Proactive Wake-up Scheduler based on Recurrent Neural Networks

Soheil Rostami*, Hoang Duy Trinh†, Sandra Lagen†, Mário Costa*, Mikko Valkama‡, and Paolo Dini†
*Huawei Technologies Oy (Finland) Co. Ltd, Helsinki, Finland
†Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Barcelona, Spain
‡Department of Electrical Engineering, Tampere University, Finland
emails: {soheil.rostami1, mariocosta}@huawei.com, {hdtrinh, slagen, pdini}@cttc.es, {mikko.valkama}@tuni.fi

***Abstract*—Recently, the wake-up scheme has been proposed to enhance the energy-efficiency of 5G mobile devices and prolong its battery lifetime while reducing the buffering delay. The existing wake-up optimization mechanisms use off-line methods and are tied to specific traffic models. In this paper, a novel concept of wake-up scheduling is introduced to further reduce the energy-efficiency of mobile devices and to deal with realistic traffic. The main idea is to use a fixed configuration of the wake-up scheme and adjust the scheduling of the wake-up signals dynamically. For this, a proactive wake-up scheduler is proposed to take online decisions based on traffic prediction. Towards this end, a framework to predict packet arrivals based on recurrent neural networks is developed. Numerical results show that for given delay requirements of video, audio streaming, and mixed traffic flow, the proactive wake-up scheduler reduces the power consumption of the baseline wake-up scheme without scheduler by up to 36%, 28% and 9%, respectively.**

***Index Terms*—5G, machine learning, wake-up scheme, energy efficiency, LSTM.**

I. Introduction

The emerging fifth generation (5G) mobile networks have shown a promising capability to offer futuristic mobile applications and services. Such services require an increase in data rates and enhanced quality-of-service (QoS) compared with current wireless standards, and they are realized in New Radio (NR) based 5G systems by adopting higher transmission bandwidth, higher modulation orders, advanced coding techniques, and sophisticated multi-antenna schemes [1]. However, the utilization of such computationally intensive techniques comes at the cost of higher energy consumption and can deplete the battery of mobile devices very quickly.

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 [2], [3] and NR based 5G systems [4]. However, it has been shown in [5] that the time period that a DRX-enabled mobile device spends monitoring the physical downlink control channel (PDCCH) without any data allocation has a major impact on its battery lifetime. In order to reduce the energy consumption of unscheduled cycles in DRX, the wake-up scheme (WuS) has been recently proposed in [6]. In WuS, 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 arrive at the transmission buffer of the base station, the wake-up indicator is assumed to be sent at the next upcoming wake-up instant. In our previous work [7], we introduced an off-line method to optimize the WuS configuration (i.e., the wake-up cycle period) based on a delay bound under the assumption of Poisson traffic.

Instead, in this paper we introduce a novel concept called *wake-up scheduling* to further reduce the power consumption of the mobile device. The main idea is of using a fixed WuS configuration and then adjusting the scheduling of the wake-up signals dynamically by determining whether to wake-up the device or not. In particular, we propose a proactive scheduler, which takes on-line 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 on-line wake-up scheduling decisions with traffic prediction capabilities into WuS. In addition, differently to previous works in [6], [7], the proposed scheduler is not tied to specific traffic models.

The rest of this paper is organized as follows. Section II briefly reviews the WuS principle of operation[1] and introduces the proposed wake-up scheduling concept. Then, Section III presents the proactive scheduler. These are followed by simulation results and conclusions in Sections IV and V, respectively. Terminology-wise, according to NR specification [1], we use gNB to refer to a base station and UE to denote a mobile device.

II. Wake-up Scheduling Concept

A. Review of wake-up scheme

In WuS, the cellular modem is configured with a wake-up receiver (WRx), as a companion low-complex single-purpose

This work has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 675891 (SCAVENGE), Tekes TAKE-5 project, and Spanish MINECO grant TEC2017-88373-R.

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

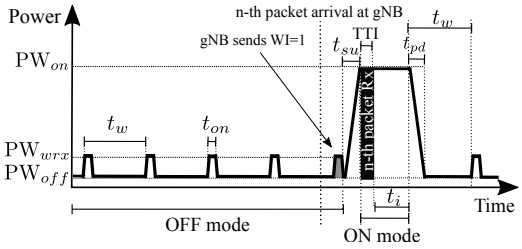

Figure 1. Operation and corresponding parameters of WuS, without scheduler.

