

Received December 2, 2017, accepted January 5, 2018, date of publication January 9, 2018, date of current version March 15, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2791466

Optimistic DRX for Machine-Type Communications in LTE-A Network

HUI-LING CHANG, (Student Member, IEEE), AND MENG-HSUN TSAI^{ID}, (Member, IEEE)

Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 70101, Taiwan

Corresponding author: Meng-Hsun Tsai (e-mail: tsaimh@csie.ncku.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant 106-2221-E-006-007- and Grant 105-2221-E-006-186-. This paper is an extended version of our conference paper in Proceedings of the 2016 IEEE International Conference on Communications [1].

ABSTRACT In long-term evolution-advanced (LTE-A) network, the third-generation partnership project has proposed machine-type communications (MTC) as a new paradigm where devices transfer data among themselves with limited human interaction. For devices in MTC, power saving has become an important issue because many devices are powered by battery. In LTE-A network, the standard power saving mechanism, called discontinuous reception (DRX), is designed for normal mobile users, not for MTC devices. In this paper, we propose an optimistic DRX (ODRX) mechanism to be suitable for MTC devices. ODRX considers the radio resource control connection release and re-establishment to save more power. We also introduce the optimistic flag to allow longer sleep periods. Analytical and simulation models are proposed to investigate the performance of ODRX, and ODRX is then compared with the standard DRX and dynamic DRX (DDRX) through simulation experiments. The results show that ODRX outperforms standard DRX and DDRX by gaining significant extra power saving with little extra wake up latency. We also propose guidelines to configure ODRX parameters.

INDEX TERMS Discontinuous reception (DRX), long term evolution-advanced (LTE-A), machine-type communications (MTC), power saving.

I. INTRODUCTION

Long term evolution-advanced (LTE-A), proposed by the third generation partnership project (3GPP), is the most promising fourth generation mobile network. In LTE-A network, the 3GPP has proposed machine-type communications (MTC) as a new paradigm where devices transfer data among themselves with limited human interaction. MTC is actually the same concept of Internet of things (IoT) extensively known by the public. Nowadays there are many electronic products which will bring human life much more convenient if they are combined with MTC. This is why MTC is widely discussed and applied in many aspects, such as monitoring system, industrial automation, and smart home. At the same time, MTC accompanies new technical issues.

For devices in MTC, power saving has already become an important issue. Devices basically use wireless communications networks. In most situations, they keep silent until there are data delivered from/to them. Additionally, small data transmissions is a specified feature in MTC [2], which implies the devices only spend little time to receive the data. However, if they are active all the time, which means they retain the

radio resource and keep the connection with networks so that data can be sent/received immediately, most of the power is wasted. Actually, many devices are powered by battery so how to prolong the battery life is pretty important [3]–[6]. Moreover, the devices in some applications are out of reach after setting up to work, such as water quality monitors. Therefore, the battery of those devices is hard to be replaced with new one. From another point of view, the size of devices is proportional to their power consumption [4]. Generally speaking, the more energy the devices consume, the bigger the battery is designed. Nonetheless, the devices in many applications are designed as small as possible. Additionally, the low cost is also another important consideration [5]. As a result, the electricity must be used more efficiently to extend the battery life. Last but not least, the total amount of energy consumption is getting higher and higher because the world tends to have more and more Internet-connected devices. (The current projection of the device number by 2020 is about 30 billion [7].) How to save more energy absolutely becomes a significant concern in device design.

In the meantime, the 3GPP proposes a power-saving mechanism, called discontinuous reception (DRX) [8], to allow user equipments (UEs) to receive data only at specified time slots and turn off the radio module at other time slots. When DRX mode is turned on in a UE, inactivity timer is started. Note that DRX mode can be applied in not only RRC connected mode [8] but also RRC idle mode [9]. If some data arrive at the UE, the timer is re-started. Once the timer is expired (i.e., no data arrival during countdown of the timer), the UE enters a DRX cycle. In the DRX cycle, the UE monitors physical downlink control channel (PDCCH) only in the period called *on duration*. If data arrival is in on duration, the UE receives the data, enters active mode, and starts inactivity timer. Otherwise, the data need to wait until the on duration of next DRX cycle.

Unfortunately, current DRX mechanism is designed for normal usage of mobile users, not for MTC. For a mobile user, latency is usually more important than power saving. However, in many cases, latency is not that important for an MTC application. In this paper, we propose an optimistic DRX (ODRX) mechanism to allow more sleep periods for MTC devices. To allow more sleep periods, ODRX considers to release the radio resource control (RRC) connection and re-establish the connection when the MTC device is paged. In other words, ODRX allows latency incurred from re-establishment of RRC connection as long as more power could be saved. ODRX also follows the new policy to decide what state the devices will enter when the inactivity timer is expired. To investigate performance of ODRX, we propose analytical and simulation models for ODRX, and we compare with the standard DRX and dynamic DRX (DDRX) through simulation experiments.

This paper is an extended version of our conference paper [1]. The differences between these two versions are identified as follows:

- 1) Multiple numbers of short cycles and long cycles are adopted by introducing two counters N and M with thresholds n and m , respectively. Precisely, Chang *et al.* [1] propose a simplified scheme where $n = 1$ and $m = 1$. The details are described in Section III.
- 2) Derivation of analytical model with n and m , described in Section IV, is much more complicated than that described in [1].
- 3) Effects of n and m on power saving factor and wake up latency are studied in Section V. Based on the results, we suggest a policy to choose the n and m values.
- 4) To compare with existing adaptive methods, we have added experiments on Dynamic DRX in Section V, which is not studied in [1].
- 5) To obtain more insights on the cycle length, we have added experiments on the effects of long cycle length T_L and short cycle length T_S , as shown in Fig.7.

This paper is organized as follows. Related works are described in Section II. Section III introduces ODRX. In Section IV, we present an analytical model to investigate

the performance of ODRX. The analytical model is used to validate against the simulation model. Based on the validated simulation model, we investigate performance of ODRX in Section V. Finally, we conclude this paper in Section VI.

II. RELATED WORK

In this section, we first describe existing works on DRX for normal usage. Then we discuss several works on DRX for MTC applications.

The two key output measures in DRX are power saving factor and wake up latency. The trade-offs between these two measures have been widely discussed [10]–[15]. Jha *et al.* [10] and Koc *et al.* [11], [12] try to optimize the static DRX settings to obtain more power saving. In [13] and [14], dynamic DRX configuration is adopted to maximize the power saving factor when the mean packet waiting time is given as a constraint. Hsieh *et al.* [15] claim that analytical models in existing works are not accurate due to simplified assumption, and then propose an analytical model which is more realistic and better conforms to the 3GPP specifications. Results in [15] have shown that the simplified analytical models in existing works are accurate enough when packet arrival rate is small, but not accurate when packet arrival rate is large.

Under different traffic models, the DRX configurations are investigated and analyzed in [16]–[19]. Web traffic has been first studied in [16]. Then multimedia traffic is studied in [17], and self-similar traffic is studied in [18]. Moreover, multicast transmissions which demand to guarantee the quality of service (QoS) are investigated in [19].

The above works focus on normal usage of mobile users, and are not able to be used directly for MTC applications.

Since packet sizes are assumed to be much smaller than usual and packet interval time is assumed to be much longer in MTC applications [2], DRX mechanism is re-investigated for MTC applications in [20] and [21]. Through semi-Markov chain, the effects of DRX parameters on power saving factor and wake-up latency are studied. The results in [20] indicate that proper settings of DRX parameters are important in MTC applications. The difference between the two papers is that Wu *et al.* [21] consider the DRX in RRC idle mode but Zhou *et al.* [20] do not. Wu *et al.* [21] claim that RRC idle mode plays a major role in power saving rather than RRC active mode. However, the effect of RRC idle mode is not quantitatively analyzed in [21].

