DRaMA: Device-specific Repetition-aided Multiple Access for Ultra-Reliable and Low-Latency Communication

Kitaek Lee, Sundo Kim, Junseok Kim, and Sunghyun Choi
Department of ECE and INMC, Seoul National University, Seoul, Korea
{klee, sdkim, jskim14}@mwnl.snu.ac.kr, schoi@snu.ac.kr

Abstract—In Fifth-Generation (5G) New Radio (NR), satisfying a scenario called Ultra-Reliable and Low-Latency Communication (URLLC) is one of main goals. URLLC requires very high reliability with low latency, 1 ms user plane latency and 99.999% reliability. In Long Term Evolution-Advanced (LTE-A) system, however, latency requirement required by URLLC cannot be satisfied because resource allocation process increases latency. In this paper, we propose DRaMA, a grant-free uplink transmission scheme for low latency required in URLLC. With its own preamble and repetition pattern, a device repeatedly transmits the same packet using different frequency resources across consecutive mini-slots right after sending the preamble. We develop a mathematical model of collision probability to design repetition patterns, which determine the device-specific frequency resources. We demonstrate that DRaMA satisfies URLLC requirements in most cases by using system level simulation.

I. INTRODUCTION

With the advent of the Internet of Things (IoT) era, cellular systems are evolving in different directions than before. In addition to the peak data rate, which is the most important factor in current mobile communication system, so called Long Term Evolution-Advanced (LTE-A) system, there are other factors to consider such as latency and connection density for cellular IoT. In particular, low latency is emerging as a very important issue in the IoT applications. Therefore, in Fifth-Generation (5G) New Radio (NR) defined by 3GPP, satisfying low latency becomes a major goal. A service that requires very low latency and high reliability in 5G is called Ultra-Reliable and Low-Latency Communication (URLLC). Specifically, URLLC services require 1 ms user plane latency and $10^{-5}$ Packet Error Rate (PER), i.e., 99.999% reliability [1].

Examples of URLLC applications requiring very low latency and high reliability include factory automation, process automation, and smart grid [2, 3]. These applications feature intermittent traffic with very small packet size and low latency requirement. For example, in factory automation scenario, if an unintended misbehavior of a machine is detected, it needs to be reported to a controller immediately. Such occasional problems can be reported with very small amount of data, but should be reported as quickly as possible.

Current LTE-A systems have unavoidable latency because device should be allocated resource to transmit uplink data. In particular, LTE-A device should perform four-step resource allocation process for uplink transmission, thus incurring high latency [4]. Therefore, it is not appropriate to use grant-based uplink transmission through four-step resource allocation process for latency-sensitive URLLC applications.

A possible way to reduce latency for such an environment is grant-free uplink transmission without resource allocation process. Unlike the grant-based transmission scheme in which orthogonal resources are used by each device for uplink transmission, each device performs a contention-based transmission in a grant-free uplink transmission scheme. Uplink transmissions of multiple devices using same resource can lead to a collision, thus resulting in transmission failure. Therefore, grant-free uplink transmission scheme works well only when the collision problem is resolved. That is, the collision probability should be kept very low.

In this paper, we propose a grant-free Medium Access Control (MAC) protocol, called Device-specific Repetition-aided Multiple Access (DRaMA), which supports low latency uplink transmission. Each device that performs uplink transmission in DRaMA is assigned 1) an orthogonal preamble in advance for device identification and 2) a repetition pattern for performing uplink data transmission. A device using DRaMA transmits uplink data repeatedly using a predetermined repetition pattern right after the preamble transmission without a resource allocation. The repetition pattern determines which frequency resources to use for each transmission. Apparently, the probability of successful packet transmission increases by repeatedly transmitting the same packet in multiple mini-slots. In this manner, the collision problem is resolved so that grant-free uplink transmission becomes feasible. Also, DRaMA can operate robustly to the channel error by using repetition transmission. In addition, the performance of DRaMA is further enhanced by designing near-optimal repetition patterns in terms of the collision probability.

The rest of the paper is organized as follows. In Section II, we introduce resource allocation process in LTE-A and mini-slot in 5G. In Section III, we summarize the related work. Section IV provides a detailed description of the proposed DRaMA protocol. In Section V, PER analysis of DRaMA is presented, and then, how to design a near-optimal repetition pattern is presented. After evaluating the performance in Section VI, we conclude the paper in Section VII.
II. Preliminaries

In the current LTE-A system, device must be allocated resources from evolved Node B (eNB) for uplink transmission. The resource allocation process consists of four steps, and device can perform uplink data transmission only after the following resource allocation process.

