Flag-Assisted Early Release of RRC Scheme for Power Saving in NB-IoT System

Hui-Ling Chang, Chung-Ying Hsieh, and Meng-Hsun Tsai(✉)

National Cheng Kung University, Tainan, Taiwan
momo@imslab.csie.ncku.edu.tw, P76054290@mail.ncku.edu.tw,
tsaimh@csie.ncku.edu.tw

Abstract. As the 5G standard is about to be completed, the IoT applications will have the unprecedented development. Massive and various IoT devices sense and interact with the environment. Most of those devices are battery-powered and some of them are deployed at inaccessible locations, so how to reduce the power consumption is a critical issue. 3GPP proposed eDRX and two transport optimization mechanisms to help devices reduce power consumption. In this paper, based on the CP CIoT EPS Optimization, a flag-assisted Early Release of RRC scheme is proposed. The message flow is modified to make IoT devices enter RRC_IDLE early. The result shows that the flag-assisted Early Release of RRC can help IoT devices save power further.

Keywords: NB-IoT · CP CIoT EPS Optimization · Power saving · eDRX

1 Introduction

As the Fifth Generation (5G) standard is about to be completed and the worldwide telecom operators have announced the commercial 5G would be launched between 2018 and 2022 [1], the Internet of Things (IoT) applications will have the unprecedented development. Cisco estimated that IoT connections will be more than 14.6 billion by 2022 [2]. In the near future, smart home, smart city, smart factory, smart mobility, health care, etc. will be realized in our daily life [3]. A variety of devices with sensor, actuator, or both have the ability to communicate with each other and interact with the environment. Most of the IoT applications are small data transmission services. The Third Generation Partnership Project (3GPP) proposed the Narrow-Band IoT (NB-IoT) as the radio access technology in 5G for the classic IoT model of static, low-data-rate, and delay-tolerant nodes. NB-IoT supports the network connection which characterizes massive connection, low cost, ultra-low power consumption, and coverage enhancement.

How to reduce the power consumption is always a critical issue, because a majority of IoT devices are battery-powered and some of them are deployed at

D.-J. Deng et al. (Eds.): SGIoT 2019, LNICST 324, pp. 3–15, 2020.
https://doi.org/10.1007/978-3-030-49610-4_1

inaccessible locations. Even though the battery of some devices, e.g. smart fitness band, can be replaced or charged easily, prolonging the battery life in order to reduce the charging frequency is still important due to the user friendliness. In terms of power efficiency, NB-IoT has targeted a ten-year battery life with the capacity of 5 Wh [4], which is a quite challenging goal.

To achieve this target, 3GPP introduced extended Discontinuous Reception (eDRX) in Release 13. In legacy LTE, the devices adopting DRX periodically sleep to turn off the radio module for the sake of saving power and wake up to monitor the Physical Downlink Control Channel (PDCCH). The substantial difference between DRX and eDRX is that the device adopting eDRX spends more time to stay in sleep period, which greatly reduces the power consumption. The maximum cycle length in idle mode DRX is only 2.56 s while the cycle length can be up to 2.91 h in NB-IoT eDRX mechanism [5].

Moreover, 3GPP also proposed two transport optimization mechanisms in Release 13 to reduce the signal transmission [6]. One is User Plane Cellular IoT Evolved Packet System (UP CIoT EPS) Optimization which allows user equipments (UEs) to reduce the bearer setup signal (i.e., Access Stratum (AS) security setup and Radio Resource Control (RRC) reconfiguration messages) after an initial RRC connection is established by suspending and resuming procedures. The other one is Control Plane (CP) CIoT EPS Optimization. Although CP is used to transmit the signal message, CP CIoT EPS Optimization makes the user data encapsulated in Non-Access Stratum (NAS) message. As a result, the UEs avoid AS security setup and the UP bearer establishment. By these optimizations, NB-IoT has the ability to efficiently support massive Machine-Type Communication (mMTC) and further reduce the power consumption. Since most applications of NB-IoT are small data transmission, establishing user plane is unnecessary. 3GPP also specifies that UEs adopting NB-IoT must support CP CIoT EPS Optimization, while UP CIoT EPS Optimization is optional. For convenience, CP CIoT EPS Optimization is called CP CIoT for short in the following content.

The analysis in [7] shows that compared to using conventional Service Request (SR) procedure, adopting CP CIoT increases the battery life up to five years. However, this procedure still can be improved. Although eDRX can be performed in both RRC_CONNECTED and RRC_IDLE states, the power consumption in the RRC_CONNECTED Sleep mode is 200 times higher than that in RRC_IDLE Standby mode [8]. Obviously, the devices should enter RRC_IDLE as soon as possible after they transport the data. Therefore, based on CP CIoT, we proposed flag-assisted Early Release of RRC (ER-RRC) scheme in this paper. The existing message flow is modified for the IoT devices to stay in RRC_IDLE state longer and further save more power.

In LTE/LTE-A, some literatures analyzed the trade-off between the power saving and the latency by tuning the DRX parameters and tried to optimize the parameter setting under different traffic models to save power [9,10]. In our previous work [11], Optimistic DRX (ODRX) was introduced to appropriately skip the short cycles, which results in extra 20% power saving with just about 70 ms extra delay. In NB-IoT, the power saving issue is still popular. The authors in [12] analyzed the trade-off between the tracking area update (TAU) cost and the paging cost, and then optimized the length of eDRX cycle. The signaling cost is mitigated so the processing power is also reduced. A prediction-based power saving mechanism is developed in [13] to allocate resource in advance by observing the uplink occurrence and processing delay. The device can send uplink data without a scheduling request. The transmission time is reduced and thus the power can be saved. A group-based DRX is introduced to provide better power saving for a large-scale NB-IoT system [14]. The group leader adopts the DRX scheme different from that of the group members. The leader wakes up more often than the members do to monitor the downlink channel. The power of total IoT devices in the whole system can be saved and the signaling congestion can be avoided.

