Wake-up Scheduling for Energy-Efficient Mobile Devices

Soheil Rostami, Hoang Duy Trinh, Sandra Lagen, Mário Costa, *Member, IEEE*, Mikko Valkama, *Senior Member, IEEE*, and Paolo Dini

Abstract

Recently, discontinuous reception mechanisms (DRX) and wake-up schemes (WuS) have been proposed to enhance the energy efficiency of 5G mobile devices and prolong the battery lifetime. The existing DRX and WuS use commonly pre-configured parameters that cannot be adjusted dynamically. In this paper, a novel wake-up scheduling (WuSched) concept is introduced to further improve the energy efficiency of WuS-enabled mobile devices while controlling the buffering delay in a dynamic manner. The main idea of WuSched is to use a fixed configuration of the wake-up scheme and adjust the scheduling of the wake-up signals dynamically based on actual traffic arrivals. For this purpose, two different optimization approaches of the wake-up scheduling concept are proposed, analyzed, and compared, namely offline and online wake-up schedulers (WuSched-Offline and WuSched-Online). First, the WuSched-Offline is analyzed analytically for Poisson traffic arrivals and optimized (offline) to balance the average delay and power consumption. Second, the WuSched-Online is proposed to take online decisions based on traffic prediction, which is able to deal with general and more complex traffic models. Towards this end, we develop a framework for the prediction of packet arrivals based on recurrent neural networks. Numerical results show that both wake-up schedulers outperform the ordinary WuS-based system where wake-up scheduler is not deployed. In particular, for predefined delay requirements of video streaming, audio streaming, and mixed traffic flow, the WuSched-Online reduces the power consumption of the baseline WuS by up to 36%, 28% and 9%, respectively. Results also show that the WuSched-Offline has slightly better energy efficiency than the WuSched-Online in the case of Poisson packet arrivals, as it is optimized for that, while its power consumption is slightly higher than that of the WuSched-Online scheduler for realistic traffic scenarios.

S. Rostami is with Huawei Technologies Oy (Finland) Co. Ltd, Helsinki, Finland, and also with Tampere University, Finland. E-mail: soheil.rostami1@huawei.com, soheil.rostami@tuni.fi.

H. Trinh, S. Lagen and P. Dini are with Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Barcelona, Spain. E-mails: {hdtrinh, sandra.lagen, paolo.dini}@cttc.es.

M. Costa is with Huawei Technologies Oy (Finland) Co. Ltd, Helsinki, Finland. E-mail: mariocosta@huawei.com.

M. Valkama is with the Department of Electrical Engineering, Tampere University, Finland. E-mail: mikko.valkama@tuni.fi.

Limited subset of initial results presented at IEEE ICC 2020 [1].

Index Terms

5G, machine learning, wake-up scheme, energy efficiency, scheduling, LSTM.

I. Introduction

The emerging fifth generation mobile networks (5G) have a promising capability to offer super-fast and ultra-low latency connectivity to the end users, and are expected to enable a wide range of futuristic mobile applications and services such as augmented/virtual reality, cloud gaming, and ultra-high-definition video streaming [2]. Such magnificent improvements are vital to accommodate the ever-growing needs for increased data rates and enhanced quality-of-service (QoS). In particular, they are realized in New Radio (NR) based 5G systems by adopting larger transmission bandwidths, higher modulation orders, advanced coding techniques, and sophisticated multi-antenna schemes [3]. However, the utilization of such computationally intensive techniques comes commonly at the cost of higher energy consumption that can deplete the mobile devices' battery power rather quickly, which in itself is one of the major causes of dissatisfaction for the users [4].

In general, the cellular modem is one of the primary energy-consuming elements of mobile devices, while the other units only contribute when they are used intensively [5], [6]. Furthermore, in current and future traffic trends, the data traffic of mobile users is mainly downlink-dominated [7]. Therefore, the development of power-saving mechanisms for cellular modems in receive mode has paramount importance in order to extend the mobile devices' functionalities in 5G networks and beyond. To this end, the 3rd generation partnership project (3GPP) has specified discontinuous reception (DRX) as the *de facto* power-saving mechanism for long-term evolution (LTE) based fourth-generation (4G) systems [8], [9] and NR based 5G systems [3], [10]. DRX enables the mobile device to reduce energy consumption by switching off the radio-frequency (RF) circuitry and other modules for long periods, activating them only for short intervals [11]. However, it has been shown in [12] that the time period for which a mobile device monitors the physical downlink control channel (PDCCH) without any data allocation has still a major impact on the battery consumption. Thus, further power-saving mechanisms are of large importance.

A. Wake-up based Access and State-of-the-Art

In the context of non-cellular networks, different power-saving mechanisms have been extensively studied and implemented, with specific focus on the low-power wide-area networks

(LPWAN) and wireless sensor networks (WSN) [13]. In this context, duty cycling has been the major mechanism for energy conservation in LPWANs/WSNs [14], [15]. In duty cycling, which resembles cellular DRX, nodes wake up and sleep periodically, thus leading to idle listening and potential overhearing. Therefore, to reduce idle listening, the concept of *wake-up radio* based access has been recently studied, e.g., [16], [17]. Demirkol *et al* [18] provided a comprehensive overview and insight into wake-up receiver (WRx), and investigated the benefits achieved with WRx along with the challenges observed in WSNs. In addition, they presented an overview of state-of-the-art hardware and networking protocol proposals as well as classification of WRx schemes. Moreover, authors in [19] introduced the concept of wireless-powered wake-up receiver, reducing the energy consumption of the wireless node considerably. The proposed receiver scavenges the RF energy from the received signal to power its sensor, communication and processing blocks. The proposed scheme can be utilized for a wide range of energy-constrained wireless applications such as wireless sensor actuator networks and machine-to-machine communications. Due to the large energy saving potential of such wake-up radio based methods, similar concepts are raising increasing interest also in cellular networks, primarily 5G NR [20], in which this paper is also focused on.

In order to reduce the energy consumption of unscheduled cycles in DRX, cellular wake-up schemes (WuS) have been recently proposed, e.g., in [5], [21]. In cellular WuS, or WuS for short, the mobile device monitors a narrow-band wake-up signaling periodically (every wake-up cycle) at specific time instants and subcarriers, which indicates to the device whether to process the upcoming PDCCH or remain in sleep mode. As soon as a packet arrives at the transmission buffer of the base station, the wake-up indicator is assumed to be sent at the next upcoming wake-up instant. Furthermore, a low-complexity WRx is required to decode the corresponding wake-up signaling and to acquire the necessary time and frequency synchronization [5], [22]. Additionally, in [22], synchronization is one of our main design factors in the design of wake-up signaling and WRx. To this end, we utilized built-in self-synchronizing signal structure and assumed high-power high-precision oscillator to remove the need for a separate synchronization stage for WRx. Our extensive simulation results [5], [22] verify that the proposed scheme can achieve very low misdetection (less than 1%) and false alarm rates for signal-to-noise ratios (SNRs) even below 0 dB. Furthermore, very high-quality synchronization can be obtained down to SNRs of -4 dB [22]. We also showed that the impact of such negligible errors is very low on power consumption and buffering delay. Furthermore, in our previous work [23], [24], we

introduced an offline method to optimize the WuS configuration (i.e., the wake-up cycle period) based on a delay bound under the assumption of Poisson traffic. In cases where traffic dynamics vary over time, the WuS optimization method in [23], [24] requires reconfiguration of the WuS parameters, which need to be communicated to the mobile device, and thus increases the control signaling overhead as well as the associated energy consumption.

B. Contributions and Novelty

In this paper, we introduce a novel concept called *wake-up scheduling* (WuSched) to further improve the energy efficiency of mobile devices in cellular networks. The main idea is in starting with a fixed WuS configuration and then adjusting the scheduling of the wake-up signals dynamically by determining whether to wake-up the mobile device or not. More precisely, in wake-up scheduling, the network does not send the wake-up indicator to the mobile device as soon as there is one (or more) packet arrival(s), but rather it may wait to send it while at the same time taking different QoS and other requirements into account, specifically the latency constraint and the mobile device power consumption. The proposed concept not only concerns to the physical layer (PHY), but mainly, it uses WuS as a mechanism to reduce energy consumption at PHY and then uses adequately scheduled wake-up signals from the medium access control (MAC) layer. In particular, offline and online optimizations of the wake-up scheduler parameters are proposed in this paper, namely WuSched-Offline and WuSched-Online. The offline optimization (WuSched-Offline) is based on the assumption that traffic arrivals follow a Poisson distribution and it is analyzed analytically. The objective is to reduce the power consumption of the mobile device while satisfying delay requirements. The optimal solution for the tunable operational parameter of the WuSched-Offline, which is referred to as the *buffer size threshold* and which only concerns the network side (so that it can be easily reconfigured based on traffic dynamics), is obtained in closed form. Then, for a general and thus very likely more complex traffic models, an online optimization is proposed through the WuSched-Online. It uses a proactive scheduler that takes decisions every wake-up cycle based on traffic predictions over a forecast horizon. A multi-step Long Short-Term Memory (LSTM) neural network is trained with data from real user applications and tailored for traffic prediction purposes. To the best of our knowledge, this is the first attempt to introduce online wake-up scheduling decisions with traffic prediction capabilities into the wake-up scheme. Unlike previous works [5], [23], [24], the WuSched-Online is not tied to any specific traffic models and operates dynamically.

Table I: Most important variables and mathematical operations used throughout the article.

Variable / Operation	Definition
PW_{wrx} / PW_{on} / PW_{off}	power consumption of WRx / modem at ON /OFF modes
t_p	inter-packet arrival time
t_r	residual time between γ^{th} packet arrival time and end of w-cycle
c	current TTI
λ	packet arrival rate
t_{su} / t_{pd}	start-up / power-down time of cellular module
t_w	wake-up cycle
t_i	maximum allowable length of inactivity timer
ω	length of inactivity timer
t_{on}	on-duration time
t_a	delay windows size
γ	buffer size threshold
e_t/t_t	overall energy consumption/length of transitional states
L	length of scheduling cycle
$L_e/L_d/L_a$	length of empty/dormant/active period
N	number of packet arrivals during scheduling cycle
N_d/N_a	number of packet arrivals during dormant/active period
T_n	inter-arrival times of nth and n+1th packets at gNB
A_n	packet arrival times of nth packets at gNB
D_n	nth packet's buffering delay
D_n	estimated delays of nth packet (buffered or forecast)
W_n	time duration between decoding nth and n+1th packets by UE
H_n	$(W_n-1)\big(W_n+1-2(T_n-D_n)\big)$
$C_0/C_1/C_2$	constant values
X_d	set of packet arrivals during dormant period
X_1 / X_2 / X_3	$\{n\|T_n \leq D_n+1\}$ / $\{n\|D_n+1<T_n\leq D_n+1+t_i\}$ / $\{n\|D_n+1+t_i<T_n\}$
P_c	average power consumption
$\overline{\mathrm{D}}$	average buffering delay
$\overline{\mathrm{D}}_{\max}$	maximum tolerable average delay or delay bound
$\widehat{\mathrm{D}}$	estimated delay for k packets including served, buffered and forecast packets
k	number of packets that delay estimator uses to calculate average delay
p	number of past TTIs that traffic predictor observes in every w-cycle
z	size of dataset
$\mathrm{E}[.]$	expectation value of random variable
$\mathrm{Var}[.]$	variance of random variable
$\mathrm{Cov}[.,.]$	covariance of two random variables
$\mathrm{Pr}[.]$	probability
$\{.\}^C$	absolute complement of $\{.\}$
x_t	packet arrival time on t^{th} TTI
$x_t\|_{t_1}^{t_2}$	set of elements of x_t from $t=t_1$ to $t=t_2$

The rest of this paper is organized as follows. Section II summarizes the WuS principle of operation[1], and introduces the proposed wake-up scheduling concept. Section III mathematically models and optimizes offline the parameters of the wake-up scheduler (WuSched-Offline) for Poisson traffic. Then, the online optimization of the wake-up scheduler (WuSched-Online), which is valid for any traffic distribution, is presented and described in Section IV. These are followed by simulation results and conclusions in Sections V and VI, respectively. Finally, some proofs related to the WuSched-Offline are reported in the Appendices. For readers' convenience, the most relevant variables and mathematical operations used throughout this paper are listed in Table I. Terminology-wise, we use gNB to refer to the base-station unit and UE to denote the mobile device, according to NR specifications [3].