1) A device sends Scheduling Request (SR) message to its eNB via Physical Uplink Control Channel (PUCCH).
2) The eNB sends an uplink grant to the device via Physical Downlink Control Channel (PDCCH).
3) The device sends a Buffer Status Report (BSR) to eNB via Physical Uplink Shared Channel (PUSCH).
4) The eNB allocates PUSCH resources to the device for the data transmission.
5) The device sends uplink data to the eNB via the allocated PUSCH resources.

The four-step process is required to allocate uplink resources to each device. However, the four-step process causes high latency, and hence, it is not suitable for URLLC services.

A method of reducing latency at physical layer is being considered. 3GPP has agreed on using shortened Transmission Time Interval (TTI) that is minimum scheduling unit of transmission [5]. In the current LTE-A, one TTI corresponds to 14 OFDM symbols (1 ms), whereas for 5G, two OFDM symbol lengths (0.143 ms) are considered one TTI. A scheduling unit composed of two OFDM symbol is called a mini-slot. We use a mini-slot as one TTI for DRaMA to satisfy URLLC requirements.

III. Related Work

There have been several proposals for reducing latency in the cellular systems with efficient resource allocation process [6, 7]. Although these schemes provide scheduling method for low latency, the latency due to the resource allocation process still remains.

Contestion-based uplink transmission schemes without resource allocation process have also been proposed. In [8], Sparse Code Multiple Access (SCMA) is proposed. SCMA scheme uses sparse codewords so that a multi-user detection based on Message Passing Algorithm (MPA) is enabled. Therefore, SCMA enables contention-based uplink transmission by enabling eNB to successfully decode data of all devices even if data of multiple devices is transmitted using the same resources. However MPA receiver still has high complexity, despite efforts to reduce the complexity. Therefore, if the number of devices increases, it is not suitable to use SCMA.

As a solution to reduce latency, Persistent Scheduling (PS) or Semi-Persistent Scheduling (SPS) may be another option [9]. While SPS and PS have been originally developed to support such applications as VoIP, which transmit periodic packets, it can reduce latency by reducing scheduling overhead. However, since SPS and PS pre-schedule orthogonal resources to devices, it is inefficient for URLLC applications with non-periodic packet arrivals.

IV. DRaMA: Proposed Grant-free Uplink Transmission Scheme

A. System Model

In this paper, we consider a situation where tens to hundreds devices perform uplink transmission when an event occurs. We consider an environment where all devices are in the RRC_CONNECTED state [10]. In an example such as a smart grid or a factory automation environment where devices are externally powered and latency is an important factor, the IoT devices can be considered to maintain the RRC_CONNECTED state. Therefore, all devices using DRaMA are assumed to be uplink synchronized. We assume that all devices have the same type of traffic. Each device has an unique preamble and a repetition pattern for uplink transmission, which are allocated in the initial setup stage.

Uplink transmission using DRaMA is composed of two phases: preamble phase and data transmission phase. The preamble phase takes up one mini-slot and is the stage where the device sends its own preamble before sending uplink data. The preamble and the repetition pattern have a one-to-one correspondence relationship. Therefore, the preamble serves not only as a device ID but also as a pattern ID. Also, the preamble is used as a reference signal for channel estimation.

In the data transmission phase, one mini-slot is required to transmit one packet. Accordingly, the data transmission phase takes up as many mini-slots as the number of repeated transmissions, and each device actually transmits an uplink packet. We refer to the length of the data transmission phase (i.e., the number of repetitions) as $T$. An uplink transmission process including the preamble phase and the data transmission phase is referred to as one cycle. Detailed discussion is provided in Section V.

