IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 3, 2021, accepted March 3, 2021, date of publication March 8, 2021, date of current version March 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3064321

Small-Cell Assisted Group Paging for Massive MTC in LTE Networks: Design and Analysis

ANH-TUAN H. BUI[1], (Graduate Student Member, IEEE), CHUYEN T. NGUYEN[2], TAKAFUMI HAYASHI[3], (Senior Member, IEEE), AND ANH T. PHAM[1], (Senior Member, IEEE)

[1]Computer Communications Laboratory, The University of Aizu, Aizuwakamatsu 965-8580, Japan
[2]School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi 100000, Vietnam
[3]Department of Computer Science, Nihon University, Koriyama 963-8642, Japan

Corresponding author: Anh-Tuan H. Bui (pham@u-aizu.ac.jp)

This work was supported by the Japanese Society for the Promotion of Science's Grants-in-Aid Scientific Research under Grant 18K11269.

ABSTRACT Long-Term Evolution cellular networks are the main enabler for the massive Machine-Type Communications service and therefore must handle a large number of Machine-Type Devices (MTDs). To control the number of devices allowed to contend on the Physical Random Access Channel (PRACH), the group paging scheme that divides the MTDs into smaller groups and lets the network sequentially trigger the groups has been studied. However, as the number of PRACH preambles is limited, a group's size must be kept relatively small compared to the MTD population. This paper exploits the possibility that a significant portion of the MTDs is also covered by densely deployed small-cells such that a Small-cell Base Station (SBS) may act as a representative for its MTDs during the preamble transmission step to reduce the load on PRACH. Once the SBS succeeds, its MTDs then contend locally to send their own signaling messages on the corresponding reserved uplink resources. Computer simulations show that the manageable group size can be significantly increased at a reasonable cost on the Physical Uplink Shared Channel. A theoretical model to quickly predict the effect of the ratio of MTDs that are under the coverage of the SBSs is also derived and verified.

INDEX TERMS Group paging, LTE, massive Machine-Type Communications, random access protocols, small cells.

I. INTRODUCTION

Fifth-generation access networks are expected to offer three major services covering a multitude of applications in both human-centric and machine-centric domains. The services, their requirements, and some example applications are together depicted in Fig. 1. Among the three, the enhanced Mobile Broadband (eMBB) and the ultra Reliable Low-Latency Communications (uRLLC) services are going to be handled by the state-of-the-art New Radio (NR) access technology specifically designed to meet their demands [1]. The massive Machine-Type Communications (mMTC) service characterized by billions of ubiquitous Machine-Type Devices (MTDs), on the contrary, will be supported by the existing Long-Term Evolution (LTE) cellular networks that have matured in terms of geographical coverage and market adoption. Nevertheless, the LTE standard is originally designed for human-centric communications with at most a few hundreds of high-rate users per cell. The integration of a massive number of low-rate MTDs into LTE networks can, therefore, result in a random access overload issue described below. All abbreviations used henceforth are summarized in Table 1 for convenience.

The associate editor coordinating the review of this manuscript and approving it for publication was Engang Tian.

A. RANDOM ACCESS OVERLOAD ISSUE

When an idle LTE device needs to access the network, it randomly selects and sends one among a set of orthogonal preamble sequences over a radio channel specifically reserved for the random access purpose, i.e., the Physical Random Access Channel (PRACH). Upon successfully decoding the preamble, the evolved NodeB (eNB) accordingly 1) reserves a Resource Block (RB) on the Physical Uplink Shared Channel (PUSCH) for the device to send the first actual message of the connection establishment process (also known as Msg3) and 2) responds with a Random Access Response (RAR) informing the device about such uplink resource grant and the timing adjustment that the device needs

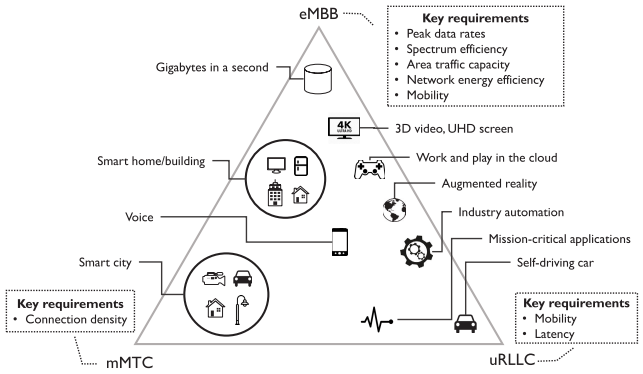


FIGURE 1. Three use cases identified in IMT-2020 vision, based on [2].

TABLE 1. The list of abbreviations used in the paper.

Abbreviations	Meaning
3GPP	Third-generation Partnership Project
CH, CM	Cluster head, Cluster member (respectively)
DQ	Distributed Queue
eNB	evolved NodeB (an LTE base station)
GID, GP	Group's ID, Group paging (respectively)
HARQ	Hybrid automatic repeat request
IMSI	International Mobile Subscriber Identity
mcMTD, scMTD	macro-cell MTD, small-cell MTD (respectively)
PI, PM, PO	Paging interval, Paging message, Paging occasion (respectively)
MME	Mobility Management Entity
PRACH	Physical random access channel
PUSCH	Physical uplink shared channel
RA	Random access
RAO	Random access opportunity (a subframe where the PRACH appears)
RAP	Random access procedure
RAPID	Random access preamble ID
RAR	Random access response
RB	Resource block
SBS	Small-cell base station
SIB	System Information Block
TC-RNTI	Temporary cell-radio network temporary identifier

to apply to properly transmit on the reserved RB. Due to the randomness of the preamble selection, however, multiple devices may send the same preamble in the same Random Access Opportunity (RAO) and cause a preamble collision. In such case, the BS cannot decode the preamble [3] and will not send back an uplink grant during the RAR window. The devices involved in the collision must, therefore, repeat the preamble step after backing off for a random period. A device that undergoes a predefined number N_{PTmax} of consecutive preamble transmission failures will withdraw and is considered "blocked" from the network. This process is known as the contention-based Random Access Procedure (RAP).

Since the LTE technology was designed under an assumption of relatively few users per cell, the PRACH's bandwidth is fixed at 6 RBs (1.08 MHz), from which only up to 64 usable preamble sequences can be created. Furthermore, collisions between the devices contending over this limited number of preambles are resolved by a simple random-backoff protocol. These design choices, while reasonable for human-centric communications, are obviously not favorable to the mMTC service whose expected device density is 1 million MTDs per square kilometer [2]. The studies conducted by both the Third Generation Partnership Project (3GPP) and the literature [3]–[5] in fact show that when tens of thousands of MTDs arrive to the network in a bursty manner, the PRACH is overloaded and most of the MTDs are blocked after experiencing N_{PTmax} consecutive preamble collisions.

It is emphasized that the overload issue is particularly severe for the PRACH and not the PUSCH because of the fact that the number of PRACH resources (preambles) is limited. The PUSCH, on the contrary, has a very wide bandwidth. Furthermore, the MTDs must use the PRACH resources in a contention-based manner. Once an MTD has successfully sent a preamble, however, it becomes synchronized in the uplink direction and also uniquely identifiable by the eNB, and can thus be addressed and scheduled without any contentions on the PUSCH.

B. RELATED WORKS

1) PUSH- VS. PULL-BASED SOLUTION

There is a rich literature focusing on mitigating the PRACH overload issue. All proposals thus far can be largely categorized into push-based and pull-based solutions, respectively, based on whether the random access traffic on PRACH is generated by the MTDs or the network. Push-based solutions assume that the MTDs proactively initiate the RAP while the network only tries to control the resultant PRACH traffic load in a reactive manner, and are thus suitable for event-driven mMTC applications. The canonical push-based solution is the Access Class Barring (ACB) scheme officially implemented in the LTE specifications. In the scheme, a device that needs to access the network can only initiate the RAP if it passes a probabilistic test. Otherwise, the device is barred for a random period before it can retake the test [6]. This baseline ACB scheme can significantly reduce the blocking probability at the cost of a higher access delay. The ACB-based works in literature, e.g., [7]–[9], further suggest dynamically varying the passing probability based on an estimate of the number of backlogged MTDs to reduce the delay cost of the baseline. In this paper, however, we mainly focus on the pull-based solutions which are discussed below.

Pull-based solutions, as opposed to the push-based ones, let the network explicitly triggers MTDs into initiating the RAP. As such, these solutions are more appropriate for data-collecting mMTC applications. In an LTE network, pull-based solutions are realized via the paging functionality that allows the core network, particularly the Mobility Management Entity (MME), to directly "call" for an idle device with a known identification (ID). That is, the MME sends a Paging Message (PM) containing the device's ID to the eNB who, in turns, waits for a Paging Occasion (PO), which is a subframe where the idle device wakes up and monitors the downlink channels, to come and broadcasts the PM. Upon receiving such a message, the device initiates the RAP to access the network. Note that the device must perform the

RAP despite being uniquely identifiable beforehand by the eNB because it is not yet synchronized in the uplink direction and thus cannot be directly scheduled on the PUSCH.

Since there are only up to 4 POs per 10ms and 16 device IDs per PM [10], it would take a significant amount of time and radio resources to page the massive MTD population. To overcome such paging limitations, the Group Paging (GP) scheme that allows the MME to page the MTDs on a collective basis has been proposed. The scheme divides the MTD population into smaller groups identified by the Group IDs (GIDs). The MTDs of the same group shall monitor the same POs and simultaneously initiate the RAP if their common GID is found in a PM. A triggered group is also assigned a time window known as the Paging Interval (PI) to execute the contention-based RAP. When the PI expires, the next group is paged/triggered. The MTDs that are yet to finish the RAP at that point must then withdraw and are also considered as blocked devices. The GP scheme allows the MME to effectively control the number of devices that are allowed to initiate the RAP, and is the canonical example of a pull-based solution.

In practice, the location of POs for a device is a function of 1) the device's International Mobile Subscriber Identity (IMSI), 2) the minimum of the paging cycle configured by the MME and the cell paging cycle provided in the System Information Block 2 (SIB2), and 3) the parameter *nB* in the SIB2 [10]. With that in mind, the GP scheme can be realized by having the MME configure an attached MTD with its own triplet, which consists of a GID, a paging cycle associated with the GID, and a separate *nB*. An MTD configured this way then uses the GID, the associated paging cycle, and the configured *nB* instead of its IMSI, the minimum of the two paging cycles, and the SIB2's *nB*, respectively, to determine its POs. Furthermore, MTDs of a group also re-obtain the SIBs prior to monitoring their PO in a new paging cycle in order to update the possibly outdated cell configurations. When the MME needs to page a group, it sends the GID-containing PM and the relevant triplet to the eNB, who then uses the received triplet to derive the POs' location while ignoring the cell paging cycle and *nB*. By doing so, MTDs of the same group will always monitor the same PO and can be simultaneously triggered by a single PM. Furthermore, the MME can also derive the triplets such that POs of two different groups in the same paging cycle are offset by at least the duration of PI.

Nevertheless, it has been shown that the GP system performs poorly because all devices of the triggered group initiate the RAP simultaneously in the very first RAO [11]. The most straightforward remedy is therefore to distribute the instances where the devices initiate the RAP over the available RAOs of the PI. For example, [11] lets each MTD randomly pick an RAO within the PI to initiate the RAP to achieve a uniform RAP initiations distribution over the PI. The works in [12], [13] further prove that when there is a constant number M_{arv} of MTDs initiating the RAP in each RAO, the system eventually converges to a stable state where the average number of successful MTDs per RAO also becomes a constant. The optimal value of M_{arv} that maximizes the average number of successful MTDs per RAO in the stable state can thus be derived. Then, under an assumption that the group size is known beforehand, the eNB proactively prevents a portion of the triggered MTDs from initiating the RAP (if necessary) and lets each of the remaining MTDs randomly choose an RAO within the PI to initiate the RAP so that the average number of devices initiating the RAP per RAO is kept at the optimum M_{arv}. In [14] where each MTD is associated with an access success probability requirement, Wei et. al formulate and solve an optimization problem to decide which MTD is allowed to initiate the RAP at which RAO of the PI in order to maximize the overall access success probability while still satisfying the requirements of all allowed devices. On the other hand, [15] divides the PI into several access cycles where backlogged devices can only send preambles in a cycle according to an access probability that is updated on a per-cycle basis, taking into account the number of backlogged MTDs and the constraints on resource and energy. Hybrid schemes combining either the probabilistic access control or the RAP initiation redistribution method with a Tree-based contention resolution protocol are proposed and analyzed in [16]. A consecutive GP scheme is introduced in [17] where the eNB may page the same group consecutively up to a certain number of times, and MTDs who fail to access in a PI may retry in the very next PI. The authors of [18], [19] study the problem of dynamic preamble allocation in the conventional GP setting in order to avoid preamble wastage and improve the PRACH efficiency, i.e., the ratio of the average number of successful MTDs to the total number of preamble allocated for the MTDs during the PI. The number of preambles (up to R) allocated to the MTDs in an RAO is adjusted based on the number of backlogged MTDs which, in turns, is estimated via a simple subtraction rule and a complicated theoretical model for [18] and [19], respectively.

