Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 6474768, 7 pages
https://doi.org/10.1155/2017/6474768

Research Article

A Reinforcement Learning Approach to Access Management in Wireless Cellular Networks

Jihun Moon[1] **and Yujin Lim**[2]

[1] *Department of Computer Science, University of Suwon, San 2-2, Wau-ri, Bongdam-eup, Hwaseong, Gyeonggi-do 445-743, Republic of Korea*
[2] *Department of Information Technology Engineering, Sookmyung Women's University, 100 Cheongpa-ro 47-gil, Yongsan-gu, Seoul 04310, Republic of Korea*

Correspondence should be addressed to Yujin Lim; yujin91@sookmyung.ac.kr

Received 25 February 2017; Accepted 9 April 2017; Published 14 May 2017

Academic Editor: Syed Hassan Ahmed

In smart city applications, huge numbers of devices need to be connected in an autonomous manner. 3rd Generation Partnership Project (3GPP) specifies that Machine Type Communication (MTC) should be used to handle data transmission among a large number of devices. However, the data transmission rates are highly variable, and this brings about a congestion problem. To tackle this problem, the use of Access Class Barring (ACB) is recommended to restrict the number of access attempts allowed in data transmission by utilizing strategic parameters. In this paper, we model the problem of determining the strategic parameters with a reinforcement learning algorithm. In our model, the system evolves to minimize both the collision rate and the access delay. The experimental results show that our scheme improves system performance in terms of the access success rate, the failure rate, the collision rate, and the access delay.

1. Introduction

In smart city applications, many smart and mobile devices connect with one another and operate in an adaptive manner. The devices generate relevant city data in bursts and in unexpected manners on a massive scale. Because Long Term Evolution-Advanced (LTE-A) cellular networks provide wide coverage and low latency, LTE-A is considered to be one of the most promising communication infrastructures for smart city applications. However, radio resources in this communication infrastructure are too limited to serve a large amount of data. 3rd Generation Partnership Project (3GPP) specifies that Machine Type Communication (MTC) should be used to handle the congestion problem caused by small amounts of data being transmitted from a large number of devices within a short period of time [1–3]. MTC traffic includes periodic-update traffic and event-driven traffic. Event-driven triggering brings about bursts and unpredictable traffic flows, and these add to the congestion problem [4].

When a device has data to send, it needs to synchronize with a base station, that is, Evolved Node B (eNB), and to reserve a Random Access Channel (RACH). The RACH is a sequence of physical radio resources (RA slots). To do this, a device follows a four-step random access procedure [5, 6]. First, a device with data to send selects a random access preamble randomly as a digital signature from a predefined set of preambles and sends the preamble to eNB. Then, eNB responds with a Random Access Response (RAR) message to synchronize subsequent uplink transmission. However, if multiple devices send the same preamble in the same RA slot at the first step, a collision occurs, and they will receive the same RAR message. After receiving the RAR message, a device sends a connection request message along with a scheduling request. Finally, eNB acknowledges the connection request message. If a device receives the acknowledgement message successfully, it proceeds to data transmission. If a device encounters a preamble collision, it does not receive the message from eNB and will initiate a new RA procedure after a fixed backoff time. When the number of unsuccessful attempts of a device reaches the predefined maximum value (*Maximum Number of Preamble Transmission* [1]), the device finally fails the RA process.

As the number of devices attempting random access in the same RA slot increases, the numbers of preamble collisions and access delays increase as well. The access delay is the time between the generation of access request and the completion of the random access procedure. As a result of such a delay, the congestion becomes heavy. 3GPP specifies the use of the Access Class Barring (ACB) scheme to tackle the congestion problem [7]. ACB is a well-known scheme that restricts RA attempts. ACB operates on two strategic parameters: the barring factor and the barring duration. Based on the current congestion status, eNB regulates the RA attempts of MTC devices using these two parameters. Thus, the control of the parameters is vital in protecting the system from the excessive connectivity of a large number of devices. However, 3GPP does not specify how to control the parameters dynamically.

In this paper, we model the problem of determining ACB parameter values by using a reinforcement learning algorithm. This algorithm is able to follow unexpected changes and traffic bursts rapidly. Through the learning algorithm, we propose a scheme for dynamically and autonomously controlling the parameters. The experimental results show that the use of our scheme is sufficient to resolve the congestion problem in terms of access success rate, failure rate, collision rate, and access delay.

The rest of the paper is organized as follows. In Section 2, we discuss related studies. In Section 3, we propose an access management scheme that uses a reinforcement learning algorithm. In Sections 4 and 5, we evaluate the performance of our scheme and conclude the paper with our plans for future research.

