An Energy-Efficient Sleep Scheduling with QoS Consideration in 3GPP LTE-Advanced Networks for Internet of Things

Jia-Ming Liang, Jen-Jee Chen, Hung-Hsin Cheng, and Yu-Chee Tseng

***Abstract*—With the design of data communications in mind, 3GPP LTE-Advanced (LTE-A) is probably the most promising technology for *Internet of Things (IoT)*. For IoT applications, continuous low-rate streaming data may be reported from devices over a long period of time, imposing stringent requirements on power saving. To manage power consumption, 3GPP LTE-A has defined the *Discontinuous Reception/Transmission (DRX/DTX)* mechanism to allow devices to turn off their radio interfaces and go to sleep in various patterns. Existing literature has paid much attention to evaluate the performance of DRX/DTX; however, how to tune DRX/DTX parameters to optimize energy cost is still left open. This paper addresses the DRX/DTX optimization, by asking how to maximize the sleep periods of devices while guarantee their QoS, especially on the aspects of traffic bit-rate, packet delay, and packet loss rate in IoT applications. Efficient schemes to optimize DRX/DTX parameters and schedule devices' packets with the base station are proposed. The key idea of our schemes is to balance the impacts between QoS parameters and DRX/DTX configurations. Simulation results show that our schemes can guarantee traffic bit-rate, packet delay, and packet loss rate while save energy of UEs.**

***Index Terms*—Long Term Evolution-Advanced (LTE-Advanced), Discontinuous Reception/Transmission (DRX/DTX), Internet of Things (IoT), Power Saving, Quality of Service, Sleep Scheduling.**

I. Introduction

Internet of Things (IoT) is a general idea to integrate numerous devices or machines with the Internet. For IoT applications, such as video surveillance [1] and smart metering [2], devices need to report various events and streaming data to a central server over a long period of time in an efficient and robust way. Thus, the 3GPP LTE-Advanced (LTE-A), which is designed with wireless data communications in mind, is the most promising technology for IoT applications. To accommodate various streaming data of IoT applications, the LTE-A standard has defined several *quality-of-service (QoS)* classes for different traffic characteristics on the aspects of traffic bit-rate, tolerable delay, and packet loss rate [3]–[5]. On the other hand, since IoT devices need to continuously report data over a long period of time, the requirements on power saving are more stringent. To save the energy of devices (also called *user equipments, UEs*), the LTE-A standard has defined the *Discontinuous Reception/Transmission (DRX/DTX)* mechanism to allow devices to turn off their radio interfaces and go to sleep when no data needs to be received or transmitted from/to the base station (also called *evolved Node B, eNB*). The key property of the DRX/DTX mechanism is to work in coordination with an eNB and its UEs and regulate UEs to wake up periodically to receive/transmit data from/to the eNB. Then, UEs can turn off their wireless transceivers during the non-wake-up period to save energy. Particularly, each UE adopts a specific timer to prolong its wake-up period whenever it sees the data coming before the timer expires. Thus, some data posing unexpected delay can still be received/transmitted after the regular wake-up periods. However, how to tune DRX/DTX parameters to minimize UEs' energy costs is still left as an open issue in LTE-A.

In this paper, we address the DRX optimization problem[1] with the consideration of UEs' QoS requirements. The objective is to maximize UEs' sleep periods (i.e., non-wake-up periods) to save their energy while satisfy their QoS requirements in terms of traffic bit-rate, packet delay, and packet loss rate. We propose an efficient sleep scheduling scheme and a packet scheduling method to tackle this problem. The key idea of these schemes is to balance the impacts between QoS parameters and DRX configurations.

Major contributions of this paper are three-fold. First, this is the first work to address the joint optimization on energy saving and QoS guarantee for IoT applications in the 3GPP LTE-A network. In addition to the generic traffic features, the extra packet delay posed by IoT devices is also considered. Second, we develop an efficient sleep scheduling scheme to optimize the DRX mechanism which can fit all the packet delay probability models and effectively mitigate the packet loss issue and guarantee the traffic bit-rate requirements of IoT applications. In addition, a DRX-aware packet scheduling scheme is also proposed to well cooperate with the proposed sleep scheme to improve the performance on energy saving and QoS satisfaction. Extensive simulations show that our schemes can satisfy UEs' QoS while incurring lower energy consumption as compared to existing results. Third, through the simulation results, we give a constructive summary to conclude the limitation of the current DRX mechanism in the standard for IoT applications and provide some suggestions

J.-M. Liang, H.-H. Cheng, and Y.-C. Tseng are with the Department of Computer Science, National Chiao-Tung University, Hsin-Chu 30010, Taiwan (e-mail: jmliang@cs.nctu.edu.tw, hunghsin@cs.nctu.edu.tw, yctseng@cs.nctu.edu.tw).

J.-J. Chen is with the Department of Electrical Engineering, National University of Tainan, Tainan 70005, Taiwan. (e-mail: jjchen@mail.nutn.edu.tw).

[1]The DTX optimization problem is similar to the DRX problem.

TABLE I
STANDARDIZED QCI CHARACTERISTICS IN LTE-A

QCI	Resource Type	Packet Delay Budget	Packet Loss Rate	Example Services
1	GBR	100 ms	10^{-2}	Conversational Voice
2	GBR	150 ms	10^{-3}	Conversational Video (Live Streaming)
3	GBR	50 ms	10^{-3}	Real-Time Gaming
4	GBR	300 ms	10^{-6}	Non-Conversational Video (Buffered Streaming)
5	Non-GBR	100 ms	10^{-6}	IMS Signaling
6	Non-GBR	300 ms	10^{-6}	Video (Buffered Streaming), TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7	Non-GBR	100 ms	10^{-3}	Voice, Video (Live Streaming), Interactive Gaming
8	Non-GBR	300 ms	10^{-3}	Video (Buffered Streaming), TCP-based
9	Non-GBR	300 ms	10^{-6}	Video (Buffered Streaming), TCP-based

for the devolvement of future standard.

The rest of this paper is organized as follows. Related work is discussed in Section II. Preliminaries are given in Section III. Section IV presents our schemes. Extensive simulation results are given in Section V. Conclusions are drawn in Section VI.

II. Related Work

In the literature, performance analyses of the DRX mechanism in LTE-A networks are conducted in [6]–[10], which all show that enabling DRX can significantly save UEs' energy. Reference [11] uses hierarchical cascaded power gating and multi-level clock gating to reduce the power consumption at the physical layer in DRX cycles. Reference [12] proposes a "light sleep" approach to turn off UEs' power amplifiers to further reduce the consumed power in wake-up periods. In reference [13], a packet scheduling scheme is proposed for the eNB which prefers allocating resource to the UE whose "inactivity" timer is going to expire first. Thus, the selected UE is more likely to catch packets in time before sleeping, thus reducing its packet loss rate. To reduce UEs' power cost, [14] tries to derive the optimal number of active slots in a frame according to the physical structure when DRX operates. However, these studies [11]–[14] neglect the coordination between various traffic characteristics and DRX configurations. Reference [15] proposes a dynamic DRX scheme which continuously lengthens the DRX cycle and inactivity timer if no data needs to be received by UEs. However, it costs a large amount of signaling overheads to negotiate these adjustments between the eNB and UEs. Reference [16] proposes an autonomous scheme incurring low signaling overheads which can adaptively adjust DRX cycles to capture the UE's incoming traffic characteristic to improve energy efficiency. In [17], the *channel quality identifier (CQI)* is considered to adjust the DRX inactivity timer for UEs with different CQIs to improve system utility. However, both [16] and [17] do not consider the higher-level QoS features such as the traffic bit-rate and packet loss rate, which are mandatory in LTE-A network. These observations motivate us to address the DRX optimization problem.

III. Preliminaries

In this section, we first introduce the QoS features designed in the LTE-A. Then, we describe the operation of the DRX mechanism. Finally, we formally define our DRX optimization problem.

A. QoS in LTE-A

In the LTE-A network, there are two types of flows:

- *Guaranteed-Bit-Rate* (GBR)
- *Non-Guaranteed-Bit-Rate* (non-GBR).

