A Markovian Model for Multi-Station TWT Configuration with General Batch Size Distributions

Andreea-Carina Deaconu, Thomas Begin, Isabelle Guérin Lassous
Université Claude Bernard Lyon 1, ENS de Lyon, CNRS, LIP, UMR 5668, 69342, Lyon Cedex 07, France
Email: {carina.deaconu, thomas.begin, isabelle.guerin-lassous}@ens-lyon.fr

***Abstract*—We present a Markovian performance model for IEEE 802.11 WLANs using Target Wake Time (TWT) to reduce energy consumption. The model captures TWT configuration, AP buffer size, physical data rate, number of competing stations and packet arrivals (including variable-size batches), and produces closed-form expressions for packet throughput, packet loss rate, mean queue length, mean queuing delay, and an energy reduction factor. We validate the model extensively against ns-3 discrete-event simulations (with a recent TWT patch) and two alternative modeling approaches, demonstrating high accuracy across a wide range of TWT configurations, traffic patterns, and buffer sizes. Finally, we show how the model can be integrated into an algorithm to select TWT parameters that balance energy savings and QoS requirements.**

***Index Terms*—IEEE 802.11; Wi-Fi; WLAN; energy saving.**

I. Introduction

Wireless local area networks (WLANs) have become ubiquitous in modern life, seamlessly connecting numerous and diverse connected devices including laptops, smartphones, sensors, cameras, and TVs with various quality of service (QoS) requirements. WLANs generally operate in infrastructure mode, where access points (APs) provide connectivity to stations (STAs) over shared radio channels. These operations are specified by the IEEE 802.11 standard, widely known as Wi-Fi technology. In recent years, reducing the energy consumption of WLANs has emerged as a critical research priority. These efforts are part of global initiatives to reduce the greenhouse gas emissions of information technology (IT), or they simply aim to extend battery recharge intervals, enhancing end-user experience.

The IEEE 802.11 standard has specified several energy-saving mechanisms through successive amendments. The most recent of them, Target Wake Time (TWT), stands out as the most promising approach. TWT enables a set of STAs and their AP to schedule specific periods during which communications are suspended so that the STAs can safely enter into a low-power, doze state and conserve energy.

However, implementing TWT presents significant challenges. Although the 802.11 standard specifies the message format to negotiate the TWT options and parameters, IEEE does not provide algorithms to adequately select the TWT parameters values. In fact, finding an adequate TWT configuration leads to a difficult balance: it must preserve the QoS requirements of STAs while maximizing dozing opportunities. Furthermore, the suitability of a TWT parameter setting is highly context-dependent, influenced by factors such as the traffic issued by competing STAs, the physical data rate in use, and the size of buffers. In other words, a TWT configuration that performs well in one WLAN environment may be ineffective in another.

This work was funded by the French National Research Agency (ANR) under the project ANR NEMIoT (ANR-23-CE25-0006).

A. Related literature

The configuration of TWT has been the subject of several studies aimed at optimizing its configuration, particularly the duration of the service periods (SPs), during which STAs can transmit and receive frames. We provide a more detailed definition of the SP notion, central to our work, in Section II-D. In [27], the authors developed a simple yet effective TWT configuration strategy by alternating between two fixed SP durations: a long SP shared by all STAs and a short SP restricted to a single STA. Several works have explored dynamic TWT configuration strategies. For instance, [22] introduced an adaptive mechanism that adjusts SP duration based on traffic flow performance and category, though their methods lacks an explicit optimization objective. Similarly, [28] proposed a one-SP-per-STA scheduling approach where SP duration is derived from the total number of STAs and the physical data rate of the radio channel. Other studies, such as [9], [16], rely on pre-defined SP duration values selected from a discrete list. Some solutions assume orthogonal frequency-division multiple access (OFDMA) as the underlying communication scheme, leveraging this assumption to design algorithms that minimize collision probability between the STAs [6], or maximize STA throughput [11]. In multi-AP WLANs, [21] investigated coordinated TWT SP scheduling among APs to reduce overall interference.

Simulation-based evaluations have also been conducted. [26] implemented TWT in the ns-3 simulator to analyze the impact of TWT parameters on WLAN performance and energy consumption. Similarly, [25] compared multiple TWT schedulers using ns-3 simulation. [18] conducted experimentation studies to measure the variation of energy consumption in real testbeds with and without TWT.

Closer to our work, the solution of [15] introduced an analytical model for TWT to guide parameter configuration. However, their model assumes that SP durations are sufficiently long to sustain WLAN traffic, limiting its applicability

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

under high load conditions. Additionally, it only handles packet arrival in constant-sized batches, and was validated using a simplified, homemade simulator. In contrast, our work proposes a more versatile analytical performance model for TWT that (i) supports arbitrary SP durations, (ii) handles varying batch sizes, and (iii) is validated using ns-3, enhanced by the TWT implementation from [26].

B. Our contributions

In this work, we present a Markovian model to evaluate the performance of WLANs operating under TWT and to guide TWT parameter configuration. Our contributions are threefold: (i) We develop a queuing-theory-inspired Markovian model that accounts for the TWT configuration, the number of competing STAs, the AP buffer size, the physical data rate of the radio channel, and the packet arrival process, possibly as packet batches of varying sizes. The model provides closed-form expressions for key performance metrics: throughput, packet loss rate, mean queue length, mean queuing delay, and the energy reduction factor; (ii) We validate the good accuracy of our Markovian model across diverse settings (*e.g.,* diverse TWT configurations, traffic types and buffer sizes) by comparing its performance estimates against those obtained from the discrete-event simulator ns-3, enhanced with a recent patch to handle TWT [26], as well as those from two alternative modeling approaches; (iii) We demonstrate how our Markovian model can be readily integrated into an algorithm for configuring TWT parameters according to the WLAN environment and the QoS requirements of the STAs.

The remainder of this paper is organized as follows. In Section II, we revisit key aspects of WLANs and provide a concise description of TWT. We use Section III to describe the WLAN under study and the associated modeling assumptions. Section IV presents our proposed Markovian model, which evaluates the performance of STAs operating under TWT in a WLAN. In Section V, we assess the accuracy of our model by comparing its performance against those obtained from ns-3, as well as to two alternative modeling approaches. Section VI demonstrates how our performance model can guide the selection of adequate TWT configurations to achieve energy savings while meeting QoS requirements. Finally, Section VII concludes the paper.

II. Wi-Fi communications under TWT

A. CSMA/CA

In a WLAN, the radio channel is shared among the communicating devices (STAs and APs), using a listen-before-talk approach. Specifically, radio channel access follows the distributed coordination function (DCF) based on the carrier sense multiple access with collision avoidance (CSMA/CA) mechanism and the binary exponential backoff (BEB) algorithm. Before initiating a transmission, a device senses the radio channel to determine if it is idle. If the channel is busy, the device defers its transmission and initializes a random backoff countdown. This countdown decrements exclusively while the channel remains idle, pauses whenever channel activity is detected, and resumes once the medium becomes available again. Note that although OFDMA was introduced with the release of the 802.11ax amendment, CSMA/CA remains by far the predominant multiple access scheme in use for WLANs.

