Multi-Link Scheduling with Restricted Target Wake Time in Wi-Fi 7

Doğanalp Ergenç, Mateusz Zakrzewski, Falko Dressler
Telecommunication Networks, TU Berlin, Germany
ergenc@ccs-labs.org, m.zakrzewski@campus.tu-berlin.de, dressler@ccs-labs.org

***Abstract*—Demand for time-sensitive, resilient wireless networking is rapidly growing for critical systems and applications. Addressing this demand, Wi-Fi 7 introduces new features, including multi-link operation (MLO) and restricted target wake time (R-TWT). While MLO enables simultaneous use of multiple channels across 2.4, 5, and 6 GHz bands for increased reliability and spectral efficiency, R-TWT helps coordinate Wi-Fi 7 networks for timely channel access and improved energy efficiency. Although individual benefits of these features have been demonstrated in the literature, their combined use remains largely unexplored. In this paper, we propose scheduling heuristics to coordinate R-TWT service periods across multiple links in MLO devices. Moreover, we present an open-source multi-link R-TWT simulation framework implemented in OMNeT++. The experiments in this framework demonstrate that our heuristics achieve predictable latency and low jitter in various scenarios, and significantly reduce energy consumption compared to the non-scheduled approach.**

***Index Terms*—Wi-Fi, MLO, R-TWT, time-sensitive, scheduling**

I. Introduction

Emerging critical wireless applications increasingly demand resilient low-latency communication. Several wireless networking technologies, such as the upcoming 6G, DECT, and Wi-Fi standards, introduce new features to address these requirements. IEEE 802.11be (Wi-Fi 7), for example, introduces multi-link operation (MLO) to utilize the 2.4, 5 and 6 GHz bands simultaneously over multiple radio interfaces, maximizing throughput and enhancing reliability under varying link conditions. Another prominent feature, restricted target wake time (R-TWT), extends the energy-saving mechanism target wake time (TWT) from Wi-Fi 6 by adding protected service periods (SPs) for scheduled communication. This enables stations to have dedicated, contention-free transmission windows during which they are *awake*, while remaining in a *doze* state at other times to improve energy efficiency. Such mechanisms are particularly beneficial for time-sensitive applications.

Although the individual benefits of MLO and R-TWT have been investigated in the literature [1], [2], [3], their joint operation and design principles have not yet been widely explored. Here, MLO offers increased scheduling opportunities for R-TWT service periods across multiple links, providing greater reliability under varying channel congestion or link quality conditions [4]. Figure 1 illustrates an example scenario with six stations (STAs) employing multi-link R-TWT over the 2.4, 5 and 6 GHz bands (L_1, L_2, and L_3, respectively). In the figure, green interfaces of the STAs are awake, while

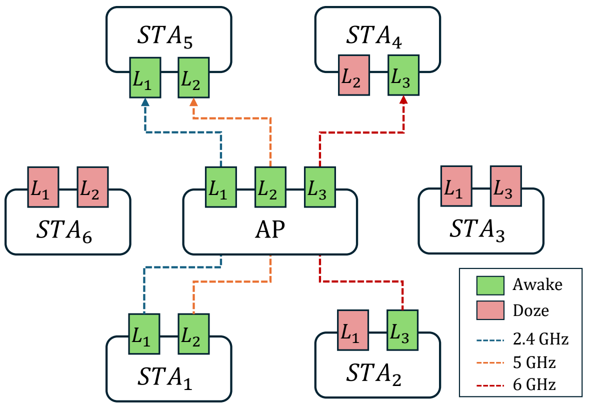


Fig. 1: An example scenario for multi-link R-TWT.

red ones are in the doze state during a particular SP. STA_1 communicates with STA_5 over L_1 and L_2 simultaneously, whereas STA_2 and STA_5 are connected over L_3. All other STAs remain in the doze mode and wake up in subsequent SPs. The access point (AP) stays awake to relay data traffic for links connecting awake (interfaces of) STAs. The SPs are scheduled by the AP for the given requirements, such as traffic load and latency tolerance, acting as a centralized scheduler.

Alongside its benefits, such scheduling requires a careful consideration of heterogeneous link conditions, such as varying channel capacities and congestion levels across multiple links. Moreover, the co-existence of MLO and R-TWT functionalities necessitates an extended design of Wi-Fi network stack. Addressing these challenges, this paper investigates the benefits of multi-link R-TWT scheduling, combining MLO and R-TWT. We extend our previous MLO implementation [5] in the OMNeT++ simulator with R-TWT functionality to evaluate scheduling heuristics under diverse network settings.

Our main contributions can be summarized as follows:

- The design of a combined MLO and R-TWT stack, along with its open-source implementation for the OMNeT++ simulator[1] (Section IV).
- A set of scheduling heuristics for allocating R-TWT service periods over asymmetrical MLO links (Section V).
- An extensive evaluation and parameter study demonstrating latency, jitter, and energy-efficiency improvements achieved by these heuristics (Section VI).

[1]The implementation is available at https://github.com/tkn-tub/wifi-rtwt-mlo-omnet.

ISBN: 978-3-903176-79-9

II. Related Work

The key features in this work, MLO and R-TWT, have been investigated in the literature to optimize Wi-Fi networks for various metrics. Several MLO studies focus on aspects such as traffic allocation methods over multiple links to improve throughput [6] or reduce latency [2]. López-Raventós and Bellalta [1], for instance, propose different heuristics to find the optimal link-switching method that satisfies the given throughput requirements. While this work reveals the benefits of flexible link selection enabled by MLO, another study in [2] investigates its limitations under different operational modes (see Section III-A) and traffic conditions. The results show that latency improvements vary significantly depending on load asymmetry over different links.