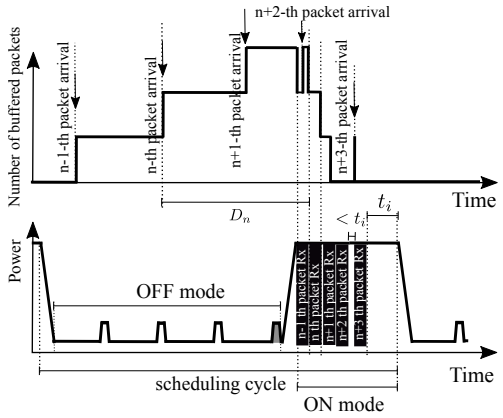

Figure 2. Wake-up scheduling concept, for the case that the gNB sends WI=1 once the buffered packets is 3.

receiver in order to decode the wake-up signaling [6]. 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 a 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 to not 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 [6]. 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 duration 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.

In WuS, if there is 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 properly optimized for the upcoming traffic, the immediate waking up of the UE can either adversely increase its energy consumption and eventually decrease 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) [7].

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 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 gNB for a given UE is larger than a predefined buffer size threshold, or until the estimated average buffering delay exceeds a predefined threshold. The former condition is illustrated in Fig. 2, where gNB does not send WI=1 until the number of buffered packets reaches to 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. In this paper, we focus on the latter condition in order to allow the network to meet maximum tolerable delays of the target applications, as explained in the next section in detail.

The main motivation behind not sending WI=1 as soon as a packet arrives at the gNB but rather waiting and sending the packets consecutively, is that the state-of-the-art modems suffer from large start-up and power-down stages [6]. 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.

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 OFF and ON modes. The scheduling cycle starts from expiry of the inactivity timer of the previous scheduling cycle, and ends by the expiry of the current cycle's inactivity timer. During the ON mode, the modem consumes the high power of PW_{on}, and either it is processing the packets or its inactivity timer is running. For the modem during OFF mode, packets are buffered, and it consumes low power of PW_{off}.

The 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 is at most one packet per TTI. TTI of 1 ms is assumed. Also, we assume that packets are served individually based on first-input first-output.

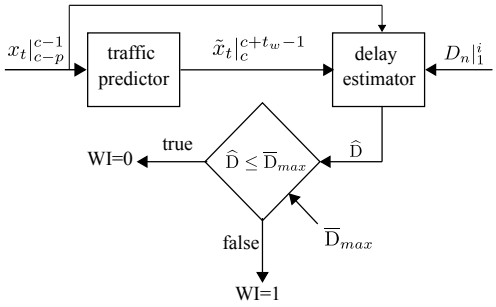

Figure 3. Overall block diagram of proactive wake-up scheduler.

III. Proactive Scheduler

Proactively knowing the packet arrival times for a forecast horizon, allows the UE to remain at OFF mode for longer periods. The proposed scheduler balances power consumption and packet delay by adaptively and autonomously determining when to send the WI, according to the traffic pattern and a maximum tolerable delay (denoted by $\overline{\mathrm{D}}_{max}$). The proactive scheduler 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. The proposed 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 $[c, c+t_w)$.

Furthermore, in 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 ($\widehat{\mathrm{D}}<\overline{\mathrm{D}}_{max}$), 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.

The overall block diagram of the proposed proactive wake-up scheduler is shown in Fig. 3. The different modules and variables are described below.

A. Dataset from real traces

In this paper, the performance of the proactive wake-up scheduler 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 [8]. 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.

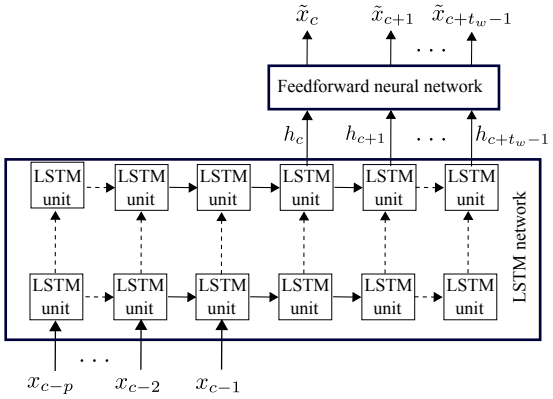

Figure 4. Proposed architecture for the packet arrival time prediction.

Our dataset includes two columns: the Identifier of the UE, and the timestamp of the packet arrival (with TTI granularity). The classifier introduced in [9] 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 (video and audio traffics have varying packet arrival rates up to 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 [10] to predict the next packet arrivals over a finite horizon. We choose LSTM since it has been proven in [10]–[12] to have lower prediction errors than other time series forecasting approaches, such as auto regressive integrated moving average (ARIMA) [13].

