Reducing State Information by Sharing IMSI for Cellular IoT Devices

Manabu Ito, Nozomu Nishinaga, Yoshinori Kitatsuji, and Masayuki Murata, *Member, IEEE*

***Abstract*—To provide cellular communications for a large number of Internet of Things (IoT) devices, the mobile core system, called evolved packet core (EPC) in the case of a 4G system, requires a large amount of computational resources. This is because the EPC needs to maintain communication state information for IoT devices, even if they rarely send data. This paper proposes a method for reducing state information. The proposed method assigns the same international mobile subscriber identity to multiple IoT devices, which upload data with the same cycle, and manages devices uploading data at different timings. This paper investigates the impact of factors affecting the performance of the proposed method through several simulation-based verifications and shows that the proposal can reduce the required state information to less than 0.5% compared to the current EPC method.**

***Index Terms*—Cellular networks, 4G mobile communication, information management, international mobile subscriber identity (IMSI), Internet of Things (IoT), telecommunication control, time division utilization.**

I. INTRODUCTION

MOBILE network technologies for the Internet of Things (IoT) are expected to provide wide area coverage and low power communications at a low cost (for devices and usage) [1]. IoT services, which must meet these requirements, include metering of gas and water utilities, and tracking logistics containers, cars, high-value bicycles, and pets. The "IoT devices" used in these services have no power source, are dispersed geographically, and some of them move around frequently.

Widely deployed cellular networks have made possible low power consumption for IoT devices and cost reductions for inexpensive IoT devices and services. To reduce the power consumption of devices, a sleep mode called power saving mode (PSM) [2], where the device does not respond to calls from the network, has been introduced to devices. The PSM enables devices to operate for more than ten years on two long life AA batteries [3]. To reduce the cost of devices, simplified radio frequency (RF) hardware has been developed [4]; one that uses a single antenna and half-duplex communication to ensure the minimum bandwidth and data rate requirements for IoT services. Several methods for reducing the costs associated with cellular network usage have been studied [5], including the effective utilization of hardware resources using virtualization techniques and optimization of communication controls. However, these methods are still not sufficient for the realization of inexpensive IoT services.

The evolved packet core (EPC), which is a representative mobile core system that constitutes the cellular network at present, controls data paths (called "bearers") for each accommodated user terminal such as smartphones and IoT devices. The EPC continuously maintains the connection status (also referred to as state information) of bearers for each user terminal even if it rarely sends data. This requires a large number of computational resources to store and process a large amount of state information when the EPC needs to support a large number of IoT devices. Additionally, a rapid increase in demand for computational resources creates difficulties in reducing the EPC's capital expenditure/operational expense (CAPEX/OPEX). The best way to address this problem would be reducing the amount of state information as much as possible.

The method in [6] reduces the number of IoT devices that establish a bearer in the EPC by having them communicate with the EPC via a gateway. However, in the case of providing IoT services over a widely distributed area, this method may impose a burden on the device side. Appropriate devices need to be selected as gateways in order to efficiently reduce state information. In addition, this method may not reduce state information to any significant degree unless mobile devices and device density are taken into account.

This paper proposes a communication control method that aims to reduce the amount of state information retained in the EPC without adding additional sophistication to the connections among IoT devices. The proposed method is an approach based on the enhancement of the EPC (in contrast to the previous method [6] that is based on a user-side approach). The proposed method assigns the same international mobile subscriber identity (IMSI) to multiple IoT devices, which upload data with the same cycle, and manages devices uploading at different times. From the EPC's perspective, it sees a device that alternates between movement and communication. This is because bearers are established and released with

Manuscript received February 19, 2016; revised April 16, 2016 and May 21, 2016; accepted June 22, 2016. Date of publication July 7, 2016; date of current version January 10, 2017.

M. Ito is with the National Institute of Information and Communications Technology, Koganei, Tokyo 184-8795, Japan, and also with the Graduate School of Information Science and Technology, Osaka University, Suita 565-0871, Japan (e-mail: mn-ito@nict.go.jp).

N. Nishinaga is with the National Institute of Information and Communications Technology, Tokyo 184-8795, Japan (e-mail: nisinaga@nict.go.jp).

Y. Kitatsuji is with the KDDI Research and Development Laboratories, Inc., Fujimino 356-8502, Japan (e-mail: kitaji@kddilabs.jp).

M. Murata is with the Graduate School of Information Science and Technology, Osaka University, Suita 565-0871, Japan (e-mail: murata@ist.osaka-u.ac.jp).

Digital Object Identifier 10.1109/JIOT.2016.2587823

a single IMSI by multiple IoT devices in different locations (this is not a handover, which is the process of maintaining the communication as devices move). To prevent communication timings from overlapping, time slots at periodic intervals are introduced in the EPC. The proposed method controls, by using time slots, the timings of the registration process of IoT devices powered up (specifically powered up at random), thereby controlling the timing of uploading data after the registration process. This paper shows that the proposal can reduce the required state information to less than 0.5% compared to the current EPC method. This conclusion is based on the observation that two hundred or more IoT devices sharing a single IMSI can upload data in turns. Preliminary results of this work have been published in [7]. This paper investigates the impact of important factors provisionally configured in our previous work, carefully configures them, and, as a result, shows the improved performance of the proposed method.

The rest of this paper is organized as follows. Section II describes related work aimed at reducing state information in the EPC. Section III introduces the current procedure that IoT devices follow for uploading data through the EPC. Section IV explains our proposed method for the communication control of IoT devices that share a single IMSI. Section V presents performance evaluations of the proposed method made by using simulation experiments, and discusses related issues. Section VI presents our conclusions.

II. Related Work

Methods that can reduce state information in the EPC are categorized into two types: first, methods that modify the EPC; and second, methods that utilize gateway devices in access networks. The former is referred to as EPC-side methods, and the latter is referred to as device-side methods.

The EPC-side methods include the enhancement of standardized functions [8], [9], the use of implementation by separating the *C*-plane from the *U*-plane [10], [11], and a new architecture [12]. In the enhancement of standardized functions, when devices do not transmit data, the communication tunnels that constitute bearers in the EPC are released (in general, the communication tunnels are maintained based on the number of devices irrespective of communication status of the devices). The methods can reduce the number of the communication tunnels, which are constantly established in the EPC. In *C/U*-planes separation, data paths are controlled without using communication tunnels. The enhancement of standardized functions [8], [9] and *C/U*-planes separation [10], [11] have limited effectiveness in reducing state information. That is, although state information maintained in *U*-plane can be reduced, that maintained in *C*-plane cannot be reduced. In contrast, the new architecture [12] can reduce state information in *C*-plane. In this architecture, the mobile core network comprises general switches and middle boxes, and the architecture has adopted a routing control model that is based on a software defined network (SDN) that controls data paths without using communication tunnels. In addition, this architecture reduces the amount of information (rules for routing control) maintained in a whole mobile core network by aggregating rules for routing control. However, this architecture cannot reduce information such as users' profiles and locations of devices. Considering market trends, cellular networks need to support IoT services before migrating from the current EPC to the new architectures based on an SDN (e.g., 5G networks). It takes some time to migrate from the current EPC to a new architecture. Consequently, it is favorable to use methods based on the current standardized architecture.

In the device-side method [6], devices communicate with the EPC through gateway devices appropriately chosen among the devices by using device-to-device communication. This method can reduce the number of devices that establish bearers in the EPC, thereby reducing the amount of state information in the EPC. However, to provide IoT services over a wide area, it is necessary that multiple gateway devices be optimally selected from devices to improve the reduction of state information. This is because there are limitations to the number of devices that can be managed by a single gateway device and the communication distance between the gateway device and other devices. Although the combined use of multihop communication technologies may improve the flexibility of selection of gateway devices, some calculation processes, such as routing control and congestion control in the device, are required, thereby reducing the battery life of the device. In addition, moving devices cannot always establish network connectivity via the gateway devices, and a region of low device density requires a certain number of gateways due to the limitation of communication distance between devices. These prevent state information from being effectively reduced. In contrast, the proposed method does not require any assistance from devices due to the independence of the moving devices and device density. Even so, the proposed method can reduce the amount of state information to the same degree or better than the device-side method (details in Section V-E). Table I shows summary of comparison between the proposed method and related works.