[1]Throughout this work, the term WuS refers to 'WuS without scheduler', which is used as a baseline reference method.

II. WAKE-UP SCHEDULING CONCEPT

A. WuS Overview

In WuS, the cellular modem is configured with a WRx, as a companion low-complex single-purpose receiver in order to decode the wake-up signaling [5]. WuS allows the terminal to reduce the energy consumption by switching off the modem for long periods of time, activating the modem (ON mode) only for short intervals to decode data and control plane signals.

At every wake-up cycle (w-cycle), represented as t_w, the WRx monitors the wake-up signaling for a specific on-duration time (t_{on}) to determine if any data is scheduled or not (see Fig. 1). Occasionally, based on the interrupt signal from WRx, the modem switches ON, decodes both PDCCH and physical downlink shared channel (PDSCH), and performs connected-mode procedures. The wake-up signaling on each w-cycle is represented by 1-bit, referred to as wake-up indicator (WI), where 0 indicates WRx not to wake up the modem (remaining in OFF mode) and 1 triggers WRx to wake up the modem (moving to ON mode) because there is a packet to receive [5]. When WI=1 is sent to WRx, the gNB expects the target mobile device to decode the PDCCH with a time offset equal to the start-up time (t_{su}). After successful decoding of PDCCH/PDSCH, the UE initiates its inactivity timer with a duration of t_i. After the inactivity timer is initiated, if a new PDCCH message is received before the expiration of inactivity timer, the UE re-initiates its inactivity timer. However, if there is no PDCCH message received before the expiration of the inactivity timer, a sleep period starts (modem goes through transitional periods of power down, with a duration of t_{pd}).

In WuS, if there are one or more packet arrivals during the sleep state, the gNB sends WI=1 to the target UE at the next upcoming wake-up instant (as shown in Fig. 1). However, if the WuS configuration (namely, t_w and t_i) is not correctly optimized for the upcoming traffic, the immediate waking up of the UE can either adversely increase its energy consumption, eventually decreasing the benefits of using WuS (meaning that the UE can tolerate longer w-cycles), or even create a worst-case scenario, in which the UE may not even satisfy its delay requirements (implying the need for shorter w-cycles) [24].

B. Wake-up Scheduling

In our proposal, both w-cycle (t_w) and inactivity timer (t_i) are configured semi-statically, and the desired power and delay trade-off is achieved by adjusting the wake-up instant. More

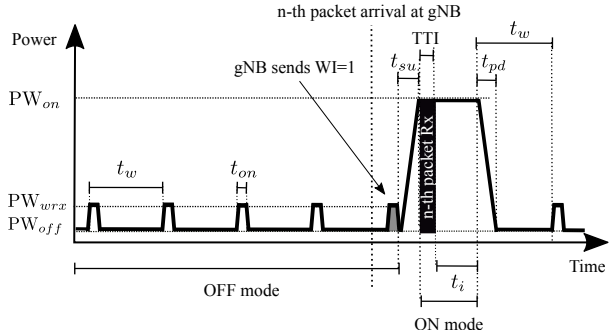


Figure 1: Operation and corresponding parameters of WuS, without scheduler.

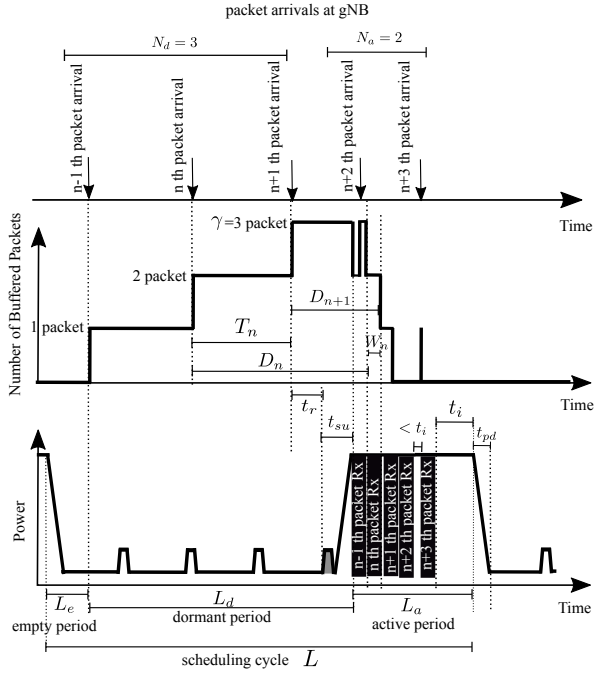


Figure 2: An example of wake-up scheduling and corresponding definitions when $\gamma = 3$. The number of packet arrivals during the corresponding scheduling cycle is equal to $N = 5$ ($N_d = 3$ and $N_a = 2$).

precisely, the wake-up scheduler does not send WI=1 as soon as there is a packet in the w-cycle, but waits until some condition is met; for instance, until the number of buffered packets at the gNB for a given UE is larger than a predefined buffer size threshold (γ), or until the estimated average buffering delay exceeds a predefined threshold ($\overline{\mathrm{D}}_{max}$). The former condition is the core part of the WuSched-Offline and is illustrated in Fig. 2, where the gNB does not send WI=1 until the number of buffered packets reaches to $\gamma = 3$, and it takes four w-cycles to reach the threshold. This way, instead of switching ON the UE for three times, it is switched ON only once after the fourth w-cycle. Note that the buffer size threshold γ influences the packet delays and so it establishes a trade-off in between the energy consumption and the experienced delays. On the other hand, the latter condition mentioned above is used in the WuSched-Online, in order to allow the network to meet maximum tolerable delays of the target applications.

The main motivation behind not sending WI=1 as soon as a packet arrives at the gNB but instead waiting and sending the packets consecutively, is that the state-of-the-art modems suffer from large start-up and power-down stages [5]. Therefore, it is desired in terms of energy-efficiency that once the modem is at ON mode, it receives multiple packets and not a single packet. Although, waiting for longer times to buffer packets can eventually increase the buffering delay. This extra buffering delay should not be problematic as long as the average delay is maintained within a maximum bound. It is worth mentioning that WuS is a specific example of the WuSched-Offline when $\gamma = 1$.

Under the wake-up scheduling, the ON and OFF periods of the UE vary based on its traffic dynamics. For this purpose, we define the *scheduling cycle* as the length of a full cycle of *empty*, *dormant* and *active periods*. The scheduling cycle starts from the expiry of the inactivity timer of the previous scheduling cycle and ends by the expiry of the current cycle's inactivity timer. The scheduling cycle's length (L) is a random variable that depends on the buffer size threshold and the packet arrivals. During each scheduling cycle, only a single WI of 1 is sent to the target UE. We assume N (random variable) packets in the scheduling cycle are served (equivalent to the overall number of packet arrivals in the corresponding scheduling cycle).

In order to help the readers to follow up, the different periods of the scheduling cycle are illustrated in Fig. 2, and defined in what follows:

- **empty period:** It starts right after the beginning of the scheduling cycle and lasts until the arrival of the first packet of such scheduling cycle at the gNB. During the empty period, the number of buffered packets is zero. The length of the empty period is a random variable that we refer to as L_e.
- **dormant period:** It starts as soon as the first packet arrives and lasts until the end of the start-up stage. During the dormant period, packets are buffered at the gNB until the number of buffered packets reaches γ. As a result, by the end of the corresponding w-cycle, the modem is switched ON and, after the start-up stage, the UE is ready to receive the packets. The length of the dormant period is a random variable, denoted by L_d, and the number of packets buffered during the dormant period is referred to as N_d, which is greater than or equal to γ.
- **active period:** It starts after the end of the start-up period and lasts until the end of the scheduling cycle. During the active period, the modem is at ON mode and consumes the high power of PW_{on}, and either it is processing the packets or its inactivity timer is running. The

active period's length is a random variable that is denoted by L_a. The number of packets that arrive at the gNB during the active period is referred to as N_a. During the active period, the UE serves $N = N_d + N_a$ packets, before it enters the next scheduling cycle. The relationship between the length of the different periods of each scheduling cycle is $L = L_e + L_d + L_a$.

For the modem during OFF mode, packets are buffered, and it consumes low power of PW_{off}. In general, the UE power consumption in different operating states is highly implementation dependent, while also depends on the operational configurations. Stemming from the specifically-designed narrow-band WuS signal structure, the WRx power consumption (PW_{wrx}) is generally much lower than that of the modem during ON mode (PW_{on}). Following [5], [22], [25], PW_{wrx}=57 mW, PW_{on}=850 mW, and PW_{off}=16 mW can be considered as representative numbers, while the start-up/power-down periods read t_{su}=15 ms, and t_{pd}=10 ms. Additionally, regarding the WuS parameters, we consider t_{on}=3/14 ms and t_i=1 ms [5]. Furthermore, since the on-duration period of WuS signaling is very short, only three OFDM symbols [22], the WRx contribution to the device energy consumption is very minor. Therefore, in our system model, WRx power consumption is eventually ignored, i.e., we consider $\text{PW}_{wrx} \approx 0$. However, it is noted that in later numerical results, non-zero WRx power consumption is considered.

The wake-up scheduler can be located at the network side (e.g., MAC layer of the gNB), and hence all the computationally intensive processing is performed by the network. Without loss of generality, we assume that the UE can process a single packet (regardless of its size) per transmission time interval (TTI) and that the packet arrival rate (λ) is at most one packet per TTI. TTI of 1 ms is assumed. In general, because NR supports wide bandwidth operation, packets can be served in a very short time duration. In addition, in case the user packet sizes are small, packet concatenation in NR for duration of a TTI is used, so that all packet arrivals in a relatively short time window can be served in a single TTI. Accordingly, we assume that radio-link control entity (located at the gNB) concatenates all those packets arriving during the slot, and as soon as the BBU is triggered on, the device can receive and decode the concatenated packets for a duration of a single TTI. During the corresponding slot, if there was a new packet arrival, the BBU starts serving the corresponding packet by the end of current slot time. Also, we assume that packets are served individually based on first-input first-output (FIFO). One of the key components of the 5G NR design is a flexible self-contained slot-based framework that allows delivering significantly lower latency than LTE. This slot structure framework includes the opportunity for uplink and downlink scheduling, data, and acknowledgement to occur in the

same slot. In other words, in each time slot, UEs can send their acknowledgment to network, and network can decide to re-transmit the packet or not in next inactivity period. In our work, we assume that the self-contained slot-based framework is utilized.

In the case of multimedia packet-data traffic, there is not a strong need to provide a maximum delay budget per packet. Rather, from a user perspective, the delay over the radio interface should simply be lower than maximum average packet delay ($\overline{\mathrm{D}}_{max}$), whose value is set based on the service type. Even in case of typical constant-rate services such as voice and video, (short-term) exceeding delays are often not an issue, as long as the average delay remains constant, assuming averaging over some relatively short time interval. Moreover, maximum delay requirements are mainly used for ultra reliable and low latency communications (URLLC). However, since our main focus in this paper is on multimedia type traffic, we consider average packet delay as QoS indicator of services.

III. Offline Optimization of Wake-Up Scheduling for Poisson Traffic

In this section, the average power consumption and buffering delay of the wake-up scheduler are derived as a function of the buffer size threshold (γ) and the packet arrival rate of a Poisson process (λ). Then, γ is optimized for a given λ and a maximum delay bound ($\overline{\mathrm{D}}_{max}$).

