

# Collision Reduction for Machine Type Communications in LTE-A Networks

Jihun Moon<sup>1</sup> and Yujin Lim<sup>2,\*</sup>

<sup>1</sup>*Department of Computer Science, University of Suwon*

<sup>2</sup>*Department of Information Technology Engineering, Sookmyung Women's University*

<sup>1</sup>*kakasio@suwon.ac.kr, <sup>2</sup>yujin91@sookmyung.ac.kr*

## Abstract

*LTE-A networks provide excellent wireless connectivity for MTC (Machine Type Communication). In order to connect a large number of MTC devices into one network, many signaling messages should be transmitted. The signaling channel is overloaded for the messages and accesses of MTC devices are failed. 3GPP specifies access class barring scheme to resolve the overload problem but leaves a specific algorithm to configure a parameter as an implementation issue. In this paper, we present an algorithm to dynamically configure a parameter based on the number of colliding devices in access class barring scheme. Through simulations, it is shown that our algorithm improves the performance compared to the original access class barring in terms of access success, access failure, collision, and access delay.*

**Keywords:** *Access class barring, Collision reduction, LTE-A, Machine-type communications, Overload problem.*

## 1. Introduction

In MTC (Machine Type Communication), devices are able to autonomously exchange information with no human intervention [1, 2]. MTC applications support automated remote monitoring such as smart metering, mobile healthcare, and intelligent transport system. Recently, with the widespread introduction of LTE-A (Long Term Evolution – Advanced), migration of MTC devices to LTE-A is under investigation by many cellular operators. Since a large number of MTC devices attempt to access a wireless channel within a short period of time, signaling messages sharply increase and congestion is an inevitable consequence. To guarantee a system performance, the congestion should be controlled [3].

Once a device is activated to transmit data, it tries to reserve RACHs (Random Access Channels). When a huge number of devices try to access RACHs simultaneously, signaling flow increases tremendously. Particularly, because data transmissions for automated remote monitoring are synchronized by duty cycles, it causes a severe signaling congestion. The congestion increases packet collision and access delay and decreases a service quality of MTC applications. To tackle the congestion problem, ACB (Access Class Barring) and EAB (Extended Access Barring) schemes are presented by 3GPP (3rd Generation Partnership Project) [4]. In ACB, eNB (evolved Node-B) periodically broadcasts a barring factor ( $0 \leq p \leq 1$ ) and a barring duration. A device attempting RACH access selects a random number between 0 and 1. If the selected number is equal to or greater than the barring factor, the access attempt is postponed for the barring duration. When the value of  $p$  is high, the contention between the unbarred devices increases and the packet collision

---

\* Corresponding Author: Yujin Lim

increases. When the value of  $p$  is low, the contention between unbarred devices decreases but the time which the barred devices finally succeed in their random access increases. In EAB, devices are categorized into several ACs (Access Classes). One of ACs is allocated to each device. The eNB periodically broadcasts a barring bitmap that identifies which ACs are permitted to access. A device attempting a random access checks the received bitmap. If the AC of the device is not permitted to access, the device is barred and will not initiate the random access procedure until the bitmap is updated.

In ACB and EAB, a specific algorithm needs to configure the operating parameters like barring factor or barring bitmap. Even though many researchers try to propose specific algorithms, it is still open issue for further research [3]. In this paper, we present an algorithm to dynamically configure the barring factor based on the number of colliding devices in ACB. The experimental results show that our algorithm configures the barring factor dynamically according to the number of colliding devices. Besides the algorithm improves the system performance compared to the original scheme in terms of access success, access failure, collision, and access delay.

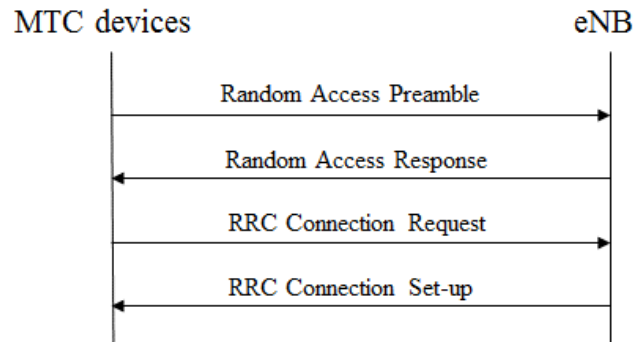
## 2. Related Work

A device attempting RACH access starts to a random access procedure in a predefined time resource (we call it 'time slot'). The frequency resources are composed of  $M$  orthogonal preambles in a time slot. The time and frequency resources are indicated as random access opportunity. If more than one devices use the same preamble to access RACH in a time slot, a collision happens. The colliding device tries again with a randomly chosen preamble in a new slot. The trial for random access is repeated until the maximum number of preamble transmission is reached.