In [22], QoS requirements on traffic bit-rate, packet delay, and packet loss rate are taken into account. A QoS-aware DRX mechanism is proposed to achieve those requirements by sacrificing some power saving. However, power saving is much more important than QoS requirements in many MTC applications as long as MTC devices are powered by battery.

Because of the characteristics of the MTC, the DRX mechanism can not be directly applied on the MTC devices. In most MTC applications, the power saving issue is more important than the latency issue. In this paper, we propose an optimistic DRX mechanism to save more power by designing

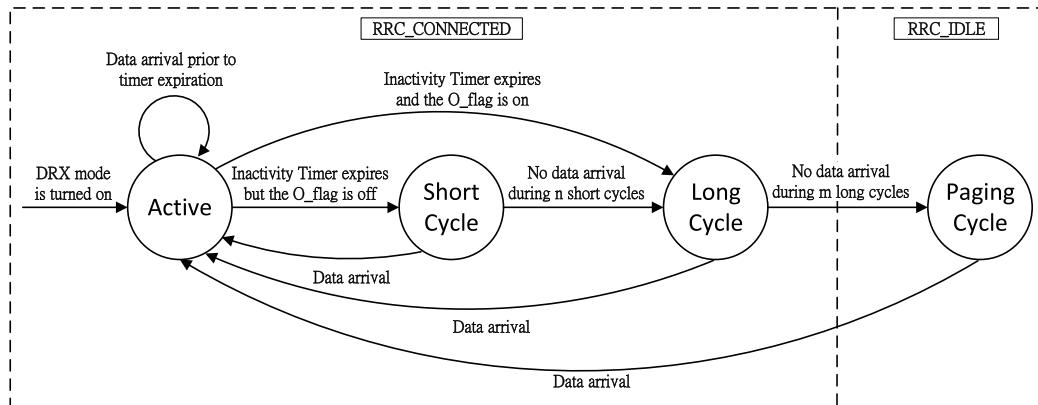


FIGURE 1. Finite state machine of ODRX.

different rule, and considering both RRC_CONNECTED and RRC_IDLE modes for MTC applications. Then we compare the proposed mechanism with existing works in terms of power saving factor and wake up latency.

III. THE OPERATION OF PROPOSED ODRX

This section introduces the operation of optimistic DRX (ODRX) for MTC applications. Fig. 1 illustrates the finite state machine of the ODRX mechanism exercised in an MTC device. The states are elaborated as follows.

A. ACTIVE

At the beginning of the DRX mode, the MTC device is at **Active** state, and the *inactivity timer* (which is denoted as T_0) is started. There is a flag, called *optimistic flag* (O_{flag}), set to Off. When the device is at **Active** state, it keeps monitoring PDCCH continuously to see whether there are data or not. If the device receives data from network before the inactivity timer expires, the device stays at **Active** state, restarts the timer, and sets the O_{flag} to Off. After the inactivity timer expires, the O_{flag} is checked immediately. If the flag is Off, the device enters **Short Cycle** (The period of a short cycle is denoted as T_S) and starts *DRX Short Cycle Counter* (denoted as N). Otherwise, the device enters **Long Cycle** (with period T_L) and starts *DRX Long Cycle Counter* (denoted as M).

B. SHORT CYCLE

The device monitors PDCCH only in *On Duration* (with period T_{ON}) period and then goes to sleep for a period which is called *Opportunity for DRX* (according to above notation, this period is denoted as $T_S - T_{ON}$). If the device observes data in On Duration, the device will return to **Active** state after reception. When the DRX Short Cycle Counter expires (i.e., $N = n$, where n is the short cycle counter threshold), the device enters **Long Cycle** and starts DRX Long Cycle Counter. That is if there are no received data in n short cycles, the device enters the next state to save more power. On the other hand, if some data arrive in sleep period, the device receives the data at the beginning of the On Duration of the next short cycle.

C. LONG CYCLE

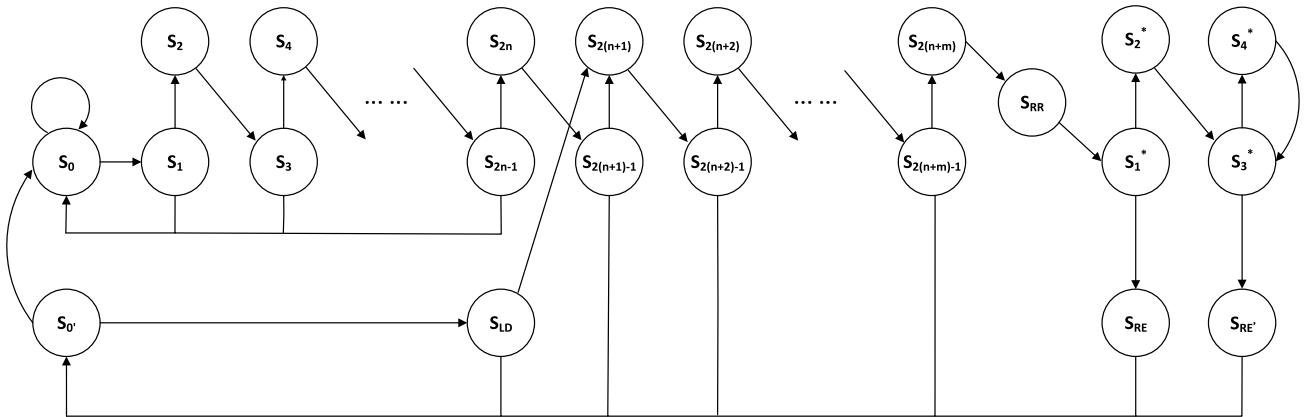
The behavior of the device at **Long Cycle** is basically the same as that at **Short Cycle** except that Opportunity for DRX in **Long Cycle** is much longer. In addition, if the device returns to **Active** state because of data arrival in **Long Cycle**, the O_{flag} is set to On. Once the DRX Long Cycle Counter expires (i.e., $M = m$, where m is the long cycle counter threshold), the device releases the radio resource and enters **Paging Cycle** (with period T_P).

D. PAGING CYCLE

The device monitors PDCCH for a single subframe (denoted as T_{sub}) in each paging cycle. The eNB may page the device at the subframe if some data arrive at the eNB before the subframe. If the device is paged at the subframe, the device needs to reestablish the connection and then returns to **Active** state. The O_{flag} is also set to On.

The main “optimistic” concepts are two parts. One is the adoption of **Paging Cycle**. Another is the usage of the O_{flag} . If there are no data delivered to/from the device for a long time, keeping connection with the network wastes not only the radio resource but also the device power. However, in **Paging Cycle**, the resource (i.e., the RRC connection) is not occupied and the device can save more power without RRC connection. In addition, as long as the device comes back to **Active** state from **Long Cycle** or **Paging Cycle**, next time when the inactivity timer expires, the device enters **Long Cycle** directly instead of **Short Cycle**. Since many MTC applications incur data transmission occasionally, skipping **Short Cycle** is more battery-efficient.

If the device actually needs to receive data frequently after it just returns from **Long Cycle** or **Paging Cycle**, it can still stay at **Active** state or **Short Cycle** by the operation of the O_{flag} . It is assumed that data transmission is intensive because of continuous data arrival at the eNB and the device just comes back to **Active** state from **Long Cycle** or **Paging Cycle**. The O_{flag} is on. Due to the consecutive data packet, data will arrive before the inactivity timer expires, so the O_{flag} will be set to off. Then the device enters **Short Cycle**

**FIGURE 2.** Semi-Markov chain for ODRX.

instead of **Long Cycle** when the inactivity timer expires for the next time.