B. Bursty traffic

While some applications generate relatively smooth traffic (*e.g.,* constant bitrate streams), others exhibit bursty behavior with significant variability in packet inter-arrival times. Packet bursts may be caused for several reasons: (i) TCP congestion control that tends to group packets together, leading to bursts of transmission [20], (ii) Modern websites, where loading a single webpage requires fetching numerous embedded objects (*e.g.,* images, JavaScript, CSS files) via a cascade of sequential or concurrent HTTP requests, and (iii) Video-on-demand services (*e.g.,* YouTube, Netflix) that transmit batches of packets, corresponding to multiple video frames, in rapid succession.

C. TWT options

TWT can be configured with various options, which are established during a TWT agreement between the AP and the STA running TWT. In this paper, we consider the following TWT configuration:

- **Individual** - Each STA running TWT will negotiate its own TWT setting with the AP. The alternative, which requires OFDMA for frame transmission, is the broadcast option, where the AP sets up a shared TWT session for a group of stations.
- **Non trigger-enabled** - The STA can use either classical CSMA/CA or OFDMA to access the radio channel. If the trigger-enabled option is selected, the STA must use OFDMA. Consequently, in a non trigger-enabled configuration, the STA initiates upstream transmissions autonomously, rather than waiting for an explicit Trigger frame from the AP.
- **Unannounced** - The AP can transmit downstream data to the STA during the TWT service period without waiting for an explicit frame from the STA indicating that it has awakened. Conversely, if the announced option is selected, the STA must first transmit a frame to notify the AP that it is awake.
- **Implicit** - TWT settings are periodically and automatically renewed. If the explicit option is selected, TWT settings must be periodically updated.

For further details, we refer the reader to [19].

D. TWT service periods and their parameters

TWT defines service periods (SPs), during which a STA may exchange traffic, and restricted periods (RPs), during which it may not. Note that, in theory, restrictions may apply only to the STA's upstream traffic, since TWT was originally designed as a scheduling mechanism to avoid contention in the access to the radio channel. However, since TWT is now primarily used for energy saving, both upstream and downstream traffic is generally restricted during RPs. This

allows the STA to safely enter a doze state (*i.e.,* turn off the radio module of its network interface card (NIC)) without the risk of missing a frame.

With our selected TWT options, the configuration of a TWT agreement concludes with the setting of three parameters defined in the IEEE 802.11 standard, that determine the timing of SPs and RPs:

- `Target Wake Time` defines the start of the first SP;
- `TWT Wake Interval` defines the periodicity of SPs;
- `Minimum TWT Wake Duration` defines their length.

Fig. 1 illustrates TWT and the influence of its SPs on the transmission times of frames destined for a STA, using a simple example. Note that TWT allows the AP to prematurely interrupt a SP by using a special frame with the "End Of Service Period" (EOSP) flag enabled. But this naturally only applies to downstream traffic, and since this paper focuses on TWT for energy saving at the STAs, we assume that SPs always run to completion.

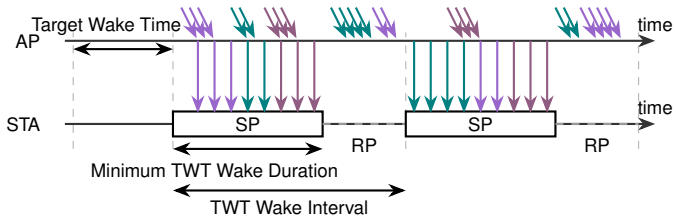


Fig. 1: Target Wake Time (TWT) mechanism diagram.

III. System description and assumptions

We now present the WLAN under study, along with the key assumptions made regarding its configuration and operation.

We analyze a WLAN comprising one AP and N STAs operating under TWT, all competing for radio channel access. All STAs share identical TWT parameter configurations and thus the same SPs. Note that if STAs belong to different (non-overlapping) SPs, one may analyze each SP separately. For the sake of conciseness, we (re-)define the key TWT parameters as follows: SP_{start} is the timestamp of the first SP, T_{SP} is the duration of an SP, and T_{RP} is the duration of an RP. Mapped to the IEEE 802.11 standard vocabulary (see Section II-D), they are: SP_{start} = `Target Wake Time`, T_{SP} = `Minimum TWT Wake Duration`, and T_{RP} = `TWT Wake Interval` - T_{SP}.

The AP periodically broadcasts a type of management frame, named beacon, that contains general information (*e.g.,* the Service Set Identifier (SSID), the supported physical data rates). We denote by BI the beacon interval duration. By default, this value is set to 102.4 ms [23].

For traffic modeling, consider the i-th STA, $i \in \{1, \ldots, N\}$. Its upstream traffic (from the i-th STA to the AP) generates individual packets and is modeled as a Poisson process with rate γ_i. The total upstream traffic rate can be calculated as the sum of all individual rates: $\gamma = \sum_{i=1}^{N} \gamma_i$.

The downstream traffic from the AP to the i-th STA arrives in batches, with batch arrivals following a Poisson process at rate λ_i. Let A_i be the random variable representing the batch size (in packets) for the i-th STA, that takes integer values in $[1, B_i]$. Its probability mass function (PMF) is denoted by the q_i vector, satisfying: $\sum_{a=1}^{B_i} q_i(a) = 1$. The mean batch size, $\bar{A}_i$, can be easily computed: $\bar{A}_i = \sum_{a=1}^{B_i} q_i(a) \cdot a$.

These assumptions on the traffic enable modeling a wide range of traffic patterns, from sporadic single-packet arrivals (by considering that all batches are of size 1, *i.e.,* $q_i(1) = 1, i \in \{1, \ldots, N\}$), to bursty traffic with variable-sized batch arrivals (*e.g.,* for video streaming, by setting the batch size distribution so as to match the one of the video traffic).

We denote by $1/\mu_i$ the mean transmission time of a downstream frame to the i-th STA (including the reception of the acknowledgement), in the absence of competing traffic (*i.e.,* backoff delays drawn by the AP are never suspended and frame collisions never occur). This value depends on several factors, including the 802.11 amendment, channel bandwidth, frame size, modulation and coding scheme (MCS), and the number of spatial streams (NSS). Nonetheless, the computation of μ_i, known as the mean transmission rate, is standard and can be derived from the IEEE 802.11 specifications; see Appendix A for details.

Finally, we denote by K the AP buffer size (in packets).

Table I summarizes the main notation of this paper.

TABLE I: Main notation of this paper.

Notation	Description
N	Number of STAs
SP_{start}	Timestamp of the first SP of a STA
T_{SP}	Minimum TWT Wake Duration or the wake duration of a STA within a TWT interval (ms)
T_{RP}	Sleep duration of a STA within a TWT interval (ms)
BI	Beacon interval duration (ms)
γ_i	Upstream traffic from the i-th STA (packets/s)
γ	Total upstream traffic rate (packets/s)
μ_i	Mean transmission rate of the downstream traffic to the i-th STA (packets/s)
$\bar{\mu}$	Mean transmission rate of the overall downstream traffic (packets/s)
μ'	Mean transmission rate of the overall downstream traffic used in the model (packets/s)
λ_i	Batch arrival rate of the traffic to the i-th STA (batches/s)
B_i	Maximum possible batch size of the traffic to the i-th STA (packets)
q_i	Batch size probability vector of the traffic to the i-th STA
$\bar{A}_i$	Mean size of a batch of the traffic to the i-th STA (packets)
K	AP buffer size (packets)
$p(s)$	Steady-state probability that there are s packets in the AP buffer, $s \in \{0, \ldots, K\}$
Q	Mean number of packets found in the AP buffer
L	Packet loss rate of the total downstream traffic
X	Mean packet throughput of the total downstream traffic (packets/s)
R	Mean packet delay of the total downstream traffic (ms)
G	Energy reduction factor relative to not using TWT