To the best of authors' knowledge, no research focused on improving the message flow. In this paper, CP CIoT is enhanced to make IoT devices enter eDRX earlier. Consequently, more power can be saved.

The rest of this paper is organized as follows. In Sect. 2, CP CIoT is described in detail. The proposed ER-RRC scheme is elaborated in Sect. 3. We also proposed the analytical model to validate our simulation in Sect. 4. The performance evaluation is shown in Sect. 5. Finally, we made the conclusion in Sect. 6.

2 CP CIoT EPS Optimization

In this section, the procedure of data transport in CP CIoT is elaborated in detail. First of all, a new information element (IE) called Release Assistance Indication (RAI) is introduced for CP CIoT. When the devices intend to transmit data, RAI can be included in NAS message to indicate that no further uplink (UL) or downlink (DL) data subsequent to this UL data is expected, or only a single DL data (e.g. a response to this UL data) is expected. In the first case, the RAI is set to one while in the second case, the RAI is set to two [15]. The message flows of mobile originated (MO) data transport with two RAI values are explained as follows.

Figure 1 illustrates the MO message flow with RAI = 1.

Step 0. The device is in RRC_IDLE. In this state, the device releases radio resource and performs eDRX. In each paging cycle, the device wakes up for only 1 ms to monitor the PDCCH [16].

Steps 1–2. The device performs the random access procedure to inform the evolved node B (eNB) its transmission request.

Steps 3–5. The device establishes a RRC connection. In RRC Connection Setup Complete message, the data and RAI value (i.e., RAI = 1) are included in the NAS PDU.

Step 6. The eNB relays the NAS PDU in S1-AP Initial UE message to the Mobility Management Entity (MME). Note that the eNB cannot retrieve information from NAS message.

Step 7. The MME checks the integrity of the NAS PDU and decrypts the UL user data for further transmission.

Steps 8–11. The MME requests to re-activate the bearers for the device.

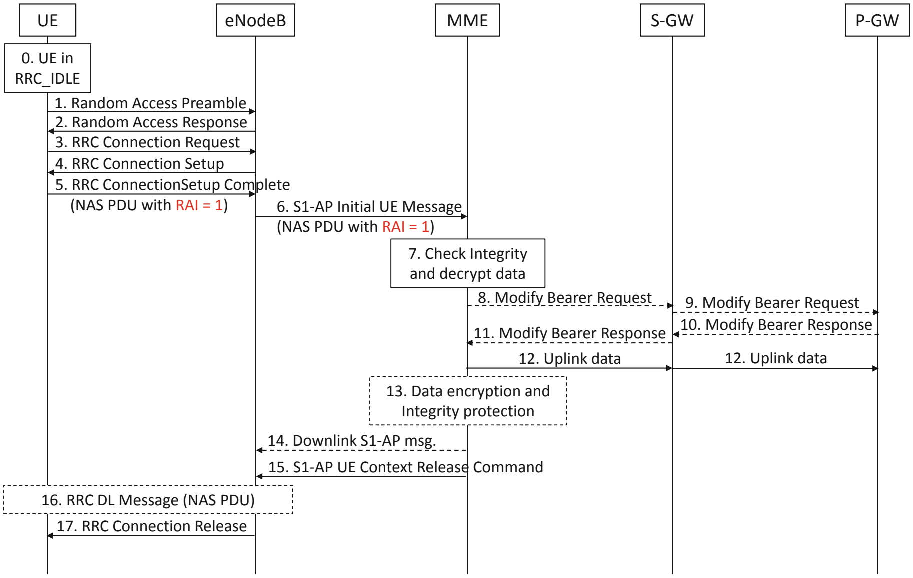
UE
eNodeB
MME
S-GW
P-GW
0. UE in RRC_IDLE
1. Random Access Preamble
2. Random Access Response
3. RRC Connection Request
4. RRC Connection Setup
5. RRC ConnectionSetup Complete (NAS PDU with RAI = 1)
6. S1-AP Initial UE Message (NAS PDU with RAI = 1)
7. Check Integrity and decrypt data
8. Modify Bearer Request
9. Modify Bearer Request
10. Modify Bearer Response
11. Modify Bearer Response
12. Uplink data
12. Uplink data
13. Data encryption and Integrity protection
14. Downlink S1-AP msg.
15. S1-AP UE Context Release Command
16. RRC DL Message (NAS PDU)
17. RRC Connection Release

Fig. 1. MO data transport in CP CIoT EPS Optimization with RAI = 1

Step 12. The MME sends the UL user data to the Packet Data Network gateway (P-GW) via the Serving gateway (S-GW). Since RAI value is set to one, all application layer data exchanges are completed with this UL data. However, the MME may have pending data for the device. Therefore, the MME checks whether there is buffered data for the device or not. If not, the following **Step 13.**, **Step 14.**, and **Step 16.** are skipped.

Step 13. The MME is aware of pending MT data so it performs data encryption and protects the integrity of this data.

Step 14. The MT data is encapsulated in a NAS PDU and sent to the eNB.

Step 15. If the MME does not have pending data for the device, it immediately requests the eNB to release the connection after sending UL data to P-GW. If the MME has pending data for the device, it requests the eNB to release the connection right after sending DL S1-AP message.