IV. ANALYTICAL MODEL

This section proposes an analytical model for ODRX by the semi-Markov chain in Fig. 2. We assume that data arrivals form a Poisson process with rate λ . In other words, the inter-arrival time T_d of data follows exponential distribution with mean $1/\lambda$. The notation used in analytical model is shown in Table 1.

TABLE 1. Notation used in analytical model.

α	Power saving factor
δ	Wake up latency
π_i	Stationary probability at S_i
H_i	State holding time of S_i
T	The average system time
N	The DRX Short Cycle Counter
M	The DRX Long Cycle Counter
t_s	The sleep ratio in short cycles
t_l	The sleep ratio in long cycles
t_p	The sleep ratio in paging cycles
T_d	Inter-arrival time of data
$p_{i,j}$	Transition probability from S_i to S_j
T_0	The Inactivity Timer
T_{ON}	Length of On Duration
T_{sub}	Length of a subframe (1 ms)
T_S	Length of a short cycle
T_L	Length of a long cycle
T_P	Length of a paging cycle
T_{RR}	Latency of RRC connection release
T_{RE}	Latency of RRC connection establishment

In the semi-Markov chain, states S_0 and S_0' denote **Active** with $O_flag=0$ and $O_flag=1$, respectively. The parameters n and m are the thresholds of the short cycle counter N and the long cycle counter M , respectively. States S_{2i-1} for $i \in [1, n]$ denote the On Duration in **Short Cycle**. States S_{2i} for $i \in [1, n]$ denote the Opportunity for DRX in **Short Cycle**. States $S_{2(n+i)-1}$ for $i \in [1, m]$ denote the On Duration in **Long Cycle**. States $S_{2(n+i)}$ for $i \in [1, m]$ denote the Opportunity for DRX in **Long Cycle**. States S_1^* , S_2^* , S_3^* and S_4^* denote

Paging Cycle. In this cycle, MTC device monitors PDCCH at states S_1^* and S_3^* , and sleeps at states S_2^* and S_4^* . State S_{RR} denotes that the device is releasing RRC connection. The time spent in S_{RR} is denoted as T_{RR} . States S_{RE} and $S_{RE'}$ denote that the device is establishing an RRC connection. The times spent in S_{RE} and $S_{RE'}$ are both denoted as T_{RE} . Note that state holding times of states S_{RR} , S_{RE} and $S_{RE'}$ are used to estimate extra latency for accuracy. Finally, SLD is the state for On Duration in the first long cycle. In order to facilitate analysis, we separate SLD from $S_{2(n+1)-1}$. If the Inactivity Timer expires when the device is in S_0' , the device enters **Long Cycle** directly (SLD).

We first explain our output measures in subsection IV-A. Then we derive the stationary probabilities π_i in subsection IV-B and the state holding times H_i in subsection IV-C. Based on the derived stationary probabilities and state holding times, we derive our output measures in subsection IV-D.

A. THE OUTPUT MEASURES

We analyze the performance of ODRX by considering the power saving factor (denoted as α) and the wake up latency (denoted as δ). Adopting DRX in MTC aims at saving more power. The power saving factor is used to estimate the proportion of sleep time in the entire system. In Fig. 2, the device sleeps at S_{2i} for $i \in [1, n]$, $S_{2(n+i)}$ for $i \in [1, m]$, S_1^* , S_2^* , and S_4^* .

The wake up latency is another important indicator to judge the performance, which is used to estimate the average delay of data. When the device is sleeping, the latency of data (buffered at eNodeB) accumulates. In the proposed mechanism, we try to sacrifice little latency in exchange for more power saving.

From Fig. 2, the whole system time, T , is expressed as

$$T = \sum_{i=0}^{2(n+m)} \pi_i H_i + \sum_{i=1}^4 \pi_i^* H_i^* + \pi_0 H_0' + \pi_{LD} H_{LD} + \pi_{RR} H_{RR} + \pi_{RE} H_{RE} + \pi_{RE'} H_{RE'} \quad (1)$$

where π_i (or π_i^*) is the stationary probability of S_i (or S_i^*) and H_i (or H_i^*) is the state holding time of S_i (or S_i^*).

The average sleep period in short cycles, t_s , is

$$t_s = \sum_{i=1}^n \pi_{2i} H_{2i}. \quad (2)$$

Similarly, the average sleep period in long cycles, t_l , is

$$t_l = \sum_{i=1}^m \pi_{2(n+i)} H_{2(n+i)}. \quad (3)$$

The average sleep period in paging cycles, t_p , is

$$t_p = \pi_2^* H_2^* + \pi_4^* H_4^*. \quad (4)$$

According to above description, the power saving factor, α , is defined as

$$\alpha = \frac{t_s + t_l + t_p}{T}. \quad (5)$$

By substituting (1), (2), (3) and (4) into (5), we obtain the power saving factor α .

The data arrivals in each cycle form a Poisson process since the inter-data intervals follow an exponential distribution. In average, the data arrive over a time interval at the center, so the latency caused by the sleep periods of short cycle and long cycle (excluding the last long cycle) are $(T_S - T_{ON})/2$ and $(T_L - T_{ON})/2$, respectively. Specifically, the latency caused by the last long cycle is $(T_L - T_{ON})/2 + T_{RR} + T_{sub} + T_{RE}$. In addition, the latency caused by S_{RR} , S_1^* (S_3^*), S_2^* (S_4^*), and S_{RE} ($S_{RE'}$) are $T_{RR}/2 + T_{sub} + T_{RE}$, $T_{sub}/2 + T_{RE}$, $(T_P - T_{sub})/2 + T_{sub} + T_{RE}$, and $T_{RE}/2$, respectively. Finally, the wake up latency, δ , is expressed as

$$\begin{aligned} \delta = \frac{1}{T} & \left[\frac{T_S - T_{ON}}{2} t_s + \frac{T_L - T_{ON}}{2} t_l \right. \\ & + (T_{RR} + T_{sub} + T_{RE}) \pi_{2(n+m)} H_{2(n+m)} \\ & + \left(\frac{T_{RR}}{2} + T_{sub} + T_{RE} \right) \pi_{RR} H_{RR} \\ & + \left(\frac{T_P - T_{sub}}{2} + T_{sub} + T_{RE} \right) t_p \\ & + \left(\frac{T_{sub}}{2} + T_{RE} \right) (\pi_1^* H_1^* + \pi_3^* H_3^*) \\ & \left. + \frac{T_{RE}}{2} (\pi_{RE} H_{RE} + \pi_{RE'} H_{RE'}) \right]. \end{aligned} \quad (6)$$

