



# Uplink Access Control in Narrowband IoT

Ren-Hung Hwang<sup>(✉)</sup>, Min-Chun Peng, and Bo-Hao Tu

National Chung Cheng University, Chiayi, Taiwan  
rhhwang@cs.ccu.edu.tw

**Abstract.** In order to provide low-power wide area network (LPWAN) services, 3GPP adopted the Narrow-Band Internet of Things (NB-IoT) standard in 2016. NB-IoT is expected to become the transmission communication standard for providing a large number of IoT devices in 5G networks. However, development of NB-IoT is still in its earlier stage and encounters several challenges. First, NB-IoT is designed for machine type communication. Generally, connection and transmission delays are not the primary consideration for this type of communication. Thus, it is not able to meet different delay requirements of different types of IoT applications. For example, for life-threatening or life-saving applications, they would require very high reliable and low latency transmission of emergency messages. Secondly, when a user equipment wants to associate to a NB-IoT network, it must synchronize with the regional base station (eNB) through the random access channel (RACH) procedure. A large number of IoT devices will cause a big challenge to the RACH procedure. Therefore, in this paper, we aim to improve the RACH procedure to handle a large number of IoT devices without affecting the transmission delay of emergent messages. We propose a Dynamic RACH Resource Allocation (DRRA) scheme which integrates with resource allocation scheme and Access Class Barring (ACB) scheme to improve the delay and throughput of the RACH procedure. Our simulation results show that the proposed DRRA scheme is able to achieve higher access success rate, higher system throughput, and low transmission delay for emergent message as compared to the original RACH procedure.

**Keywords:** NB-IoT · RACH · LPWAN · IoT

## 1 Introduction

Internet of Things (IoT) is one of the booming technologies in recent years. IoT industries, such as smart meter, smart grid, smart city, and healthcare, is expected to grow radiantly in next 10 years. Wireless communication technologies play a key role in meeting the flexible and ubiquitous communication demand of IoT devices. In particular, because the wide deployment scope and limited battery life of IoT devices, several Low-Power WAN (LPWAN) technologies have been developed to fit the demand of long distance but small data volume transmission. Among them, NB-IoT proposed by 3GPP becomes the most promising LPWAN technology due to its reliable infrastructure and licensed communication band [1].

In the NB-IoT network, devices need to perform a Random Access (RA) procedure [2] to communicate with an eNB before data transmission using preambles. There are two types of RA procedures, contention-based and contention-free. They use different set of preambles. The contention-based RA procedure is initiated by devices and used when the devices have demands for accessing the network. On the other hand, contention-free RA procedure is initiated by eNB and assign dedicated preamble to device directly. In most cases, devices need to perform contention-based RA procedure when first attached to an eNB. Only after the RA procedure, devices can achieve the uplink frequency synchronization and access the NB-IoT network. There are four steps in the contention-based RA procedure.

#### **Step1. Random Access Preamble (MSG1)**

The resource allocation in the frequency domain is a set of starting preambles. Preambles are carried in the Physical Random Access Channel (PRACH) which is a part of an uplink resource of a NB-IoT network. Each starting preamble is equivalent to the first NB-IoT PRACH (NPRACH) symbol group and associated with a specific 3.75 kHz tone. The set of starting preambles is determined by a subcarrier offset and a number of spanned subcarriers. The time-domain allocation is defined by a periodicity, a starting time with the period, and the number of repetitions associated with the NPRACH resource. Allocation of the PRACH in a subframe is determined by the prach-ConfigIndex of System Information Block 2 (SIB2) announced by an eNB [3]. Devices then randomly choose one preamble for contention-based RA procedure and send it to the eNB. If a device does not receive the Random Access Response (RAR) in a RAR window, its contention for RA is failed.

#### **Step2. Random Access Response (MSG2)**

When an eNB receives a preamble from a device, a RAR message is sent to the device. The RAR consists of the preamble index, uplink time synchronization, uplink grant, and Temporary Cell-Radio Network Temporary Identifier (TC-RNTI). Once a device succeeds in the RA procedure, its TC-RNTI is changed to C-RNTI. During this step, it is possible that more than one device selects the same preamble. In this situation, if the preamble can be received correctly, a RAR will be replied and the contention is solved in Step 4. However, due to the signal collision, the eNB is more likely not able to receive the preamble. In this study, we assume that when more than one device sends the same preamble, the RA will fail due to collision.

