

1-1-2021

Joint optimization of target wake time mechanism and scheduling for IEEE802.11ax

MEHMET KARACA

Follow this and additional works at: <https://journals.tubitak.gov.tr/elektrik>



Part of the [Computer Engineering Commons](#), [Computer Sciences Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

KARACA, MEHMET (2021) "Joint optimization of target wake time mechanism and scheduling for IEEE802.11ax," *Turkish Journal of Electrical Engineering and Computer Sciences*: Vol. 29: No. 3, Article 22. <https://doi.org/10.3906/elk-2005-218>

Available at: <https://journals.tubitak.gov.tr/elektrik/vol29/iss3/22>

This Article is brought to you for free and open access by TÜBİTAK Academic Journals. It has been accepted for inclusion in Turkish Journal of Electrical Engineering and Computer Sciences by an authorized editor of TÜBİTAK Academic Journals. For more information, please contact academic.publications@tubitak.gov.tr.

Joint optimization of target wake time mechanism and scheduling for IEEE 802.11ax

Mehmet KARACA* 

Department of Electrical and Electronics Engineering, Faculty of Engineering, TED University, Ankara, Turkey

Received: 31.05.2020

Accepted/Published Online: 09.04.2021

Final Version: 31.05.2021

Abstract: IEEE 802.11ax as the newest wireless local area networks (WLANs) standard brings enormous improvements in network throughput, coverage and energy efficiency in densely populated areas. Unlike previous IEEE 802.11 WLAN standards where power saving mechanisms have a limited capability and flexibility, 802.11ax comes with a different mechanism called target wake time (TWT) where stations (STAs) wake up only after each TWT interval and different STAs can wake up at different time instance depending on their application requirements. As an example, for a periodic data arrival occurring in IoT applications, STA can wake up by following the data period and go to sleep mode for a much longer time, and STAs with high traffic volume can have shorter TWT interval to wake up more frequently. Moreover, as multiuser transmission capability is added to 802.11ax, multiple STAs can have the same TWT interval and wake up at the same time, and hence there is a great opportunity to have collision-free transmission by scheduling multiple STAs on appreciate TWT intervals to reduce energy consumption and also increase network throughput. In this paper, we investigate the problem of STAs scheduling and TWT interval assignment together to reduce overall energy consumption of the network. We propose an algorithm that dynamically selects STAs to be served and assigns the most suitable TWT interval given STAs' traffic and channel conditions. We analyze our algorithm through Lyapunov optimization framework and show that our algorithm is arbitrarily close to the optimal performance at the price of increased queue sizes. Simulation results show that our algorithm consumes less power and support higher traffic compared to a benchmark algorithm that operates randomly for TWT assignment.

Key words: IEEE 802.11ax, power saving, TWT mechanism, Lyapunov optimization, random traffic

1. Introduction

IEEE 802.11 based WLANs has been the most popular way to reach wireless internet for their great flexibility and cost efficiency. Since the last decade, the increasing demand for data-hungry services and the deployment of Wi-Fi hotspots almost everywhere has triggered the new standardization activities, and the first signal was the need for the development of IEEE 802.11ac [1] that can provide gigabit rates with the enhancements made in mostly PHY layer. However, the improvements in PHY layer has been reaching to its limits, and also the dense network scenarios has become one of the main challenges, MAC layer improvements for 802.11 based WLANs has been emerged. To this end, IEEE 802.11ax [(High Efficiency WLAN (HEW)) task group [2] has started the new standardization activity, which aims at enhancing PHY and MAC layers in a radical way and is expected to end at 2020.

The main target of 802.11ax is to increase MAC throughput per station at least four times compared to 802.11ac, while providing spectral efficiency and improved energy saving in dense networks. The MAC structure

*Correspondence: mehmet.karaca@tedu.edu.tr

of the previous 802.11 WLANs is mostly based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism that is simple but inefficient especially when the network is dense and the traffic is high. As a result, new MAC structure has started to be developed, and orthogonal frequency division multiple access (OFDMA) has been selected as one of the new access schemes for 802.11ax to improve the performance of dense networks by offering multiuser transmission capability. Also, multiuser MIMO (MU-MIMO) that enables simultaneous multiple transmission at the same time and frequency resource can be realized for uplink transmission with 802.11ax [3].

Moreover, in order to improve the energy efficiency of STAs, the IEEE 802.11ax utilizes a new power saving mechanism which does not exist with the previous WLANs standards but was already introduced in 802.11ah [4, 5], which is called target wake time (TWT) mechanism. With TWT mechanism each STA and access point (AP) agree on a common set of parameters that include the periodicity of wake-up (i.e. TWT interval) and how long they stay awake. This mechanism allows STAs to receive/transmit data at only the wake-up times, and in the rest of the time they can stay in sleep mode. There are two types of TWT mechanism defined in the standard. With broadcast TWT mechanism, the TWT parameters are carried on beacon frames and STAs need to wake up to receive the beacon in order to know their TWT information.

With individual TWT, STAs can have different wake-up period depending on their traffic conditions [6], which is one of the key differences from the previous power saving mechanisms that STAs can have only a common beacon period as their wake-up interval. Also, the difference from 802.11ah is that as multiuser transmission is possible with 802.11ax, multiple STAs can share the same wake-up period and simpler scheduling scheme among STAs can be developed to allocate TWT intervals to STAs and hence the contention between STAs can be reduced significantly with individual TWT mechanism [6]. We note that the assignment of wake-up periods to STAs and the scheduling are up to the implementation and not specified in the standard. Another feature of TWT mechanism is that the agreement between STAs and the AP on the TWT parameters can be done for uplink, downlink or both of these transmissions. In this paper, for individual TWT mechanism we focus on the joint TWT interval assignment and scheduling to further reduce energy consumption with a careful consideration of traffic condition of each STA.