The WuSched-Offline can be modeled as a stationary GI/G/1[2] FIFO queuing system [26]. We use such system's properties to analyze the wake-up scheduler's average delay and power consumption. In this section, packet arrivals are modeled as according to a Poisson process for analytic simplicity and due to its attractive theoretical properties.

Let us refer to the packet inter-arrival times of the nth and n+1th packets at the gNB as T_n, where T is exponentially distributed, and hence $\mathrm{E}[T] = 1/\lambda$ and $\mathrm{Var}[T] = 1/\lambda^2$. Furthermore, we define the nth packet's buffering delay caused by the wake-up scheduler as D_n. Based on Fig. 2, the following expression is always valid,

$$D_{n+1} = W_n + D_n - T_n, \tag{1}$$

where W_n is the time duration between decoding nth and n+1th packets by UE.

Depending on the relation between T_n and D_n, three disjoint sets of packets can be defined,

- X_1: If n $\in X_1$, the n+1th packet arrives before the end of serving nth packet ($T_n \leq D_n + 1$). Therefore, the UE serves n+1th packet immediately after serving nth packet, i.e., $W_n = 1$.

[2]In queuing theory, GI/G/1 represents the queue length in a system with a single server where inter-arrival times have a general distribution and service times have a general distribution.

All packet arrivals during the dormant period (referred to as X_d) are part of X_1 (last packet of the dormant period may or may not[3] belong to X_1). Therefore, $X_d - \{N_d\} \subseteq X_1$.

- X_2: If n $\in X_2$, the n+1th packet arrives after inactivity timer is triggered and before its expiry time ($D_n + 1 < T_n \leq D_n + 1 + t_i$). In such conditions, n+1th packet is served immediately, $D_{n+1} = 0$, and based on (1), then $W_n = T_n - D_n$.
- X_3: If n $\in X_3$, the n+1th packet arrives after inactivity timer is expired ($D_n + 1 + t_i < T_n$). X_3 has a single packet which is the last served packet. Therefore, n+1th packet belongs to the next scheduling cycle. As a result $W_n = L_d + T_n - D_n$, where $L_d = D_{n+1}$ is the length of the next scheduling cycle's dormant period or, equivalently, the delay of the first packet in the next scheduling cycle.

For compactness purposes, in the rest of the paper, the subscript n from random variables T_n, D_n and W_n are removed, unless there is a need to emphasize their dependence of n explicitly. The summary of W_n calculation is drawn in the second column of Table II.

We note that the WuSched-Offline is analyzed for Poisson traffic arrivals and thus it cannot strictly-speaking cover the case of retransmissions. This is because retransmissions would change the statistics of the packet arrivals (including new packets and retransmission) based on the channel quality, error model, and retransmission timings.

A. Stationary Probabilities

The stationary probabilities that the nth packet belongs to one of the three sets (X_1, X_2, X_3) need to be calculated to derive the delay and power expressions of the wake-up scheduler analytically. For this purpose, based on the definition of X_1 and X_3, we can write,

$$\Pr[n \in X_1] = \Pr[T - 1 \leq D] = \int_0^\infty (1 - e^{-\lambda(t+1)}) f_D(t) dt = 1 - \int_0^\infty e^{-\lambda(t+1)} f_D(t) dt, \quad (2)$$

and

$$\Pr[n \in X_3] = \Pr[T - t_i - 1 > D] = 1 - \int_0^\infty (1 - e^{-\lambda(t+t_i+1)}) f_D(t) dt = e^{-\lambda t_i} \int_0^\infty e^{-\lambda(t+1)} f_D(t) dt, \quad (3)$$

where $f_D(t)$ is the probability density function (PDF) of D. Therefore, based on (2) and (3), we can model their relation as follows,

$$\Pr[n \in X_3] = e^{-\lambda t_i} (1 - \Pr[n \in X_1]). \quad (4)$$

[3]In such a scenario, N_d^{th} packet belongs to either X_2 or X_3.

Table II: Summary of analysis of W_n, H_n and stationary probabilities.

$n \in$	W_n	H_n	$\Pr[n \in X_i]$
X_1	1	0	$\frac{\gamma+(e^{\lambda t_i}+C_0)\lambda}{\gamma+e^{\lambda t_i}+C_0\lambda}$
X_2	$T_n - D_n$	$-(T_n - D_n - 1)^2$	$\frac{(1-\lambda)(e^{\lambda t_i}-1)}{\gamma+e^{\lambda t_i}+C_0\lambda}$
X_3	$L_d + T_n - D_n$	${L_d}^2 - (T_n - D_n - 1)^2$	$\frac{1-\lambda}{\gamma+e^{\lambda t_i}+C_0\lambda}$

Also by using (4) and the probability assignment rule ($\sum_{j=1}^{3} \Pr[n \in X_j] = 1$), we attain,

$$\Pr[n \in X_2] = (1 - e^{-\lambda t_i})(1 - \Pr[n \in X_1]). \tag{5}$$

Furthermore, we can model the expected value of W based on all possible values of W_n (second column of Table II) by using the law of total probability formula as follow,

$$\mathrm{E}[W] = \Pr[n \in X_1] + \Pr[n \in X_2]\mathrm{E}[(T - D)|n \in X_2] + \Pr[n \in X_3]\mathrm{E}[(L_d + T - D)|n \in X_3]. \tag{6}$$

Appendices B, C, D and E include the derivations of $\mathrm{E}[(T-D)|n \in X_2]$, $\mathrm{E}[(T-D)|n \in X_3]$, $\mathrm{E}[L_d]$, and $\mathrm{E}[W]$, respectively. Then, by substituting (4), (5), (31), (33), (34) and (36) into (6), we can obtain,

$$\Pr[n \in X_1] = \frac{\gamma + (e^{\lambda t_i} + C_0)\lambda}{\gamma + e^{\lambda t_i} + C_0\lambda}. \tag{7}$$

Then, $\Pr[n \in X_3]$ and $\Pr[n \in X_2]$ can be calculated based on (4) and (5), respectively. The summary of the calculation of the stationary probabilities is drawn in the fourth column of Table II.

B. Average Holding Times

In this section, the average holding times (i.e., the length) of the empty and active periods, as well as the average number of packet arrivals during the dormant and active periods, are calculated. Note that we already derived the average length of dormant period in Appendix D.

1) Empty Period: If the nth packet belongs to X_3, then the n+1th packet is the first packet of the next scheduling cycle and hence the length of the empty period equals to $T - D - 1 - t_i$. As a result, based on (32),

$$\mathrm{E}[L_e] = \mathrm{E}[(T - D - 1 - t_i)|n \in X_3] = \frac{1}{\lambda}. \tag{8}$$

2) Dormant Period: Based on (34), $\mathrm{E}[N_d]$ can be calculated as,

$$\mathrm{E}[N_d] = \gamma + \lambda C_0 + 1, \tag{9}$$

where 1 is raised due to presence of the first packet in each scheduling cycle.

Table III: Summary of analysis of average holding times and average number of packets per period/cycle.

period/cycle	Holding Time	Number of packet arrivals
empty	$\mathrm{E}[L_e] = \frac{1}{\lambda}$	0
dormant	$\mathrm{E}[L_d] = \frac{\gamma}{\lambda} + C_0$	$\mathrm{E}[N_d] = \gamma + \lambda C_0 + 1$
active	$\mathrm{E}[L_a] = (\gamma + \lambda C_0 + 1)(1 + \frac{e^{-\lambda t_i}}{1-\lambda}) + C_2$	$\mathrm{E}[N_a] = \frac{(\gamma+\lambda C_0+1)\lambda}{1-\lambda} e^{-\lambda t_i}$
scheduling	$\mathrm{E}[L] = (\gamma + \lambda C_0)C_1 + C_2$	$\mathrm{E}[N] = (\gamma + \lambda C_0 + 1)(1 + \frac{e^{-\lambda t_i}}{1-\lambda})$

3) Active Period: During the active period, first, N_d packets are served for a duration of N_d TTIs, and then other packet arrivals, during the serving time of N_d TTIs, with average number of $N_d\lambda$ packets, are served. After some rounds, there will be a point in which the inactivity timer expires, and no buffered packets remain in the queue. Therefore, the average number of received packets during the active period can be modeled by a geometric progression as follows,

$$\mathrm{E}[N_a] = \big(\sum_{i=0}^{\infty} \lambda^i \mathrm{E}[N_d]\big) \Pr[T - D - 1 > t_i | n \in X_3] = \frac{\gamma + \lambda C_0 + 1}{1-\lambda} e^{-\lambda t_i}. \tag{10}$$

4) Scheduling Cycle: The average number of packets that is served during each scheduling cycle can be obtained as follows,

$$\mathrm{E}[N] = \mathrm{E}[N_d] + \mathrm{E}[N_a] = (\gamma + \lambda C_0 + 1)(1 + \frac{e^{-\lambda t_i}}{1-\lambda}). \tag{11}$$

Furthermore, the length of the inactivity timer (ω) is dependent on the packet inter-arrival time (t_p). If a packet arrives before t_i, ω is equal to the inter-packet arrival time, otherwise ω equals to t_i. Therefore, ω can be calculated as a function of t_p as,

$$\omega(t_p) = \begin{cases} t_p, & \text{for } \ t_p \leq t_i \ , \\ t_i, & \text{for } \ t_p > t_i \ . \end{cases} \tag{12}$$

Hence, $\mathrm{E}[\omega]$ can be expressed as,

$$\mathrm{E}[\omega] = \int_0^{\infty} \omega(t) \lambda e^{-\lambda t}(t) dt = \frac{1 - e^{-\lambda t_i}}{\lambda}. \tag{13}$$

By utilizing (11) and (13), we can obtain the average length of the active period as follows,

$$\mathrm{E}[L_a] = \mathrm{E}[N] + \mathrm{E}[\omega] = (\gamma + \lambda C_0 + 1)(1 + \frac{e^{-\lambda t_i}}{1-\lambda}) + \frac{1 - e^{-\lambda t_i}}{\lambda}. \tag{14}$$

Finally, the average length of the scheduling cycle (L) can be calculated as follows,

$$\mathrm{E}[L] = \mathrm{E}[L_e] + \mathrm{E}[L_d] + \mathrm{E}[L_a] = (\gamma + \lambda C_0 + 1)C_1 + C_2, \tag{15}$$

where C_1 and C_2 are constants given by,

$$C_1 = \frac{1}{\lambda} + 1 + \frac{e^{-\lambda t_i}}{1-\lambda}, \quad \text{and} \quad C_2 = \frac{1 - e^{-\lambda t_i}}{\lambda}. \tag{16}$$

The summary of the calculation of the average holding times and the average number of packets is shown in Table III.

C. Average Power Consumption

The average power consumption of the UE with wake-up scheduler, denoted by $\overline{\mathrm{P}}_c$, can be calculated as the ratio of the average energy consumption and the corresponding overall observation period, expressed as,

$$\overline{\mathrm{P}}_c = \frac{e_t + (\mathrm{E}[L_e] + \mathrm{E}[L_d] - t_t)\mathrm{PW}_{off} + \mathrm{E}[L_a]\mathrm{PW}_{on}}{\mathrm{E}[L]}, \tag{17}$$

where e_t and t_t are the energy consumption of transitional states and the overall time period that the UE spends on transitional periods, which respectively read as,

$$t_t = t_{su} + t_{pd}, \quad \text{and} \quad e_t = t_t \frac{\mathrm{PW}_{on} - \mathrm{PW}_{off}}{2}. \tag{18}$$

Due to the negligible value of the power consumption of the UE at OFF mode, we can further assume that $\mathrm{PW}_{off} \approx 0$. Therefore, (17) can be expanded as a function of γ as follows,

$$\overline{\mathrm{P}}_c(\gamma) = \mathrm{PW}_{on} \frac{t_t/2 + (\gamma + \lambda C_0 + 1)(C_1 - \frac{1}{\lambda}) + C_2}{(\gamma + \lambda C_0 + 1)C_1 + C_2}. \tag{19}$$

From the above equation, it is clear that the average power consumption $\overline{\mathrm{P}}_c(\gamma)$ is a strictly decreasing function with respect to γ at $\gamma \geq 1$, i.e., $\frac{d\overline{\mathrm{P}}_c(\gamma)}{d\gamma} < 0$. As expected, increasing the buffer size threshold reduces the power consumption.

