

# r-Hint: A Message-Efficient Random Access Response for mMTC in 5G Networks

Teng-Wei Huang, Yi Ren, Kate Ching-Ju Lin, and Yu-Chee Tseng

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

{tengwei, renyi, katelin, yctseng}@cs.nctu.edu.tw

**Abstract**—Massive Machine Type Communication (mMTC) has attracted increasing attention due to the explosive growth of IoT devices. Random Access (RA) for a large number of mMTC devices is especially difficult since the high signaling overhead between User Equipments (UEs) and an eNB may overwhelm the available spectrum resources. To address this issue, we propose “respond by hint” (*r-Hint*), an ID-free handshaking protocol for contention-based RA in mMTC. The core idea of *r-Hint* is to avoid sequentially notifying contending UEs of their IDs by broadcasting a *hint* in the RA Response (RAR). To do so, we exploit the concept of prime *factorization* and hashing to encode the *hint* such that UEs can extract their required information accordingly. Our simulation results show that *r-Hint* reduces the RAR message size by 20%–40%. Such reduction can be translated to around 50% improvement of spectrum efficiency in LTE-M.

**Index Terms**—communication protocol, LTE-M, MTC, Random Access, 5G

## I. INTRODUCTION

The LTE standard defines a handshaking mechanism, called Random Access (RA) procedure, which is triggered when User Equipments (UEs) attempt to access the network. This contention-based procedure consists of four steps, as illustrated in Fig. 1(a). Each contending UE first randomly selects one preamble from the available preamble sequences (typically 64 in Random Access Channel, RACH), and requests for access by sending the selected preamble (Msg.1). The eNB then replies a RA Response (RAR) message (Msg.2) including all RA preambles successfully detected in Msg.1. Then, the UE receiving Msg.2 can make a Radio Resource Control (RRC) connection request (Msg.3) and complete the RA procedure if a RRC connection setup is received (Msg.4).

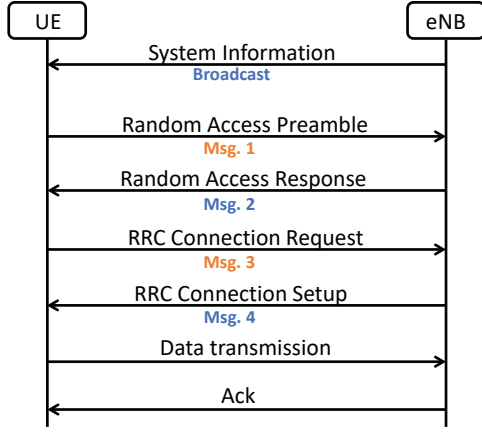
A UE may fail to connect to the eNB when its preamble is not detected or when collisions occur. If this happens, the UE needs to backoff and contend again in the next round of RACH. To increase the success probability of RA, existing work either allocates more RACHs [1]–[4] or uses Extended Access Barring (EAB) [5]–[9] to control the number of contending UEs per RACH. However, for massive MTC (mMTC) devices in an eNB, maintaining a similar success probability implies more frequent RACHs, as shown in Fig. 1(b), which significantly increases the overhead. Furthermore,

since the RAR messages are sent on the Physical Downlink Shared Channel (PDSCH), which is a data channel, the increasing overhead also occupies more resources in PDSCH and, thus, reduces the spectrum efficiency of downlinks, as shown in Fig. 1(c). In this work, our goal is to reduce the size of RAR message by a more efficient coding scheme.