A GBR flow can support real-time services, such as conversational voice, video, and gaming applications, while a non-GBR flow can support non-real-time services, such as IMS signaling and TCP-based applications [18]. A GBR flow is associated with some QoS parameters such as *guaranteed-bit-rate* and *maximum-bit-rate*. The former is the minimum reserved traffic rate (bits/s) guaranteed by the eNB. The latter is the maximum sustained traffic rate (bits/s) that the flow can not exceed. All non-GBR flows share a common QoS parameter: *aggregate-maximum-bit-rate*, which is the amount of traffic rate (bits/s) shared by all non-GBR flows of a UE. In addition, each flow (including GBR and non-GBR flows) is further associated with a QoS profile including:

- *QoS Class Identifier*
- *Packet Delay Budget*
- *Packet Loss Rate*

The *QoS Class Identifier (QCI)* is a scalar identifier to describe the traffic characteristics in terms of *packet delay budget* and *packet loss rate*. The packet delay budget is the maximum waiting time (in ms) that a packet delivered from the eNB to the UE. The packet loss rate is the probability that a packet arrives at the eNB but is not received by the UE. This may happen when a buffered packet passes its delay budget. Here, we also investigate the impact of *service-request-response (SRS) time* (in ms) for non-GBR flows. The SRS is the maximum waiting time for the service request of the applications to be delivered from the UE to the eNB. Usually, SRS time is lager than the packet delay budget. Table I summarizes the QoS characteristics in LTE-A.

B. Discontinuous Reception (DRX) Mechanism

In LTE-A, the DRX mechanism is managed by the *Radio Resource Control (RRC)*. An eNB can initiate the DRX mechanism by sending a *Command MAC control element* to a UE [19]. The DRX configurations are UE-specific. Each UE has its own configurations which are determined by the eNB.

TABLE II
SUMMARY OF THE DRX PARAMETERS

notation	definition
drxStartOffset	the subframe where the DRX cycle starts
on-duration	the number of subframes at the beginning of a DRX cycle that the UE waits to receive PDCCHs
drx-InactivityTimer	the number of consecutive subframes that the receiver is turned on after receiving a PDCCH
shortDRX-Cycle	the short periodic repetitions of the on-duration
longDRX-Cycle	the long periodic repetitions of the on-duration, which is followed by several shortDRX-Cycles
drxShortCycleTimer	the number of consecutive subframes that the UE shall follow the shortDRX-Cycle to start longDRX-Cycle

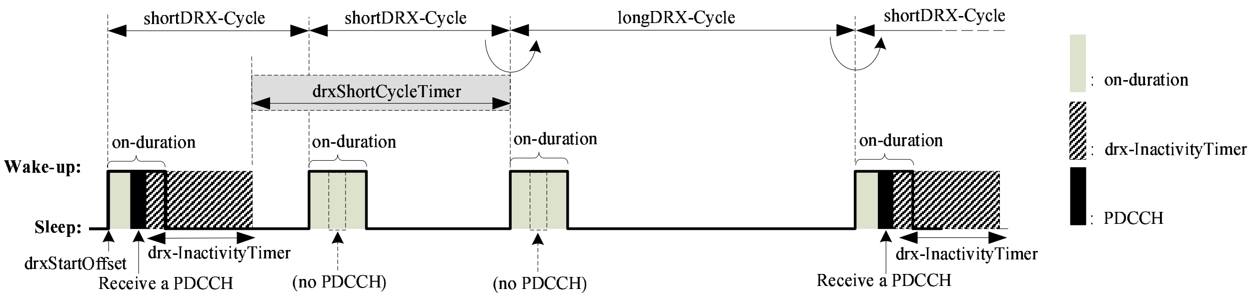


Fig. 1. An overview of the DRX operation.

When DRX is enabled, a UE wakes up and sleeps with specific patterns, as shown in Fig. 1. The basic unit of wake-up and sleeping durations is a subframe (i.e., 1 ms). When the DRX mechanism is activated, there are six parameters to be specified for each UE: 1) shortDRX-Cycle, 2) on-duration, 3) drxStartOffset, 4) drx-InactivityTimer, 5) longDRX-Cycle, and 6) drxShortCycleTimer. The shortDRX-Cycle and longDRX-Cycle are the basic operation periods (in subframes) that the UE performs wake-up and sleep operations. Usually, the length of longDRX-Cycle is a multiple of the length of shortDRX-Cycle. The on-duration is an interval (in subframes) in a cycle that the UE has to stay awake. During the wake-up period, the UE will monitor whether or not there is a *Physical Downlink Control Channel (PDCCH)* delivered from the eNB to indicate any downlink transmission to it. The drxStartOffset indicates the subframe where the first on-duration starts. The drx-InactivityTimer is used for extending the wake-up period of the UE when finds any PDCCH delivered to it. Before drx-InactivityTimer expires, if the UE monitors a new PDCCH from the eNB, the drx-InactivityTimer resets and restarts to count down again. Once the drx-InactivityTimer expires, the UE will start drxShortCycleTimer and go to sleep (by turning off its interface). During the UE's sleep period, all data for the UE will be buffered in the eNB until the next on-duration comes. If no PDCCH is monitored by the UE during several shortDRX-Cycles, the drxShortCycleTimer will expire. Once the drxShortCycleTimer expires, the shortDRX-Cycle ends and the longDRX-Cycle follows. During the longDRX-Cycle, the UE behaves similarly as it works in the shortDRX-Cycle. Once the UE monitors the PDCCH, it terminates the longDRX-Cycle and starts the shortDRX-Cycle again. The DRX parameters are summarized in Table II.

We observe that the DRX configurations, QoS requirements, and power constraints of a UE are tightly coupled with each other. For example, a shorter shortDRX-Cycle can reduce a UE's packet delay but decrease its sleep period. Also, a UE with a longer on-duration can enjoy a higher traffic bit-rate but it incurs higher power cost to the UE. Further, a UE with a larger drx-InactivityTimer can catch more packets to reduce the packet loss rate. However, it decreases the UE's sleep periods. Finally, a shorter longDRX-Cycle can reduce the SRS time of a UE but reduce its sleep period. Therefore, how to configure DRX parameters is a critical problem.

C. The DRX Optimization Problem

We consider the downlink transmissions of an eNB serving N UEs under Time Division Duplex (TDD) mode[2]. Each UE$_i$,$i = 1..N$, has admitted F_i^G GBR flows and F_i^N non-GBR flows, and each GBR flow$_j$ has a guaranteed-bit-rate R_j^G (bits/s) and all non-GBR flows share an aggregate-maximum-bit-rate R_i^N (bits/s). For each flow$_j$ (including GBR and non-GBR flows), it has a QoS profile in terms of *packet delay budget* D_j (ms) and allowable *packet loss rate* P_j^{loss}. The packet size of a flow may vary over time due to its IoT application. We assume that the packet size ranges from Q_j^{min} to Q_j^{max} (bits/packet). The expected inter-arrival time of the packets of flow$_j$ is Z_j ms. In addition, each non-GBR flow$_j$ has a service-request-response time S_j (ms) based on its IoT application, which is larger than its packet delay budget, i.e., $S_j \gg D_j$. In this paper, we assume that each packet of flow$_j$ of UE$_i$ has a remaining packet delay budget $D_j - t$ when being processed by our scheduler, where t has a probability mass function $P_{i,j}(t)$. The extra delay of t may be incurred by networks' latency (in the case of downlink transmissions) or stream processing, compressing, coding, or packing latency (in the case of uplink transmissions). In each subframe, the basic allocation unit for a UE is a *resource block (RB)*. Suppose that there are Ω RBs in a subframe. Note that the UE with a higher channel quality can receive more data bits in a RB. Let C_i (bits/RB) be UE$_i$'s channel rate which may vary over time and be measured during its wake-up period. We assume that C_i ranges from C_i^{min} to C_i^{max} (bits/RB). The DRX optimization

[2]The uplink transmissions are similar to the downlink ones.

problem asks how to schedule resources and optimize the DRX parameters of each UE$_i$, including the on-duration (O_i), drxStartOffset (L_i), shortDRX-Cycle (T_i^S), longDRX-cycle (T_i^L), drx-InactivityTimer (Γ_i^I), and drxShortCycleTimer (Γ_i^L) such that the QoS requirements of UE$_i$ (i.e., R_j^G, R_i^N, D_j, and P_j^{loss}) can be met while the sum of sleep periods of all UEs can be maximized.