2) NETWORK-LEVEL DESIGN SOLUTIONS

It is argued that while the aforementioned solutions can improve the PRACH efficiency and increase the manageable group size of the GP scheme, they still rely solely on the fixed PRACH to provide access to the MTDs and, thus, inevitably face the PRACH resource shortage problem as the MTD population keeps growing. This encourages a new research direction focusing on novel network-level designs that enable the use of additional, non-PRACH resources for mMTC access purposes.

In fact, there have been a number of designs that further divide MTDs in a paged group into *clusters* based on their geographical closeness and assign to each cluster a Cluster Head (CH). The Cluster Members (CMs) then access and communicate with the CH using short-ranged access technologies in what are called ''capillary'' solutions [20]. Examples of such designs in pull-based contexts include [21] where, upon receiving the GP message, the CMs perform the IEEE 802.11ah [22] RAP to access and send the data

to their CHs. The CHs, whose number is only a fraction of the group size, then perform random access on the PRACH following the LTE RAP and forward the aggregated data to the eNB. This design thus effectively makes additional uses of both the IEEE 802.11ah access technology and its frequency resources for mMTC random access. In [23], the CMs of a cluster are assumed to be organized beforehand so that they can transmit their data in a sequential manner (without contentions) to their CH using an unspecified technology over the PUSCH RB explicitly reserved for the CH once the CH has successfully completed the LTE RAP. Capillary architectures have also been spotted in push-based contexts. For example, [24] assumes that the CHs support the functionality of an eNB and lets the CMs initiate the LTE RAP as soon as their data arrive in order to access and send the data to the CH. The CH then initiates the LTE RAP to access and forward the aggregated data to the real eNB, which effectively results in a nested LTE-in-LTE system. [25] employs the ZigBee access technology for establishing the intra-cluster link between a CM and its CH. However, the CM does not send its data to the CH over the link, but is instead scheduled in the time domain by the CH to perform the LTE RAP so as to avoid contending with other CMs of the same cluster on the PRACH. On the other hand, [26] investigates the impacts of the unreliability of intra-cluster links (technology not specified), which may force a device to invoke the LTE RAP and directly access the eNB despite residing within a cluster, on the overall system performance.

The capillary approach represents a powerful solution class that can exploit non-PRACH resources for mMTC random access. However, they come at the expense of frequency planning for the intra-cluster communications, and occasionally require that the MTDs support more than one access technologies, e.g., [21], [25]. There thus exists an alternative, complementary design approach which assumes no intra-cluster communications and that MTDs are LTE-only, but can still exploit the cluster topology to make use of additional resources in the native LTE carrier bandwidth. In push-based contexts, such approach is pioneered by [27] in which several fixed-position CHs are placed within an LTE cell. A CH periodically sends a *dedicated* preamble to the eNB over the PRACH while the CMs stay silent. Upon successfully receiving a dedicated preamble, the eNB reserves *multiple* RBs on the PUSCH. As the CMs are in the close proximity of the CH, they can also apply the associated timing adjustment to appropriately transmit Msg3 on the reserved RBs. However, an individual CM is not scheduled since it is not yet uniquely identified at this point and, therefore, must randomly select one of the reserved RBs to send its Msg3. When multiple CMs use the same reserved RB, an Msg3 collision happens and involved CMs must backoff for a random period before repeating from the start. The MTDs that need to access the network but are not located near any CHs, meanwhile, simply follow the normal LTE RAP. As such, the design additionally exploits the native PUSCH resources for the random access purpose without requiring the MTDs to also support non-LTE access technologies. In pull-based contexts, however, there is no such design to the best of our knowledge.

C. OUR PROPOSAL AND CONTRIBUTIONS

It is seen that although the approach of exploiting additional native PUSCH resources for the mMTC random access by means of non-capillary designs is promising, there has yet been any of such attempts in pull-based contexts. Furthermore, we would argue that it is not efficient to directly apply the design of [27] to a pull-based, more specifically a GP system. Firstly, in a GP system, it is known that all triggered devices simultaneously arrive in the very first RAO and there is no other new arrival during the PI. A CH therefore should not blindly send the dedicated preambles periodically but only until all of its first-RAO CMs have successfully sent their Msg3 on the PUSCH. Indeed, if the CH continues to send preambles afterwards, the eNB will continue to reserve PUSCH RBs that are then unused by any devices and cause a significant PUSCH resource wastage. Secondly, even if we assume that the CHs do stop sending the dedicated preambles after all of the CMs have succeeded, the set of preambles dedicated to the CHs will still remain inaccessible to the non-clustered MTDs. This leads to a resource wastage on the PRACH as the total number of preambles (dedicated and non-dedicated) is fixed.

We are thus motivated to propose a GP system design that can efficiently exploit additional native PUSCH resources for the mMTC random access without relying on capillary operations. Indeed, in a typical 5G heterogeneous network, a large number of small-cells implementing the NR access technology are deployed for the eMBB service and may cover a substantial portion of each group as depicted in Fig. 2. These NR small-cells are not accessible by any of the LTE-only MTDs and thus, cannot be used in a capillary manner. We let the Small-cell Base Stations (SBSs) play the role of the CHs as in [27]. However, in our system, the SBSs and the macro-cell-only MTDs (henceforth referred to as mcMTDs) will have to contend over the same set of non-dedicated PRACH preambles. Once an SBS successfully delivers a preamble and obtains multiple PUSCH RBs from the eNB, its covered MTDs (the CMs), henceforth referred to as small-cell MTDs

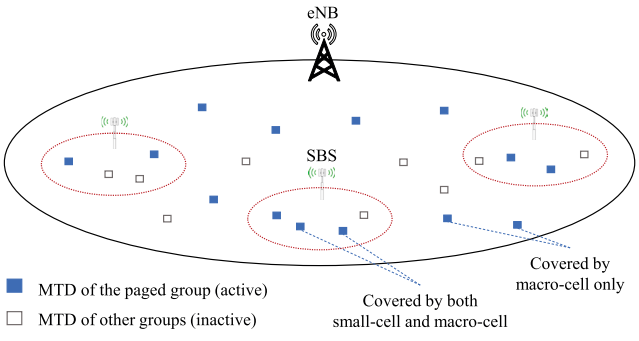


FIGURE 2. Heterogeneous cells layout.

or scMTDs, then contend to transmit their own Msg3 over those RBs. Collisions that happen in either the preamble stage (between the mcMTDs and the SBSs) or the Msg3 stage (between the scMTDs of the same SBS) are handled by an existing Distributed Queue (DQ) mechanism to further enhance the performance. Our contributions are summarized as follows.

- *Design of a non-capillary GP system taking advantage of the cluster topology*. Compared to the push-based counterpart in [27], our system has three main design differences. Firstly, we do not assume dedicated preambles for the CHs. Secondly, our system does not force the CHs to transmit preamble periodically. Instead, they only need to start sending preambles upon receiving the GP message and continue doing so until all triggered devices under their coverage are resolved. Thirdly, contentions are handled using an existing DQ protocol instead of the conventional random-backoff protocol. Computer simulations show that when compared to an optimal GP-based solution that only relies on PRACH preambles, the proposed system can support a remarkably higher group size, given the same PI, at a reasonable additional cost on the PUSCH.
- *Formulation of a theoretical model* to quickly predict the system's behaviors under various configurations to find desirable operating points. Since the SBSs in our system are not assigned dedicated preambles but compete over the same set of non-dedicated preambles with the mcMTDs and stop once their scMTDs have been resolved, system modeling becomes significantly more complicated. Given the group size, our proposed model can capture not only the access delay and blocking rate of the system, but also the additional cost on PUSCH with high accuracy as verified via comparison with computer simulations. Thus, it serves as a useful tool for system planners to quickly quantify the tradeoff between the performance gain and additional resource cost on PUSCH, from which appropriate system parameter settings can be selected.

The rest of this paper is organized as follows. In the next section, we review the conventional LTE RAP and propose a modified one to accommodate the SBSs. An existing DQ protocol to resolve the contentions and a way to incorporate such protocol into the modified RAP are also described in the same section. The theoretical model is formulated in Section III and validated by computer simulations in Section IV. Finally, Section V concludes the paper.

II. SMALL-CELL ASSISTED GROUP PAGING (SCAGP)

In this section, we thoroughly describe our proposal which includes both a modified RAP to accommodate the SBSs and an existing DQ-based mechanism to handle possible contentions. However, we will first start with a detailed description of the contention-based RAP that was briefly mentioned in section I-A.

A. THE LTE RANDOM ACCESS PROCEDURE

The RAP is a four-message handshaking procedure between an MTD and the eNB to establish a physical link required for carrying out the higher-layer connection establishment process. The message flow is as follows.

- Msg1 (RA preamble): The MTD randomly selects one from $R \leq 64$ orthogonal preamble sequences and sends it to the eNB (over the PRACH) in the nearest RAO. When multiple MTDs send the same preamble in the same RAO, a preamble collision occurs and renders the preamble undecodable [3].
- Msg2 (RA Response): An RAR is a downlink message containing multiple acknowledgments, each of which addresses a successfully decoded preamble. The RARs for an RAO are transmitted within a window that starts exactly 2 subframes after the RAO and lasts for W_{RAR} subframes. Each RAR consumes one subframe in the window and may contain up to N_{RAR} acknowledgments. Carried in an acknowledgment are the ID of a successful preamble (Random Access Preamble ID, RAPID), a temporary identifier (Temporary Cell Radio Network Temoporary Identifier, TC-RNTI) to uniquely identify the corresponding device at the physical layer, a timing adjustment instruction to help the device synchronize in the uplink direction, and an one-RB resource grant on the PUSCH for the device's transmission of the Msg3. A backoff indicator BI may also be included to instruct the MTDs whose preambles' RAPIDs are not found in any of the RARs to backoff for a random period of $(0, BI]$ milliseconds before retrying from the first step.
- Msg3 (RRC Connection Request): This is the first actual message of the connection establishment process. After decoding the relevant acknowledgment, the MTD applies the timing adjustment and transmits Msg3 on the reserved PUSCH RB. The message contains the device's core network-level ID (not the TC-RNTI) and a reason for requesting an access. This Msg3 is protected by the Hybrid Automatic Repeat Request (HARQ) protocol. That is, if the transmission suffers from uncorrectable errors, the eNB reschedules the MTD to retransmit on a new reserved PUSCH RB in the future. Note that at this point, the MTD is uniquely identifiable to the eNB at the physical layer (via the TC-RNTI) and can thus be rescheduled if needed.
- Msg4 (Contention Resolution): When the eNB correctly receives an Msg3, it echoes back the decoded higher-layer ID via Msg4 as an acknowledgment. The RAP is considered successful upon the correct reception of this message at the MTD's side. Msg4 is also protected by the HARQ protocol.

If an MTD fails to transmit its preambles N_{PTmax} times, it terminates the RAP and is temporarily blocked from the network.

B. THE PROPOSED MODIFIED RA PROCEDURE

In heterogeneous settings, the possibility that many MTDs lie in the overlap of both a traditional LTE macro-cell and a NR small-cell (see Fig. 2) can be exploited to reduce contentions during the preamble transmission step. Before going into details, let us make the following assumptions about our proposed system.