TABLE I
Comparison Between Related Works and Proposed Method

Methods	Features			
	Solution type	**Amount of reduction of state information**	**Lead time to market**	**Prolong battery life**
Enhancement of standardized functions [8][9]	EPC-side	Low (Only U-plane)	short	Favorable
C/U-planes separation [10][11]	EPC-side	Low (Only U-plane)	long	Favorable
New architecture [12]	EPC-side	Medium ~ Large (C/U-plane)	long	Favorable
Proposed	EPC-side	Significant (C/U-plane)	short	Favorable
Gateway-based [6]	Device-side	Large (C/U-plane)	short	-

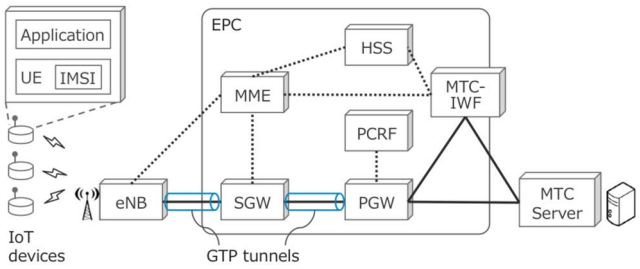

Fig. 1. Standard cellular network architecture.

III. IoT Communication Using Cellular Network

A. EPC and Communication Procedures for IoT Devices

Fig. 1 shows the EPC architecture with a newly standardized function for IoT devices (which we refer to as "devices," unless specifically distinguishing between IoT devices and devices). The device is equipped with user equipment (UE) that has the ability to access the EPC, and an application for recording sensor's data and transmitting the data. The UE has an IMSI, which is used as a unique identification number in the EPC. For machine type communication (MTC) such as IoT, an MTC interworking function (MTC-IWF) is added [13]. This function serves as an intermediary between the EPC and MTC server where the applications for IoT services run. The MTC server is liberated from the responsibility of managing the IMSIs and IP addresses of devices. The MTC server uses external identifiers (EIDs), which are newly defined, for identifying and calling the devices (called "device triggering").

The packets that are transmitted and received between devices and MTC server are routed using two GPRS tunneling protocol (GTP) tunnels, which constitute a bearer. There are base stations (called "eNBs") for a long term evolution (LTE) access network, a serving gateway (SGW), and a packet data network gateway (PGW) on the data path, wherein the primary function is packet routing/forwarding and policy enforcement. The SGW serves as a mobility anchor that enables devices to have seamless communication when the device moves from one eNB to another. The PGW performs traffic monitoring, billing, and access control, and acts as a gateway to external networks. The GTP tunnels are created, removed, and updated by the eNB, SGW, and PGW in a coordinated manner using a mobility management entity (MME) and a policy and charging rules function (PCRF). The MME is a *C*-plane function, which is responsible for mobility management and user authentication via a home subscriber server (HSS), which has subscriber profiles and the corresponding information between the EID and IMSI. The PCRF provides the PGW with QoS authorization based on user's profile and/or the media information (e.g., the TCP/IP 5-tuple and media codecs) received from external networks. Table II summarizes function of blocks in Fig. 1.

When the power is turned on, the device registers itself with the EPC—also known as the "attach" procedure. In this procedure, the EPC registers the location of the device, assigns it an IP address, and establishes a bearer. After the completion of the attach procedure, state information (IP address, location, bearer IDs, external network information, QoS, etc.), which is indexed by the IMSI, is maintained in the functions (eNB, MME, SGW, and PGW) that constitute the EPC. When there is no data on the data path after the Attach procedure or data transmission, only the GPRS tunnel between the eNB and SGW is released, whereas the tunnel between the SGW and PGW is allowed to remain. The device then enters idle mode. IoT devices often use PSM, where they do not respond to calls from the network (details in the next section), to save battery. After returning from PSM, a device executes a tracking area update (TAU) to update its location. If the device receives paging messages from the EPC shortly after performing the TAU, the device re-establishes the GPRS tunnel between the eNB and SGW and the bearer returns to being available.

TABLE II
Responsibilities of Function Blocks

Block/Module	Responsibilities
UE	The UE has an ability to access the EPC.
Application	The application records sensor's data and transmits the data.
IMSI	The IMSI is used as a unique identification number in the EPC.
eNB	The eNB is a base station for a LTE access network.
MME	The MME is responsible for mobility management and user authentication via a HSS.
SGW	The SGW serves as a mobility anchor that enables devices to have seamless communication when the device moves from one eNB to another.
PGW	The PGW performs traffic monitoring, billing, and access control, and acts as a gateway to external networks.
PCRF	The PCRF provides the PGW with QoS authorization.
HSS	The HSS has subscriber profiles and the corresponding information between an EID and IMSI.
EID	The EID is newly defined for identifying and calling IoT devices.
MTC-IWF	The MTC-IWF serves as an intermediary between the EPC and an MTC server.
MTC Server	The MTC server identifies IoT devices by EIDs.

B. Power Saving Mode

In PSM, the devices switch off RF modules, thereby entering sleep mode, during which the devices do not respond to paging messages. The device activates PSM using two timers, which are obtained in the attach or TAU procedure. The first timer (called T3324) is the time during which the device remains in idle mode following the attach or TAU procedure. The second timer (called T3412) is the periodic TAU timer. Subsequent to the EPC releasing radio resource and eNB-to-SGW tunnel, T3324 and T3412 are initiated. Until T3324 expires, the device can remain in idle mode and respond to the paging messages or any other signaling messages from the EPC. Once T3324 expires, the device will then enter the PSM for the duration of T3412. The device can cancel PSM

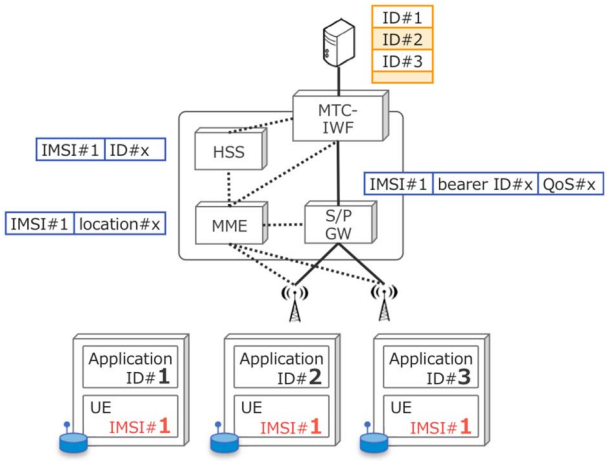

Fig. 2. Overview of proposed method.

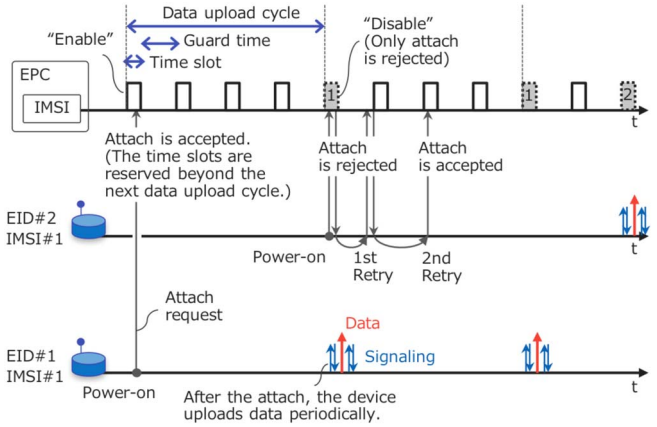

Fig. 3. Overview of the control of communication timings.

by conducting a TAU or by sending a service request that is triggered by the application layer to re-establish its bearer.

IV. Proposed Methods

To aggregate cellular communication lines (bearers for IoT devices) in the EPC, this paper proposes a method that allows a single communication line to be shared among the IoT devices using time division. Note that the IoT devices communicate with a MTC server at a constant frequency. Fig. 2 gives an overview of the proposed method. Its key feature is that multiple devices have the same IMSI (lower layer ID) and transmit data to the MTC server in turns (at different times). The application ID (upper layer ID) is responsible for identifying IoT devices. From the perspective of the EPC, a single device alternatives between movement and communication (uploading data). Therefore, the proposed method can reduce the amount of state information managed in the EPC.