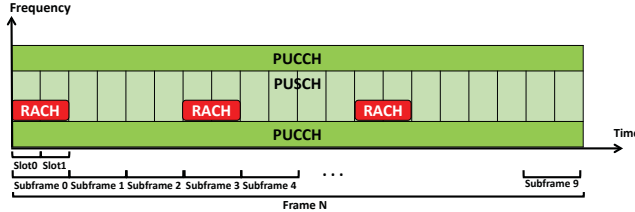
The root cause of inefficiency in the current RAR design is its one-to-one coding scheme to notify each detected contending UE of its Random Access Preamble Identifier (RAPID) and Temporary Cell Radio Network Temporary Identifiers (TC-RNTI). This problem becomes even worse for mMTC where much more UEs need to share a limited 1.4MHz bandwidth (only 6 RBs per subframe) in LTE-M [10], thereby very likely consuming most of the spectrum resources for one-to-one notification. To address this concern, we ask a fundamental question: Is it possible to avoid using an one-to-one notification mechanism when it comes to need informing a set of UEs some individual (personalized) information?

In this work, we propose “respond by hint” (*r-Hint*), an ID-free handshaking protocol for RA in mMTC. The key idea of *r-Hint* is to exploit prime *factorization* and hashing to encode the hint. Fundamentally, some integers can be factorized into a product of unique primes, giving UEs an opportunity to extract their individual information. By combining *factorization* with hashing, we can further assign identities to UEs without specifically announcing the aforementioned RAPID and TC-RNTI. We also show how to combine our hint scheme with a naive bitmap method to significantly reduce the size of RAR message. Our simulation results show that our design reduces the RAR message size by 20%–40%. Such overhead reduction can be translated to around 50% improvement of spectrum efficiency in LTE-M.

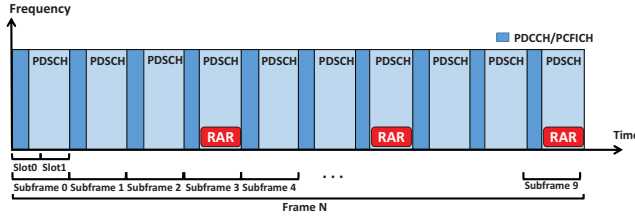
The rest of the paper is organized as follows. Section II reviews the LTE RA procedure and some related work. The design of our *r-Hint* is described in Section III. Section IV shows our simulation results. Section V concludes this work.



(a) Contention-based RA procedure in LTE



(b) Uplink frame structure in LTE



(c) Downlink frame structure in LTE

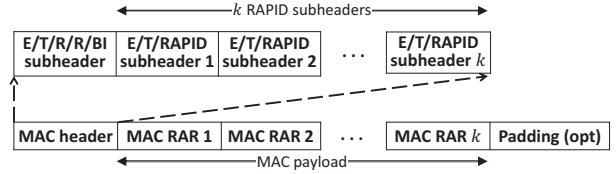
Fig. 1: Contention-based RA procedure and frame structure in LTE

## II. BACKGROUND AND RELATED WORK

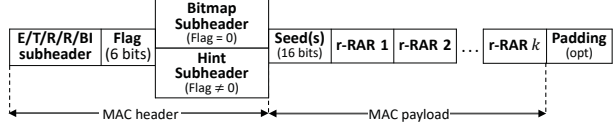
### A. RAR Message Format

Recall the contention-based RA procedure in Fig. 1(a). If the eNB successfully detects the RA preamble of a UE from Msg.1, it will reply the RAR message (Msg.2) on PDSCH. Since multiple UEs may be successfully detected, the current LTE adopts a one-to-one notification mechanism, as shown in Fig. 2(a).

1) *MAC Header*: A MAC header consists of multiple subheaders. Definitions of these symbols are summarized in Table I. The first subheader, Backoff Indicator (BI), is for all the contending UEs to learn when the next RACH will appear. Then,  $k$  RAPID subheaders follow, where  $k$  is the number of preambles successfully detected. Each RAPID subheader specifies the 6-bit ID of the corresponding preamble being detected (as there are 64 preambles). Contending UEs should scan all the RAPIDs and check if their selected preambles appear in any RAPID. If a UE finds its target RAPID, say with index  $i$ , it then locates the  $i$ -th MAC RAR in the payloads to extract its individual information. In other words, the