IV. The Proposed Scheme

In this section, we present our three-stage (TS) scheme to the DRX optimization problem. Once the parameters (i.e., T_i^S,T_i^L,O_i, L_i, Γ_i^I, Γ_i^L) of each UE$_i$ are determined, they will be sent to each UE. On the other hand, a packet scheduling is proposed for the eNB to cooperate with UEs. Our TS scheme maintains three key properties to reduce UEs' wake-up periods. First, we make all UE$_i$'s DRX cycle be an integer multiple of others'. This reduces UEs' unnecessary wake-up periods incurred by resource competition. Second, we also optimize the drx-InactivityTimer (we use "InactivityTimer" for short) and help UEs to catch the packets posing unexpected delays and thus to meet their delay budgets. Third, we allow UEs to go to "deep" sleep when their service-request-response times are not violated. As the results, our scheme can save significant energy and is quite suitable for IoT applications. The details of the scheme are described as follows.

A. Stage 1: Determining T_i^S and T_i^L

To decide T_i^S of each UE$_i, i = 1..N$, we first find the strictest delay budget of each UE$_i$ (denoted as D_i^{min}):

$$D_i^{min} = \min_j \{D_j | flow_j \in UE_i\}. \tag{1}$$

Without loss of generality, let $D_1^{min} \leq D_2^{min} \leq ... \leq D_N^{min}$. Let $T_1^S \leq D_1^{min}$. We determine $T_i^S, i = 2..N$, as follows:

$$T_i^S = \left\lfloor \frac{D_i^{min}}{T_{i-1}^S} \right\rfloor \times T_{i-1}^S. \tag{2}$$

Eq. (2) implies that $T_i^S \leq D_j$ for all flow$_j$ in UE$_i$ because $T_i^S \leq D_i^{min} \leq D_j$. Since all flows' packets of UE$_i$ are served with a cycle T_i^S, this guarantees that all packets will meet their delay budgets. Also note that Eq. (2) ensures T_i^S to be an integer multiple of T_{i-1}^S for $i = 2..N$. This can help UEs to interleave their wake-up periods and avoid the competition for resources among UEs. Here, T_1^S is the basic cycle and the allocation pattern will repeat after T_N^S/T_1^S basic cycles due to our arrangement (this will be clear later on).

To decide T_i^L of each UE$_i, i = 1..N$, we first find the strictest service-request-response time, denoted as S_i^{min}, among all non-GBR flows in UE$_i$:

$$S_i^{min} = \min_j \{S_j | flow_j \in UE_i\}. \tag{3}$$

Since the size of longDRX-Cycle T_i^L of UE$_i$ should be an integer multiple of the size of its shortDRX-Cycle T_i^S, and T_i^L must be less than or equal to S_i^{min}, we set $T_i^L, i = 1..N$, as follows:

$$T_i^L = \left\lfloor \frac{S_i^{min}}{T_i^S} \right\rfloor \times T_i^S. \tag{4}$$

Note that Eq. (4) implies that $T_i^L \leq S_i^{min} \leq S_j$ for all non-GBR flow$_j \in$ UE$_i$. Therefore, once a service request arrives in a long cycle T_i^L, it can guarantee the request to be served within $T_i^L \leq S_j$. Therefore, the service response time of all non-GBR flows in UE$_i$ can be met.

B. Stage 2: Determining O_i, Γ_i^I, and Γ_i^L

To determine the on-duration O_i of each UE$_i, i = 1..N$, we first calculate the sum of the maximum packet sizes of the flows in UE$_i$, whose delay budget is equal to its shortDRX-Cycle length T_i^S, i.e., $\sum_{D_j = T_i^S, \forall flow_j \in UE_i} Q_j^{max}$. Then, O_i is set as follows:

$$O_i = \max \left\{ \left\lceil \frac{\sum_{D_j = T_i^S, \forall flow_j \in UE_i} Q_j^{max}}{C_i^{min} \times \Omega} \right\rceil, 1 \right\}. \tag{5}$$

We can see that by reserving O_i subframes as UE$_i$'s on-duration, the most urgent packet of flow$_j$ is able to be served during the shortDRX-Cycle. Here, the most urgent packet is the packet with the delay budget equal to T_i^S and arrives at the beginning of the shortDRX-Cycle. This packet has to be received by UE$_i$ before the cycle ends; otherwise, it will be dropped. Note that Eq. (5) also implies that UE$_i$ uses the least number of necessary wake-up subframes by reserving necessary resource for urgent packets only, which can reduce the periodic wake-up periods of UEs.

For determining the InactivityTimer Γ_i^I of each UE$_i, i = 1..N$, we first model the *expected packet loss rate*, denoted by $\mathbb{E}_{i,j}(\cdot)$, for flow$_j$ in UE$_i$ by making use of its packet delay probability $P_{i,j}(t)$. Then, a temporal InactivityTimer for each flow$_j$ is chosen to satisfy the flow's packet loss rate. Finally, the best InactivityTimer is determined for UE$_i$ to meet all its flows' packet loss rate. The detail of the procedure is described as follows.

- Let M_j be the number of packets of flow$_j$ that should arrive during D_j ms (thus, $M_j = \max\{\lfloor \frac{D_j}{Z_j} \rfloor, 1\}$). Each packet $m, m = 1..M_j$, may pose delay t_m ms, $t_m = 1..D_j$ (we regard a packet to be lost if it is delayed over D_j ms). Let $\hat{t}_m$ be the subframe number that packet m arrives to the eNB and can be served by the eNB. Without loss of generality, we regard the first subframe after UE$_i$'s first on-duration ends as the subframe number 1. So we have

$$\hat{t}_m = t_m + (m-1) \times Z_j + T_{offset} + \Delta_j, \tag{6}$$

where T_{offset} is the expected subframe number of the first arrived packet of flow$_j$ after UE$_i$'s first on-duration ends and Z_j is the expected packet inter-arrival time of flow$_j$. Note that $\Delta_j = \left\lceil \frac{n_j^{RB}}{a_j^{RB}} \right\rceil$ is the expected latency for serving a packet of flow$_j$ through the network bandwidth shared by all UEs in the network, where $n_j^{RB} = \frac{Q_j^{min} + Q_j^{max}}{C_i^{min} + C_i^{max}}$ is the average number of RBs to serve a packet of flow$_j$ in UE$_i$ and a_j^{RB} is the average number of available RBs that can allocate to the packet per subframe defined by

$$a_j^{RB} = \left(\frac{FlowRate_j}{\sum_{i=1..N} R_i^N + \sum_{i=1..N} \sum_{flow_j \in UE_i} R_j^G} \right) \times \Omega. \tag{7}$$

Note that the first part of Eq. (7) is the average number of available RBs per subframe for flow$_j$ and $FlowRate_j$ is the admitted bit-rate of flow$_j$:

$$FlowRate_j = \begin{cases} R_j^G, & \text{if flow}_j \text{ is GBR} \\ \frac{R_i^N}{F_i^N}, & \text{if flow}_j \text{ is non-GBR.} \end{cases} \tag{8}$$