Since the early announcement of the new IEEE 802.11 standard, the attention from both industry and academia to 802.11ax has increased, and the work [7, 8] explain and investigate the new features coming with 802.11ax but the interest in these studies is not the power saving mechanism. There is a rich literature on the power saving mechanisms of 802.11 based networks. However, 802.11ax comes with its unique features and one of them is TWT mechanism. This mechanism is introduced to save more power by allowing users to have different sleeping periods. In [9, 10], the authors are interested in reducing power consumption of 802.11ax based networks but by not directly attacking the TWT mechanism. Specifically, the work in [9] tries to save energy by developing a scheduling scheme for the case when OFDMA is used. Also, a cloud-based centralized management is proposed for the energy saving of a dense network with many 802.11ax based APs in [10]. The "clock drift" problem which is the misalignment of the scheduling times of STAs within TWT mechanism and its effect on the power consumption is investigated in [11]. The most relevant studies to our work are found in [6, 12, 13]. A detailed explanation and performance assessment are given in [6], and it also highlights that a collision-free transmission can be achieved by a proper employment of a TDMA-type scheduling for TWT mechanism, which is left as a open problem. Also, in [6] a simple scheduling algorithm, which assigns users to TWT sessions randomly, is given but this algorithm shows only the performance of a generic TWT scheme without having any sophisticated mechanism (i.e. no consideration of dynamic network conditions). In [12], the optimization of

broadcast TWT mechanism is performed for energy saving and the authors in [13] aims at reducing energy for uplink multiuser transmission. The works in [12, 13] do not attempt to optimize TWT interval of STAs (e.g., wake-up period) and also do not consider a scenario where different STA may have different traffic conditions. Also in [12, 13], a full-buffer scenario is assumed, in which the traffic conditions of users is same and they have always data, and the authors try to address the problem of reducing congestion. It is still an open problem that how TWT mechanism of 802.11ax should be optimized according to time-varying network conditions to save more power and to control the users' traffic. In our work, we attack this problem and take into account the fact that users traffic conditions change over time and hence TWT interval is dynamically determined. Moreover, a scheduling algorithm which assigns users to the most suitable TWT interval depending on their dynamic traffic and channel conditions is designed.

Our contributions are summarized as follows:

- We aim at reducing the energy consumption of 802.11ax network where individual TWT is employed as the power saving mechanism. We improve the system model of the individual TWT by giving more flexibility for the assignment of TWT interval.
- We formulate the problem of joint multiuser scheduling and TWT interval assignment for energy saving as a stochastic optimization problem by taking into account the traffic condition of each STA. Also, we propose an algorithm that solves our problem by assigning TWT intervals to each STA dynamically. We find upper bounds on the optimal energy consumption and queue sizes of STAs via Lyapunov optimization framework.
- Through simulations, we demonstrate the performance of our algorithm by comparing it with a benchmark algorithm that randomly assigns TWT intervals to STAs. We also depict the trade-off between the average energy consumption and the average queue delay.

2. System model and problem definition

We consider a fully-connected 802.11ax based WLAN network where there is an AP serving M users. We assume random channel gains between the AP and STAs that are independent across time and STAs. As there is only a discrete finite set of F Modulation and coding Schemes (MCS) available in practice, only a fixed set of data rates denoted as $\mathcal{R} = r_1, r_2, \dots, r_F$ can be supported.

Recall that with the new sleeping and power saving mechanism introduced in 802.11ax, each STA and AP agree a common set of TWT parameters (individual TWT), and STAs can only wake up at the specified time to minimize their energy consumption. Next, we explain two important TWT parameters that are most relevant to this work and also represented in Figure 1:

- TWT session: the duration of time (in seconds) over that a STA wakes up and receive or transmit data. It is assumed to be same for all STAs and denoted as t_{up} .
- TWT interval: the time interval between two consecutive wake-up time of a STA.

More specifically, STAs wake up periodically to be served by the AP. We consider individual TWT with which different STAs can have different periodicity depending on their traffic and application requirements. Also, unlike previous studies, we consider that this periodicity is not static but can be changed dynamically on the agreement of STAs and the AP. Specifically, at the beginning of a epoch that starts at $t = nT$