D. Average Buffering Delay

By squaring both sides of (1) and using basic sum and multiplications, we can obtain the following equation,

$${D_{n+1}}^2 = H_n + (T_n - D_n)^2 - 2(T_n - D_n) + 1, \tag{20}$$

where

$$H_n = (W_n - 1)\big(W_n + 1 - 2(T_n - D_n)\big). \tag{21}$$

Then, by averaging both sides of (20), we get,

$$\mathrm{E}[D] = \frac{\mathrm{E}[T^2] - 2\mathrm{E}[T] + 1 - 2\mathrm{Cov}[D,T] + \mathrm{E}[H]}{2(\mathrm{E}[T] - 1)} = \frac{1}{\lambda} + \frac{1}{2(1/\lambda - 1)} - \frac{\mathrm{Cov}[D,T]}{1/\lambda - 1} + \frac{\mathrm{E}[H]}{2(1/\lambda - 1)}. \tag{22}$$

In Appendices G and H, we present the calculations of $\mathrm{Cov}[D,T]$ and $\mathrm{E}[H]$. Finally, the average delay can be obtained by replacing (43) and (51) into (22), as follows,

$$\begin{aligned}\overline{\mathrm{D}}(\gamma) = \mathrm{E}[D] = \frac{1}{\lambda} + \frac{\lambda}{2(1-\lambda)} - \frac{1}{\lambda(1 - \lambda + e^{-\lambda t_i})} + \frac{e^{\lambda t_i}}{\lambda((1-\lambda)e^{\lambda t_i} + 1)(\gamma + \lambda C_0 + 1)} + \\ \frac{-e^{\lambda t_i} + \frac{\gamma - 1}{2}}{\lambda(\gamma + e^{\lambda t_i} + C_0\lambda)} + \frac{1}{\gamma + e^{\lambda t_i} + C_0\lambda}\left[\frac{\gamma^2}{2\lambda} + \gamma C_0 + \frac{\lambda C_0^2}{2} + \frac{\lambda t_w^2}{24}\right].\end{aligned} \tag{23}$$

For presentation purposes, we represent $\mathrm{E}[D]$ as $\overline{\mathrm{D}}(\gamma)$. Similar to $\overline{\mathrm{P}}_c(\gamma)$, the derivative of $\overline{\mathrm{D}}(\gamma)$ with respect to continuous variable γ can be calculated (refer to Appendix I), from which it can be concluded that the average buffering delay $\overline{\mathrm{D}}(\gamma)$ is a strictly increasing function with respect to γ at $\gamma \geq 1$, i.e., $\frac{d\overline{\mathrm{D}}(\gamma)}{d\gamma} > 0$.

As expected, contrary to the behavior of $\overline{\mathrm{P}}_c(\gamma)$, increasing the buffer size threshold increases the buffering delay. Therefore, a clear energy-delay trade-off appears in the selection of γ for the wake-up scheduler.

E. Offline Optimization of Wake-Up Scheduler

From the system-level point of view, the tunable parameter of the WuSched-Offline is the buffer size threshold ($\gamma \geq 1$), assuming a fixed configuration of the w-cycle and the inactivity timer. For the sake of presentation compactness, we will not investigate how to set both parameters; readers can refer to our recent work in [24]. The remaining parameters of the wake-up scheduler (t_{on}, t_{pd}, t_{su}) depend on physical constraints and signal design, and accordingly, we assume them to be fixed as well. Based on these assumptions, we focus on optimizing the buffer size threshold (γ) in order to minimize the UE's power consumption while satisfying a specific delay requirement (i.e., average buffering delay should be less than or equal to a maximum tolerable delay, $\overline{\mathrm{D}}_{max}$), under Poisson traffic model assumption, for given values of t_w, t_i, t_{on}, t_{pd} and t_{su}.

By using the analytical models of the power consumption and the buffering delay, as well as their behaviour as a function of γ (i.e., $\overline{\mathrm{P}}_c(\gamma)$ in (19) is a decreasing function and $\overline{\mathrm{D}}_(\gamma)$ in (23) is an increasing function), and by following a similar approach as the one in [24], the optimal buffer size threshold (γ^*) can be easily obtained. The result is included in the next Theorem 1.

Theorem 1. *The optimal buffer size threshold that minimizes the UE's power consumption while satisfying a specific delay requirement is* $\gamma^* = \lfloor \gamma_m \rfloor$*, being* γ_m *the boundary point of the delay constraint, i.e.,* $\overline{D}(\gamma_m) = \overline{D}_{max}$.

Proof. Thanks to $\frac{d\overline{\mathrm{P}}_c(\gamma)}{d\gamma} < 0$ and $\frac{d\overline{\mathrm{D}}(\gamma)}{d\gamma} > 0$, we can easily show that $\gamma = \lfloor \gamma_m \rfloor$ is the optimal solution to minimize the UE's power consumption subject to a specific delay requirement, as detailed next. Fig. 3 (a) and Fig. 3 (b) show the decreasing trend of the power consumption and the increasing behaviour of the delay constraint as a function of γ, respectively, which satisfies $\frac{d\overline{\mathrm{P}}_c(\gamma)}{d\gamma} < 0$ and $\frac{d\overline{\mathrm{D}}(\gamma)}{d\gamma} > 0$. Consider an arbitrary point C in the interior of the feasible region for γ ($\gamma_C < \lfloor \gamma_m \rfloor$ where $\overline{\mathrm{D}}(\gamma_m) = \overline{\mathrm{D}}_{max}$). As it can be seen from Fig. 3, there is always a

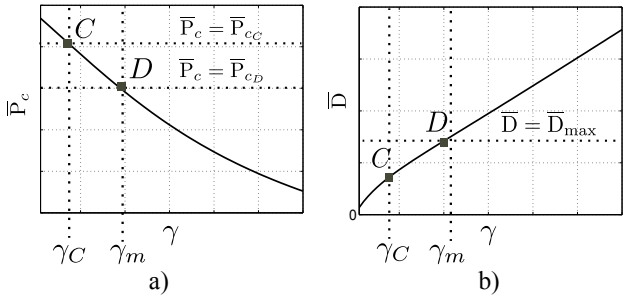


Figure 3: Schematic proof of Theorem 1.

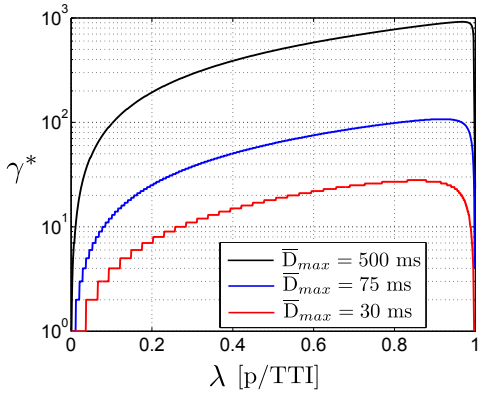


Figure 4: Optimal value of buffer size threshold as function of packet arrival rate for $t_w = 15$ ms and $t_i = 1$ ms.

point close to the boundary of the delay constraint, denoted by D ($\gamma_D = \lfloor \gamma_m \rfloor$), where its power consumption $\overline{\mathrm{P}}_{c_D}$ is lower than that of C ($\overline{\mathrm{P}}_{c_D} < \overline{\mathrm{P}}_{c_C}$). Then, we can conclude that under a given delay constraint, the point $\lfloor \gamma_m \rfloor$ always exists and attains the lowest power consumption within the feasible region, and hence it is the optimal solution. The γ_m can be calculated using any standard root-finding algorithm that meets $\overline{\mathrm{D}}(\gamma_m) = \overline{\mathrm{D}}_{max}$. □

Fig. 4 shows how γ^* changes when λ varies for delay bounds of 30 ms, 75 ms and 500 ms, for $t_i = 1$ ms and $t_w = 15$ ms. It clearly shows that by increasing λ, γ^* increases too. The high buffered size threshold reduces energy consumption; however, if the packet arrival rate is low, configuring a high buffer size threshold can increase buffering delay and cannot satisfy the maximum delay bound. As a result, a smaller γ should be configured for a low λ to satisfy the delay requirement. For high λ, it is necessary to increase γ to reduce energy consumption. Similarly, for higher delay bounds, γ can be configured high, due to the much-relaxed delay requirements. Interestingly, for high packet arrival rates close to 1 p/TTI, γ reduces to one, implying that the UE is on ON mode most of the time (because of the inactivity timer, most of the time the UE does not enter to OFF mode). This is the main reason for limiting λ for less than 1 p/TTI. Therefore, the wake-up scheduler is not effective anymore for packet arrival rates

close to or beyond 1 p/TTI. Instead, other power-saving mechanisms, such as microsleep, could be used. Finally, as can be observed in Fig. 4, γ^* (precisely γ_m) has a linear trend concerning λ for lower packet arrival rates, and this can be exploited to reduce the computational complexity of root-finding algorithms.

IV. Online Optimization of Wake-Up Scheduling based on Traffic Prediction

In this section, we present the online optimization of the wake-up scheduler, which aims at trading-off in between power consumption and packet delay in a dynamic manner by adaptively and autonomously determining when to send the WI, according to the traffic pattern and a maximum tolerable delay ($\overline{\mathrm{D}}_{max}$). Differently from the WuSched-Offline that was presented and modeled analytically in Section III, the WuSched-Online does not assume any a priori knowledge about the traffic statistics, and thus it is general and can be applied to all traffic distributions as well as mixed traffic combinations.

Proactively knowing the packet arrival times for a forecast horizon, allows the UE to remain at OFF mode for longer periods. In this regard, the proposed wake-up scheduler increases the sleep period of the UE as much as possible in a greedy manner by not sending WI=1 until the average buffering delay approaches $\overline{\mathrm{D}}_{max}$. For this purpose, the average delay is estimated for k packets, in every w-cycle.

In the proposed scheme, *traffic predictor* forecasts the packet arrival times of the target UE for the forecast horizon of one w-cycle based on past packet arrival times. In other words, the traffic predictor observes the session's packet arrival time for p previous TTIs until beginning of the current TTI (c) and then predicts the packet arrival times for the upcoming w-cycle with TTI indexes of $[c, c+t_w)$. Note that, differently from the WuSched-Offline, the WuSched-Online can also cover retransmissions, by taking the packet arrival times of previous retransmitted packets and then predicting packet arrivals of either new packets or retransmission packets.

Furthermore, every w-cycle, a *delay estimator* block estimates the average buffering delay ($\widehat{\mathrm{D}}$) of k packets, assuming that the UE is switched on at the end of the upcoming w-cycle. If $\widehat{\mathrm{D}}$ is higher than $\overline{\mathrm{D}}_{max}$, the network realizes that the only way to have shorter delay is by sending WI=1 promptly. Otherwise (if $\widehat{\mathrm{D}} \leq \overline{\mathrm{D}}_{max}$), it leaves the UE to remain in OFF mode for at least another w-cycle. Finally, a *delay comparator* block performs the task of comparison and decision making (i.e., whether to send WI=1 or WI=0) accordingly.