IV. Performance modeling

A. Transmission rate adjustment

Given that the N STAs composing the WLAN have different traffic rates, then the overall downstream mean transmission rate, denoted by $\bar{\mu}$, can be calculated as the weighted average of all μ_i:

$$\bar{\mu} = \sum_{i=1}^{N} \frac{\lambda_i \cdot \bar{A}_i}{\sum_{j=1}^{N} \lambda_j \cdot \bar{A}_j} \cdot \mu_i. \tag{1}$$

To account for the competing upstream traffic and the presence of RPs, we set the AP mean transmission rate that we use in our model, μ', by adjusting $\bar{\mu}$ with: (i) the proportion of the downstream traffic from the total downstream and upstream traffic and (ii) the proportion of time that the STAs are active, as follows:

$$\mu' = \bar{\mu} \cdot \frac{\sum_{i=1}^{N} \lambda_i \cdot \bar{A}_i}{\gamma + \sum_{i=1}^{N} \lambda_i \cdot \bar{A}_i} \cdot \frac{T_{SP}}{T_{SP} + T_{RP}}. \tag{2}$$

B. Continuous-time Markov chain

In our model, we track the evolution of the number of packets queued in the AP buffer and destined to the STAs under analysis. We model this quantity using the discrete variable s. The variable s ranges from 0 to K and we use a Markov chain to model its evolution as illustrated by Fig. 2. To maintain the Markov property, we assume that the frame transmission times, $1/\mu'$, are exponentially distributed.

From a given state s ($0 \leq s \leq K$), the Markov chain can transition to the state $s-1$ following a packet departure at rate μ' (if $s > 0$), or to a state $s+1$, $s+2$,..., following the arrival of a packet batch of size 1, 2,..., whose occurrence rate is $\sum_{i=1}^{N} \lambda_i \cdot q_i(1)$, $\sum_{i=1}^{N} \lambda_i \cdot q_i(2)$,... Note that packet losses due to buffer overflow occur whenever a packet batch of size greater than $K - s$ arrives at the AP buffer (in which case the current state s transitions to state K).

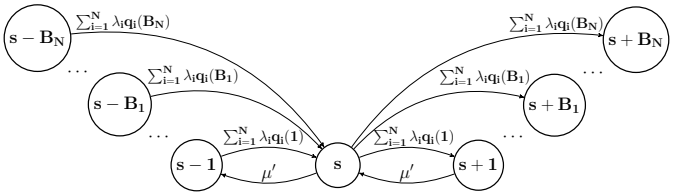


Fig. 2: Markov chain transitions for a state s, where $\max(B_i) \leq s \leq K - \max(B_i)$, $i \in [1, N]$.

With this state description, we denote by $p(s)$ the steady-state probability of state s. In other words, $p(s)$ is the probability that s packets for the STA under analysis are currently queued in the AP buffer.

We can then derive the following balance equations:

$$\begin{cases} p(0) \cdot \sum_{i=1}^{N} \lambda_i = p(1) \cdot \mu', \text{ for } s = 0 \\ p(s) \cdot (\mu' + \sum_{i=1}^{N} \lambda_i) = p(s+1) \cdot \mu' + \\ \qquad + \sum_{x=0}^{s-1} p(x) \sum_{i=1}^{N} \lambda_i q_i(s-x), \text{ for } 1 \leq s \leq K-1 \\ p(K) \cdot \mu' = \sum_{x=0}^{K-1} p(x) \sum_{i=1}^{N} \lambda_i \sum_{a=K-x}^{B_i} q_i(a), \text{for } s = K \end{cases} \tag{3}$$

By definition, the probabilities $p(s)$ must also verify the normalizing condition:

$$\sum_{s=0}^{K} p(s) = 1. \tag{4}$$

The steady-state values of $p(s)$ must satisfy both Eq. 3 and Eq. 4. For this paper, we employ an iterative procedure to obtain their values: starting with an initial, feasible solution for the p vector (*e.g.,* an uniform distribution), we iteratively update its values using each equation, until convergence is achieved (*e.g.,* when the current and previous values of p differ by less than 10^{-6}). In our experience, the number of iterations may be as few as two for simple Poisson packet arrivals, whereas dozens may be needed when packets arrive in batches of varying sizes. The Python code with our implementation is available at [5].

C. Performance metrics

Having obtained the values of $p(s)$, we can derive the following performance metrics:

- the mean number of packets queued in the buffer:

$$Q = \sum_{s=0}^{K} p(s) \cdot s \tag{5}$$

- the packet loss rate, calculated as the expected number of packets dropped with the arrival of a batch, divided by the expected value of a batch size:

$$L = \frac{\sum_{s=0}^{K} \sum_{i=1}^{N} \sum_{a=1}^{B_i} p(s) \cdot \lambda_i \cdot q_i(a) \cdot \max(s + a - K, 0)}{\sum_{i=1}^{N} \sum_{a=1}^{B_i} \lambda_i \cdot q_i(a) \cdot a} \tag{6}$$

- the throughput (in packets/s), by computing the mean packet arrival rate and subtracting the loss:

$$X = \left[\sum_{i=0}^{N} \sum_{a=1}^{B_i} \lambda_i \cdot q_i(a) \right] \cdot (1 - L) \tag{7}$$

- the mean packet delay, by applying Little's law [7], [8]:

$$R = \frac{Q}{X} \tag{8}$$

- the energy reduction factor, calculated as the proportion of time spent in the doze state:

$$G = 1 - \frac{T_{SP}}{T_{SP} + T_{RP}} \tag{9}$$

V. Model validation

To assess our model accuracy, we compare its estimations for the performance metrics with the values obtained from network simulation, as well as with two alternative modeling approaches. Section V-A details the choice of the simulator and the simulation setup. Section V-B describes the other two compared solutions. Section V-C and Section V-D analyze two different scenarios and discuss the obtained results.

A. Simulation setup

As the ground truth for our evaluation, we use the discrete-event simulator ns-3 [3], which is widely adopted by the research community. Since there is currently no implementation of the TWT mechanism in its official release, we rely on the enhancement of the Wi-Fi module from the 2024 edition of Workshop on ns-3 [26].

We extend the Wi-Fi module patch with additional functionalities to support our target scenario, the final code being available here [2]. We implement an application that can generate the traffic batches at exponential inter-arrival times. We add UDP support alongside TCP, as the industry-standard QUIC protocol inherently relies on UDP for real-time video streaming.

We use the IEEE 802.11ax amendment (Wi-Fi 6), a frequency band of 5 GHz, a channel width of 20 MHz, a guard interval (GI) of 800 ns, 1 spatial stream and the high-efficiency (HE) MCS index of 7. We adequately increase the following timeout thresholds, to ensure they do not interfere with the scenario's execution: the ARP table timeout (how long an entry can be held in the cache before it needs to be renewed by a re-resolution) and the Wi-Fi MAC queue maximum delay (how much a packet can remain in the queue before it is expired, therefore dropped). No more than one MPDU (MAC protocol data unit) and one MSDU (MAC service data unit), respectively is transmitted at a time (*i.e.,* there is no frame aggregation). The maximum video application payload size is 1472 bytes.