#### **Step3. Scheduled Transmission (MSG3)**

When a device receives a RAR, it replies with a MSG3 to the eNB. The uplink resource allocated in the RAR is used to transmit the MSG3, which consists of the device ID and the RRC connection request.

#### **Step4. Contention Resolution (MSG4)**

When an eNB receives a MSG3, it has to decide which devices succeed in the contention. The eNB transmits the MSG4 to devices that send the MSG3. The MSG4 consists of the device ID and the RRC connection setup. If a device finds that the device ID in the MSG4 is itself, its contention succeeds. The TC-RNTI of the device changes to C-RNTI and the status changes to RRC\_CONNECTED. Otherwise, the device knows its contention failed and starts the RA procedure all over again.

In order to achieve maximum energy loss of 144 dB, 154 dB, and 164 dB against Maximum Coupling Loss (MCL), there are three coverage levels in NB-IoT. The system information sets the specifications for each CE Level, such as the period in which the channel appears, the position of the channel on the frequency axis, the position of the channel on the time axis, the number of subcarriers used by the channel, and the maximum number of retries for the preamble. The base station will determine the threshold of the two reference signals (Reference Signal Received Power, RSRP). The transmission device will find out which CE level it is based on the measured RSRP. UE will perform random access procedure with corresponding NPRACH configuration.

Although in reality, there are many types of IoT devices and applications with very different traffic characteristics and QoS requirements, such as transmission delay and reliability, differentiating the RACH procedure for different types of IoT devices has received very little attention in the literature. In this study, we classify NB-IoT devices (applications) into two categories, the emergency transmission type and the regular transmission type. As the number of NB-IoT devices grow rapidly, it is a big challenge to guarantee the emergency transmission type device to connect to the eNB with a high successful rate and short delay. We aim to design the RACH procedure to handle a large number of IoT devices without affecting the transmission delay of emergent messages. We propose a Dynamic RACH Resource Allocation (DRRA) scheme which integrates with resource allocation scheme and ACB scheme to improve the delay and throughput of the RACH procedure. As compared to the enhanced RA procedure proposed by 3GPP [4], our simulation results show that the proposed DRRA scheme is able to achieve higher access success rate, higher system throughput, and extremely low transmission delay for emergent message.

The remainder of this paper is organized as follows. Section 2 surveys state-of-the-art works on NB-IoT RACH procedure. Section 3 presents the proposed Dynamic RACH Resource Allocation (DRRA), including the Resource Allocation scheme, and the Access Class Barring (ACB) scheme. Simulation results are shown in Sect. 4. Finally, conclusions and future research are given in Sect. 5.

## 2 Related Works

Surveys on recent advances in NB-IoT standards, key technologies and open issues can be found in [5–7]. Quite a few recent works on NB-IoT RACH procedure focused on efficient power consumption issues [8]. Performance modeling on the standard RACH procedure is done in [9]. The authors in [10] proposed a method to determine the required number of preambles for a target RA request arrival rate, but details of the RACH procedure, such as retransmission, were ignored. Preamble repetition was analyzed using stochastic geometry from physical signal aspect in [11] which showed little improvement in a heavy traffic scenario. Authors in [12] also showed that fewer repetitions with more retransmissions can yield higher successful RA probability. Thus, in this work, we will focus on the effect of retransmission.

In general, six basic mechanisms were proposed by 3GPP to enhance the RACH procedure [4], include ACB, separate RACH resource for MTC, dynamic allocation of

RACH resources, MTC specific backoff, slotted access, and pull based scheme. Most of the previous works studied how to improve the RACH procedure from one of the above mechanisms. Different from previous works, in this paper, we combine the first three mechanisms to enhance the RACH procedure with goals to provide large number of simultaneous IoT devices access and low delay transmission of emergency messages.

### 3 Dynamic RACH Resource Allocation (DRRA) Scheme

#### 3.1 System Overview

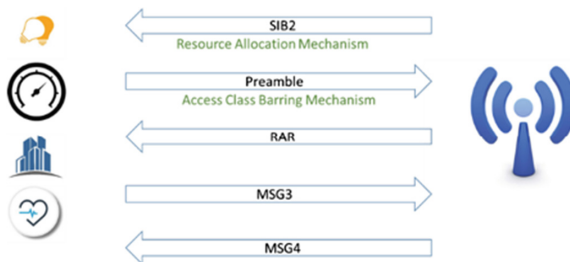
In this section, a system architecture overview is given of the proposed DRRA. The following assumptions are made.