Note that it is assumed that devices sharing a single IMSI are used in the same IoT service. In other words, the devices accommodated in the same IoT service have the same communication policy (upload cycle, traffic, etc.). After completing the attach procedure following power-on, devices upload their data to the MTC server periodically. Note that time taken to upload their data does not change largely. Data retransmissions do not occur frequently due to the assumption of a LTE access network in which high reliability is ensured.

This section presents a control method designed to prevent communication timings from overlapping (Section IV-A) and a method for providing device triggering in a limited case where the devices support the PSM (Section IV-B). Section IV-C describes additional functions and procedures for realizing the proposed methods on standard cellular network architecture.

A. Method for Controlling Communication Timings

Wireless access networks and the EPC are subject to load fluctuations. The load fluctuations vary the time it takes devices to communicate with the EPC and MTC server (the devices establish a bearer, upload data, and release the bearer). Therefore, to prevent periods of communication from overlapping, a certain time interval between the periods is required. To ensure this, time slots at periodic intervals are introduced in specific IMSIs stored by HSS in the EPC. Note that a time slot is defined as the time period in which a specific IMSI is active. The EPC is enabled to accept attach requests caused by the random switching-on of devices with a specific IMSI while only in time slots, thereby controlling the timings of uploading data following the Attach procedure. Fig. 3 provides an overview of the control method. Note that two devices have the same IMSI (IMSI#1), and EID#1 and EID#2 are assigned to them, respectively (the way of assigning EIDs is described in Section IV-C). The data upload cycle is the period during which the devices upload data. The number of time slots in this cycle is equal to the maximum number of devices sharing a single IMSI. A guard time is inserted between the time slots. The guard time blocks the Attach procedure triggered by a request that arrives late in a time slot from overlapping the next time slot. The duration of each data upload cycle, time slot, and guard time, which depends on IoT services, can be managed by the EPC itself (it is up to the implementation method to choose which nodes manage the parameters) or the MTC server (described in Section IV-C). The time slots have two statuses, "enable" (solid line boxes in Fig. 3)—where the attach request is accepted, and "disable" (dotted line in boxes in Fig. 3)—where the Attach request is rejected. After accepting the attach request from the device (EID#1), the HSS changes the status of the time slot to disable from enable. The disable state continues beyond the next data upload cycle. This reservation process for a time slot is inspired by packet reservation multiple access [24]. After the attach procedure is completed, the device (EID#1) uploads data at intervals of the data upload cycle. In the example of Fig. 3, a device (EID#2) powered up and sends an attach request at the same timing that the device (EID#1) uploads data. In this case, the EPC rejects the attach request due to the disable state, while the EPC accepts a bearer establishment request for uploading data. When the EPC rejects the attach request, the EPC replies with attach reject messages that contain a back-off timer to the device (EID#2). The back-off timer value can be determined using various algorithms. The value can be random for the sake

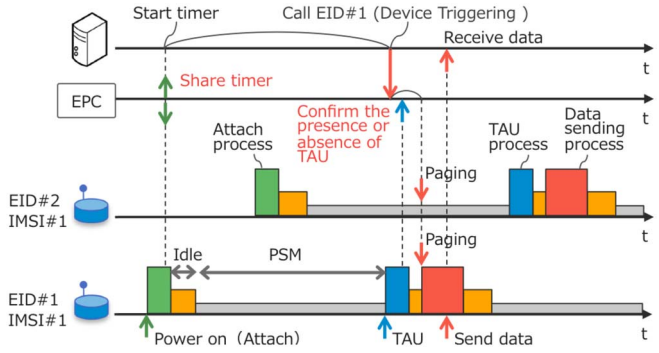

Fig. 4. Overview of device triggering using PSM.

of simplicity, or it can be calculated to determine the time elapsed before the next enable (vacant) time slot (although the calculation makes management cumbersome and complicated). After the back-off timer expires, the device will retry the attach procedure. Attach reject messages containing the back-off timer are also answered when attach request messages arrive at the EPC while in the guard times. The devices repeat this retransmission process until a attach request is accepted. In this way, the EPC controls the timing of the attach by fitting it into the enable time slot, thereby preventing communication timings following the attach from overlapping.

B. Device Triggering Using PSM

This section presents a method for providing device triggering. To change the behavior of the application running on a device or improve the reliability of the application (recover the application suspended due to some sort of failure), the MTC-server-initiated requests need to be delivered to the devices.

In general, assigning the same IMSI to multiple devices makes it difficult to enable device triggering from the MTC server. This is because the devices, which are assigned the same IMSI, respond to paging messages when they reside in the same paging area (the paging message is broadcast within the paging area composed of several eNBs). Provided that all the devices support PSM and their communication timings (non-PSM time) do not overlap, device triggering can invoke a specific device. This is because devices in PSM do not respond to paging messages.

To synchronize device triggering with the timing of the non-PSM, the EPC need to notify the timer (T3324 in Section III-B) to the MTC server. In the standardized procedure, this timer is notified only to a device during the attach procedure. In addition, to eliminate an inconvenience caused by the time lag between when the device and MTC server send messages after the timer expires, a function of buffering the messages need to be introduced to the EPC. Fig. 4 shows an overview of device triggering using the PSM. Note that two devices have the same IMSI (IMSI#1), and EID#1 and EID#2 are assigned to them, respectively (the way of assigning EIDs is described in Section IV-C). After the timer in the MTC server expires, the MTC server sends a data transmission request to the device by specifying its EID (EID#1). The EPC receives this request and subsequently identifies IMSI#1 corresponding to EID#1 and determines if a TAU corresponding to IMSI#1 has been performed. If no TAU exists no later than a short time (for fewer than one seconds), the request is buffered. Otherwise, paging messages are sent to the devices for the request to be transmitted. Although the paging messages reach the devices sharing IMSI#1, only the device (EID#1) that returns from PSM responds to the paging messages, thereby, establishing the data path in the EPC. Subsequently, the device receives the data transmission request and sends data to the MTC server.

C. Applying the Proposed Methods to Standard Architecture

This section presents the procedure for applying the proposed method to the standard architecture shown in Section III-A. In the EPC, HSS, MME, and MTC-IWF need to be enhanced. The MTC server also needs to include new operations. To simplify the management of time slots in the EPC, the MTC server is mainly responsible for the management of time slots. The MTC server activates each EID in turn for a particular period and changes the status of the IMSI, stored in the HSS, in synchronization with this activation. Once an EID is assigned to a device, the MTC server stops changing the status of the IMSI when the EID's turn comes. This eliminates the need for the HSS to manage time slots and each EID, thereby preventing the amount of information in the HSS from increasing. Only the status (enable/disable) of the IMSI and the back-off timer value are added in the HSS.

Note that the back-off timer value is determined using a random number algorithm for the sake of simplicity. There are several well-known back-off algorithms using random number algorithm. Uniform back-off algorithm (hereinafter referred to as the "uniform") selects back-off timers in a constant range. Binary exponential and adaptive back-off algorithms (hereinafter referred to as the "binary exponential" and "adaptive," respectively) select back-off timers in a variable range. In binary exponential, the range is determined at devices and is incremented in a binary exponential manner according the number of retransmissions. By contrast, in adaptive, the range is determined at the EPC and varies depending on the rate of the attach requests. These back-off algorithms are detailed in Section V-A.