The duration of a simulation is 100s, including the 10s warmup period. We run each simulation 15 times, with different, random seeds, to obtain the 95% confidence intervals for the performance metrics of the traffic issued by the AP: the packet loss rate of the total downstream traffic (L), the mean packet delay of the total downstream traffic (R), and the reduction factor of the energy consumption (G). The energy reduction factor is relative to a baseline simulation with the same parameters, in which TWT is disabled.

B. Alternative modeling approaches

We also compare the performance estimations obtained by our model to other two modeling approaches: a bufferless approach and an existing solution [15].

1) A bufferless modeling approach: In a bufferless model, also called a loss system, no queuing room exists for packets waiting transmission (*i.e.,* no buffer is available). This approach has been successfully applied to the dimensioning of call centers, yielding the well-known Erlang loss formula [13]. It has also found applications in admission control [12] [14].

To adjust this bufferless approach to our case, we calculate the arrival rate, λ, as the sum of all downstream rates, expressed in packets/s: $\lambda = \sum_{i=1}^{N} \lambda_i \cdot \bar{A}_i$. We consider an adjusted transmission rate, μ'': (i) as the proportion of the time that the STAs are active and (ii) as a proportion of the sum of the total downstream and upstream rates, only if it exceeds the system transmission rate:

$$\mu'' = \begin{cases} \bar{\mu} \cdot \frac{T_{SP}}{T_{SP}+T_{RP}}, \text{if } \lambda + \gamma \leq \bar{\mu} \\ \bar{\mu} \cdot \frac{\lambda}{\lambda+\gamma} \cdot \frac{T_{SP}}{T_{SP}+T_{RP}}, \text{otherwise} \end{cases} \quad (10)$$

If the transmission rate is greater or equal to the arrival rate, the loss rate is 0. Otherwise, the mean loss rate is calculated as the difference between the arrival rate and the transmission rate, normalized by the arrival rate: $max(0, 1 - \frac{\mu''}{\lambda})$. The mean packet delay is equal to the mean transmission time of a packet: $1/\mu''$. Finally, the energy reduction factor is the proportion of time that the STAs sleep during a TWT interval: $1 - \frac{T_{SP}}{T_{SP}+T_{RP}}$.

2) Probability tree model: In [15], the authors develop an analytical solution based on the construction of a large probability tree model. Following the Python implementation of their solution [1], we adapt it to our scenarios. The model takes into account the traffic rates of a single STA with a TWT agreement. In our case, since all STAs have the same service periods, we use the global rates of the downstream/upstream traffic. The downstream batch rate is simply the sum of all individual rates: $\sum_{i=1}^{N} \lambda_i$, while the downstream batch size is calculated as a weighted batch mean size: $\frac{\sum_{i=1}^{N} \lambda_i \cdot \bar{A}_i}{\sum_{i=1}^{N} \lambda_i}$. For the upstream traffic rate, we use γ as the batch rate, with a batch size of 1. For the transmission rate, we use $\bar{\mu}$. As for the size of the AP and STAs buffers, we simply use the same values as in the following considered scenarios.

We limit the code iterations to a coverage of 95% of the probability tree, or a maximum of 10^6 visited leaves, or a maximum of 10^5 visited nodes. We obtain values for the packet loss rate and the mean packet delay that we compare to the results of our model. To obtain the energy reduction factor, we divide the computed power consumption with TWT by the power consumption of the awake state, *i.e.,* when not using TWT at all: $1 - \frac{P}{P_{awake}}$, using the notation from [15].

C. Scenario 1: Poisson traffic

In our first analyzed scenario, we consider one AP and N=5 STAs. The AP sends downstream traffic to all 5 STAs, following Poisson processes with the following rates: λ_1 = 1000, λ_2 = λ_4 = 100, λ_3 = λ_5 = 200. Only the first STA sends upstream traffic at rate γ = γ_1 = 100. By definition, all batches have size 1, *i.e.,* there are individual packet arrivals. We consider two possible sizes of the AP buffer, namely 100 and 5000 packets. The size of the STAs buffers has a negligible effect, all STAs having a buffer size of 500 packets.

All devices have the same TWT agreement: the same SP_{start}, the same T_{SP}, and the same T_{RP}. In this scenario, we maintain the synchronization of the STAs with the beacon interval duration, as done in [26]. In that case, the first SP will start immediately after the first received beacon, and we must have: $T_{SP} + T_{RP} = BI = 102.4$ ms. T_{RP} should not be less than 3.6 ms, as this is the minimum time a NIC needs to enter the doze state. [15] With respect to this, we analyze RP durations ranging from 3.6 ms up to 100 ms.

We now discuss the obtained performance metrics, shown in Fig. 3. For an AP buffer size of 100, from the simulation results in Fig. 3a, we observe, as expected, an increasing trend in the packet loss rate, with respect to the increase in the sleep duration of the STAs, T_{RP}. From the simulation results in Fig. 3b, we can also observe an increase in the mean packet delay, as the sleep time of the STAs increases. In Fig. 3c, we can notice that the energy reduction is directly proportional to the ratio of time that the STAs sleep during a TWT interval.

Increasing the AP buffer size to 5000 results in a slight reduction in packet loss. E.g., for RP durations between 50 and 100 ms, the loss rate is 3-4% lower in Fig. 3d relative to Fig. 3a. The larger AP buffer size also causes a significant increase in packet delay, since the packets may spend more time waiting in the buffer. For instance, for RP durations between 50 and 90 ms, the packet delay has values between 50 and 400 ms for the buffer size of 100 in Fig. 3b, while the delay has values between 3 and 18s for the same RP durations and the AP buffer size of 5000 in Fig. 3e. By comparing Fig. 3c and Fig. 3f, we can see that the change in the buffer size has little impact on the energy reduction, therefore we can conclude that it depends mainly on the TWT parameters.

Overall, we can observe that our model closely matches the three performance metrics (packet loss rate, packet delay and energy reduction factor) obtained from the simulations, for both the AP buffer sizes of 100 and 5000. There is an exception in the packet delay for the biggest analyzed T_{RP} duration, *i.e.,* 100 ms, in Fig. 3b and Fig. 3e. Note that this corresponds to the extreme case of a SP as short as 2.4 ms. Naturally, the T_{RP} durations that induce high loss rates or severe packet delays are undesirable in practical WLAN operations. However, their analysis illustrates the model's capacity to accurately detect the transition point at which system performance degrades and packet loss escalates.

The alternative solutions, *i.e.,* the bufferless approach and the probability tree model, do not predict as accurately the packet loss rate for different RP durations (see Fig. 3a and Fig. 3d). They also do not account for the change in the size of the buffer, as noticed from the constant values of the packet delay in Fig. 3b and Fig. 3e.

D. Scenario 2: Weibull traffic

In the second scenario, we keep the same WLAN configuration: one AP and N=5 STAs. As opposed to the first scenario, we generate downstream traffic to the first, third and fifth STAs in packet batches, in order to simulate a bursty traffic. To define the batch sizes, we use the Weibull distribution [17], inspired by [26], but with exponential inter-arrival times. More details regarding the batch size distribution can be found in Appendix B.

The remaining downstream traffic for the second and fourth STAs follows the same Poisson processes as in the first scenario: λ_2 = λ_4 = 100 and batch sizes of 1. We try to maintain the same mean arrival rates in packets/s as in the first scenario, in order to analyze the impact of traffic bursts to the performance metrics, compared to individual packet arrivals.