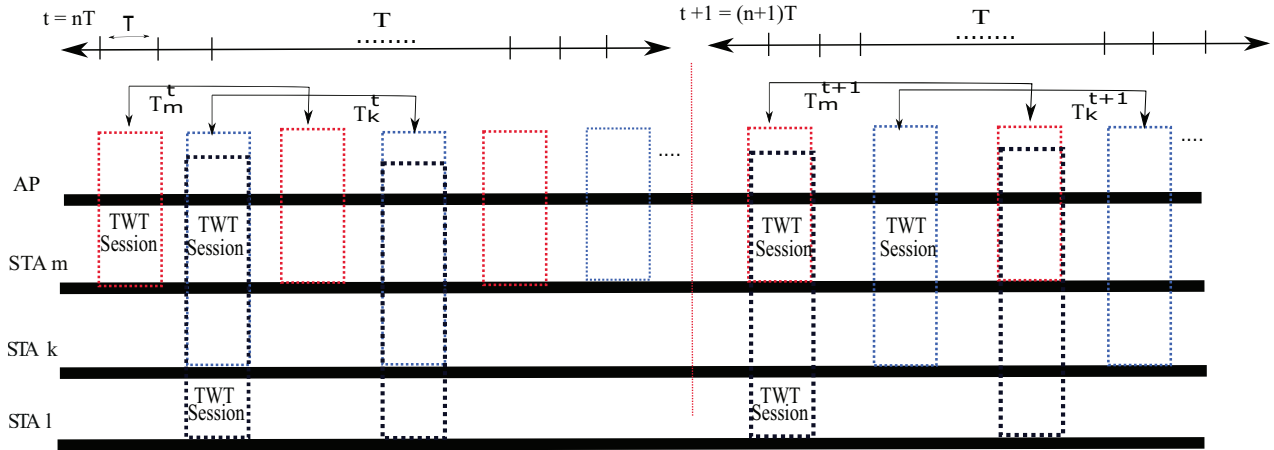


Figure 1. Dynamic TWT interval assignment.

where $n = 1, 2, \dots$, TWT Wake-up interval of a STA m denoted as T_m^t is determined. Then, the STA m wakes up at every T_m^t until the next epoch. Each epoch lasts T seconds, where $T > T_m^t$ for all t and m . At the beginning of the next epoch, T_m^t is updated if necessary.

For instance, as shown in Figure 1, the TWT wake-up interval for STA m at the beginning of epoch n (i.e. at time $t = nT$) is T_m^t . However, at the next epoch, $t + 1 = (n + 1)T$ it is set to $T_m^{t+1} > T_m^t$, and STA m has less number of TWT sessions and more time to sleep. To depict the integration of the multiuser capability in TWT, we also show in Figure 1 that STA k and STA l have the same TWT interval at time t and wake up at the same time.

TWT wake-up interval takes values from a predefined discrete set, $T_m^t \in \{T_1, T_2, \dots, T_L\}$ for all m and t . For a given T_m^t , the number of TWT sessions that STA m can utilize at time $t = nT$ (over epoch n) is given as:

$$N_m^t = \left\lfloor \frac{T}{T_m^t} \right\rfloor,$$

where $\lfloor x \rfloor$ is the largest integer that is smaller than x . Since T_m^t takes only discrete values from a finite set, T as a system parameter and the set $\{T_1, T_2, \dots, T_L\}$ can be chosen such that $N_m^t = \frac{T}{T_m^t} - 1$. We further assume that each T seconds is divided into mini-slots denoted as τ , $\tau \in \{0, 1, 2, \dots\}$, and τ is multiple integer of all possible values of T_m^t . We assume that τ is equal to t_{up} .

We now determine the energy consumption for STA m at a mini time slot τ . Note that when STA m wakes up, it becomes awake for a duration of t_{up} seconds. t_d and t_u are the fractions of t_{up} that are used for downlink and uplink transmission, respectively. Also, the power consumption during downlink and uplink are denoted as P_d and P_u , respectively. Then, the total energy consumption during a single TWT session (e.g., mini-slot τ) is given by,

$$E_s = P_d \times t_d t_{up} + P_u \times t_u t_{up}.$$

¹Another parameter of TWT mechanism is the first time for a STA to start TWT session. Here, for analytical simplicity we assume TWT interval can fully represent the number of TWT sessions.

Note that when $\tau = iT_m^t$, where $i > 0$ is an integer, STA m wakes up to start a TWT session. When $\tau \neq iT_m^t$, STA m is at sleep mode. The energy consumption of STA m at a mini slot τ is given as,

$$E_m(\tau) = \begin{cases} E_s & ; \text{if } \tau = iT_m^t \\ E_{sleep} & ; \text{otherwise,} \end{cases} \quad (1)$$

where E_{sleep} is the energy consumed at sleep mode over τ seconds and $E_{sleep} = P_{sleep} \times \tau$, where P_{sleep} is the power consumption at sleep mode.

During the epoch n starting at $t = nT$, there are N_m^t number of TWT sessions for STA m . Thus, the STA m consumes E_m^t amount of energy over the epoch n , where

$$\sum_{\tau=1}^{T/\tau} E_m(\tau) = E_m^t = E_s \times N_m^t + \left(\frac{T}{\tau} - N_m^t \right) E_{sleep}, \quad (2)$$

where $\left(\frac{T}{\tau} - N_m^t \right) E_{sleep}$ accounts for the total energy consumption of STA m at sleep mode since there are $\left(\frac{T}{\tau} - N_m^t \right)$ mini-slots for sleeping over the epoch. Then, the time average energy consumption over all STAs is determined as follows:

$$E_{avg} = \limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{i=0}^{t-1} \sum_{m=1}^M \mathbb{E}[E_m^t]. \quad (3)$$

At each mini-slot τ , data arrives to the queue of each STA. Let $A_m(\tau)$ be the amount of data (i.e. in bits) arriving into the queue of STA m at slot τ . We assume that $A_m(\tau)$ is a stationary process and it is independent across STAs and time slots. We denote the arrival rate vector as $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_M)$, where $\lambda_m = \mathbb{E}[A_m(\tau)]$. Let $\mathbf{Q}(\tau) = (Q_1(\tau), Q_2(\tau), \dots, Q_M(\tau))$ denote the vector of queue sizes, where $Q_m(\tau)$ is the queue length of user m at time slot τ . A queue is strongly stable if

$$\limsup_{S \rightarrow \infty} \frac{1}{S} \sum_{\tau=0}^{S-1} \mathbb{E}(Q_m(\tau)) < \infty. \quad (4)$$