Step 16. The eNB sends the RRC DL message with NAS PDU where the MT data is included to the device.

Step 17. The RRC connection between the device and the eNB is released. In the meantime, the S1 connection between the eNB and the MME for the device is also released. All the bearers are torn down. The eNB removes the context of the device and the device goes back to RRC_IDLE.

Figure 2 shows the MO message flow with RAI = 2, which is similar to that with RAI = 1. Because the RAI value in the NAS PDU is two (in **Step 5.**), the MME should receive DL data from P-GW (in **Step 13.**). After receiving this DL data, the device finishes transmission activities with the network and goes back to RRC_IDLE.

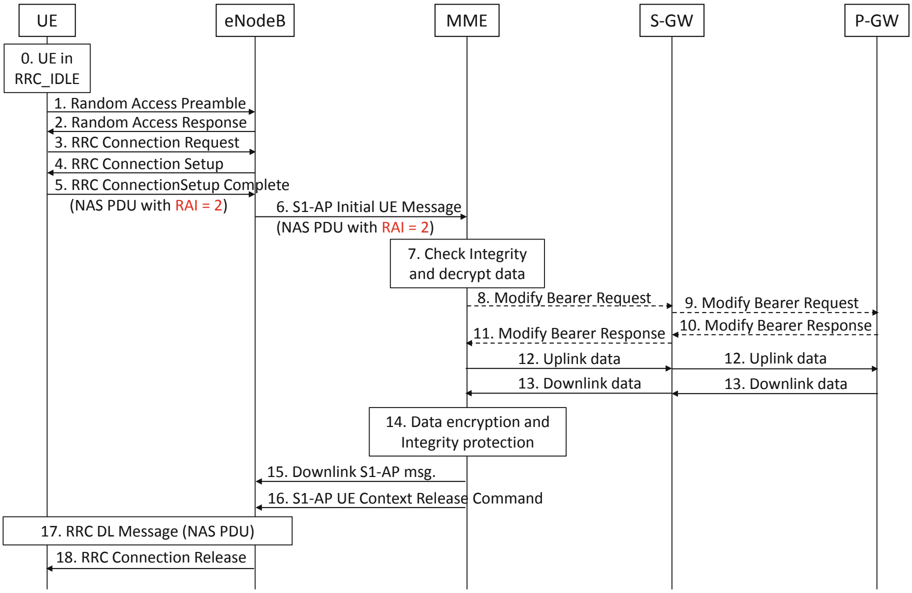
UE
eNodeB
MME
S-GW
P-GW
0. UE in RRC_IDLE
1. Random Access Preamble
2. Random Access Response
3. RRC Connection Request
4. RRC Connection Setup
5. RRC ConnectionSetup Complete (NAS PDU with RAI = 2)
6. S1-AP Initial UE Message (NAS PDU with RAI = 2)
7. Check Integrity and decrypt data
8. Modify Bearer Request
9. Modify Bearer Request
10. Modify Bearer Response
11. Modify Bearer Response
12. Uplink data
12. Uplink data
13. Downlink data
13. Downlink data
14. Data encryption and Integrity protection
15. Downlink S1-AP msg.
16. S1-AP UE Context Release Command
17. RRC DL Message (NAS PDU)
18. RRC Connection Release

Fig. 2. MO data transport in CP CIoT EPS Optimization with RAI = 2

Note that the device can also transmit data without RAI. In this situation, the basic message flow is the same as that in Fig. 1 (from **Step 0.** to **Step 12.**). After the MME uplinks the data to the P-GW, the connection between the device and the network is not released immediately. The eNB starts the inactivity timer (denoted as T_0). If the device sends or receives data before T_0 expires, the eNB re-starts T_0. Otherwise, the eNB sends the RRC Connection Release command to the device after T_0 expires.

3 Flag-Assisted Early Release of RRC

The operation of the flag-assisted ER-RRC scheme is described in this section. In many IoT applications, the devices sense the environment and report the variation to the remote application server. These devices have MO data much more than MT data. That is when a device uplinks data with RAI = 1, it is far more likely that no pending data is at the MME. Therefore, the eNB can inform the device to enter RRC_IDLE earlier. However, for some cases, the MME might do additional paging if the device releases the RRC connection too early to receive the pending data.

In the flag-assisted ER-RRC scheme, a flag is used to indicate the device has pending data when last time the device enters RRC_IDLE. Additionally, two new IE are added to help the eNB decide whether to release the RRC connection or not. One is called *earlyReleaseRRC* and the other one is called *pendingData*. The modified message flow with ER-RRC is illustrated in Fig. 3. The details are described in the following steps.