The upstream traffic rates are the same as in the first scenario: γ = γ_1 = 100. We analyze the same AP buffer sizes, of 100 and 5000 packets, for the same buffer size at the STAs, of 500 packets. We analyse the same TWT configurations as in the first scenario, with RP durations ranging from 3.6 to 100 ms, and $T_{SP} + T_{RP} = BI = 102.4$ ms.

We now discuss the results for the performance metrics, shown in Fig. 4. Similarly to Scenario 1, we can observe an increase in the values of all the performance metrics as the RP duration increases.

Then, for a relatively small size of the AP buffer, of 100 packets, the loss rates obtained with Weibull traffic in Fig 4a are significantly higher than the loss rates obtained with Poisson traffic in Fig 3a. With Poisson traffic, in the simulation we obtain zero packet loss for RP durations less than or equal to 30 ms, whilst with Weibull the loss rate remains greater than 12% even for the smallest analyzed RP duration, of 3.6 ms. By increasing the AP buffer size from 100 to 5000, the loss rates obtained with Weibull traffic in Fig 4d approach the loss rates obtained with Poisson traffic for the same AP buffer size in Fig 3d. The packet delays with Weibull traffic for both AP buffer sizes in Fig 4b and Fig 4e are similar to packet delays with Poisson traffic in Fig 3b and Fig 3e.

As for the first scenario, our model closely matches the performance metrics obtained with the simulations, for both AP buffer sizes and for the range of RP durations, except for the packet delay of the longest analyzed RP value, namely 100 ms. The alternative modeling approaches are less sensible to the buffer size and to the impact of bursts, as they do not allow specifying a general batch size distribution.

VI. Model exploitation to guide the configuration of TWT

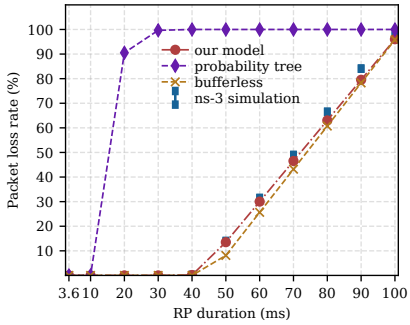
In this section, we demonstrate how one can use our model to minimize energy consumption while maintaining a set of QoS requirements, expressed as a maximum tolerable packet loss rate and a maximum tolerable mean packet delay. In return, we suggest an adequate configuration of the TWT parameters.

A. Problem formulation and solution

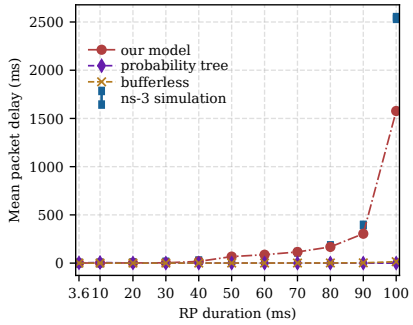
We formulate the search of an adequate TWT parameter configuration as a root-finding problem (*i.e.,* solving an $f(x) = 0$ equation) for an objective function parameterized by T_{RP}. To maintain the synchronization with the AP beacons, T_{SP} is constrained by the relation: $T_{SP}+T_{RP} = BI = 102.4$ ms, and SP_{start} occurs right after the transmission of the next beacon.

The objective function runs the model calculations (fed with the respective scenario parameters) and returns the deviation of its predicted performance metrics from the required QoS values. The result is negative if the QoS conditions are met, and positive otherwise. Since the RP duration is directly proportional to energy savings, as observed in Fig. 3 and Fig. 4, the zero-crossing point represents the optimal operational value that minimizes overall energy consumption, while maintaining QoS compliance.

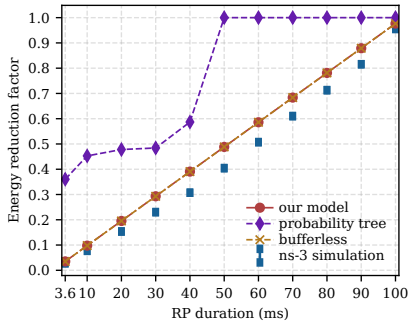
One classic root-finding algorithm is binary search (also known as the bisection method). We propose Algorithm 1, in which we use the Chandrupatla method [10], for a faster execution time [24, p. 105,111]. We resort to the implementation from the SciPy Python module [4].



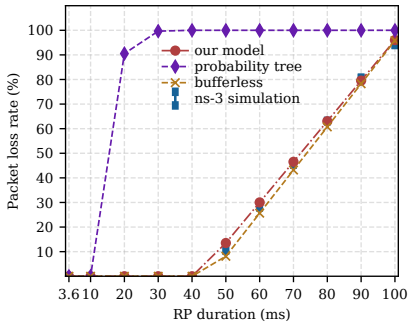
(a) AP buffer size 100 - packet loss rate.



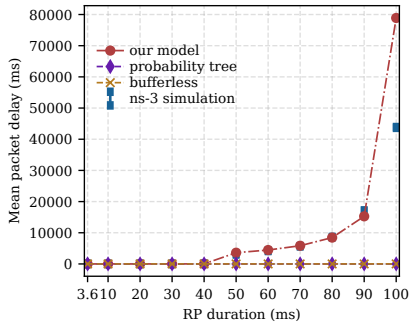
(b) AP buffer size 100 - mean packet delay.



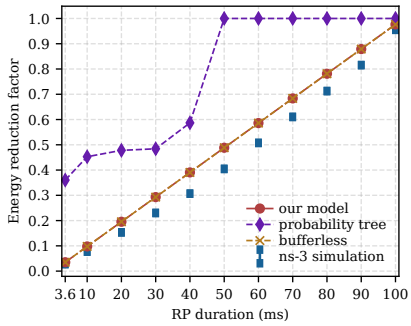
(c) AP buffer size 100 - energy reduction.



(d) AP buffer size 5000 - packet loss rate.

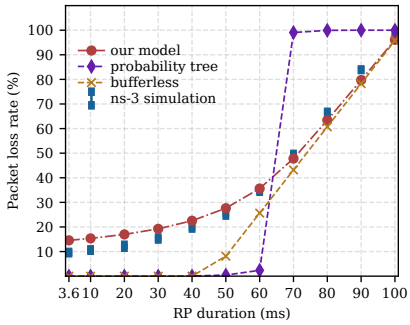


(e) AP buffer size 5000 - mean packet delay.

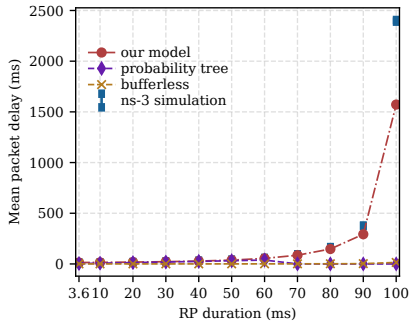


(f) AP buffer size 5000 - energy reduction.

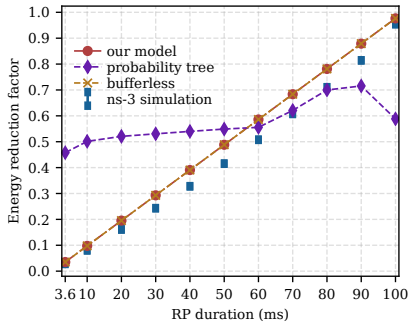
Fig. 3: Scenario 1: Poisson traffic. Our model against simulation results, for different AP buffer sizes.