B. Proposed Preamble

Basically, the preamble used in the proposed scheme has the same principle as random access preamble of LTE-A system. In the LTE-A system, Zadoff-Chu (ZC) sequence [11] is used as the random access preamble. The eNB can successfully decode all the preambles even if the eNB receives multiple different preambles simultaneously, since different ZC sequences have very low cross correlation. Unlike the random access preamble of LTE-A with a subcarrier spacing of 1.25 kHz, the proposed preamble has the same subcarrier spacing as an OFDM symbol for data transmission, i.e., 15 kHz. Therefore, the duration of ZC sequence of the proposed preamble can be shortened by one OFDM symbol duration length for data transmission. In addition, if two OFDM symbols are used, the proposed preamble can be transmitted in a mini-slot duration. We propose a method to increase the number of preambles by using more frequency resources up to full bandwidth and using a smaller value of the cyclic shift. Using the above method, it is possible to generate a much larger number of preambles than 64 in the current LTE-A so that those preambles can be uniquely allocated to devices.
C. Repetition Pattern

A repetition pattern used determines which frequency resource to use when the device repeats transmission in multiple mini-slots. The minimum unit of frequency resource is determined by the required packet size. For the convenience, we refer to the minimum unit of frequency resource as a frequency slot. One frequency slot is composed of one or more RBs depending on the packet size and MCS. In addition, we define the number of available frequency slots in the system bandwidth as $F$. Fig. 1 shows examples of repetition patterns where one resource block or two resource blocks are used as one frequency slot. A repetition pattern is represented by a $F \times T$ matrix. Each element of the matrix yields 1 if the $f$th frequency slot in the $t$th mini-slot is selected, or 0, otherwise. If multiple devices transmit uplink data using the same frequency slot in a specific mini-slot, a collision occurs and transmission may fail. Therefore, how to avoid collisions should be considered when designing repetition patterns.

We define the similarity of two specific repetition patterns, that is the number of mini-slots in which the selected frequency slots of the two patterns are the same. The similarity of two specific patterns which are represented by matrices, respectively, $P$ and $Q$ can be expressed as

$$sim(P, Q) = tr(P^T \cdot Q).$$

(1)

Two devices with high similarity patterns have high influence on each other when transmitting simultaneously. Also, only if the similarity of two patterns is 1 or 0, collision probability is independent in all mini-slots. Obviously, it is best if all devices are assigned patterns with zero similarity, but it might be impossible because the number of available patterns is limited. Therefore, the similarity of all patterns should be minimized while satisfying the required number of patterns.

D. Uplink Transmission with DRaMA

When an uplink packet arrives in a device, the device waits for a nearest preamble phase and transmits its preamble. By sending the preamble, the device informs the eNB of its device ID and the pattern to be used for the uplink transmission. As described above, the eNB can successfully decode all the transmitted preambles. Immediately after transmitting the preamble, the device transmits an uplink packet repeatedly using the predetermined repetition pattern.

Fig. 2 shows a simple example of uplink transmission of four devices using DRaMA. In this example, a repetition pattern is represented by a $3 \times 3$ matrix. Devices A, B, C, and D use repetition patterns as shown in Fig. 2(a). In Fig. 2(b), the only device A transmits a packet in the first DRaMA transmission cycle, devices A, B, and C transmit in the second transmission cycle, and devices A, B, C, and D transmit in the third transmission cycle. In the second and third cycle, we find that devices suffer from collisions. However, each device successfully transmits a packet in at least one transmission by using a different repetition pattern.

V. REPEATED PATTERN DESIGN

In this section, we discuss how to design repetition pattern to improve the performance of DRaMA in terms of the collision probability and packet error probability. It is important to design good repetition pattern rather than repeating transmission randomly, because strictly low PER is required by URLLC. The PER performance of DRaMA is closely related to the collision probability. We design an near-optimal repetition pattern and a pattern assigning rule. Whenever a repetition pattern is assigned to a new device, the repetition pattern is determined with the rule of minimizing the sum of the collision probabilities.

A. Numerical Analysis of PER Performance

We start by numerically analyzing PER performance of DRaMA. In this work, we assume that packet arrivals of each device follow a Poisson process. Therefore, the collision
probability of the $k^{th}$ device when the $k^{th}$ device uses the $f^{th}$ frequency slot in the $t^{th}$ mini-slot, $\varphi_{\text{col},k}^{(t)}$, is expressed as

$\varphi_{\text{col},k}^{(t)} = 1 - e^{-(N_f^{(t)}-1)\lambda_{\text{sec}}}$, \hspace{1cm} (2)

where $N_f^{(t)}$ is the number of devices included in the $t^{th}$ mini-slot and the $f^{th}$ frequency slot, and $\lambda_{\text{sec}}$ (/cycle) is the packet arrival rate. $\lambda_{\text{sec}}$ (/s) and $\lambda_{\text{yc}}$ (/cycle) are the same packet arrival rates in different units. Note that $\lambda_{\text{yc}} = \frac{1}{2} \cdot \frac{T+1}{1000} \cdot \lambda_{\text{sec}}$. That is, the collision probability is equal to the probability that any devices other than the $k^{th}$ device transmits simultaneously with the $k^{th}$ device.