A device attempting a random access follows a random access procedure to reserve resource in random access opportunity as shown in Figure 1. First, the device randomly selects a preamble sequence from a predefined set of preamble sequences. It sends the chosen preamble to allow eNB to estimate the transmission timing of the device. Then, when eNB receives the preamble which was transmitted by the device, it replies with RAR (Random Access Response) message to adjust the transmission timing of the device. If more than one devices transmitted the same preamble in the first step, they will receive the RAR messages. Third, the device to be received the RAR message sends RRC (Radio Resource Control) request and scheduling request. If more than one devices sent the same preamble in the first step, they transmit the same RRC and scheduling requests in this step. Finally, eNB transmits a contention-resolution message. If the collision occurs and eNB decodes one of collided packets, eNB acknowledges the device. Unacknowledged devices fail the random access. The failed devices attempt again with newly chosen preambles in another time slot based on a uniform back-off algorithm.

Many RAN (Random Access Network) congestion control mechanisms have been proposed. In [5], the random access attempts from devices are rejected selectively according to the average admission rate. It estimates the reject probability based on the traffic load measuring by a proportional integrative derivative controller. In [6], a prioritized random access with dynamic access barring is described. It pre-allocates channel resources for devices with class-dependent back-off procedures in order to prevent a large number of simultaneous random access requests. Among the proposed mechanisms, ACB-based mechanism is known as simple and efficient scheme to distribute the access attempts. In [7], eNB calculates the packet transmission probability based on the traffic load. Using the probability, eNB selects

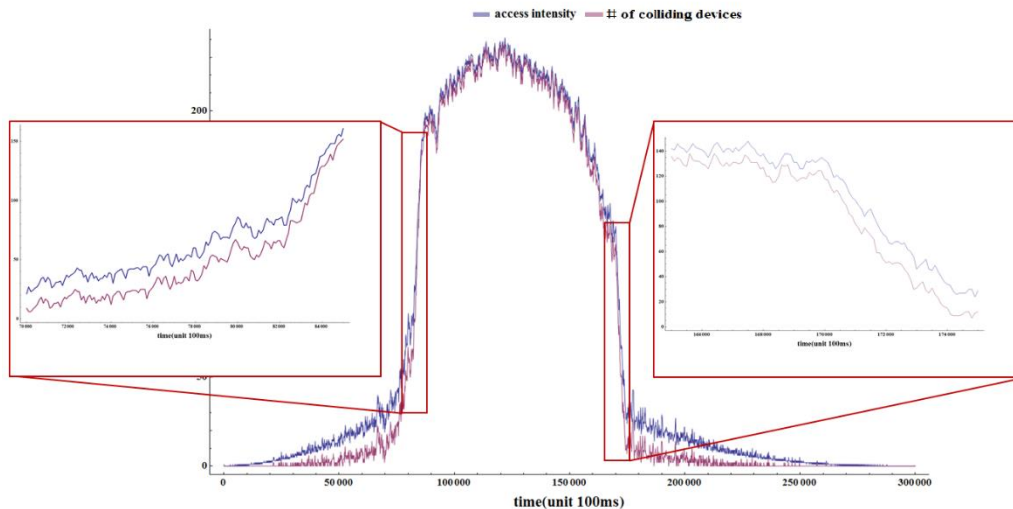
the optimal value of the barring factor. In [8], it presents an analytical model to estimate the expected total access delay of all the devices. It needs the knowledge of the number of backlogged devices to calculate the optimal value of the barring factor. In [9], cooperative mechanism is proposed to control the congestion among multiple cells. The eNBs cooperate to control the barring factor in each cell for global stabilization and access load sharing.