Fig. 5 shows the call flow of an attach procedure (modifications to the standard procedure are indicated in red). An MTC server periodically updates the status of an IMSI, which is stored in an HSS, via an MTC-IWF. When a device is powered on, an attach request arrives at an EPC. In the EPC, the HSS checks the status of the IMSI. If the status is disable, an attach reject containing a back-off timer is returned to the device. The device resends the attach request after the back-off timer expires. If the status is enable, the HSS changes the status to disable so that the HSS can reject the attach requests from other devices. After completing the attach, an application in the device notifies the MTC server of the completion of registration. At this time, if the device has a static EID, the device notifies the MTC server about it; otherwise, the device requests the assignment of an EID. The MTC server updates

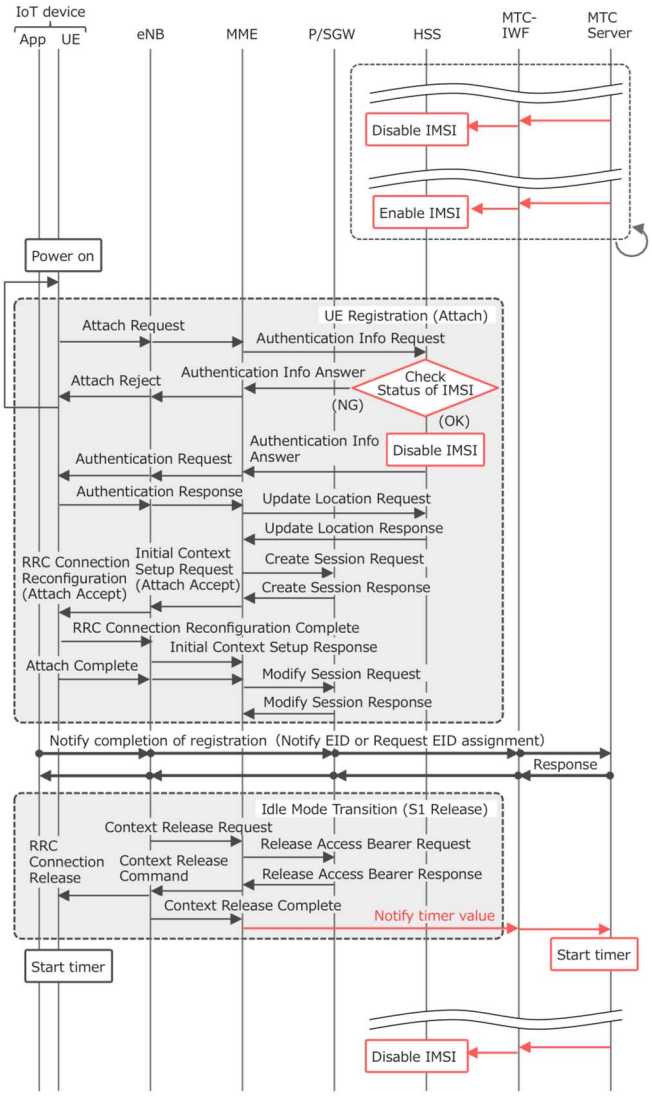

Fig. 5. Call flow in attach procedure.

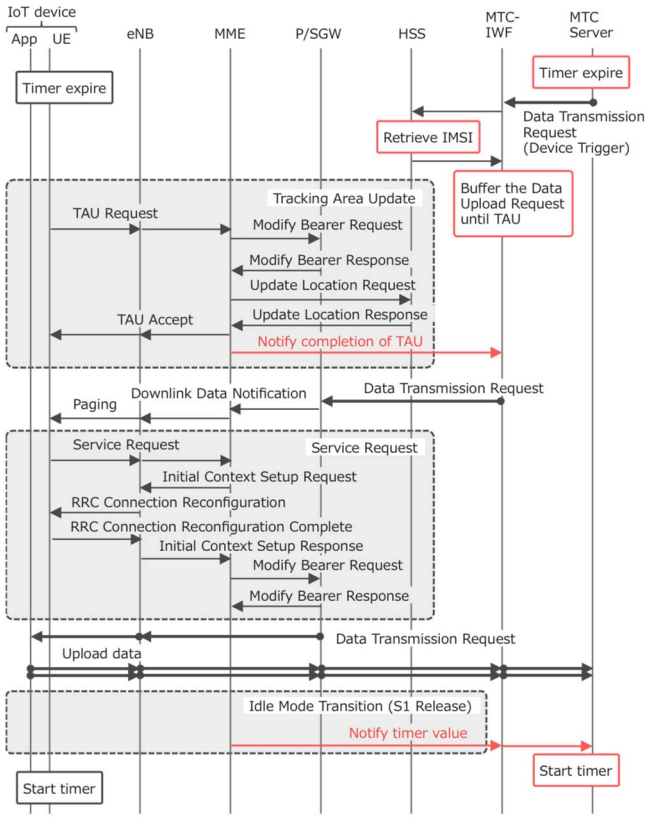

Fig. 6. Call flow in data upload procedure using device triggering.

the binding between the time slot and the EID. The MTC server also stops updating the status of the IMSI in the HSS during the time slot to which the EID is assigned. When the bearer is released in the EPC ("S1 release" in Fig. 5), the MME notifies the MTC server of the timer value (T3324) which the device in the attach is notified of. Note that the eNB starts the procedure of S1 release upon detecting user inactivity using a timer (RRC inactivity timer). Although the period of the timer is usually a few seconds to a few tens of seconds in commercial networks [17]–[19], this period needs to be shortened to increase the number of IoT devices sharing a single IMSI. The MTC server triggers the timer. When this timer expires, the MTC server calls the device using the EID.

Fig. 6 shows the call flow of a data upload procedure using device triggering (modifications to the standard procedure are indicated in red). The MTC server manages each timer of the EID. When a timer expires, the MTC server performs device triggering on the device of the EID corresponding to the timer. The MTC server sends a data transmission request to the device. The request arrives at the MTC-IWF. The MTC-IWF retrieves the IMSI corresponding to the EID, contained in the request, from the HSS. Note that the EID is composed of a group id and device id, and the HSS stores only the correspondence information between the group id and the IMSI. This eliminates the need for every EID that is related to the IMSI to be stored, thereby reducing the amount of information retained in the HSS. After retrieving the IMSI, the MTC-IWF determines the presence of the TAU corresponding to the IMSI. If no TAU exists no later than a short time, the MTC-IWF buffers the data transmission request. When the device returns from the PSM and performs the TAU, the MME notifies the MTC-IWF of the completion of the TAU. The MTC-IWF then transfers the buffered request to the device. The request goes through a PGW and reaches an SGW. To re-establish the eNB-to-SGW tunnel for transferring the request to the device, paging messages are broadcast. The device receiving the paging messages re-establishes the data path, receives the data transmission request, and uploads data to the MTC server. After uploading, the device releases the eNB-to-SGW tunnel. This behavior is repeated periodically.

V. Simulation-Based Evaluation

This section evaluates the effectiveness of sharing a single IMSI using packet level simulations. In the proposed method, an appropriate value for the guard time needs to be configured to prevent communication timings from overlapping. In addition, the back-off algorithms and the time slot values have an effect on the number of retransmitted attach requests, resulting in variations in the amount of time spent on attach completion and load operations on the EPC.

First, we determined the value for the guard time taking the background traffic into consideration (Section V-B). Then, we compared the back-off algorithms (Section V-C). Finally, we evaluated the number of devices sharing the single IMSI at different time slot values (Section V-D).

A. Simulation Setup and Parameters

We implemented a discrete-event driven simulation of the packet delivery process using the queuing network system in C++ (OMNeT++) [20]. The call flows were simulated as shown in Figs. 5 and 6. Each node in the EPC was modeled as a queuing server. The queuing servers identifies received messages and transmit them to the next queuing servers. Thus, messages go through the queuing servers according to the call flows shown in Figs. 5 and 6. Messages contained only the parameters required for this evaluation. Realistic processes at each node were substituted for processing delays. To simplify the network configuration of the simulation, all devices were connected to a single eNB. The communication pattern of the device was assumed such that the device recorded the sensor's data (100 byte) in intervals of one second and uploaded the data (called "sensing data" hereafter) to a MTC server at half-hour intervals. This communication pattern is categorized as a high-frequency IoT service [1]. Although there were several communication patterns, we focused only on the high-frequency IoT service to evaluate the performance limitations of the proposed method. Note that the proposed method is designed on the assumption that a single IMSI is shared among the same communication patterns (as described in Section IV). That is, IoT devices in a different communication pattern are assigned different IMSIs. Consequently, we assumed that there is only one communication pattern in order to evaluate the number of IoT devices sharing a single IMSI. The data transmission speed was assumed to be 1 Mb/s and took about 1.44 s to upload the sensing data.

The measurements were carried out with varying the number of devices powered-on (the number of devices sharing a single IMSI). The power-on timings assumed the cases where the devices were uniformly powered on for 60 or 1200 s (except for the evaluation of the guard time in Section V-B).

The simulation was conducted 1000 times with different random seeds (in general, the precision error becomes a few percent) for individual power-on patterns and varying the number of devices that were powered on. The parameter settings used in our simulation are listed in Table III. The processing times and link delay times were determined with reference to [16].

We evaluated the following back-off algorithms.