B. STATIONARY PROBABILITIES

The transition probability from S_i to S_j is denoted as $p_{i,j}$. The stationary probability of state i , π_i , can be calculated based on the “flow rate” concept [23]. Based on the concept, we can get the stationary probabilities of states in Fig. 2:

$$\begin{aligned} \pi_i &= \pi_0 \prod_{j=1}^i p_{j-1,j}, \quad i \in [1, 2n+1], \\ \pi_{2(n+1)} &= \pi_0 \prod_{j=1}^{2(n+1)} p_{j-1,j} + \pi_{LD} p_{LD,2(n+1)}, \\ \pi_i &= \pi_{2(n+1)} \prod_{j=2n+3}^i p_{j-1,j}, \quad i \in [2n+3, 2(n+m)], \end{aligned}$$

$$\pi_{RR} = \pi_{2(n+m)} p_{2(n+m),RR},$$

$$\pi_1^* = \pi_{RR} p_{RR,1^*},$$

$$\pi_2^* = \pi_1^* p_{1^*,2^*},$$

$$\pi_3^* = \pi_2^* p_{2^*,3^*} + \pi_4^* p_{4^*,3^*},$$

$$\pi_4^* = \pi_3^* p_{3^*,4^*},$$

$$\pi_{RE} = \pi_1^* p_{1^*,RE},$$

$$\pi_{RE'} = \pi_3^* p_{3^*,RE'},$$

$$\begin{aligned} \pi_{0'} &= \pi_{LD} p_{LD,0'} + \sum_{i=1}^m \pi_{2(n+i)-1} p_{2(n+i)-1,0'} \\ &+ \pi_{RE} p_{RE,0'} + \pi_{RE'} p_{RE',0'}, \end{aligned}$$

$$\pi_{LD} = \pi_{0'} p_{0',LD}. \quad (7)$$

The inter-arrival time T_d of data follows exponential distribution with mean $1/\lambda$. If there is no data arrival before T_0 expires when the device is at S_0 , transition from S_0 to S_1 occurs. The state transition probability $p_{0,1}$ is expressed as

$$p_{0,1} = \Pr[T_d > T_0] = e^{-\lambda T_0}. \quad (8)$$

Similarly,

$$p_{1,2} = \Pr[T_d > T_{ON}] = e^{-\lambda T_{ON}},$$

$$p_{2i-1,2i} = \Pr[T_d > T_S] = e^{-\lambda T_S}, \quad i \in [2, n+1],$$

$$p_{2i-1,2i} = \Pr[T_d > T_L] = e^{-\lambda T_L}, \quad i \in [n+2, n+m],$$

$$\begin{aligned} p_{1^*,2^*} &= \Pr[T_d > (T_L - T_{ON} + T_{RR} + T_{sub})] \\ &= e^{-\lambda(T_L - T_{ON} + T_{RR} + T_{sub})}, \end{aligned}$$

$$p_{3^*,4^*} = \Pr[T_d > T_P] = e^{-\lambda T_P},$$

$$p_{0',LD} = \Pr[T_d > T_0] = e^{-\lambda T_0},$$

$$p_{LD,2(n+1)} = \Pr[T_d > T_{ON}] = e^{-\lambda T_{ON}}. \quad (9)$$

On the other hand, if data arrive before T_0 expires when the device is at S_0 , the device stays at S_0 . The state transition probability $p_{0,0}$ is expressed as

$$p_{0,0} = \Pr[T_d < T_0] = 1 - e^{-\lambda T_0}. \quad (10)$$

Similarly,

$$\begin{aligned} p_{1^*,RE} &= \Pr[T_d < (T_L - T_{ON} + T_{RR} + T_{sub})] \\ &= 1 - e^{-\lambda(T_L - T_{ON} + T_{RR} + T_{sub})}, \end{aligned}$$

$$p_{3^*,RE'} = \Pr[T_d < T_P] = 1 - e^{-\lambda T_P},$$

$$p_{1,0} = \Pr[T_d < T_{ON}] = 1 - e^{-\lambda T_{ON}},$$

$$p_{2i-1,0} = \Pr[T_d < T_S] = 1 - e^{-\lambda T_S}, \quad i \in [2, n],$$

$$p_{2n+1,0'} = \Pr[T_d < T_S] = 1 - e^{-\lambda T_S},$$

$$p_{2i-1,0'} = \Pr[T_d < T_L] = 1 - e^{-\lambda T_L}, \quad i \in [n+2, n+m],$$

$$p_{0',0} = \Pr[T_d < T_0] = 1 - e^{-\lambda T_0},$$

$$p_{LD,0'} = \Pr[T_d < T_{ON}] = 1 - e^{-\lambda T_{ON}}. \quad (11)$$

The device enters S_3 certainly when it is at S_2 , so $p_{2,3} = 1$. Similarly, $p_{2i,2i+1} = 1$, $i \in [2, n+m-1]$ and $p_{2(n+m),RR} = p_{RR,1^*} = p_{2^*,3^*} = p_{4^*,3^*} = p_{RE,0'} = p_{RE',0'} = 1$.

Finally, from (7)-(11), we have

$$\begin{aligned}
 \pi_1 &= e^{-\lambda T_0} \pi_0, \\
 \pi_{2i} &= \pi_{2i+1} = e^{-\lambda(T_0 + T_{\text{ON}} + (i-1)T_S)} \pi_0, \quad i \in [1, n], \\
 \pi_{2n+3} &= \pi_{2(n+1)}, \\
 \pi_{2(n+i-1)} &= \pi_{2(n+i)-1} = e^{-\lambda(i-2)T_L} \pi_{2(n+1)}, \quad i \in [3, m], \\
 \pi_{2(n+m)} &= e^{-\lambda(m-1)T_L} \pi_{2(n+1)}, \\
 \pi_{\text{RR}} &= e^{-\lambda(m-1)T_L} \pi_{2(n+1)}, \\
 \pi_1^* &= e^{-\lambda(m-1)T_L} \pi_{2(n+1)}, \\
 \pi_2^* &= e^{-\lambda(mT_L - T_{\text{ON}} + T_{\text{RR}} + T_{\text{sub}})} \pi_{2(n+1)}, \\
 \pi_3^* &= \frac{e^{-\lambda(mT_L - T_{\text{ON}} + T_{\text{RR}} + T_{\text{sub}})}}{1 - e^{-\lambda T_P}} \pi_{2(n+1)}, \\
 \pi_4^* &= \frac{e^{-\lambda(mT_L - T_{\text{ON}} + T_{\text{RR}} + T_{\text{sub}} + T_P)}}{1 - e^{-\lambda T_P}} \pi_{2(n+1)}, \\
 \pi_{\text{RE}} &= \left[1 - e^{-\lambda(T_L - T_{\text{ON}} + T_{\text{RR}} + T_{\text{sub}})} \right] \\
 &\quad \times e^{-\lambda(m-1)T_L} \pi_{2(n+1)}, \\
 \pi_{\text{RE}'} &= e^{-\lambda(mT_L - T_{\text{ON}} + T_{\text{RR}} + T_{\text{sub}})} \pi_{2(n+1)}, \\
 \pi_{0'} &= \frac{\pi_{2(n+1)} + (1 - e^{-\lambda T_S}) e^{-\lambda(T_0 + T_{\text{ON}} + (n-1)T_S)} \pi_0}{1 - e^{-\lambda T_0} + e^{-\lambda(T_0 + T_{\text{ON}})}}, \\
 \pi_{\text{LD}} &= \frac{e^{-\lambda T_0} \pi_{2(n+1)}}{1 - e^{-\lambda T_0} + e^{-\lambda(T_0 + T_{\text{ON}})}} \\
 &\quad + \frac{(1 - e^{-\lambda T_S}) e^{-\lambda(2T_0 + T_{\text{ON}} + (n-1)T_S)} \pi_0}{1 - e^{-\lambda T_0} + e^{-\lambda(T_0 + T_{\text{ON}})}}. \quad (12)
 \end{aligned}$$