Achieving low latency and low energy consumption are two common optimization goals in TWT and R-TWT scheduling. For instance, Nurchis and Bellalta [3] evaluate single-link TWT scheduling in a setup where STAs are grouped into two or four TWT sessions, with each group waking up every 20 ms. They show that as traffic demand increases, scheduled traffic experiences lower latency than the conventional contention-based approach. Mozaffariahrar et al. [7] quantify how SP length influences power consumption. For a 100 ms wake interval, for instance, a 5120 µs SP allows an STA to spend approximately 94% of the cycle in the sleep state, significantly reducing its energy consumption. Notably, they also present the first open-source implementation of TWT for the ns-3 network simulator. In another study, Haxhibeqiri et al. [8] implement R-TWT on an SDR platform, demonstrating the effectiveness of precise time synchronization and scheduling for time-sensitive communication. They show that, within a given R-TWT schedule, mixed-criticality frames can be delivered with bounded latency. Apart from such experimental works, Belogaev et al. [9] and Bankov et al. [10] introduce analytical models to estimate packet delay distribution and throughput in R-TWT settings, respectively.

Finally, limited works investigate the co-existence of MLO and R-TWT. Shafin et al. [11] propose allowing aligned TWT SPs across multiple links to increase the channel access probability for critical traffic. However, they do not conduct any further evaluation beyond a preliminary analysis. Jin et al. [12] jointly optimize TWT and MLO by balancing the trade-offs between energy consumption and transmission delay. Their link scheduling strategy minimizes energy consumption compared to another MLO heuristic, highlighting the benefits of the multi-link TWT mechanism.

Compared to the majority of works studying either MLO or (R-)TWT, we aim to demonstrate *their joint* benefits. Beyond the throughput and energy-efficiency improvements with which these features are mainly associated, we propose a multi-link R-TWT scheduler to achieve predictable latency, concerning asymmetrical link and traffic conditions. Lastly, we present our implementation in the OMNeT++ simulator as the first open-source framework for the combined MLO and R-TWT features. Table I summarizes these novelties.

TABLE I: Comparison of related work. "∼" represents a partial or no availability.

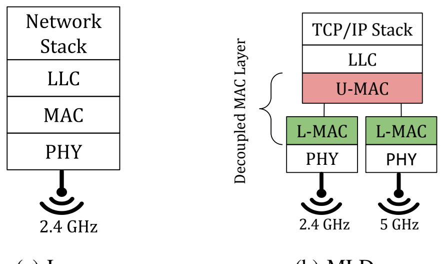
Work	**Features**		**Evaluation**	**(Open-source) Implementation**
	MLO	(R-)TWT		
[1]	✓	✗	Throughput	✗
[2]	✓	✗	Latency	✗
[3]	✗	✓	Latency	✗
[7]	✗	✓	Energy	∼
[8]	✗	✓	Latency, throughput	∼
[9]	✗	✓	Latency	✗
[10]	✗	✓	Throughput	✗
[11]	✓	✓	✗	✗
[12]	✓	✓	Energy	✗
This work	✓	✓	Latency, energy	✓

III. Fundamentals

A. Multi-link Operation

In the IEEE 802.11be standard, multi-link devices (MLDs) utilize the 2.4, 5 and 6 GHz frequency bands over multiple interfaces simultaneously. In an MLD, the medium access control (MAC) layer is decoupled into an upper MAC (U-MAC) and lower MAC (L-MAC), as shown in Figure 2. The U-MAC acts as a coordinator for multiple links. It buffers packets from higher layers and performs aggregation and distribution across interfaces via packet or flow assignment strategies [13]. The U-MAC also manages frame sequencing, link association and authentication, and other multi-link control procedures. For this coordination, it operates in tandem with the management modules of the associated interfaces.

In contrast, each L-MAC is instantiated per radio interface. It handles standard link-layer functions, including channel access based on individual channel parameters and constructing and verifying MAC headers. In current Wi-Fi, the primary contention-based channel access mechanism is enhanced distributed channel access (EDCA), an extension of hybrid coordination function (HCF). It follows a listen-before-talk approach while adding traffic prioritization through different access categories (ACs). EDCA maintains a EDCA functions (EDCAFs) for each AC with distinct parameters. In particular, the EDCAF with smaller Contention Window (CW) size benefits from a faster channel access. To improve channel efficiency, IEEE 802.11e also introduces the Block ACK (BA), where after a predefined number of packets are sent, the receiver sends a BA for all received frames at once.



(a) Legacy (b) MLD

Fig. 2: Wi-Fi stack in legacy STAs and MLDs.

MLO also introduces unique challenges. MLDs with insufficiently isolated interfaces may experience internal interference during simultaneous transmission and reception. To mitigate this, the standard defines different operational modes. In this work, we employ the Simultaneous Transmit and Receive (STR) mode, assuming independent channel access on each link to enable concurrent transmission and reception.

B. *Target Wake Time*

TWT improves the energy efficiency of STAs by putting radios to sleep, i.e., doze mode, when not in use. TWT assigns SPs to STAs during which they wake up to transmit or receive, and sleep otherwise. There are two types of TWT agreements: individual and broadcast. In the former, when an STA wants to use the TWT, it sends a request to the AP, with a desired SP in terms of wake time and duration. The AP checks its suitability, e.g., comparing with the SPs of other STAs, and responds with the agreed TWT values back. Then, the STA is only required to wake up for the negotiated SP. In case of the broadcast agreement, the AP advertises available SPs with beacons and each STA requests to join one of them. The key parameters for an SP are the target wake time (the absolute time for the first wake-up), SP cycle length or wake interval (the time between consecutive SPs), and wake duration (the time the STA remains awake during each SP). The cycle length must lie between 512 µs and 65 536 µs [8], while the minimal wake duration is 256 µs [3].