2. Related Work

Several proposals for tackling the RA congestion problem are discussed. In ACB [7], eNB broadcasts the barring factor ($0 \leq p \leq 1$) and barring duration to its cell based on the current congestion level. A device with data to send generates a random number ($0 \leq q \leq 1$). If $q \leq p$, the device gets permission to access RACH. Otherwise, the RA attempt is barred for the barring duration. However, there is tradeoff with respect to barring factor p. If severe congestion occurs in a cell, eNB sets p to an extremely low value and most devices are barred. This results in an unacceptable access delay. On the other hand, if eNB sets p to an extremely high value, most of the preambles encounter collisions. This results in unacceptable data transmission failure. Thus, the barring factor is an important factor in determining system performance.

3GPP specifies the use of Extended ACB (EAB) as well as ACB. In EAB [8], devices are grouped into a set of ACs (Access Classes). eNB broadcasts a barring bitmap for the ACs periodically. A device with data to send compares its AC with the bitmap. If the bit that corresponds to the AC of the device is set, the device is barred from transmitting data until the bit changes. In this case, the scheduling policy among the ACs is a factor determining system performance. In current cellular networks, eNB alone determines the ACB barring factor to stabilize each cell. In [9], a cooperative mechanism is proposed to control congestion globally over multiple cells. The barring factor of each eNB is decided cooperatively among all eNBs. This is done for global stabilization and for access load sharing.

In order to maintain a high service quality for HTC (Human Type Communication), 3GPP specifies the use of two different schemes: the MTC specific backoff scheme and the separate RA resources scheme [10]. In the MTC specific backoff scheme, a dedicated backoff parameter is set for the MTC devices. The backoff scheme discourages the devices to attempt random access for certain duration of time. The backoff value for HTC devices is shorter than it is for MTC devices. In the separate RA resources scheme, RA slots are allocated to HTC and MTC devices separately. Both of the schemes focus on reducing the impact of RACH congestion on HTC devices. Thus, MTC devices may experience serious congestion because the amount of resources available is reduced.

In addition to the solutions specified by 3GPP, various other congestion solutions are proposed. In [11], a congestion-aware admission control scheme is proposed. It rejects RA requests from MTC devices selectively according to the congestion level, which is directly induced from the incoming packet processing delay at the application layer. In [12], RACH resources are preallocated to different MTC classes using class-dependent backoff procedures to prevent a large number of simultaneous RACH access attempts. Dynamic access barring according to the traffic load level is proposed for collision avoidance. Under this barring, the access attempts of devices transmitting for the first time are delayed.

3. Proposed Scheme

To tackle the congestion problem, we adopt Q-learning (QL) algorithm. The algorithm utilizes a form of reinforcement learning to solve Markovian decision problems without possessing complete information [13, 14]. Because QL finds solutions through the experience of interacting with an environment, we use it to model the ACB barring factor [15]. In other words, we control the ACB barring factor adaptively with QL.