C. STATE HOLDING TIME

When the device is at S_0 , there are two cases for data arrival: (i) no data arrive before the expiration of T_0 ; (ii) data arrive before the expiration of T_0 . In case (ii), assume that data arrive at the i th subframe of T_0 with probability p_i , and the holding time is T_i . Then

$$H_0 = p_{0,1} T_0 + \sum_{i=1}^{T_0} p_i T_i, \quad (13)$$

where

$$p_i = \Pr[i-1 < T_d < i] = e^{-\lambda(i-1)} - e^{-\lambda i}, \quad i \in [1, T_0] \quad (14)$$

and

$$T_i = H_0 + i. \quad (15)$$

From (13)-(15), we have

$$H_0 = \frac{1 - e^{-\lambda T_0}}{e^{-\lambda T_0}(1 - e^{-\lambda})}. \quad (16)$$

The calculation of $H_{0'}$ is similar to that of H_0 with the only difference that $T_i = i$. We can get

$$H_{0'} = \frac{1 - e^{-\lambda T_0}}{1 - e^{-\lambda}}. \quad (17)$$

Similarly,

$$H_1 = \frac{1 - e^{-\lambda T_{\text{ON}}}}{1 - e^{-\lambda}} \quad (18)$$

and

$$H_{\text{LD}} = \frac{1 - e^{-\lambda T_{\text{ON}}}}{1 - e^{-\lambda}} \quad (19)$$

When the device is at S_{2i-1} , $i \in [2, n+1]$, there are three cases for data arrival: (i) data arrive in the sleep period of last short cycle; (ii) no data arrive within the period of T_{ON} ; (iii) data arrive within the period of T_{ON} . In case (i), the device stays for a single subframe in On Duration to receive data with probability p_S . In case (iii), assume that data arrive at the j th subframe of T_{ON} with probability p_j^{ON} , and the holding time is T_j^{ON} . Then

$$\begin{aligned}
 H_{2i-1} &= p_S T_{\text{sub}} + p_{2i-1,2i} T_{\text{ON}} \\
 &\quad + \sum_{j=1}^{T_{\text{ON}}} p_j^{ON} T_j^{ON}, \quad i \in [2, n+1]. \quad (20)
 \end{aligned}$$

where

$$\begin{aligned}
 p_S &= \Pr[T_d < (T_S - T_{\text{ON}})] = 1 - e^{-\lambda(T_S - T_{\text{ON}})}, \\
 p_j^{ON} &= \Pr[T_S - T_{\text{ON}} + j-1 < T_d < T_S - T_{\text{ON}} + j] \\
 &= e^{-\lambda(T_S - T_{\text{ON}} + j-1)} - e^{-\lambda(T_S - T_{\text{ON}} + j)}, \quad j \in [1, T_{\text{ON}}], \quad (21)
 \end{aligned}$$

and

$$T_j^{ON} = j, \quad j \in [1, T_{\text{ON}}]. \quad (22)$$

From (20)-(22), we have

$$\begin{aligned}
 H_{2i-1} &= \frac{e^{-\lambda(T_S - T_{\text{ON}})} - e^{-\lambda T_S}}{1 - e^{-\lambda}} + 1 - e^{-\lambda(T_S - T_{\text{ON}})}, \\
 &\quad i \in [2, n+1]. \quad (23)
 \end{aligned}$$

Similarly,

$$\begin{aligned}
 H_{2i-1} &= \frac{e^{-\lambda(T_L - T_{\text{ON}})} - e^{-\lambda T_L}}{1 - e^{-\lambda}} + 1 - e^{-\lambda(T_L - T_{\text{ON}})}, \\
 &\quad i \in [n+2, n+m]. \quad (24)
 \end{aligned}$$

H_{2i} , $i \in [1, n]$ and H_{2i} , $i \in [n+1, n+m]$ are the sleep periods of a device during a short cycle and a long cycle, respectively. Hence,

$$\begin{aligned}
 H_{2i} &= T_S - T_{\text{ON}}, \quad i \in [1, n] \\
 H_{2i} &= T_L - T_{\text{ON}}, \quad i \in [n+1, n+m]. \quad (25)
 \end{aligned}$$

Since the device monitors PDCCH for a single subframe in each paging cycle, $H_1^* = H_3^* = T_{\text{sub}} = 1$. Similarly,

$$H_2^* = H_4^* = T_P - T_{\text{sub}}. \quad (26)$$

Finally, based on the definitions of S_{RR} , S_{RE} , and $S_{\text{RE}'}$, $H_{\text{RR}} = T_{\text{RR}}$ and $H_{\text{RE}} = H_{\text{RE}'} = T_{\text{RE}}$.

D. VALIDATION OF α AND δ

From the above derivation, we can obtain the values of T , α , and δ by substituting (12), (16)-(19), and (23)-(26) into (1), (5), and (6). The analytical model is validated against discrete event simulation experiments carried out in Matlab based simulator. The discrete event simulation model with $n = 1$ and $m = 1$ is described in [1]. As shown in Table 2, the analytical analysis is consistent with the simulation results.

TABLE 2. Validation of simulation and analytical models ($T_0 = 20$ ms, $T_{\text{ON}} = 80$ ms, $T_S = 160$ ms, $T_L = 320$ ms, $T_P = 640$ ms, $T_{\text{RR}} = 1$ ms, $T_{\text{RE}} = 97$ ms).

$1/\lambda$ (n, m)	1,000			100,000		
	(1, 1)	(1, 5)	(10, 5)	(1, 1)	(1, 5)	(10, 5)
α (Ana.)	0.8625	0.7714	0.7548	0.9965	0.9933	0.9933
α (Sim.)	0.8625	0.7718	0.7539	0.9965	0.9933	0.9933
Error	0%	0.05%	0.11%	0%	0%	0%
δ (Ana.)	331.47	166.64	158.24	415.75	411.62	411.62
δ (Sim.)	331.34	166.59	157.75	415.78	411.63	411.63
Error	0.04%	0.03%	0.31%	0%	0%	0%

TABLE 3. Parameter setting used in simulation experiments.

Parameter	Value
$1/\lambda$	1000 ms
T_0	20 ms
T_{ON}	80 ms
T_S	160 ms
T_L	320 ms
T_P	40 ms
T_{RR}	1 ms
T_{RE}	97 ms
n	1
m	5
n^* (for DRX)	16
k (for DDDRX)	30

V. PERFORMANCE EVALUATION

Based on the simulation experiments validated against the analytical model, this section investigates the performance of DRX mechanisms. In subsection V-A, we observe the effects of three ODRX parameters, including T_P , n , and m , on δ and α . Then we propose guidelines to configure these three parameters. To show the feasibility of ODRX, we compare performance of ODRX with that of standard DRX [20] and Dynamic DRX [13] in subsection V-B. Table 3 shows the default parameter setting used in our simulation experiments. Note that the setting of the short cycle (T_S) is in the range defined in [24] and the setting of the long cycle (T_L) is a multiple of the short cycle length, which is also defined in [24]. Moreover, the setting of the latency of RRC connection release (T_{RR}) and the latency of RRC connection establishment (T_{RE}) is based on the study in [25].

A. THE ESTIMATION OF ODRX PARAMETERS

The effect of T_P on α and δ is shown in Fig. 3. For all λ values, α converges between $T_P = 40$ ms and $T_P = 80$ ms (shown by the range between two vertical dashed lines). We also observe that the convergent boundary is consistent for various (n, m) pairs (not shown in this paper). Exactly, when T_P is set to be larger than 80 ms, there is no further help in enhancing power saving but wake up latency vainly increase. Based on this observation, we choose $T_P = 40$ ms as the optimal value in our following experiments.

In our simulation experiments, we observe that the optimal setting of the n and m values is quite different under variant data arrival rate λ . Without loss of generality, we set $1/\lambda = 1000$ ms to show how to choose appropriate n and m values.