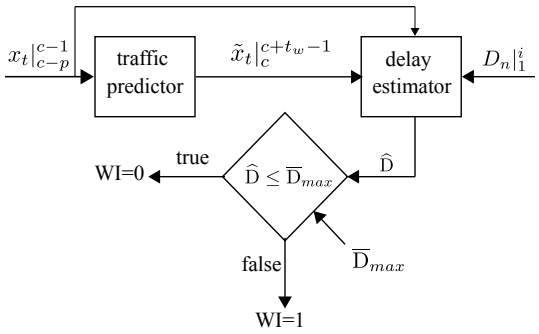


Figure 5: Overall block diagram of the WuSched-Online.

The overall block diagram of the proposed WuSched-Online is shown in Fig. 5. The different modules and variables are described below.

A. Dataset from Real Traces

In this paper, the performance of the WuSched-Online is investigated using real video and audio streaming traces. For this, we monitored one operative network in Spain during one month using the online watcher presented in [27]. We have selected only those traces gathered during the night hours (1am - 6am) to be sure that the selected cell is serving very few users. This allows us to assume that our traces are not affected by the packet scheduler at the base station, since an adequate number of radio resources per TTI is available to accommodate all the transmitting UEs.

Our dataset includes two columns: the Identifier of the UE, and the timestamp of the packet arrival (with TTI granularity). The classifier introduced in [28] is used to properly select the traces of the apps of interest. The collected dataset consists of 1500 sessions of different traffic type. For the sake of comparison, we also generated Poisson traffic with mean packet arrival rate of 0.2 p/TTI, and added them to the dataset.

B. Traffic Predictor

The traffic prediction can be formulated as a time series forecasting problem, where the packet arrivals at each TTI are defined as the values of the time series. The dataset with size z for a particular traffic type is represented by $x_t|_1^z$, where x_t indicates the packet arrival time during the t^{th} TTI. In this work we tailor a stacked LSTM neural network architecture [29] to predict the next packet arrivals over a finite horizon. We choose LSTM since it has been proven in [29]–

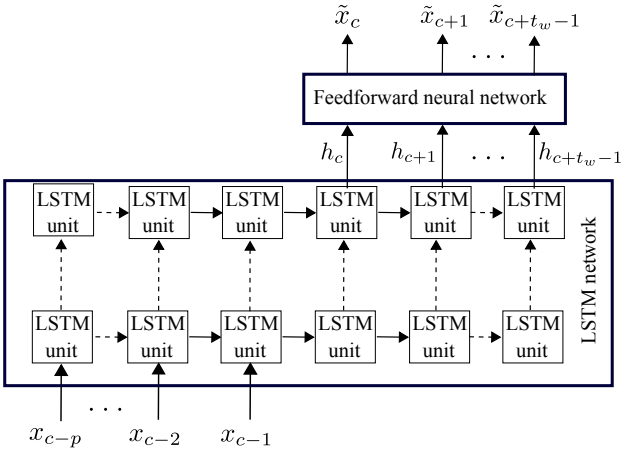


Figure 6: Proposed architecture for the packet arrival time prediction.

Table IV: Training hyperparameters

Initial learning rate	0.001
Number of epochs	100
Number of LSTM hidden states	64
Number of LSTM hidden layers	5
Number of feed-forward hidden layers	1
Optimization algorithm	Adam
Loss function	MAPE

[31] to have lower prediction errors than other time series forecasting approaches, such as auto regressive integrated moving average (ARIMA) [32].

In the proposed architecture, multiple LSTM units are concatenated to form one layer of the LSTM network. Each unit computes the operations on single TTI and transfer the output to the next LSTM unit. The number of concatenated units indicates the number of TTIs (p) that are considered before making the prediction. The proposed architecture for the traffic predictor is depicted in Fig. 6. The LSTM unit of each layer extracts a fixed number of features, which are passed to the next layer. The depth of the network (e.g., the number of layers) is to increment the accuracy of the prediction, which is done by the last fully connected layer.

As shown in Fig. 5 and 6, the proposed network observes $x_t|_{c-p}^{c-1}$ and, then, predicts the traffic in the upcoming w-cycle $\tilde{x}_t|_c^{c+t_w-1}$ by delaying the prediction for the duration of t_w. Finally, the output of the LSTM network ($h_t|_c^{c+t_w-1}$) is fed to a fully connected neural network that performs the actual prediction. The last feed-forward layer applies the softmax activation function, which is needed during the training phase to optimize the weights of the network neurons [30]. The first layer size corresponds to p observed TTIs, while the last layer output has a length equal to future horizon t_w.

The traffic predictor is trained using the dataset in Section IV-A and specified for each of the

considered traffic type. In particular, we have trained the LSTM for four traffic profiles: Youtube videos, Spotify audios, Mixed Youtube/Spotify, and Poisson traffic. The implementation of the traffic prediction algorithm was performed in Python, using Keras and Tensorflow, as backend. The chosen hyperparameters are reported in Table IV. The number of hidden layers is fixed to 5, which is the number giving a good trade-off between prediction accuracy and model complexity. For the training part, we used the Adam's algorithm [33] as optimizer and the Mean Absolute Percentage Error (MAPE) as loss function. We define the MAPE as follows,

$$\text{MAPE} = \frac{100\%}{t_w} \sum_{t=c}^{c+t_w-1} \frac{|\tilde{x}_t - x_t|}{x_t}, \tag{24}$$

where $\tilde{x}_t$ is the predicted packet arrival time on the t^{th} TTI.

C. Delay Estimator

We categorize packet arrivals during past observation $[c{-}p, c)$ and forecast horizon $[c, c{+}t_w)$ into three disjoint sets: (1) already served packets with index of $1{\leq}n{\leq}i$, (2) buffered packets with index of $i{+}1{\leq}n{\leq}j$ where $j{\leq}p$, and (3) forecast packet arrivals for upcoming w-cycle with index of $j{+}1{\leq}n{\leq}k$, where $k{-}j{\leq}t_w$. Delay estimator utilizes the served packets' delay times (D_n, for $1{\leq}n{\leq}i$), and estimated delays of buffered and forecast packets ($\bar{D}_n$, for $i{+}1{\leq}n{\leq}k$), to estimate the average buffering delay ($\widehat{\mathrm{D}}$), as follows,

$$\widehat{\mathrm{D}} = \frac{\sum_{n=1}^{i} D_n + \sum_{n=i+1}^{k} \bar{D}_n}{k}. \tag{25}$$

Finally, the decision whether to send WI=1 or not is decided by comparing $\widehat{\mathrm{D}}$ with $\overline{\mathrm{D}}_{max}$. If the estimated delay is larger than maximum delay bound, WI=1 is sent to the target UE.

V. Numerical Results

In this section, a set of numerical results are provided in order to evaluate the accuracy of the traffic predictor used for the online optimization of the wake-up scheduler (WuSched-Online, in Section V-A) and validate the functionality of the proposed wake-up schedulers (WuSched-Offline and WuSched-Online) for different traffic patterns including Poisson traffic (in Section V-B) and realistic traffic (in Section V-C).

As previously mentioned, four traffic types are considered: video streaming, audio streaming, mixed audio/video streaming, and Poisson traffic. One of the distinguishing features of the video and audio streaming is their low playback latency. The average latency to have high quality playback of a track is 265 ms [34]. Accordingly, for audio streaming, we assume that the

maximum delay bound ($\overline{\mathrm{D}}_{max}$) is 265 ms. Similarly, we assume that the maximum delay bounds for video streaming, mixed flow and Poisson traffic are 40 ms, 40 ms, and 30 ms, respectively. Furthermore, for the numerical results, the UE power consumption model similar to [5], [8], [22], [25] is deployed, for which PW_{wrx}=57 mW, PW_{on}=850 mW, PW_{off}=16 mW, t_{su}=15 ms, and t_{pd}=10 ms. Regarding the WuS parameters, we assume $t_{on}=3/14$ ms and t_i=1 ms [5].

Three different sets of performance results, in terms of power consumption and delay, are presented. Namely, (1) wake-up scheme without scheduler ('WuS') that is considered as a benchmark scheme, (2) offline optimization of the wake-up scheduler ('WuSched-Offline'), and (3) online optimization of the wake-up scheduler ('WuSched-Online'). Furthermore, to verify the performance of the WuSched-Offline under Poisson traffic model, both the results obtained from mathematical analysis ('ana. WuSched-Offline') given in Theorem 1 and simulation results ('sim. WuSched-Offline') are provided in Section V-B.

According to Theorem 1 and (23), it is necessary for the WuSched-Offline to know the packet arrival rate a priori in order to calculate the optimal buffer size threshold. Therefore, in this work we assume that packet arrival rate is estimated based on an exponential moving average, as proposed in [35]. Authors in [35] introduce an approach to estimate the packet arrival rate, and they show that their method converges to the actual packet arrival rate under a wide range of traffic types.

A. Prediction Accuracy

In this section, we seek to evaluate the accuracy of predictions of the proposed traffic predictor as a function of the number of previous observations (p), the length of the horizon (t_w), and the type of applications generating the traffic. For that, we use the MAPE in (24) to quantify the accuracy of traffic prediction.

The impact of t_w and p on the prediction errors is illustrated in Fig. 7. For shorter w-cycles, the predictions follow the actual values closely, whereas for larger w-cycles, the prediction error is bigger: longer forecast horizons (t_w) decrease the accuracy of the predictor, as expected. Furthermore, as it can be observed, the MAPE reduces with a larger number of observations (p) for all four traffic types. Also, the accuracy decreases (i.e., MAPE increases) based on the different traffic type. The accuracy rate is smaller for Poisson packet arrivals than for video and audio traffics, due to its simpler traffic pattern. For Poisson traffic, the MAPE increases around

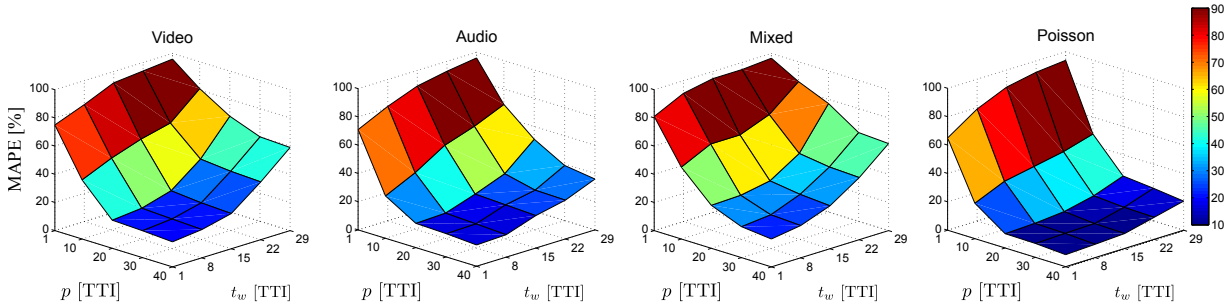


Figure 7: MAPE as function of number of past observations p and forecast horizon t_w for different traffic types.

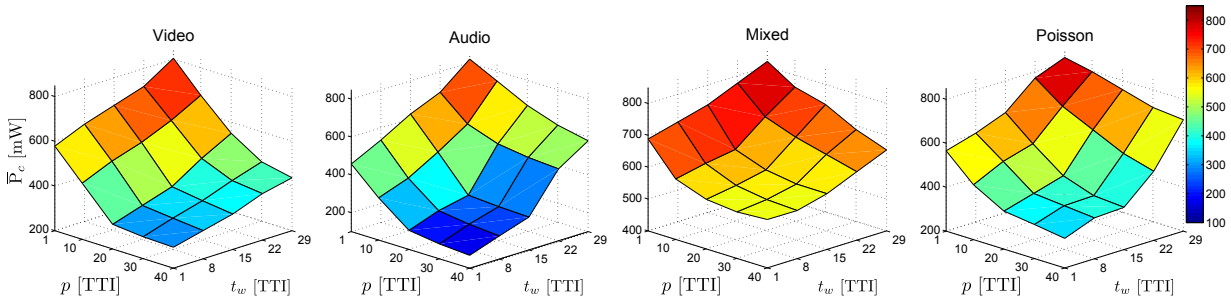


Figure 8: Power consumption of the WuSched-Online as function of number of past observations p and forecast horizon t_w for different traffic types, while maintaining the corresponding delay requirements of each traffic ($k = 45$ packets).