Then, the expected packet loss rate $\mathbb{E}_{i,j}(\cdot)$ with the temporal InactivityTimer $\hat{\Gamma}_j^I$ of flow$_j$ in UE$_i$ can be expressed as follows:

$$\begin{aligned} &\mathbb{E}_{i,j}(\hat{\Gamma}_j^I, [t_1, t_{2,}.., t_m, .., t_{M_j}], D_j, T_i^S) \\ &= \sum_{t_m=1..D_j, \forall m} Prob([t_1, t_{2,}.., t_m, .., t_{M_j}]) \\ &\quad \times Loss(\hat{\Gamma}_j^I, [t_1, t_{2,}.., t_m, .., t_{M_j}], D_j, T_i^S), \end{aligned}$$

where $Prob(\cdot)$ and $Loss(\cdot)$ are the probability and the packet loss ratio function of the packet delay distribution $[t_1, t_{2,}.., t_m, .., t_{M_j}]$ such that

$$Prob([t_1, t_{2,}.., t_m, .., t_{M_j}]) = \prod_{m=1..M_j} P_{i,j}(t_m), \tag{9}$$

and

$$\begin{aligned} &Loss\left(\hat{\Gamma}_j^I, [t_1, t_2, .., t_m, .., t_{M_j}], D_j, T_i^S\right) \\ &= \frac{M_j - \left(\sum_{m=1..M_j} (\phi_m + \eta_m)\right)}{M_j}, \end{aligned} \tag{10}$$

where

$$\phi_m = \begin{cases} 1, & \text{if } X_m \le D_j \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

and

$$\eta_m = \begin{cases} 1, & \text{if } X_m > D_j \text{ and } Y_m \le \hat{\Gamma}_j^I \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

In Eq. (10), the denominator is the total number of arrival packets during the delay budget and the numerator is the number of packets failed to be received by UE$_i$ due to the expiration of InactivityTimer $\hat{\Gamma}_j^I$. In addition, ϕ_m (in Eq. (10) and Eq. (11)) is an indicator that returns 1 if the arrival of packet m can be received by UE$_i$'s on-duration of the cycle; otherwise, it returns 0. Term η_m (in Eq. (10) and Eq. (12)) is also an indicator that returns 1 if the arrival of packet m can be received by the InactivityTimer; otherwise, it returns 0. Note that X_m (in Eq. (11) and Eq. (12)) is used to evaluate the waiting time if packet m is received until the next on-duration of UE$_i$, i.e., $X_m = \left\lceil \frac{\hat{t}_m}{T_i^S} \right\rceil \times T_i^S - T_{offset} - (m-1) \times Z_j$. Term Y_m in Eq. (12) is the interval between the arrivals of packet m and packet y, where packet y satisfies $\hat{t}_y \le \hat{t}_m$ and is successfully received by the on-duration or InactivityTimer of UE$_i$, i.e.,

$$Y_m = \begin{cases} \hat{t}_m - (\lceil \frac{\hat{t}_y}{T_i^S} \rceil \times T_i^S), & \text{if } \phi_y = 1 \\ \hat{t}_m - \hat{t}_y, & \text{if } \eta_y = 1. \end{cases} \tag{13}$$

Thus, it implies that once packet m arrives at the subframe nearby that of packet y which is received successfully by the UE's on-duration or the InactivityTimer, the packet m can also be received by UE$_i$ through the extended InactivityTimer triggered by packet y.

- Then, we choose a temporal InactivityTimer $\hat{\Gamma}_j^{I*}$ to meet the required packet loss rate of flow$_j$, i.e.,

$$\hat{\Gamma}_j^{I*} = \min \left\{ \hat{\Gamma}_j^I | \mathbb{E}_{i,j}(\hat{\Gamma}_j^I) \le P_{i,j}^{loss}, \hat{\Gamma}_j^I = 0, 1.. \right\}. \tag{14}$$

Note that a shorter InactivityTimer $\hat{\Gamma}_j^{I*}$ can potentially reduce the wake-up period of the UE when the required packet loss rates are the same.

- Finally, the best InactivityTimer Γ_i^{I*} is chosen for UE$_i$ so as to satisfy the packet loss rate of all the flows in UE$_i$:

$$\Gamma_i^{I*} = \max \left\{ \hat{\Gamma}_j^{I*} | flow_j \in UE_i \right\}. \tag{15}$$

Note that to speed up the calculation of the best InactivityTimer for UEs, we can use a larger unit $\rho \ge 1$ (in ms) for packet delay budget, InactivityTimer, and cycle length, thus the expected packet loss rate incurred by the temporal InactivityTimer $\hat{\Gamma}_j^I$ can be rewritten as $\mathbb{E}_{i,j}(\frac{\hat{\Gamma}_j^I}{\rho}, [t_1, t_{2,}.., t_m, .., t_{M_j}], \frac{D_j}{\rho}, \frac{T_i^S}{\rho})$. This can significantly reduce the computational cost. In addition, because the best temporal InactivityTimers of the flows are the same if the flows are with the same traffic characteristics (i.e., packet inter-arrival time, packets loss rate, packet delay budget, and packet delay distribution), we can record these information and corresponding best InactivityTimers for further accelerating the computation.

For determining Γ_i^L of each UE$_i, i = 1..N$, we set $\Gamma_i^L = \lceil \frac{S_i^{max}}{T_i^S} \rceil \times T_i^S - ((f - \text{drxStartoffset}) \% T_i^S)$, where f is the subframe number to trigger drxShortCycleTimer. Here, depending on the applications behavior, if no packet arrives over the maximal service request-time, i.e., $S_i^{max} = \max\{S_j | flow_j \in UE_i\}$, it has a higher probability that there will be no packet arrival later. This feature can be used for the UE to go to deep sleep for further conserving energy.

C. Stage 3: Determining L_i

To determine L_i of each UE$_i, i = 1..N$, we first define the "crowded" degree for each cycle. Then, we recursively assign each UE a less crowded cycle in which the UE starts its DRX operation to avoid resource contention. Finally, for each cycle, to mitigate the resource competition among UEs, we disperse these UEs by assigning different drxStartOffsets (we use "startoffsets' for short) in the cycle, thus reducing unnecessary wake-up periods of the UEs. The detail of the procedure is described as follows.

- First, let s_k be the first available subframe in each cycle $k, k = 1..\frac{T_N^S}{T_1^S}$. Initially, set $s_k = 1$. Then, we define the

"crowded" degree $\mathbb{C}_k$ for cycle k as the total amount of the resident UEs' least wake-up time, where the least wake-up time of UE$_i$ in each shortDRX-Cycle is $O_i+\Gamma_i^I$. Initially, set $\mathbb{C}_k=0$.
- Second, we recursively assign each UE$_i$ a temporal startoffset $\hat{L}_i$, which is composed of two parameters, φ_i^{cycle} and φ_i^{offset}, where φ_i^{cycle} is UE$_i$'s resident cycle and φ_i^{offset} is the startoffset in the φ_i^{cycle}-th cycle, i.e., $\hat{L}_i=(\varphi_i^{cycle}-1)\times T_1^S+\varphi_i^{offset}$. For UE$_i$, we choose the cycle k^* which is with the smallest $\mathbb{C}_{k*}$ among the first $\frac{T_i^S}{T_1^S}$ cycle. Then, we give a temporal startoffset by assigning $\varphi_i^{cycle}=k^*$ and $\varphi_i^{offset}=s_{k^*}$ and update $s_{k^*}=s_{k^*}+O_i+\Gamma_i^I$. For $n_i=1..\frac{T_N^S}{T_i^S}-1$, we repeatedly update $s_{k^*+n_i\times\frac{T_i^S}{T_1^S}}=s_{k^*+n_i\times\frac{T_i^S}{T_1^S}}+O_i+\Gamma_i^I$ due to its cyclic feature. Finally, update $\mathbb{C}_k$ for each of these updated cycles accordingly.
- Third, based on the results of previous step, we proportionally redistribute the drxStartOffset L_i for each UE$_i$. First, we derive $\psi=\max_{k=1..\frac{T_N^S}{T_1^S}}\{\mathbb{C}_k\}$. Then, we assign each UE$_i$'s L_i according to the ratio $\frac{T_1^S}{\psi}$ such that $L_i=(\varphi_i^{cycle}-1)\times T_1^S+\lfloor\frac{T_1^S}{\psi}\times\varphi_i^{offset}\rfloor$.