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. 4. 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. 3 and 4, 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

Table I
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

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 [11]. 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 III-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 I. 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 [14] as optimizer and the Mean Absolute Percentage Error (MAPE) as loss function. We define the MAPE as follows,

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

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 ($\widehat{\text{D}}$), as follows,

$$\widehat{\text{D}} = \frac{\sum_{n=1}^{i} D_n + \sum_{n=i+1}^{k} \bar{D}_n}{k}. \tag{2}$$

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

IV. NUMERICAL RESULTS

In this section, a set of numerical results are provided in order to evaluate the accuracy of the traffic predictor and validate the functionality of the proactive scheduler, for different traffic patterns.

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 [15]. Accordingly, for audio streaming, we assume that the maximum delay bound ($\overline{\text{D}}_{max}$) 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.

Power consumption of the UE in different operating states is highly dependent on the implementation, and also its operational configurations. For the numerical results, the power consumption model used in [6] and [16] is employed, for which $\text{PW}_{wrx} \approx 0$ mW, PW_{on}=850 mW, $\text{PW}_{off} \approx 0$ mW, t_{su}=15 ms, and t_{pd}=10 ms. Regarding the WuS parameters, we assume t_{on}=1/14 ms and t_i=1 ms [6].

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 Eq. (1) to quantify the accuracy of traffic prediction.

The impact of t_w and p on the prediction errors is illustrated in Fig. 5. 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 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. 5, 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 proactive scheduler, which is measured in terms of the estimated delay over a certain number of packets k (i.e., $\widehat{\text{D}}$ in Eq. (2)). In particular, the impact of traffic prediction errors on the estimated delay depends on p, k and t_w. To

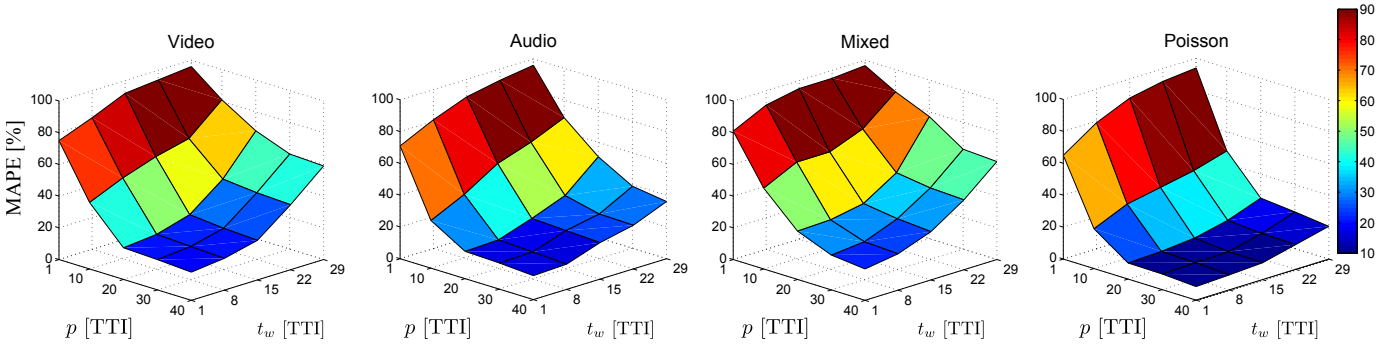

Figure 5. MAPE as function of number of past observations p and forecast horizon t_w for different traffic types.

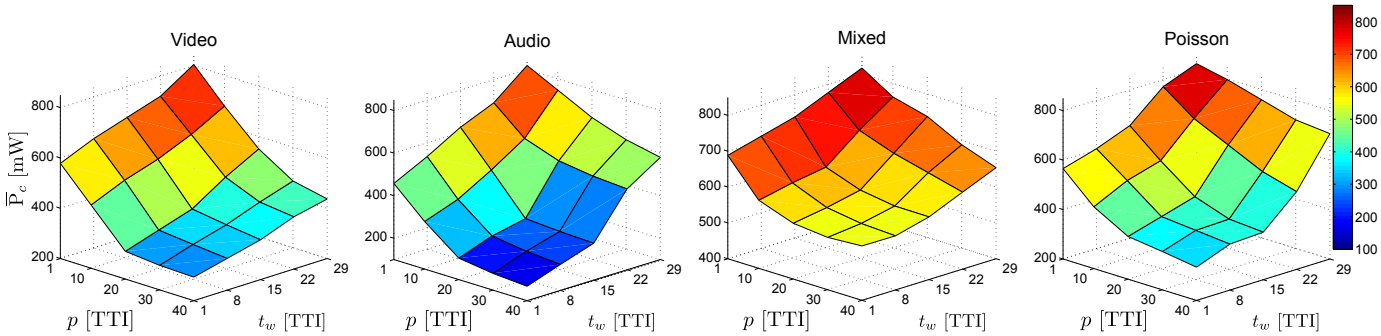

Figure 6. Power consumption of proactive scheduler as function of number of past observations p and forecast horizon t_w for different traffic types, while maintaining the corresponding delay requirements of each traffic ($k = 45$ packets).