- eNBs are aware of preamble collisions  
It is assumed that eNBs are aware of the collisions of the preambles when they happened, and will not reply RAR to those devices.
- Service types and classes  
It is assumed that a device collects the same service type of data. There are two service types with different delay and reliability requirements (see Table 1).

**Table 1.** Types of IoT devices (applications).

Type	Data type	Application
Type A	Emergent data	Life-threatening or life-saving applications (e.g., alarms)
Type B	Regular data	Environmental sensing data (e.g., smart meter)

Figure 1 shows the overview of the DRRA scheme. Before a device accesses a NB-IoT network, it receives a SIB2 from an eNB, which allocates the RACH resource based on the traffic condition in the previous frame. With the RACH resource allocation, the device attempts to transmit a preamble in a RA-slot. In a heavy traffic load condition, an ACB scheme is triggered. The device has to follow the ACB policy before sending a preamble. The ACB policy sets a probability and the device can only send a preamble according to this probability.



**Fig. 1.** Architecture of the DRRA scheme

### 3.2 RACH Resource Allocation Scheme

The DRRA scheme aims to control two types of the RACH resources. One is the number of preambles in a RA-slot, and the other is the number of RA-slots in a frame. They are adjusted to reflect the traffic load to guarantee high successful access probability.

In a NB-IoT network, an eNB broadcasts the prach-ConfigIndex in the SIB2, which then allows the eNB to adjust the number of RA-slots in a frame. In addition to the RACH resource adjustment, the eNB can also control the distribution of preambles for different service classes. As a result, the devices are aware of which preambles they can choose and when to begin the RA procedure.

Table 2 shows the notations used in the DRRA scheme. Arrival rates of Type A and Type B devices are assumed to follow Poisson process.

**Table 2.** Notations used in the DRRA scheme.

Notation	Meanings
$N_{Preamble}$	Number of contention-based preambles in a RA-Slot
$N_{TypeA}$	Number of preambles allocated for Type A devices in a RA-Slot
$N_{TypeB}$	Number of preambles allocated for Type B devices in a RA-Slot
$\hat{\lambda}_{TypeA}$	Estimated arrival rate of Type A devices per preamble in a frame
$\hat{\lambda}_{TypeB}$	Estimated arrival rate of Type B devices per preamble in a frame
$X_0$	No. of preambles for Type A devices which is not selected by any device
$X_1$	No. of preambles for Type A devices which is selected by only one device
$Y_0$	No. of preambles for Type B devices which is not selected by any device
$Y_1$	No. of preambles for Type B devices which is selected by only one device
$P_{b,TypeA}$	Estimated blocking probability of Type A devices (i.e., prob. of failed RA)
$P_{b,TypeB}$	Estimated blocking probability of Type B devices
$P_{b,TypeA}^G$	Guarantee blocking probability of Type A devices
$P_{b,TypeB}^G$	Guarantee blocking probability of Type B devices

In order to estimate the arrival rate of these two types of devices, the eNB keeps track the number of preambles that are not used by any devices ( $X_0$ ,  $Y_0$ ) and only one device ( $X_1$ ,  $Y_1$ ). Following the Poisson distribution, arrival rate per preamble of Type A (B) can be estimated from  $X_0(Y_0)$  and  $X_1(Y_1)$  by following equations, respectively.

$$e^{-\hat{\lambda}_{TypeA}} = \frac{X_0}{N_{TypeA}} \quad (1)$$

$$\hat{\lambda}_{TypeA} e^{-\hat{\lambda}_{TypeA}} = \frac{X_1}{N_{TypeA}} \quad (2)$$

$$e^{-\hat{\lambda}_{TypeB}} = \frac{Y_0}{N_{TypeB}} \quad (3)$$

$$\lambda_{\text{TypeB}} e^{-\lambda_{\text{TypeB}}} = \frac{Y_1}{N_{\text{TypeB}}} \quad (4)$$