Another cause of packet transmission failure channel error. We define that $\varphi_{\text{err},k}^{(t)}$ is channel error probability of the $k^{th}$ device in the $t^{th}$ mini-slot. Since channel error and collision are independent, the probability of packet transmission failure in one mini-slot is expressed as

$\varphi_{\text{err}}^{(t)} = \varphi_{\text{err},k}^{(t)} + \varphi_{\text{col},k}^{(t)} - \varphi_{\text{err},k}^{(t)} \varphi_{\text{col},k}^{(t)}$. \hspace{1cm} (3)

Since URLLC requires tight latency constraint, if packet transmission is not successful within latency constraint, it is defined as transmission failure. That is, assuming that the time duration of $s$ mini-slots is latency constraint, the device should transmit its packet within $s$ mini-slots. If the waiting time to send a preamble after packet arrival is too long, it is impossible to use all the repetitions. That is, the number of opportunities of transmission repetition depends on the packet arrival time. Assume that $t_w$ is the number of mini-slots from a mini-slot in which a packet arrives to a mini-slot just before the preamble phase. That is, $t_w$ is the number of mini-slots that a device should wait to send preamble. Then, the number of opportunities, $t_o$, is expressed as

$t_o = \max \left(0, \min(s - t_w - 1, T)\right)$. \hspace{1cm} (4)

Assuming that i) the transmission fails in the mini-slot in which collision occurs, and ii) all the collision probabilities are independent, the transmission failure probability $P_{\text{fail,k}}$ in one cycle is expressed as

$P_{\text{fail,k}} = 1 - \sum_{t_w=1}^{T+1} \frac{1}{T+1} \left(1 - \prod_{t=1}^{t_w} \varphi_{\text{err}}^{(t)} \right)$. \hspace{1cm} (5)

We design repetition patterns to ensure that the collision probabilities in different mini-slots are independent and the sum of the collision probabilities of the devices is minimized by using (2) and (5).

**B. Frequency Hopping Repetition Pattern**

We propose a repetition pattern using frequency hopping technique. The number of frequency slots to jump before each transmission is constant and a frequency slot is determined by using $\text{mod}(\cdot, F)$ operation. The frequency slot used for transmission in the first mini-slot in a specific repetition pattern is referred to as ‘root frequency slot’ and the number of frequency slots to jump before each transmission is referred to as ‘hopping degree.’ Root frequency slot index has a value from 1 to $F$ and hopping degree has a value from 0 to $F - 1$. Then, a repetition pattern with the root frequency slot index $r$ and the hopping degree $h$, respectively, is expressed as a $F \times T$ matrix, $M^{(r,h)}$, with the following elements.

$m_{i,j}^{(r,h)} = \begin{cases} 1, & \text{if } i = \text{mod}(r + (j - 1)h - 1, F) + 1, \\ 0, & \text{otherwise}. \end{cases}$ \hspace{1cm} (6)

Assuming that $h$ has a value from 0 to $F - 1$, the number of patterns that make the similarity of all patterns less than or equal to 1 is expressed as

$\# \text{ patterns w/ similarity } 1 \text{ or } 0 = \frac{F^2}{\min_{1 \leq t \leq T - 1} \text{gcd}(F, t)}$. \hspace{1cm} (7)

If determining $F$ be a prime number, we can make $F^2$ patterns that make the similarity of all patterns less than or equal to 1 regardless of $T$. In addition, if we group $F$ patterns with the same $h$ but different $r$ into a group, we divide the patterns into $F$ pattern groups. Two patterns in the same group have a similarity of 0, and two patterns in different groups have a similarity of 1, thus the collision probabilities are independent in all mini-slots. At the initial setup stage, different patterns are assigned to devices with the following rule.

1) One pattern group is selected, and all of the $F$ patterns in the group are assigned to the devices.
2) Another pattern group is selected, and the process is repeated until all devices are assigned a pattern.