C. *Restricted Target Wake Time*

In TWT, STAs wake up during their negotiated SPs but are not prohibited to access the medium at other times. This approach provides energy savings for STAs that use TWT, yet the channel remains non-exclusive to the members of a particular TWT SP. In contrast, R-TWT was introduced in IEEE 802.11be for exclusive channel access. It extends TWT by dividing airtime into non-overlapping SPs, ensuring that only the designated STAs can wake and access the medium during each period. To achieve this, R-TWT operates under centralized AP control, with the AP distributing a global schedule. That is, AP coordinates the SP agreements so that STAs not assigned to the current SP refrain from medium access (typically by remaining in doze state), preventing collisions. Accordingly, this mechanism enables collision-free medium access, supporting time-sensitive applications.

R-TWT inherits the negotiation structure of TWT described above. In our implementation, we do not employ the full standard-compliant message structure, but rather a simplified TWT agreement scheme that exchanges only the subset of information required for transmission scheduling.

IV. System Design

For our multi-link R-TWT framework, we consider a network with one AP and multiple pairs of STAs MLDs, as illustrated in Figure 1. Each pair consists of a sender STA and a receiver STA for a specific traffic flow. The AP coordinates R-TWT scheduling over multiple links through these phases:

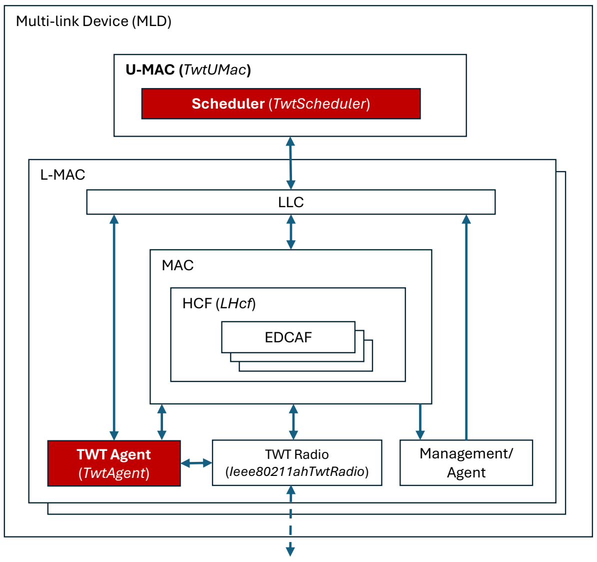


Fig. 3: MLD architecture with integrated R-TWT.

1) **Bootstrapping:** Senders transmit their link requests, including flow requirements and associated MLO links, during an initial period.
2) **Scheduling:** After collecting all requests, the AP computes non-overlapping SPs for each pair of STAs, indicating awake and doze times for both senders and receivers at multiple links.
3) **Offloading:** The AP distributes the computed schedules to STAs, including a broadcast SP for beaconing.
4) **Communication:** STAs configure their radios to wake at their assigned SPs per link for transmissions and receptions. The AP relays frames only to awake receivers during their dedicated SPs.

Executing these steps requires several extensions to the U-MAC and L-MAC at MLDs. In this section, we describe these extensions along with the relevant implementation details of our open-source framework in the OMNeT++. A view of the extended Wi-Fi stack is also shown in Figure 3. The most significant components are highlighted in color with bold text, and the relevant simulation modules are shown in italics.

A. *U-MAC Extensions*

To enable R-TWT over MLO, the U-MAC manages link associations and TWT agreements, and computes schedules accordingly. In our design, we extend the existing U-MAC implementation in [5] (`TwtUMac` in Figure 3) and introduce a new TWT scheduler module (`TwtScheduler`) for the AP. The following sections describe these extensions, excluding the scheduler details, which are presented in Section V.

1) Link association and R-TWT agreement: Initially, the AP tracks Wi-Fi association frames to discover STAs and their associated links (bootstrapping phase). All management frames originally received by the L-MAC management module are forwarded to the U-MAC for this purpose. After association, an STA's U-MAC initiates R-TWT negotiation by sending a *link requirement request* to the AP, based on flow requirements specified in the configuration file. This request includes flow data rate, wake time preference, and destination MAC address (and other fields). Here, the flow data rate should be derived from the application deployed at the STA or manually configured. Negotiation occurs over the 2.4 GHz link, designated as the control channel for R-TWT, but could be carried over any link(s) specified by the AP.

The AP forwards collected link requirement requests to its scheduling module. Once all requests are collected, the scheduler computes a conflict-free schedule consisting of transmission and reception time slots (scheduling phase) and responds with a *TWT response* for each request (offloading phase). Scheduled SPs also include dedicated time slots for beaconing, during which all STAs are awake to receive beacons and other control packets for normal Wi-Fi operations. Each response includes: source MAC address, a flow identifier, link identifier for multi-link scheduling, wake time and duration of the SP, and and schedule cycle interval.

Upon receiving a TWT response, the STA's U-MAC configures the L-MACs of the associated links. Each link then follows its SP settings, configuring radios for the specified wake and sleep times (communication phase).

2) Queue management: After R-TWT scheduling, the wake times of a flow's sender and receiver may not necessarily overlap. In such cases, the AP buffers received frames until the corresponding receiver interfaces are awake. To support this, we modify the AP's packet buffer into multiple logical queues, one for each receiver STA. When the U-MAC acknowledges an awake link of a receiver, the L-MAC pulls the next frame from that receiver's logical queue at the U-MAC. This ensures frames are transmitted in the correct order and on time with respect to the scheduled SPs.

B. L-MAC Extensions

L-MAC originally comprises LLC, MAC, management, and radio modules in the design of [5]. We introduce a new TWT agent module (`TwtAgent`) to handle wake and doze operations according to SP schedules, and modify the channel access and radio modules to adapt frame transmissions.