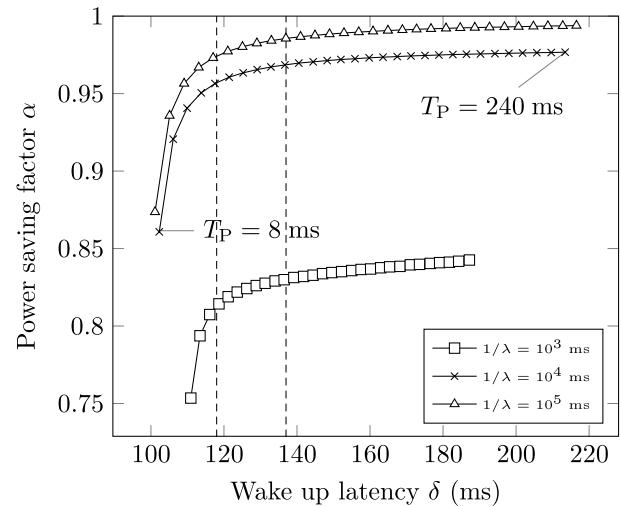


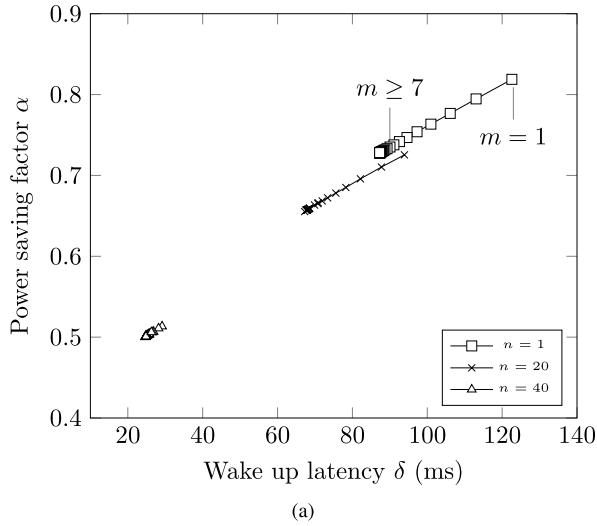
FIGURE 3. Effect of T_P on δ and α ($n = 1, m = 5$).

Effects of n and m on α and δ are shown in Fig. 4. When n or m is small, α and δ are large. This is because the device enters paging cycle easily. In other words, data arrive in sleep period with higher probability and the device can save more power with larger latency. For all n values, α and δ converge around $m = 7$. For all m values, α and δ converge around $n = 50$. That is, for example, compared to the case $m = 7$, setting m value to be larger than 7 does no further help in reducing wake up latency or increasing power saving. In Fig. 4(a), when n is small (see the line with $n = 1$), δ is bounded between 92 and 121. When $n = 40$, δ is bounded within 25 and 28 because the device hardly enters paging cycle.

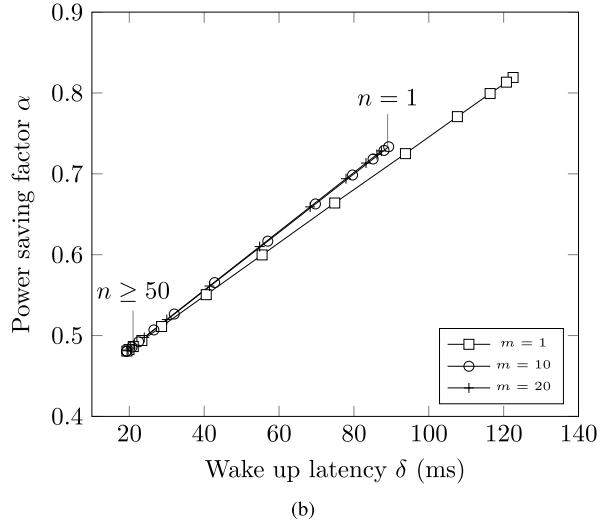
When ODRX is deployed in the real environment, five major ODRX parameters (T_P, T_S, T_L, n , and m) are able to be configured. While adjusting all the parameters is not an efficient way, we summarize the results to provide the guideline to set and adjust the parameters as follows. The paging cycle length, T_P , is optimal between 40 ms and 80 ms. The short cycle length, T_S , should be less than or equal to 320 ms; otherwise, the long cycle length, T_L , has no effect on the power saving factor and the wake up latency. Since the optimal setting of n and m is varying under different arrival rate, selecting n and m which can meet the requirements (e.g., the delay budget) need to be on the basis of the estimated arrival rate. In practice, we suggest to first set n value based on the wake up latency requirement, and then set m value to achieve the target power saving factor. Note that T_P, T_S , and T_L can be also adjusted to be suitable for the devices in the real environment. However, in a more efficient way, we suggest to appropriately adjust n and m rather than the others.

B. PERFORMANCE COMPARISON AMONG DRX MECHANISMS

In standard DRX, when the inactivity timer expires, the device enters short cycle. A short cycle counter threshold n^* is defined to determine when the device shall enter long cycle. If no data arrive within n^* short cycles, the device



(a)



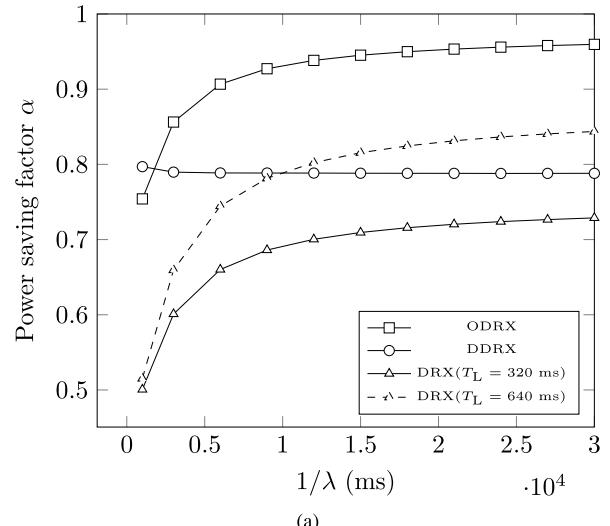
(b)

FIGURE 4. Effects of n and m on α and δ . (a) Effect of m . (b) Effect of n .

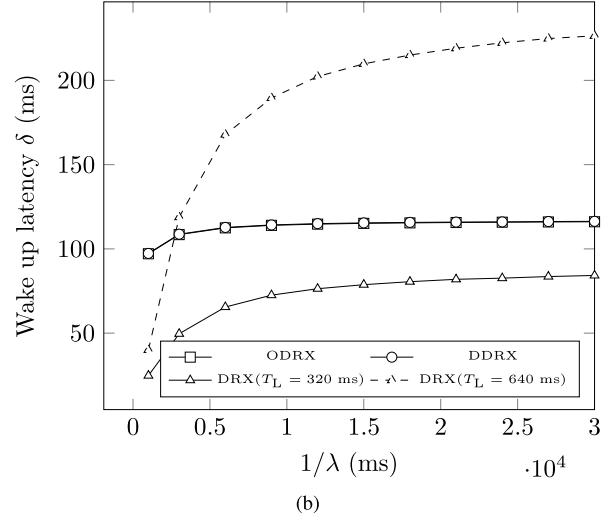
enters long cycle. If there are data buffered at the eNB when the device is in short or long cycle, the device receives data and returns back to active state. Based on the suggested parameter setting in [20], we set n^* to 16 in this paper.