Moreover, if every queue in the network is stable then the network is called to be stable. The dynamics of the queue of STA m is

$$Q_m(\tau) = \max[Q_m(\tau) - R_m(\tau), 0] + A_m(\tau). \quad (5)$$

We note that the transmission rate of $R_m(\tau)$ can take positive value, i.e. $R_m(\tau) > 0$ at time instances $\tau = iT_m^t$ as STA m wakes up only at these instances. When $\tau \neq iT_m^t$, then $R_m(\tau) = 0$. Λ denotes the capacity region of the network, which is the largest possible set of rates $\boldsymbol{\lambda}$ that can be supported by a joint scheduling and TWT interval assignment algorithm with ensuring the network stability, $\boldsymbol{\lambda} \in \Lambda$.

As the multiuser transmission capability of 802.11ax is enhanced with the introduction of OFDMA and MU-MIMO in uplink, more than one STA can be assigned to a TWT interval. We assume that at most K STAs can share the same TWT interval, and let $N_{T_l}(t)$ be the number of STAs that are assigned to TWT interval T_l at time t , $l \in \{1, 2, \dots, L\}$. Therefore, in total up to $L \times K$ STAs can have a TWT interval assigned over each epoch. With our setup, the control decision of the network for STA m is the determination of T_m^t .

That is to say at the beginning of each time t (i.e. epoch), the network determines the TWT wake-up interval for each STA. Then, we consider the following optimization problem:

$$\min E_{avg} \tag{6}$$

$$\text{s.t.:1) Network stability} \tag{7}$$

$$2) T_m^t \in \{T_1, T_2, \dots, T_L\} \quad \forall t, m \tag{8}$$

$$3) N_{T_l}(t) \leq K \quad \forall l \in \{1, 2, \dots, L\}, \forall t \tag{9}$$

The constrain in (8) ensures that TWT interval decided for a STA m takes values from a predefined discrete set. Also, the third constraint in (9) guarantees that at most K STAs can be assigned to a TWT interval. At the beginning of each epoch, the system determines TWT wake-up interval for $L \times K$ number of STAs and other STAs can go to sleep mode during whole T seconds to save energy.

The problem (6)–(9) aims at minimizing the total average energy consumption of STAs by properly choosing TWT wake-up interval with the consideration of their traffic conditions. Intuitively, it is expected that the solution of the problem must strike a balance between the energy consumption and queue size. In other words, when the channel condition is good and queue size is large it can reduce the TWT interval, and when queue size is low it prefers a higher TWT interval for more energy saving. The problem constitutes a stochastic optimization problem and we next propose a solution based on Lyapunov optimization technique [14].

3. Joint TWT interval and scheduling algorithm (JTWSA)

In this work, we employ Lyapunov drift and optimization tools [14] that enables us to deal with performance optimization and network stability problems together in a unified framework. The Lyapunov optimization tool allows us to minimize the drift (i.e. stabilizing the network) and optimize the given objective (i.e. minimizing energy) simultaneously. Also, it provides an online algorithm that only depends on the current network information (i.e. no future information such as channel and traffic conditions of the users is needed). In order to use this tool, first, a Lyapunov function is determined, and it is common in literature to choose a square function of queue sizes as the Lyapunov function:

$$L(\mathbf{Q}(t)) \triangleq \frac{1}{2} \sum_{m=1}^M Q_m^2(t)$$

Lyapunov function acts as a measurement tool for the total queue sizes in the network. Another important concept in this framework is Lyapunov drift that measures how much the expected values of the queue sizes vary within a one single slot. Here, we use the conditional T-slot Lyapunov drift that is the expected variation in the Lyapunov function over T slots and it is given as:

$$\Delta_T(t) \triangleq \mathbb{E} \{L(\mathbf{Q}(t+T)) - L(\mathbf{Q}(t)) | \mathbf{Q}(t)\}. \tag{10}$$

To stabilize the network, we need to make sure that the drift does not grow indefinitely and hence ultimate aim is to minimize the drift to obtain a stable network. For a more tractable analysis, an upper bound for the drift is determined and then this bound is minimized.

In order to handle both the network stability and an objective optimization together, drift-plus penalty method [14] is employed. With this method, our penalty (or cost) $V \sum_{m=1}^M \mathbb{E}[E_m^t | \mathbf{Q}(t)]$ (i.e. T slot drift-plus-

penalty) is added to the drift (10):

$$\Delta_T(t) + V \sum_{m=1}^M \mathbb{E}[E_m^t | \mathbf{Q}(t)], \tag{11}$$

where V is a control parameter between the queue delay and the average system penalty. Now, in the Lyapunov optimization framework, we turn our energy minimization and network stability problem to be the minimization of (11). Instead of directly attacking to minimize (11), in the following Lemma we first derive an upper bound on (11) and then we propose our algorithm that minimizes the bound.

Lemma 1 *Given $V > 1$, and at time $t = nT$, for any feasible control decision $(T_1^t, T_2^t, \dots, T_M^t)$, we have the following bound,*

$$\begin{aligned} \Delta_T(t) + V \sum_{m=1}^M \mathbb{E}[E_m^t | \mathbf{Q}(t)] \leq & B_1 + \mathbb{E} \left[\sum_{\tau=t}^{t+T-1} \sum_{m=1}^M Q_m(\tau) A_m(\tau) | \mathbf{Q}(t) \right] \\ & - \mathbb{E} \left[\sum_{\tau=t}^{t+T-1} \sum_{m=1}^M Q_m(\tau) R_m(\tau) - V \sum_{m=1}^M E_m^t | \mathbf{Q}(t) \right] \end{aligned} \tag{12}$$

where $B_1 = \frac{MT(R_{max}^2 + A_{max}^2)}{2}$.