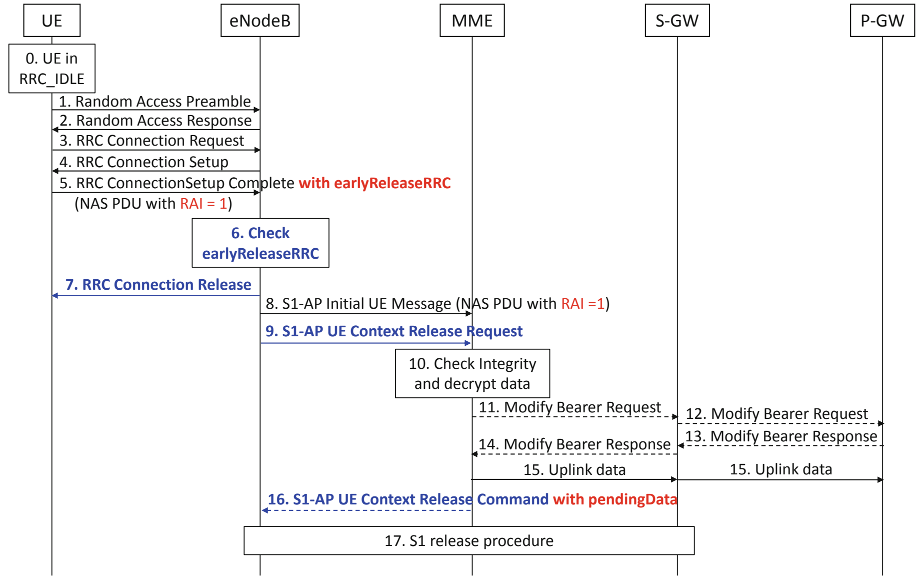
UE
eNodeB
MME
S-GW
P-GW
0. UE in RRC_IDLE
1. Random Access Preamble
2. Random Access Response
3. RRC Connection Request
4. RRC Connection Setup
5. RRC ConnectionSetup Complete with earlyReleaseRRC (NAS PDU with RAI = 1)
6. Check earlyReleaseRRC
7. RRC Connection Release
8. S1-AP Initial UE Message (NAS PDU with RAI =1)
9. S1-AP UE Context Release Request
10. Check Integrity and decrypt data
11. Modify Bearer Request
12. Modify Bearer Request
13. Modify Bearer Response
14. Modify Bearer Response
15. Uplink data
15. Uplink data
16. S1-AP UE Context Release Command with pendingData
17. S1 release procedure

Fig. 3. MO data transport with ER-RRC scheme

Steps 0–4. These steps are the same as that in Fig. 1.

Step 5. In RRC Connection Setup Complete message, *earlyReleaseRRC* is a new IE for the device to set as true, if the device sets RAI value to one.

Step 6. The eNB checks the *earlyReleaseRRC* value before relaying the NAS PDU to the MME.

Step 7. Once the *earlyReleaseRRC* is set to true, the eNB checks the flag. If the flag is false, the eNB immediately releases the RRC connection to the device. The device goes back to RRC_IDLE after receiving this message. In contrast, if the flag is true, the eNB follows CP CIoT message flow instead of ER-RRC.

Steps 8–9. The eNB relays the NAS PDU in S1-AP Initial UE message to the MME. At the same time, the eNB requests the MME to release UE-associated logical S1 connection.

Steps 10–15. The integrity of the data is checked. Before sending the data to P-GW, the MME decrypts the data and requests to re-activate the bearer for the device.

Steps 16–17. The final step is to release the S1 connection between the eNB and the MME. S1-AP UE Context Release Command is one of the messages in S1 release procedure. The *pendingData* is added in S1-AP UE Context Release Command and sent to the eNB. If the MME is aware of pending data, it sets the new IE, *pendingData*, to true. Once the eNB observes that *pendingData* is true, it sets the flag of the device to true. Otherwise, the eNB sets the flag of the device to false.

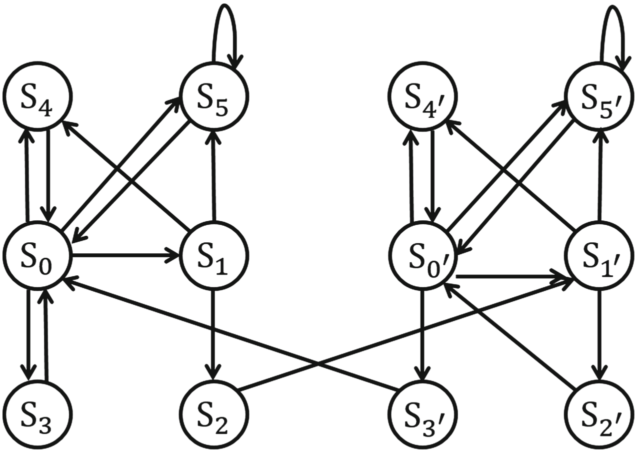

Fig. 4. The Markov chain for the flag-assisted ER-RRC

4 The Analytical Model

An analytical model for flag-assisted ER-RRC by the Markov chain in Fig. 4. UL data arrivals and DL data arrivals are assumed to form a Poisson process with rates λ_u and λ_d, respectively. In other words, the inter-arrival time of UL data, T_u, and the inter-arrival time of DL data, T_d, both follow the exponential distribution with mean $1/\lambda_u$ and $1/\lambda_d$, respectively. The UL data can be classified into three type. Type 1 with the probability α represents the data with RAI = 1 and *earlyRelease*. Type 2 with the probability β represents the data with RAI = 2. Type 3 with the probability γ represents the data without RAI value. Note that $\alpha + \beta + \gamma = 1$. The notations used in the analytical model are shown in Table 1.

In the Markov chain, states, S_i for $i \in [0, 5]$, mean the flag is off. If the device transports Type 1 data, the eNB will allow the device to use ER-RRC scheme. In contrast, states, $S_{i'}$ for $i \in [0, 5]$, mean the flag is on. The eNB makes the device use CP CIoT. States S_0, $S_{0'}$, S_1, and $S_{1'}$ are RRC_IDLE state. The device in S_0 and $S_{0'}$ has no buffered data at the MME while in S_1 and $S_{1'}$, it does have. The rest states, S_i and $S_{i'}$ for $i \in [2, 5]$ are RRC_CONNECTED state where the device wakes up for data transmission. S_2 and $S_{2'}$ denote that the device transmits Type 1 data before the buffered DL data is sent to it. S_3 and $S_{3'}$ denote that the device transmits Type 1 data without buffered data at the MME. S_4 and $S_{4'}$ denote that the device transmits Type 2 data. S_5 and $S_{5'}$ denote that the device transmits Type 3 data or it is paged to receive the buffered data.