1) TWT Agent: This module maintains the wake and sleep schedule for its corresponding interface and initiates transmissions via internal self-messaging of OMNeT++. The agent schedules these messages to notify the MAC and radio modules of the start and end of each SP. Upon receiving the signals, the MAC pulls buffered packets from the U-MAC and attempts channel access, while the radio switches between sleep and transmission/reception states as needed. This separates the scheduling logic from the other modules, simplifying the implementation.

2) Channel Access: The MAC module uses HCF (`LHcf`), which employs EDCA through multiple EDCAFs (see Section III-A), for channel access. Channel access primarily depends on two trigger signals at the L-MAC: one from the U-MAC indicating a buffered frame, and another from the TWT Agent indicating wake mode. If both are present, HCF pulls a packet from the corresponding receiver STA's U-MAC

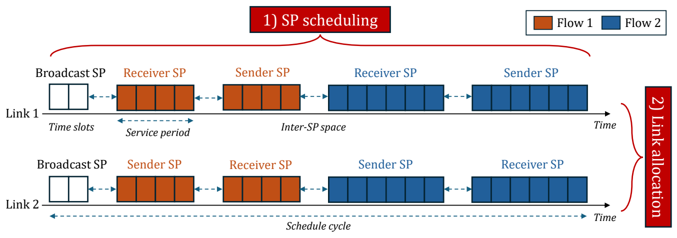


Fig. 4: Per-flow multi-link SP scheduling.

buffer and enqueues it in the appropriate EDCAF based on the frame's AC. When the channel is idle, the packet is transmitted following EDCA with two particular settings. First, since the scheduler guarantees non-overlapping SPs, we set a minimal CW size for EDCAFs for faster channel access. Second, we enable BAs to reduce control overhead.

3) TWT Radio: Finally, this modules extends the standard Wi-Fi radio implementation in OMNeT++ with dynamic wake and sleep capabilities to support R-TWT SPs. The TWT Radio (`Ieee80211ahTwtRadio`) manages power states based on signals from the TWT Agent, waking or sleeping whenever a corresponding signal is received. During the transition from awake to sleep, a short offset time is used to avoid interfering with ongoing transmissions. This module leverages OMNeT++'s built-in state-based energy consumption framework, consuming 1 mW in sleep, 2 mW in idle, 5 mW while listening, 10 mW while receiving, and 100 mW while transmitting. The experiment results in this paper are based on this model.

V. Multi-link R-TWT Scheduler

The multi-link R-TWT scheduler (`TwtScheduler`) computes dedicated SPs for given traffic flows based on varying link characteristics. It performs two functions (see Figure 4):

1) **SP scheduling:** Calculates distinct time slots, i.e., a SP, for the sender and receiver of each flow within a schedule cycle, according to the flow requirements.
2) **Link allocation:** Assigns the calculated time slots over the selected links between sender and receiver to minimize latency and prevent queuing overflow at the AP.

Note that our scheduling heuristics assume a homogeneous network of identical MLDs with two asymmetrical links at 2.4 and 5 GHz. However, their main principles can easily be generalized for more than two links.

A. SP Scheduling

In scheduling, a schedule cycle S defines a set of ordered SPs α^f for each flow $f \in F$. This cycle is divided into time slots of fixed length $T = 256\,\mu s$, corresponding to the minimum TWT wake duration defined in the standard. Accordingly, during SP scheduling, for each flow $f \in F$, the scheduler determines the number of *consecutive* time slots that form a dedicated R-TWT SP for both the sender and the receiver of that flow. The receiver's SP is then used by the AP to relay the frames buffered during the sender's SP. Here, the receiver should remain awake for only the minimum duration needed to receive all frames, while still maximizing its energy savings. The scheduler first computes the number of free time slots t_{free} within a schedule cycle, excluding two dedicated beaconing slots (see Section IV-A) and the inter-SP gaps. The latter is a buffer period between consecutive SPs to prevent overlapping transmissions and is fixed to two time slots as a design parameter. Accordingly, t_{free} is calculated as:

$$t_{\text{free}} = (S_l/T) - (|F| \cdot 2 \cdot 2) - 2 \tag{1}$$

where S_l is the length of the schedule cycle, and $|F|$ is the number of flows. Each flow typically requires two SPs (both the sender and the receiver), thus also necessitating two inter-SP gaps per link, except for specific cases discussed in the next section. Note that F is known to the scheduler after the bootstrapping phase (see Section IV).

Next, the scheduler follows a link-agnostic approach to assign the available time slots t_{free} to the SPs. During the bootstrapping phase, the scheduler learns the associated links of all STAs, the total channel capacity C across multiple links in terms of achievable throughput in bitrate, and the requested data rate f_d for each flow. The achievable throughput can be approximated using (i) recent transmission statistics, (2) PHY layer feedback including MCS and RSSI values, and (3) frame success/retry ratios. Accordingly, it first computes the achievable *goodput* G as the available link rate for data transmission after deducting control overhead:

$$G = C \cdot \frac{t_{\text{free}}}{S_l/T} \tag{2}$$

Then, the length of an SP (i.e., the number of required time slots) for a flow f is calculated as:

$$\alpha^f = \left\lceil \frac{f_d}{G} \right\rceil \tag{3}$$

Given these formulas, we employ a max-min fairness heuristic to assign SPs for all R-TWT requests, following the description in [14]. The algorithm is summarized in Algorithm 1. It performs the assignments starting from the flow with the lowest data rate and proceeding to the highest. This ensures that high-demand flows do not initially occupy the schedule cycle, thereby maximizing the number of flows that can obtain an SP. Iterating over the ordered set $F^* = \{f^1, f^2, \ldots, f^n \mid f_d^i \leq f_d^{i+1}\}$, the scheduler computes α^f and compares it with the *fair share* t_{fair}. The fair share represents the ratio of the remaining available time slots to $|F|$, limiting the SP length per flow to avoid rapidly assigning all available resources. If the required number of slots for f is smaller than t_{fair}, a SP of length α^f is assigned to the sender and receiver of f (line 3). Otherwise, they are assigned a shorter but *fair* SP, $\alpha^f = t_{\text{fair}}$ (line 5). In both cases, t_{fair} is recalculated based on the remaining available slots. Finally, the algorithm outputs α^*, the list of computed SP lengths for all flows.