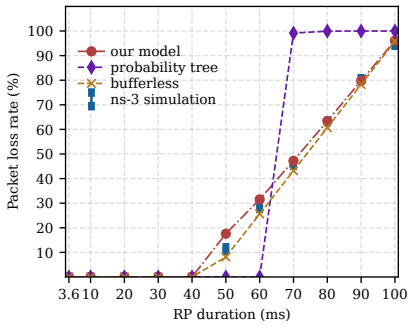
(a) AP buffer size 100 - packet loss rate.



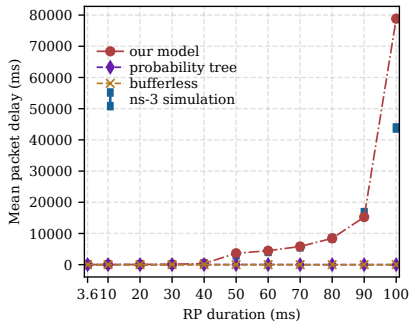
(b) AP buffer size 100 - mean packet delay.



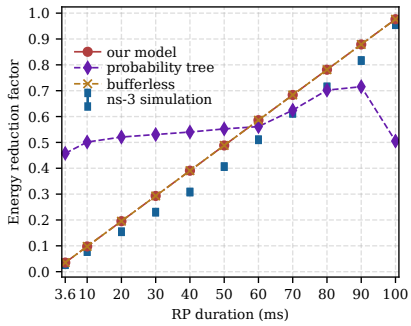
(c) AP buffer size 100 - energy reduction.



(d) AP buffer size 5000 - packet loss rate.



(e) AP buffer size 5000 - mean packet delay.



(f) AP buffer size 5000 - energy reduction.

Fig. 4: Scenario 2: Weibull traffic. Our model against simulation results, for different AP buffer sizes.

Algorithm 1: Selecting adequate TWT parameters based on the QoS requirements

Input: QoS requirements: (maximum tolerable loss rate L_{max}, maximum tolerable delay R_{max}); scenario parameters *systemParams*: traffic rates, AP buffer size, physical data rate,...

Output: The values for `TWT Wake Interval` and `Minimum TWT Wake Duration`

Function *objectiveFunction(systemParams,* T_{RP}, L_{max}, R_{max}*) : float* **is**
  $T_{SP} \leftarrow BI$ - T_{RP};
  (L, R) $\leftarrow$ runModel(*systemParams*, T_{SP}, T_{RP});
  lossDeficit $\leftarrow$ L - L_{max};
  delayDeficit $\leftarrow$ R - R_{max};
  // Return the worst constraint violation magnitude
  return max(lossDeficit, delayDeficit) ;
end
$BI \leftarrow$ 102.4 ms;
$T_{RP_min} \leftarrow$ 3.6 ms;
$T_{RP_max} \leftarrow$ 100 ms;
$T_{RP_result} \leftarrow$ apply Chandrupatla method with *objectiveFunction* and [T_{RP_min}, T_{RP_max}];
`TWT Wake Interval` $\leftarrow BI$;
`Minimum TWT Wake Duration` $\leftarrow BI$ - T_{RP_result};
return `TWT Wake Interval`, `Minimum TWT Wake Duration`;

B. Use cases

In this subsection, we compare the solution proposed by Algorithm 1 with the optimal solution found in a brute force manner with ns-3, *i.e.,* we run multiple simulations with consecutive T_{RP} values, with step 0.01, and choose the largest one for which the QoS requirements are still met. We consider a tolerance of 10^{-6} for the model loss rate calculation.

1) Use case 1: Smooth traffic: For the first use case, we consider only Poisson, individual packet arrivals, identical to Scenario 1 of Section V-C. Expecting the traffic to be an uncongested flow, we enforce strict QoS requirements of L_{max} = 0% and R_{max} = 50 ms, for an AP buffer size of 100 packets.

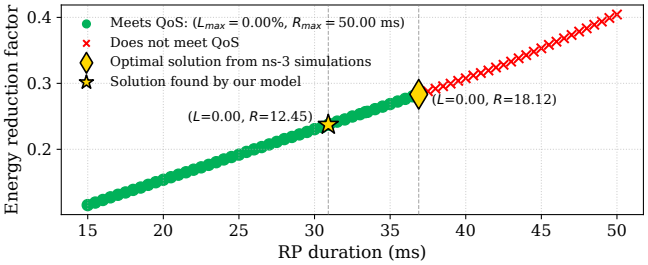
The T_{RP} duration found with Algorithm 1 is 30.91 ms, with the corresponding TWT parameters: `TWT Wake Interval` = BI = 102.4 ms, `Minimum TWT Wake Duration` = $BI - T_{RP}$ = 71.49 ms.

In Fig. 5a we plot the performance metrics obtained from simulations for a range of T_{RP} values, their relation with the QoS criteria and the optimal solution. The RP duration found with our model, T_{RP} = 30.91 ms, does not violate the QoS requirements and is close to the optimal solution determined from the simulations, T_{RP} = 36.89 ms. This results in an energy reduction of over 30%.

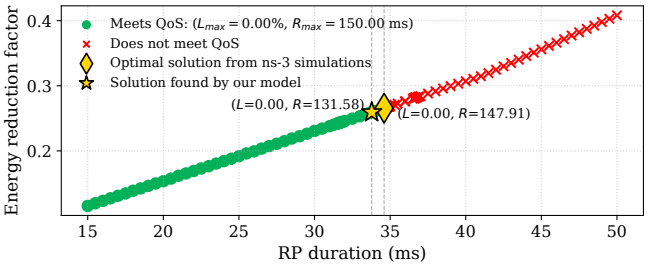
2) Use case 2: Bursty traffic: For this use case, we expect bursty traffic to some of the STAs, identical to Scenario 2 (Section V-D). We assume a higher delay, R_{max} = 150 ms, and a L_{max} of 0%.

The T_{RP} duration found with Algorithm 1 is 33.78 ms, with the corresponding TWT parameters: `TWT Wake Interval` = BI = 102.4 ms, `Minimum TWT Wake Duration` = $BI - T_{RP}$ = 68.62 ms.

In Fig. 5b we plot the performance metrics obtained from the simulations for a range of T_{RP} values, their relation with the QoS criteria and the optimal solution. The RP duration found with our model, T_{RP} = 33.78 ms, is close to the optimal solution determined from the simulations, T_{RP} = 34.60 ms. Using TWT with the proposed configuration results in an energy reduction of over 30%.



(a) Use case 1: Smooth traffic.



(b) Use case 2: Bursty traffic.

Fig. 5: Goodness of the TWT configuration suggested by our model versus the one found by an exhaustive search with ns-3 simulations.

VII. Conclusions

In this paper, we investigate the performance modeling of a WLAN operating under the energy-saving mechanism TWT. We develop a comprehensive Markovian model that accounts for many factors, including the TWT configuration, the AP buffer size, the physical data rate of the radio channel, the number of competing STAs, and the packet arrival process, possibly represented as variable-sized packet batches. The model yields closed-form expressions for key performance metrics such as throughput, packet loss rate, mean queue length, mean queuing delay, and the energy reduction factor. Through extensive evaluation across different scenarios and using the discrete-event simulator ns-3, we establish the good accuracy of our Markovian model across various configurations of TWT, buffer sizes and traffic pattern. Furthermore, we illustrate how our model can guide the selection of adequate TWT parameter configurations through two use cases.