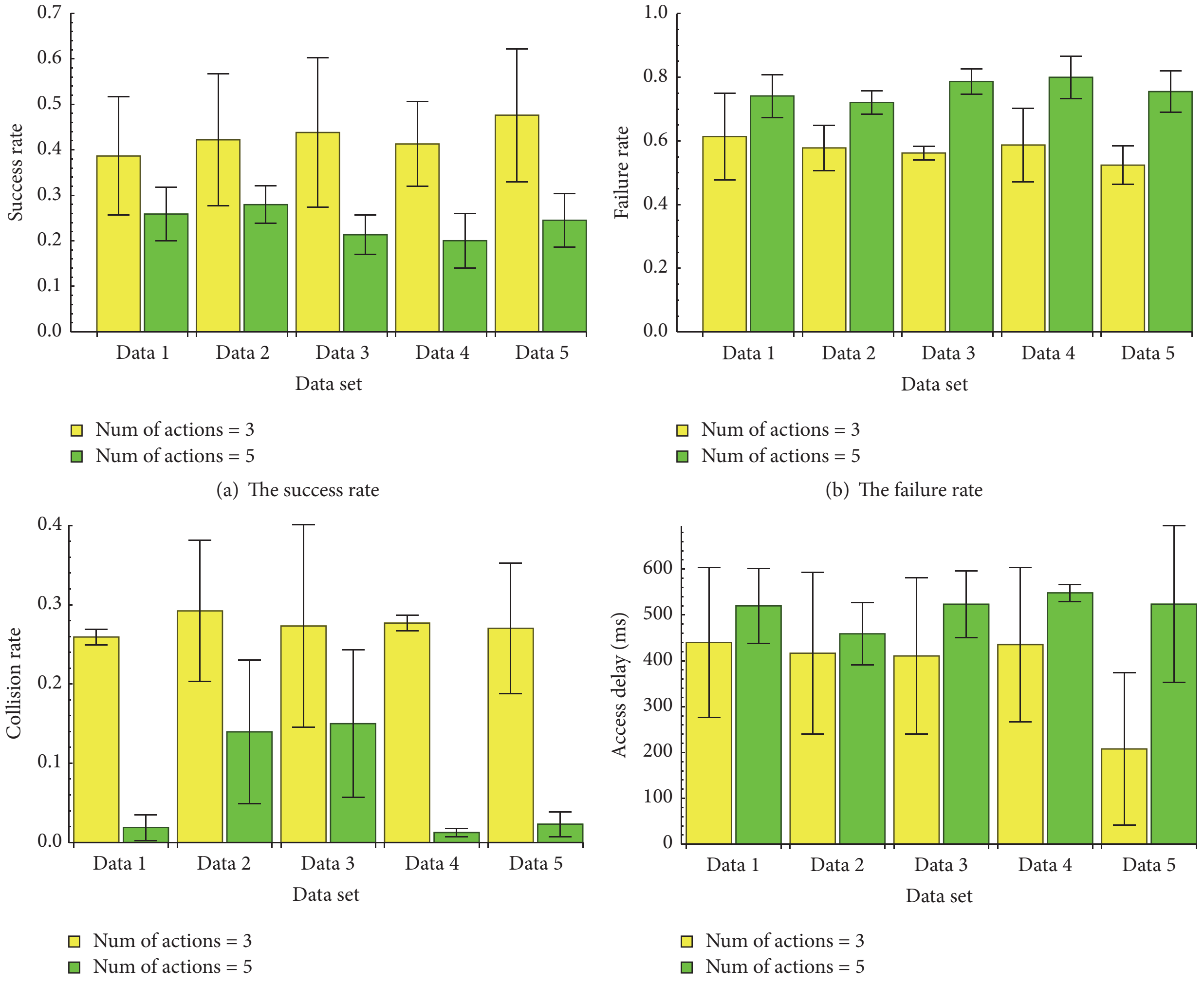
Let S denote a finite set of possible environment states and let A denote a finite set of admissible actions to be taken. At RA slot t, eNB perceives the current state $s_t = s \in S$ of the environment and takes an action $a_t = a \in A$ based on both the perceived state and its past experience. The action a_t changes the environmental state from s_t to $s_{t+1} = s' \in S$. When that happens, the system receives the reward r_t.

The goal of the QL algorithm is to find an optimal policy for state s that optimizes the rewards over the long run. The algorithm estimates the Q-value $Q(s,a)$ as the cumulative discounted reward. Using the Q-values, the algorithm finds the optimal Q-value $Q^*(s,a)$ in a greedy manner. The Q-value is updated as

$$Q(s,a) = Q(s,a) + \alpha \cdot \Delta Q(s,a), \tag{1}$$

where α ($0 \leq \alpha \leq 1$) is the learning rate. When α is 0, the Q-value is not updated. When α is a high value, learning occurs quickly, as in

$$\Delta Q(s,a) = \left\{ r + \gamma \cdot \max_{a' \in A} Q(s',a') \right\} - Q(s,a), \tag{2}$$



(a) The success rate

(b) The failure rate

(c) The collision rate

(d) The access delay

Figure 1: A performance comparison for various number of admissible actions.

where γ $(0 \leq \gamma \leq 1)$ is the discount factor that weighs immediate rewards more heavily than future rewards.

In this paper, we model a QL algorithm to control the barring factor in order to minimize both the number of RACH collisions and the access delay. A collision occurs when two or more devices transmit the same preamble in the same time slot. We define a set of possible states, a set of admissible actions, and the rewards. First, we define the access success rate R_t^{succ} $(0 \leq R_t^{\text{succ}} \leq 1)$ as a set with states S. The access success rate is defined as the number of devices that successfully access RACH divided by the number of devices contending in a given RA slot. R_t^{succ} is divided evenly into $|S|$ states. Each state s has three possible actions: increasing or decreasing the p value by $\delta_i \in \Delta$ or maintaining the current p value. The Δ indicates a finite set of unit values for p. To balance the exploration and exploitation of learning, an ϵ-greedy method [16] is applied to our QL algorithm. In other words, we can select a random action with probability ϵ or we can select an action with probability $1-\epsilon$ that gives an optimal $Q(s,a)$ in the state s. We define the reward in order to minimize the collision rate (R_t^{col}, $0 < R_t^{\text{col}} < 1$) and the access delay (delay_t, $0 < \text{delay}_t < \text{delay}_{\max}$). R_t^{col} represents the number of colliding devices divided by the number of contending devices. The reward given when action a is taken at state s in RA slot t is