1) *Uniform:* All back-off timers are chosen in the range $[0, M]$, where M is the maximum back-off range. In the evaluations, we choose $M = 60$ and $M = 120$.
2) *Binary Exponential:* The back-off timer is uniformly distributed in the range $[0, 2^{i-1}w]$, where w is the initial back-off range and i is the number of attach reject received by the device. This means that the back-off range is incremented in a binary exponential manner according to the number of rejections. In the evaluations, we chose $w = 30$ and put a cap on the maximum back-off range ($i \leq 4$).
3) *Adaptive:* The back-off time is uniformly distributed in the range $[0, r(T_S + T_G)]$, where T_S is time slot value, T_G is guard time value, and r is the number of attach requests received by the EPC during T_S+T_G. This means that the back-off range changes depending on the rate of attach requests. In the evaluations, the rate was calculated per $T_S + T_G$ seconds or per the timing when the EPC received ten attach requests if the number of requests was less than ten during $T_S + T_G$.

We evaluated performance using the following two metrics: 1) maximum time spent on attach completion and 2) attach reject messages per unit time.

TABLE III
PARAMETER SETTINGS

Parameters	Values	
Number of devices	100 – 240	
RRC inactivity timer	2 s	
Processing time	*C-plane*	*U-plane*
Device	0.004 s	0.004 s
eNB	0.004 s	0.0015 s
MME	0.004 s	-
SGW, PGW	0.004 s	0.001 s
HSS	0.004 s	-
MTC-IWF	0.004 s	0.001 s
MTC server	-	0.004 s
Link delay time (wireless)	0.001 s	
Link delay time (eNB-SGW)	0.0075 s	
Link delay time (in the EPC)	0.001 s	
Link delay time (EPC-MTC server)	0.001 s	

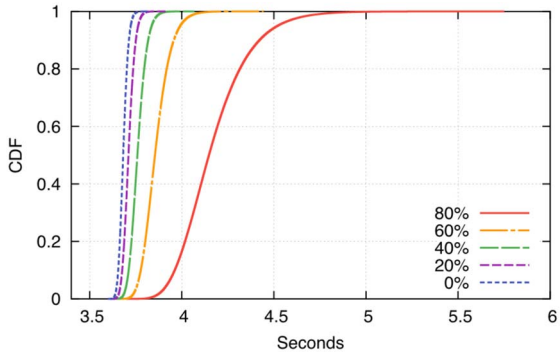

Fig. 7. Occupation time of IMSI in uploading data. Percentages refer to the utilization of each node in the EPC.

B. Evaluation of Guard Time

The guard time needs to be longer than the time required for the devices to upload data. This time includes not only the time spent uploading data but also the time taken to establish communication tunnels. The latter time is influenced by the load of the EPC.

In this simulation, background packets were transmitted to each node (eNB, MME, and S/PGW) in the EPC. The sending rates were changed, with the result that the utilization of each node was from 0 to 80%. One hundred devices uploaded

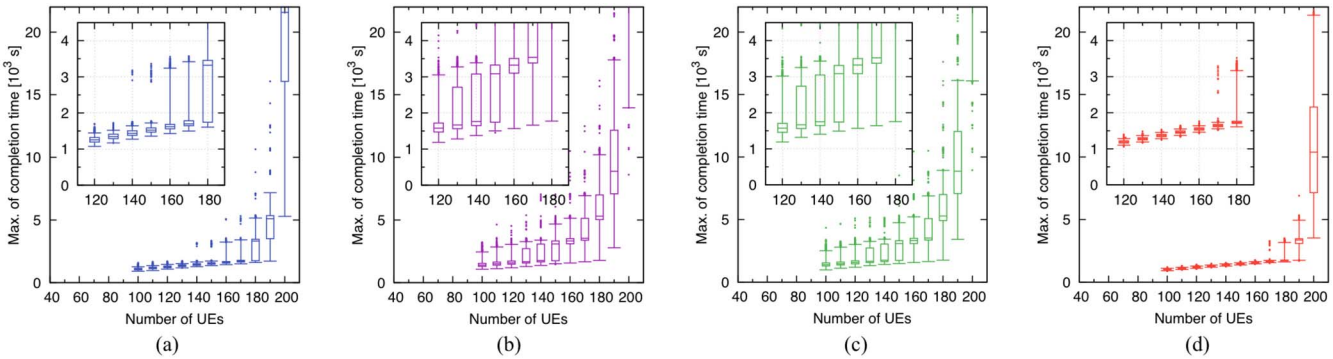

Fig. 8. Maximum time of attach completion in the case where the devices were powered on during a short power-on period (60 s) in the back-off algorithms. (a) Uniform [0 s, 60 s]. (b) Uniform [0 s, 120 s]. (c) Binary exponential. (d) Adaptive.

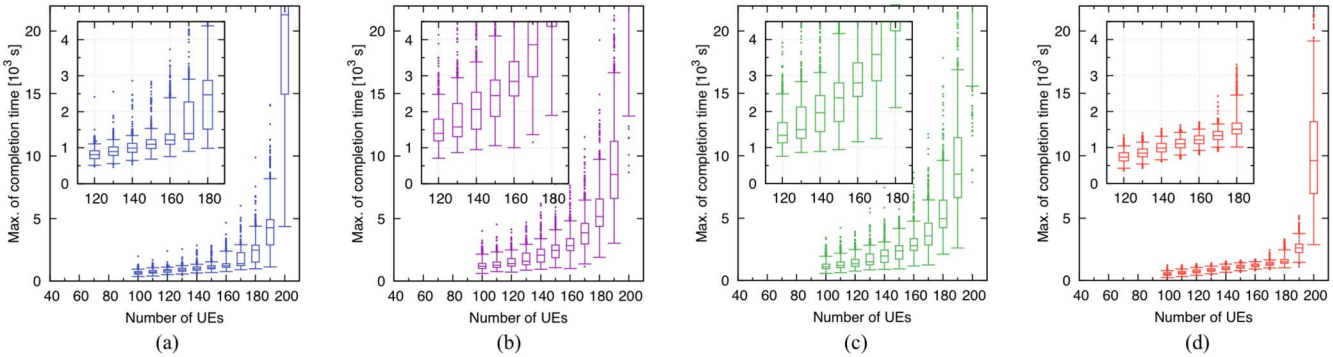

Fig. 9. Maximum time of attach completion in the case where the devices were powered on during a long power-on period (1200 s) in the back-off algorithms. (a) Uniform [0 s, 60 s]. (b) Uniform [0 s, 120 s]. (c) Binary exponential. (d) Adaptive.

data in turn at ten-second intervals. The time that elapsed from "TAU request" to "notify timer value" (see Fig. 6) was measured.

Fig. 7 shows the occupation time of the IMSI in uploading data. In our assumed IoT service, a six-second guard-time is enough. This is because the EPC restricts requests when the utilization of the node becomes 60∼80% in actual operation [21]–[23]. In the following simulations, the guard time was configured to 6 s.

C. Comparison of Back-off Algorithms

In the comparison of back-off algorithms, the time slot was set to three seconds. This configuration ensured that a single IMSI is occupied by each device for 9 s (equal to the sum of the time slot and guard time) in a half-hour period (1800 s), that is, a single IMSI can be shared among 200 devices using time division.

Figs. 8 and 9 show the maximum time of attach completion in each back-off algorithm. The data are displayed as a boxplot, showing the median value (line), interquartile range (boxes), and 5%–95% percentile (whiskers), and outliers (dots). The maximum time increased exponentially beyond the ratio of the number of devices sharing the IMSI to the maximum number (200 devices), although the ratio differed depending on the back-off algorithms and the power-on patterns. The difference among back-off algorithms showed a similar tendency in the case where the devices were powered on during a short power-on period (Fig. 8) and also where there

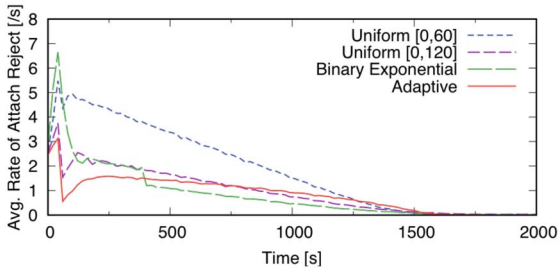

Fig. 10. Attach reject rate per unit time in the case where 160 devices were powered on during a short power-on period (60 s).

was a long power-on period (Fig. 9). Adaptive back-off algorithms made the maximum time of attach completion smaller than other algorithms [Figs. 8(d) and 9(d)]. Although the uniform back-off algorithm, where the range of random values of the back-off timer is small [Figs. 8(a) and 9(a)], reduced the maximum time to a certain extent, the small back-off timer caused the load on the EPC to increase (this is shown in the next paragraph). The binary exponential back-off algorithms was ineffective in reducing the maximum time. The maximum time of attach completion was a function of the range of the back-off timer values.