**Figure 1. The Random Access Procedure for MTC Communications**

### 3. Proposed Algorithm

We present a dynamic algorithm to configure a barring factor for ACB. We have motivated that the variance in access intensity does not match exactly the variance in collision, as shown in Figure 2 [10]. The access intensity is the number of random access attempts during a time period. When the access intensity starts to increase, the number of colliding devices increases lately. Besides, the access intensity starts to decrease, the number of colliding devices decrease early. Thus, it is reasonable that the barring factor is controlled based on the variance in collision, not the variance in access intensity.



**Figure 2. The Curve of Access Intensity and Preamble Collision**

In order to model the variance in the number of colliding devices, we adopt the Gaussian distribution which is often used to represent continuous random variables whose distributions are not known [11]. The barring factor in  $i$ -th time slot is adapted by

$$p_i = \frac{1}{e^{\varphi_i \log AI_{max} AI_i}} \quad (1)$$

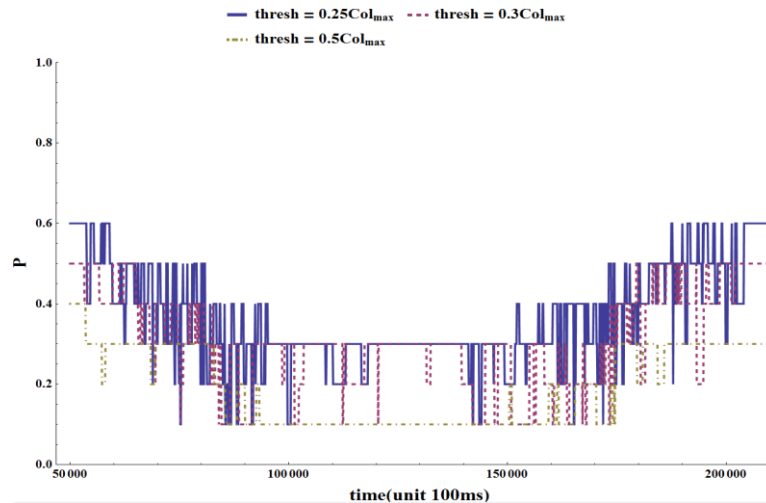
where  $AI_i$  is the access intensity in  $i$ -th time slot and  $AI_{max}$  is the maximum of the access intensity which the application allows. However, in the control of the barring factor based on the Gaussian distribution, the barring factor is changed too swiftly. Thus, we scale the barring factor according to the difference between the number of colliding devices in adjacent time slots as below.

$$\varphi_i = \frac{|Col_i - Col_{i-1}|}{\sigma_{Col}^2} \quad (2)$$

where  $Col_i$  indicates the number of colliding devices in  $i$ -th time slot. The  $\sigma_{Col}^2$  is the variance of the number of colliding devices over a sliding window of length  $k$ . The  $k$  is the operating parameter of our algorithm. We have considered the difference between the number of colliding devices in adjacent time slots. Thus, even though the collision is high, if the difference between the values in adjacent time slots is low, the barring parameter is set to high value. To solve the problem, we set the barring factor by

$$p_i = \begin{cases} p_{i-1} & \text{if } |Col_i - Col_{i-1}| \leq thresh \\ \frac{1}{e^{\varphi_i \log AI_{max} AI_i}} & \text{else} \end{cases} \quad (3)$$

where  $thresh$  is the operating parameter and we set it by using the maximum number of the colliding devices in a time period, i.e.,  $Col_{max}$ . Figure 3 shows the curve of the barring factor with varying  $thresh$ . When the  $thresh$  decreases,  $p$  is set to high value. As shown in the figure, the value of  $p$  with  $thresh = 0.5Col_{max}$  is adapted the most dynamically according to the variance in access intensity.



**Figure 3. The Values of Barring Factor by Varying  $thresh$**

#### 4. Experimental Results

To evaluate the performance of our algorithm, we adopt the load for smart electric metering application which has considered the household density in central and urban areas of London [1]. It is representative example of MTC applications with tight synchronization. In the application, the access intensity generated from synchronized smart meters in urban London is 35,670 households/cell. As well as the access intensity, the frequency of meter-reading reports considerably influences the system load. The

reporting frequency ranges from 5 mins to 24 hours. We set it to 5 mins. The simulation parameters for the RACH of LTE networks are shown in Table 1.