D. Packet Scheduling at the eNB

At the eNB, we design a DRX-aware scheduling scheme to cooperate with the proposed TS scheme. This scheduler will keep aware of UEs' DRX operations and maintain a additional virtual queue for each UE to collect the packets which will be due before its current cycle ends. Specifically, when allocating data in a subframe, the eNB will allocate stringent data first. A packet is considered stringent if it is in the virtual queue and will be dropped in the next subframe. Also, for those UEs whose InactivityTimers will expire at the next subframe, our packet scheduler will allocate them one RB if their virtual queues are not empty. Finally, the remaining RBs of the subframe will be allocated to the UEs with buffered data and are with higher channel rate as compared to their average channel rates to improve transmission efficiency.

E. Time Complexity Analysis

We analyze the time complexity of the proposed TS scheme as follows. In the stage one, it costs $O(F_i^G+F_i^N)$ to find the strictest delay for UE$_i$ and costs $O(N\log N)$ to sort all UEs' strictest delays. Then, to determine UEs' shortDRX-Cycle lengths costs $O(N)$. Similarly, it costs $O(F_i^N)$ to find the strictest SRS time for each UE$_i$. Then, to determine UEs' longDRX-Cycle lengths costs $O(N)$. Let $F=\sum_{i=1..N}(F_i^G+F_i^N)$ be the total number of flows in the network and thus the stage one totally costs $\big(O(\sum_{i=1..N}(F_i^G+F_i^N))+O(N\log N)+O(N)\big)+\big(O(\sum_{i=1..N}F_i^N)+O(N)\big)=O(F+N\log N)$.

In the stage two, to sum up the maximum packet size of UE$_i$'s flows costs $O(F_i^G+F_i^N)$. Thus, all UEs cost $O(F)$ to determine all their on-duration lengths. Next, it costs $O((\frac{D_j}{\rho})^{M_j}\cdot M_j)$ to calculate the expected packet loss rate for each flow$_j$ because flow$_j$ has $(\frac{D_j}{\rho})^{M_j}$ delay distributions, where $\rho\geq 1$ is the unit to speed up the calculation for expected packet loss rates and each distribution costs $O(M_j)$ to calculate the value of $Loss(\cdot)$. Because the network has $O(F)$ flows, each flow costs at most $\frac{T_i^S}{\rho}(\leq\frac{D_j}{\rho})$ times to derive its expected packet loss rate to find the best InactivityTimer. Thus, all UEs costs $O(F)\cdot O(\frac{D_j}{\rho}^{M_j+1}\cdot M_j)$ to find their best InactivityTimers to satisfy their target packet loss rates. Finally, it costs $O(N)$ to determine the drxShortCycleTimer for all UEs. Thus, the stage two totally costs $O(F)+O(F)\cdot O(\frac{D_j}{\rho}^{M_j+1}\cdot M_j)+O(N)=O(F)\cdot O(\frac{D_j}{\rho}^{M_j+1}\cdot M_j)+O(N)$.

In the stage three, for each UE, it costs $O(\frac{T_N^S}{T_1^S})$ to choose the least crowded cycle among $\frac{T_N^S}{T_1^S}$ cycles and costs $O(\frac{T_N^S}{T_1^S})$ when updating the crowded degrees of all cycles. Since we have N UEs, it costs O($N\cdot\frac{T_N^S}{T_1^S}$) to determine the cycle where each UE starts its DRX operation. Finally, to proportionally redistribute N UEs' startoffsets over $\frac{T_N^S}{T_1^S}$ cycles costs O($N\cdot\frac{T_N^S}{T_1^S}$). Thus, the stage three totally costs $O(N\cdot\frac{T_N^S}{T_1^S})+O(N\cdot\frac{T_N^S}{T_1^S})=O(N\cdot\frac{T_N^S}{T_1^S})$.

Therefore, the time complexity of the TS scheme incurred by the three stages is $O(F+N\log N)+\Big(O(F)\cdot O(\frac{D_j}{\rho}^{M_j+1}\cdot M_j)+O(N)\Big)+O(N\cdot\frac{T_N^S}{T_1^S})$. We should note that the terms M_j, $\frac{D_j}{\rho}$, and $\frac{T_N^S}{T_1^S}$ are very small constant values as compared to N (i.e, M_j, $\frac{D_j}{\rho}$, and $\frac{T_N^S}{T_1^S}\ll N$) and the total number of flows F is usually constant times of N (i.e, $O(F)=O(N)$).

On the other hand, for the proposed packet scheduling scheme, it costs $O(F\cdot M)$ to find the stringent data among F flows, where each flow cumulates at most M packets in each subframe. Next, it totally costs $O(N)$ to allocate one RB to the UEs whose InactivityTimers will expire in the next subframe. Finally, it costs $O(N\log N)$ to sort N UEs according to the designate priority. Thus, the DRX-aware scheduling scheme totally costs $O(F\cdot M)+O(N)+O(N\log N)=O(N\log N)$ due to $M\ll N$ and $O(F)=O(N)$.

TABLE III
THE CHANNEL QUALITY IDENTIFIER (CQI) SUPPORTED IN LTE-A NETWORKS [20].

CQI index	modulation	code rate × 1024	bits per resource block
1	QPSK	78	12.79
2	QPSK	120	19.69
3	QPSK	193	31.67
4	QPSK	308	50.53
5	QPSK	449	73.67
6	QPSK	602	98.77
7	16QAM	378	124.03
8	16QAM	490	160.78
9	16QAM	616	202.13
10	64QAM	466	229.36
11	64QAM	567	279.07
12	64QAM	666	327.79
13	64QAM	772	379.97
14	64QAM	873	429.68
15	64QAM	948	466.59

TABLE IV
TRAFFIC ADOPTED IN THE SIMULATION [21]–[24].

Scenario	Applications	Flow Type	QoS Class Identifier	Traffic Bit-Rate	Packet Delay Budget	Packet Loss Rate
SN1 (General Traffic)	VoIP (G.711)	GBR	1	64 Kbps	100 ms	10^{-2}
	IPTV (H.264)	GBR	4	128 Kbps	300 ms	10^{-6}
	HTTP/FTP	non-GBR	6	169 Kbps	300 ms	10^{-6}
SN2 (IoT Traffic)	Voice Surveillance (AMR)	GBR	1	12.2 Kbps	100 ms	10^{-2}
	Video Surveillance (QVGA)	GBR	2	20 Kbps	150 ms	10^{-3}
	Other IoT Services (e.g. smart meter)	non-GBR	6	10 Kbps	300 ms	10^{-6}

V. PERFORMANCE EVALUATION

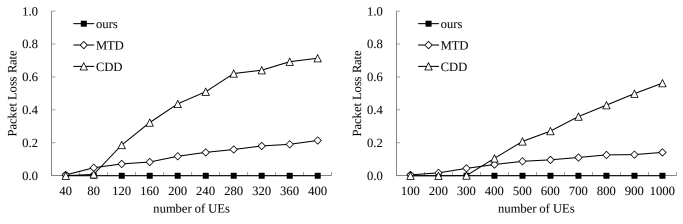
In the section, we present our simulation results to verify the effectiveness of the proposed scheme. We develop a simulator in JAVA language. The system parameters of the simulator are listed below. The frame duration is 10 ms. The channel bandwidth is 10 MHz. Thus, we have $\Omega = 100$ RBs in each subframe. Fifteen channel qualities are adopted in the simulation, as shown in Table III. Six types of applications are considered in the simulation. The QoS parameters of these applications are shown in Table IV. We also consider three types of UEs which adopt different applications. The first type of UEs adopts only one GBR flow. The second type of UEs adopts only one non-GBR flow. The third type of UEs adopts both one GBR and one non-GBR flows. The number of these three types of UEs are the same. The packet delay is modeled by the normal distribution with a mean of 0 and a standard deviation of 0.2 times of a flow's delay budget. The channel quality of each UE will vary over time. We generate the channel condition randomly in each subframe for each UE from Table III. Note that the unit to calculate the InactivityTimer for our scheme is $\rho = 5$ and the basic cycle is $T_1^S = (D_1^{min})/2$.