Figs. 10 and 11 show the number of attach reject per unit time in each back-off algorithm when 160 devices share a single IMSI. Under the condition that the devices were powered on during a long period (see Fig. 11), the back-off algorithms had little effect on the attach reject rate. In contrast, under

TABLE IV
ANALYTICAL COMPARISON OF BACK-OFF ALGORITHMS

Back-off Algorithms	Maximum of completion time	Attach reject rate	Other characteristics
Uniform	Depends on maximum back-off range (T_{max})	Becomes high as making the maximum of completion time short and conversely	Simple to implement
Binary Exponential	Becomes long due to the fact that T_{max} is increased by retransmissions	Becomes least due to the fact that T_{max} is increased by retransmission	Need to count Attach reject at IoT devices
Adaptive	Becomes short due to the fact that T_{max} is decreased when the Attach request rate is low	Adjusted by the Attach request rate	Need to figure out the Attach request rate for each IMSIs at EPC

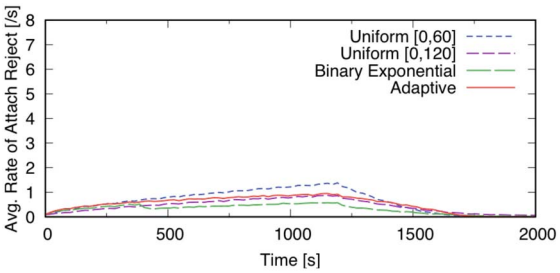

Fig. 11. Attach reject rate per unit time in the case where 160 devices were powered on during a long power-on period (1200 s).

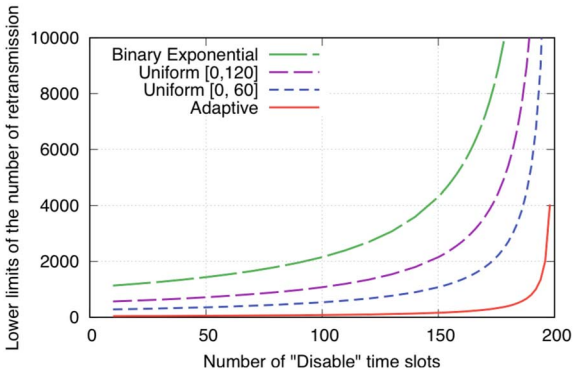

Fig. 12. Lower limits of the number of retransmissions to the number of disable time slots.

the condition that the devices were powered on during a short period (see Fig. 10), the back-off algorithms influenced the attach reject rate. The uniform back-off algorithm, where the back-off range is small, caused a high attach reject rate while the adaptive back-off algorithm decreased the attach reject rate.

In order to analytically compare the maximum time of attach completion in each back-off algorithm, we express the probability that the request from the last device arrives within an enable (vacant) time slot. Let $T_{\max}$ be the maximum back-off range, l_s be the duration of a time slot, l_g be the duration of a guard time, N_s be the number of time slots per data upload cycle, and N_d be the number of disable time slots per data upload cycle. Thus, the probability of request occurrence is $2/T_{\max}$, the probability that requests arrives within a time slot is $l_s/(l_s + l_g)$, and the probability of an enable time slot is $(N_s - N_d)/N_s$. The probability that requests arrive within an enable time slot, p, can be expressed as

$$p = \frac{2}{T_{\max}} \cdot \frac{l_s}{l_s + l_g} \cdot \frac{N_s - N_d}{N_s}. \tag{1}$$

The probability of not completing the attach procedure after n retransmission, q, can be given by

$$q = (1 - p)^n. \tag{2}$$

We evaluate n when q is lower than threshold α. The number of retransmissions (n) to satisfy $q \leq \alpha$ is expressed as

$$n \geq \frac{\log \alpha}{\log(1 - p)}. \tag{3}$$

Obviously, n increases as $T_{\max}$ is incremented. The values of $T_{\max}$ depend on the back-off algorithms. To evaluate the optimal number of devices sharing a single IMSI, we obtain lower limits of the number of retransmissions (n) for the number of disable time slots (N_d) from (3) under the condition $q \leq 0.05$. Fig. 12 shows the relation of lower limits of n to N_d. The lower limits of n increases exponentially beyond a certain value of N_d. Thus, it is desirable to share a single IMSI among the number of devices corresponding to this certain value.

Table IV shows analytical comparisons of the back-off algorithms, including a summary of above evaluations.

The adaptive back-off algorithm can be of some benefit in preventing the number of attach rejects from increasing in some circumstances. The attach reject causes an increase in the number of signaling messages (from "attach request" to "attach reject" as shown in Fig. 5). The condition wherein the devices were powered on during a short period will increase the load on the EPC. To reduce the impact on the EPC as much as possible, it will be necessary to use the adaptive back-off algorithm instead of the uniform or binary exponential back-off algorithm.

D. Impact of Time Slot on the Number of Devices Sharing IMSI

The number of devices sharing a single IMSI was evaluated in the case of where the adaptive back-off algorithm was used. The guard time was 6 s and the time slot value was varied.

Figs. 13 and 14 show the maximum time of attach completion for various time slots. The dotted lines are the average values and the fill areas are the range of 1–99th percentile. The short time slot reduced the maximum time where the

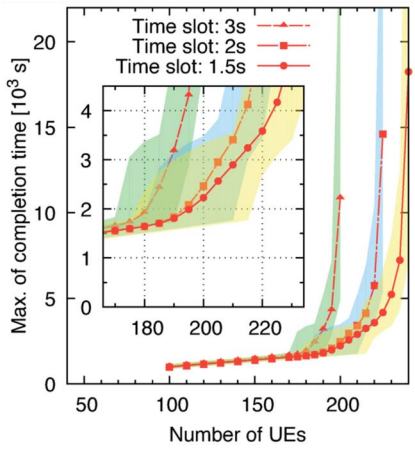

Fig. 13. Maximum time of attach completion in the case where the devices were powered on during a short power-on period (60 s).

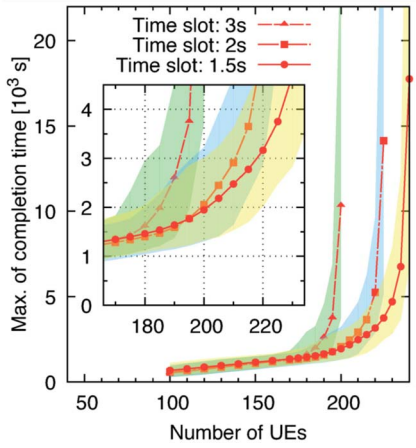

Fig. 14. Maximum time of attach completion in the case where the devices were powered on during a long power-on period (1200 s).

devices were powered on during a short power-on period and also during a long power-on period. This is a result of increasing the maximum number of devices sharing the IMSI. The impacts of reducing the time slot on the EPC are shown in Figs. 15 and 16. The difference between curves in Figs. 15 and 16 is induced by changes in the back-off range in the adaptive back-off algorithm. In the case where the devices were powered on for a short power-on period (60 s), attach requests are initially sharply concentrated. This results in a long back-off range, inducing a low rate of attach requests. The back-off range, in turn, becomes short, resulting in a high rate of attach requests. In the case where the power-on period is long (1200 s), there is no concentration of attach requests. This does not cause large fluctuations in the back-off range. In this situation, the rate declines after the power-on period as shown in Fig. 16. Varying the time slot had little effect on the attach reject rate regardless of the scenario (power-on period and the number of devices powered-on). This is because although the duration of time slots becomes short, the number of time slots increases.

We evaluate the tradeoff between the number of devices sharing a single IMSI and the duration of a time slot using (3) in the previous section (Section V-C). Fig. 17 shows the relation of lower limits of n to N_d under the condition $q \leq 0.05$. In the case where the number of IoT devices sharing a single IMSI is not large, it is desirable to make a time slot longer.