15% when t_w increases from 10 to 30 TTIs for given $p = 20$ TTIs; however, for other traffics the accuracy reduction is high and MAPE increases around 50% for the same t_w change.

As shown in Fig. 7, from prediction accuracy point of view, it is desirable to reduce t_w and enlarge p. However, in terms of power consumption, such a reduction of the w-cycle would contribute to a higher energy consumption due to frequent checking of wake-up signaling. Additionally, a higher number of past observations p involves a longer memory length of the LSTM network and a large amount of information that must be stored for a precise traffic prediction. As a result, the floating point operations per second (FLOPS) of the LSTM network increases. This complexity overhead can become very high, especially if the number of users per cell increases.

Note that different parameters of the traffic predictor can be configured in such a way that they provide adequate precision for the WuSched-Online, which is measured in terms of the estimated delay over a certain number of packets k (i.e., $\widehat{\mathrm{D}}$ in (25)). In particular, the impact of traffic prediction errors on the estimated delay depends on p, k and t_w. To ensure efficient usage of the forecast horizon and, at the same time, limit the long-term differences in the quality-of-service to an acceptable level, k should be set longer than t_w for the upcoming w-cycle. At the same time, k should be sufficiently short so that prediction errors are not strongly noticed by a user. In this work, we set k to 45 packets.

From (25), it can be inferred that the estimated delay has lower sensitivity with respect to prediction accuracy. To illustrate this, we evaluate the impact of the prediction errors on the actual WuSched-Online performance. Fig. 8 depicts the power consumption of the WuSched-Online as a function of p and t_w, for each traffic type, considering the associated maximum delay bounds. It can be observed that configuring p and t_w to 20 and 15 TTIs, respectively, can achieve reasonable power saving. Indeed, further reducing t_w and/or further increasing p beyond such values, reduces the power consumption slightly. Accordingly, for the rest of paper, we assume k=45 packets, t_w=15 TTIs, p=20 TTIs.

B. Performance Evaluation: Poisson Packet Arrivals

In this section, we investigate the performance of the three methods (WuSched-Online, WuSched-Offline, and WuS) in terms of average buffering delay and average power consumption when traffic follows a Poisson pattern, and packet arrival rate (λ) is increased from 0 to 1 p/TTI. For this purpose, Fig. 9 and 10 show the average delay and power consumption of proposed mechanisms under two different delay bounds of 23 ms and 30 ms, respectively.

Fig. 9 (a) depicts the average packet delay experienced by the WRx-enabled UE when packet arrival rates vary. As it can be observed, the average delay for WuS is about $\overline{\mathrm{D}}_{max} = 23$ ms for lower arrival rates. Note that, in case of WuS, the average delay is dependent on start-up period and w-cycle. For the WuSched-Offline, the experienced delay follows closely the maximum delay bound for wider range of packet arrival rates, and is slightly shorter than the maximum tolerable delay. This is because of selecting the greatest integer less than or equal to the optimal buffer size threshold of the optimization problem. For the WuSched-Online, the actual average delay is slightly higher than the maximum delay bound. The main reason for such negligible excess delay is the unavoidable errors in the traffic predictions, whose impact depends on the w-cycle. In practice, to compensate for such small excess delay, the delay bound can be set slightly smaller than the actual average delay requirement. Finally, for larger arrival rates, all three methods' delays reduce sharply. This is because of the inactivity timer, which causes the UE to remain on active state most of the time, due to high arrival rates, and therefore the overall delay reduces to the packet processing delay.

Fig. 9 (b) compares the average power consumption of the three methods under Poisson arrivals. As it can be seen, the simulated results of the WuSched-Offline closely follow the analytical results. Interestingly, one may observe that that the optimal buffer size threshold

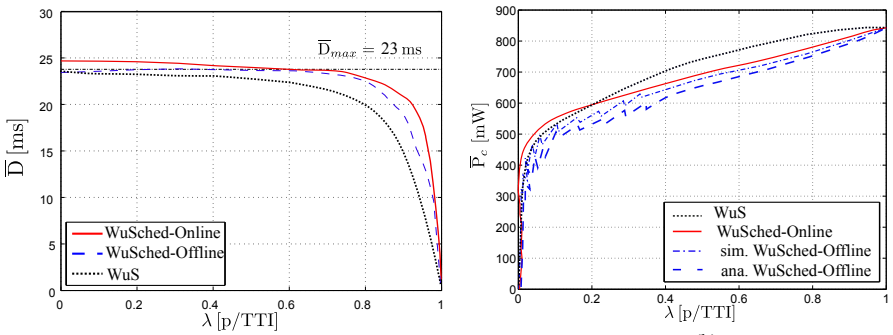
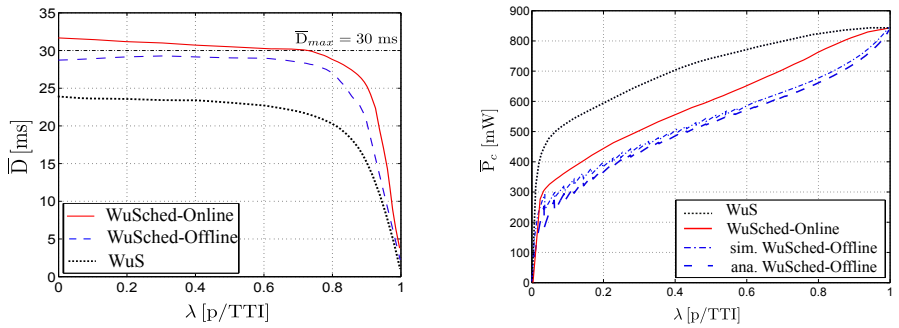


Figure 9: (a) Average buffering delay and (b) power consumption of WuS, WuSched-Offline and WuSched-Online, as function of packet arrival rate for $\overline{\mathrm{D}}_{max} = 23$ ms.

increases when increasing λ, as shown in Fig. 4. Based on $\frac{d\overline{\mathrm{P}}_c}{d\gamma} < 0$, it is expected that the average power consumption would decrease when increasing λ, however Fig. 9 (b) contradicts it. This can be justified by the fact that at same time that γ^* increases, λ also increases, which increases the power consumption due to frequent packet processing, and it is a dominant contributor to the mean power consumption than the power reduction due to increasing γ. Additionally, there are some sharp reductions on the power consumption for lower packet arrival rates, caused by increasing γ with one unit. Furthermore, WuS and WuSched-Offline yield similar power consumption for lower packet arrivals, however, it is clear that WuSched-Offline consumes less power than WuSched-Online and WuS for larger packet arrival rates. This shows that there is need to reconfigure and optimize WuS for different packet arrival rates. Also, the WuSched-Online outperforms WuS for higher packet arrival rates. Finally, for high packet arrival rates, all three methods approach to a fully modem ON scenario with power consumption of 850 mW.

Similar to Fig. 9, Fig. 10 is drawn to show the buffering delay and average power consumption of the proposed methods under 30 ms delays. As it can be observed in Fig. 10 (a), the average delay for WuS is much lower than for the $\overline{\mathrm{D}}_{max} = 30$ ms case. However, the proposed wake-up schedulers behave consistently, and adapt themselves to new delay requirement, similar to Fig 9 (a). Furthermore, Fig. 10 (b) compares the average power consumption of the three methods. It is clear that WuSched-Offline consumes less power than WuSched-Online and WuS. Also, the WuSched-Online outperforms WuS.



(a) (b)

Figure 10: (a) Average buffering delay and (b) power consumption of WuS, WuSched-Offline and WuSched-Online, as function of packet arrival rate for $\overline{\mathrm{D}}_{max} = 30$ ms.

C. Performance Evaluation: Realistic Traffic

In this section, the average power consumption and the buffering delay of the three methods (WuSched-Online, WuSched-Offline, and WuS) are evaluated for different realistic traffic patterns.

Fig. 11 shows the empirical cumulative distribution function (CDF) of packet delay for the four different traffic types. Generally, the video streaming's session is much longer than that of the audio traffic, and packets arrive burstly (implying high self-similarity). As it can be observed in video results of the WuSched-Online, a large number of packets are served with near to zero delay, and the reason is due to the consecutive packet arrivals that are served while the inactivity timer is triggered. At the same time, a large number of packets are served with delays larger than the maximum delay budget of video (40 ms), and this comes from the fact that the WuSched-Online is a greedy method and waits until the average buffering delay approaches to $\overline{\mathrm{D}}_{max}$. As compared to the WuSched-Online, WuSched-Offline achieves similar average buffering delay (sketched with dashed vertical lines), however it has packets with longer delays (e.g., for video, there are packets with delays over 65 ms). Furthermore, WuS has a lower and consistent delay regardless of the traffic types. However, this comes at cost of an extra energy consumption (as it will be shown in Table V).

For mixed traffic flow (aggregation of video and audio traffics), the average delays are similar to video traffic rather than to audio traffic. The reason is that the delay bound plays a pivotal role in the operation of wake-up scheme, which is the same for both traffics. The small difference between mixed and video traffic comes from the inaccuracy of the traffic predictor. Additionally, the WuSched-Offline satisfies the delay requirements by optimizing the buffer size threshold based on estimated packet arrival rate and delay bound. As shown in Fig. 11, the average delays

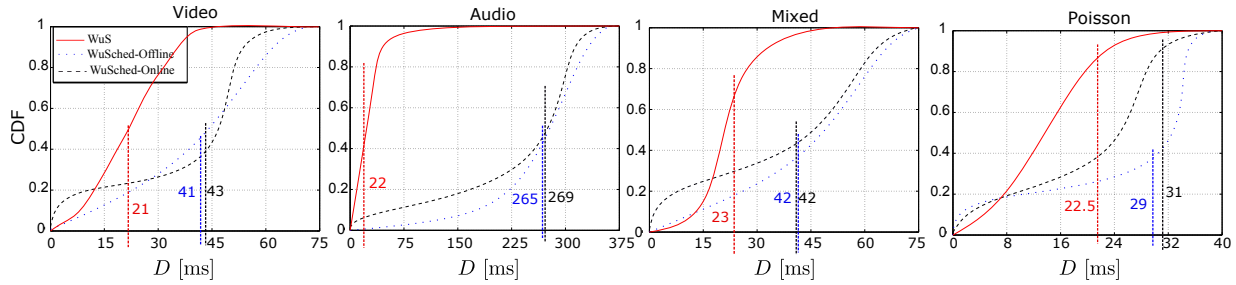


Figure 11: The CDF graphs of buffering delay of packets for all three methods under different traffic types. The dashed lines and corresponding numbers represent average delays caused by the particular method.

of the WuSched-Online for different traffic types are slightly higher than $\overline{\mathrm{D}}_{max}$, which is stemmed from prediction inaccuracy. Therefore, in order to satisfy the delay requirements, $\overline{\mathrm{D}}_{max}$ for the WuSched-Online could be set slightly lower than the actual delay requirements.

To complete the study, Table V shows the average delay and the average power consumption in third and fourth columns, respectively. It is clear that the average power consumption of WuS for all traffic types is higher than that of the WuSched-Online; however, it achieves a much lower buffering delay. Furthermore, the WuSched-Offline only outperforms the WuSched-Online for the case of Poisson traffic, and for rest of realistic types the WuSched-Online outperforms the WuSched-Offline. To illustrate the benefits of the wake-up schedulers better, we define the wasted energy (E_w) as the ratio (in percentage) of the energy that the UE consumes for transitory states plus inactivity timer over the overall energy consumption of the UE. Note that the rest of energy is consumed for processing the packets. The wasted energy E_w is shown in the fifth column of Table V. As it can be observed, the gain of the WuSched-Online is coming from having less amount of wasted energy, owing to the use of an intelligently and greedily strategy so that packets are served mainly in a consecutive manner without the need for frequent start ups and power downs. For the case of Poisson arrivals, both wake-up schedulers have similar CDF shape, with a small difference that is stemmed from prediction errors. Moreover, it can be observed that audio streaming requires lower power consumption than the rest of traffic types, due to the small packet arrivals per given time period. Furthermore, due to the fact that packets in video streaming and mixed traffic flow have much higher self-similarity characteristics, the wasted energy is slightly lower than that of other traffics.