Proof The proof starts with finding an upper bound for the Lyapunov drift given in (10) by using the following fact: for user m , the following inequality holds.

$$Q_m^2(t+1) - Q_m^2(t) \leq R_m^2(t) + A_m^2(t) + 2Q_m(t)[A_m(t) - R_m(t)] \tag{13}$$

By summing (13) over $[t, t+T-1]$ and knowing that $R_m(t) \leq R_{max}$ and $A_m(t) \leq A_{max}$ for all m and t we obtain

$$Q_m^2(t+T) - Q_m^2(t) \leq TR_{max}^2 + TA_{max}^2 - 2 \left[\sum_{\tau=t}^{t+T-1} Q_m(\tau)[R_m(\tau) - A_m(\tau)] \right]. \tag{14}$$

Then, by taking the conditional expectation of (14) with respect to $\mathbf{Q}(t)$ and summing over all users, and dividing by 1/2 we have

$$\Delta_T(t) \leq B_1 - \mathbb{E} \left[\sum_{\tau=t}^{t+T-1} \sum_{m=1}^M Q_m(\tau)[R_m(\tau) - A_m(\tau)] | \mathbf{Q}(t) \right]. \tag{15}$$

Finally, we add the penalty term $V \sum_{m=1}^M \mathbb{E}[E_m^t | \mathbf{Q}(t)]$ to both sides of (15) and rearranging the resulting terms, we have the bound in (12). This completes the proof. \square

The Lyapunov optimization framework now helps to solve our primary problem by finding an algorithm that minimizes the right hand side (R.H.S) of the bound (12). Clearly, the solution of following problem minimizes the R.H.S of (12).

Problem 1:

$$\max_{T_1^t, T_2^t, \dots, T_M^t} \mathbb{E} \left[\sum_{\tau=t}^{t+T-1} \sum_{m=1}^M Q_m(\tau)[R_m(\tau) - A_m(\tau)] - V \sum_{m=1}^M E_m^t | \mathbf{Q}(t) \right]$$

However, the solution of Problem 1 requires a prior knowledge of queue size and transmission rates of STAs, i.e. future information. Particularly, obtaining the future traffic information is not easy, and hence we follow the same idea proposed in [15] to find a looser but more relaxed bound without needing the future queue size information. With this idea, we approximate the future queue sizes as the current observation, i.e. $Q_m(\tau) = Q_n(t)$ for all $\tau \in [t, t + T - 1]$ for all STA m , and now we aim at maximizing the following problem:

Problem 1.1:

$$\max_{T_1^t, T_2^t, \dots, T_M^t} \mathbb{E} \left[\sum_{\tau=t}^{t+T-1} \sum_{m=1}^M Q_m(t) R_m(\tau) - V \sum_{m=1}^M E_m^t | \mathbf{Q}(t) \right]$$

Now, the solution of Problem 1.1 needs only queue size information at the beginning of $t = nT$ but it still needs future transmission rates over T seconds. Fortunately, for WLANs, the coherence time of the channel is long enough since STAs are usually stationary and the mobility is low. Also, transmission rates can be estimated with the advanced machine learning techniques [16]. We assume that the transmission rates change every T seconds, hence the Problem 1.1 is reduced to the following deterministic optimization problem as follows:

Problem 1.2:

$$\max_{T_1^t, T_2^t, \dots, T_M^t} \sum_{m=1}^M Q_m(t) R_m(t) N_m^t - V E_m^t \tag{16}$$

By using (2), we can rewrite (16) as

$$\max_{T_1^t, T_2^t, \dots, T_M^t} \sum_{m=1}^M Q_m(t) R_m(t) N_m^t - V N_m^t (E_s - E_{sleep}). \tag{17}$$

The following joint TWT assignment and scheduling algorithm (JTWSA) solves (17).

JTWSA :

- Input: $V, P_d, P_u, P_{sleep}, t_{up}$. At every $t = nT$ do:
- Step 1: Among all users, find

$$\mathcal{S}(t) = \operatorname{argmax} -(L \times K) \{Q_m(t) R_m(t) - V(E_s - E_{sleep})\},$$

where $\operatorname{argmax} -(L \times K)$ is the operation of choosing the first $L \times K$ elements of a set of real numbers sorted in decreasing order.

- Step 2:
 - Start assigning TWT intervals T_1, T_2, \dots, T_L sorted in increasing order to the STAs in $\mathcal{S}(t)$ if only $Q_m(t) R_m(t) > V(E_s - E_{sleep})$. For instance, the TWT interval of the first K STAs in $\mathcal{S}(t)$ is set to T_1 , for the next K STAs in $\mathcal{S}(t)$, it is set to T_2 so on. If the number of STAs is assigned to TWT interval is K , go to the next TWT interval.
 - If $Q_m(t) R_m(t) \leq V(E_s - E_{sleep})$, set the TWT interval for that STA m to T .
- Step 3: After determining T_m^t for every STA, perform transmit/receive operation and updates queue until next $t = (n + 1)T$.

As JTWSA optimally minimizes the R.H.S of (12), we have the following Theorem that shows performance of JTWSA.

