Performance Evaluation of Radio Access Network Overloading from Machine Type Communications in LTE-A Networks

Ming-Yuan Cheng¹, Guan-Yu Lin¹, Hung-Yu Wei¹ and Chia-Chun Hsu²

¹Graduate Institute of Electrical Engineering, National Taiwan University, Taipei, Taiwan.
²MediaTek Inc., Hsinchu, Taiwan

Abstract—Featuring massive number of devices, Machine Type Communications (MTC) poses great challenges for radio access network (RAN) with its overloading problem that has been aggressively addressed by 3GPP as an essential working item. In this article, we first introduce random access procedure in LTE-A. Then we address the issue of RACH overload coming from massive number of MTC devices. To resolve RAN overload, several easy-to-implement RAN level solutions are proposed. Based on simulation of MTC in Long Term Evolution- Advanced (LTE-A) architecture, we compare the overload resolution capability of different overload resolution mechanisms and conclude with observations for further solution design.

Index Terms—RAN overload resolution; machine type communication; massive access; LTE-A networks

I. INTRODUCTION

Machine-to-Machine (M2M) applications, also known as machine type communication (MTC) applications, refer to automated applications involving with communication among machines without the necessity of human intervention. With its wide range of applications as classified in [1], M2M has become a promising technology. Features widely deployed backbone network, cellular network is the best choice to provide ubiquitous M2M services. Yet the traditional cellular networks are originally designed and optimized for voice call and human-to-human (H2H) communications, and thereby fall far short of the requirement of M2M applications. As suggested in [2][3], M2M applications characterize a massive deployment, which generates a huge amount of data/signaling traffic, congesting the available radio resource, and thereby overloading the communication networks. Additionally, the problem of congestion not only occurs in the radio access part, but also in the core network part, which poses tremendous difficulties to the normal operations of the mobile network, penalizing both MTC and non-MTC devices. To resolve the severe congestion of MTC devices in radio access network (RAN) level, some congestion resolution mechanisms have been proposed [4–7]. However, up to now no exhaustive comparison of these mechanisms is given. To discover the difference between resolution mechanisms, we implement these methods into the simulation scenario as proposed in [8]. Further, we evaluate these approaches by different performance metrics, including access success probability, access delay, and radio resource usage.

The article is organized as the following. First, we will give a brief overview of random access procedure and the description of simulation parameters settings in Section II and III respectively. After that, several overload control methods are mentioned in section IV, and Section V shows the simulation results to evaluate the performance between different overload control methods. Section VI concludes this article.

II. SIGNALING FLOW OVER CONTROL PLANE

The system architecture of LTE is illustrated in Fig. 1. Since RAN overload occurs in E-UTRAN part, in this section we will go through the signaling flow of the random access procedure in E-UTRAN, which is used by UE to establish the RRC connection. The overall signaling flow of RRC connection is shown in Fig. 2 [9]. RACH procedure [10][11], as the random access part of RRC connection, consists of the following four steps:

1) Random-access preamble transmission: The first step consists of transmission of a random-access preamble, allowing the eNodeB to estimate the transmission timing of the terminal. The network broadcasts information to all terminals...
in which time-frequency resources random-access preamble transmission is allowed (i.e., the PRACH resources, in SIB-2). As part of the first step of the random-access procedure, the terminal selects one preamble to transmit on the PRACH.

2) Random-access response: The second step consists of the network transmitting a timing advance command to adjust the terminal transmit timing, based on the timing estimated in the previous step. Besides uplink synchronization, the second step also assigns uplink resources to the terminal to be used in the third step in the random access procedure.

3) RRC Connection Request (Msg3): The third step consists of the transmission of the mobile-terminal identity to the network using the UL-SCH similar to normal scheduled data. The exact content of this signaling depends on the state of the terminal, in particular whether it is previously known to the network.

4) RRC Connection Setup (Msg4): The fourth and also the final step consists of the transmission of a contention-resolution message from the network to the terminal on the DL-SCH. This step also resolves the contention originating from the multiple terminals attempting to access the systems that share the same random-access resource.

Since RAN overload mainly occurs in RACH procedure, the remaining steps of RRC connection is not mentioned here.

III. RAN OVERLOAD CONTROL METHOD

As described in the previous sections, all of the UEs must follow the signaling flow of RACH procedure in control plane. Here, we will go through the behaviors of UE as in Fig. 3 and their corresponding parameters [9].