The computational complexity of the WuSched-Offline can be less than that of the WuSched-Online due to not using the predictive framework, which requires additional processing. However, the computational complexity for a cell can be most likely kept feasible even for larger UE pop-

Table V: Average delay, power consumption and the wasted energy for different methods and traffic types.

Method	Traffic	$\overline{\mathrm{D}}$ [ms]	$\overline{\mathrm{P}}_c$ [mW]	E_w [%]
WuS	Poisson	23	600	36
	Video	21	625	44
	Audio	22	405	48
	Mixed	23	655	16
WuSched-Offline	Poisson	29	399	7
	Video	41	450	22
	Audio	265	335	37
	Mixed	42	606	10
WuSched-Online	Poisson	31	450	15
	Video	43	395	12
	Audio	269	290	26
	Mixed	42	590	7

ulations, especially in applications such as machine-type-communication, where group-specific wake-up signaling could be utilized – instead of UE-specific, which further reduces the signaling overhead. Those users that may have similar traffic type can be grouped and network can utilize the same wake-up sequences and same predictive entities. Overall, the computing capabilities in the base-stations and other network entities are continuously growing, hence we believe that executing the predictive entity is feasible when the networks evolve.

VI. Conclusions

In this work, the concept of wake-up scheduling and two optimizations (offline and online) of its parameters are proposed. The offline optimization of the wake-up scheduler is analyzed mathematically for Poisson packet arrivals. On the other hand, the feasibility of the online optimization of the wake-up scheduler based on user traffic prediction has been investigated. For this purpose, a traffic predictor which leverages on LSTM networks is also proposed. A detailed and extensive analysis comparing the power consumption and buffering delay of both wake-up schedulers was carried out, under different traffic types and various design parameters. Both wake-up schedulers were shown to facilitate a lower energy consumption compared to the wake-up scheme without scheduler. Moreover, the online optimization of the wake-up scheduler outperforms the offline one for realistic traffic types. These promising results motivate jointly considering user traffic prediction and wake-up scheduler in order to reduce the energy consumption of users under different traffic conditions.

Based on the numerical results provided in this paper, our view regarding the wake-up scheduling is that there is no 'One-Size-Fits-All Solution', unless the UE is well-defined and narrowed to a specific traffic type. Further interesting research areas include extending the proposed framework to autonomously combine and utilize different wake-up schedulers and

power saving mechanisms together, and selecting the method that better fits for particular circumstances. While FIFO was considered in this work which does not discriminate between different traffic QoS requirements, our future work will consider the weighted fair queuing for wake-up scheduling in order to satisfy the diverse QoS requirements of different services.

APPENDIX A

In this section, we prove that if T has an exponential distribution with mean $1/\lambda$, then $T' = T - t$ has the same distribution as T for $t > 0$. Due to fact that T has an exponential distribution, it has memory-less property as follows ($s \geq 0$),

$$\Pr[T > s + t | T > t] = \Pr[T > s] = e^{-\lambda s}. \tag{26}$$

Furthermore, by assuming $T' = T - t$, and based on the above equation, we can write,

$$\Pr[T' > s | T' > 0] = \frac{1 - F_{T'}(s)}{\Pr[T' > 0]} = e^{-\lambda s}, \tag{27}$$

where $F_{T'}(s)$ is the CDF of T'. Additionally, by expanding (27), we can obtain,

$$F_{T'}(s) = 1 - \Pr[T' > 0] e^{-\lambda s}. \tag{28}$$

Since s is assumed to be non-negative, therefore, $F_{T'}(0) = 0$, and based on (28), we can conclude that $\Pr[T' > 0] = 1$. As a result, we can can express that T' has an identical exponential distribution with T.

APPENDIX B

By replacing t with $D + 1$ and using the proof explained in Appendix A, we can state that $T - D - 1$ has an exponential distribution that is the same as that of T, and hence,

$$\mathrm{E}[(T - D - 1) | (D + 1 < T \leq D + 1 + t_i)] = \mathrm{E}[T | (T \leq t_i)]. \tag{29}$$

Furthermore, based on the law of total expectation, for any exponentially distributed random variable we can write,

$$\mathrm{E}[T | (T \leq t_i)] = \frac{\mathrm{E}[T] - \Pr[T > t_i]\mathrm{E}[T | (T > t_i)]}{\Pr[T \leq t_i]} = \frac{1/\lambda - e^{-\lambda t_i}(1/\lambda + t_i)}{1 - e^{-\lambda t_i}} = \frac{1}{\lambda} - \frac{t_i e^{-\lambda t_i}}{1 - e^{-\lambda t_i}}. \tag{30}$$

Then, based on (29) and (30),

$$\mathrm{E}[(T - D) | n \in X_2] = \mathrm{E}[T | (T \leq t_i)] + 1 = \frac{1}{\lambda} - \frac{t_i e^{-\lambda t_i}}{1 - e^{-\lambda t_i}} + 1. \tag{31}$$

APPENDIX C

Similarly to the derivation of $\mathrm{E}[(T - D) | n \in X_2]$ in Appendix B, by utilizing Appendix A, replacing $t = D + 1 + t_i$, and assuming $n \in X_3$ or equivalently $T > D + 1 + t_i$, we can write,

$$\mathrm{E}[(T - D - 1 - t_i) | (T > D + 1 + t_i)] = \mathrm{E}[T], \tag{32}$$

so that,

$$\mathrm{E}[(T - D) | n \in X_3] = \mathrm{E}[T] + 1 + t_i = \frac{1}{\lambda} + 1 + t_i. \tag{33}$$

APPENDIX D

The first packet that arrives in each scheduling cycle has to wait for the arrival of other $\gamma - 1$ packets plus the time period until the end of the w-cycle (referred to as t_r, as shown in Fig. 2) as well as WRx's on time (t_{on}) and the start-up period (t_{su}). Since Poisson arrivals are independently and uniformly distributed on any interval of time, we can assume that the arrival instant of the γ-th packet is uniformly distributed along the last w-cycle, which can be justified due to the relatively short length of t_w. Hence, an average extra delay of $t_w/2$ is introduced. Consequently, the mean transmission time of the first packet is delayed as follows (which is equivalent to the average holding time of the dormant period),

$$\mathrm{E}[L_d] = \frac{\gamma}{\lambda} + C_0, \tag{34}$$

where C_0 is a constant that can be obtained as follows,

$$C_0 = -\frac{1}{\lambda} + \frac{t_w}{2} + t_{on} + t_{su}. \tag{35}$$

APPENDIX E

By averaging both sides of (1), and assuming a stationary system, we can obtain,

$$\mathrm{E}[W] = \mathrm{E}[T] = \frac{1}{\lambda}. \tag{36}$$

APPENDIX F

In this section, we prove that $\mathrm{E}[T^2|(T > t_i)] = t_i^2 + 2t_i/\lambda + 2/\lambda^2$. By using the result in Appendix A, the PDF of conditional exponential distribution $T|(T > t_i)$ is the same as T with time-shift t_i, i.e., $\lambda e^{-\lambda(t-t_i)}$. Therefore, the expected value of T^2 can be obtained as follows,

$$\mathrm{E}[T^2|(T > t_i)] = \int_{t_i}^{+\infty} t^2 \lambda e^{-\lambda(t-t_i)} dt = t_i^2 + 2t_i/\lambda + 2/\lambda^2. \tag{37}$$

APPENDIX G

Due to the independence of D_n and T_n for $n \in X_d^C \cup \{N_d\}$, the covariance of D and T is zero for those packets arriving during the active period (see second row of (41)). Similarly, if $\gamma = 1$, the covariance of D and T is zero for all values of n, as written in the second row of (41). However, it is obvious that D_n for the dormant period (except the last packet in the dormant period) depends on the following packet arrivals until the end of the dormant period (provided that γ is greater than one),

$$D_n = T_n + T_{n+1} ... + T_{N_d-1} + t_r + t_{on} + t_{su} + n - 1, \tag{38}$$

for all $n \in X_d - \{N_d\}$.

In order to find $\mathrm{Cov}[D, T]$, we follow a similar approach as the one described in [26] for GI/G/1 queuing system. According to the law of total covariance, the covarinace relation between any three random variables (i.e., n, D, T) can be written as follows,

$$\mathrm{Cov}[D, T] = \mathrm{E}\big[\mathrm{Cov}[D, T|n]\big] - \mathrm{Cov}\big[\mathrm{E}[D|n], \mathrm{E}[T|n]\big]. \tag{39}$$

The above equation for the exponentially distributed T can be simplified to the following,

$$\mathrm{Cov}[D,T] = \mathrm{E}\big[\mathrm{Cov}[D,T|n]\big] - \mathrm{Cov}\big[\mathrm{E}[D|n], \frac{1}{\lambda}\big] = \mathrm{E}\big[\mathrm{Cov}[D,T|n]\big]. \tag{40}$$

Furthermore,

$$\mathrm{Cov}[D,T|n] = \begin{cases} \mathrm{Var}[T] = \dfrac{1}{\lambda^2}, & \text{for } n \in X_d - \{N_d\} \text{ and } \gamma \geq 2, \\ 0,. & \text{for } n \in X_d^C \cup \{N_d\} \text{ or } \gamma = 1. \end{cases} \tag{41}$$

Proof. By utilizing (38), the additive law of covariance, and also due to the independence of different inter-arrival times for $n \in X_d - \{N_d\}$, then,

$$\mathrm{Cov}[D,T|n] = \mathrm{Cov}[T_n, T_n] + \cdots + \mathrm{Cov}[T_{N_d}, T_n] + \mathrm{Cov}[n-1, T_n] = \mathrm{Cov}[T_n, T_n] = \mathrm{Var}[T]. \tag{42}$$

□

Based on (40) and (41), by averaging $\mathrm{Cov}[D,T|n]$ over the N packets of the scheduling cycle, we can obtain the covariance of D and T ,

$$\begin{aligned} \mathrm{Cov}[D,T] = \mathrm{E}\big[\mathrm{Cov}[D,T|n \in (X_d - \{N_d\})]\big]\, \Pr[n \in (X_d - \{N_d\})] + \\ \mathrm{E}\big[\mathrm{Cov}[D,A|n \in (X_d^C \cup \{N_d\})]\big]\, \Pr[n \in (X_d^C \cup \{N_d\})] = \\ \frac{\mathrm{E}[N_d]-1}{\mathrm{E}[N]\lambda^2} = \frac{1-\lambda}{\lambda^2(1-\lambda+e^{-\lambda t_i})}\left(1 - \frac{1}{\gamma + \lambda C_0 + 1}\right). \end{aligned} \tag{43}$$

APPENDIX H

The expected value of H_n (already expressed in (21)) can be calculated by using the law of total probability formula, as follows (summarized in third column of Table II),

$$H_n = \begin{cases} 0, & \text{for } \ n \in X_1. \\ -(T_n - D_n - 1)^2, & \text{for } \ n \in X_2. \\ {L_d}^2 - (T_n - D_n - 1)^2, & \text{for } \ n \in X_3. \end{cases} \tag{44}$$

Therefore,

$$\mathrm{E}[H] = -\Pr[n \in X_2]\mathrm{E}\big[(T-D-1)^2 | n \in X_2\big] + \Pr[n \in X_3]\mathrm{E}\big[{L_d}^2 - (T-D-1)^2 | n \in X_3\big]. \tag{45}$$