As future works, we intend to extend our modeling framework to account for scenarios involving multiple WLANs operating on the same radio channel. A key challenge in such an environment is coordinating the TWT configurations of the different WLANs to minimize overlap in their awake periods, thereby reducing contention. Addressing this coordination problem will require developing new scheduling techniques and optimization strategies.

REFERENCES

[1] Source code: Probability tree model source code. https://github.com/EstGue/Performance_modeling_TWT, 2025.

[2] ns-3 TWT Patch Enhancements. https://github.com/carrrina/ns-3-twt-patch, 2026.

[3] ns-3.43. The Network Simulator ns-3. https://www.nsnam.org/, 2026.

[4] scipy.optimize.elementwise.find_root. find_root - SciPy v1.17.0 Manual https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.elementwise.find_root.html, 2026.

[5] Source code: TWT Markovian model. https://github.com/carrrina/twt-markovian-model, 2026.

[6] Jiyang Bai et al. Adaptive Uplink OFDMA Random Access Grouping Scheme for Ultra-Dense Networks in IEEE 802.11ax. In *2018 IEEE/CIC International Conference on Communications in China (ICCC)*, 2018.

[7] Gunter Bolch et al. *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons, 2006.

[8] Alexandre Brandwajn and Thomas Begin. *Computer Performance Modeling*. World Scientific, 2025.

[9] Fabio Busacca et al. Target Wake Time scheduling for time-sensitive and energy-efficient Wi-Fi networks. *IEEE Transactions on Mobile Computing*, 25(3), 2026.

[10] Tirupathi R. Chandrupatla. A new hybrid quadratic/bisection algorithm for finding the zero of a nonlinear function without using derivatives. *Advances in Engineering Software*, 28(3):145–149, 1997.

[11] Qinghua Chen et al. A Target Wake Time Based Power Conservation Scheme for Maximizing Throughput in IEEE 802.11ax WLANs. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, 2019.

[12] Sally Floyd. Comments on measurement-based admissions control for controlled-load services. Technical report, Lawrence Berkeley Laboratory, 1996.

[13] Noah Gans et al. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & service operations management*, 2003.

[14] S. Georgoulas et al. An integrated bandwidth allocation and admission control framework for the support of heterogeneous real-time traffic in class-based IP networks. *Computer Communications*, 2008.

[15] Esther Guérin et al. An analytical method to configure TWT in WLANs. In *2025 International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2025.

[16] Mehmet KARACA. Joint optimization of target wake time mechanism and scheduling for IEEE802.11ax. *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, 29, 2021.

[17] CD Lai, DNP Murthy, and M Xie. Weibull distributions. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3(3):282–287, 2011.

[18] Ruofeng Liu and Nakjung Choi. A first look at Wi-Fi 6 in action: Throughput, latency, energy efficiency, and security. *Proc. ACM Meas. Anal. Comput. Syst.*, 7(1), March 2023.

[19] Maddalena Nurchis and Boris Bellalta. Target wake time: Scheduled access in IEEE 802.11 ax WLANs. *IEEE Wireless Communications*, 2019.

[20] Vern Paxson and Sally Floyd. Wide area traffic: the failure of poisson modeling. *IEEE/ACM Transactions on networking*, 3(3):226–244, 2002.

[21] Xiaohong Peng et al. Access point coordination based TWT scheduling for the next generation WLAN. In *2024 13th International Conference on Communications, Circuits and Systems (ICCCAS)*, 2024.

[22] Wenxun Qiu et al. Category-based 802.11ax Target Wake Time solution. *IEEE Access*, 2021.

[23] Salem Sati and Kalman Graffi. Adapting the beacon interval for opportunistic network communications. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.

[24] Philipp O. J. Scherer. *Computational Physics: Simulation of Classical and Quantum Systems*. Springer International Publishing, Cham, Switzerland, 3rd edition, 2017.

[25] Virgo Jaya Sianipar et al. Impact of target wake time (TWT) on energy efficiency in Wi-Fi 6 networks for real-time and non-real-time applications. In *2025 3rd International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE)*, 2025.

[26] Shyam Krishnan Venkateswaran et al. IEEE 802.11ax target wake time: Design and performance analysis in ns-3. In *Proceedings of the 2024 Workshop on ns-3*.

[27] Shyam Krishnan Venkateswaran et al. Poster: Accordion: Toward a limited contention protocol for Wi-Fi 6 scheduling. MobiHoc '23, New York, NY, USA, 2023. Association for Computing Machinery.

[28] Changmok Yang et al. Target Wake Time scheduling strategies for uplink transmission in IEEE 802.11ax networks. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*.

APPENDIX A
ANALYTICAL FRAME TRANSMISSION RATE CALCULATION

A. General formula

We compute the mean transmission rate, μ, as the number of packets that can be transmitted over a period of time.

We consider the aggregate MAC protocol data unit (A-MPDU) mechanism enabled, with a number of f maximum aggregated MPDUs, and the aggregate MAC service data unit (A-MSDU) disabled. We denote by $PSDU_{size}$ the maximum size of a PLCP Service Data Unit (PSDU), and by $MPDU_{size}^{(i)}$ the size of the i-th MPDU from a single transmission. Together, they satisfy:

$$PSDU_{size} = \sum_{i=1}^{f} MPDU_{size}^{(i)}. \tag{11}$$

We denote by T_{data} the necessary time to transmit exactly $PSDU_{size}$ and by $T_{overhead}$ the transmission overhead, that includes inter frame space (IFS) time intervals and the time to receive the acknowledgement (ACK). μ can be calculated as:

$$\mu = \frac{f}{T_{data} + T_{overhead}}. \tag{12}$$

To compute $MPDU_{size}^{(i)}$, we start from the application payload size (L_{data}) and we add the lengths of the protocol headers: the length of the layer 4 header (L_{H4}), the length of the layer 3 header (L_{H3}), the length of the layer 2 header (subdivided into the logical link control (LLC) header and the MAC header) (L_{H2}), the size of the frame check sequence (FCS), FCS_{size}, and, finally, the size of the MPDU-delimiter, $MPDU_{delim}$. Each non-final MPDU is added a padding, so as the total size in bytes to be divisible by 4. With PAD_i being the size of the padding, we obtain the following equation (the order of the addition terms is not necessarily the order in which the corresponding fields are found in the frame):

$$\begin{aligned} MPDU_{size}^{(i)} = L_{data} + L_{H4} + L_{H3} + L_{H2} \\ + FCS_{size} + MPDU_{delim} + PAD_i. \end{aligned} \tag{13}$$

To compute the time it takes to transmit an amount of data, we need to account for the number of bits that can be transmitted in one OFDM symbol and the duration of one symbol. Since a data frame and an ACK frame are usually not transmitted at the same rate, we denote by N_{DBPS} the number

of bits per symbol and by T_{sym} the symbol duration for data frames, and by N_{DBPS_ACK} and T_{sym_ACK} the respective values in the case of ACK frames. We also need to add the physical (PHY) layer overhead, *i.e.,* the physical layer's preamble and header. We denote by T_{PHY} the PHY overhead for a data frame and by $T_{LegacyPHY}$ the PHY overhead for an ACK frame. We compute T_{data} as:

$$T_{data} = T_{PHY} + T_{sym} \cdot \left\lceil \frac{PSDU_{size}}{N_{DBPS}} \right\rceil. \quad (14)$$