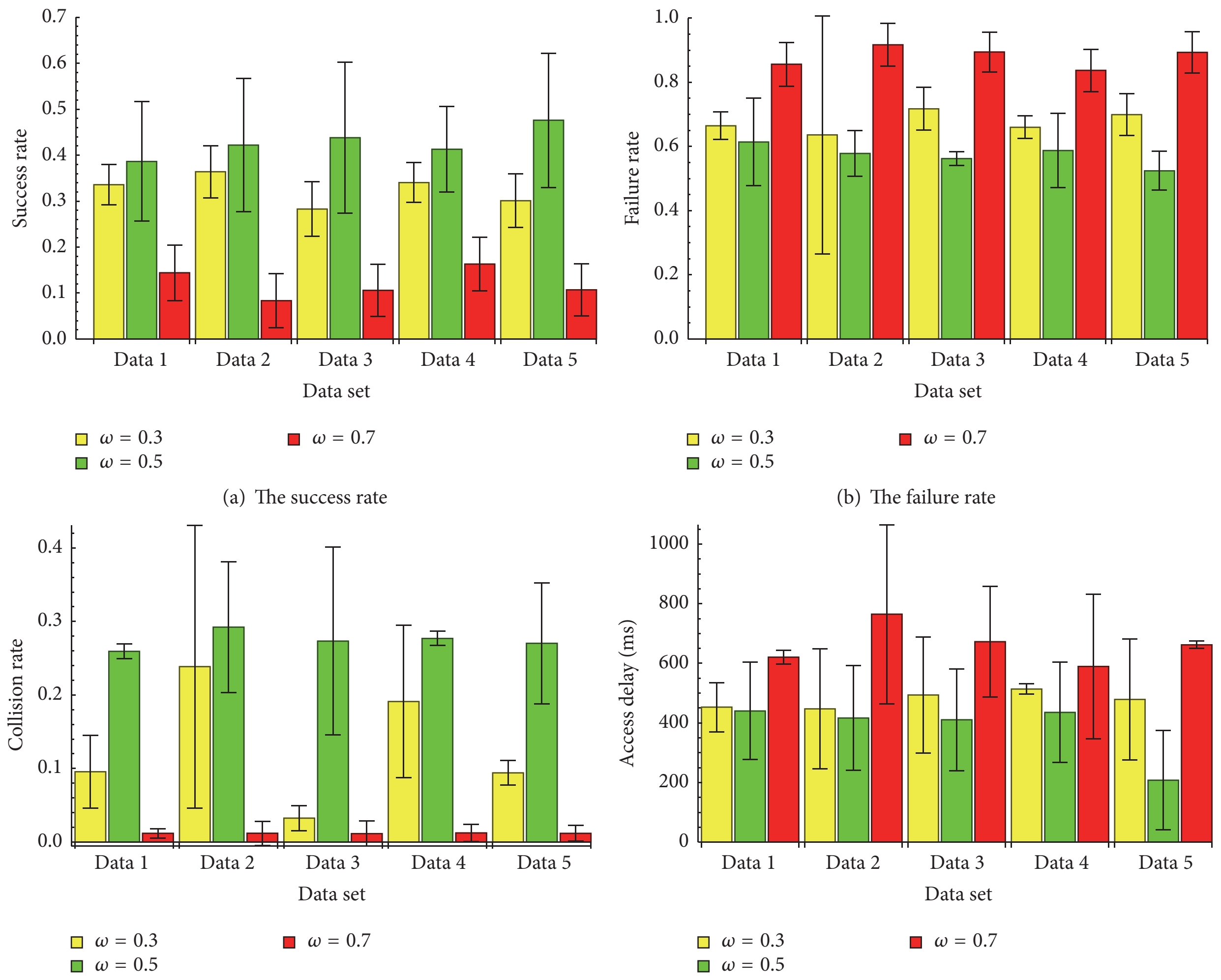
$$r_t(s,a) = \omega \cdot \frac{1}{R_t^{\text{col}}} + (1-\omega)\frac{\text{delay}_{\max}}{\text{delay}_t}, \qquad (3)$$

where $\text{delay}_{\max}$ is the maximum access delay that the system allows and ω is a smoothing factor $(0 \leq \omega \leq 1)$.

4. Performance Analysis

In this paper, we extend the work of our previous study [15] and evaluate the performance of our access management scheme in terms of access success rate, collision rate, failure rate, and access delay.