Considering the battery life of the devices, it is preferable for the devices to complete the attach and activate the PSM as soon as possible after powering-on. For example, assuming that the expected time of attach completion is 3600 s with a probability of 99%, the sharing of the IMSI among approximately 200 devices meets the expectation for any power-on pattern when the time slot is less than or equal to 2 s, as can be seen from Figs. 13 and 14. This result shows that the proposed method can reduce the amount of state information in the EPC by two orders of magnitude.

E. Discussion

Table V shows analytical comparisons between our proposed method and the previous method by which devices communicate with the EPC through the devices selected as gateway (hereinafter referred to as the "GW-based method"). In the GW-based method, devices require strategies appropriate to the device-side situation or the number of aggregating communication lines decreases. In the case of the IoT service targeting moving devices, the GW-based method needs to select appropriate devices as gateways to increase the opportunities for moving devices to obtain the connectivity using gateway devices. The GW-based method also needs to determine the number of gateways according to device density. In an area that is not densely packed with IoT devices, the number of gateway devices is determined by taking the limitation of communication distances in to account. On the other hand, in an area that is densely packed with IoT devices, the limitation of the number of devices that can be managed by one gateway device is an important factor in determining the number of gateways. In addition, the loads on the gateway devices need to be considered.

In the proposed method, the network (the EPC and the MTC server) is responsible for management. For moving devices, the proposed method does not need additional functions since the EPC inherently supports mobility management. The strategy for dealing with device density relies on area designs (capacity and coverage of eNB). For load fluctuations in wireless access networks or the EPC, the MTC server plans and determines the durations of time slots and guard times, which affect the number of aggregating communication lines. The presumption of large load fluctuations leads to a longer time slot and guard time, thereby reducing the number of devices sharing a single IMSI. When load fluctuations are restrained by constructing a dedicated network using virtualization technologies, the number of devices sharing a single IMSI can be increased. In this paper, the time slot and guard time were set based on the assumption that the EPC had a certain load. To evaluate the performance of the proposed method in detail, it will be necessary to evaluate the load tolerance by varying the durations of the time slot and guard time.

TABLE V
ANALYTICAL COMPARISON

Methods	Handling for improvement of aggregation			Limitations
	Moving devices	Device density	Load	
GW-based	Mobility management is required on the device side	The limitation of distance and gateway capacity need to be considered	The load on the gateway needs to be considered	· Difficult to prolong battery life
Proposed	Mobility management is inherently supported in the EPC	Relies on area design of base stations	MTC server needs to determine time slot and guard time	· Load due to retransmission · Only devices under the same IoT service

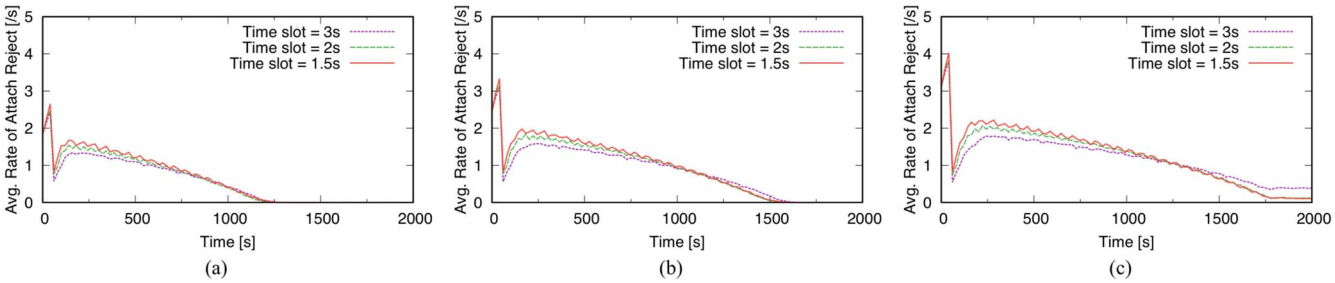

Fig. 15. Attach reject rate per unit time in the case where the devices were powered on during a short power-on period (60 s). (a) UE = 120. (b) UE = 160. (c) UE = 200.

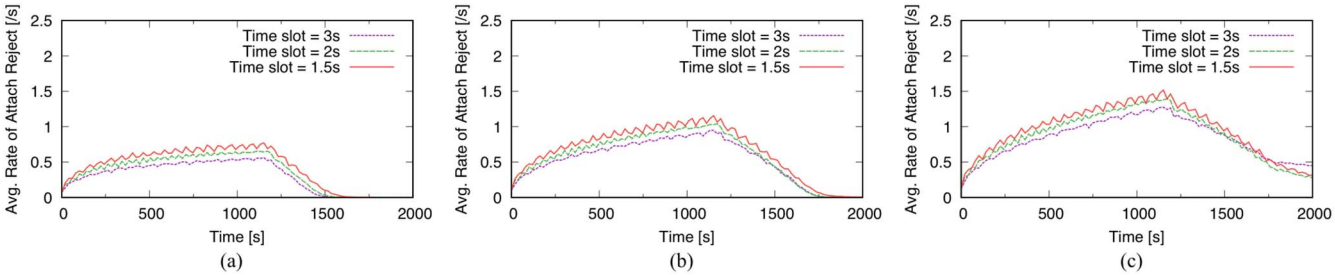

Fig. 16. Attach reject rate per unit time in the case where the devices were powered on during a long power-on period (1200 s). (a) UE = 120. (b) UE = 160. (c) UE = 200.

Considering the impact on devices or the EPC, both methods have certain limitations. The GW-based method makes it difficult to prolong battery life of devices using PSM and requires engineering for selecting appropriate gateway devices. On the other hand, the proposed method increases the load on the EPC to some extent. This is because the proposed method may result in the retransmission of the attach request, which is sent from the powered-on devices. In addition, the proposed method is intended only for devices under the same IoT services.

Table VI shows a numerical comparison of the amount of state information in the EPC for a IoT service used to track bicycles. The number of bicycles in a dense zone (Tokyo, Japan: 13 500 km^2) and in a nondense zone (Hokkaido, Japan: 83 500 km^2) is 9 000 000 (666.7/km^2) and 2 800 000 (33.5/km^2), respectively [14]. The communication pattern for uploading location information of bicycles is the same as the pattern used in the simulation (as shown in Section V-A). For numerical calculations with the GW-based method, completely optimal situations were assumed. That is, it was assumed that each device selected as a gateway was placed in an area of 1.4 km mesh (note that the maximum distance between the gateway and devices is 1 km) so that all devices communicated with the EPC via the gateway devices. In the dense zone, multiple gateway devices were placed in one area due to the limitation on the number of devices that can be managed by one gateway device (150 devices [15]). In the proposed

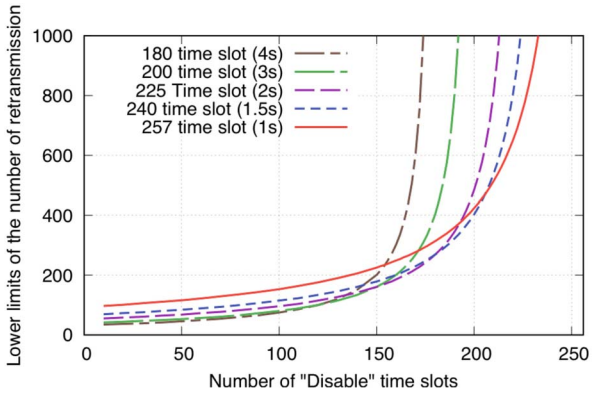

Fig. 17. Lower limits of the number of retransmissions to the number of disable time slots while varying the duration of a time slot.

TABLE VI
NUMERICAL COMPARISON IN THE AMOUNT OF STATE INFORMATION IN BICYCLE TRACKING SERVICE

Methods	Amount of state information in the EPC		Remarks
	Dense zone	Non-dense zone	
Current EPC	9000 K [context]	2800 K [context]	-
GW-based	60 K [context]	42.5 K [context]	No. of devices per GW: 150 [15] Communication distance: 1 km
Proposed	45 K [context]	14 K [context]	No. of device sharing an IMSI: 200 (from the simulation results)

method, the number of devices sharing a single IMSI was assumed to be 200 devices, which was obtained from the simulation results. This comparison shows that the proposed method can reduce the amount of state information as well as or better than the GW-based method in some situations.

VI. CONCLUSION

To reduce the amount of state information in the EPC by aggregating communication lines in the EPC without gateway devices, this paper proposed a method that assigns the same IMSI to multiple IoT devices and prevents the communication timings from overlapping. Our simulations showed that the proposed method provided cellular communication lines for 200 or more IoT devices using only a single IMSI, thereby reducing the amount of state information in the EPC to a significant degree.