In DDRX, C_w is the given mean wake up latency constraint. The average latency and sleep proportion of each k packets are calculated. The length of sleep cycle is dynamically adjusted to approximate the calculated average latency to C_w . If the length of sleep cycle does not need to be changed, the inactivity timer is then adjusted to enhance the sleep proportion. To compare with ODRX, the given mean wake up latency, C_w , is set to the observed δ of ODRX. The parameter k is set to 30.

In Fig. 5, we compare ODRX with DRX and DDRX under different $1/\lambda$. This figure illustrates that both α and δ increase as $1/\lambda$ increases in ODRX and DRX, while both performance measures maintain in a small range (e.g., α is in (0.79, 0.8)) in DDRX. The observation for ODRX and DRX is consistent with the intuition that when arrival rate is small (i.e., $1/\lambda$ is large), the device can have more sleep. Under the same T_L



(a)



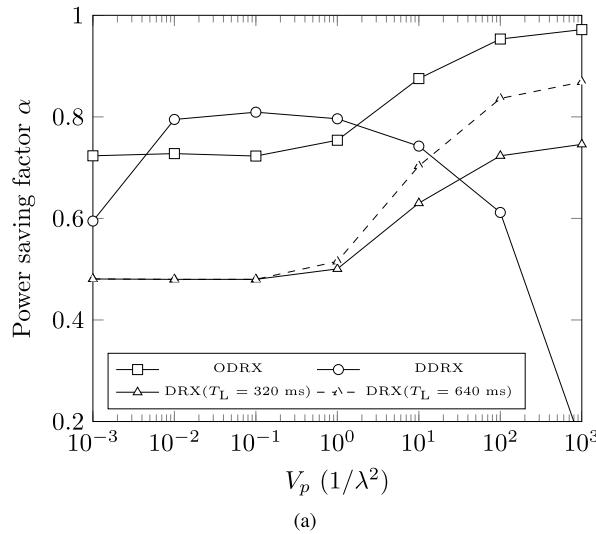
(b)

FIGURE 5. Effect of $1/\lambda$ on α and δ ($n = 1, m = 5$). (a) α . (b) δ .

value, the device in ODRX can save more power than that in DRX by sacrificing a few amount of wake up latency. For example, when $T_L = 320$ ms, ODRX can gain extra 23.1% of power saving (95.9% – 72.8%) by sacrificing 31.8 ms of latency (116 ms – 84.2 ms) when $1/\lambda = 30000$ ms. When $1/\lambda = 1000$ ms, ODRX can gain extra 25.4% of power saving (75.4% – 50%) by sacrificing 72.4 ms of latency (97.1 ms – 24.7 ms). Fortunately, in many MTC applications, 70 ms of wake up latency can be ignored, but 25% of power saving is noticeable. Note that this phenomenon also occurs in other n^* values (e.g., $n^* = 1$), and the details are ignored in this paper.

Besides, we observe that, in DRX, over-increasing T_L does little help to increase α , but increase δ significantly. For example, when $1/\lambda = 30000$ ms, increasing T_L from 160 ms to 320 ms can save extra 11.5% power (84.3% – 72.8%), but also introduce 169% latency ((226.4 ms – 84.2 ms)/84.2 ms).

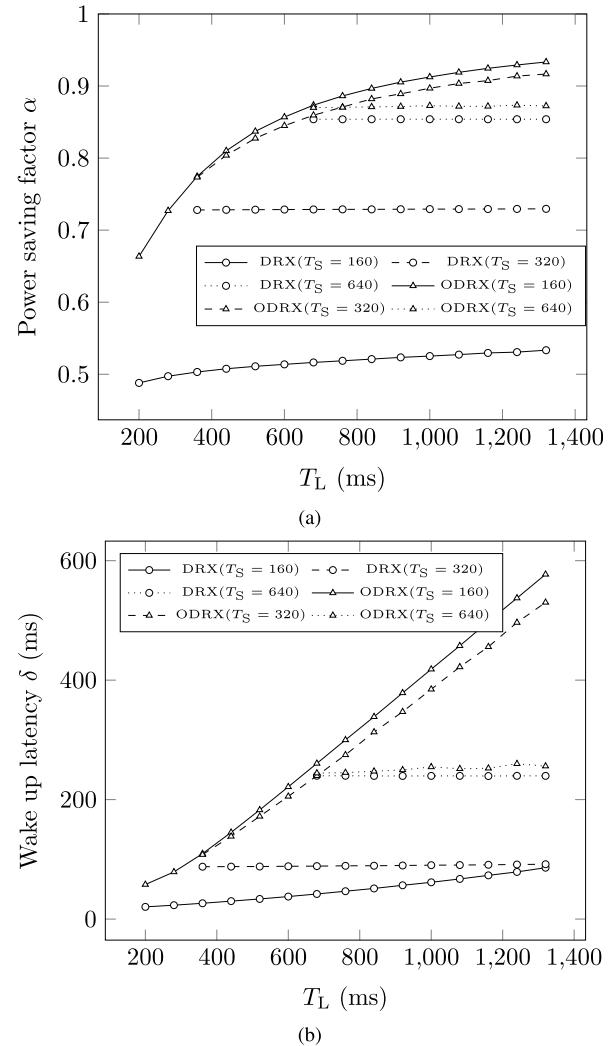
In Fig. 5, under the same latency, ODRX can save more power than DDRX as long as $1/\lambda > 2000$ ms.

FIGURE 6. Effect of V_p on α and δ ($n = 1, m = 5$). (a) α . (b) δ .

This phenomenon is due to the late adjustment of DDRX. In DDRX, cycle length is dynamically adjusted to meet the latency constraint every k packets. In average, if $1/\lambda = 2000$ ms and $k = 30$, the cycle length is adjusted every 60 seconds, which is approximately 150 to 300 cycles. Although we may choose a smaller k values (e.g., $k = 1$), the smaller k may introduce large amount of computation and power consumption (which contradict the major goal of DRX mechanisms). Therefore, DDRX is not suitable for MTC applications with small data arrival rate.

To study DRX performance under different data traffic, we suppose that the inter-data-arrival time T_d has the Gamma distribution with mean $1/\lambda$ and variance V_p . The Gamma distribution is considered because the distribution of any positive random variable can be approximated by a mixture of Gamma distributions (see [26, Lemma 3.9]). Fig. 6 shows the performance of DRX mechanisms under different V_p .

In DRX and ODRX mechanisms, α increases as V_p increases. When V_p becomes large, more cycles with many

FIGURE 7. Effects of T_S and T_L on α and δ ($n = 16, m = 7$). (a) α . (b) δ .

arrivals and without any arrival are observed. In this case, the device is more likely to enter long cycle (for standard DRX) and paging cycle (for ODRX), and thus save more power. In DDRX, α increases and then decreases as V_p increases. The results show that the dynamic adjustment in DDRX works well when the data arrivals are regular, but does not work well when the data arrivals become irregular.

Finally, we observe the effects of short cycle length and long cycle length in ODRX and DRX on α and δ . The results are shown in Fig. 7.

In Fig. 7, α and δ in ODRX significantly increase as T_L increases, while α and δ in DRX are insensitive to T_L . Due to the O_flag, the device has higher probability to enter long cycle, which results in saving more power. On the other hand, T_L has the little impact on α and δ in DRX because the device stays in short cycle more often than long cycle. The results show that the “optimistic” behavior introduced in ODRX can effectively improve power saving.

VI. CONCLUSIONS AND DISCUSSIONS

In this paper, we proposed optimistic DRX (ODRX) mechanism for MTC devices to save more power. Two major “optimistic” features are introduced in ODRX: (1) RRC connection release and re-establishment are considered for power saving. (2) An optimistic flag is used to make the device enter long cycle directly. We then proposed analytical and simulation models to investigate performance of ODRX. We also proposed guidelines to configure parameters n , m and T_p . Finally, we compare ODRX with standard DRX and dynamic DRX (DDRX) through simulation experiments.