4.1 Output Measures

We analyze the performance of flag-assisted ER-RRC by considering the power saving factor (denoted as ω) and the average latency of DL data (denoted as δ). The power saving factor is used to estimate the proportion of RRC_IDLE time in the entire system. Because a device staying more time in RRC_IDLE can save more power, ω should be as large as possible. However, the device in RRC_IDLE cannot receive DL data immediately, which causes DL latency. The target is to enhance the power saving factor but not to cause unaffordable latency.

The whole system time is expressed by $T = \sum_{i=0}^{5} (\pi_i H_i + \pi_{i'} H_{i'})$, where π_i ($\pi_{i'}$) is the stationary probability of the state S_i ($S_{i'}$) and H_i ($H_{i'}$) is the average state holding time of the state S_i ($S_{i'}$). The time that a device stays in RRC_IDLE in the entire system is $T_{idle} = \pi_0 H_0 + \pi_1 H_1 + \pi_{0'} H_{0'} + \pi_{1'} H_{1'}$. Evidently, $\omega = \frac{T_{idle}}{T}$. By the definition, $\delta = \frac{T_{latency}}{N}$, where $T_{latency}$ expressed by $\pi_1 H_1 + \pi_2 H_2 + \pi_{1'} H_{1'}$ is the accumulated latency of each DL data and N expressed by T/λ_d is the number of DL data in the entire system.

4.2 Stationary Probabilities

The transition probability from S_i to S_j ($S_{i'}$ to $S_{j'}$) is denoted as $p_{i,j}$ ($p_{i',j'}$). Take $p_{0,1}$ and $p_{0',1'}$ as an example, the device in S_0 ($S_{0'}$) transits to S_1 ($S_{1'}$) if the DL data arrive earlier than UL data. $p_{0,1}$ and $p_{0',1'}$ can be expressed by $\Pr[T_d < T_u]$. Then, $p_{0,1} = p_{0',1'} = \int_0^\infty \Pr[t_d < t_u | t_u = t] \Pr[t_u = t] dt = \frac{\lambda_d}{\lambda_u + \lambda_d}$. In the same way, we have $p_{0,3} = p_{0',3'} = \alpha \frac{\lambda_u}{\lambda_u \lambda_d}$, $p_{0,4} = p_{0',4'} = \beta \frac{\lambda_u}{\lambda_u \lambda_d}$, $p_{0,5} = p_{0',5'} = \gamma \frac{\lambda_u}{\lambda_u \lambda_d}$, $p_{1,2} = p_{1',2'} = \alpha(1 - e^{-\lambda_u T_{wb}})$, $p_{1,4} = p_{1',4'} = \beta(1 - e^{-\lambda_u T_{wb}})$, $p_{1,5} = p_{1',5'} = \gamma \frac{\lambda_u}{\lambda_u + \lambda_d} + e^{-\lambda_u T_{wb}}$, $p_{5,5} = p_{5',5'} = \frac{\gamma \lambda_u + \lambda_d}{\lambda_u + \lambda_d} \left(1 - e^{-(\lambda_u + \lambda_d) T_0}\right)$, and $p_{5,0} = p_{5',0'} = \frac{(\alpha + \beta) \lambda_u}{\lambda_u + \lambda_d} \left(1 - e^{-(\lambda_u + \lambda_d) T_0}\right) + e^{-(\lambda_u + \lambda_d) T_0}$. Since the device enters $S_{1'}$ when it is in S_2, $p_{2,1'} = 1$. Similarly, $p_{3,0} = p_{4,0} = p_{2',0'} = p_{3',0} = p_{4',0'} = 1$.

By the definition of stationary probability, $\pi_j = \sum_{S_i \in \mathbb{S}} p_{i,j} \pi_i$. If $S_i \in \mathbb{S}$, a device can transit from S_i to S_j directly. Because the stationary probability of each state can be expressed by a function of π_0 and $\sum_{i=0}^{5} (\pi_i + \pi_{i'}) = 1$, all the stationary probabilities can be solved.

4.3 State Holding Time

We assume that the UL or DL data arrive at the jth subframe after the device enters RRC_IDLE (i.e., S_0 and $S_{0'}$) with probability p_j. Then H_0 and $H_{0'}$ can be expressed by $\sum_{j=1}^{\infty} p_j \times j$, where $p_j = \int_{j-1}^{j} f(t, \lambda_u)[1 - F(t, \lambda_d)] dt + \int_{j-1}^{j} f(t, \lambda_d)[1 - F(t, \lambda_u)] dt$. $f(t, \lambda_u)$ ($f(t, \lambda_d)$) is the probability density function (PDF) of the UL (DL) data arrivals, and $F(t, \lambda_u)$ ($F(t, \lambda_d)$) is the cumulative distribution function (CDF) of the UL (DL) data arrivals. p_j can be reformulated as $e^{-(\lambda_u + \lambda_d)(j-1)} - e^{-(\lambda_u + \lambda_d) j}$, $j \in [1, \infty)$. Therefore, $H_0 = H_{0'} = \frac{1}{1 - e^{-(\lambda_u + \lambda_d)}}$.