Algorithm 1 Max-min fairness scheduling.

```
Require: $F^*$ ordered flows, $G$ goodput, $t_{\text{free}}$ time slots in SP
 1: $t_{\text{fair}} \leftarrow t_{\text{free}} \;/\; |F^*|$
 2: for $i \leftarrow 0$ to $|F|$ do
 3:     $\alpha^f \leftarrow \lceil \frac{f_d}{G} \rceil$ {Equation (3)}
 4:     if $\alpha^i > t_{\text{fair}}$ then
 5:         $\alpha^i \leftarrow t_{\text{fair}}$
 6:     end if
 7:     $t_{\text{free}} \leftarrow t_{\text{free}} - \alpha^i$
 8:     $t_{\text{fair}} \leftarrow t_{\text{free}} \;/\; (\; |F^*| - i \;)$
 9: end for
10: return $\alpha^* = \{\alpha^f \mid \forall f \in F\}$
```

B. Link Allocation

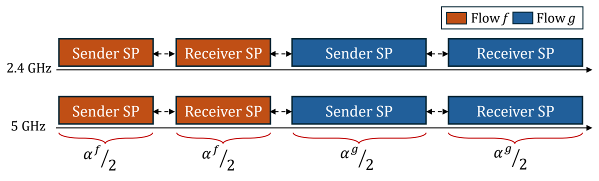
The second function of the scheduler is to allocate the calculated SPs in α^* to multiple links. It must guarantee that both the sender and receiver of a flow have their SPs in an effective temporal order such that the AP can relay the sender's transmissions to the receiver quickly without depleting its buffers (see Section IV-A) or delaying packets to the next cycle. For this, we propose three link allocation heuristics as illustrated in Figure 5 and summarized below:

- **Symmetrical allocation** distributes the SP length of a flow equally across two links. On each link, the sender is scheduled first and the receiver is scheduled afterward.
- **Asymmetrical allocation** modifies the calculated SP schedule for the sender and receiver of a flow to exploit asymmetrical link characteristics, and parallelizes transmission and reception over multiple links.
- **Cross-symmetrical allocation** is similar to the symmetrical but assigns sender and receiver SPs across links in a crossed manner to prevent long packet queues at the AP.

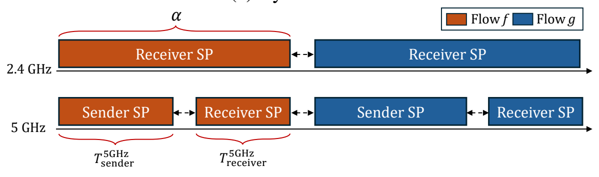
Each heuristic has particular advantages and disadvantages. Symmetrical allocation is the simplest but the least efficient for AP buffer management and is therefore prone to queuing delay. Asymmetrical allocation trades additional complexity for improved performance, since it requires adjustments over the calculated SPs. Cross-symmetrical allocation offers the best trade-off for two links but can become complex to configure for networks with more links. We discuss these aspects in detail in the following.

1) Symmetrical: As illustrated in Figure 5a, this heuristic allocates the half of the calculated SP of a flow f to the sender, i.e., $\alpha^f/2$, at both 2.4 and 5 GHz links in parallel. A subsequent SP of $\alpha^f/2$ is then allocated to the receiver so that it can immediately retrieve frames the buffered at the AP in the previous SP (of the sender). Despite its simplicity, the symmetrical link allocation causes the frames received from both links accumulating at the AP for $\alpha^f/2$ duration. Although it may not exhaust the computational resources of the AP, i.e., a limited queue and buffer size, as only one sender transmits at a given time, an increased queue occupation can impact end-to-end latency and jitter as discussed in our evaluation.

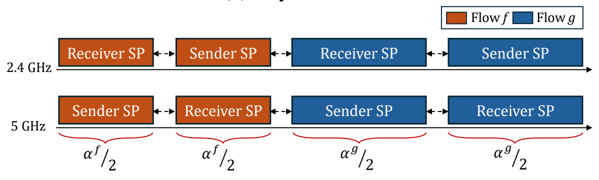
2) Asymmetrical: In the symmetrical allocation, the time slots of a flow's SPs are distributed equally across links, regardless of their asymmetrical conditions. However, when one link is faster than the other, e.g., the 5 GHz link typically providing a higher bitrate than the 2.4 GHz, the sender transmits most frames over the faster link using fewer time slots than the allocated SP length. In parallel, the AP relay these



(a) Symmetrical



(b) Asymmetrical



(c) Cross-symmetrical

Fig. 5: Link allocation heuristics.

frames through the slower link within the receiver's SP, which may not sustain the data rate transmitted over the faster one.