We define $T_{overhead}$ as the sum of:

- $T_{DIFS/AIFS}$ - the IFS waited by a device before transmitting, that can be either T_{DIFS}, for DCF inter-frame space (DIFS), or T_{AIFS}, for arbitration inter-frame space (AIFS);
- T_{BO} - the mean backoff duration waited by the device;
- T_{SIFS} - the short inter-frame space (SIFS), *i.e.,* the minimum waiting time required between receiving a frame and responding;
- T_{ACK} - the time it takes to receive the ACK (of size ACK_{size}), including the physical layer overhead:

$$T_{ACK} = T_{LegacyPHY} + T_{sym_ACK} \cdot \left\lceil \frac{ACK_{size}}{N_{DBPS_ACK}} \right\rceil; \quad (15)$$

$$T_{overhead} = T_{DIFS/AIFS} + T_{BO} + T_{SIFS} + T_{ACK}. \quad (16)$$

When collisions do not occur, the backoff value is selected randomly following an uniform distribution from [0, CW_{MIN}], giving the expected value of $CW_{MIN}/2$. The mean backoff duration is $T_{BO} = \frac{CW_{MIN}}{2} \cdot T_{slot}$, where T_{slot} is the duration of the slot. Also, we assume the propagation delay is negligible due to the short-range nature of Wi-Fi communications.

B. Application to our scenario

In our scenario, we consider the same mean transmission rate, μ, between the AP and each of the STAs, and we assume the transmission of only one MPDU at a time, *i.e.,* f=1.

We set the following values: L_{data}=1472 bytes, $L_{H4} = 8$ bytes (the UDP header), $L_{H3} = 20$ bytes (the IP header), L_{H2}=8+26=32 bytes, $FCS_{size} = 4$ bytes, $MPDU_{delim} = 4$ bytes, and $PAD_1 = 0$, since we have only one, final MPDU. We obtain: $PSDU_{size} = MPDU^{(1)}_{size} = 1548$ bytes.

We use the 802.11ax amendment (Wi-Fi 6) and the AC_BE (Best Effort) category from the EDCA (Enhanced Distributed Channel Access), which gives us the standard values: $T_{slot} = 9\mu$s, $T_{SIFS} = 16\mu$s, $T_{AIFS} = T_{SIFS} + 3 \cdot T_{slot} = 43\mu$s, $CW_{MIN} = 15$, $T_{BO} = \frac{15}{2} \cdot 9 = 67.5\mu$s.

We use the the high-efficiency (HE) MCS index of 7, a frequency band of 5 GHz, a channel width of 20 MHz, a guard interval (GI) of 800 ns, and 1 spatial stream. In Wi-Fi 6, the base OFDM symbol takes 12.8 μs to transmit, to which we add the GI to obtain: $T_{sym} = 12.8 + 0.8 = 13.6\mu$s.

For a HE MCS 7 on a 20 MHz, there are 242 data subcarriers. It uses 64-QAM modulation, which carries $\log_2(64) = 6$ bits per subcarrier, and the coding rate is $5/6$, which gives: $N_{DBPS} = 242 \cdot 6 \cdot \frac{5}{6} = 1210$ bits per symbol.

An ACK frame is transmitted at the legacy non-high-throughput (non-HT) mode, so that it could be decoded by every device on the channel. The legacy mode has a base OFDM symbol duration of 3.2 μs, to which GI is added: $T_{sym_ACK} = 4\mu$s. In a legacy 20 MHz channel, a 64-FFT scheme is used. Out of 64 total subcarriers, exactly 48 are dedicated to carrying data bits. We transmit at a 24 Mbps legacy rate, which uses 16-QAM modulation, so it maps $\log_2(16) = 4$ bits per subcarrier signal. The coding rate for 24 Mbps is defined by the standard as $1/2$. We obtain: $N_{DBPS_ACK} = 48 \cdot 4 \cdot \frac{1}{2} = 96$ bits per symbol. Disabled A-MPDU implies a normal ACK: $ACK_{size} = 14$ bytes.

We are left to determine the physical layer overheads. The legacy PHY overhead consists of: a legacy short training field (L-STF) of 8μs, a legacy long training field (L-LTF) of 8μs, followed by a single 4 μs OFDM symbol, called the legacy signal field (L-SIG): $T_{LegacyPHY} = 8 + 8 + 4 = 20\mu$s. Now we have: $T_{ACK} = 20 + 4 \cdot \lceil \frac{14 \cdot 8}{96} \rceil = 28\mu s$.

The HE PHY overhead contains the legacy $T_{LegacyPHY}$ block, followed by: a repeated legacy signal field (R-SIG) of 4μs, a HE signal A field (HE-SIG-A) of 8μs, a HE short training field (HE-STF) of 4μs, and a HE long training field (HE-LTF), calculated with the formula: $HE\text{-}LTF = (2 \cdot FFT_{period} + GI) \cdot NSS$, where FFT_{period} has the same duration as a legacy OFDM symbol, 3.2 μs. We obtain: $T_{PHY} = \lceil 20 + 4 + 8 + 4 + (2 \cdot 3.2 + 0.8) \cdot 1 \rceil = 44\mu$s.

Putting it all together: $T_{data} = 44 + 13.6 \cdot \lceil \frac{1548 \cdot 8}{1210} \rceil = 193.6\mu s$; $T_{overhead} = 43 + 67.5 + 16 + 28 = 154.4\mu s$; $\mu = \frac{1}{(193.6 + 154.4) \cdot 10^{-6}} = 2873$ packets/s.

APPENDIX B
SCENARIO 2: WEIBULL TRAFFIC DETAILS

A basic Weibull distribution is characterized by two parameters: a scale α and a shape β; $\alpha, \beta > 0$. A shape $\beta > 1$ will give more weight to larger values, while a shape $\beta < 1$ will give less weight to larger values. We denote by α_i the scale and by β_i the shape of the Weibull distribution used to generate the downstream traffic to the i-th STA. We generate values for the batch sizes expressed in bytes. We set a minimum batch size, B_{lower} = 12 bytes (so that the corresponding frame with its headers attains the minimum possible size of 64 bytes) and a maximum batch size, B_{upper} = $15 \cdot 10^5$ bytes. We denote by $CDF(b)$ the cumulative distribution function (CDF) of b bytes, which represents the probability to generate a batch size of less than or equal to b bytes. We normalize the values to the range $[CDF(B_{upper}) - CDF(B_{lower})]$. We obtain $q_i(a)$, the probability of occurrence of a batch size of a packets, $a \in \{1, \ldots, B_i\}$, using the video application payload size (L_{data}), according to the formula:

$$q_i(a) = \frac{CDF[min(a \cdot L_{data}, B_{upper})] - CDF[(a-1) \cdot L_{data}]}{CDF(B_{upper}) - CDF(B_{lower})} \quad (17)$$

The numerical values used in Scenario 2 are: α_1 = 43247.57, $\alpha_3 = \alpha_5$ = 26210.65; $\beta_1 = \beta_3 = \beta_5$ = 0.8099; $\bar{A}_1$ = 32.50, $\bar{A}_3 = \bar{A}_5$ = 19.51; λ_1 = 30, $\lambda_3 = \lambda_5$ = 10.