Theorem 1 (*Lyapunov Optimization*) *Suppose λ is an interior point in Λ , then there exists a constant $\epsilon > 0$ such that $\lambda + \epsilon \in \Lambda$. Then, under JTWSA, we have the following bounds:*

$$\limsup_{T' \rightarrow \infty} \frac{1}{T'} \sum_{t=0}^{T'-1} \mathbb{E}[E_m^t] \leq E_{avg}^* + \frac{B_2}{V} \tag{18}$$

and

$$\limsup_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m=1}^M \mathbb{E}[Q_m(nT)] \leq \frac{B_2 + VE_{max}}{\epsilon}, \tag{19}$$

where E_{avg}^* is the optimal solution of problem (6)-(7)-(8)-(9), $B_2 = \frac{MT^2(R_{max}^2 + A_{max}^2)}{2}$ and $E_{max} = MTE_s$.

Proof The proof has a well-established structure in the literature and to avoid redundancy, we omit the details here and refer the readers to the proof in Theorem 5.4 of [14] and Theorem 2 in [15]. The sketch of the proof is as follows: first, it shows the existence of a stationary randomized algorithm that is optimal and achieves the minimum time average energy by choosing T_m^t independent from $\mathbf{Q}(t)$ but according to a fixed probability distribution known to the network. Then, it is shown that JTWSA is better than this randomized algorithm in minimizing the R.H.S of (12) and thus it is also optimal. \square

Theorem 1 implies that the average energy under JTWSA approaches to the minimum optimal energy E_{avg}^* as V increase, while the average queue length also increase with higher values of V . Next, we verify our findings with simulation results.

4. Numerical results

We consider an IEEE 802.11 ax based WLAN where there is an AP serving $M = 50$ STAs. We assume that all TWT sessions are used for uplink transmissions and choose the parameters given in Table. We create our simulation environment and obtain our results in MATLAB.

The network operates as follows: at the beginning of each T seconds, the queue sizes and transmission rates for each STA is determined, and then the network assigns TWT intervals to STAs according to JTWSA. At most $K = 5$ STAs can be assigned to same TWT interval thanks to the multiuser transmission capability of 802.11ax. As there are $L = 9$ different TWT intervals and $K = 5$, at most 45 STAs can wake up for transmission over each epoch and other 5 STAs go to sleep mode during entire T seconds. After assigning TWT intervals, STAs wake up at the determined intervals and transmit their data to AP. We compare JTWSA with a benchmark algorithm that assigns TWT intervals to STAs randomly at the beginning of every T seconds. Next, we show the performance of JTWSA and the benchmark algorithm in terms of average queue sizes and energy consumption.

Figures 2a and 2b depict the total average queue size and energy consumption under JTWSA and the benchmark algorithm when T is 1 s, respectively. From 2a, we see that as the arrival rate increases the total average queue size increases as well. However, there is a sudden increase in the queue size with the benchmark algorithm when the arrival rate is around 1, which indicates that the network is not stable any more and the benchmark cannot support any arrival rate higher than 1. On the other hand, JTWSA can keep stabilizing the

Table . Simulation parameters and values.

Simulation parameters	Values
T	1 s, 2 s
P_u	1 W
P_{sleep}	0.15 W
t_{up}	1 ms
t_d	0
t_u	1
R	[10, 20, 50, 100, 150, 200] Mbps
M	50 STAs
K	5 STAs
V	[1000, 5000]
λ	Poission traffic. Arrival period [5 4.5 4 3.5 3 2.5 2 1.5 1 0.5] s
File size	25 KBytes
TWT values	Total 9 nonoverlapping TWT intervals. Minimum and maximum TWT intervals: 50 ms and 450 ms

network up to the arrival rate of 2. Theorem 1 points that when the JTWSA operates with higher V values, a larger queue size is resulted in. Figure 2a verifies that the queue size with $V = 5000$ is higher compared to the case when $V = 1000$.

Figure 2b shows the total average energy consumption of JTWSA and the benchmark algorithm. Note that STAs can wake up even their queue sizes and data rates are low with the benchmark algorithm and hence the energy conservation is expected to be lower. This intuition is demonstrated in Figure 2b where the energy consumption with the benchmark is higher than JTWSA. Also, as Theorem 1 suggests that as we increase V , we get closer to the minimum energy consumption with JTWSA. Clearly, from Figure 2b, a higher V value yields lower energy consumption for JTWSA. Therefore, by choosing the control parameter V properly, a desired trade-off between queue delay and energy consumption can be achieved.

Figures 3a and 3b depict the impact of a higher value of T on the queue size and energy consumption. Specifically, we set $T = 2$ s and according Theorem 1, both the queue size and energy consumption increase as we increase T since B_2 depends on T . By comparing Figures 2a and 3a, one can conclude that the queue size is larger with higher values of T . Similar result can be deduced for the energy consumption from Figures 2b and 3b. We note that in order to close the minimum energy consumption when T is high, we also need a higher V values. It can also be seen that the impact of a higher T value on the energy consumption is less significant but the impact on the queue size cannot be ignored. Although choosing lower T values seems to be better in terms of the average queue size and energy consumption, it will increase signaling overhead since the AP must communicate and agree with each STA on the TWT interval at every T second. This trade-off can be resolved given the operator preferences for a target objective and performance. Lastly, Figure 4 shows how long the users stay in sleep mode with different arrival rates, V values and the algorithms. With the benchmark algorithm, the average sleep duration does not depend on the traffic condition since the TWT intervals are assigned randomly, and the average sleep duration is around 328 ms. That is to say, more than 60% of the time the users under the benchmark algorithm stay active. On the other hand, the users under JTWSA stay in active mode depending on their traffic conditions. Specifically, when the traffic is low (i.e. arrival rate is low