We adopted the traffic model for a smart metering application as an experimental scenario in which a large



(a) The success rate

(b) The failure rate

(c) The collision rate

(d) The access delay

FIGURE 2: A performance comparison for various values of ω.

number of devices access RACHs in a highly synchronized manner [1]. Smart metering is one of smart city applications. In the model, the housing density of an urban area of London located within a single cell was used as the density of meters. We set the number of meters N to 35,670. Each meter requested one data transmission for reading frequency T. We set the reading frequency to 5 min. 3GPP defines two different traffic models for smart metering applications. We adopted a Beta distribution based model for our experiments. The number of meters that start the RA procedure in the tth RA slot is defined as

$$n_t = N \int_{t}^{t+1} p(t)\, dt, \tag{4}$$

where $p(t)$ follows the Beta distribution. The $p(t)$ is defined as

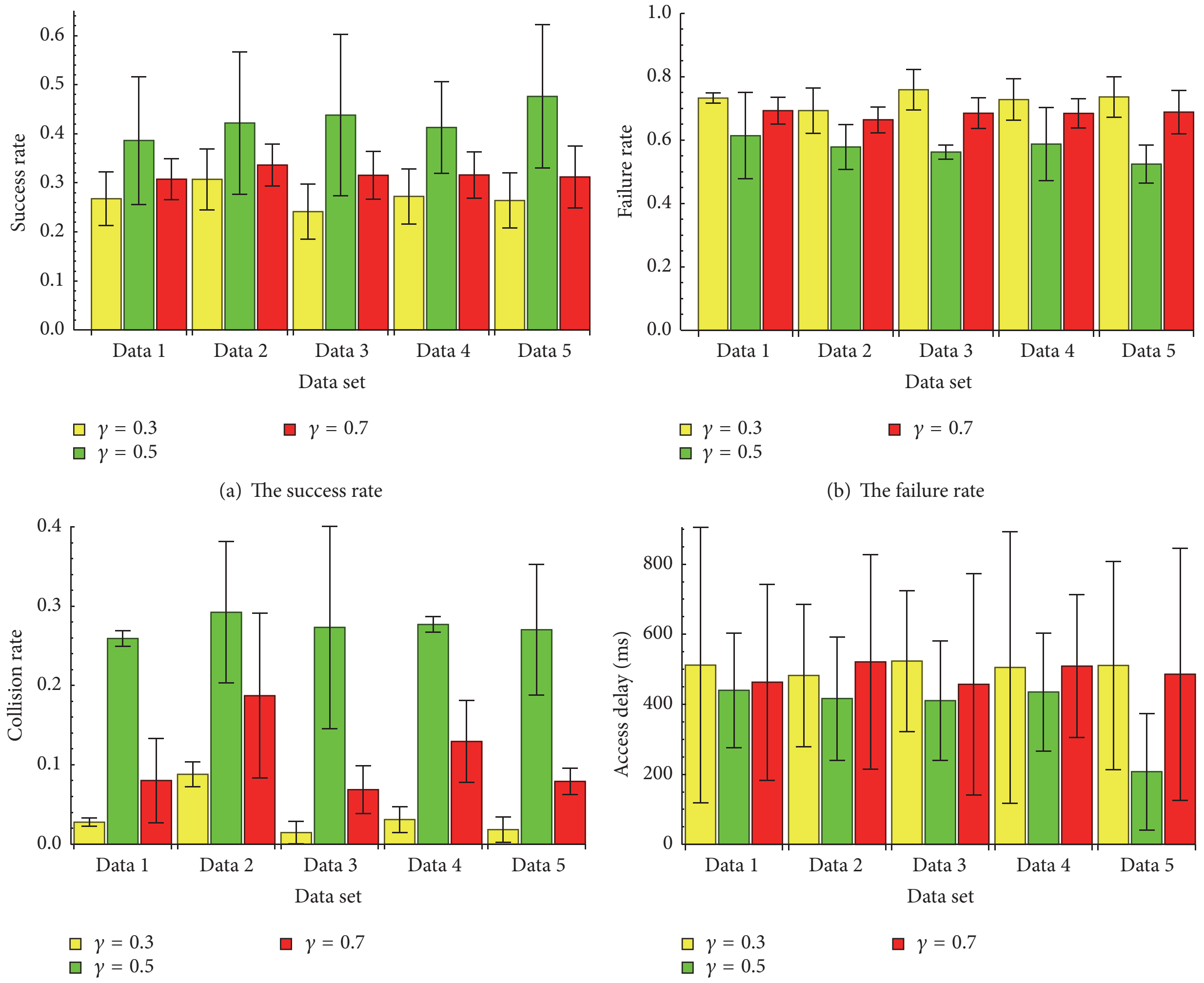
$$p(t) = \frac{t^{\alpha} \cdot (T-t)^{(\beta-1)}}{T^{\alpha+\beta-1} \cdot \text{Beta}(\alpha, \beta)}, \tag{5}$$

where Beta(α, β) is the Beta function with $\alpha = 3$ and $\beta = 4$.