A. UE Behaviors

When UE is ready to transmit preamble, it will check whether the current time slot is random access slot (in ten milliseconds, UE has two opportunity to send preamble). Otherwise, UE will wait until the next random access slot is coming. After sending the preamble, the UE will increase the number of transmission times by one, and start the random access response (RAR) timer, and then wait for RAR. Once receiving the random access response in the RAR window, UE will process TA (Timing Alignment), UL grant and C-RNTI and prepare for send RRC Connection Request (i.e., Msg3). On the other hand, if UE cannot receive random access response in RAR window (i.e., eNB fails to detect the preamble from the UE), UE will check whether its number of preamble transmission times is smaller than the maximum number of preamble transmission. If yes, UE will randomly choose a time slot based on Backoff Indicator and prepare for the next preamble transmission. If not, UE shall stop performing RACH procedure and indicate a random access problem to the upper layers.

B. Simulation assumption and parameters

In our simulation, arrival time of M2M devices follow two types of distribution, uniform and beta distribution in 60 seconds and 10 seconds respectively. In each cell, the number of available preamble is 54 (total 64 preamble sequence minus 10 dedicated ones). All parameters in Table I follow the settings in [8]. Totally, the resource in PDCCH (Physical Downlink Control Channel) is 16 CCEs (Control Channel Elements). We assume 8 CCEs are located at common search space, which is used for carrying the common control information including paging information, system information and random access procedures. The common search space is monitored by all UEs in a cell. The other 8 CCEs are located at UE-specific search space in our simulation assumption. In each time slot, eNodeB can only contains one RAR (which will consume 8 CCEs at common search space in PDCCH), it only allows up to 3 UEs. Paging message will consume 4CCEs in common search space. Msg4 will consume 4 CCEs,
which is located at UE-specific search space in PDCCH for one UE. If the CCEs in common search space are not used by RAR, the eNodeB can use the resource to send Msg4. The preamble detection probability depends on the preamble transmission power. UE will increase its transmission power in each retransmission to increase the detection probability. For preamble transmission, [8] assumes that if more than two UEs choose the same preamble sequence, these UEs cannot receive RAR due to preamble collision. In other words, capture effect is not considered here. The parameter, mac-ContentionResolutionTimer, is the window size for Msg3.

ContentionResolutionTimer, is the window size for Msg3.

TABLE I

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of MTC devices</td>
<td>5000, 10000, 30000</td>
</tr>
<tr>
<td>MTC devices arrival distribution</td>
<td>Uniform distribution over 60s, Beta distribution over 10s</td>
</tr>
<tr>
<td>Cell bandwidth</td>
<td>5MHz</td>
</tr>
<tr>
<td>PRACH Configuration</td>
<td>6</td>
</tr>
<tr>
<td>Total number of preambles</td>
<td>54</td>
</tr>
<tr>
<td>Maximum number of preamble transmission</td>
<td>10</td>
</tr>
<tr>
<td>Number of UL grants per RAR</td>
<td>3</td>
</tr>
<tr>
<td>Number of CCEs allocated for PDCCH</td>
<td>16</td>
</tr>
<tr>
<td>Number of CCEs per PDCCH</td>
<td>4</td>
</tr>
<tr>
<td>Preamble detection probability (in case of no collision)</td>
<td>1 - ( \frac{i}{T} ), where ( i ) indicates the ( i )-th preamble transmission</td>
</tr>
<tr>
<td>ra-ResponseWindowSize</td>
<td>5 subframes (5ms)</td>
</tr>
<tr>
<td>mac-ContentionResolutionTimer</td>
<td>48 subframes (48ms)</td>
</tr>
<tr>
<td>Backoff Indicator</td>
<td>30ms</td>
</tr>
<tr>
<td>HARQ retransmission probability for Msg3 and Msg4 (non-adaptive HARQ)</td>
<td>10%</td>
</tr>
<tr>
<td>Maximum number of HARQ TX for Msg3 and Msg4 (non-adaptive HARQ)</td>
<td>5</td>
</tr>
<tr>
<td>Maximum number of UE in Paging List</td>
<td>16</td>
</tr>
</tbody>
</table>

We assume that all MTC devices activate during \( t = 0 \) and \( t = T \). The beta distribution applied in Table I is then defined as:

\[
p(t) = \frac{t^{\alpha-1}(T-t)^{\beta-1}}{T^{\alpha+\beta-1}\text{Beta}(\alpha, \beta)}, \quad \alpha > 0, \beta > 0
\]

where \( p(t) \) follows the beta distribution, \( \int_0^T p(t)dt = 1 \). The access intensity in the \( i \)-th access slot is then expressed as \( N \int_{t_{i-1}}^{t_i} p(t)dt \), where \( t_i \) is the time of the \( i \)-th access opportunity, and \( N \) is the total number of MTC devices in the cell. In this study, \( \alpha = 3 \) and \( \beta = 4 \) are applied.