1) The SBSs may send preambles to the eNB while their covered MTDs do not. The scMTDs also know the SBS-IDs of the SBSs that they belong to.
2) All R preambles are equally accessible to both SBSs and mcMTDs. The eNB may distinguish whether a singleton, i.e., non-colliding, preamble is from an mcMTD or an SBS and if the preamble is from an SBS, the eNB also knows the ID of that SBS [28].
3) The eNB can distinguish whether an Msg3 on a PUSCH RB is erroneous due to the poor channel condition or due to multiple MTDs using the same RB (collision).
4) The bandwidth of the PUSCH is sufficiently wide.

In our system, an SBS will compete with other SBSs and the mcMTDs on behalf of its scMTDs during Msg1 stage, while local competition between its scMTDs actually happens at the Msg3 stage after the SBS successfully obtains the uplink resource from eNB. The proposed modified RAP is thus as follows

- Msg1: The mcMTDs and the SBSs whose MTDs are not yet succeeded on the PUSCH randomly select their preambles among the same set of R available ones and send them to the eNB (over the PRACH) in the nearest RAO.
- Msg2: Upon successfully decoding a preamble, the eNB sends an acknowledgment (via an RAR). The content of an acknowledgment depends on whether the preamble is sent by an mcMTD or an SBS. If the preamble is from an mcMTD, then the acknowledgment contains the RAPID, a unique TC-RNTI, a timing adjustment instruction, and an one-RB resource grant on the PUSCH. Otherwise, it contains the SBS-ID, a timing adjustment instruction, a N_b-RB resource grant on the PUSCH, and N_b unique TC-RNTIs associated with the N_b RBs. Note that whenever an SBS transmits a preamble in an RAO, its scMTDs also monitor the RARs of that RAO to see if the SBS has succeeded, i.e., if there is any RAR containing the ID of the SBS.
- Msg3: Based on the one-RB grant, the successful mcMTD transmits its Msg3 on the single reserved RB. On the other hand, each scMTD of the successful SBS randomly selects one out of N_b PUSCH RBs reserved for the SBS (and also assumes the TC-RNTI associated with the selected RB) to deliver its own Msg3. When multiple scMTDs send Msg3 on the same PUSCH RB, a *Msg3 collision* occurs and the message is severely corrupted. The eNB will indicate the PUSCH RBs that suffer from collisions via an assumed new feedback format[1] on the downlink channel T_f subframes later. If an Msg3 transmission on a certain RB is not involved in a collision but requires an HARQ retransmission, the eNB reschedules the corresponding device as in the normal RAP. The rescheduling is possible because at this point the device (whether an mcMTD or an scMTD) is uniquely identifiable thanks to the one-to-one association between a TC-RNTI and a PUSCH RB.
- Msg4: Same as in conventional RA procedure.

Although this modified RAP is our paper's highlight, it remains unclear how contentions (preamble and Msg3 collisions) are handled, how an SBS knows if its scMTDs are not yet fully resolved so that it may continue requesting for PUSCH resources, and how the scMTDs of an SBS know in which RAO the SBS sends a preamble so that they can monitor the RARs accordingly. In the following part, we describe an *existing* DQ-based contention resolution protocol [29] to provide an answer to the mentioned ambiguities.

C. DISTRIBUTED QUEUEING-BASED CONTENTION RESOLUTION

The traditional contention-based RAP in section II-A resolves preamble collisions via a random backoff protocol, which results in a very high blocking probability for the MTDs [12]. The DQ protocols, on the other hand, resolve contentions by organizing colliding devices into a *logical* queue. When collisions happen in an RAO, involved MTDs are randomly split into G smaller subsets and "pushed" to the queue's end. Then, in each RAO, only devices of the head subset may leave the queue to perform retransmission. This prevents the subsets from interfering with each other and helps DQ protocols achieve a very low blocking probability. It is noted that the logical queue is realized using two counters, namely p_Q and DQ. The p_Q counter is maintained at each individual MTD and represents its position inside the queue. An MTD whose $p_Q = 0$ is currently at the queue's head. The DQ counter, meanwhile, is kept at the eNB and represents the queue's length. The counters are updated after each RAO as follows.

For DQ (at the eNB): $DQ^{\text{new}} = \max\left(DQ^{\text{old}} - 1, 0\right) + G$ to reflect the removal of the head subset (if any) and the addition of G newly created subsets to the queue.

For p_Q (at each MTD):

- If $p_Q^{\text{old}} > 0$, the device is still queuing, and thus $p_Q^{\text{new}} = p_Q^{\text{old}} - 1$ to reflect head subset's removal.
- Otherwise, the device has transmitted in the RAO as its $p_Q^{\text{old}} = 0$. If the device is involved in a collision, then $p_Q^{\text{new}} = \max\left(DQ^{\text{old}} - 1, 0\right) + g$ where g is a random integer between $[0, G - 1]$. This is to reflect that the device has chosen the g-th subset and rejoined the queue from the end. DQ^{old}, G, and the preamble statuses are included in the RARs.

[1] Detailed design of this new feedback format is not considered in the paper

The number of subsets G is chosen based on an estimate $\hat{m}_c$ of the number of colliding MTDs[2] such that the average size of a created subset is kept at a designated level d. More specifically, $G = \max\left(\lfloor \hat{m}_c/d \rceil, 1\right)$ where $\lfloor\cdot\rceil$ is the nearest integer operator. That is, if $\hat{m}_c > d$, the colliding MTDs will be randomly divided into $G = \lfloor \hat{m}_c/d \rceil \geq 1$ subsets, which ensures that the average size of a newly created subset is approximately d. Otherwise, $G = 1$, i.e., no division is needed, and all colliding MTDs join the queue's end as a single new subset because their number is already below d. When the subsets' size is much lower than the designated level d, however, it may be possible to merge some subsets to keep their size close to d. Therefore, the eNB also keeps an estimation $\hat{m}_e$ of the number of MTDs of the tail subset. The colliding MTDs in an RAO will merge with this tail subset if the estimated size after merging, i.e., $\hat{m}_e + \hat{m}_c$, does not exceed d. To realize this logic, the eNB and each MTD apply the following additional update rules.

- At the eNB: If $\hat{m}_c + \hat{m}_e^{\text{old}} \leq d$, subset merging occurs and no new subset is created, i.e., $G = 0$, and thus $\hat{m}_e^{\text{new}} = \hat{m}_e^{\text{old}} + \hat{m}_c$. Otherwise, subset merging does not happen and $G = \max\left(\lfloor \hat{m}_c/d \rceil, 1\right)$ subsets are created at the queue's end as usual. $\hat{m}_e$ is thus updated as $\hat{m}_e^{\text{new}} = \lfloor \hat{m}_c/G \rceil$.
- At each MTD: If a colliding MTD sees $G = 0$ in the RAR, it set $p_Q^{\text{new}} = DQ^{\text{old}} - 2$ to merge with the tail subset.

Note that if there is only one subset in the queue, i.e., $DQ^{\text{old}} = 1$, the merging operation will also not occur because after the only subset exits the queue to perform preamble retransmission in the RAO, there would be no other subsets to merge with even if the resultant number of colliding MTDs in the RAO is low enough. In this case, $G = \max\left(\lfloor \hat{m}_c/d \rceil, 1\right)$ and $\hat{m}_e^{\text{new}} = \lfloor \hat{m}_c/G \rceil$ as usual.

An example of the DQ mechanism is portrayed in Fig. 3. For demonstration purposes, we assume that there are $R = 4$ preambles, that the designated subset size is $d = R$, and that the eNB knows exactly the number of colliding MTDs in an RAO and in each subset. In the first RAO, the paged group of 16 MTDs simultaneously send their preambles and none succeeds due to preamble collisions. All MTDs are thus randomly divided $G = \lfloor 16/4 \rceil = 4$ subsets and "pushed" to the queue. The head subset of 5 MTDs then exits the queue to retry in RAO 2 where two of the MTDs involve in a preamble collision. The eNB knows that the colliding MTDs should not be further divided because $2 < d$. It also knows that since the tail subset has 4 MTDs, merging is not allowed as $4 + 2 > d$. Therefore, $G = 1$ so that the two colliding MTDs rejoin the queue as a single new subset of size 2 from the end. In RAO 3, the next subset of size 4 retries and two MTDs experience collision. The eNB allows subset merging in this case because $2 + 2 \leq d$ and sets $G = 0$ so that the two colliding MTDs "merge" with the tail subset instead of rejoining as a new separated subset. This continues until all MTDs succeed, i.e., until the queue becomes empty.

[2]Readers are referred to section III-A of [29] for details about the MAC-layer estimation method

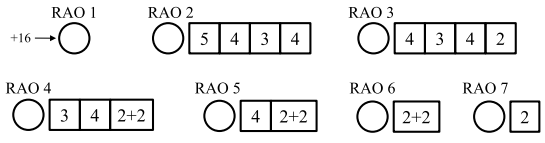


FIGURE 3. DQ protocol operation.

D. INTERWORKING BETWEEN DQ AND MODIFIED RAP

In our system, the eNB employs the aforementioned DQ mechanism to handle contentions during both Msg1 and Msg3 stage, and maintains two corresponding queue types. The first type, namely Msg1 queue, is used for resolving the entities that contend over the R preambles on the PRACH. Each Msg3 queue, meanwhile, is reserved for the scMTDs of a certain SBS to contend locally over the SBS's obtained N_b RBs on PUSCH. The differences between the two queue types are summarized in Table 2. Note that for the sake of fairness, an scMTD will also terminate the RAP upon exceeding N_{PTmax} Msg3 collisions. To smoothly incorporate these two queue types into the already well-defined RAP, however, attentions to details must be paid.

Let us first consider the Msg1 queue. Following the description thus far, it may seem natural that both the mcMTDs and the SBSs should participate in the Msg1 queue as equals. However, doing so might result in a prolonged delay because scMTDs, which constitute a sizable portion of the population, cannot resume their Msg3 queuing processes until the SBSs get to the Msg1 queue's head and successfully transmit preambles. We therefore propose to let the SBSs reside permanently at the Msg1 queue's head, i.e., their p_Q is always 0, so that they may send preambles as soon as needed. This obviously introduces additional contentions and estimation error that negatively affect Msg1 queue's performance. Nevertheless, since the number of SBSs is relatively low, the impact is negligible while the delay of scMTDs can be significantly reduced. Fixing p_Q of the SBSs at 0 also helps the scMTDs know exactly at which RAO their SBSs send a preamble so that they may monitor the RARs correspondingly.

It should also be noted that there can be multiple Msg1 queues due to the RAO spacing constraint. That is, a single Msg1 queue cannot utilize all RAOs if the next RAO comes before the queue is updated, i.e., before the completion of the current RAR window. As an example, let us assume that there is one RAO every 5 subframes, says, at subframe 1, 6, 11, 16, and that $W_{RAR} = 5$. A single Msg1 queue whose mcMTDs transmit in subframe 1 then cannot be updated until the end of subframe 8. The RAO in subframe 6 thus becomes unusable for the queue. To fully utilize all RAOs, there should be two Msg1 queues. One queue operates on subframes 1, 11, while the other uses subframes 6, 16 so that from either Msg1 queue's perspective, the RARs for an RAO

TABLE 2. A comparison between the Msg1 queue and an Msg3 queue.

	The Msg1 queue	An Msg3 queue
Contenders	The mcMTDs and the SBSs	The scMTDs of the same SBS
Resources	R preambles	N_b PUSCH RBs
Optimal designated subset size d_{opt}	$\underset{d\in\mathbb{Z}_{>0},d\leq R}{\text{argmin}} \left\lvert d\left(1-\frac{1}{R}\right)^{d-1} - \min\left\{R\left(1-\frac{1}{R}\right)^{R-1}, W_{\text{RAR}}\times N_{\text{RAR}}\right\}\right\rvert$ [29]	N_b
Time unit	An RAO that periodically appears on the time domain	An "Msg3 slot" that only appears each time the relevant SBS succeeds in the Msg1 queue
Feedback message	The RARs	The new feedback format assumed in section II-B

always come before the next RAO's arrival. Each mcMTD and SBS then randomly chooses an Msg1 queue to associate with upon entering the system. This multi-queue concept has been mentioned in our previous works [30].