**Table 1. Simulation Parameters for RACH in LTE FDD**

Parameter	Setting
Cell bandwidth	5 MHz
PRACH configuration index	6
Total number of preambles	64
Maximum number of preamble transmission	10
Number of UL grants per RAR	3
Number of CCEs allocated for PDCCH	16
Number of CCEs per PDCCH	4
Ra-ResponseWindowSize	5 subframes
mac-ContentionResolutionTimer	48 subframes
Backoff Indicator	20ms

The  $N$  devices in a cell attempt to transmit data to eNB during a period time  $T$ . The random access intensity follows distribution  $p(t)$ . The number of devices which generate the access requests in the  $i$ -th time slot is given by

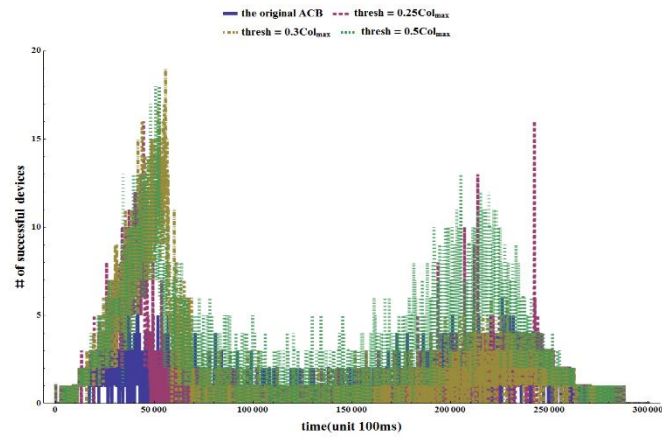
$$AI(i) = N \int_{t_i}^{t_i+1} p(t) dt \quad (4)$$

where  $t_i$  is the  $i$ -th time slot and  $p(t)$  follows the Beta distribution with the parameters  $\alpha = 3$  and  $\beta = 4$ . The  $p(t)$  is given by

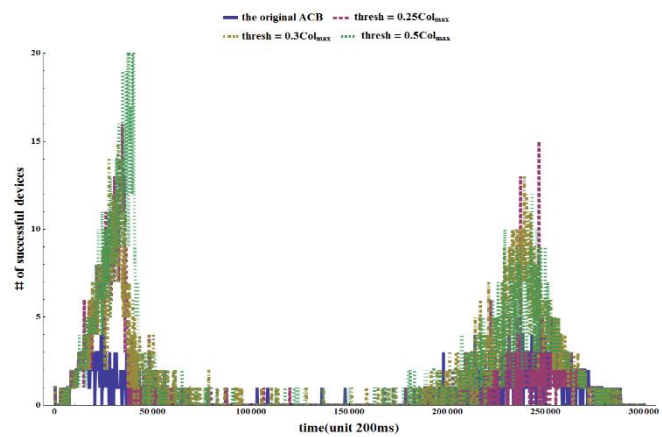
$$p(t) = \frac{t^{\alpha}(T-t)^{\beta-1}}{T^{\alpha+\beta-1}Beta(\alpha,\beta)} \quad (5)$$

where  $Beta(\alpha, \beta)$  indicates the Beta function.

In [12], a mechanism has been proposed to determine when congestion control for the random access is enabled or disabled. The eNB calculates the congestion coefficient which is the difference between 1 and the ratio of the number of RAR sent to the number of preamble transmissions in a certain time duration. When the congestion coefficient exceeds a certain threshold, e.g., 0.4 in a moving window, the congestion control is activated. On the contrary, the control is deactivated when the congestion coefficient goes below the threshold. In our experiments, we adopt the mechanism and set the threshold to 0.4.