(a) The format of RAR message in LTE



(b) The format of RAR message in *r-Hint*

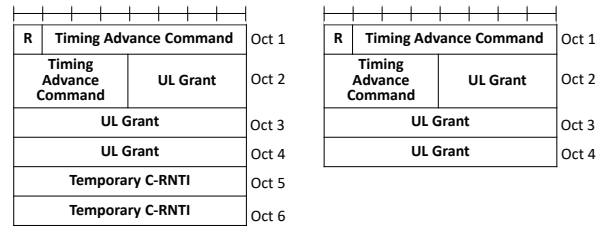
Fig. 2: Comparison with format of RAR message between LTE and *r-Hint*

Symbol	Definition
$E$	A flag indicating whether the next MAC header exists
$T$	A flag indicating whether the MAC sub-header contains a RAPID or a BI
$R$	Reserved bits
$BI$	Backoff Indicator, a 4-bit field indicating the time interval between the current and the next RACH
$RAPID$	A 6-bit field identifying a successfully decoded RA preamble identifier

TABLE I: Definitions of MAC sub-header in RAR message

order of RAPIDs matches exactly the order of MAC RARs in an one-to-one way. Note that UEs picking the same preambles will read the same MAC RARs, which leads to collisions in Msg.3. Then Msg.4 will resolve the collisions. A UE not finding its RAPID in the current RAR message will follow the backoff procedure and perform RA in the next RACH.

2) *MAC RAR*: Fig. 3(a) illustrates the format of MAC RAR. It includes timing and resource granting information for Msg.3 and TC-RNTI, a temporary ID assigned to the UE(s) for selecting the corresponding preamble. If there is no collision occurs in Msg.3, this TC-RNTI will become C-RNTI, the final unique ID



(a) Fields in each MAC RAR (b) Fields in each *r-RAR*

Fig. 3: Comparison with payload of RAR message between LTE and *r-Hint*

for this UE. The length of TC-RNTI is 16 bits, and the number of available IDs is 65523 (some reserved for special purposes). Due to its one-to-one notification nature, the size of RAR message grows linearly as the number of detected UEs increases. This overhead is large especially for mMTC in 5G. Our objective is to avoid the one-to-one notification mechanism in RAR message.

### B. Related Work

The existing works on improving RA efficiency can be classified into two categories: dynamic RACH adjustment and EAB control. References [1], [2] propose some mathematical models to derive performance gains by increasing the number of RACHs. In [3], a dynamic resource allocation algorithm is proposed to achieve a balance between the available data communications bandwidth and the RA success probability. The scheme in [4] determines the number of RACHs per frame based on applications' traffic model.

On the other hand, EAB control schemes have been intensively studied [5]–[9]. The schemes in [5], [6] dynamically adjust the EAB parameters to control the number of contending UEs in each RACH based on predicted traffic patterns. Reference [7] preallocates RA resources for different MTC classes to reduce collisions among UEs. The work [8] exploits access class barring and timing advance information to relieve RA loading. Another approach [9] leverages congestion control to alleviate collisions. While all the above solutions focus on collision avoidance, we investigate how to reduce the message size of RAR and improve spectrum utilization.

### III. *r-Hint* PROTOCOL FOR RAR

We now describe our *r-Hint* for reducing the size of RAR message. Our design is divided into two parts:

- Removing the use of RAPIDs in MAC Subheader.
- Removing the use of TC-RNTIs in MAC RARs.

The main idea is to avoid sequentially listing all notifications by encoding them in a shared message. Fig. 2(b) illustrates the proposed format of RAR message. The RAPIDs part is replaced by a *Flag* field followed by two options. We combine a naive *Bitmap* scheme and a *Prime Factorization* scheme. When the 6-bit *Flag* = 0, the *Bitmap* scheme is applied. The length of *Bitmap* is the number of preambles,  $N$  (typically  $N = 64$ ). When *Flag*  $\neq 0$ , the *Prime Factorization* scheme is applied and the value of *Flag* is the length of the *hint* subheader. The  $k$  MAC RARs are replaced by a *Seed* field followed by  $k$  "reduced RARs" (r-RARs). The *Seed* field is for UEs to decode their TC-RNTIs. Each r-RAR is the same as the original MAC RAR except that the *TC-RNTI* field is removed.