On the other hand, multiple Msg3 queues may also exist. The reason is however not due to the same constraint but the fact that an Msg3 queue is used to resolve the scMTDs of a *single* SBS. The number of Msg3 queues is thus equal to the number of SBSs in the system. An example of such interworking is shown in Fig. 4 where $T_f = 8$ms and, for simplicity, RAO periodicity and W_{RAR} are respectively set to 10ms and 5ms so that there is only one Msg1 queue ($\text{Q}_{msg1,\#1}$). Let us now focus on a particular SBS, says the SBS4. In subframe 1, which is an RAO as denoted by a colored square, the SBS4 sends a preamble. Since its preamble is not chosen by any other SBSs or mcMTDs in the same RAO, the SBS4 gets an acknowledgment containing an N_b-RB grant from eNB in subframe 4. After a 5ms delay to process the acknowledgment [5], the scMTDs that are covered by the SBS4 *and* reside at the head subset of the corresponding Msg3 queue ($\text{Q}_{msg3,\#4}$) apply the timing adjustment and randomly select their RBs (and assume the TC-RNTIs associated with the selected RBs) from the N_b reserved PUSCH RBs to deliver their Msg3. $T_f = 8$ subframes later, those scMTDs receive the new format feedback containing results of transmission on the RBs, $\text{Q}_{msg3,\#4}$'s length, and corresponding G from the eNB. $\text{Q}_{msg3,\#4}$ can then be updated accordingly. Since SBS4 also sees the length of $\text{Q}_{msg3,\#4}$ via the new feedback format, it knows that such queue is not yet empty, i.e., there are unresolved scMTDs, and thus continues to send a preamble at the nearest RAO in subframe 21. This time it is involved in a preamble collision and does not get any acknowledgment from eNB. Since the SBSs are prioritized, SBS4 stays at $\text{Q}_{msg1,\#1}$'s head to retry in the very next RAO at subframe 31 instead of to rejoining the queue from the end.

E. OTHER CONSIDERATIONS

To account for wireless channel impairments and power ramping effect, the 3GPP simulation setup assumes that a singleton (non-colliding) preamble is detected with probability $(1 - 1/e^n)$ where n is the number of preamble transmissions thus far at the corresponding entity [3]. Additionally, even when the singleton preamble is detected, the limitation in the number of acknowledgments that can be sent during a RAR window, i.e., $W_{\text{RAR}} \times N_{\text{RAR}}$, may result in the entity not being acknowledged. In our paper, we reasonably assume that the SBSs have advance channel estimation capability and enough power budget so that a singleton preamble sent by an SBS is always detected. On the other hand, both mcMTDs and SBSs may be affected by the acknowledgment limitation. The entities whose preamble transmissions fail due to any reasons other than collisions are allowed to retry in the very next RAO seen by their chosen Msg1 queue. An Msg3 transmission, on the contrary, is not subjected to the mentioned limitations.

III. PROPOSED THEORETICAL MODEL

In this section, we aim to construct a mathematical model to quickly predict the proposed system's behaviors. For convenience, random quantities and their deterministic equivalents are symbolized by calligraphic and normal letters, respectively.

A. DRIFT APPROXIMATION

Let $\mathcal{M}_i[n]$ be the number of MTDs who will transmit for the n-th time in the i-th time unit. Also, let $\mathcal{L}_i$ be the queue's length, and the estimated size of the tail subset at the beginning of the i-th time unit, respectively. If we assume a perfect estimation and ignore subset merging, the (simplified) DQ contention process can be described via the evolution of a multi-dimensional discrete-time stochastic process $\vec{\mathcal{M}}_i = \left\{\mathcal{M}_i[1], \ldots, \mathcal{M}_i[N_{\text{PTmax}}], \mathcal{L}_i : i \in \mathbb{Z}_{\geq 0}\right\}$ as follows.

$$\begin{cases} \mathcal{M}_{i+\mathcal{L}_i-1+g}[2] = \text{Bino}\left(\mathcal{M}_{i,c}[1], \frac{1}{\mathcal{G}_i}\right) \\ \cdots \\ \mathcal{M}_{i+\mathcal{L}_i-1+g}[N_{\text{PTmax}}] = \text{Bino}\left(\mathcal{M}_{i,c}[N_{\text{PTmax}}-1], \frac{1}{\mathcal{G}_i}\right) \\ \mathcal{L}_{i+1} = \mathcal{L}_i - 1 + \mathcal{G}_i, \end{cases} \tag{1}$$

where $\mathcal{M}_{i,c}[n]$ and $\mathcal{G}_i = \max\left(\left\lfloor \frac{\sum_{n=1}^{N_{\text{PTmax}}} \mathcal{M}_{i,c}[n]}{d} \right\rceil, 1\right)$ are the numbers of MTDs colliding during their n-th transmission and of newly created subsets in this i-th time unit, respectively, while g is any integer $\in [1, \mathcal{G}_i]$. The system of equations (1) suggests that this process is a general adapted process for which a definitive solution in transient conditions is not available. Furthermore, the state space of such process is prohibitively large due to a massive number of devices and the involvements of $(N_{\text{PTmax}} + 1)$ many random variables, even if we factor in the fact that the number of permissible states can be slightly reduced due to the random variables'

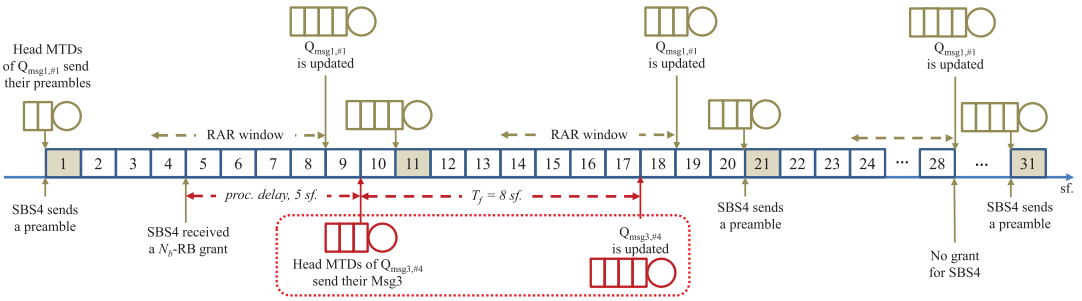


FIGURE 4. Interworking between DQ-based contention resolution protocol and modified RAP.

TABLE 3. The list of key variables in the proposed theoretical model.

Variables	*Meaning*
$M_i[n]$, $M_{i,s}[n]$, $M_{i,c}[n]$	Number of scMTDs that transmit, succeed, and collide in their n-th Msg3 attempts in the i-th Msg3 slot of the relevant Msg3 queue, respectively.
M_i, $M_{i,s}$, $M_{i,c}$	Total versions of $M_i[n]$, $M_{i,s}[n]$, and $M_{i,c}[n]$, respectively, summing over all n from 1 to N_{PTmax}.
$\hat{M}_{i,c}, \hat{M}_{i,e}$	Estimated number of scMTDs that collide and reside in the tail subset in the i-th Msg3 slot of the relevant Msg3 queue, respectively.
L_i	The length of the relevant Msg3 queue at the beginning of its i-th Msg3 slot.
$P_j[n]$, $P_{j,s}[n]$, $P_{j,c}[n]$, $P_{j,f}[n]$	Number of mcMTDs that transmit, succeed, collide, and fail in their n-th preamble attempts in the j-th local RAO of the relevant Msg1 queue, respectively.
$N_j[m]$, $N_{j,s}[m]$, $N_{j,c}[m]$, $N_{j,f}[m]$	Number of SBSs that have successfully accumulated m acknowledgements and will transmit, succeed, collide, and fail in their preamble attempts in the j-th local RAO of the relevant Msg1 queue, respectively.
P_j, $P_{j,s}$	Total number of entities (both mcMTDs and SBSs) that transmit preambles and that are acknowledged in the j-th local RAO of the relevant Msg1 queue, respectively.
$\hat{P}_{j,c}, \hat{P}_{j,e}$	Estimated number of colliding entities and of the tail subset's size in the j-th local RAO of the relevant Msg1 queue, respectively.
$L_{j,P}$	The length of the relevant Msg1 queue at the beginning of its j-th local RAO.

correlated nature. To work around the problem, we use a drift approximation method to approximate the stochastic process $\mathcal{M}_i$ by a deterministic discrete-time equivalent $\vec{M}_i$ which describes the transient evolution of the drift of $\mathcal{M}_i$ itself. More specifically, $\vec{M}_i$ is defined as the multi-dimensional deterministic process $\left\{M_i[1], \ldots, M_i[N_{\text{PTmax}}], L_i : i \in \mathbb{Z}_{\geq 0}\right\}$ whose evolution is expressed as

$$\begin{cases} M_{i+L_i-1+g}[n] = \mathbb{E}\left[\mathcal{M}_{i+\mathcal{L}_i-1+g}[n] \middle| \vec{\mathcal{M}}_i = \vec{M}_i\right] \\ L_{i+1} = \mathbb{E}\left[\mathcal{L}_{i+1} \middle| \vec{\mathcal{M}}_i = \vec{M}_i\right]. \end{cases} \tag{2}$$

Later simulations will show that such approximation works reasonably well under practical settings. Note that the method itself is not new and has been used in [31] to study an approximated model of the conventional RAP with backoff-based contention resolution. Nevertheless, the formulation of such a model for the modified RAP with DQ-based contention resolution, as elaborated in the following sections, is indeed our original contribution. The key variables in our model are summarized in Table 3.

B. A MODEL FOR AN Msg3 QUEUE

We are now ready to construct a deterministic model for the Msg3 queuing process. Let us additionally denote by $\hat{M}_{i,e}$ the estimated size of the tail subset in the i-th "Msg3 slot" and, in a slight abuse of notation, consider the process $\vec{M}_i = \left\{M_i[1], \ldots, M_i[N_{\text{PTmax}}], \hat{M}_{i,e}, L_i : i \in \mathbb{Z}_{\geq 0}\right\}$. Since all MTDs of the paged group simultaneously initiate the modified RAP, the initial state is $\vec{M}_0 = \left\{\frac{N \cdot c}{N_{\text{SBS}}}, 0, \ldots, 0\right\}$ where N is the number of MTDs in the paged group, c is the portion of the population that are covered by the SBSs, and N_{SBS} is the number of SBSs.