To address this, in asymmetrical allocation, the AP *switches* to the faster link to deliver the remaining frames within the same SP. This requires splitting the calculated SP on the faster link into two parts: one for the sender and another for the receiver, as illustrated in Figure 5b. In the figure, $T^{\text{5GHz}}_{\text{sender}}$ denotes the portion of the SP α on the 5 GHz link used by the sender to transmit all frames. The AP then relays these frames during (i) the entire α period on the 2.4 GHz link and (ii) an additional $T^{\text{5GHz}}_{\text{receiver}}$ duration on the 5 GHz link. To ensure the transmitted and received data volumes are equal, we formulate:

$$T^{\text{5GHz}}_{\text{sender}} \cdot R^{\text{5GHz}} = \alpha \cdot R^{\text{2.4GHz}} + T^{\text{5GHz}}_{\text{receiver}} \cdot R^{\text{5GHz}} \tag{4}$$

where R^{5GHz} and $R^{\text{2.4GHz}}$ denote the bitrates of the 5 GHz and 2.4 GHz links, respectively. When considering that the overall SP should be equal to the sum of the split time slots for the sender and the receiver at the 5 GHz link, including the two time slots for inter-SP spacing, we have:

$$\alpha = T^{\text{5GHz}}_{\text{sender}} + T^{\text{5GHz}}_{\text{receiver}} + 2 \cdot 256\,\mu\text{s} \tag{5}$$

The split duration of the SP can then be calculated as:

$$T^{\text{5GHz}}_{\text{sender}} = \frac{\alpha \cdot (R^{\text{5GHz}} + R^{\text{2.4GHz}}) - 2 \cdot 256\,\mu\text{s} \cdot R^{\text{5GHz}}}{2 \cdot R^{\text{5GHz}}} \tag{6}$$

Finally, the asymmetrical link allocation uses Equation (6) to split the SP on the faster link, while allocating the entire SP to the receiver at the slower link. Note that this also saves an inter-SP buffer time on the latter.

3) Cross-symmetrical: This heuristic combines the benefits of the symmetrical and asymmetrical allocation approaches. Similar to the symmetrical one, it first splits the calculated SP equally between the sender and receiver of the respective flow. It then allocates these partial SPs for parallel transmission and reception, as illustrated in Figure 5c, resembling the asymmetrical allocation. Although this method performs a simpler multi-link allocation without explicitly considering link characteristics, it still prevents frame accumulation in the AP's buffer via simultaneous transmission and reception.

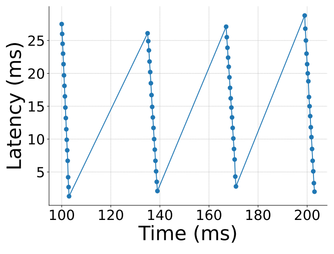
VI. Evaluation

For our experiments, we use OMNeT++ v6.1 with INET v4.5.4, in which we extend with our multi-link R-TWT framework. The simulation parameters are summarized in Table II. In the experiments, the network consists of one AP and 8–16 uniformly distributed STAs in a 50 m × 50 m area. They are grouped into sender–receiver pairs as described in the previous sections. All STAs are MLO-capable, operating over two links at 2.4 and 5 GHz, each with varying bitrates. For traffic generation, we use two models, uniform and non-uniform, under congestion and non-congestion settings. In the uniform model, all senders transmit UDP traffic at identical data rates between 2 and 16 Mbit/s, depending on the network size. In the non-congestion setting, the total offered load remains below the achievable network throughput. In the congestion setting, the load increases up to the network capacity. In the non-uniform model, one flow generates a higher load than the others, acquiring more channel time after contention. Each experiment is repeated 10 times for 10 s of simulated time.

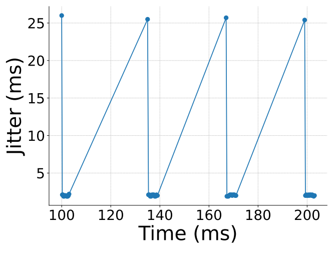
As one of the main design parameters of the scheduler, we investigate the impact of the schedule cycle length. The minimal cycle length for n STAs is $n\cdot$ 2048 µs, corresponding to eight time slots. Based on our observations, this is the minimum value at which all heuristics can effectively allocate SPs across the links. Finally, we evaluate the worst-case latency and jitter between sender and receiver STAs, and energy consumption for our proposed methods. While the first two metrics represent predictable latency, the last one highlights the energy-saving benefits of R-TWT. The latency is measured between the sender's and receiver's U-MACs, capturing both transmission and queuing delays. Before the results, we first present a validation scenario to demonstrate the basic operation of the implemented R-TWT schedulers.

TABLE II: Simulation parameters.

Parameter	**Value**
# STAs	4-32 (in different scenarios)
Area	50 m × 50 m
# Flows ($\lvert F \rvert$)	2-16
Flow rate (f_d)	2-16 Mbit/s UDP traffic
Packet size	1000 B
#MLO links	2 (2.4 GHz, 5 GHz)
Bitrate	52 Mbit/s (2.4 GHz), 130 Mbit/s (5 GHz)
Bandwidth	20 and 40 MHz
CW w/o scheduling	$\text{CW}_{\text{min}} = 15, \text{CW}_{\text{max}} = 1023$
CW with scheduling	$\text{CW}_{\text{min}} = 0, \text{CW}_{\text{max}} = 3$
Cycle durations	16 384, 32 768 and 65 536 µs
Duration	10 s
Repetition	10

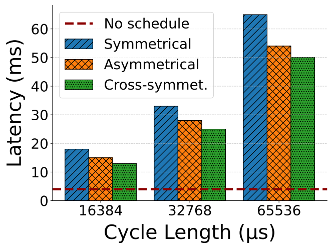


(a) Latency

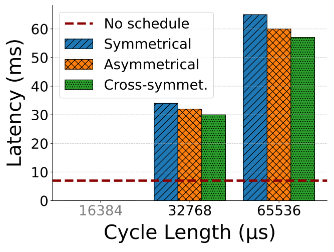


(b) Jitter

Fig. 6: Latency and jitter timeline for a schedule cycle of 32 768 µs.