#### A. MAC Subheader

We design two schemes, namely *Bitmap* and *Prime Factorization*, to encode RAPIDs. The former is more

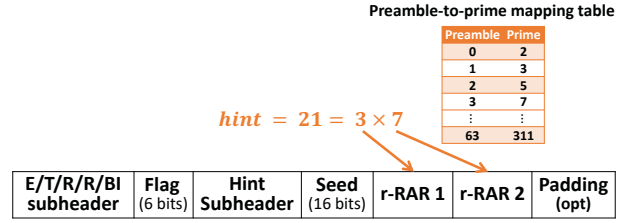


Fig. 4: The example of *Prime Factorization* scheme

suitable for a larger number of preambles are detected, while the latter is more suitable for a moderate number of detected preambles. The value of *Flag* determines which scheme is applied.

1) *Bitmap Scheme*: This scheme simply uses an  $N$ -bit bitmap (mask) to represent the results of preamble detection, where  $N$  is the number of available preambles used per RACH. Specifically, if the eNB detects the  $i$ -th preamble in Msg.1, it sets the  $i$ -th bit of the bitmap to '1'. Otherwise, it is set to '0'. All the contending UEs can check the bitmap and learn whether their preambles are successfully detected or not. On the other hand, to locate the correct r-RAR in the payloads, each UE simply counts the number of '1's in the bitmap before its own '1's. This number  $k$  implies the number of r-RARs in the payloads before its r-RAR.

2) *Prime Factorization Scheme*: The basic idea is to map each preamble position to an unique prime. Therefore, for 64 preambles, we need 64 primes. Fig. 4 shows the preamble-to-prime mapping table of the first 64 primes. This table can be embedded in the System Information Block (SIB) or pre-installed in UEs. Let  $P_1, P_2, \dots, P_k$  be the preambles being detected. The eNB calculates the *product*  $P = Prime(P_1) \times Prime(P_2) \times \dots \times Prime(P_k)$ , where  $Prime(P_i)$  is the prime mapped to  $P_i$ . The eNB then puts  $P$  in the Hint Subheader. When a contending UE receives  $P$ , it simply factorizes  $P$  and checks if  $Prime(P_i)$  is a factor of  $P$ , where  $P_i$  is its selected preamble. If so, it can further locate its r-RAR by counting the number of prime factors which are smaller than  $Prime(P_i)$ . This number represents how r-RARs are ordered in the MAC payload.

Fig. 4 illustrates an example, where two preambles, '1' and '3', are detected. Then  $P = Prime(1) \times Prime(3) = 21$ . UEs can factorize 21 into  $3 \times 7$ . So only UEs sending preambles '1' and '3' will locate their r-RARs and proceed to Msg.3. Note that  $P$  is a product of primes and its value depends on the number of preambles successfully detected. In the worst case, about 417 bits are needed to represent  $P$ . *Flag* is to serve this purpose. When more than 64 bits are needed, the eNB can choose the *Bitmap* scheme for lower cost of notification.

#### B. MAC RAR

Recall the MAC RAR in Fig. 3(a). One of its purpose is to notify UEs their TC-RNTIs, which is 16 bits. Give

$n$  UEs to be notified, it will take  $16n$  bits, introducing significant overhead. So, are there ways to notify the UEs *without* sending their TC-RNTIs? The basic idea is to use hashing function as an agreement between the eNB and the UEs. Assume that there exists a pre-defined hashing function  $h(\cdot)$  known by all UEs. Given the same input, the hashing function generates the same 16-bit output as TC-RNTI. In LTE-A, the eNB assigns different TC-RNTI to the UEs of which selected preambles being detected. In our design,  $TC-RNTI_i$  can be easily obtained by  $TC-RNTI_i = h(Prime(P_i))$ . As both the eNB and the UE know  $Prime(P_i)$ , the corresponding  $TC-RNTI_i$  is computed without transmitting it. Note that even if the eNB chooses the *Bitmap* scheme, each UE still knows its  $Prime(P_i)$  due to the mapping table. Therefore, with this hashing scheme, we can discard the 16-bit TC-RNTI field in each MAC RAR. The r-RAR (refer to Fig. 3(b)) only needs to carry the data other than TC-RNTIs.