Let us consider the i-th Msg3 slot. For compactness, the total numbers of scMTDs transmitting, succeeding, and colliding in the i-th Msg3 slot are referred to as $M_i = \sum_{n=1}^{N_{\text{PTmax}}} M_i[n]$, $M_{i,s} = \sum_{n=1}^{N_{\text{PTmax}}} M_{i,s}[n]$, and $M_{i,c} = \sum_{n=1}^{N_{\text{PTmax}}} M_{i,c}[n]$, respectively, where $M_{i,s}[n]$ and $M_{i,c}[n]$ are respectively the numbers of scMTDs that succeed and collide in their n-th Msg3 transmission in the i-th Msg3 slot. When M_i devices contend over N_b RBs, the probability that a device does not collide with the others can be written as $\delta(M_i, N_b) = (1 - 1/N_b)^{M_i - 1}$. Since this probability does not depend on n, the average number of successful n-th time scMTDs can be written as

$$M_{i,s}[n] = M_i[n]\delta(M_i, N_b). \tag{3}$$

Concretely, (3) is only applicable when $M_i \in \mathbb{N}$. However, we relax that constraint and assume that (3) works for $M_i \in \mathbb{R}_{\geq 0}$. The average number of scMTDs who collide in their n-th Msg3 attempts in this i-th Msg3 slot is then

$$M_{i,c}[n] = M_i[n](1 - \delta(M_i, N_b)). \tag{4}$$

Now it is necessary to compute the number of newly created subsets G_i. As described in section II-C, G_i is based on the an estimate $\hat{M}_{i,c}$ of $M_{i,c}$ and the designated subset size d. To simplify the model, a perfect estimation, i.e., $\hat{M}_{i,c} = M_{i,c}$ is oftentimes a reasonable relaxation. However, when all N_b PUSCH RBs are in collisions, the MAC-level estimation method cannot estimate $M_{i,c}$ and must interpolate $\hat{M}_{i,c}$ as the smallest integer multiple (denoted by U_{N_b}) of N_b that is higher than the estimate obtained when there are $N_b - 1$ colliding RBs [29]. For example, given $N_b = 10$, U_{N_b} is found as 50. To reflect such limit while keeping the model simple, we assume that $\hat{M}_{i,c} = M_{i,c}$ only if $M_{i,c} \leq U_{N_b}$ and $\hat{M}_{i,c} = U_{N_b}$ otherwise. Consequently,

$$G_i = \max\left(\left\lfloor \frac{\hat{M}_{i,c}}{d_M} \right\rceil, 1\right) = \max\left(\left\lfloor \frac{\min\left(M_{i,c}, U_{N_b}\right)}{d_M} \right\rceil, 1\right), \tag{5}$$

where d_M is the designated subset size of the Msg3 queues. Note that G_i in (5) is the number of subset that will be created given that the subset merging operation does not happen, i.e., given that either $\hat{M}_{i,c} + \hat{M}_{i,e} > d_M$ or $L_i = 1$. When $\hat{M}_{i,c} + \hat{M}_{i,e} \leq d_M$ and $L_i > 1$, subset merging occurs and $G_i = 0$. The system of equations describing the evolution of can thus be written as, for $n \in [2, N_{\text{PTmax}}]$ and $g \in [0, G_i - 1]$,

$$\begin{cases} M_{i+L_i+g}[n] = M_{i,c}[n-1]/G_i \\ \hat{M}_{i+1,e} = \hat{M}_{i,c}/G_i \\ L_{i+1} = L_i - 1 + G_i, \end{cases} \tag{6}$$

if $\hat{M}_{i,e} + \hat{M}_{i,c} > d_M$ or $L_i = 1$, and

$$\begin{cases} M_{i+L_i-1}[n] = M_{i+L_i-1}[n] + M_{i,c}[n-1] \\ \hat{M}_{i+1,e} = \hat{M}_{i,e} + \hat{M}_{i,c} \\ L_{i+1} = L_i - 1, \end{cases} \tag{7}$$

otherwise. Note that $M_i[1] = 0$ for all $i \geq 1$. Eqs.(3)−(7) allow us to update several future values of the elements of $\vec{M}_i$ given the current state.

To finish the model, we need to specify a condition upon meeting which $\vec{M}_i$ is terminated. The DQ contention process should finish when the last remaining subset is resolved. We therefore terminate the process at the i_t-th Msg3 slot where $M_{i_t} < 1$ and $L_{i_t} = 1$.

C. COMPLETE MODEL FOR MODIFIED RAP WITH DQ

The model for an Msg3 queue cannot be directly reapplied to an Msg1 queue. Indeed, to correctly model the Msg1 queues, significant modifications are needed to account for the SBSs, various phenomena, e.g., the limited number of acknowledgments during an RAR window, and timing details. In this section, we elaborate the formulation of such a model.

The key observation to model the SBSs' involvement is that an SBS only transmits preambles when the corresponding Msg3 queue is not yet empty. Since on average an Msg3 queue lasts for i_t "Msg3 slots", i.e., until the corresponding SBS has obtained i_t acknowledgments from the eNB, we make the most important approximation here to assume that each SBS stops sending preambles when it has cumulatively received i_t acknowledgments and consider the process $\vec{P}_j$ defined as

$$\left\{P_j[1], \ldots, P_j[N_{\text{PTmax}}], N_j[0], \ldots, N_j[i_t - 1], \hat{P}_{j,e}, L_{j,P} : j \in \mathbb{Z}_{\geq 0}\right\}, \tag{8}$$

where $P_j[n]$, $\hat{P}_{j,e}$, and $L_{j,P}$ is the number of mcMTDs transmitting their n-th preamble, the estimated size of the tail subset of the Msg1 queue, and the Msg1 queue's length in RAO j, respectively. Meanwhile, $N_j[m]$ is the number of SBSs that will compete in RAO j and have accumulated m acknowledgments thus far. The initial state of $\vec{P}_j$ has $P_0[1] = N(1-c)/N_{\text{Q}}, N_0[0] = N_{\text{SBS}}/N_{\text{Q}}$, where N_{Q} is the number of Msg1 queues, and all other quantities equal 0.

In the j-th local RAO, the number of transmitting entities, denoted by P_j, is

$$P_j = \sum_{n=1}^{N_{\text{PTmax}}} P_j[n] + \sum_{m=0}^{i_t-1} N_j[m]. \tag{9}$$

The number of successful entities given that there are P_j transmitting entities and R preambles cannot be derived in a straightforward way using $\delta\left(P_j, R\right)$ as in (3). In order for an entity to be successful, its preamble transmission must not only be singleton (not involved in a collision), but also detected at the eNB and acknowledged during the RAR window (see section II-E). To find the number of successful entities, we proceed as follows. Let us first assume that $P_j \in \mathbb{N}$ and condition on the event that exactly $k \in N$ among the P_j transmitting entities are singleton. The conditional average numbers of detected n-th time mcMTDs and m-th time SBSs can then be approximated by $\frac{kP_j[n]p_n}{P_j}$ and $\frac{kN_j[m]}{P_j}$, respectively, where $p_n = (1 - 1/e^n)$ is the probability that a singleton n-th time mcMTD is detected at the eNB. The total number of entities that are acknowledged is thus

$$\min\left(\sum_{n=1}^{N_{\text{PTmax}}} \frac{kP_j[n]p_n}{P_j} + \sum_{m=0}^{i_t-1} \frac{kN_j[m]}{P_j}, N_{\text{UL}}\right), \tag{10}$$

where $N_{\text{UL}} = W_{\text{RAR}} \times N_{\text{RAR}}$ is the maximum number of acknowledgments that can be sent during the RAR window. Thus, the probability that a detected entity in the j-th RAO is also acknowledged is

$$p_{j,\text{ack}} = \frac{\min\left(\sum_{n=1}^{N_{\text{PTmax}}} \frac{kP_j[n]p_n}{P_j} + \sum_{m=0}^{i_t-1} \frac{kN_j[m]}{P_j}, N_{\text{UL}}\right)}{\sum_{n=1}^{N_{\text{PTmax}}} \frac{kP_j[n]p_n}{P_j} + \sum_{m=0}^{i_t-1} \frac{kN_j[m]}{P_j}}. \tag{11}$$

The unconditional average numbers of acknowledged n-th time mcMTDs and m-th time SBSs, respectively denoted by $P_{j,s}[n]$ and $N_{j,s}[n]$, are then expressed as

$$P_{j,s}[n] = \sum_{k=0}^{\min(P_j,R)} \frac{kP_j[n]p_n}{P_j} \cdot p_{j,\text{ack}} \cdot S_R\left(P_j; k\right), \tag{12}$$

$$N_{j,s}[m] = \sum_{k=0}^{\min(P_j,R)} \frac{kN_j[m]}{P_j} \cdot p_{j,\text{ack}} \cdot S_R\left(P_j; k\right), \quad (13)$$

where $S_R\left(P_j; k\right)$ is the probability that there are exactly $k \in N$ singleton entities among the $P_j \in \mathbb{N}$ transmitting ones, given the R preambles. Similarly, the number of colliding n-th time mcMTDs and m-th time SBSs, respectively denoted by $P_{j,c}[n]$ and $N_{j,c}[m]$, are expressed as

$$P_{j,c}[n] = \sum_{k=0}^{\min(P_j,R)} \left(P_j[n] - \frac{kP_j[n]}{P_j}\right) S_R\left(P_j; k\right), \quad (14)$$

$$N_{j,c}[m] = \sum_{k=0}^{\min(P_j,R)} \left(N_j[m] - \frac{kN_j[m]}{P_j}\right) S_R\left(P_j; k\right). \quad (15)$$

Note that the probability $S_R\left(P_j; k\right)$ is readily available from, e.g., equation (10) in [31]. Since P_j in our model is actually in $\mathbb{R}_{\geq 0}$ domain, we perform linear interpolations of (12)-(15) between $\lceil P_j \rceil$ and $\lfloor P_j \rfloor$ to obtain the final $P_{j,s}[n]$, $P_{j,c}[n]$, $N_{j,s}[m]$, and $N_{j,c}[m]$. The numbers of n-th time mcMTDs and m-th time SBSs whose preamble transmissions fail due to reasons other than collisions, respectively denoted by $P_{j,f}[n]$ and $N_{j,f}[m]$, are then simply

$$\begin{cases} P_{j,f}[n] = P_j[n] - P_{j,s}[n] - P_{j,c}[n] \\ N_{j,f}[m] = N_j[m] - N_{j,s}[m] - N_{j,c}[m]. \end{cases} \quad (16)$$

We can now derive the number of created groups $G_{j,P}$. Let us denote the total number of colliding entities in the j-th RAO by $P_{j,c}$ and the corresponding MAC-level estimate by $\hat{P}_{j,c}$. Note that by the definitions, we have $P_{j,c} = \sum_{n=1}^{N_{\text{PTmax}}} P_{j,c}[n] + \sum_{m=0}^{i_t-1} N_{j,c}[m]$. When all R preambles are in the collision state, the eNB cannot estimate $P_{j,c}$ and must interpolate $\hat{P}_{j,c}$ to the smallest integer multiple of R (denoted by U_R) that is higher than the estimate obtained when there are $R-1$ colliding preambles. To keep the model simple, however, we assume that the eNB knows the exact value of $P_{j,c}$ but only up to U_R. Consequently, $G_{j,P}$ is approximated as

$$G_{j,P} = \max\left(\left\lfloor \frac{\hat{P}_{j,c}}{d_P} \right\rceil, 1\right) = \max\left(\left\lfloor \frac{\min\left(P_{j,c}, U_R\right)}{d_P} \right\rceil, 1\right), \quad (17)$$

where d_P is the designated subset size of the Msg1 queues.

The next task is to formulate a system of equations to describe the evolution of the process $\vec{P}_j$. Note that although the colliding mcMTDs will join the queue as $G_{j,P}$ new subsets, the mcMTDs who fail due to reasons other than collisions can retry in the next RAO. Thus, we have the following evolution equations (in strict order) when subset merging does not occur, i.e., when $\hat{P}_{j,c} + \hat{P}_{i,e} > d_P$ or $L_{j,P} = 1$,

$$\begin{cases} P_{j+L_{j,P}+g}[n] = P_{j,c}[n-1]/G_{j,P} \\ P_{j+1}[n] = P_{j+1}[n] + P_{j,f}[n-1] \\ \hat{P}_{j+1,e} = \hat{P}_{j,c}/G_{j,P} \\ L_{j+1,P} = L_{j,P} - 1 + G_{j,P}, \end{cases} \quad (18)$$

for $n \in [2, N_{\text{PTmax}}]$ and $g \in \left[0, G_{j,P} - 1\right]$. Note that $P_j[1] = 0$ for $j \geq 1$. The second equation in (18) is justified by the fact that the next RAO should have already been occupied by queuing mcMTDs from either the current j-th RAO (since we carry out (18) in a strict order) or a previous RAO. On the contrary, when $\hat{P}_{j,c} + \hat{P}_{i,e} \leq d_P$ and $L_{j,P} > 1$, the subset merging operation is carried out and (18) is re-written as

$$\begin{cases} P_{j+L_{j,P}-1}[n] = P_{j+L_{j,P}-1}[n] + P_{j,c}[n-1] \\ P_{j+1}[n] = P_{j+1}[n] + P_{j,f}[n-1] \\ \hat{P}_{j+1,e} = \hat{P}_{j,e} + \hat{P}_{j,c} \\ L_{j+1,P} = L_{j,P} - 1. \end{cases} \quad (19)$$