In our simulation, we additionally include voice calls traffic, which is Poisson arrival distribution with parameter 7 calls per second. The maximum number of UE in paging list is the maximum UE in a paging message. Note that all the introduced methods for RAN overload resolution only apply to M2M devices, and the original voice calls always follow the default parameters.

IV. METHOD DESCRIPTION

The basic concept of contention resolution is to spread massive random access load on a relatively long period of time according to delay requirement of MTC devices. Based on this concept, some RAN level solutions have been proposed in [12, 13], but no exhaustive comparison is provided to evaluate/compare their performance. In the following subsection, we individually introduce how these solutions work.

A. Push based methods

Each M2M device is configured to transfer its readings if certain condition(s) are satisfied. Data center of individual utility company configures and controls the reporting policy for their M2M devices. Reading report is initiated and performed at individual device governed by the configured policy. Therefore, the push based approach is a distributed control model. We introduce some push based methods as follows.

1) Backoff Indicator adjustment: As suggested by [4], large M2M backoff indicator value could allocate massive RACH preamble transmissions to a long period of time, so that preamble collisions can be reduced but the delay may increase. Thus, in this approach we apply different BI values to investigate the tradeoff.

2) P-persistent approach: In this approach, each MTC device is given a predefined value \( p \). Each time when a MTC device attempts to start the random access procedure, it will randomly generate a random number between 0 and 1. If the generated number is smaller than the value of device \( P \)-persistent, then it can start the RACH procedure and transmit RACH preamble. Otherwise, the device needs to back off and waits until another interval to try again. The approach is suggested by [5] for contention resolution.

3) Maximum number of preamble transmission adjustment: In this approach, we adjust the maximum number of preamble transmission. We aim to investigate whether MTC devices can increase the success rate of RACH procedure by configuring larger maximum number of preamble transmission.

4) Wait timer adjustment: For this alternative, we add a new parameter called wait timer. (Note that the wait timer here is different from extended wait timer defined in RRC level.) Wait timer here means the additional period M2M devices have to wait after they fail to receive RAR, Msg3 and Msg4. In this period, UE cannot send anything to eNB.

5) Access Class Barring: This approach classifies MTC devices based on their access classes (AC 0 ∼ 9). To be specific, M2M devices in different access class apply different Backoff Indicators, persistent value, or wait timer [5–7]. Note that since currently no approaches to differentiate MTC devices is standardized, in this work we assume all MTC devices are homogeneous and always apply the same parameters.

B. Pull based methods(Paging)

In pull based approach, RACH procedure is triggered by the eNodeB rather than the UE. In paging method, all M2M devices in idle mode listen to the paging message. Only when IDs of M2M devices are included in the paging message can
they initialize random access procedure. In spite of the elimination of preamble collision, resource efficiency of RACH is limited by the maximal number of device ID that can be included in a paging message, which is set as 16 in our simulation. Besides, paging method will cost of extra paging resource as tradeoff.

V. SIMULATION RESULT

As shown in [9], all the initial results of the uniform distribution cases show 100% success probability. As for beta distribution, fewer number of M2M devices still has an 100% success rate. Thus, we focus on the scenario with beta distributed 30,000 M2M devices arrival. Further, since up to now related simulation scenarios about prioritized congestion resolution mechanisms are not yet agreed by 3GPP, simulation results about access class barring are not presented in this section.

A. Steps by Steps analysis

We hereby study beta distribution with 30,000 M2M devices. In this subsection, we analyze the UL and DL part respectively.

1) Random-access preamble transmission analysis: In RACH preamble transmission, we divide the preamble usage into three states: no UE use this preamble (empty), only one UE use this preamble, (success usage), and more than two UEs use the same preamble (collided preamble.) Thus, we can define the following three metrics:

a) Success probability for preamble: Success probability is defined as the number of preamble successfully received at eNodeB divided by the total request number of RACH preamble in that time slot.

b) Preamble resource utilization: Preamble resource utilization is the number of success usage divided by total number of preambles.

c) Collision probability of preamble: The preamble collision probability is defined as the number of collided preamble divided by the total number of preambles.