In our QL model, we divided R_t^{succ} evenly into 4 states: s_1 for $0 \leq R_t^{\text{succ}} < 0.25$, s_2 for $0.25 \leq R_t^{\text{succ}} < 0.5$, s_3 for $0.5 \leq R_t^{\text{succ}} < 0.75$, and s_4 for $0.75 \leq R_t^{\text{succ}} \leq 1$. $\text{delay}_{\max}$ was set to the size of an RA slot multiplied by the *Maximum Number of Preamble Transmissions*. We set the learning rate α in (1) to 0.9. For the ϵ-greedy method, we set ϵ to 0.01. The basic RACH capacity parameters for LTE FDD networks followed [1].

Our scheme was trained using five different datasets. Each dataset included four training sets and one test set. The training sets contained the RA requests generated by the meters in a series of four reading frequencies. After the training, we measured the performance metrics of the test set. In the figures, the plotted values indicate the averages of values measured from the five datasets.

Figure 1 shows the scheme's performance with respect to various numbers of admissible actions. We defined the set elements $\Delta = \{\delta_1 = 0.2, \delta_2 = 0.1\}$ as operators to be used in actions. We used only δ_1 when there were three admissible actions: increasing p by δ_1, decreasing p by δ_1, and maintaining the current value of p. We used both δ_1



(a) The success rate

(b) The failure rate

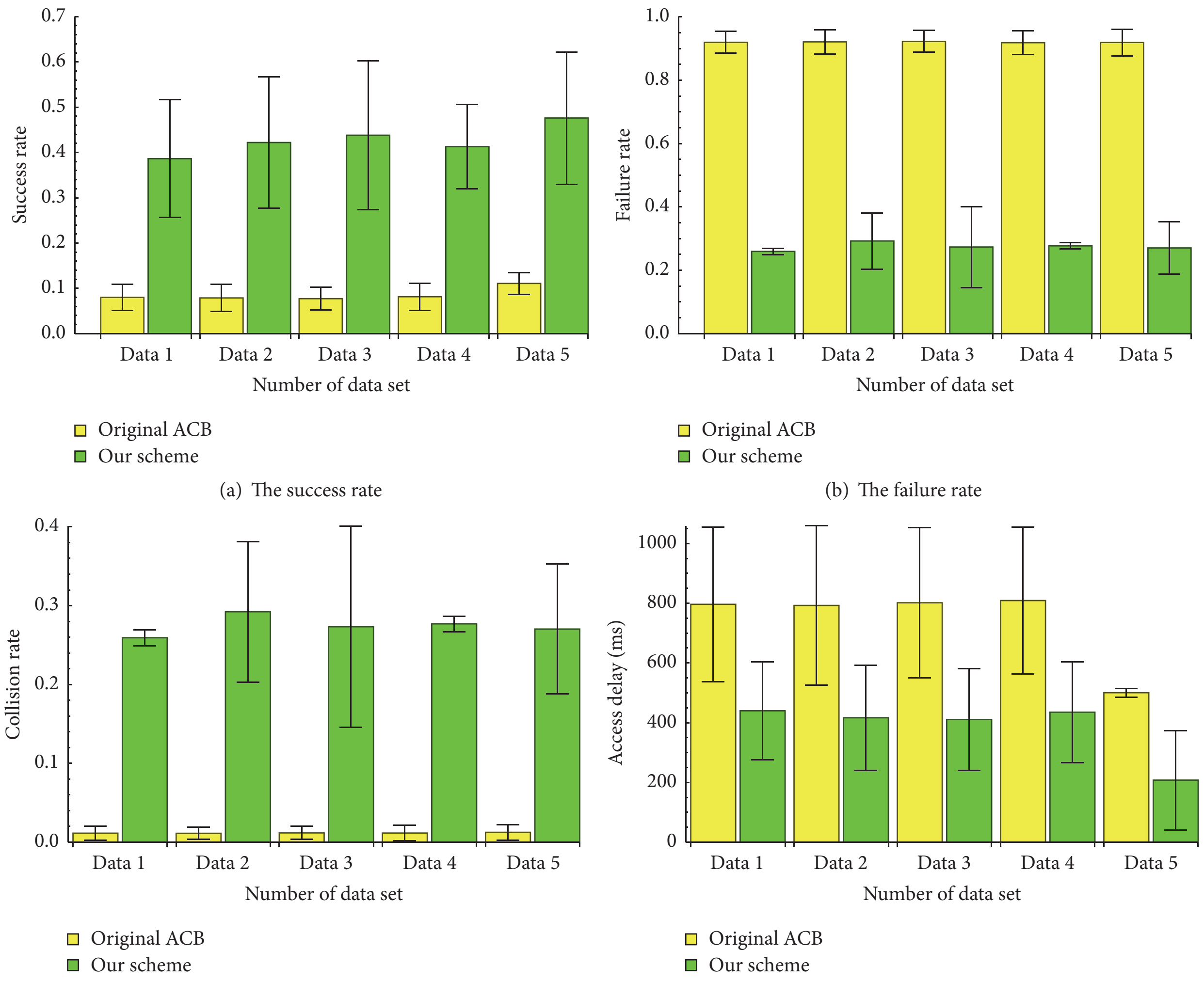
(c) The collision rate

(d) The access delay

FIGURE 3: A performance comparison for various values of γ.