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 Eq. (2), 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 proactive scheduler performance. Fig. 6 depicts the power consumption of the proactive scheduler 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

In this section, to validate the functionality of the proactive scheduler, the average power consumption of WuS with and without the proactive scheduler are compared for different user traffics. Two 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, and (2) wake-up scheme with proactive scheduler ('Pro.').

Fig. 7 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 seen for video results of proactive scheduler, 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 proactive scheduler is a greedy method and waits until the average buffering delay approaches to $\overline{\mathrm{D}}_{max}$. As compared to the proactive scheduler, 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 II).

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.

To complete the study, Table II shows the average delay and the average power consumption in third and fourth columns,

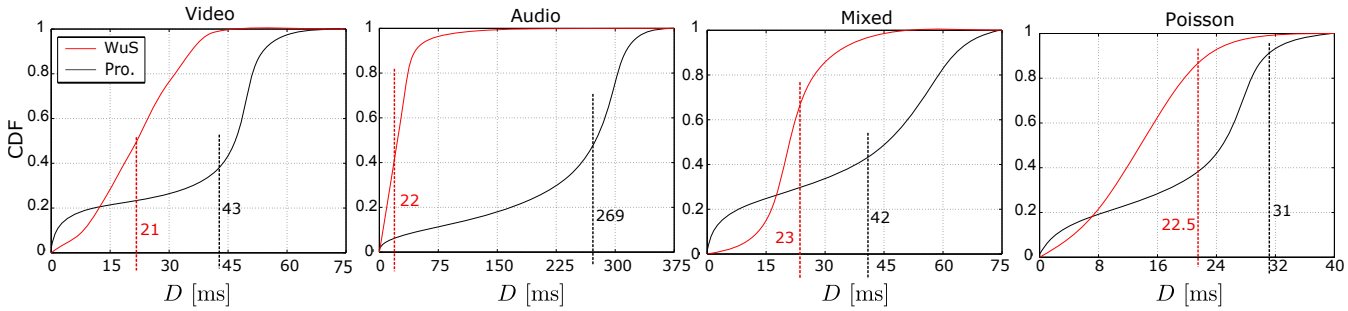

Figure 7. The CDF graphs of buffering delay of packets for WuS and proactive scheduler under different traffic types. The dashed lines and corresponding numbers represent average delays caused by the particular method.

Table II
AVERAGE DELAY, POWER CONSUMPTION, AND PERCENTAGE OF WASTED ENERGY FOR WUS AND PROACTIVE SCHEDULER UNDER DIFFERENT 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
Pro.	Poisson	31	450	15
	Video	43	395	12
	Audio	269	290	26
	Mixed	42	590	7

respectively. It is clear that the average power consumption of WuS for all traffic types is higher than that of the proactive scheduler; however, it achieves a much lower buffering delay. To illustrate the benefits of the proactive scheduler 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 II. As it can be observed, the gain of the proactive scheduler 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. 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.

V. CONCLUSIONS

In this work, the concept of wake-up scheduler, and in particular proactive scheduler are proposed. The feasibility of proactive scheduler based on user traffic prediction has been investigated. For this purpose, a traffic predictor which leverages on LSTM networks is proposed. Simulation results show that proactive scheduler has lower energy consumption than the wake-up scheme without scheduler. Moreover, the promising results motivate jointly considering user traffic prediction and wake-up scheduler in order to reduce the energy consumption of users under different traffic circumstances.

REFERENCES

[1] "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.

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

[3] "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. [Online]. Available: http://www.3gpp.org.

[4] "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. [Online]. Available: http://www.3gpp.org.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. IEEE INFOCOM 2017*, May 2017, pp. 1–9.

[11] 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.

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

[13] Y. Shu, M. Yu, J. 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.

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

[15] 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.

[16] C. C. Tseng, H. C. Wang, F. C. Kuo, K. C. Ting, H. H. Chen, and G. Y. Chen, "Delay and power consumption in LTE/LTE-A DRX mechanism with mixed short and long cycles," *IEEE Trans. on Vehicular Tech.*, vol. 65, no. 3, pp. 1721–1734, March 2016.