Without applying any RACH overload resolution method, the probability that $S$ among $N$ simultaneously sent preambles are received successfully (i.e. without preamble collision) can be calculated as

$$ P(S|N, R) = \frac{1}{R^N} \binom{N}{S} \binom{R-S}{S} \frac{S^S_i}{R^i} \sum_{i=0}^{\min(N-S, R-S)} \frac{(N-S)^i}{i!} (-1)^i (R-S-i)^{N-S-i} $$

(2)

Fig. 4(a) shows that preamble resource utilization reaches to the peak when the number of request preamble slightly exceeds the total number of RACH resources. When the number of request preamble decreases from the number of peak position, resource utilization drops since most preamble sequences are not used. On the other hand, when the number of request preamble raises from the peak, preamble sequences would suffer from high RACH collision probability, leading to low resource utilization. In Fig.4(b) we observe that the collision probability is an increasing function of the number of sent preambles, but success probability is not a decreasing function of the number of sent preambles. This is because when the number of request preamble is low, some devices may still fail during the first or second preamble transmission because they apply transmission power that is too low to be detected by the eNB. On the other hand, when the number of request preamble is high, failure preamble transmission is dominated by RACH collision rather than the small transmission power. Thus, the success probability declines and the collision probability rises when the access number increases. The preamble analysis is only considered with the number of preamble in one time slot. For different methods, their location will at the different parts in Fig. 4. For paging method, the maximum number of preamble request in one time slot is 16 due to the limitation of paging list. Thus, the distribution of the points will only locate at the part where the number of request preamble is low. For original scheme, the distribution might locate at where the number of request preamble is high. The methods we propose are to find a suitable parameter to reach the optimal point.

2) Downlink resource consumption analysis: As shown in Table II, we calculate the total CCE consumption of different methods, and we can find that paging method consumes the most CCEs among all the methods due to paging message. Also, all the push based methods consume more PDCCH resource as success rate increases. Higher success probability will lead to more CCEs consumption.
B. Overall analysis

In this subsection, we define some metrics to evaluate the performance of random access procedure.

1) Average Access Success Probability: The probability for a UE/MTC device to successfully complete the random access procedure within the maximum number of preamble transmissions.

2) Average Access Delay: The time difference between the packet arrival and the completion of the random access procedure, for the successfully accessed MTC devices.

In the original scenario, the success rate of both H2H and M2M are very low without applying any contention resolution method. In Table II, we compare the success probability, average delay, and total PDCCH consumption corresponding to several RAN overload resolution methods. Note that since both MTC devices and H2H users apply the same RACH procedure, the success probability and collision probability for MTC and H2H traffic are the same. From Table II, we can observe that the success rate drops sharply from the original one if we increase the maximum transmission number. This is because when RACH resources are overloaded, higher retry times could not increase the overall success probability. Instead, frequent retransmission produces more collided preambles, resulting in a much severer RAN overload. Then we draw a comparison between backoff indicator, persistent value, and wait timer methods. Table II shows that all of the three methods can achieve a much higher success rate than the original scenario by appropriate parameter tuning, with the tradeoff of increasing delay. Finally, we investigated the performance of the paging method. By pull based approach, RACH resource congestion is avoided and the success rate can reach to 100% because RACH preamble traffic is reshaped. However, the trade-off of the perfect success rate is the long delay and high consumption of downlink resource. Two main reasons lead to a long delay: First, the number of M2M devices that can be paged is limited by the setting of the paging occasion and the capacity of paging message. Thus, RACH resource utilization may not be efficient. Second, a M2M device may not be prepared to send preambles when it is paged because the eNB does not perceive whether a M2M device has data to send. As a result, an unprepared MTC device may have to wait for a long period of time for re-page because it takes longer for massive devices to be paged once. These two reasons are essential to the implementation of pull-based contention resolution methods.

VI. CONCLUSION

This article addresses the RAN overload problem of MTC in 3GPP LTE, along with the performance of several easy-to-implment RAN overload resolution mechanisms. Based on simulation settings agreed in [9], we carry out a comprehensive performance comparison between these mechanisms, which is not yet done by previous works. Note that there are still some extended issues worthy of further study, such as core network congestion and signaling resource shortage resulting from MTC.

REFERENCES