The DL data arrival time in each eDRX cycle follows an uniform distribution because the time intervals between DL data follow an exponential distribution. On average, the data arrive at the midpoint of an eDRX cycle. That is when a device transits from S_0 to S_1 (or from $S_{0'}$ to $S_{1'}$), it will stay in S_1 ($S_{1'}$) for half of an eDRX cycle on average. Therefore, the average waiting time for the buffered data (denoted as T_{wb}) is $T_{eDRX}/2$, where T_{eDRX} is the length of an eDRX cycle. When a device is at S_1 (or $S_{1'}$), it will leave S_1 ($S_{1'}$) if either the device is paged to receive the buffered data or the device has UL data to transmit before paged. In the latter case, we assume the UL data arrive at the jth subframe after the device enters S_1 (or $S_{1'}$) with probability p_j. $p_j = \Pr[j-1 < T_u < j] = e^{-\lambda_u (j-1)} - e^{-\lambda_u j}, j \in [1, T_{wb}]$. Then, $H_1 = H_{1'} = \Pr[T_u > T_{wb}]T_{wb} + \sum_{j=1}^{T_{wb}} p_j \times j = \frac{1-e^{-\lambda_u T_{wb}}}{1-e^{-\lambda_u}}$.

The time a device spends in S_4 and $S_{4'}$ is three parts: (i) RRC establishment and bearer setup time (including sending the UL data) (denoted as T_{set}); (ii) the latency of the expected DL data (denoted as T_{ed}); (iii) RRC release time (denoted as T_{RR}). More specifically, T_{ed} is the device waiting time for the expected DL data after the device transmits its UL data. We assume T_{ed} follows exponential distribution with mean $1/\lambda_{ed}$. The calculation of part (ii) is the same as H_1 with substituting T_0 for T_{wb}. Thus $H_4 = H_{4'} = T_{set} + \frac{1-e^{-\lambda_{ed} T_0}}{1-e^{-\lambda_{ed}}} + T_{RR}$.

The time a device spends in S_5 and $S_{5'}$ contains T_{set} and four kinds of activities time (T_{RR} included). (1) Type 1 data arrival before T_0 expires. (2) Type 2 data arrival before T_0 expires. (3) Type 3 data or DL data arrival before T_0 expires. (4) no any data arrival. Let $H_5 = H_{5'} = T_{set} + H_{act}$. Note that in $S_{5'}$, since the device is in RRC_CONNECTED, it is supposed to be no buffered data at the MME. Therefore, the eNB informs the device to go back to RRC_IDLE earlier as the device is in case (1). The holding time for (1) is $\sum_{j=1}^{T_0} p_j (j + T_1 + T_{RR})$, where T_1 is the data transmission time. The holding time for (2) is $\sum_{k=1}^{T_0} p_k (k + \frac{1-e^{-\lambda_{ed} T_0}}{1-e^{-\lambda_{ed}}} + T_{RR})$. The holding time for (3) is $\sum_{l=1}^{T_0} p_l (l + H_{act})$. The holding time for (4) is $(\Pr[T_d > T_u > T_0] + \Pr[T_u > T_d > T_0])(T_0 + T_{RR})$. $p_j = \frac{\alpha\lambda_u}{\lambda_u+\lambda_d}(e^{-(\lambda_u+\lambda_d)(j-1)} - e^{-(\lambda_u+\lambda_d)j}), j \in [1, T_0]$. $p_k = \frac{\beta\lambda_u}{\lambda_u+\lambda_d}(e^{-(\lambda_u+\lambda_d)(k-1)} - e^{-(\lambda_u+\lambda_d)k}), k \in [1, T_0]$. $p_l = \frac{\gamma\lambda_u+\lambda_d}{\lambda_u+\lambda_d}(e^{-(\lambda_u+\lambda_d)(l-1)} - e^{-(\lambda_u+\lambda_d)l}), l \in [1,\, T_0]$. Referring to the reports [17,18], $T_1 = 0.005\,\mathrm{s}$. Then, we have $H_5 = H_{5'} = \big\{\frac{1-e^{-(\lambda_u+\lambda_d)T_0}}{1-e^{-(\lambda_u+\lambda_d)}} + e^{-(\lambda_u+\lambda_d)T_0} T_{RR} + \frac{\alpha\lambda_u}{\lambda_u+\lambda_d}(1 - e^{-(\lambda_u+\lambda_d)T_0})(T_1 + T_{RR}) + \frac{\beta\lambda_u}{\lambda_u+\lambda_d}(1 - e^{-(\lambda_u+\lambda_d)T_0})(\frac{1-e^{-\lambda_{ed}T_0}}{1-e^{-\lambda_{ed}}} + T_{RR})\big\} / \big\{1 - \frac{\gamma\lambda_u+\lambda_d}{\lambda_u+\lambda_d}(1 - e^{-(\lambda_u+\lambda_d)T_0})\big\}$.

In states S_2 and S_3, the device wakes up to transmits Type 1 data and goes back to RRC_IDLE. In states $S_{2'}$ and $S_{3'}$, the device has to wait for the MME checking if there are buffered data before going back to RRC_IDLE. Moreover, the device in $S_{2'}$ must spend time to receive the buffered data. Referring to [17,18], $H_2 = H_3 = 0.0335\,\mathrm{s}$, $H_{2'} = 0.0717\,\mathrm{s}$, and $H_{3'} = 0.0715\,\mathrm{s}$.

4.4 Validation

From above deviation, ω and δ can be obtained. The analytical model is validated against discrete event simulation experiments carried out in C++ based simulator. Table 2 shows that the analytical analysis is consistent with the simulation results.