One might concern that a hash function may not always guarantee one-to-one mapping. That is, different inputs may hash to the same TC-RNTI, leading to a conflict. To avoid this, the eNB broadcasts a seed  $s$  to control the hashing results. The *Seed* field is to carry the value of  $s$ , which is used as one of the input in the hashing operation, i.e.,  $TC-RNTI_i = h(Prime(P_i), s)$ . The eNB may try multiple seeds to avoid collision. Since the range of TC-RNTI is large (65523), the collision is very unlikely. In the rare case that collisions occur, the eNB can choose not including these UEs in the current RAR message, in which case they will contend in the next RACH. We will derive the numerical analysis in next section to check how many iterations of seed searching are required to guarantee nearly conflict-free TC-RNTI assignment.

#### IV. PERFORMANCE EVALUATION

We conduct simulations to evaluate the performance of our designs, in terms of the overhead of RAR, the computational complexity of conflict-free TC-RNTI assignment and the overall spectrum utilization.

##### A. The Overhead of MAC PDU

We first compare the overhead of the RAR message in our design to that in the conventional LTE. In the conventional LTE, each RAPID sub-header and MAC RAR cost 8 bits and 48 bits, respectively, for every detected preamble. Our design drops those RAPID sub-headers, but adds the cost of a 6-bit ‘Flag’, a hint/bitmap subheader, and a 16-bit ‘Seed’. By following the analysis in [11], we set the number of available preambles in each RACH to 64. Hence, the *Bitmap* scheme costs 86 bits, while the size of the *Prime Factorization* scheme equals the actual length of the *hint* plus 22 (for flag and seed). In addition, in our design, the MAC RAR for each detected preamble costs only 32 bits since the TC-RNTI field is discarded. We report the average result

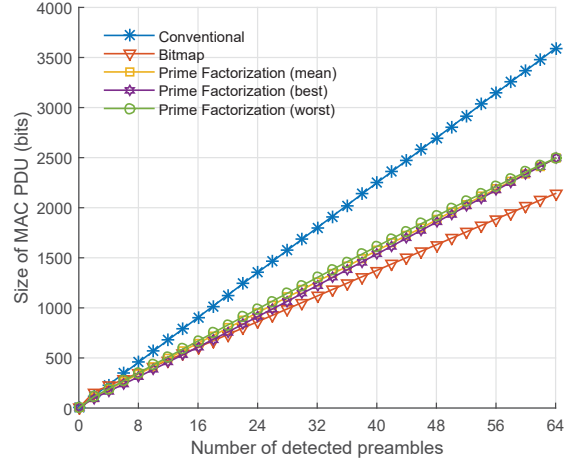


Fig. 5: Cost of MAC PDU

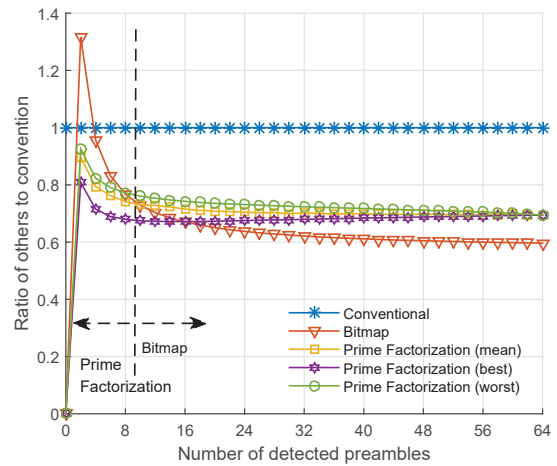


Fig. 6: Ratio of proposed scheme and conventional scheme

of 100 RACHs. Since the size of the *hint* depends on the product value of the primes corresponding to the selected preambles, we not only evaluate the mean cost but also calculate the cost of the best case (i.e., the selected preambles mapping to the smallest primes) and the cost of the worst case (i.e., the selected preambles mapping to the largest primes).