$\lambda_{\text{TypeA}}$  can be derived either from (1) or (2). In this work,  $\lambda_{\text{TypeA}}$  is estimated by taking the average of the values from (1) and (2) if both of them yield valid values. Similarly,  $\lambda_{\text{TypeB}}$  is calculated in the same way. The blocking probabilities of Type A and Type B devices can then be estimated by using (5) and (6) where  $P_A = (\lambda_{\text{TypeA}}) e^{-(\lambda_{\text{TypeA}})}$  and  $P_B = (\lambda_{\text{TypeB}}) e^{-(\lambda_{\text{TypeB}})}$ .

$$P_{b,\text{TypeA}} = 1 - \frac{\sum_{i=1}^{N_{\text{TypeA}}} \binom{N_{\text{TypeA}}}{i} \times i \times P_A^i \times (1 - P_A)^{N_{\text{TypeA}}-i}}{\lambda_{\text{TypeA}}} \quad (5)$$

$$P_{b,\text{TypeB}} = 1 - \frac{\sum_{i=1}^{N_{\text{TypeB}}} \binom{N_{\text{TypeB}}}{i} \times i \times P_B^i \times (1 - P_B)^{N_{\text{TypeB}}-i}}{\lambda_{\text{TypeB}}} \quad (6)$$

They are compared with the predefined guaranteed blocking probabilities of  $P_{b,\text{TypeA}}^G$  and  $P_{b,\text{TypeB}}^G$  to determine as to whether the traffic load is too heavy or not. If the traffic load is too heavy, the RACH resource and preamble distribution are adjusted to give priority to guarantee the blocking probability of type A devices. The new values of  $N'_{\text{TypeA}}$ , and  $N'_{\text{TypeB}}$  will be used to calculate the updated arrival rates (per preamble),  $\lambda'_{\text{TypeA}}$  and  $\lambda'_{\text{TypeB}}$ , based on Eqs. (7) and (8).

$$\lambda'_{\text{TypeA}} = \frac{\lambda_{\text{TypeA}} * N_{\text{TypeA}}}{N'_{\text{TypeA}}} \quad (7)$$

$$\lambda'_{\text{TypeB}} = \frac{\lambda_{\text{TypeB}} * N_{\text{TypeB}}}{N'_{\text{TypeB}}} \quad (8)$$

With the new arrival rates and number of allocated preambles, new blocking probability is calculated again using Eqs. (5) and (6). This process is repeated until suitable  $N'_{\text{TypeA}}$ , and  $N'_{\text{TypeB}}$  are found. Algorithm 1 shows the pseudo code of the RACH Resource Allocation Scheme. In Algorithm 1, when traffic load is low, lines (4)–(9) reallocate preambles to both types of devices according to their arrival rates. On the other hand, if the traffic load is high, lines (10)–(17) increases the preambles allocated to type A traffic first which guarantees the transmission of emergent messages. Finally, lines (19)–(30) adjusts the number of preambles allocated to type B under the constraint of available preamble left over. Notably, if the traffic load is extremely high, we will run out of preambles and cannot guarantee the probability of success probability of RA for either type. In this case, we will need to apply Access Class Barring

Scheme proposed in the next section to restrain the arrival rate of IoT devices. In the future work, we will also consider the option of adding RACH resources on non-anchor carriers.

---

**Algorithm 1** RACH Resource Allocation Scheme

---

**input**  $X_0, X_1, Y_0, Y_1, N_{TypeA}, N_{TypeB}, P_{b,TypeA}^G, P_{b,TypeB}^G$   
**output new**  $N_{TypeA}, N_{TypeB}$