(a) time slot = 100ms

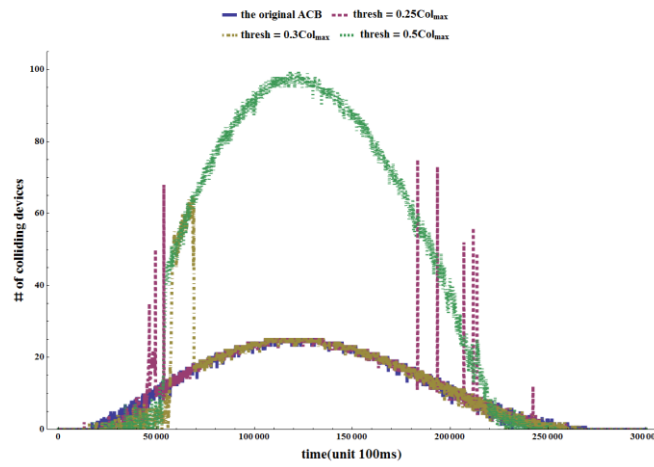


(b) time slot = 200ms

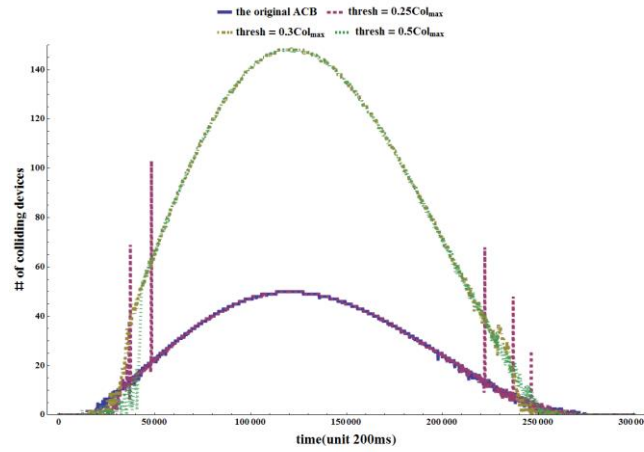
**Figure 4. The Performance Comparison in the Number of Successful Devices**

Figure 4 shows the number of devices that successfully access a random channel in each time slot. For experiments, we set the operating parameter  $k$  in (2) to 10 moving windows. The parameter  $thresh$  in (3) is set to  $0.5Col_{max}$ ,  $0.3Col_{max}$ , and  $0.25Col_{max}$ . For the performance comparison, we use the original ACB scheme. In the scheme, when the congestion control is activated, the barring factor is set to 0.1 until the control is deactivated. In the time slot with 100ms, the number of successful devices in our algorithm is about 2 times better than that of the original ACB, on average. In the time slot with 200ms, the number of successful devices in our algorithm is about 1.5 times better than that of the original ACB, on average. Figure 5 shows the number of the colliding devices in each time slot. The number of colliding devices in our algorithm is about 2 times larger than that of the original ACB. In the original ACB, the barring factor is set to 0.1 when ACB scheme is activated and the many devices are barred. Thus the collision between unbarred devices is reduced. However, even though the collision decreases, it is not mean that the performance is improved. Figure 6 shows the number of devices that finally fail to access a random channel even though the random access trial reaches the maximum number of preamble transmission. In the time slot with 100ms, the number of failed devices in our algorithm is about 20% smaller than that of the original ACB. In the time slot with 200ms, our algorithm is about 10% better than that of the original ACB. This shows that our algorithm manipulate the barring factor dynamically according to the variance in collision. As the result, the number of successful

devices increases and the number of failed devices. With varying *thresh*, as the value of *thresh* increases, the performance is improved. Even though the collision is high, if the value of *thresh* is too low, *p* is set to high value and the contention increases.