Table 1. Notations used in analytical model

λ_u	UL data arrival rate
λ_d	DL data arrival rate
T	The whole system time
T_u	Inter-arrival time of UL data
T_d	Inter-arrival time of DL data
T_{idle}	Total time a device stays in RRC_IDLE in the entire system
$T_{latency}$	The accumulated latency of each DL data
T_{wb}	The average waiting time for the buffered data
T_{eDRX}	The length of an eDRX cycle
T_{ed}	The latency of the expected DL data.
T_{set}	RRC establishment and bearer setup time (including sending the UL data)
T_{RR}	RRC release time
T_0	The inactivity timer at the eNB
α	The probability of Type 1 UL data
β	The probability of Type 2 data
γ	The probability of Type 3 data
ω	The power saving factor
δ	The average latency of DL data
π_i $(\pi_{i'})$	The stationary probability of state S_i $(S_{i'})$
H_i $(H_{i'})$	The average state holding time of state S_i $(S_{i'})$
N	The number of DL data in the entire system
$p_{i,j}$ $(p_{i',j'})$	The transition probability from S_i to S_j $(S_{i'}$ to $S_{j'})$
$1/\lambda_{ed}$	The mean latency of expected DL data

Table 2. Validation of simulation and analytical models ($T_0 = 10\,\text{s}$, $T_{eDRX} = 10.24\,\text{s}$, and $1/\lambda_{ed} = 5\,\text{s}$)

$1/\lambda_d$	$10,000\,\text{s}$			
$1/\lambda_u$	$10,000\,\text{s}$		$100\,\text{s}$	
RAI setting	$\alpha = \beta = 0$ $\gamma = 1$	$\alpha = 0.4$ $\beta = \gamma = 0.3$	$\alpha = \beta = 0$ $\gamma = 1$	$\alpha = 0.4$ $\beta = \gamma = 0.3$
ω (Ana.)	0.997986	0.99856	0.894281	0.956778
ω (Sim.)	0.997988	0.99856	0.903243	0.957415
Error	0.00%	0.00%	1.00%	0.07%
δ (Ana.)	5.10577	5.10975	4.46125	4.86886
δ (Sim.)	5.10966	5.11149	4.46717	4.86010
Error	0.08%	0.03%	0.13%	0.18%

5 Performance Evaluation

In Fig. 5, the effect of eDRX cycle length on power saving factor and DL latency with three different data arrival rates is observed. Note that the devices set RAI to one for all UL data (i.e., $\alpha = 1$ and $\beta = \gamma = 0$) in this experiment since only RAI value is equal to one, the ER-RRC may be applied. The T_{eDRX} value starts from 10.24 s and the value is doubled each time up to 655.36 s.

Intuitively, when the cycle length becomes large, both power saving factor and DL latency increase. It means that the device spends more time to stay in RRC_IDLE to save more power while the DL data have to wait for longer time to be transmitted to the device. The DL latency is bounded to $1/\lambda_u$ because no matter how large the cycle length is, the DL data can be transmitted when the device has to transport the UL data.

However, it is noticeable that if the arrival rate is too small to consume power, increasing T_{eDRX} cannot well improve ω but makes δ worse. Instead, if the arrival rate is large, setting T_{eDRX} according to the delay budget can greatly enhance ω. Take $1/\lambda = 100$ s for example, if the delay budget is 60 s, T_{eDRX} can be set to 160 s.

Figure 6 shows the effect of the UL arrival rate on ω and δ. We compared flag-assisted ER-RRC scheme with CP CIoT and Always ER which means the eNB makes the device release the connection early every time when the device transmits the Type 1 data. In terms of ω, both flag-assisted ER-RRC and Always ER outperform CP CIoT, because the device in CP CIoT has to wait for the MME checking whether there are buffered data or not, which consumes more power. Although flag-assisted ER-RRC has well performance as Always ER do, δ in flag-assisted ER-RRC is smaller than that in Always ER. The latency is bounded under 5.12 s.

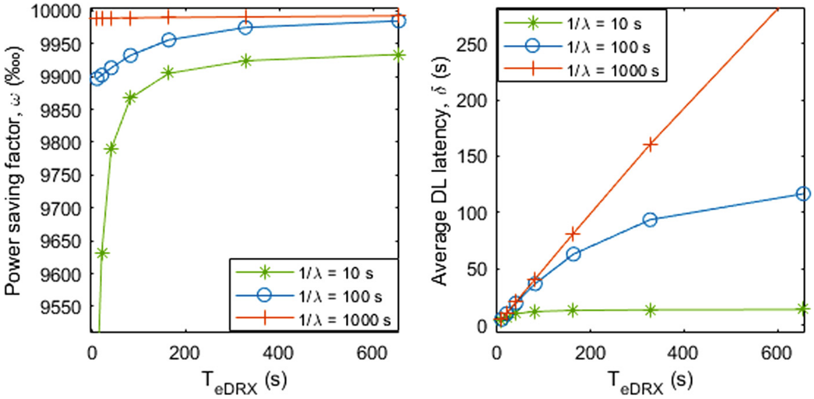

Fig. 5. The effect of eDRX cycle length on ω and δ ($\lambda_u = \lambda_d = \lambda$, $\alpha = 1$, and $T_0 = 1$)