**1: Begin**  
**2:** Calculate  $\lambda_{TypeA}$  and  $\lambda_{TypeB}$  based on  $X_0 \cdot X_1$  by (1),(2), (3),(4)  
**3:** Calculate  $P_{b,TypeA}$  and  $P_{b,TypeB}$  based on (5), (6)  
**4: if**  $P_{b,TypeA} \leq P_{b,TypeA}^G$  &&  $P_{b,TypeB} \leq P_{b,TypeB}^G$  // traffic load is light  
**5:**  $\alpha = \lambda_{TypeA} / (\lambda_{TypeA} + \lambda_{TypeB})$  //proportional share the extra RACH resources  
**6:**  $N_{TypeA} = N_{TypeA} + \text{floor}(\alpha \times (N_{Preamble} - N_{TypeA} - N_{TypeB}))$  //floor function  
**7:**  $N_{TypeB} = N_{Preamble} - N_{TypeA}$  and exit the algorithm  
**8: else if**  $P_{b,TypeA} > P_{b,TypeA}^G$  // give priority to increase  $N_{TypeA}$   
**9:** Find the smallest  $N'_{TypeA}$  such that  $N_{TypeA} \leq N'_{TypeA} \leq N_{Preamble}$  and  $P'_{b,TypeA} \leq P_{b,TypeA}^G$  where  $\lambda'_{TypeA}$  and  $P'_{b,TypeA}$  are calculated by (7) and (5) (use  $\lambda'_{TypeA}$ )  
**10: if**  $P'_{b,TypeA} > P_{b,TypeA}^G$  when  $N'_{TypeA} = N_{Preamble}$  //not sufficient preambles  
**11:**  $N_{TypeA} = N_{Preamble}$  and  $N_{TypeB} = 0$  and exit the algorithm  
**12: else**  
**13:**  $N_{TypeA} = N'_{TypeA}$   
**14: endif**  
**15: endif**  
**16: // final check with  $N_{TypeB}$**   
**17: if**  $P_{b,TypeB} \leq P_{b,TypeB}^G$  &&  $N_{TypeB} \leq (N_{Preamble} - N_{TypeA})$  //reduce  $N_{TypeB}$   
**18: repeat steps 5-7**  
**19: else //increase  $N_{TypeB}$  under the constraint that  $N_{TypeB} \leq (N_{Preamble} - N_{TypeA})$**   
**20:** Find the smallest  $N'_{TypeB}$  such that  $N'_{TypeB} \leq (N_{Preamble} - N_{TypeA})$  and  $P'_{b,TypeB} \leq P_{b,TypeB}^G$  where  $\lambda'_{TypeB}$  and  $P'_{b,TypeB}$  are calculated by (8) and (6)  
**21: if**  $P'_{b,TypeB} > P_{b,TypeB}^G$  when  $N'_{TypeB} = (N_{Preamble} - N_{TypeA})$  //not sufficient preambles  
**22:**  $N_{TypeB} = (N_{Preamble} - N_{TypeA})$   
**23: else**  
**24:**  $N_{TypeB} = N'_{TypeB}$   
**25: repeat steps 5-7**  
**26: endif**  
**27: endif**  
**28: end //end of algorithm**

---

### 3.3 Access Class Barring (ACB) Scheme

The ACB scheme is adopted to restrain the RA requests. Since we expect the number of type A devices will be relatively small and they provide life-saving applications, the proposed ACB scheme only applies to the type B devices. In the ACB scheme, an ACB barring factor is set to control devices to perform the RA procedure. A device generates a random number and if the number higher than the ACB barring factor, it has to give up this trial and restart the RA procedure later.

Next, the detail steps of the proposed ACB scheme are presented. Initially, the number of RA requests in a frame sent from type B devices is  $\lambda_{TypeB} * N_{TypeB}$ . After setting up an ACB barring factor, the arrival rate of type B devices is reduced to  $(\lambda_{TypeB} \cdot \rho_{ACB})$  and the blocking probability can then be calculated by Eq. (6).

The goal of the ACB scheme is to find a suitable ACB barring factor such that the number of RA requests is expected to lower than the system capacity, which is one arrival per preamble. Thus, given current per preamble arrival rate of type B device,  $\lambda_{TypeB}$ , the ACB barring factor is set according to following equation.

$$\rho_{ACB} = \begin{cases} \frac{1}{\lambda_{TypeB}}, & \text{if } \lambda_{TypeB} > 1 \\ 1, & \text{if } \lambda_{TypeB} \leq 1 \end{cases} \quad (9)$$

## 4 Performance Evaluation

Performance is evaluated via simulations which are written in C language.

### 4.1 Simulation Parameters and Performance Metrics

We follow the same parameter settings proposed in [6], as shown Table 3.