We need to calculate $\mathrm{E}[(T-D-1)^2|n \in X_2]$, $\mathrm{E}[(T-D-1)^2|n \in X_3]$ and $\mathrm{E}[L_d^2]$ before calculating $\mathrm{E}[H]$.

a) $\mathrm{E}[(T-D-1)^2|n \in X_2]$: Similar to (29), by utilizing Appendix A, we can obtain,

$$\mathrm{E}[(T-D-1)^2|n \in X_2] = \mathrm{E}[T^2|T < t_i]. \tag{46}$$

Furthermore, similar to (30) by utilizing the law of total expectation, we can obtain,

$$\mathrm{E}[T^2|(T \leq t_i)] = \frac{\mathrm{E}[T^2] - \Pr[T > t_i]\mathrm{E}[T^2|(T > t_i)]}{\Pr[T \leq t_i]} = \frac{2/\lambda^2 - e^{-\lambda t_i}(t_i^2 + 2t_i/\lambda + 2/\lambda^2)}{1 - e^{-\lambda t_i}}. \tag{47}$$

where $\mathrm{E}[T^2|(T > t_i)] = t_i^2 + 2t_i/\lambda + 2/\lambda^2$, and its proof is included in Appendix F.

b) $\mathrm{E}[(T-D-1)^2|n \in X_3]$: Similar to (33), thanks to the memory-less property of Poisson distribution, we can obtain,

$$\mathrm{E}[(T-D-1-t_i)^2|n \in X_3] = \mathrm{E}[T^2] = \frac{2}{\lambda^2}. \tag{48}$$

Furthermore, by utilizing (33) and (48), we get,

$$\begin{aligned}\mathrm{E}[(T-D-1)^2|n\in X_3] = \mathrm{E}[(T-D-1-t_i)^2|n\in X_3]+\\ 2t_i\mathrm{E}[(T-D)|n\in X_3] - t_i^2 - 2t_i = t_i^2 + 2t_i/\lambda + 2/\lambda^2.\end{aligned} \tag{49}$$

c) $\mathrm{E}[L_d^2]$: We can calculate $E[L_d^2]$, based on (34) and (35) as follows,

$$\mathrm{E}[L_d^2] = \mathrm{Var}[L_d] + \mathrm{E}[L_d]^2 = \frac{\gamma-1}{\lambda^2} + \frac{t_w^2}{12} + (\frac{\gamma}{\lambda}+C_0)^2, \tag{50}$$

where $\mathrm{Var}[t_r] = \frac{t_w^2}{12}$ is the variance of uniformly distributed t_r.

Then, by substituting (4), (5), (46), (49) and (50) into (45) while using basic sums and multiplications, we finally obtain,

$$\mathrm{E}[H] = -\frac{(1-\lambda)(e^{\lambda t_i}-1)}{\gamma+e^{\lambda t_i}+C_0\lambda}\left[\frac{2/\lambda^2}{1-e^{-\lambda t_i}}\right] + \frac{1-\lambda}{\gamma+e^{\lambda t_i}+C_0\lambda}\left[\frac{\gamma-1}{\lambda^2}+\frac{t_w^2}{12}+(\frac{\gamma}{\lambda}+C_0)^2\right]. \tag{51}$$

APPENDIX I

In this section, we prove that $\frac{d\overline{\mathrm{D}}(\gamma)}{d\gamma} > 0$ for $\gamma \geq 1$. For this purpose, the derivative of (23) with respect to γ is calculated as follows,

$$\frac{d\overline{\mathrm{D}}(\gamma)}{d\gamma} = \frac{e^{\lambda t_i}}{\lambda} z_1(\gamma) + z_2(\gamma) + z_3(\gamma), \tag{52}$$

where,

$$z_1(\gamma) = \frac{-1}{((1-\lambda)e^{\lambda t_i}+1)(\gamma+\lambda C_0+1)^2} + \frac{3/2}{(\gamma+e^{\lambda t_i}+C_0\lambda)^2}, \tag{53}$$

$$z_2(\gamma) = \frac{C_0\lambda+1}{\lambda(\gamma+e^{\lambda t_i}+C_0\lambda)^2}, \tag{54}$$

$$z_3(\gamma) = \frac{\frac{\gamma^2}{2\lambda}+(\frac{\gamma}{\lambda})(e^{\lambda t_i}+C_0\lambda)+C_0(e^{\lambda t_i}+C_0\lambda)-\frac{\lambda C_0^2}{2}-\frac{\lambda t_w^2}{24}}{(\gamma+e^{\lambda t_i}+C_0\lambda)^2}. \tag{55}$$

$z_1(\gamma)$ is positive because it can be shown that $\frac{3}{2}((1-\lambda)e^{\lambda t_i}+1)(\gamma+\lambda C_0+1)^2 \geq (\gamma+e^{\lambda t_i}+C_0\lambda)^2$. $z_2(\gamma)$ is always positive, because according to (35), $C_0\lambda+1>0$ is met. $z_3(\gamma)$ is always positive, because its numerator (refer to it as $N_{z_3}(\gamma)$) is an increasing function with respect to γ, and $N_{z_3}(1) \geq 0$ is met for all values of the parameters, so that we can conclude that $z_3(\gamma)$ is always positive for $\gamma \geq 1$. Since $z_1(\gamma)$, $z_3(\gamma)$ and $z_3(\gamma)$ are positive, then $\frac{d\overline{\mathrm{D}}(\gamma)}{d\gamma} > 0$ is demonstrated.

REFERENCES

[1] S. Rostami, H. D. Trinh, S. Lagen, M. Costa, M. Valkama, and P. Dini, "Proactive wake-up scheduler based on recurrent neural networks," in *Proc. IEEE ICC 2020*, June 2020.

[2] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, February 2014.

[3] "TS 38.300, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NR; NR and NG-RAN overall description," 3GPP, Tech. Rep., Jan. 2019.

[4] "Designing mobile devices for low power and thermal efficiency," Qualcomm Technologies Technologies, Inc., Tech. Rep., Oct. 2013.

[5] S. Rostami, K. Heiska, O. Puchko, J. Talvitie, K. Leppanen, and M. Valkama, "Novel wake-up signaling for enhanced energy-efficiency of 5G and beyond mobile devices," in *Proc. IEEE Globecom 2018*, Dec 2018, pp. 1–7.

[6] M. Lauridsen, P. Mogensen, and T. B. Sorensen, "Estimation of a 10 Gb/s 5G receiver's performance and power evolution towards 2030," in *Proc. IEEE VTC 2015 Fall*, Sept 2015, pp. 1–5.

[7] "IMT traffic estimates for the years 2020 to 2030," ITU-R, Tech. Rep., July 2015.

[8] M. Lauridsen, "Studies on mobile terminal energy consumption for LTE and future 5G," Ph.D. dissertation, Aalborg University, Jan. 2015.

[9] "LTE; evolved universal terrestrial radio access (e-utra);physical layer procedures," 3GPP TS 36.213 version 10.1.0 Release 10, Tech. Rep., APR. 2010.

[10] "5G;NR; User Equipment (UE) procedures in idle mode and in RRC inactive state," 3GPP TS 38.304 version 15.1.0 Release 15, Tech. Rep., Oct. 2018.

[11] A. T. Koc, S. C. Jha, R. Vannithamby, and M. Torlak, "Device power saving and latency optimization in LTE-A networks through DRX configuration," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2614–2625, May 2014.

[12] "UE power consideration based on days-of-use," Qualcomm Incorporated, R1-166368, Tech. Rep., Aug. 2016.

[13] A. Froytlog, T. Foss, O. Bakker, G. Jevne, M. A. Haglund, F. Y. Li, J. Oller, and G. Y. Li, "Ultra-low power wake-up radio for 5G IoT," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 111–117, March 2019.

[14] R. M. Sandoval, A. Garcia-Sanchez, J. Garcia-Haro, and T. M. Chen, "Optimal policy derivation for transmission duty-cycle constrained lpwan," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3114–3125, Aug 2018.

[15] T. Dinh, Y. Kim, T. Gu, and A. Vasilakos, "L-mac: A wake-up time self-learning mac protocol for wireless sensor networks," *Computer Networks*, vol. 105, 05 2016.

[16] N. S. Mazloum and O. Edfors, "Performance analysis and energy optimization of wake-up receiver schemes for wireless low-power applications," *IEEE Transactions on Wireless Communications*, vol. 13, no. 12, pp. 7050–7061, Dec 2014.

[17] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, "Has time come to switch from duty-cycled mac protocols to wake-up radio for wireless sensor networks?" *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 674–687, April 2016.

[18] I. Demirkol, C. Ersoy, and E. Onur, "Wake-up receivers for wireless sensor networks: benefits and challenges," *IEEE Wireless Communications*, vol. 16, no. 4, pp. 88–96, Aug 2009.

[19] S. Rostami, K. Heiska, O. Puchko, K. Leppanen, and M. Valkama, "Wireless powered wake-up receiver for ultra-low-power devices," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2018, pp. 1–5.

[20] "Study on UE power saving," 3GPP TR 38.840 version 1.0.0, Tech. Rep., Mar. 2019.

[21] M. Lauridsen, G. Berardinelli, F. M. L. Tavares, F. Frederiksen, and P. Mogensen, "Sleep modes for enhanced battery life of 5G mobile terminals," in *Proc. IEEE VTC 2016 Spring*, May 2016, pp. 1–6.

[22] S. Rostami, K. Heiska, O. Puchko, K. Leppanen, and M. Valkama, "Novel wake-up scheme for energy-efficient low-latency mobile devices in 5G networks," *IEEE Transactions on Mobile Computing*, 2020.

[23] S. Rostami, S. Lagen, M. Costa, P. Dini, and M. Valkama, "Optimized wake-up scheme with bounded delay for energy-efficient MTC," in *Proc. IEEE Globecom 2019*, Dec 2019, pp. 1–6.

[24] S. Rostami, S. Lagen, M. Costa, M. Valkama, and P. Dini, "Wake-up radio based access in 5G under delay constraints: Modeling and optimization," *IEEE Trans. on Communications*, vol. 68, no. 2, Feb 2020.

[25] C. C. Tseng *et al.*, "Delay and power consumption in LTE/LTE-A DRX mechanism with mixed short and long cycles," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1721–1734, March 2016.

[26] D. P. Heyman and K. T. Marshall, "Bounds on the optimal operating policy for a class of single-server queues," *Oper. Res.*, vol. 16, no. 6, pp. 1138–1146, Dec. 1968.

[27] N. Bui and J. Widmer, "Owl: a reliable online watcher for lte control channel measurements," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 25–30.

[28] H. D. Trinh, A. Fernandez Gambin, L. Giupponi, M. Rossi, and P. Dini, "Classification of mobile services and apps through physical channel fingerprinting: a deep learning approach," *arXiv preprint arXiv:1910.11617*, 2019.

[29] J. Wang *et al.*, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. IEEE INFOCOM 2017*, May 2017, pp. 1–9.

[30] H. D. Trinh, L. Giupponi, and P. Dini, "Mobile traffic prediction from raw data using LSTM networks," in *Proc. IEEE PIMRC 2018*, Sep. 2018, pp. 1827–1832.

[31] A. Azari, P. Papapetrou, S. Denic, and G. Peters, "User traffic prediction for proactive resource management: Learning-powered approaches," 2019.

[32] Yantai Shu, Minfang Yu, Jiakun Liu, and O. W. W. Yang, "Wireless traffic modeling and prediction using seasonal arima models," in *Proc. IEEE ICC 2003*, vol. 3, May 2003, pp. 1675–1679 vol.3.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] G. Kreitz and F. Niemela, "Spotify – large scale, low latency, p2p music-on-demand streaming," in *Proc. IEEE P2P 2010*, Aug 2010, pp. 1–10.

[35] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks," *SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 118–130, Oct. 1998.