(a) 8 STAs.



(b) 16 STAs.

Fig. 7: 99th percentile latency vs. number of STAs and cycle length in *uniform and non-congested* traffic scenarios.

A. Validation

For validation, we simulate a network with eight STAs and uniform traffic. The asymmetrical link allocation is employed with a schedule cycle of 32 768 µs. Figure 6 shows the latency and jitter of the consecutive frames of a flow over 200 ms.

In both figures, we observe a clear pattern of wake and doze periods corresponding to the realization of SPs. Figure 6a reflects the queueing behavior, where packets are buffered at the STA during doze periods and transmitted during the SP. The maximum latency of approximately 30 ms is bounded by the cycle duration, demonstrating that the scheduler provides a predictable latency. In Figure 6b, the jitter spikes at the beginning of each SP result from the latency difference between the last packet of one SP and the first packet of the next. Then, it remains stable due to the contention-free access.

B. Worst-case Latency

We measure the 99th percentile latency for our heuristics with different schedule cycle lengths to represent worst-case latency. We also compare them with a *non-scheduled scenario*, where all MLDs perform independent EDCA at each link and the frames are equally distributed over the links. Figure 7 shows the experiment results for *uniform and non-congested* traffic scenarios at different network sizes of 8 and 16 STAs.

In Figure 7a, the worst-case latency for all schedulers is higher than the non-scheduled scenario, which is indicated by the red, dashed line. Expectedly, when the network is not congested, STAs can easily content for the channel and waiting for SPs becomes an overhead. From left to right, the latency increases proportionally with the cycle length for all types of schedulers. The reason is that, in the worst case, a packet generated right an SP must wait for the next cycle. Nevertheless, all worst-case latency measurements remain below the cycle length and shows the predictably achieved by the scheduling. The results are also similar for 16 STAs as shown in Figure 7b. Here, we could not use the 16 384 µs cycle length as it is too short to be divided among 16 STAs.

TABLE III: 99th percentile latency under non-uniform traffic.

Cycle (µs)	Scheduler	Latency (ms)
-	No schedule	2110 ± 470.4
16384	Symmetrical	16.98 ± 0.01
	Asymmetrical	15.23 ± 0.01
	Cross-symmetrical	14.16 ± 0.01
32768	Symmetrical	33.17 ± 0.01
	Asymmetrical	30.09 ± 0.02
	Cross-symmetrical	28.19 ± 0.02
65536	Symmetrical	65.67 ± 0.01
	Asymmetrical	58.97 ± 0.04
	Cross-symmetrical	56.76 ± 0.02

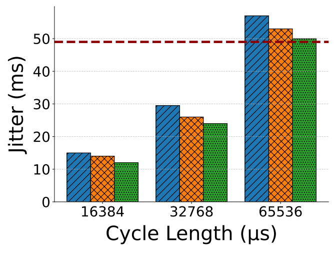
The latency differences between the link allocation heuristics are also notable. In Figure 7, the symmetrical allocation (blue, diagonal hatched) results in the highest latency due to queueing delays: while the sender STA transmits on both links simultaneously, the AP must buffer the received frames until the receiver's SP. The asymmetrical allocation performs better by parallelizing transmission at the sender and reception at the receiver. Finally, the cross-symmetrical allocation achieves up to 41 % and 13 % lower latency than the symmetrical and asymmetrical schedulers, respectively, as it enables simultaneous transmission and reception over both links.

We also measure the worst-case latency under the non-uniform traffic scenario to examine whether our scheduler can ensure fair transmission opportunities for all STAs in the presence of high-demand flows. Table III summarizes these results for a network with eight STAs. In non-scheduled scenarios, the worst-case latency reaches up to 2110 ms with a substantial deviation. This indicates that some STAs and potentially the AP experience long queueing delays, as the heavy-load flow dominates the medium. In contrast, all schedulers effectively bound the latency within the given cycle lengths, ensuring predictable performance even in complex traffic conditions. These results also confirm the effectiveness of max–min fairness strategy in SP scheduling (see Section V-A).

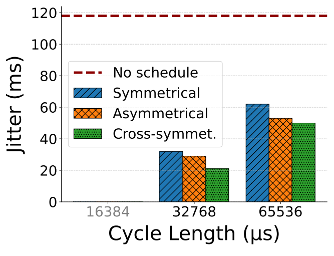
C. Jitter

Scheduled channel access enhances network stability, characterized by small and stable jitter. Accordingly, Figure 8 presents the 99th-percentile jitter for the *uniform and congested* traffic, following a similar depiction as the previous results.

For both 8 and 16 STAs, all schedulers perform much better than the non-scheduled scenario (red, dashed line), eliminating latency variation caused by random channel access. As seen in Figure 8b, the difference is more pronounced for 16 nodes, where channel contention is higher. For every cycle length and network size, the cross-symmetrical allocation achieves the lowest jitter. Since both the sender and receiver of a flow remain active throughout the entire SP, queueing delay is reduced with this heuristic. In contrast, the symmetrical allocation dozes the sender in the middle of an SP, causing frames to



(a) 8 STAs.



(b) 16 STA.

Fig. 8: 99th percentile jitter vs. number of STAs and cycle length in *uniform and congested* traffic scenarios.

buffer for a longer period. Lastly, the asymmetrical allocation performs in between, as it allows the sender to transmit new packets for a longer duration than the symmetrical approach but still does not utilize the entire SP.

The results also indicate that longer cycle lengths increase jitter, due to the extended waiting time for the first packet in the subsequent SP (see Section VI-A). Although a shorter cycle length is preferable for this reason, it makes scheduling SPs more difficult as the number of STAs increases.