The simulation results show that ODRX is quite suitable for many non-realtime MTC applications. Precisely, ODRX can save more than 95% power when data arrival rate is small and more than 75% power when data arrival rate is large. Compared to standard DRX, the device in ODRX can remarkably save more power by sacrificing inconspicuous latency. Precisely, ODRX can save more than 20% extra power in most cases by sacrificing at most 73ms wake up latency. On the other hand, ODRX outperforms DDRX in power saving under the same latency requirements.

In the past few years, the standardization of the 5G network is heatedly discussed although the 4G network is just widely deployed. By 2020, the number of both mobile terminals and IoT connections will grow rapidly [6]. The variety of machine-oriented applications is also growing at an unforeseen pace. Massive machine type communications is one of the major use cases for 5G. A large number of connected devices, a very long battery life, non-delay-sensitive data, and small data transmission are all the characteristics of this use case [6]. For example, the vending machines are occasionally requested to generate small reports in non-delay-sensitive manner. Among the solutions to low power consumption described in [27], applying DRX mechanism in the devices is one of the handiest ways. To the best of the authors’ knowledge, DRX mechanism in 5G standards is very similar to that in LTE-A. That is, based on the results in Section V, implementing ODRX with appropriate parameter setting (according to different requirements of applications, such as delay budget) is able to dramatically reduce the power consumption in 5G.

In conclusion, the ODRX implementation enables the LTE-A as well as 5G devices to prolong the battery life and achieve energy efficiency. Certainly, ODRX application in 5G still needs further study.

REFERENCES

- [1] H.-L. Chang, S.-L. Lu, T.-H. Chuang, C.-Y. Lin, M.-H. Tsai, and S.-I. Sou, “Optimistic DRX for machine-type communications,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
- [2] *Service requirements for Machine-Type Communications (MTC); Stage 1*, document TS 22.368, 3GPP, Jan. 2015.
- [3] Z. Abbas and W. Yoon, “A survey on energy conserving mechanisms for the Internet of Things: Wireless networking aspects,” *Sensors*, vol. 15, no. 10, pp. 24818–24847, 2015.
- [4] R. Want, B. N. Schilit, and S. Jenson, “Enabling the Internet of Things,” *Computer*, vol. 48, no. 1, pp. 28–35, 2015.
- [5] A. Osseiran *et al.*, “Scenarios for 5G mobile and wireless communications: The vision of the METIS project,” *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 26–35, May 2014.
- [6] W. Xiang, K. Zheng, and X. S. Shen, *5G Mobile Communications*. Cham, Switzerland: Springer, 2016.
- [7] A. Nordrum, “The Internet of fewer Things [News],” *IEEE Spectr.*, vol. 53, no. 10, pp. 12–13, Oct. 2016.
- [8] *Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) Protocol Specification*, document TS 36.321, 3GPP, Mar. 2017.
- [9] *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) Procedures in Idle Mode*, document TS 36.304, 3GPP, Nov. 2016.
- [10] S. C. Jha, A. T. Koc, R. Vannithamby, and M. Torlak, “Adaptive DRX configuration to optimize device power saving and latency of mobile applications over LTE advanced network,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 6210–6214.
- [11] A. T. Koc, S. C. Jha, R. Vannithamby, and M. Torlak, “Optimizing DRX configuration to improve battery power saving and latency of active mobile applications over LTE-A network,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2013, pp. 568–573.
- [12] 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 Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2614–2625, May 2014.
- [13] S.-R. Yang, “Dynamic power saving mechanism for 3G UMTS system,” *Mobile Netw. Appl.*, vol. 12, no. 1, pp. 5–14, 2007.
- [14] R. M. Karthik and A. Chakrapani, “Practical algorithm for power efficient DRX configuration in next generation mobiles,” in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1106–1114.
- [15] P.-J. Hsieh, G.-Y. Lin, C.-Y. Chen, and H.-Y. Wei, “Accurate modeling of the DRX mechanism with predetermined DRX cycles based on the 3GPP LTE standard,” *Mobile Netw. Appl.*, vol. 21, no. 2, pp. 259–271, 2016. [Online]. Available: <http://dx.doi.org/10.1007/s11036-015-0662-8>
- [16] J. Wu, T. Zhang, and Z. Zeng, “Performance analysis of discontinuous reception mechanism with web traffic in LTE networks,” in *Proc. IEEE 24th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2013, pp. 1676–1681.
- [17] K. Wang, X. Li, and H. Ji, “Modeling 3GPP LTE advanced DRX mechanism under multimedia traffic,” *IEEE Commun. Lett.*, vol. 18, no. 7, pp. 1238–1241, Jul. 2014.
- [18] K. Wang, X. Li, H. Ji, and X. Du, “Modeling and optimizing the LTE discontinuous reception mechanism under self-similar traffic,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5595–5610, Jul. 2016.
- [19] J.-M. Liang, P.-C. Hsieh, J.-J. Chen, and Y.-C. Tseng, “Energy-efficient DRX scheduling for multicast transmissions in 3GPP LTE-advanced wireless networks,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2013, pp. 551–556.
- [20] K. Zhou, N. Nikaein, and T. Spyropoulos, “LTE/LTE-A discontinuous reception modeling for machine type communications,” *IEEE Wireless Commun. Lett.*, vol. 2, no. 1, pp. 102–105, Feb. 2013.
- [21] J. Wu, T. Zhang, Z. Zeng, and H. Chen, “Study on discontinuous reception modeling for M2M traffic in LTE-A networks,” in *Proc. 15th IEEE Int. Conf. Commun. Technol. (ICCT)*, Nov. 2013, pp. 584–588.
- [22] J.-M. Liang, J.-J. Chen, H.-H. Cheng, and Y.-C. Tseng, “An energy-efficient sleep scheduling With QoS consideration in 3GPP LTE-advanced networks for Internet of Things,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 3, no. 1, pp. 13–22, Mar. 2013.
- [23] L. Kleinrock, *Queueing Systems: Theory*, vol. 1. New York, NY, USA: Wiley, 1976.
- [24] *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification*, document TS 36.331, 3GPP, Mar. 2017.
- [25] S. Mohan, R. Kapoor, B. Mohanty. (2011). *Latency in HSPA Data Networks*. [Online]. Available: <https://www.qualcomm.com/media/documents/qualcomm-research-latency-hspa-data-networks>
- [26] F. P. Kelly, *Reversibility and Stochastic Networks*. New York, NY, USA: Wiley, 1979.
- [27] H. Sharatiadari *et al.*, “Machine-type communications: Current status and future perspectives toward 5G systems,” *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 10–17, Sep. 2015.



HUI-LING CHANG (S'17) received the B.S. degree from National Cheng Kung University, Taiwan, in 2014, where she is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering. Her research interests include wireless sensor networks and mobile networks.



MENG-HSUN TSAI (S'04–M'10) received the B.S., M.S., and Ph.D. degrees from National Chiao Tung University in 2002, 2004, and 2009, respectively. In 2012, he joined the University of Southern California, as a Visiting Scholar. He is currently an Associate Professor with the Department of Computer Science and Information Engineering, National Cheng Kung University. His current research interests include design and analysis of mobile networks, software defined networking, and performance modeling. He was a recipient of the Exploration Research Award of Pan Wen Yuan Foundation in 2012. He received the Outstanding Contribution Award from the IEEE Taipei Section in 2010.

• • •