**Table 3.** Simulation parameter setting.

CE level	Subcarrier	Repetition	Periodicity
CE0	48	1	40 ms
CE1	24	2	80 ms
CE2	24	4	160 ms

$P_{b,TypeA}^G$  is set to 0.01 and  $P_{b,TypeA}^G$  is set to 0.1 unless otherwise stated. Simulation time is 10 min and each simulation is performed 30 times to obtained average performance metric and 95% confidence interval. The confidence interval is very small, thus not shown in following results.

The following metrics are used to evaluate the performance of a RACH scheme.

- Access success probability: it is the average successful probability of a RA request per RACH period.
- Average access delay: it is the average access delay given a RA request is successful.
- Average throughput: it is the average number of successful RA requests per RACH period.

In the following simulation experiments, we will compare the performance of three schemes: the original scheme proposed in 3GPP NB-IoT without ACB, the ACB scheme proposed in [13] which is designed to cope with massive IoT devices, and our DRRA scheme.



## 4.2 Simulation Case 1: Single CE Level

In the first simulation case, we valid the basic features of the DRRA scheme. We assume there are only CE level 0 devices, and the MaxTry parameter is set to 5. The MaxTry parameter defines how many times a device will retry the RA procedure after failures. Preamble is randomly reselected when a device performs a retry. The arrival rate of CE level 0 devices varies from 10 to 30. For each arrival rate, half of them are type A devices.

Figure 2 shows the average access probability of three schemes. When arrival rate is higher than 20, the system becomes congested. As we can observed from Fig. 2, ACB scheme outperforms the other two schemes when traffic load is light. The rationale is that the ACB scheme restrain the arrival rate even the system is not overloaded. Figure 3 confirms this situation as the average throughput of the ACB scheme is lower than the other two schemes. When the system becomes overloaded with RA requests, the proposed DRRA outperforms the other two schemes both in access success probability and average throughput. This confirms that the proposed DRRA scheme is able to adapt to traffic load better than the other two schemes. Notably, even when the arrival rate is 30, the DRRA scheme is still able to yield a very high throughput.

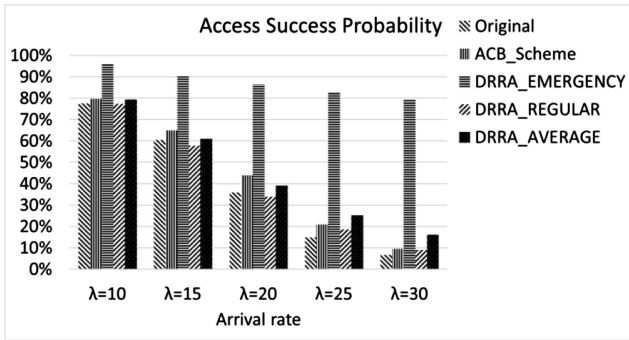


Fig. 2. Average access success probability of three schemes

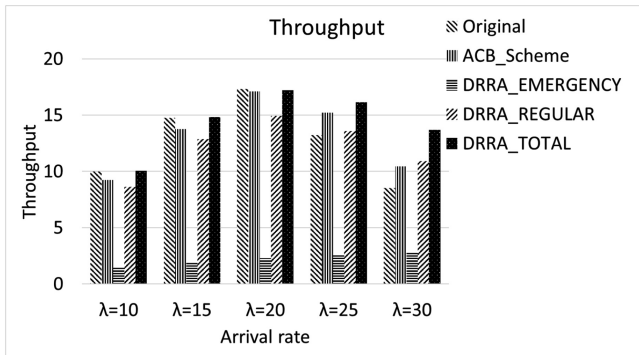


Fig. 3. Average throughput of three schemes

Figure 4 compares the average access delay of three schemes. As the DRRA scheme gives priority to type A devices, emergent data is able to be sent in a very short delay. This shows the superior feature of the proposed DRRA scheme. For type B devices, when the traffic load is high, the DRRA scheme also yields smaller delay than the other two schemes.