and δ_2 when there were five admissible actions: increasing p by δ_1 or δ_2, decreasing p by δ_1 or δ_2, and maintaining the current value of p. For the reward in (3), we set ω to 0.5 and we set the discount factor γ in (2) to 0.5. For the access success rate, the failure rate, and the access delay, the scheme with three admissible actions showed about 78%, 25%, and 26% better performances, respectively, than the scheme with five admissible actions did. The failure rate was calculated by taking the number of devices that ultimately failed RA attempts because the preamble transmission counter had reached *Maximum Number of Preamble Transmission* and dividing it by the number of contending devices. The scheme with three admissible actions showed a collision rate of about 4 times that of the scheme with five admissible actions. The performance with respect to various numbers of admissible actions is mainly influenced by the granularity of δ. When the granularity is properly coarse (e.g., $\delta = 0.2$), the barring factor swiftly copes with the variance of the number of meters trying to access RACH. However, when the granularity is too fine (e.g., $\delta = 0.1$), the barring factor does not promptly respond to the variance of RA requests. The issue to determine the level of granularity is still open.

Figure 2 shows the scheme's performance with respect to various values of ω, which influenced the rewards, as shown in (3). We considered three admissible actions involving δ_1 with $\gamma = 0.5$. For the access success and failure rates, the scheme with $\omega = 0.5$ showed about 30% and 16% better performances than those of the scheme with $\omega = 0.3$. Moreover, the scheme with $\omega = 0.5$ showed about 3 times and 35% better performance than the scheme with $\omega = 0.7$ did. For collision rate, the scheme with $\omega = 0.7$ showed about 11 times and 24 times better performances than those with $\omega = 0.3$ and $\omega = 0.5$, respectively. This is because the rewards added more weight to the collision rate when $\omega = 0.7$. For access delay, the scheme with $\omega = 0.5$ showed about 25% and 73% better performances than those of the schemes with $\omega = 0.3$ and $\omega = 0.7$, respectively. In these cases, the rewards added weight to access delay, and the scheme with $\omega = 0.3$ showed about 38% better performance than that of the scheme with $\omega = 0.7$. As shown in the figure, the performance



(a) The success rate

(b) The failure rate

(c) The collision rate

(d) The access delay

Figure 4: A performance comparison with the original ACB.

is significantly influenced by ω. When the weight is more added to the collision rate, the collision rate is improved. When the weight is more added to access delay, the access delay is improved. The mechanism to dynamically control the smoothing factor needs to improve the performance and we leave it for our future research.

Figure 3 shows the scheme's performance with respect to various values for the discount factor γ in (2). We used three admissible actions involving δ_1 with $\omega = 0.5$. When γ was low, weight was added to quicker rewards. For the success rate, failure rate, and access delay, the scheme with $\gamma = 0.5$ showed about 37%, 22%, and 25% better performances than the scheme with $\gamma = 0.3$ did. Compared to the scheme with $\gamma = 0.7$, the scheme with $\gamma = 0.5$ showed about 26%, 17%, and 22% better performances, respectively. For collision rate, the scheme with $\gamma = 0.3$ showed about 88% and 68% better performances than those of the schemes with $\gamma = 0.5$ and $\gamma = 0.7$, respectively.

To evaluate the scheme's performance, we compared our scheme to the original ACB [7]. In the original ACB, when congestion is detected, eNB regulates meters' RA attempts by setting the barring factor to 0.1. For our scheme, we used three admissible actions involving δ_1 with $\omega = 0.5$ and $\gamma = 0.5$. In Figure 4, our scheme shows a success rate of about 5 times better than that of the original ACB. In terms of the failure rate and access delay, our scheme showed about 70% and 50% better performances, respectively. For collision rate, the original ACB showed a very low value compared to that of our scheme. In the ACB, because most meters have restricted access to RACH and the competition for channel resources is reduced, the success and collision rates decrease, but both the failure rate and the access delay increase.