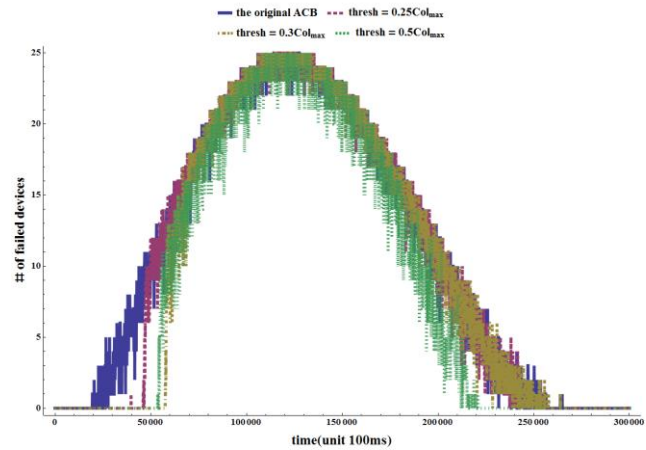


(a) time slot = 100ms

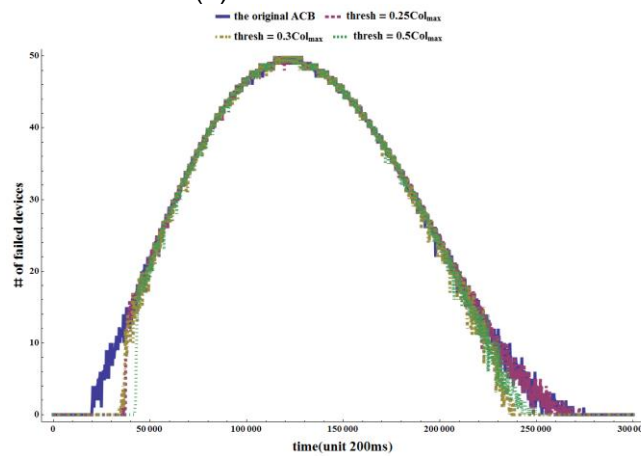


(b) time slot = 200ms

**Figure 5. The Performance Comparison in the Number of Colliding Devices**



(a) time slot = 100ms

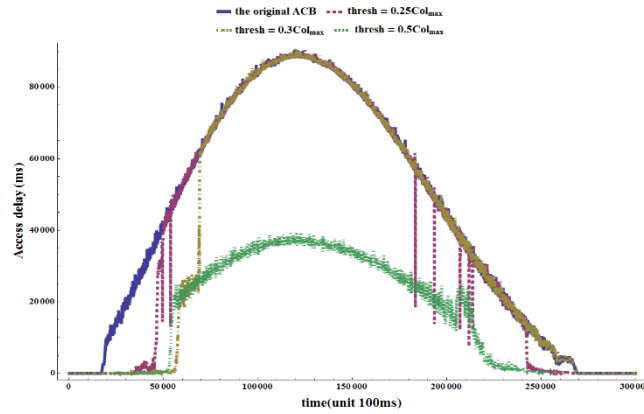


(b) time slot = 200ms

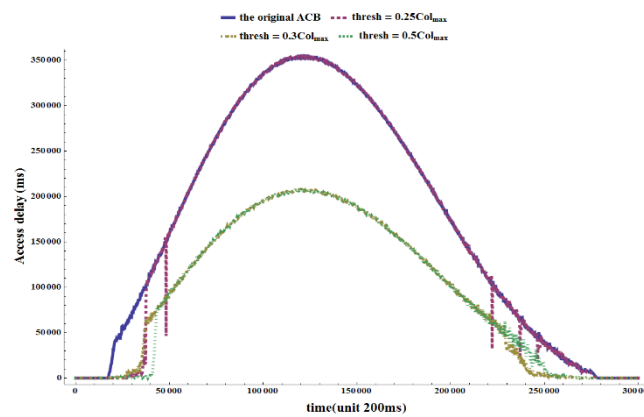
**Figure 6. The Performance Comparison in the Number of Failed Devices**

Figure 7 shows the access delay that indicates the time between the first random access attempt and the completion of the random access procedure for the successful devices. The access delay of our algorithm is about 30% or 40% better than that of the original ACB in 100ms or 200ms time slot, respectively. Because the value of  $p$  is too low in the original ACB, the number of barred devices and the number of repetitions for random access procedure increases. Thus, the access delay also increases. With varying  $thresh$ , the access delay with  $0.5Col_{max}$  is about 45% better than these with  $0.3Col_{max}$  and  $0.25Col_{max}$  in 100ms time slot. In 200ms time slot, the access delay with  $0.5Col_{max}$  is about 35% better than other cases. When the value of  $thresh$  is too low, the number of failed devices increases and the access delay also increases.