Fig. 5 plots the overhead of comparison schemes for various numbers of detected preambles. The figure shows that our scheme reduces the size of MAC PDU, especially when the number of detected preambles increases. Fig. 6 shows the ratio of the size of our designed format to that of the conventional LTE. The results verify that the *Bitmap* scheme has the fixed-length subheader and, thereby, is more efficient for a larger number of detected preambles. With proper adaptation, we can switch to use *Prime Factorization* when the number of the detected preambles is smaller than 9. Overall, the hybrid scheme

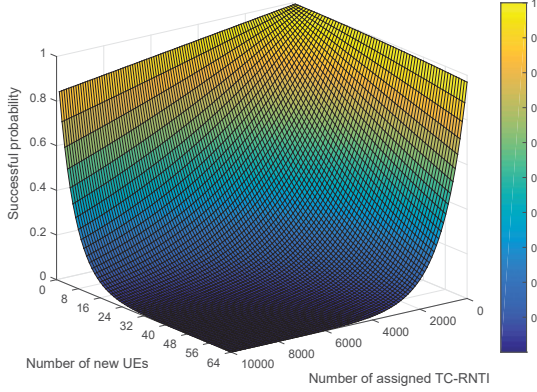


Fig. 7: Successful probability of unique ID assignment

reduces the size by about 20%–40%.

### B. Probability of Conflict-Free Seed Searching

Since the temporary C-RNTI might turn to the permanent C-RNTI for future data communications, the hashed IDs should be conflict-free not only inside a RACH but also across all the RACHs. We aim at deriving the probability of allocating  $k$  unique TC-RNTIs to  $k$  new UEs in a RACH, assuming that there exist  $M$  available TC-RNTIs in total and  $n$  out of  $M$  have been allocated to other UEs. Since each UE may be hashed to an integer value between  $[1, M]$ , the number of possible TC-RNTI assignments for the  $k$  UEs hence equals  $M^k$ . However, to ensure conflict-free, each of the  $k$  UEs should be assigned a distinct TC-RNTI from the  $(M - n)$  empty TC-RNTIs, i.e., not yet occupied by the  $n$  existing UEs. Hence, the number of all the feasible assignments should be  $k$ -permutations of  $(M - n)$ , i.e.,  $P(M - n, k) = \frac{(M - n)!}{(M - n - k)!}$ . The probability of successful conflict-free TC-RNTI assignment equals

$$\mathbb{P}_{succ} = \frac{P(M - n, k)}{M^k} = \frac{(M - n)! / (M - n - k)!}{M^k}. \quad (1)$$

Fig. 7 plots the successful probability for different scenarios of  $k$  (number of new UEs) and  $n$  (number of existing UEs) when we only pick a *random seed once*. We set the number of available unique TC-RNTIs  $M$  to 65,523. The figure shows that, as expected, the success probability of conflict-free TC-RNTI assignment decreases when either the number of new UEs or the number of occupied TC-RNTIs grows.

To increase the success probability, the eNB can test several seeds, defined as a random variable  $X$ , and check whether a conflict-free seed can be found in the  $x$ -th iteration of searching. Since each seed searching can be deemed as a Bernoulli trial with the success probability equal to  $\mathbb{P}_{succ}$ , the probability of finding a conflict-free seed in the  $x$ -th trial follows the geometric distribution, which can be expressed as follows:

$$\Pr(X = x) = (1 - \mathbb{P}_{succ})^{x-1} \mathbb{P}_{succ}. \quad (2)$$

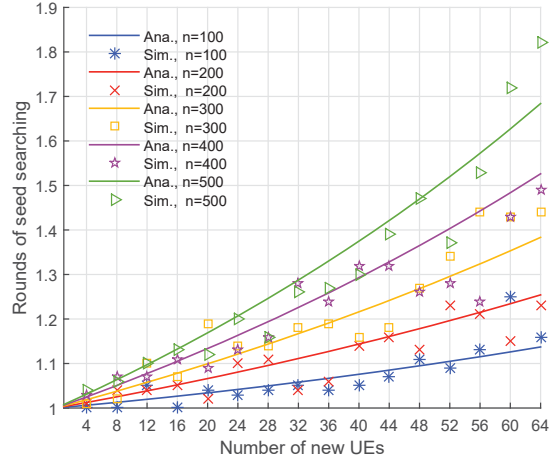


Fig. 8: Cost of seed searching

Then, the expected number of searching iterations required to obtain a conflict-free seed will be  $E[X] = \frac{1}{\mathbb{P}_{succ}}$ .

Fig. 8 plots the average rounds of seed searching to get a successful trail. The figure shows both the numerical result and the average simulation result of 100 RACHs for every setting. The results demonstrate that, for a moderate number of existing UEs, the expected iterations of seed searching is below 2, meaning that the cost of obtaining a conflict-free TC-RNTI assignment is negligible. The figure also shows that the simulation results match the numerical analysis quite well. We further testify the success probability of a fixed searching cost, i.e., 100 rounds of seed searching, as the number of existing UEs scales up. The results in Fig. 9 plots the conflict-free probability of 100 RACHs for various network scales. The results demonstrate that, when the number of contending new UEs is relatively small, e.g.,  $k \leq 20$ , with this reasonable searching cost, the conflict-free probability can be nearly 100% even when the number of allocated IDs exceeds 10,000. Even when the number of contending UEs in a RACH becomes larger, we can still find a unique TC-RNTI assignment as the number of assigned TC-RNTIs is fairly large.

### C. Overall Spectrum Utilization

So far we only evaluate the overhead saving for a single RACH. We finally check how such overhead reduction improves the overall spectrum efficiency when the network scales up, e.g., in mMTC. In particular, to avoid collisions in RA procedure, the eNB can reserve more resources in a PDSCH to create more RACH opportunities and then uniformly distribute UEs among different RACHs. By doing this, the number of contending UEs in each RACH can be controlled to maintain an acceptable success probability. However, creating more RACH opportunities also increases the overhead and,

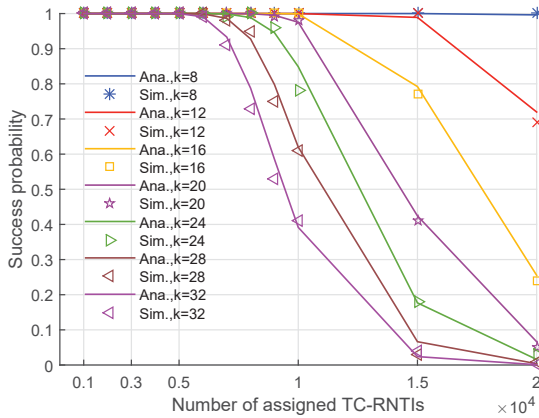


Fig. 9: Probability of searching unique TC-RNTI assignment

hence, reduces spectrum utilization for data communications. We then check what is the percentage of resources allocated to the RAR messages when the system targets for an expected success probability of 95%, 90% and 80%, respectively. We believe those settings should be reasonable for a mMTC scenario as it may not be efficient to distribute massive UEs to an extremely large number of RACHs, leading to low spectrum utilization.

Fig. 10 illustrates the percentage of resources occupied by the RAR messages in a 1.4MHz LTE PDSCH (i.e., 6 RBs in each sub-frame). The figure shows that the overall overhead increases significantly as the number of UEs grows, possibly leaving only 10% of the spectrum resources available for data communications. Our hybrid scheme can reduce the overhead by around 50%. More importantly, the overhead of the conventional scheme is almost the same even when the expected success probability decreases. This is because, even though the number of required RACHs decreases for a smaller targeting success probability, the number of UEs per RACH instead increases, leading to a longer RAR message per RACH. By contrast, the cost of our RAR message becomes smaller when the number of UEs per RACH increases. As a result, the overall overhead decreases if we lower the targeting success probability, i.e., reducing the number of RACHs. In other words, our scheme provides a chance to balance the trade-off between the random access success probability and the overhead.