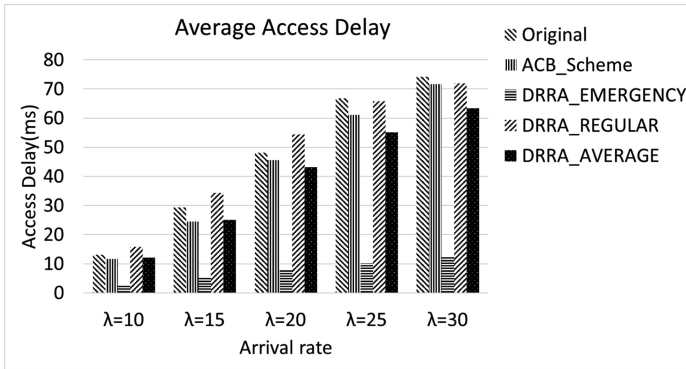


Fig. 4. Average access delay of three schemes

### 4.3 Simulation Case 2: Three CE Levels

In the second simulation case, we simulate IoT devices distributed in 3 CE levels. Five sets of arrival rates are simulated, they are (11, 3.5, 2.5), (13, 4, 3), (15, 4.5, 3.25), (17, 5, 3.5), (19, 5.5, 4), where three numbers in each set are the RA request arrival rates of devices of three CE levels, respectively. In this simulation, we show that the MaxTry affects the performance of the RACH procedure significantly. On one hand, increase the MaxTry will increase the chances of RA trials. On the other hand, increase the MaxTry will also increase the traffic load which results in higher chance of collision and unsuccess access probability.

Figure 5 shows the effect of MaxTry on the average throughput of devices of CE level 0. As we can see that the throughput increases as the MaxTry increases initially. However, when the MaxTry is too large, e.g., 7, the throughput decreases as the MaxTry increases, especially for scenario 5 where the traffic load is high.

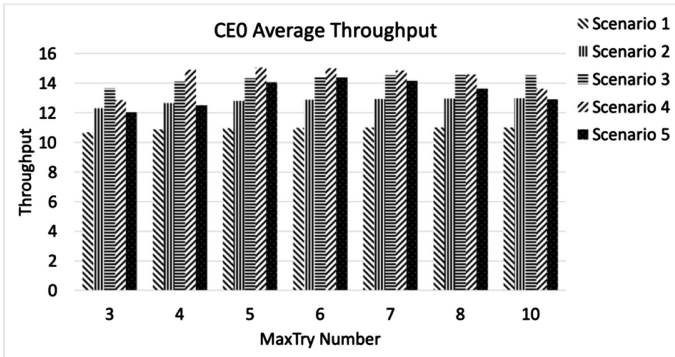


Fig. 5. Effect of MaxTry on throughput of CE level 0 devices

When the RA request of a CE level 0 device failed after MaxTry times, the device will change its CE level to 1 and continue the RACH procedure. If it failed on this level again, it will increase its CE level to 2. Figures 6 and 7 show the effect of MaxTry on the average throughput of CE level 1 and 2, respectively. The MaxTry affects the throughput of these two levels significantly. Only a small number of MaxTry can increase the throughput of CE level 1. For CE level 2, the average throughput decreases as the MaxTry increases, for most of the scenarios.