We compare our scheme against the *Counter-Driven DRX (CDD)* scheme [16] and the *Multiple-Threshold DRX (MTD)* scheme [17], which are the most relative schemes to the topic of this paper. The rationale of **CDD** scheme is to dynamically adjust each UE's cycle length to capture the UE's incoming traffic to improve energy efficiency and data receiving latency. The rationale of **MTD** scheme is to dynamically adjust each UE's InactivityTimer to accommodate different CQIs that the UE perceives to maintain energy efficiency while increasing the traffic rate satisfaction. Specifically, the **CDD** scheme adjusts each UE's cycle length based on two predefined counters and two thresholds. Thus, if the UE consecutively wakes up but does not receive the data delivered from the eNB, the first counter of the UE is increased. Otherwise, the UE resets its first counter. Once the UE's first counter reaches the predefined threshold $\hat{M}_{UE}$, the UE enlarges its cycle length to improve its sleep efficiency because the incoming traffic for the UE seems sparse. Contrarily, if the UE consecutively wakes up and receives the data from the eNB, the second counter of the UE is increased. Otherwise, the UE resets its second counter. Once the second counter reaches the predefined threshold $\hat{N}_{UE}$, the UE decreases its cycle length to reduce the packet receiving latency. Note that in the simulation we choose $\hat{M}_{UE} = 10$ and $\hat{N}_{UE} = 15$ for **CDD**, which is recommended in [16] for the best performance on energy saving and packet receiving latency. On the other hand, the **MTD** scheme fixes each UE's cycle length and adjusts their InactivityTimers based on the predefined SINR-thresholds for the *channel quality identifier (CQI)* that they perceive. If the UE's SINR is persistently smaller than the low SINR-threshold of the CQI that it perceives, the UE's InactivityTimer length will be increased because the UE has a lower CQI which needs more time to receive the incoming data. This can improve the UE's rate satisfaction. Contrarily, once the UE's SINR is persistently higher than the high SINR-threshold of the CQI that it perceives, the UE's InactivityTimer length will be decreased because the UE has a higher CQI which needs less time to receive the incoming data. This can improve the UE's sleep efficiency.

We consider two scenarios with different types of traffic: general traffic (SN1) and IoT traffic (SN2). The general traffic, which requires higher data rate, includes VoIP (G.711), IPTV (H.264), and HTTP/FTP services. The IoT traffic, which requires lower data rate, includes the applications for audio surveillance (AMR), video surveillance (QVGA), and smart metering. In the following results, the duration of each experiment is at least 6000 subframes.

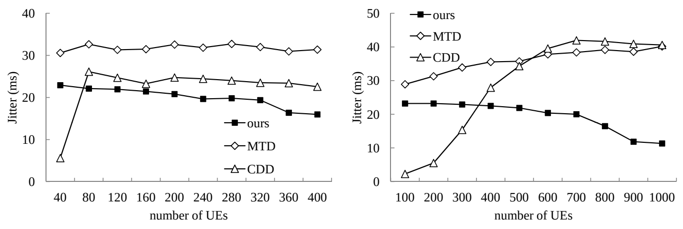
A. Packet Loss Rate

We first compare the average packet loss rate under different numbers of UEs. Fig. 2(a) and Fig. 2(b) show the results for SN1 and SN2, respectively. In both Fig. 2(a) and Fig. 2(b), we can see that the packet loss rate of most schemes increases when the number of UEs increases. This is because the network is getting saturated and it becomes difficult to serve all UEs' packets under the consideration of packet delay budgets. The **CDD** scheme incurs the highest packet loss rate because it only adjusts DRX cycles for UEs but neglects to tune their InactivityTimers. Once the UE's packet delay budget is used up at the middle of its cycle, the UE will fail to receive the packet. On the other hand, the **MTD** scheme has the lower packet loss rate because the UEs can adjust their InactivityTimers when they are under different channel conditions. It is important to note that our scheme outperforms other schemes. The packet loss rate of our scheme is even lower than 10^{-7} when the network is saturated (i.e., 400 UEs in SN1 and 1000 UEs in SN2). This is because our scheme can optimize UEs' InactivityTimers according to their target packet loss rates and cooperate with the DRX-aware scheduling to serve the urgent data first to fully utilize the resource. Thus, the packet loss rates of UEs can be exactly guaranteed.



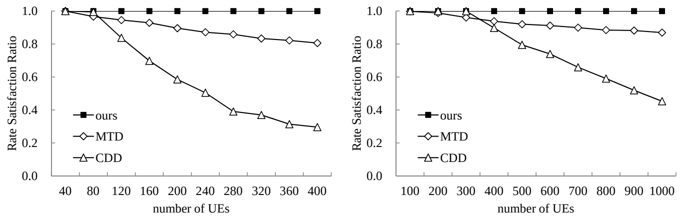
(a) SN1 (General Traffic) (b) SN2 (IoT Traffic)

Fig. 2. The impact of number of UEs on packet loss rate in scenarios SN1 and SN2.



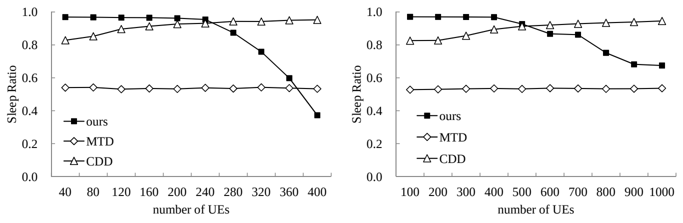
(a) SN1 (General Traffic) (b) SN2 (IoT Traffic)

Fig. 3. The impact of number of UEs on jitter in scenarios SN1 and SN2.



(a) SN1 (General Traffic) (b) SN2 (IoT Traffic)

Fig. 4. The impact of number of UEs on rate satisfaction ratio of UEs in scenarios SN1 and SN2.



(a) SN1 (General Traffic) (b) SN2 (IoT Traffic)

Fig. 5. The impact of number of UEs on sleep ratio in scenarios SN1 and SN2.

B. Jitter

We then measure jitter under different numbers of UEs, where jitter is defined as the standard deviation of packet delivery latency [25]. Fig. 3(a) and Fig. 3(b) show the results for SN1 and SN2, respectively. In Fig. 3(a) and Fig. 3(b), the **CDD** scheme and **MTD** scheme have higher jitter because **CDD** increases UE's cycle lengths if UEs do not receive their packets (due to dropped) and **MTD** assigns InactivityTimers for UEs disregarding the network traffic load, thus the buffered packets have to wait until the next on-duration comes, which results in longer delivery latency. We should note that our scheme outperforms other schemes in most cases. This is because our scheme can optimize the InactivityTimers by giving longer InactivityTimers for UEs when the network traffic load becomes heavy. Thus, the UEs can receive the target packets earlier, as compared to other schemes.

C. Rate Satisfaction Ratio

Next, we investigate the average rate satisfaction ratio of UEs, which is defined by the amount of *satisfied* rate (including GBR and non-GBR flows in the UE) over the total amount of admitted rates of UEs. When the satisfaction ratio is 1, it means that the scheme can successfully satisfy the required rate of UEs. Fig. 4(a) and Fig. 4(b) show the results for SN1 and SN2, respectively. As can be seen, all schemes have a rate satisfaction ratio of 1 when the number of UEs is less than 40 in SN1 and 200 in SN2, respectively, because the network is under non-saturated. The **CDD** scheme incurs the lowest rate satisfaction when the number of UEs is larger than 80 in SN1 and 300 in SN2, respectively, because this scheme loses lots of packets, especially for those time-aware surveillance data, which will not be retransmitted. On the other hand, the **MTD** scheme has better rate satisfaction because the **MTD** scheme assigns InactivityTimers for UEs based their CQI and thus the UEs can receive more packets even if they stay in a bad channel condition. We should note that our scheme performs the best. The satisfaction ratio of our scheme is still 1 when the number of UEs is 400 in SN1 and 1000 in SN2, respectively. This is because our scheme can optimize DRX parameters in terms of cycle length and InactivityTimer based on the network traffic load, and cooperates with a DRX-aware scheduler to serve the UEs with high channel quality to improve their rate satisfactions.