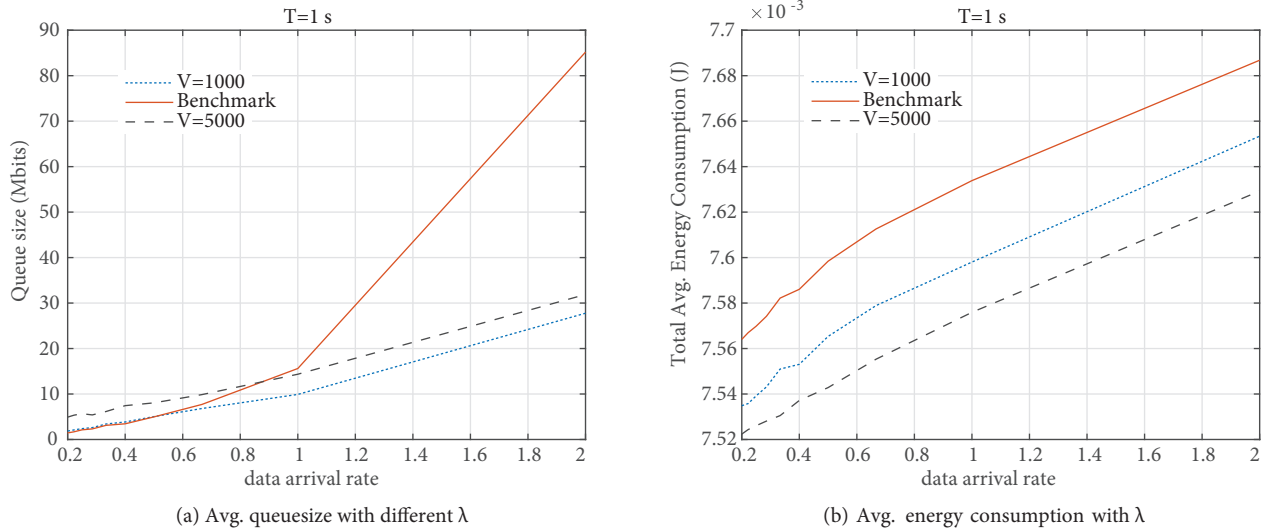


Figure 2. Queue sized and energy consumption when T = 1.

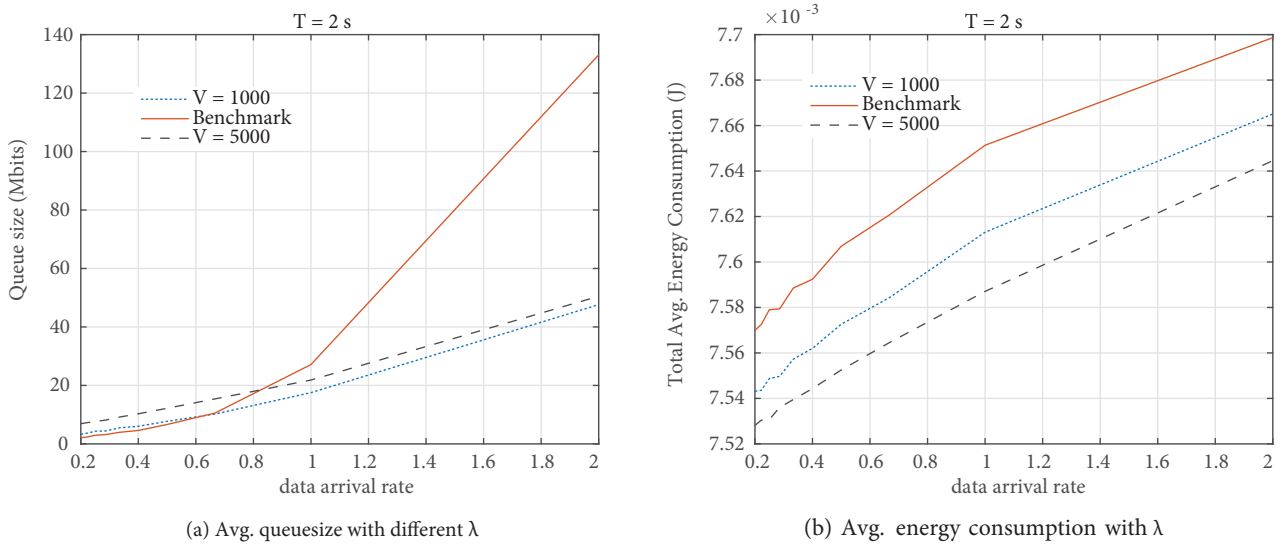


Figure 3. Queue sized and energy consumption when T = 2.

), for instance the arrival rate is 0.2, the users stay in sleep mode more than 80% of the time. As the traffic rate increases, they tend to stay in active mode more often in order to empty their queues and as a result, their sleep duration decreases. This also explains why JTWSA consumes less power than the benchmark algorithm. Also, with higher values of V , the users stay in sleep mode longer and hence save more energy.