If arrival rate is very low (e.g., $\lambda_{\text{sec}} = 1$ /s), the frequency hopping repetition patterns satisfy the following lemma.

**Lemma 1.** If we consider assigning a frequency slot to a new device in a mini-slot, the sum of the collision probabilities is minimized by assigning a frequency slot assigned to the minimum number of devices to the device.

**Proof.** Assume that there exist $K$ devices in total, and $N_f^{(t)} (\leq K)$ devices are assigned to the $f^{th}$ frequency slot out of $F$ frequency slots. We consider the situation where the $(K + 1)^{th}$ device is newly included in the $f^{th}$ frequency slot. Packet arrival rate of each device is $\lambda_{\text{yc}}$ (/cycle). We define $\Phi_k^{(f)}$ as the sum of the collision probabilities of $K$ devices already included. Besides, we define $\Psi_k^{(f)}$ as the sum of the collision probabilities of $(K + 1)$ devices when the $(K + 1)^{th}$ device is included in the $f^{th}$ frequency slot. Since only the collision probabilities of the devices that are included in the $f^{th}$ frequency slot change, $\Psi_k^{(f)}$ is then expressed as

$\Psi_k^{(f)} = \Phi_k^{(f)} + \left( N_f^{(t)} + 1 \right) \left( 1 - e^{-N_f^{(t)} \lambda_{\text{yc}}} \right) - \sum_{l=0}^{f-1} N_f^{(t)} \lambda_{\text{yc}} \left( e^{-\lambda_{\text{yc}}} - 1 - \lambda_{\text{yc}} \right)$, if $\lambda_{\text{yc}} \ll 1$. \hspace{1cm} (8)
If $N_f^{(t)} \lambda_{\text{cy}}$ is less than 2, $\Psi_{K+1}^{(t)}(f)$ is an increasing function of $N_f^{(t)}$. Since $\lambda_{\text{cy}}$ is very small, it is reasonable that $N_f^{(t)} \lambda_{\text{cy}}$ is less than 2. Therefore, to minimize the sum of the collision probabilities, the $(K+1)^{th}$ device should be assigned to the frequency slot $f^*$ that is expressed as

$$f^* = \arg \min_{1 \leq f \leq F} \Psi_{K+1}^{(t)}(f)$$

$$= \arg \min_{1 \leq f \leq F} N_f^{(t)},$$

(9)

That is, allocating the frequency slot containing the smallest number of devices to the new device minimizes the sum of the collision probabilities.

Obviously, the frequency hopping patterns allocated to all devices by the above-described pattern allocation rule allocate a pattern having a frequency slot including the smallest number of devices in each mini-slot to a new device. Also, since the collision probabilities in different mini-slots are all independent, such frequency hopping patterns minimize the sum of the collision probabilities in all mini-slots whenever a device is assigned a repetition pattern.

C. Determining Number of Repetitions

The first thing to notice before determining the number of repetitions is that simply increasing the number of repetitions cannot be the best way. As the number of repetitions increases, the length of one cycle becomes longer so that average number of devices performing uplink transmission in a cycle increases and the collision probability also increases. In addition, as the length of one cycle increases, the average waiting time in the Tx buffer increases and the average latency increases. Therefore, it is necessary to determine an appropriate number of repetitions considering both latency constraint and collision probability. Since the exact channel is unknown in each transmission, we assume that channel error probability is fixed when determining the number of repetitions. First, we determine expected failure probability and find the number of repetitions that make the expected failure probability to be minimized. The expected failure probability is determined as $F_{\text{fail}}(t)$, which is the average failure probability of all devices. In fact, failure probability $P_{\text{fail},k}$ is a monotonic increasing function of $r_{\text{err},k}(t)$, so that any fixed channel error probability does not affect determining the number of repetitions. Therefore, for all $T$ with $2 \leq T \leq F$, we determine the number of repetitions to minimize the expected failure probability. Therefore, the number of repetitions $T^*$ satisfying the above condition is expressed as

$$T^* = \arg \min_{2 \leq T \leq F} F_{\text{fail}}(T).$$

(10)