Updating the $N_j[m]$, on the other hand, is more complicated due to the timings between messages of the modified RAP. In the example of Fig. 4, if the SBS4 fails in the first RAO (either due to a collision or an insufficient number of acknowledgments), it will retransmit in the second RAO at subframe 11 thanks to the prioritization rule. On the other hand, if the SBS4 succeeds and receives an acknowledgment in either of the subframes $\{4, 5, 6, 7\}$, the corresponding Msg3 queue can be updated before the third RAO at subframe 21 and the SBS4 can continue sending a preamble in that third RAO if needed. Otherwise, the SBS4 receives the acknowledgment in subframe 8, which will cause it to miss the third RAO and have to wait until the fourth RAO to continue requesting for resources. To simplify the model, we will assume the timing diagram in Fig. 4 to derive subsequent equations, but the extension to other timing diagrams is trivial. With that in mind, the evolution of $N_j[m]$ can be expressed as

$$\begin{cases} N_{j+1}[m] = N_{j+1}[m] + N_j[m] - N_{j,s}[m] \\ \qquad \forall m \in [0, i_t - 1] \\ N_{j+2}[m] = N_{j+2}[m] + \rho \cdot N_{j,s}[m-1] \\ \qquad \forall m \in [1, i_t - 1] \\ N_{j+3}[m] = N_{j+3}[m] + (1-\rho) \cdot N_{j,s}[m-1] \\ \qquad \forall m \in [1, i_t - 1], \end{cases} \quad (20)$$

where ρ is the probability that an acknowledged SBS receives its acknowledgment in the first four subframes of the RAR window. To calculate the probability ρ, let us first denote the total number of acknowledged entities in this j-th RAO by $P_{j,s} = \sum_{n=1}^{N_{\text{PTmax}}} P_{j,s}[n] + \sum_{m=0}^{i_t-1} N_{j,s}[m]$. If the number of acknowledged entities are within the capacity of the first four subframes, i.e., $P_{j,s} \leq 4 \cdot N_{RAR}$, then $\rho = 1$. Otherwise $4 \cdot N_{RAR} < P_{j,s} \leq N_{UL}$, there will be $P_{j,s}$ acknowledgments, $4 \cdot N_{RAR}$ of which are carried by the first four subframes. Thus $\rho = 4 \cdot N_{RAR}/P_{j,s}$. (8)−(20) together completely describe the evolution of $\vec{P}_j$. The updating process terminates in an RAO j_t where $P_{j_t} < 1$ and $L_{j_t,P} = 1$.

As a last step, we attempt to extract the relevant performance metrics from the model. In a GP system, the task of computing the average access delay reduces to counting the number of successful MTDs in each RAO. In the j-th RAO, the number of successful mcMTDs is $\sum_{n=1}^{N_{\text{PTmax}}} P_{j,s}[n]$. The

number of scMTDs "succeeding" in an RAO, meanwhile, relates to $N_{j,s}[m]$ as follows. When an SBS who has successfully obtained m acknowledgments until now succeeds in getting another acknowledgment, there will subsequently be $M_{m+1,s}$ successful scMTDs in the $(m+1)$-th "Msg3 slot" manifested from the new N_b-RB uplink grant. Therefore, the *total* number of successful MTDs in the j-th RAO, denoted by $P_{j,s,\text{tot}}$, is written as

$$P_{j,s,\text{tot}} = \underbrace{\sum_{n=1}^{N_{\text{PTmax}}} P_{j,s}[n]}_{\triangleq P_{j,s,\text{macro}}} + \underbrace{\sum_{m=0}^{i_t-1} N_{j,s}[m] \cdot M_{m+1,s}}_{\triangleq P_{j,s,\text{micro}}}. \quad (21)$$

It is reminded that $M_{m+1,s}$ and i_t are obtained via the Msg3 queue model described in section III-B. The average access delays of the mcMTDs, scMTDs, and overall population in the unit of "local" RAOs, i.e., RAOs seen by an Msg1 queue, are thus

$$E_{\text{macro}}[D] = \frac{\sum_{j=1}^{\min(I'_{max}, j_t)} j \cdot P_{j,s,\text{macro}}}{\sum_{j=1}^{\min(I'_{max}, j_t)} P_{j,s,\text{macro}}}, \quad (22)$$

$$E_{\text{micro}}[D] = \frac{\sum_{j=1}^{\min(I'_{max}, j_t)} j \cdot P_{j,s,\text{micro}}}{\sum_{j=1}^{\min(I'_{max}, j_t)} P_{j,s,\text{micro}}}, \quad (23)$$

$$E[D] = \frac{\sum_{j=1}^{\min(I'_{max}, j_t)} j \cdot P_{j,s,\text{tot}}}{\sum_{j=1}^{\min(I'_{max}, j_t)} P_{j,s,\text{tot}}}, \quad (24)$$

respectively, where I'_{max} is the PI's duration in the unit of local RAOs. The blocking rate denoted by P_b, meanwhile, is

$$P_b = 1 - \frac{\sum_{j=1}^{\min(I'_{max}, j_t)} P_{j,s,\text{tot}}}{N}. \quad (25)$$

On the other hand, the average number of PUSCH RBs consumed per *successful* MTDs, denoted by $\overline{n_P}$, can be approximated as (26), where p_H is the probability that a non-colliding Msg3 requires an HARQ retransmission. By viewing the transmission process of a collision-free Msg3 as a series of Bernoulli trials with success probability $1-p_H$, the average number of times a collision-free Msg3 needs for its successful reception is approximately $1/(1-p_H)$. Note that since p_H is small, the probability of a collision-free Msg3 being dropped after exceeding the number of allowed HARQ transmission is negligible. The three terms on the numerator of (26), as shown at the bottom of the page, are then explained as follows. The first term is the total number of RBs allocated for the successful SBSs in the j-th RAO. The second term is the number of *additional* RBs needed for eventual delivery of the non-colliding Msg3 of $P_{j,s,\text{micro}}$ "successful" scMTDs in the j-th RAO. Note that the RBs used for the first attempts of these Msg3 are already counted in the first term, thus the subtraction by 1. The third term, meanwhile, represents the *total* number of RBs necessary for successful mcMTDs in the j-th RAOs to deliver their non-colliding Msg3 to the eNB.

D. FIXED ADDITION TO AVERAGE ACCESS DELAY

The delay obtained from (24) is only from a contention resolution perspective. In practical implementations, there are additional delays, which are summarized as

$$\Delta D = t_w + t_{RAR} + t_{proc} + t_{msg3} + t_{msg4}, \quad (27)$$

where t_w is the average time an MTD has to wait until the first RAO of its chosen Msg1 queue, t_{RAR} is the average time the MTD has to wait from the start of an RAO in which it succeeds until the reception of the corresponding acknowledgment. t_{proc} is the time it takes from the acknowledgment reception until the MTD can actually start transmitting Msg3 on PUSCH, while t_{msg3} and t_{msg4} are average delays incurred by Msg3 and Msg4, including their HARQ retransmissions.

To calculate t_w, we assume that the global RAO period is t_{RAO} subframes. Note that $t_{\text{RAO}} \times N_{\text{Q}}$ must be less than or equal the period of the local RAOs (denoted by $t_{\text{RAO},loc}$). Since the MTDs choose its Msg1 queue randomly, t_w can be written as

$$t_w = \frac{1}{N_{\text{Q}}} \sum_{i=1}^{N_{\text{Q}}} (i-1) \cdot t_{\text{RAO}} = \frac{t_{\text{RAO}}\left(N_{\text{Q}}-1\right)}{2}. \quad (28)$$

Also, t_{RAR} can be approximated as $(2 + W_{\text{RAR}}/2)$ subframes. t_{proc} is usually assumed to be fixed at 5 subframes. The sum of $t_{msg3} + t_{msg4}$, on the other hand, is given as equation (44) in [5]. The average access delay in unit of subframes is then found as $\overline{D} = t_{\text{RAO},loc} \cdot E[D] + \Delta D$.

When ΔD is taken into account, the number of local RAOs available for access during PI in our theoretical model is reduced, i.e., our model will not count the last few RAOs because even if the MTDs successfully transmit their preambles in those RAOs, they will most likely not finish the rest of RAP in time. Thus, $I'_{max} = \lfloor (I_{max} - \Delta D)/t_{\text{RAO},loc} \rfloor$ where I_{max} is the PI's duration but in subframes unit.

IV. SIMULATION RESULTS

In this section, system-level simulations using MATLAB programming are performed to assess the performance of the modified RAP and to verify the theoretical delay model. For the sake of discussion, we compare our proposal with one of the most well-performed GP-based schemes, i.e., the optimal

$$\overline{n_P} = \frac{\sum_{j=1}^{\min(I'_{max}, j_t)} \left\{ \sum_{m=0}^{i_t-1} N_{j,s}[m] \cdot N_b + P_{j,s,\text{micro}} \cdot \left(\frac{1}{1-p_H} - 1 \right) + P_{j,s,\text{macro}} \cdot \frac{1}{1-p_H} \right\}}{\sum_{j=1}^{\min(I'_{max}, j_t)} P_{j,s,\text{tot}}} \quad (26)$$

group paging (OGP) [12]. The main idea of OGP is to assume that the eNB knows the exact number of MTDs N in the paged group and tries to redistribute them evenly over $\lfloor I_{max}/t_{\text{RAO}} \rfloor$ RAOs in the PI to keep the average number of MTDs newly initiating the RAP per RAO at an optimal level M_{arv}. If such task is not possible, i.e., $\lceil N/\lfloor I_{max}/t_{\text{RAO}} \rfloor \rceil > M_{arv}$, a portion of the paged group will be proactively prevented from initiating the RAP. Note that the OGP still assumes the conventional backoff-based contention resolution mechanism. Also, while its assumption of knowing the exact N is often not achievable in practice, OGP serves as an upper performance bound for conventional GP system and is thus often used as a benchmark in, e.g., [14], [21].

A. SIMULATION SETUP

Simulation parameters are presented in Table 4, most of which follow [12]. Note that while the expected device density of the mMTC use case is as high as 1 million per square kilometer [2], it does not necessarily mean that all of those MTDs are always active. In practice, typically only a small fraction, e.g., 1%, of the population is active at a given time. Furthermore, since the network is in charge of GIDs allocation, it can proactively limit the maximum number of MTDs per paged group to a manageable level. Therefore, in this paper, we consider a group size of up to $N = 10000$, which is in line with most existing GP-based works.

TABLE 4. Simulation Parameters.

Parameters	Values
Number of MTDs in the paged group	N = 1000 to 10000
Number of SBSs	$N_{\text{SBS}} = 20, 30, 40$
Covered ratio	c = 0 to 1
RAO periodicity	t_{RAO} = 5 ms
Subframe length	1 ms
Number of preambles	$R = 54$
Maximum number of preamble transmissions	$N_{\text{PTmax}} = 16$
RAR window size	W_{RAR} = 5 subframes
Number of acknowledgments per RAR	$N_{\text{RAR}} = 3$
Number of reserved RBs per grant for an SBS	$N_b = 10$
Preamble detection probability for n-th preamble transmission	$p_n = (1 - 1/e^n)$ for mcMTDs and 1 for SBSs
Backoff Indicator	BI = 20 ms
Retransmission probability for Msg 3 & 4	$p_H = 0.1$
Maximum number of Msg 3 & 4 HARQ transmissions	5
Round-trip time of Msg 3 (Msg 4)	8 (5) subframes

The "covered ratio" is exclusive to SCAGP and represents the total fraction of MTDs covered by small-cells. These scMTDs are assumed to distribute evenly among the SBSs, e.g., with a cover ratio of 0.5 and $N_{\text{SBS}} = 20$, each small-cell is assumed to cover 2.5% of the population. Meanwhile, the delay between a scMTD's Msg3 transmission and its reception of the new format feedback, i.e., T_f, is set to 8ms which is the same as Msg3 round-trip time in the 3GPP setup. The number N_b of RBs reserved for a successful SBS is heuristically fixed at 10 to offer a good tradeoff between the performance and the resource consumption. The designated subset sizes for the Msg1 queues and Msg3 queues, i.e., d_P and d_M, are set to their respective optimal values of 22 and 10 (see Table 2). Also, given that $R = 54$ and $N_b = 10$, we have

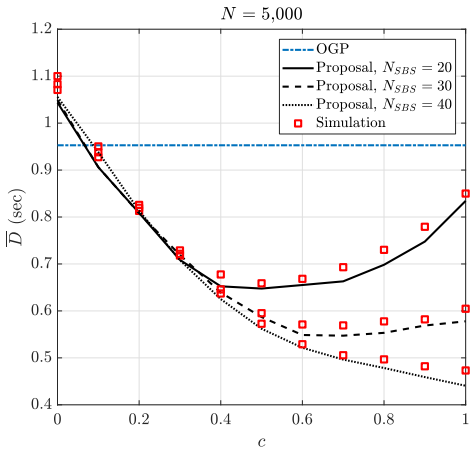


FIGURE 5. Average access delay, $I_{max} = \infty$.