5. Conclusion

In this paper, we have developed an algorithm that aims at improving the energy efficiency of the power saving mechanism of IEEE 802.11ax by optimization TWT intervals of STAs according to their traffic and channel conditions. Our algorithm assigns lower TWT interval when queue size is large and/or transmission rate is

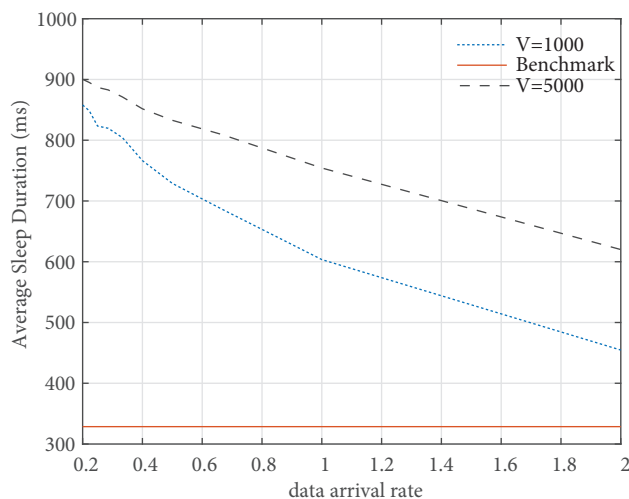


Figure 4. Average sleep duration with different traffic conditions.

high, and allows more sleeping time if traffic is low enough. We have analyzed the algorithm via Lyapunov optimization framework and found performance bounds on the energy consumption and queue sizes. We have defined two important trade-off: first, high energy saving can be achieved when a large average queue delay is allowed. Also, lower average delay can be obtained if TWT intervals are updated more frequently but the cost in this case is high signaling overhead. The impact of inaccurate transmission rates prediction on the performance of our algorithm is worth to be investigated. Also, in a multi-WLAN scenario where interference is significant, determining TWT intervals and channel allocation would be an interesting research problem.

References

- [1] IEEE Std 802.11ac-2013. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications—Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz. USA: IEEE, 2013.
- [2] Draft IEEE Standard P802.11. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications—Amendment 6: Enhancements for High Efficiency WLAN. D3.0. USA: IEEE, 2018.
- [3] Khorov E, Kiryanov A, Lyakhov A, Bianchi G. A Tutorial on IEEE 802.11ax High Efficiency WLANs. *IEEE Communications Surveys & Tutorials* 2019; 21 (1): 197-216. doi: 10.1109/COMST.2018.2871099
- [4] Khorov E, Lyakhov A, Krotov A, Guschin A. A survey on IEEE 802.11 ah: an enabling networking technology for smart cities. *Computer Communication* 2015; 58: 53-69. doi: 10.1016/j.comcom.2014.08.008
- [5] 802.11ah-2016-2017. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation. USA: IEEE, 2017.
- [6] Nurchis M, Bellalta B. Target wake time: scheduled access in IEEE 802.11ax WLANs. *IEEE Wireless Communications* 2019; 26 (2): 142-150. doi: 10.1109/MWC.2019.1800163
- [7] Deng D, Lin Y, Yan X, Zhu J, Li Y et al. IEEE 802.11ax: highly efficient WLANs for intelligent information infrastructure. *IEEE Communications Magazine* 2017; 55 (12): 52-59. doi: 10.1109/MCOM.2017.1700285
- [8] Bellalta B. IEEE 802.11ax: High-efficiency WLANs. *IEEE Wireless Communications* 2016; 23 (1): 38-46. doi: 10.1109/MWC.2016.7422404

- [9] Karaca M, Bastani S, Priyanto BE, Safavi M, Landfeldt B. Resource management for OFDMA based next generation 802.11 WLANs. In: IFIP Wireless and Mobile Networking Conference; Colmar, France; 2016. pp. 57-64. doi: 10.1109/WMNC.2016.7543930
- [10] Khorov E, Kiryanov A, Krotov A. Cloud-based Management of Energy-Efficient Dense IEEE 802.11ax Networks. In: IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom); Sochi, Russia; 2019. pp. 1-5. doi: 10.1109/BlackSeaCom.2019.8812787
- [11] Bankov D, Khorov E, Lyakhov A, Stepanova E. Clock Drift Impact on Target Wake Time in IEEE 802.11ax/ah Networks. In: Engineering and Telecommunication (EnT-MIPT); Moscow, Russia; 2018. pp. 30-34. doi: 10.1109/EnT-MIPT.2018.00014
- [12] Chen Q, Weng Z, Xu X, Chen G. A target wake time scheduling scheme for uplink multiuser transmission in IEEE 802.11ax-based next generation WLANs. IEEE Access 2019; 7: 158207-158222. doi: 10.1109/ACCESS.2019.2950464
- [13] Yang H, Deng D, Chen K. On energy saving in IEEE 802.11ax. IEEE Access 2018; 6: 47546-47556. doi: 10.1109/ACCESS.2018.2865763
- [14] Georgiadis L, Neely MJ, Tassiulas L. Resource Allocation and Cross-Layer Control in Wireless Networks. Boston, USA: Foundations and Trends in Networking, 2006.
- [15] Fang W, Yao Y, Huang L, Sharma A. B, Golubchik L et al. A comment on power cost reduction in distributed data centers: a two time scale approach for delay tolerant workloads. IEEE Transactions on Parallel and Distributed Systems 2015; 26 (5): 1495-1496.
- [16] Kulkarni A, Seetharam A, Ramesh A, Herath JD. DeepChannel: wireless channel quality prediction using deep learning. IEEE Transactions on Vehicular Technology 2020; 69 (1): 443-456. doi: 10.1109/TVT.2019.2949954