In this way, $T^*$ minimizes the expected failure probability.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of DRaMA via system level simulation using MATLAB. In the simulation, we assume that the system bandwidth is 20 MHz, supporting up to 100 RBs. A small cell environment suitable for the considered IoT environment with a cell radius of 250 m is assumed. We assume that devices transmit packets with data of 32 bytes that URLLC applications require [1], 32 byte packets can be transmitted with 4 RBs using MCS 21 and with 5 RBs using MCS 18 with a mini-slot. Therefore, using the entire 100 RBs, 25 and 20 frequency slots can be used with MCS 21 and 18, respectively. Since the largest prime number smaller than 25 is 23, the number $F$ of frequency slots in DRaMA frequency hopping pattern is 23 using MCS 21. On the other hand, since the largest prime number smaller than 20 is 19, the number of frequency slots in DRaMA is 19 using MCS 18. In addition, because URLLC features intermittent traffic, we assume packet arrival rate of 0.5/s and 1/s. The values used in the simulation are summarized in Table I.

A. Frequency Hopping Patterns vs. Pseudo-random Patterns

We comparatively evaluate the performance of DRaMA with frequency hopping patterns and pseudo-random patterns, based on PER due to the collision and the channel error (assuming fixed channel error probability of 0.001). Fig. 3(a) shows collision probability with varying number of devices using DRaMA with pseudo-random patterns and frequency hopping patterns. The collision probability of the devices using frequency hopping patterns is lower than the collision probability using pseudo-random patterns.

Fig. 3(b) shows PER with varying number of devices. We observe that DRaMA with frequency hopping patterns outperforms DRaMA using pseudo-random patterns, since pseudo-random patterns cannot guarantee independent collision probability in mini-slots. If collision probability in each mini-slot is not independent, the performance is degraded because some devices may affect collisions more.
In the URLLC scenarios with event driven packet arrival, SPS that periodically modifies scheduling is not appropriate. Therefore, we compare DRaMA with PS only. Both devices with DRaMA and devices with PS use MCS 21. Fig. 4(a) shows the average latency of DRaMA and PS with varying number of devices. We observe that the average latency of DRaMA remains constant, while the average latency of PS increases with the number of users. Since each device uses orthogonal resources in PS, scheduling period should be increased when the number of devices increases. However, the average latency in DRaMA remains constant regardless of the number of devices because the cycle length is fixed.

Fig. 4(b) shows PER of DRaMA and PS with varying number of devices. Unlike devices with DRaMA, devices with PS transmit a packet only once with a given resource. Therefore, it is observed that PS is more vulnerable to channel error than DRaMA. Also, in the case of PS, the scheduling period becomes longer as the number of devices increases so that if scheduling period is greater than or equal to the latency constraint, PER increases sharply.

C. Packet Error Rate

We evaluate the performance of DRaMA with various MCS, packet arrival rate, and the number of devices. Fig. 5(a) shows the average PER for various cases. Using (5) and channel error probability according to each MCS, the upper bound of the number of devices satisfying URLLC requirement is derived. In the case of $\lambda = 1$, the upper bounds are 213 and 240 when MCS indexes are 18 and 21, respectively. We observe that DRaMA can support URLLC requirements well in cases where the number of devices is less than the upper bound.

Fig. 5(b) shows Empirical Cumulative Density Function (ECDF) of PER for each device when the arrival rate is 0.5/s. It is observed that most devices transmit packets within 1 ms with PER lower than $10^{-5}$ except for the case where the number of devices is 300. In the case of 200 devices, only three out of 200 devices do not satisfy URLLC requirement.

VI. Conclusion

In this paper, we propose DRaMA, a novel grant-free uplink transmission scheme for URLLC services. A device using DRaMA can perform uplink transmission without resource allocation that causes high latency by using a unique preamble and a predetermined device-specific repetition pattern. We design the repetition patterns of DRaMA with good performance through collision probability and PER analysis. The performance of DRaMA is evaluated through system level simulation. In most cases, the results demonstrate that packets can be transmitted within 1 ms with PER lower than $10^{-5}$ that is required in URLLC scenario. In addition, it is observed that DRaMA performs much better when frequency hopping repetition pattern is used, rather than pseudo-random pattern is used. Also, DRaMA outperforms PS, since DRaMA satisfies very low PER requirement which PS cannot satisfy at all.

ACKNOWLEDGEMENTS

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (2015-00300, Multiple Access Technique with Ultra-Low Latency and High Efficiency for Tactile Internet Services in IoT Environments).

REFERENCES