The proposed method can reduce the amount of state information significantly without device-supported planning and engineering for aggregating communication lines. Therefore, the proposed method can help mobile network operators provide a large number of IoT devices with cellular connectivity on limited resources. This may bring about benefits especially for mobile virtual network operators that put a high priority on reducing CAPEX/OPEX to achieve low-cost communications.

In future research, we will evaluate the feasibility and performance of our method using a prototype implementation. In addition, we will study the reduction of state information by using our method in a realistic IoT service.

REFERENCES

[1] T. Rebbeck, M. Mackenzie, and N. Afonso, "Low-powered wireless solutions have the potential to increase the M2M market by over 3 billion connections key messages overall LPWA opportunity," Analysys Mason Limited, London, U.K., Tech. Rep. 5177472, 2014.

[2] "Architecture enhancements to facilitate communications with packet data networks and applications," Doc. 3GPP, TS 23.682 v12.0.0, 2013.

[3] *Nokia LTE M2M Optimizing LTE for the Internet of Things*, Nokia, Helsinki, Finland, 2014.

[4] *LTE Release 13*, Ericsson, Stockholm, Sweden, 2015.

[5] V.-G. Nguyen, T.-X. Do, and Y. Kim, "SDN and virtualization-based LTE mobile network architectures: A comprehensive survey," *Wireless Pers. Commun.*, vol. 86, no. 3, pp. 1401–1438, Feb. 2016.

[6] K. Samdanis, A. Kunz, M. I. Hossain, and T. Taleb, "Virtual bearer management for efficient MTC radio and backhaul sharing in LTE networks," in *Proc. IEEE 24th Int. Symp. PIMRC*, London, U.K., Sep. 2013, pp. 2780–2785.

[7] M. Ito, N. Nishinaga, Y. Kitatsuji, and M. Murata, "Aggregating cellular communication lines for IoT devices by sharing IMSI," in *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, May 2016.

[8] S. Sakurai, G. Hasegawa, N. Wakamiya, and T. Iwai, "Performance evaluation of a tunnel sharing method for accommodating M2M communication to mobile cellular networks," in *Proc. IEEE Globecom Workshops*, Atlanta, GA, USA, Dec. 2013, pp. 157–162.

[9] Y. Kitatsuji and H. Yokota, "On-demand session establishment for all IP-based mobile networks," in *Proc. 6th EURO-NF Conf. Next Generation Internet*, Paris, France, Jun. 2010, pp. 1–8.

[10] J. Kaippallimalil and H. A. Chan, "Network virtualization and direct Ethernet transport for packet data network connections in 5G wireless," in *Proc. IEEE Glob. Commun. Conf.*, Austin, TX, USA, Dec. 2014, pp. 1836–1841.

[11] S. Matsushima and R. Wakikawa, "Stateless user-plane architecture for virtualized EPC (vEPC)," Internet draft, draft-matsushima-stateless-uplane-vepc-04, Sep. 2015.

[12] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: Scalable and flexible cellular core network architecture," in *Proc. 9th ACM Conf. Emerg. Netw. Exp. Technol.*, Santa Barbara, CA, USA, Dec. 2013, pp. 163–174.

[13] "Architecture enhancements to facilitate communications with packet data networks and applications," 3GPP TS 23.682 v12.2.0, 2014.

[14] "Bicycle holdings trend survey," Japan Bicycle Promotion Inst., Tokyo, Japan, Tech. Rep., Mar. 2008.

[15] "Study on LTE device to device proximity services (ProSe)—Radio aspects, " Doc. 3GPP TR 36.843 v1.0.0, Nov. 2013.

[16] "Feasibility study for further advancements for E-UTRA (LTE-Advanced)," Doc. 3GPP TR 36.912 v12.0.0, Sep. 2014.

[17] "LTE RAN enhancements for diverse data applications," 3GPP TR 36.822 v11.0.0, Sep. 2012.

[18] J. Huang *et al.*, "A close examination of performance and power characteristics of 4G LTE networks," in *Proc. MobiSys*, Florence, Italy, 2012, pp. 225–238.

[19] M. Siekkinen, M. A. Hoque, J. K. Nurminen, and M. Aalto, "Streaming over 3G and LTE: How to save smartphone energy in radio access network-friendly way," in *Proc. 5th Workshop MoVid*, 2013, pp. 13–18.

[20] (May 20, 2014). *OMNeT++*. [Online]. Available: http://www.omnetpp.org

[21] "NEC virtualized evolved packet core—vEPC," NEC Corporation, Tokyo, Japan, White Paper TE-524262, 2014.

[22] *Nokia Networks Telco Cloud is on the Brink of Live Deployment*, Nokia Solutions and Netw., Helsinki, Finland, 2013.

[23] *Implement Overload Protection for Gateways and Neighboring Network Elements on the ASR5x00 Series*, document 119196, Cisco, San Jose, CA, USA, 2015.

[24] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, "Packet reservation multiple access for local wireless communications," *IEEE Trans. Commun.*, vol. 37, no. 8, pp. 885–890, Aug. 1989.

Manabu Ito received the B.E. degree in electronics, information, and communications engineering, and the M.E. degree in global information and telecommunication studies from Waseda University, Shinjuku, Japan, in 2006 and 2008, respectively. He is currently pursuing the Ph.D. degree in information science at Osaka University, Suita, Japan.

He was an Associate Research Engineer with the Mobile Network Laboratory, KDDI Research and Development Laboratories, Inc., Saitama, Japan, from 2008 to 2012. He is currently an Expert Researcher with the National Institute of Information and Communications Technology, Koganei, Tokyo, Japan. His current research interests include fast handover management in IP multimedia subsystems, architecture of evolved packet cores, and virtualization of servers and networks.

Mr. Ito is a Member of the Institute of Electronics, Information, and Communication Engineers.

Nozomu Nishinaga received the B.S. and M.S. degrees in electronics engineering and the Ph.D. degree in information engineering from Nagoya University, Nagoya, Japan, in 1994, 1996, and 1998, respectively.

He was a Research Assistant with the Information Media Education Center, Nagoya University, from 1998 to 1999. Since 1999, he has been a Researcher with the National Institute of Information and Communications Technology (NICT), Koganei, Tokyo, Japan. Since 2011, he has been the Director of the New Generation Network Laboratory, Network Research Headquarters, NICT. His current research interests include Internet architecture and wireless communications.

Dr. Nishinaga is a Member of the Institute of Electrical, Information, and Communication Engineers.

Yoshinori Kitatsuji received the B.S. and M.S. degrees in engineering science from Osaka University, Suita, Japan, in 1994 and 1997, respectively, and the D.E. degree in information engineering from the Kyushu Institute of Technology, Kitakyushu, Japan, in 2007.

He has been with KDDI Research and Development Laboratories, Inc., Saitama, Japan, since 1997, where he is currently a Group Leader with the Mobile Network Laboratory. From 2004 to 2006, he was a Research Fellow of the JGN-2 Project funded by the National Institute of Information and Communications Technology, Koganei, Tokyo, Japan. His current research interests include mobile network and network virtualization architecture, traffic engineering, and network measurement.

Dr. Kitatsuji is an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION AND SYSTEMS. He is a Senior Member of the Institute of Electrical, Information, and Communication Engineers.

Masayuki Murata (M'89) received the M.E. and D.E. degrees in information science and technology from Osaka University, Suita, Japan, in 1984 and 1988, respectively.

In 1984, he joined the Tokyo Research Laboratory, IBM Japan, Tokyo, Japan, as a Researcher. From 1987 to 1989, he was an Assistant Professor with the Computation Center, Osaka University, where he joined the Department of Information and Computer Sciences, Faculty of Engineering Science, in 1989, was an Associate Professor with the Graduate School of Engineering Science, from 1992 to 1999, has been a Professor, since 1999, and has been with the Graduate School of Information Science and Technology, in 2004. He has authored or co-authored over 300 papers in international and domestic journals and conferences. His current research interests include computer communication networks, performance modeling, and evaluation.

Prof. Murata is a Fellow of the Institute of Electrical, Information, and Communication Engineers. He is a Member of the Association for Computing Machinery, the Internet Society, and the Information Processing Society of Japan.