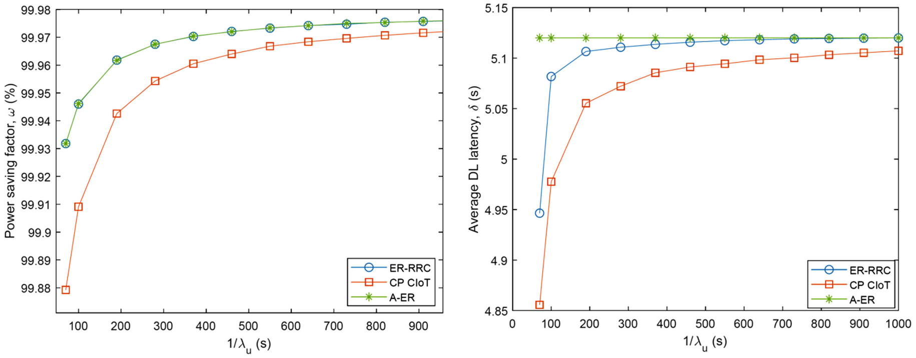

Fig. 6. The effect of UL arrival rate on ω and δ ($\alpha = 1$, $T_0 = 2\,\text{s}$, $1/\lambda_d = 10,000\,\text{s}$, and $T_{eDRX} = 10.24\,\text{s}$)

As a final remark, we designed a counter, instead of a flag, in the beginning and defined the threshold to decide whether to fall back to CP CIoT or not. In the experiments, we found that no matter how to set the parameters (i.e., T_0, T_{eDRX}, λ_u, and λ_d), the best outcome is to set the threshold to 1. Based on this result, a complex counter-based ER-RRC is simplified to a flag-assisted ER-RRC.

6 Conclusion

In this paper, the flag-assisted ER-RRC scheme is proposed to help IoT devices save more power. We proposed analytical and simulation models for flag-assisted ER-RRC, and compared with CP CIoT EPS Optimization and Always ER.

When the data arrival rate is small, it is better to set T_{eDRX} large to save power. Compared to CP CIoT, flag-assisted ER-RRC works better in terms of power saving factor by sacrificing some extra latency. The effect is quite obvious when $1/\lambda_u$ is less than $550\,\text{s}$.

Acknowledgment. This work was supported by the Center for Open Intelligent Connectivity through the Featured Areas Research Center Program within the Framework of the Higher Education Sprout Project by the Ministry of Education in Taiwan. This work was also sponsored by the R&D enhancement project "R&D of Network Behavior Security Analyses for IoT Devices on Advanced Edge Switch in an AIOT plus SDN Integrated Platform," which is executed by EstiNet Technologies Inc. and partially sponsored by Hsinchu Science Park Bureau, Ministry of Science and Technology, Taiwan, R.O.C. The work of M.-H. Tsai was supported in part by the MOST under Grant 107-2221-E-006-062 and Grant 108-2221-E-006-112, and in part by the Industrial Technology Research Institute.

References

1. GSA: Global progress to 5G - trials, deployments and launches (July 2018)
2. Cisco: Cisco visual networking index: forecast and trends, 2017–2022 white paper (February 2019)
3. Bujari, A., Furini, M., Mandreoli, F., Martoglia, R., Montangero, M., Ronzani, D.: Standards, security and business models: key challenges for the IoT scenario. Mob. Netw. Appl. **23**(1), 147–154 (2018)
4. 3GPP, TR 45.820 V13.1.0: Cellular system support for ultra low complexity and low throughput Internet of Things (November 2015)
5. 3GPP, TS 24.008 V15.3.0: Mobile radio interface layer 3 specification; Core network protocols; Stage 3 (2018)
6. 3GPP TS 23.401 V14.3.0: General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) (March 2017)
7. Andres-Maldonado, P., Ameigeiras, P., Prados-Garzon, J., Navarro-Ortiz, J., Lopez-Soler, J.M.: Narrowband IoT data transmission procedures for massive machine-type communications. IEEE Netw. **31**(6), 8–15 (2017)
8. 3GPP: NB-LTE - battery lifetime evaluation (2015)
9. Wang, K., Li, X., Ji, H.: Modeling 3GPP LTE advanced DRX mechanism under multimedia traffic. IEEE Commun. Lett. **18**(7), 1238–1241 (2014)
10. Wang, K., Li, X., Ji, H., Xiaojiang, D.: Modeling and optimizing the LTE discontinuous reception mechanism under self-similar traffic. IEEE Trans. Veh. Technol. **65**(7), 5595–5610 (2016)
11. Chang, H.-L., Tsai, M.-H.: Optimistic DRX for machine-type communications in LTE-A network. IEEE Access **6**, 9887–9897 (2018)
12. Chang, C.-W., Chen, J.-C.: Adjustable extended Discontinuous Reception (eDRX) cycle for idle-state users in LTE-A. IEEE Commun. Lett. **20**(11), 2288–2291 (2016)
13. Lee, J., Lee, J.: Prediction-based energy saving mechanism in 3GPP NB-IoT networks. Sensors **17**(9), 2008 (2017)
14. Xu, S., Liu, Y., Zhang, W.: Grouping-based discontinuous reception for massive narrowband Internet of Things systems. IEEE Internet Things J. **5**, 1561–1571 (2018)
15. 3GPP, TS 24.301 V13.12.0: Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS) (2018)
16. 3GPP, TS 36.304 V13.8.0: Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode (2017)
17. 3GPP, TR 25.912 V15.0.0: Feasibility study for evolved Universal Terrestrial Radio Access (UTRA) and Universal Terrestrial Radio Access Network (UTRAN) (2018)
18. Mohan, S., Kapoor, R., Mohanty, B.: Latency in HSPA data networks. Tech. rep., Qualcomm (2011)