D. Energy Efficiency

Finally, Figure 9 shows the average energy-consumption per packet at the senders (blue, dot) and receiver (orange, square) STAs for various number of STAs, under uniform traffic model. We compare the results only for the asymmetrical link allocation (straight) and non-scheduled (dashed) scenarios, as all schedulers perform very similarly. For the measurements, we use the power consumption model in Section IV-B.

Expectedly, multi-link R-TWT scheduling significantly reduces energy consumption for both sending and receiving STAs, as they remain in sleep mode outside their dedicated SPs. Increasing the number of STAs further improves energy savings, since each STA is assigned shorter SPs and longer dozing periods, at a potential cost of reduced throughput. Notably, in non-scheduled scenarios, energy consumption tends to increase after 16 STAs due to longer contention times. The repeated experiments also show consistent results.

VII. Conclusion

In this paper, we combined two prominent features of Wi-Fi 7, MLO and R-TWT, to achieve predictable latency and energy efficiency in wireless networks. We proposed a multi-link R-TWT scheduler that allocates dedicated service periods across multiple parallel links for Wi-Fi STAs, ensuring guaranteed channel access time. The scheduler employs different heuristics to enable simultaneous transmission and reception, thereby reducing queueing delays. We also presented the details of our open-source multi-link R-TWT framework implemented in OMNeT++. The experiments conducted with this framework show that the proposed heuristics effectively bound the worst-case latency and jitter depending on the schedule cycle length. Furthermore, we demonstrated the energy-efficiency benefits of R-TWT scheduling compared to non-scheduled scenarios. For future work, we plan to enhance our scheduler to support different traffic classes and MLO with any number of links.

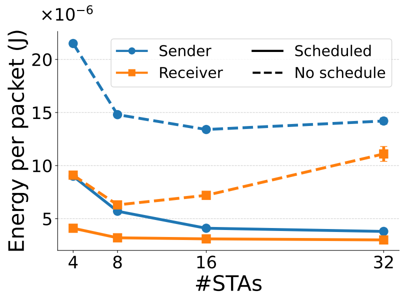


Fig. 9: Energy consumption per packet vs. increasing STA count.

Acknowledgment

This work was supported by the German Research Foundation (DFG) within the projects RESERVE with the grant number ER 1166/1-1 & DR 639/37-1 and ML4WIFI with the grant number DR 639/28-1.

References

[1] Á. López-Raventós and B. Bellalta, "IEEE 802.11be Multi-Link Operation: When the Best Could Be to Use Only a Single Interface," in *19th Mediterranean Communication and Computer Networking Conference (MedComNet 2021)*, Ibiza, Spain, Jun. 2021.

[2] M. Carrascosa, G. Geraci, E. W. Knightly, and B. Bellalta, "An Experimental Study of Latency for IEEE 802.11be Multi-link Operation," in *IEEE International Conference on Communications (ICC 2022)*, Seoul, South Korea: IEEE, May 2022.

[3] M. Nurchis and B. Bellalta, "Target Wake Time: Scheduled Access in IEEE 802.11ax WLANs," *IEEE Transactions on Wireless Communications*, vol. 26, no. 2, pp. 142–150, 2019.

[4] S. K. Venkateswaran, C.-L. Tai, A. Ahmed, and R. Sivakumar, "Target Wake Time in IEEE 802.11 WLANs: Survey, Challenges, and Opportunities," *Elsevier Computer Communications*, vol. 236, p. 108 127, 2025.

[5] D. Ergenç and F. Dressler, "An Open Source Implementation of Wi-Fi 7 Multi-Link Operation in OMNeT++," in *20th IEEE/IFIP Wireless On-demand Network systems and Services Conference (WONS 2025)*, Hintertux, Austria: IEEE, Jan. 2025, pp. 131–134.

[6] B. Tan, Y. Gao, and X. Sun, "Throughput-Optimal Multi-Link Access for Wi-Fi 7 via Multi-Agent Reinforcement Learning," in *IEEE Wireless Communications and Networking Conference (WCNC 2025)*, Milan, Italy: IEEE, Mar. 2025.

[7] E. Mozaffariahrar, M. Menth, and S. Avallone, "Implementation and Evaluation of IEEE 802.11ax Target Wake Time Feature in ns-3," in *Workshop on Ns-3 (WNS3 2024)*, Barcelona, Spain: ACM, Jun. 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*, vol. 62, no. 8, pp. 118–124, Aug. 2024.

[9] 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," in *IEEE Wireless Communications and Networking Conference (WCNC 2024)*, Dubai, United Arab Emirates, Apr. 2024.

[10] D. Bankov, A. Lyakhov, E. Stepanova, and E. Khorov, "Performance Evaluation of Wi-Fi 7 Networks with Restricted Target Wake Time," *Problems of Information Transmission (Probl. Inf. Transm.)*, vol. 60, no. 3, pp. 233–254, 2024.

[11] R. Shafin, B.-L. Ng, A. Ibrahim, P. Nayak, V. Raynam, S. Leng, and J. Han, "MLO: Broadcast TWT for MLDs," *Submission IEEE 802.11-21/0394r2*, Jun. 2020.

[12] X. Jin, Y. Long, X. Fang, R. He, and H. Ju, "Energy Consumption Optimization under Multi-link Target Wake Time scheme in WLANs," in *IEEE/CIC International Conference on Communications in China (ICCC 2022)*, Sanshui, Foshan, China, Aug. 2022, pp. 1119–1124.

[13] A. A. Abdalhafid, S. K. Subramaniam, Z. A. Zukarnain, and F. H. Ayob, "Multi-Link Operation in IEEE802.11be Extremely High Throughput: A Survey," *IEEE Access*, vol. 12, pp. 46 891–46 906, 2024.

[14] D. P. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed. Nashua, NH: Athena Scientific, 2021.