(a) time slot = 100ms



(a) time slot = 200ms

**Figure 7. The Performance Comparison in the Access Delay**

## 5. Conclusion

In this paper, we have proposed a control algorithm that adapts the barring factor according to the variance in collision. Until now researchers have proposed many methods to control the barring factor in ACB based on the access intensity. However, it is not the variance in access intensity immediately influences the system performance. Thus, we measure the number of colliding devices at each time slot and track the variance in the number of colliding devices. Based on the variance, the barring factor is configured dynamically. Experimental results shows that our algorithm is superior to the original ACB in terms of the numbers of successful devices and failed devices, and access delay, because our algorithm can control dynamically the barring factor according to the variance in the number of colliding devices

## Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2015R1D1A1A09057141)

## References

- [1] 3rd Generation Partnership Project, "Technical Specification Group Radio Access Network; Study on RAN Improvements for Machine Type Communications," 3GPP TR 37.868 V11.0.0, (2011).
- [2] 3rd Generation Partnership Project, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol Specification," 3GPP TS 36.331 V12.7.0, (2015).
- [3] M.-Y. Cheng, G.-Y. Lin, H.-Y. Wei and A.C.-C. Hsu, "Overload Control for Machine-Type Communications in LTE-Advanced System," IEEE Communications Magazine, vol. 50, no. 6, (2012), pp. 38-45.
- [4] P. Jain, P. Hedman and H. Zisimopoulos, "Machine Type Communications in 3GPP Systems," IEEE Communications Magazine, vol. 50, no. 11, (2012), pp. 28-35.
- [5] A. Ksentini, Y. Hadjadj-Aoul and T. Taleb, "Cellular-based Machine-to-Machine: Overload Control," IEEE Network, vol. 26, no. 6, (2012), pp. 54-60.
- [6] T.-M. Lin, C.-H. Lee and W.-T. Chen, "PRADA: Prioritized Random Access with Dynamic Access Barring for MTC in 3GPP LTE-A Networks," IEEE Transactions on Vehicular Technology, vol. 63, no. 5, (2014), pp. 2467-2472.
- [7] G. Wang, X. Zhong, S. Mei and J. Wang, "An Adaptive Medium Access Control Mechanism for Cellular based Machine to Machine (M2M) Communication," Proceedings of the IEEE International Conference on Wireless Information Technology and Systems (ICWITS), Honolulu, US, (2010) August 28-September 3.
- [8] S. Duan, V. Shah-Mansouri and V.W.S. Wong, "Dynamic Access Class Barring for M2M Communications in LTE Networks," Proceedings of the IEEE Global Communications Conference (GLOBECOM), Atlanta, US, (2013) December 9-13.
- [9] S.-Y. Lien, T.-H. Liau, C.-Y. Kao and K.-C. Chen, "Cooperative Access Class Barring for Machine-to-Machine Communications," IEEE Transactions on Wireless Communications, vol. 11, no. 1, (2012), pp. 27-32.
- [10] J. Moon and Y. Lim, "The Access Control of MTC Devices with Random Access Channel," Proceedings of the Applied Science and Engineering for Better Human Life, Jeju Island, Korea, (2016) August 16-19.
- [11] T.M. Cover and J.A. Thomas, "Elements of Information Theory", John Wiley and Sons, New Jersey, (2006).
- [12] Intel Corporation, "Further Performance Evaluation of EAB Information Update Mechanisms," 3GPP R2-120270, RAN WG2 Meeting #77 (2012).

## Authors



**Jihun Moon**, he is a master's degree candidate in Department of Computer Science at University of Suwon, Korea. He received his bachelor's degree in Department of Information Media at University of Suwon, Korea in 2015. Since 2015, he has been working as a researcher in Center for U-city Security and Surveillance Technology (CUSST). His research interests include IoT and communication systems.



**Yujin Lim**, she is an associate professor with Department of Information Technology Engineering at Sookmyung Women's University, Korea. She received her Ph.D. degrees in Computer Science from Sookmyung Women's University, Korea and Tohoku University, Japan in 2000 and 2013 respectively. She received her B.S. and M.S. degrees in Computer Science from Sookmyung Women's University in 1995 and 1997. From 2004 to 2015, she worked as an associate professor in Department of Information Media in University of Suwon. Her research interests include IoT, multi-agent system, and artificial intelligence.