## V. CONCLUSION

In this work, we propose an one-to-many ID-free random access response mechanism based on the concept of prime factorization. Instead of one by one announcing the identifiers of the detected preambles, an eNB in our design only needs to broadcast a single *hint* or bitmap message without specifically indicating the UE identifiers. Contending UEs then leverage factorization to parse their required information. We show via simulations that the proposed one-to-many response reduces

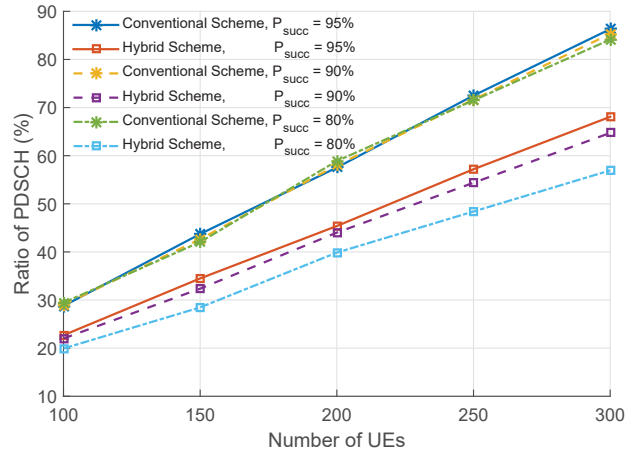


Fig. 10: Ratio of PDSCH occupied by RAR message

the signaling overhead significantly, especially when the network scales up. Such overhead saving can be translated to around 50% increases in the available bandwidth for data communications.

## REFERENCES

- [1] R.-G. Cheng, C.-H. Wei, S.-L. Tsao, and F.-C. Ren, "RACH collision probability for machine-type communications," in *Proc. IEEE VTC Spring*, 2012.
- [2] 3GPP R2-102296, *RACH Intensity of Time Controlled Devices*, 3GPP, April 2010, Vodafone, RAN2#69bis.
- [3] D. T. Wiriadmadja and K. W. Choi, "Hybrid random access and data transmission protocol for machine-to-machine communications in cellular networks," *IEEE Trans. on Wirel. Communi.*, vol. 14, no. 1, pp. 33–46, 2015.
- [4] S. Cherkaoui, I. Keskes, H. Rivano, and R. Stanica, "LTE-A random access channel capacity evaluation for M2M communications," in *Proc. IEEE Wireless Days*, 2016.
- [5] C. M. Chou, C. Y. Huang, and C.-Y. Chiu, "Loading prediction and barring controls for machine type communication," in *Proc. IEEE ICC*, 2013.
- [6] M. Tavana, V. Shah-Mansouri, and V. WS, "Congestion control for bursty M2M traffic in LTE networks," in *Proc. IEEE ICC*, 2015.
- [7] T.-M. Lin, C.-H. Lee, J.-P. Cheng, and W.-T. Chen, "PRADA: prioritized random access with dynamic access barring for MTC in 3GPP LTE-A networks," *IEEE Trans. on Veh. Technol.*, no. 5, pp. 2467–2472, 2014.
- [8] Z. Wang and V. WS, "Optimal access class barring for stationary machine type communication devices with timing advance information," *IEEE Trans. on Wirel. Communi.*, vol. 14, no. 10, pp. 5374–5387, 2015.
- [9] C. W. Chang, Y. H. Lin, Y. Ren, and J. C. Chen, "Congestion control for machine-type communications in LTE-A networks," in *Proc. IEEE GLOBECOM*, 2016.
- [10] 3GPP TS 36.141 V13.2.0, *Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) conformance testing (Release 13)*, Std., Jan. 2016.
- [11] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.