$U_R = 378$ and $U_{N_b} = 50$. As for the OGP system, the setup yields $M_{arv} = 14$. The performance of the two systems is then evaluated via three metrics:

- *Average access delay* $\overline{D}$: the average duration from the start of the paging process until a successful MTD finishes.
- *Blocking rate* P_b: the ratio between number of blocked MTDs, i.e., the number of MTDs that are unable to finish the RAP within I_{max} subframes or exceed N_{PTmax} attempts, and total number of MTDs N. To ensure a fair comparison, the prevented MTDs in the OGP system and scMTDs who undergo N_{PTmax} consecutive Msg3 collisions in our system are also considered blocked.
- *Average number of PUSCH RBs consumed per successful device* $\overline{n_P}$: the ratio between the total number of PUSCH RBs *reserved* for Msg3 transmissions and number of successful devices. It is reminded here that whenever an SBS succeeds, N_b RBs are always reserved although some of them may not be used at all due to the randomness in RB selection of scMTDs.

B. UNBOUNDED PERFORMANCE

We initially ignore PI's duration, which results in negligible blocking rate (P_b) for both systems. The corresponding average access delay $\overline{D}$ is shown in Fig. 5 where N is fixed at 5,000. It is seen that the SCAGP system offers a significantly lower delay compared to OGP when a notable part of the population is under small-cell coverage. For example, when half of the MTDs are covered by 20 SBSs, SCAGP scores an access delay of 659ms, which is well below OGP's result at 953ms. This is because the modified RAP can make use of additional resources on the PUSCH instead of solely relying on the PRACH. Note that the access delay in the unbounded scenario is an important measure because it correlates with the blocking rate when the systems are bounded by the PI.

Looking at SCAGP alone reveals that access delay decreases rapidly when c is initially increased and slowly or even rises (when N_{SBS} is low) afterward. This is because

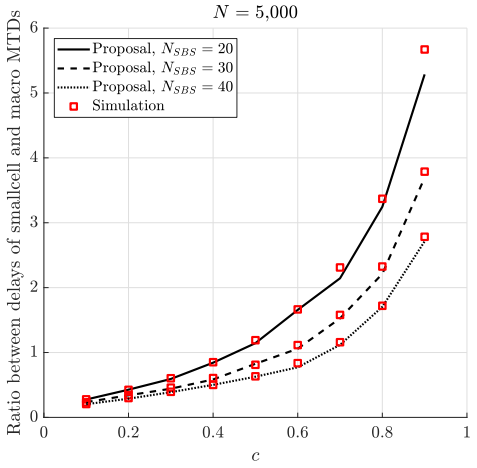


FIGURE 6. Ratio between delays of scMTDs and mcMTDs, $I_{max} = \infty$.

the load is initially shared in a balanced manner between the PRACH and the PUSCH, which results in a great improvement. When all the random access load is placed on the PUSCH, i.e., $c = 1$, the overload issue resurfaces and prevents further gain. On the other hand, given the same covered ratio, the cases with higher N_{SBS} show a lower delay since the same number of covered MTDs are now spread among more SBSs, which results in less local contention. It should however be warned that if N_{SBS} is increased without bound, the overload issue will recur on the PRACH because *all* SBSs are prioritized in Msg1 queue. We also see that the theoretical model does a good job of predicting these trends of $\overline{D}$. As such, it can be used for quickly estimating configurations of c that yield low delays. Also, despite not being shown, we have confirmed that the delays of both SCAGP and OGP increase linearly with N given any fixed c. The relative gaps between the lines in Fig. 5 thus hold regardless of N. We also verified that the theoretical model performs well when N varies.

Another aspect worth investigating is the difference in delay of scMTDs and mcMTDs, since a big discrepancy in this regard is usually not desirable. The ratio between the two delays with respect to c is thus plotted in Fig. 6. This ratio starts low but increases quickly, and the c at which it achieves the value of 1, i.e., same average delay across the MTDs, are approximately 0.5, 0.6, 0.7 for $N_{\text{SBS}} = 20, 30, 40$ respectively. Unsurprisingly, these balancing points are also the c at which the delay diminishing return starts taking effect (c.f. Fig. 5). This highlights once more the importance of balancing the PRACH and the PUSCH load. The theoretical results, which match very well with simulation, were calculated as

$$\frac{t_{\text{RAO},loc} \cdot E_{\text{micro}}[D] + \Delta D}{t_{\text{RAO},loc} \cdot E_{\text{macro}}[D] + \Delta D},$$

where $E_{\text{macro}}[D]$, $E_{\text{micro}}[D]$, and ΔD are respectively found in (22), (23), and (27) while $t_{\text{RAO},loc} = 10$ms given the environment in Table 4.

As the performance gain of SCAGP comes mostly from the ability to exploit the PUSCH, it is important to look at the average amount of PUSCH RBs required to handle one successful device plotted in Fig. 7. The number rises linearly with c. At $c = 0.5$, SCAGP already consumes one more RB per successful device compared to OGP. It is therefore advised against going any higher than this mark as there is negligible gain at the cost of significant resource consumption. Interestingly, given a fixed c, adding more SBSs does not cost more PUSCH RBs but can reduce the delay (see Fig. 5), which implies a true gain. The theoretical model, again, matches well with simulation and can thus be used by network operators to secure cost-to-performance targets.

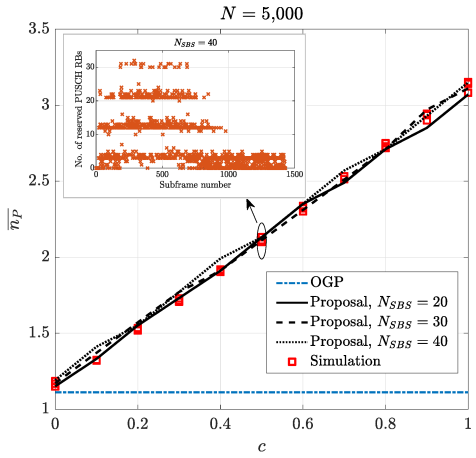


FIGURE 7. Average number of PUSCH RBs per successful device and number of PUSCH RBs reserved over time of SCAGP, $I_{max} = \infty$.

It should be noted that although one more RB *per successful device* may seem extravagant given the massive population, the total cost does not simultaneously burden the system but spreads out over time. The instant number of reserved PUSCH RBs in each subframe, taken from a single simulation run with $N_{\text{SBS}} = 40$ and $c = 0.5$, is shown in the top left corner of Fig. 7. As seen, this number hovers mostly around the 0, 10, 20, and 30 marks. This is because in the SCAGP system, when one, two, or three SBSs are to be acknowledged in the same RAR in a subframe, the number of reserved RBs 5 subframes later (after the processing delay) will spike to N_b, $2N_b$, or $3N_b$ respectively. The occasional overshoots, meanwhile, are due to some additional RBs being reserved for successful mcMTDs in the same RAR and/or Msg3's HARQ retransmissions of some previous MTDs. Note that the timings in the simulation setup prevent the overlap between the RAR windows of different Msg1 queues so that in one subframe, at most $N_{RAR} = 3$ SBSs are acknowledged. Additionally, although the maximum number of reserved RBs in a subframe of our system may reach as high as 34 RBs (equivalent to a 6.12 MHz bandwidth usage), it happens discontinuously for a short duration of only 1 second and should not severely disrupt other services running on the PUSCH. This is relevant in the 5G context where the demanding human-centric traffic

is expected to migrate toward mmWave bands, leaving more spaces for the mMTC service in current LTE bands.

C. BOUNDED PERFORMANCE

Now, we set $I_{max} = 1000$ subframes, i.e., 1 second, to see how well the two systems perform when bounded by the PI. A natural interest associated with this assumption would be the number of supportable devices and thus, the performance metrics will be mainly analyzed as functions of N at a fixed $c = 0.5$.

Starting with access delay plotted in Fig. 8 with respect to N, we see that the OGP system starts showing non-linear behaviors at $N = 3{,}000$ before becoming flat at $N \geq 4{,}000$. Such non-linearity is also observed in the SCAGP system at around $N = 4{,}000$, but the system still outperforms the OGP until $N = 6{,}000$ mark beyond which its delay converges to a value that is 14.3% higher than OGP regardless of N_{SBS}. We will later show that OGP's delay characteristic actually comes from the fact that a huge part of the paged group is prevented from initiating the RAP, and that SCAGP system can support significantly more MTDs at a lower to slightly higher $\overline{D}$ as seen in the figure. The simulation results in Fig. 8 also confirm the correctness of the theoretical model. It should be noted that in practical bounded scenarios, the data should be relevant as long as the corresponding device finishes accessing the network within the threshold I_{max}. A lower P_b is therefore usually of more importance than a shorter delay of the already successful devices.

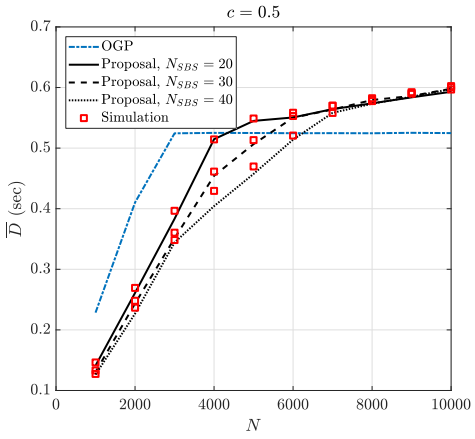


FIGURE 8. Average access delay w.r.t. N, $I_{max} = 1$s.

Given such a short I_{max}, the blocking rate is no longer negligible and is thus plotted in Fig. 9. It is seen that despite the good delay, the P_b of OGP quickly deteriorates as N is increased. SCAGP, on the other hand, is able to provide connections for many more MTDs. For example, given a target blocking rate of 0.5, the OGP system can handle a group size of approximately 5,000 while the SCAGP system can accommodate anywhere from 7,500 to 9,500 MTDs. More importantly, OGP tries to keep the number of new devices per RAO constant and thus can only admit a fixed number of MTDs given a certain I_{max}. On the contrary, the performance

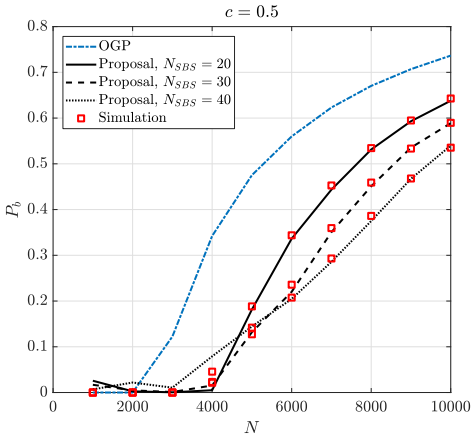


FIGURE 9. Blocking rate w.r.t. N, $I_{max} = 1$s.

of SCAGP can usually be improved by deploying more SBSs. Also, such a higher N_{SBS} is usually coupled with a higher c, which can further reduce P_b as indicated in Fig. 10. The theoretical model can capture the discussed tendencies of P_b with a very high accuracy and can thus be reliably used to quantify the tradeoff between having a bigger group size and a higher blocking rate.

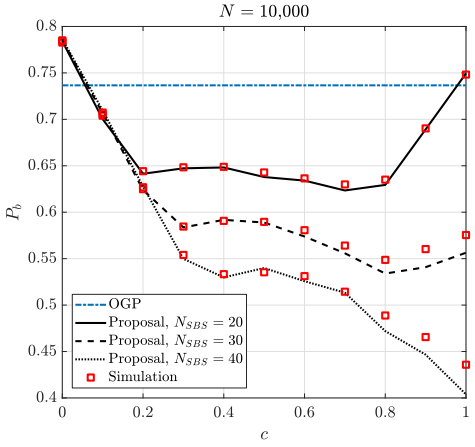


FIGURE 10. Blocking rate w.r.t. c, $I_{max} = 1$s.