D. Average Sleep Ratio

We then evaluate the average sleep ratio under different numbers of UEs. Fig. 5(a) and Fig. 5(b) show the results for SN1 and SN2, respectively. As shown in Fig. 5(a) and Fig. 5(b), the sleep ratio of our scheme decreases when the number of UEs grows, because our scheme extends UEs' wake-up periods to guarantee their QoS when the network traffic load becomes heavy. Contrarily, the sleep ratio of the **CDD** scheme increases when number of UEs increases. This is because **CDD** neglects the UEs' QoS satisfaction. Note that the **MTD** scheme has a lowest sleep ratio because this scheme adjusts InactivityTimers of UEs only based on UEs' CQI which is independent with the network traffic load (i.e., the number of UEs).

E. Power Consumption

Consequently, we measure the average power consumption of all schemes under different numbers of UEs, where the UE's power consumption is modeled according to [26] and illustrated in Fig. 6. Fig. 7(a) and Fig. 7(b) show the results for SN1 and SN2, respectively. As can be seen, the power consumption of our scheme increases when the number of UEs increases. This is because UEs need more wake-up time to receive their data to guarantee their QoS when the

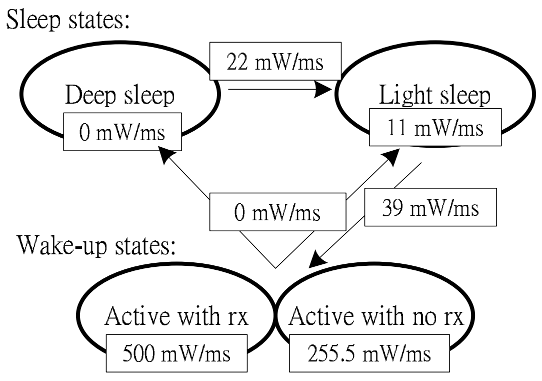
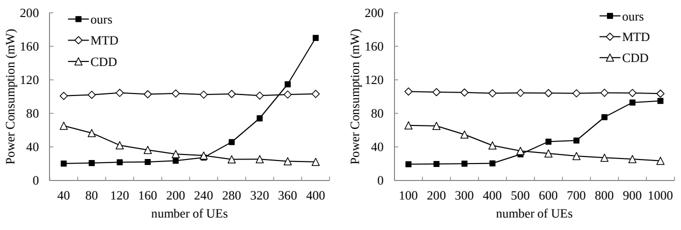


Fig. 6. The power consumption model for UEs [26].



(a) SN1 (General Traffic) (b) SN2 (IoT Traffic)

Fig. 7. The impact of number of UEs on power consumption in scenarios SN1 and SN2.

network resource is insufficient. On the other hand, the power consumption of the **CDD** scheme decreases when the number of UEs increases, because the UEs enlarge their cycle lengths without considering QoS. Finally, the power consumption of the **MTD** scheme is stable because this scheme adjusts InactivityTimers of UEs independent with the network traffic load.

F. Observations on Sleep Ratio of UEs with Multiple Flows

We now investigate the impact of the flows with different QoS parameters in the same UE on the sleep ratio performance. This is conducted especially for IoT scenarios because an IoT device may report different data via different flows according to different applications such as audio and video surveillance. In this observation, we consider five types of UEs and each UE adopts two flows with different QoS parameters listed in Table V. In particular, the type-1, type-2, and type-3 UEs have the flows with different packet delay budgets (i.e., 100 ms and 300 ms) but with the same packet loss rate (i.e., 10^{-2}). The type-4 and type-5 UEs have different packet loss rate (i.e., 10^{-2} and 10^{-6}) but with the same packet delay budgets (i.e., 100 ms). We conduct the observation through the proposed scheme to guarantee UEs' QoS. Fig. 8 shows the sleep ratio results of all types of UEs. We can see that type-3 UEs have the highest sleep ratio because their flows have looser QoS constraints (i.e., a larger packet delay budget and a higher required packet loss rate). Next, the sleep ratio of type-2 UEs is slightly lower than that of type-3 UEs because each type-2 UE has one flow with a shorter packet delay constraint. Then, the sleep ratio of type-1 UEs is lower than that of type-2 UEs because all of type-1 UEs' flows have lower packet delay budgets. We should note that the sleep ratio of type-2 UEs is close to that of type-1 UEs but not close to type-3 UEs', because the DRX mechanism enforces each UE to have only one cycle length and thus all flows of each UE are with a common cycle length limited by the strictest delay budget of the UE (i.e., 100 ms in this case). Thus, if the UE has a flow with a higher delay budget, it can not have a longer sleep period due to the limitation. Consequently, we can see that the sleep ratios of type-4 and type-5 UEs are lower than that of type-1 UEs because they require stricter packet loss rate (i.e., 10^{-6}). Note that the sleep ratio of type-4 UEs is close to that of type-5 UEs because the DRX mechanism enforces each UE to have a fixed InactivityTimer and thus all flows in the UE are with a common InactivityTimer limited by the strictest packet loss rate of the UE (i.e., 10^{-6} in this case). This strongly hurts the performance on energy saving because the UEs have to adopt the longest InactivityTimer of their flows even if it has completed to receive the packet of such flows.

TABLE V
TRAFFIC ADOPTED FOR OBSERVATIONS.

UE Type	Flow Type	Traffic Bit-Rate	Packet Delay Budget	Packet Loss Rate
1	GBR	12.2 Kbps	100 ms	10^{-2}
	GBR	12.2 Kbps	100 ms	10^{-2}
2	GBR	12.2 Kbps	100 ms	10^{-2}
	GBR	12.2 Kbps	300 ms	10^{-2}
3	GBR	12.2 Kbps	300 ms	10^{-2}
	GBR	12.2 Kbps	300 ms	10^{-2}
4	GBR	12.2 Kbps	100 ms	10^{-2}
	GBR	12.2 Kbps	100 ms	10^{-6}
5	GBR	12.2 Kbps	100 ms	10^{-6}
	GBR	12.2 Kbps	100 ms	10^{-6}

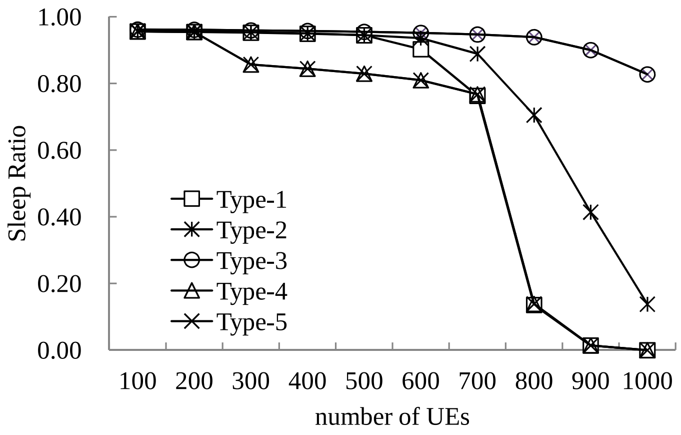


Fig. 8. The Impact of flows with different QoS in the UE on sleep ratio performance.

Based on above experiments and observations, we could summarize the limitations of the current DRX mechanism as follows. First, the current DRX supports only single cycle length and single InactivityTimer. Once the UE has the flows with different delay budgets, the UE has to wake up in each cycle even if the UE has received the packets completely in previous cycles. In addition, during the wake-up period, the UE has to wait for the timer expiring even if it has no data to receive. Second, the DRX mechanism enforces each UE to wake up and sleep only one period during a cycle. Thus, if the next data arrival of the UEs is far from the previous data arrival, the UE has to keep awake until the next data arrives. Above limitations would harm the performance on power saving. For future IoT applications, we may suggest the standard to support multiple cycle lengths and multiple wake-up/sleep patterns for each UE. Thus, the UE has the flows with different delay budgets that can be modeled by multiple sleep patterns which can best fit the traffic characteristic in the UE. So, the UE can wake up according to these patterns precisely. In addition, a special indicator is expected for the eNB to notify UEs to go to sleep immediately. This can reduce the idle wake-up period caused by waiting for InactivityTimer expiring. By above suggestions, we think it can make UEs' sleep behaviors more flexible and more efficient.