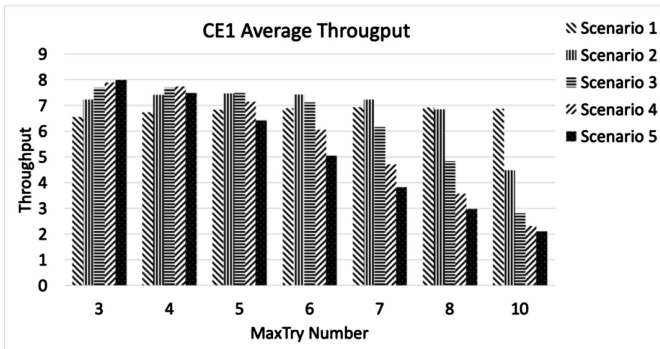


Fig. 6. Effect of MaxTry on throughput of CE level 1 devices

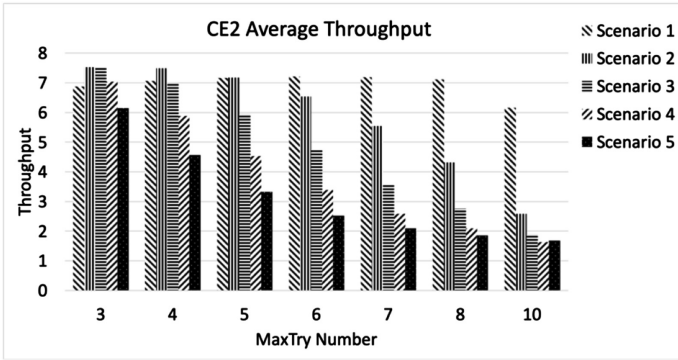


Fig. 7. Effect of MaxTry on throughput of CE level 2 devices

## 5 Conclusion and Future Works

In this paper, we proposed a Dynamic RACH Resource Allocation (DRRA) scheme which integrates with resource allocation scheme and ACB scheme to improve the performance of the RACH procedure. The DRRA scheme classify devices into two types and gives priority to devices that need to send emergent messages, e.g., life-saving alarm messages. Our simulation results show that the DRRA scheme can cope with large RA request rates and yield better throughput and delay than existing schemes. In particular, even under extremely high RA request rate, emergent devices can still have high throughput and low delay.

Several mechanisms can be integrated into the DRRA scheme and requires further study. When the RACH resources of anchor carrier are not enough to accommodate the RA requests, allocating RACH resources on non-anchor carrier has been proposed by 3GPP [5] which requires further investigation. Adjusting backoff timer can also alleviate the bursty traffic load. Repetition number is similar to the MaxTry, may increase access success probability, but consume more resources. More importantly, if collision happens, no matter how many repetitions will not help to avoid collision. Thus, how to perform repetition also requires further study. Finally, we are also investigating the effect of different traffic arrival models, such as MMPP which can capture more bursty arrival pattern. We believe the proposed DRRA can cope with MMPP arrival process better than other schemes.

## References

1. Popli, S., Jha, R.K., Jain, S.: A survey on energy efficient narrowband internet of things (NB-IoT): architecture, application and challenges. *IEEE Access* **7**, 16739–16776 (2019)
2. 3GPP: NB-IoT—Random access design. 3GPP TSG-RAN1 #83 R1-157424, Ericsson, Stockholm, Sweden (2015)
3. 3GPP: Evolved universal terrestrial radio access; radio resource control protocol. 3GPP TS 36.331 specification v13.4.0 (2017)

4. 3GPP. RAN improvements for machine-type communications. 3GPP TR 37.868 v11.0.0 (2011)
5. Ratasuk, R., Mangalvedhe, N., Xiong, Z., Robert, M., Bhatoolaul, D.: Enhancements of narrowband IoT in 3GPP Rel-14 and Rel-15. In: IEEE Conference on Standards for Communications and Networking (CSCN). IEEE, Helsinki (2017)
6. Wang, Y.-P.E., et al.: A primer on 3GPP narrowband internet of things. *IEEE Commun. Mag.* **55**(3), 127–133 (2017)
7. Feltrin, L., Tsoukaneri, G., Condoluci, M., Buratti, C., Mahmoodi, T., Dohler, M.: Narrowband IoT: a survey on downlink and uplink perspectives. *IEEE Wirel. Commun.* **26**(1), 78–86 (2019)
8. Bello, H., Jian, X., Wei, Y., Chen, M.: Energy-delay evaluation and optimization for NB-IoT PSM with periodic uplink reporting. *IEEE Access* **7**, 3074–3081 (2019)
9. Harwahu, R., Cheng, R.-G., Wei, C.-H., Sari, R.F.: Optimization of random access channel in NB-IoT. *IEEE Internet Things J.* **5**(1), 391–402 (2018)
10. Azari, A., Hossain, M.I., Markendahl, J.I.: RACH dimensioning for reliable MTC over cellular networks. In: Vehicular Technology Conference, IEEE, Sydney (2017)
11. Jiang, N., Deng, Y., Condoluci, M., Guo, W., Nallanathan, A., Dohler, M.: RACH preamble repetition in NB-IoT network. *IEEE Commun. Lett.* **22**(6), 1244–1247 (2018)
12. Harwahu, R., Cheng, R.-G., Tsai, W.-J., Hwang, J.-K., Bianchi, G.: Repetitions versus retransmissions: trade-off in configuring NB-IoT random access channels. *IEEE Internet Things J.* **6**(2), 3796–3805 (2019)
13. Leyva-Mayorga, I., Rodriguez-Hernandez, M.A., Pla, V., Martinez-Bauset, J., Tello-Oquendo, L.: Adaptive access Type Barring for efficient mMTC. *Comput. Netw.* **149**, 252–264 (2019)