The number of PUSCH RBs per *successful* MTD is shown in Fig. 11. It is seen that in the case of $N_{\text{SBS}} = 20$, the SBSs are outnumbered by mcMTDs on PRACH, and it is not until much later when the mcMTDs are mostly resolved that the SBSs can continuously seize as many RBs as they want. However, that stage is never reached due to a limited I_{max}, and the number of RBs consumed is thus low enough to offset the decrease in the number of successful MTDs, which explains the relatively stable trend of $\overline{n_P}$. This is not the case for higher N_{SBS} configurations where the SBSs can seize a large amount of RBs much sooner (c.f. Fig. 7) to serve more devices during I_{max}. Nevertheless, the increase in success rate is not enough to offset an initial higher amount of RBs consumption, causing $\overline{n_P}$ to slowly go up. The OGP system,

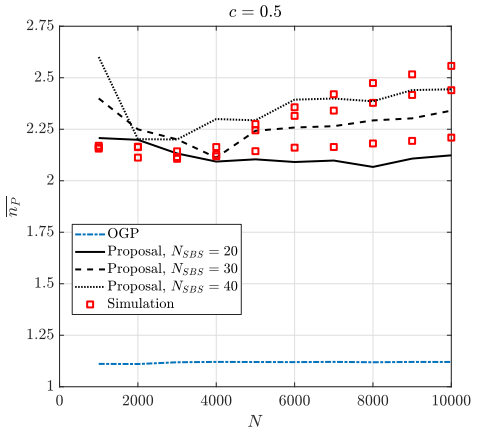


FIGURE 11. Average number of PUSCH RBs per successful device, $I_{max} = 1$s.

on the contrary, stays low and invariant for obvious reasons. It is however important to keep in mind that when the system is bounded by a short PI, a lower P_b is more likely to be favored over a low PUSCH consumption. The theoretical model grossly overestimates $\overline{n_P}$ when the number of MTDs is insufficient for our approximation in section III-C, but otherwise showcases relatively good accuracy (note the scale of $\overline{n_P}$ axis) in the interested massive access region.

Finally, it is noted that since the eNB is always in complete control of resource allocations, it can proactively fix N_b to a lower value to avoid exhausting the PUSCH RBs, if necessary. The performance loss associated with a lower N_b would then be a higher access delay in unbounded case and a higher blocking rate in the bounded case. The specific N_b needed to achieve certain performance targets can always be predicted with good accuracy using the proposed theoretical model.

V. CONCLUSION

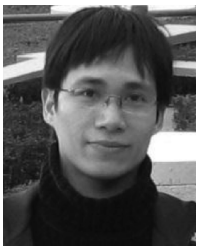
In this paper, we have exploited the possibility that a portion of the massive MTD population is covered by both a macro LTE cell and multiple small-cells to propose a random access load sharing mechanism between the PRACH and the PUSCH to address the PRACH resource shortage in the conventional GP system. In particular, the SBSs contended with each other and with mcMTDs over the PRACH preambles. Whenever an SBS succeeded, a multi-PUSCH-RB uplink grant is issued. The scMTDs of the successful SBS then contended locally to transmit Msg3 on the multiple reserved PUSCH RBs. To resolve preambles and Msg3 collisions, an existing DQ protocol is used instead of the conventional random backoff protocol. System-level simulations showed that when the load is balanced between PRACH and PUSCH, our system achieves a significant improvement in terms of access delay and blocking rate at a reasonable cost of approximately one more PUSCH RB per successful MTD. The blocking rate can be further reduce by deploying more SBSs and cover more MTDs if the associated increase in the PUSCH cost can be justified. More importantly, we also formulated a theoretical model to derive the performance metrics of the proposed system. As verified by the simulations, the model can capture the characteristics of the blocking rate, access delay, and amount of consumed PUSCH resources with high accuracy and thus can be used to predict desirable configurations.

REFERENCES

[1] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The Next Generation Wireless Access Technology*. New York, NY, USA: Academic, 2018.
[2] *Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, document ITU-R M.2083-0, ITU, IMT Vision, Sep. 2015.
[3] *Study on RAN Improvements for Machine-Type Communications*, 3rd Generation Partnership Project (3GPP), document TR 37.868 V11.0.0, Sep. 2011.
[4] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE–A suitable for M2M communications? A survey of alternatives," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 4–16, 1st Quart., 2014.
[5] I. Leyva-Mayorga, L. Tello-Oquendo, V. Pla, J. Martinez-Bauset, and V. Casares-Giner, "On the accurate performance evaluation of the LTE–A random access procedure and the access class barring scheme," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 7785–7799, Dec. 2017.
[6] *Radio Resource Control (RRC) Protocol Specification*, 3rd Generation Partnership Project (3GPP), document TS 36.331 V10.5.0, Mar. 2012.
[7] M. Tavana, V. Shah-Mansouri, and V. W. S. Wong, "Congestion control for bursty M2M traffic in LTE networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 5815–5820.
[8] H. Jin, W. T. Toor, B. C. Jung, and J.-B. Seo, "Recursive pseudo-Bayesian access class barring for M2M communications in LTE systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 8595–8599, Sep. 2017.
[9] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. S. Wong, "D-ACB: Adaptive congestion control algorithm for bursty M2M traffic in LTE networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9847–9861, Dec. 2016.
[10] *User Equipment (UE) Procedures in Idle Mode*, 3rd Generation Partnership Project (3GPP), document TS 36.304 V13.0.0, Feb. 2016.
[11] R. Harwahyu, X. Wang, R. F. Sari, and R.-G. Cheng, "Analysis of group paging with pre-backoff," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, pp. 1–9, Feb. 2015.
[12] O. Arouk, A. Ksentini, and T. Taleb, "Group paging-based energy saving for massive MTC accesses in LTE and beyond networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1086–1102, May 2016.
[13] L. Yan, Y. Li, R. Zhang, Y. Ruan, and T. Li, "Improved throughput stabilizing scheme for group paging in mMTC: A traffic scattering perspective," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2427–2431, Dec. 2019.
[14] W. Cao, A. Dytso, G. Feng, H. V. Poor, and Z. Chen, "Differentiated service-aware group paging for massive machine-type communication," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5444–5456, Nov. 2018.
[15] H. S. Jang, B. C. Jung, and D. K. Sung, "Dynamic access control with resource limitation for group paging-based cellular IoT systems," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5065–5075, Dec. 2018.
[16] H. M. Gursu, M. Vilgelm, W. Kellerer, and M. Reisslein, "Hybrid collision avoidance-tree resolution for M2M random access," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 4, pp. 1974–1987, Aug. 2017.
[17] R. Harwahyu, R.-G. Cheng, and R. F. Sari, "Consecutive group paging for LTE networks supporting machine-type communications services," in *Proc. IEEE 24th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2013, pp. 1619–1623.
[18] C.-H. Wei, R.-G. Cheng, and F. M. Al-Taee, "Dynamic radio resource allocation for group paging supporting smart meter communications," in *Proc. IEEE 3rd Int. Conf. Smart Grid Commun. (SmartGridComm)*, Nov. 2012, pp. 659–663.
[19] R.-G. Cheng, F. M. Al-Taee, J. Chen, and C.-H. Wei, "A dynamic resource allocation scheme for group paging in LTE-advanced networks," *IEEE Internet Things J.*, vol. 2, no. 5, pp. 427–434, Oct. 2015.
[20] V. B. Mišić, J. Mišić, X. Lin, and D. Nerandzic, "Capillary machine-to-machine communications: The road ahead," in *Ad-Hoc, Mobile, and Wireless Networks*. Berlin, Germany: Springer, 2012, pp. 413–423.

[21] Q. Pan, X. Wen, Z. Lu, W. Jing, and L. Li, "Cluster-based group paging for massive machine type communications under 5G networks," *IEEE Access*, vol. 6, pp. 64891–64904, 2018.
[22] M. Park, "IEEE 802.11ah: Sub-1-GHz license-exempt operation for the Internet of Things," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 145–151, Sep. 2015.
[23] G. Farhadi and A. Ito, "Group-based signaling and access control for cellular machine-to-machine communication," in *Proc. IEEE 78th Veh. Technol. Conf. (VTC Fall)*, Sep. 2013, pp. 1–6.
[24] S.-H. Wang, H.-J. Su, H.-Y. Hsieh, S.-P. Yeh, and M. Ho, "Random access design for clustered wireless machine to machine networks," in *Proc. 1st Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Jul. 2013, pp. 107–111.
[25] H.-W. Kao, Y.-H. Ju, and M.-H. Tsai, "Two-stage radio access for group-based machine type communication in LTE–A," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 3825–3830.
[26] B. Han and H. D. Schotten, "Grouping-based random access collision control for massive machine-type communication," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–7.
[27] K. Lee, J. Shin, Y. Cho, K. S. Ko, D. K. Sung, and H. Shin, "A group-based communication scheme based on the location information of MTC devices in cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 4899–4903.
[28] N. Zhang, G. Kang, J. Wang, Y. Guo, and F. Labeau, "Resource allocation in a new random access for M2M communications," *IEEE Commun. Lett.*, vol. 19, no. 5, pp. 843–846, May 2015.
[29] A.-T.-H. Bui, C. T. Nguyen, T. C. Thang, and A. T. Pham, "A comprehensive distributed queue-based random access framework for mMTC in LTE/LTE–A networks with mixed-type traffic," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12107–12120, Dec. 2019.
[30] A.-T.-H. Bui, C. T. Nguyen, T. C. Thang, and A. T. Pham, "Design and performance analysis of a novel distributed queue access protocol for cellular-based massive M2M communications," *IEEE Access*, vol. 6, pp. 3008–3019, 2018.
[31] C.-H. Wei, G. Bianchi, and R.-G. Cheng, "Modeling and analysis of random access channels with bursty arrivals in OFDMA wireless networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 1940–1953, Apr. 2015.

ANH-TUAN H. BUI (Graduate Student Member, IEEE) received the B.E. degree in communications engineering from the Hanoi University of Science and Technology, Vietnam, in 2016, and the M.S. degree in computer network systems from The University of Aizu, Japan, in 2018, where he is currently pursuing the Ph.D. degree. His study in Japan is funded by the Japanese Government Scholarship (MonbuKagakusho). His research interests include protocol designs, system modeling, and performance evaluation for next generation (5G) wireless networks.



CHUYEN T. NGUYEN received the M.S. degree in communications engineering from National Tsing-Hua University, Taiwan, in 2008, and the Ph.D. degree in informatics from Kyoto University, Japan, in 2013. In 2014, he was a Visiting Researcher with The University of Aizu, Japan. He is currently an Assistant Professor with the School of Electronics and Telecommunications, Hanoi University of Science and Technology, Vietnam. His research interests include statistics and optimization algorithms and their applications in wireless communication systems. He received the Fellow Award from the Hitachi Global Foundation, in 2016.

TAKAFUMI HAYASHI (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in applied physics from The University of Tokyo, Tokyo, Japan, in 1985, 1987, and 1992, respectively. From 1989 to 1992, he was a Research Associate with The University of Tokyo. He is currently a Professor with Nihon University. His current research interests include sequence design, signal processing, image analysis, mobile communications, intelligent infrastructure, messaging networks, secure cloud computing, smart grids, and information security management. He is a member of ACM, EURASIP, IET, and SPIE, and a Senior Member of IEICE.

ANH T. PHAM (Senior Member, IEEE) received the B.E. and M.E. degrees in electronics engineering from the Hanoi University of Technology, Vietnam, in 1997 and 2000, respectively, and the Ph.D. degree in information and mathematical sciences from Saitama University, Japan, in 2005. From 1998 to 2002, he was with NTT Corporation, Vietnam. Since 2005, he has been a Faculty Member with The University of Aizu, where he is currently a Professor and the Head of the Computer Communications Laboratory, Division of Computer Engineering. His research interests include communication theory and networking with a particular emphasis on modeling, design, and performance evaluation of wired/wireless communication systems and networks. He has authored/coauthored over 200 peer-reviewed articles on these topics. He is a member of IEICE and OSA.