VI. Conclusions

In this paper, we addressed the DRX optimization problem which considers the QoS requirements of IoT applications in LTE-A networks. An efficient three-stage scheme and a DRX-aware packet scheduling method are proposed to tackle the problem. By balancing the impacts between QoS parameters and DRX configurations, simulation results have verified our schemes. It has shown that our schemes can fully guarantee UEs' QoS requirements in terms of packet loss rate, packet delay, and traffic bit-rate while saving considerable power consumption of UEs. For future work, we will investigate the performance bounds of the standard. In addition, the flexible DRX schemes will also be studied.

Acknowledgment

This research is co-sponsored by MoE ATU Plan, by NSC grants 100-2218-E-024-001-MY3, 98-2219-E-009-019, 98-2219-E-009-005, and 99-2218-E-009-005, by AS-102-TP-A06 of Academia Sinica, by ITRI, Taiwan, and by D-Link. This work was also conducted under the "AMI Enhancement Project" of III, which is subsidized by MoEA, ROC.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 4, pp. 2787–2805, 2010.

[2] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-Based Internet of Things: Discovery, query, selection, and on-demand provisioning of web services," *IEEE Transactions on Services Computing*, vol. 3, no. 3, pp. 223–235, 2010.

[3] 3GPP TS 23.401, "General Packet Radio Service (GPRS) enhancements for E-UTRAN access," v11.1.0 (Release 11), March 2012.

[4] 3GPP TS 36.213, "Policy and charging control architecture," v11.5.0 (Release 11), March 2012.

[5] 3GPP TS 36.413, "E-UTRAN; S1 Application Protocol (S1AP)," v10.3.0 (Release 10), March 2012.

[6] L. Zhou, H. Xu, H. Tian, Y. Gao, L. Du, and L. Chen, "Performance analysis of power saving mechanism with adjustable DRX cycles in 3GPP LTE," *IEEE Vehicular Technology Conference (VTC)*, pp. 1–5, 2008.

[7] C. S. Bontu and E. Illidge, "DRX mechanism for power saving in LTE," *IEEE Communications Magazine*, vol. 47, no. 6, pp. 48–55, 2009.

[8] J. Wigard, T. Kolding, L. Dalsgaard, and C. Coletti, "On the user performance of LTE UE power savings schemes with discontinuous reception in LTE," *IEEE International Conference on Communications Workshops (ICC)*, pp. 1–5, 2009.

[9] K. Aho, T. Henttonen, J. Puttonen, L. Dalsgaard, and T. Ristaniemi, "User equipment energy efficiency versus LTE network performance," *International Journal on Advances in Telecommunications*, vol. 3, no. 4, pp. 140–151, 2010.

[10] S. Jin and D. Qiao, "Numerical analysis of the power saving in 3GPP LTE Advanced wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1779–1785, 2012.

[11] G. S. Kim, Y. H. Je, and S. Kim, "An adjustable power management for optimal power saving in LTE terminal baseband modem," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 1847–1853, 2009.

[12] K.-C. Ting, H.-C. Wang, C.-C. Tseng, and F.-C. Kuo, "Energy-efficient DRX scheduling for QoS traffic in LTE networks," *IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, pp. 213–218, 2011.

[13] B. Huang, H. Tian, L. Chen, and J. Zhu, "DRX-aware scheduling method for delay-sensitive traffic," *IEEE Communications Letters*, vol. 14, no. 12, pp. 1113–1115, 2010.

[14] R. Wang, J. S. Thompson, and H. Haas, "A novel time-domain sleep mode design for energy-efficient LTE," *International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pp. 1–4, 2010.

[15] H. Xu, H. Tian, B. Huang, and P. Zhang, "An improved dynamic user equipment power saving mechanism for LTE system and performance analysis," *Science China Information Sciences*, vol. 53, no. 10, pp. 2075–2086, 2010.

[16] E. Liu, J. Zhang, and W. Ren, "Adaptive DRX scheme for beyond 3G mobile handsets," *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–5, 2011.

[17] S. Gao, H. Tian, J. Zhu, and L. Chen, "A more power-efficient adaptive discontinuous reception mechanism in LTE," *IEEE Vehicular Technology Conference (VTC)*, pp. 1–5, 2011.

[18] M. Alasti, B. Neekzad, J. Hui, and R. Vannithamby, "Quality of service in WiMAX and LTE networks," *IEEE Communications Magazine*, vol. 48, no. 5, pp. 104–111, 2010.

[19] 3GPP TS 36.321, "E-UTRA; Medium Access Control (MAC) protocol specification," v10.5.0 (Release 10), March 2012.

[20] 3GPP TS 36.213, "E-UTRA; Physical layer procedures," v10.6.0 (Release 10), June 2012.

[21] R. Mehmooda, R. Alturkia, and S. Zeadally, "Multimedia applications over metropolitan area networks (MANs)," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1518–1529, 2010.

[22] G. Gualdi, A. Prati, and R. Cucchiara, "Performance comparison of control-less scheduling policies for VoIP in LTE UL," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2497–2501, 2008.

[23] G. Gualdi, A. Prati, R. Cucchiara, "Video streaming for mobile video surveillance," *IEEE Transactions on Multimedia*, pp. 1142–1154, 2008.

[24] S. Yang, X. Wen, W. Zheng, and Z. Lu, "Convergence architecture of Internet of Things and 3GPP LTE-A network based on IMS," *IEEE Global Mobile Congress (GMC)*, pp. 1–7, 2011.

[25] Y.-C. Tseng, J.-J. Chen, and Y.-C. Yang, "Managing power saving classes in IEEE 802.16 Wireless MANs: A fold-and-demultiplex method," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1237–1247, 2011.

[26] 3GPP TSG RAN WG2 LTE contribution, R2-071285, "DRX parameters in LTE," March 2007.

Jia-Ming Liang received his B.S. and M.S. degrees in Computer Science and Engineering from National Taiwan Ocean University and National Sun Yat-Sen University in 2004 and 2006, respectively. He obtained his Ph.D. degree in Computer Science from the National Chiao-Tung University in 2011.
Currently, he is a postdoctoral research fellow in the National Chiao-Tung University. His research interests include resource management in wireless networks, cross-layer design, and WirelessMAN technologies.

Jen-Jee Chen received his B.S. and M.S. degrees in Computer Science and Information Engineering in 2001 and 2003, respectively, from National Chiao Tung University, Hsinchu, Taiwan, and his Ph.D. degree in Computer Science in 2009 from National Chiao Tung University, Hsinchu, Taiwan. He was Visiting Scholar at the University of Illinois at Urbana Champaign during the 2007-2008 academic year and postdoctoral research fellow at the Department of Electrical Engineering, National Chiao Tung University, Taiwan, during 2010-2011. Since 2011, he joined the Department of Electrical Engineering, National University of Tainan, Taiwan, where he is currently an assistant professor. His research interests include wireless communications and networks, mobile computing, cross-layer design, and Internet of Things. Dr. Chen is a member of the IEEE and the Phi Tau Phi Society.

Hung-Hsin Cheng received M.S. degree in Computer Science from National Chiao-Tung University in 2013. His research interests include wireless communication and mobile computing. Currently, he is a software engineer.

Yu-Chee Tseng got his Ph.D. in Computer and Information Science from the Ohio State University in January of 1994. He was/is Chairman (2005-2009), Chair Professor (2011-present), and Dean (2011-present), Department of Computer Science, National Chiao-Tung University, Taiwan.

Dr. Tseng is a Y. Z. Hsu Scientific Chair Professor. Dr. Tseng received Outstanding Research Award (National Science Council, 2001, 2003, and 2009), Best Paper Award (Int’l Conf. on Parallel Processing, 2003), Elite I. T. Award (2004), and Distinguished Alumnus Award (Ohio State University, 2005), and Y. Z. Hsu Scientific Paper Award (2009). His research interests include mobile computing, wireless communication, and sensor networks. Dr. Tseng is an IEEE Fellow. He serves/served on the editorial boards of IEEE Trans. on Vehicular Technology (2005-2009), IEEE Trans. on Mobile Computing (2006-2011), and IEEE Trans. on Parallel and Distributed Systems (2008-present).