5. Conclusion

To tackle the RA congestion problem of MTC in LTE-A networks, we modelled the ACB barring factor decision problem using a Q-learning algorithm. The algorithm was able to follow unexpected and burst traffic changes rapidly. The goals of our model were minimizing both the RACH collision rate and the access delay. To meet these goals, we defined sets of possible states and of admissible actions by using both the access success rate and the unit values to change the barring factor. To evaluate the performance of our scheme, we adopted the traffic model for smart metering applications. The results show that our scheme improves system performance in terms of the access success rate, the failure rate, the collision rate, and the access delay.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by the Sookmyung Women's University Research Grants (1-1703-2030) and by the 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. S. Ali, E. Hossain, and D. I. Kim, "LTE/LTE-A random access for massive machine-type communications in smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 76–83, 2017.

[4] P. Jain, P. Hedman, and H. Zisimopoulos, "Machine type communications in 3GPP systems," *IEEE Communications Magazine*, vol. 50, no. 11, pp. 28–35, 2012.

[5] 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, pp. 38–45, 2012.

[6] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? A survey of alternatives," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 4–16, 2014.

[7] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 86–93, 2013.

[8] Intel Corporation, "Further performance evaluation of EAB information update mechanisms," 3GPP R2-120270, RANWG2 Meeting #77, 2012.

[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, pp. 27–32, 2012.

[10] 3rd Generation Partnership Project, "RACH overload solutions," 3GPP R2-103742, RANWG2 #70bis, 2010.

[11] A. Ksentini, Y. Hadjadj-Aoul, and T. Taleb, "Cellular-based machine-to-machine: overload control," *IEEE Network*, vol. 26, no. 6, pp. 54–60, 2012.

[12] J.-P. Cheng, C.-H. Lee, and T.-M. Lin, "Prioritized Random Access with dynamic access barring for RAN overload in 3GPP LTE-A networks," in *Proceedings of the IEEE GLOBECOM Workshops (GC Wkshps '11)*, pp. 368–372, Houston, Tex, USA, December 2011.

[13] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement Learning is Direct Adaptive Optimal Control," *IEEE Control Systems*, vol. 12, no. 2, pp. 19–22, 1992.

[14] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using real-time dynamic programming," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 81–138, 1995.

[15] J. Moon and Y. Lim, "Access control of MTC devices using reinforcement learning approach," in *Proceedings of the IEEE International Conference on Information Networking (ICOIN '17)*, pp. 641–643, 2017.

[16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

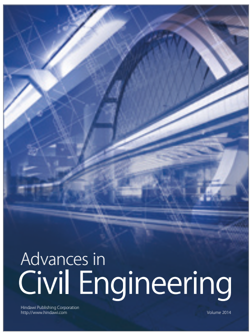
Journal of
Engineering

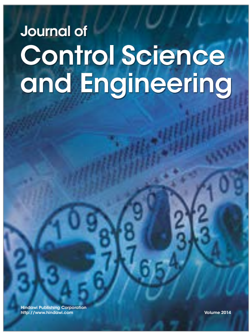
The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

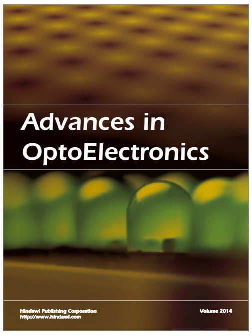
International Journal of
Distributed
Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

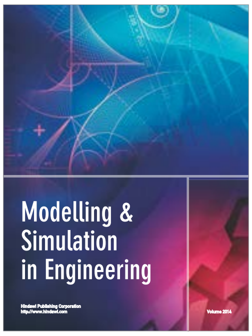
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

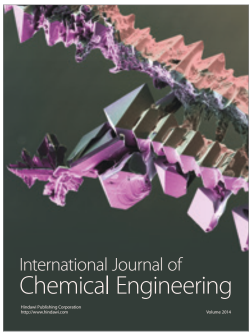

Advances in
OptoElectronics

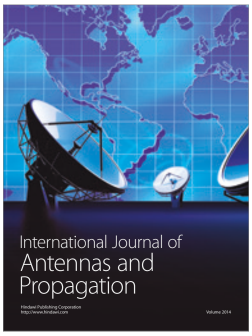
VLSI Design

International Journal of
Navigation and
Observation

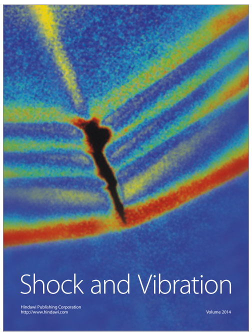

Modelling &
Simulation
in Engineering

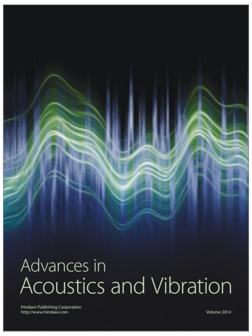
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration