Group Paging Optimization For Machine-Type-Communications

Osama Arouk*, Adlen Ksentini*, and Tarik Taleb†
* IRISA, University of Rennes 1
Campus Beaulieu, 35042 Rennes, France
Email: firstname.lastname@irisa.fr
† Communications and Networking Department, Aalto University, Finland - Email: talebtarik@ieee.org

***Abstract*— Machine-Type-Communication (MTC) is a promising service of the envisioned 5G mobile networks. However, deploying a massive number of MTC devices in these networks remains a challenge due to the overload that may appear at the Radio Access Network (RAN), hence degrading the Quality of Services (QoS) for both MTC and Non-MTC devices. One of the methods used to address the congestion's problem in RAN is Group Paging (GP), wherein a single message is used to activate a group of devices. Whilst the GP method has several advantages, its performance quickly decreases when the number of MTC devices increases. In this paper, we devise a new method, namely Traffic Scattering For Group Paging (TSFGP) to improve the performance of the GP method for massive deployment of MTC devices. Numerical results demonstrate that TSFGP highly improves the performance of GP in terms of several performance metrics, such as success probability, collision probability, and access delay.**

I. Introduction

Machine-Type-Communications (MTC), or Machine-to-Machine (M2M) communications, usually refer to systems whereby a machine communicates with another machine without any human intervention. In general, M2M technology supports a large number of applications, such as Smart City, Smart Grid, Intelligent Transport System (ITS), and health care. In the future, we may see M2M technology in nearly every aspect of our daily life. Because of the vast diversity of M2M applications, a potential number of MTC devices (50 billion M2M) are envisioned by 2020 [1]. However, deploying a massive number of MTC devices in the current cellular mobile networks, which have been initially designed and optimized for Human-to-Human (H2H) communication, will generate a huge amount of data and control traffic, causing congestion and system overload over the whole network, i.e. at both the Radio Access Network (RAN) part and the Core Network (CN) part. In addition, this problem will impact not only MTC devices but also Non-MTC connections.

Mitigating congestion and system overload has attracted a lot of attention as it represents one of the big challenges facing the deployment of MTC in the current cellular mobile networks. For example, RAN improvements for supporting a large number of MTC devices is one of the principle study items for the 3^{rd} Generation Partnership Project (3GPP) [2]. This work item has defined many RAN overload control methods to cope with the congestion's problem when deploying a large number of MTC devices. Based on whether the network or the terminal initiates the Random Access Channel (RACH) procedure, the RAN congestion control methods, approved by 3GPP, can be classified into two categories. The first one is Push based approach, wherein the terminal itself initiates the RACH procedure. Note that such approach can be also viewed as a decentralized approach. The second category corresponds to solutions where the network initiates the RACH procedure (Pull), which could be also seen as a centralized approach.

In the category of Push based schemes, there are many different methods adopting different approaches, such as separation of RACH resources, dynamic allocation of RACH resources, MTC specific backoff, and Access Class Barring (ACB). In methods adopting separation of RACH resources, the network separates the RACH resources between MTC and Non-MTC (i.e. H2H devices) not to impact the Non-MTC channels, even when massive MTC devices are deployed. However, when the network needs to gather information from a terminal, it is better to specify the resources to be used by the terminal instead of letting the MTC device do the RACH procedure by itself, i.e. using a Pull based approach. One of the Pull based approaches is paging, whereby the network sends a paging message to the terminal addressed by its individual ID. But, when a high number of MTC devices needs to be paged, sending individual paging messages becomes very costly. A solution to this problem is to use the Group Paging (GP) method, whereby a single GP message is sent to all terminals addressed by the Group ID (GID). In spite of the improvement achieved by the GP method, its performance drops when the number of paged MTC devices increases. An improvement to the GP method has been introduced in [3], whereby the aim is to control the MTC access by using a strict slot assignment schedule. However, this work focuses on the MTC devices in the RRC (Radio Resource Control) Connected mode, and does not address the MTC devices in the RRC Idle mode. In this paper, a novel approach is devised to further improve the performance of the GP method, regardless of the operating mode of the terminal. The key idea behind the proposed scheme is to scatter the paging operation of the MTC devices over a GP interval instead of letting all devices starting the RACH procedure at nearly the same time (i.e. upon receiving the GP message). It should be noted that, besides the pull and push based approaches, there are different approaches, such as the work in [4], and also the work in [5] wherein a reduction in the size of the sent messages is done based on the common information. The reduction of the size of messages is achieved to about 70%, thus alleviating the congestion not only in the RAN part, but also in the CN part [6].

The remainder of the paper is organized as follows. Section II introduces some research work related to the Random Access Channel (RACH) procedure and the RAN congestion control mechanisms. Analysis of the GP method and the analytical model of the proposed Traffic Scattering For Group Paging (TSFGP) scheme are introduced in Section III. The performance evaluation of TSFGP is presented in Section IV. Finally, concluding remarks are made in Section V.

II. Related Work

In order to better understand the RAN congestion control solutions, we start by introducing some background about the Random Access Channel (RACH) procedure in Long Term Evolution (LTE) and the LTE-Advanced (LTE-A) networks.

A. RACH procedure

Generally speaking, a User Equipment (UE) in LTE and LTE-A networks can be in one of two operating modes: RRC_IDLE or $RRC\_CONNECTED$ mode. In the RRC_IDLE mode, the UE can neither receive nor transmit data, except receiving some broadcast information. On the other hand, a UE in the $RRC\_CONNECTED$ mode can send and receive data. If a terminal in the idle mode needs to transmit some information, or there is information to be received from the network, the first procedure that is initiated is the Random Access Channel (RACH) procedure. This procedure consists of four steps:

1) *Random Access Preamble Transmission (Msg1):* The first step consists of transmitting a preamble, where the terminal randomly chooses one out of the available preambles and sends it within the dedicated Random Access (RA) slot. As the preamble is randomly chosen, it is possible to encounter the case when multiple terminals choose the same preamble, resulting in a collision. In this case, all terminals having chosen the same preamble will retry the RACH procedure.
2) *Random Access Response (Msg2):* After transmitting the preamble, the terminal monitors the Physical Downlink Control Channel (PDCCH) during a Random Access Response (RAR) window for the response message. This message contains many parameters, such as UpLink (UL) grant and Temporary Cell - Radio Network Temporary Identifier (TC-RNTI). UL grant refers to the Uplink resources to be used in the next step, while TC-RNTI is the temporary ID of the terminal within this cell, and it may be promoted to C-RNTI if the UE does not yet have one. However, if UE does not receive a response message during the RAR window, it deems that there was a collision, and subsequently retries the RACH procedure.
3) *RRC Connection Request (Msg3):* After receiving and processing $Msg2$, the UE will proceed to the next step in order to request RRC connection from the network. The terminal also sends, within this message, its ID.
4) *RRC Connection Setup (Msg4):* This step is a response to the precedent one, informing the terminal that RRC connection has been setup.

B. RAN Congestion Control

As mentioned before, there are two classes of the RAN congestion control methods: Push based and Pull based approaches [7]. In the Push based category, a UE initiates by itself the RACH procedure. Backoff method is a good example of this class. By giving a longer backoff time for MTC devices, compared to that for Non-MTC devices, the delivery of the MTC traffic will be scattered over a long time interval, which ensures enough resources for Non-MTC devices to access the channel. If it is well adjusted, the backoff method can guarantee the network availability for Non-MTC devices. Another example of this class is the Access Class Barring (ACB) method [8], where a separate Access class can be assigned for MTC devices. This method allows the network to control the access of MTC devices independently, and thus avoiding any impact of MTC traffic on the QoS of Non-MTC devices [9]. Another improvement of ACB method is the work in [10], wherein a traffic prediction is used in order to adjust the parameters of ACB method.

In the Pull based category, the network initiates the RACH procedure. Paging is a good example of this class, where the network sends a paging message to the considered terminal addressed by its individual identifier (ID). However, the current paging mechanism can page at maximum 16 terminals. Thus, the paging method would become costly if is applied to page a large number of MTC devices. For example, to page 30000 MTC devices, 1875 paging messages are needed, which will take about 9 seconds knowing that there are two paging messages every $10\,ms$. As a remedy to this limitation, MTC devices may be grouped into groups, each identified by a unique ID, i.e. Group ID (GID) [11]. Accordingly, all MTC devices belonging to the same group can be paged by only one paging message.

Given that Push based methods are considered as decentralized control schemes, the resource utilization is lower and varies over time. Further, it is difficult to regulate the network load [12]. However, the advantage of these methods is the reduction of the signaling load as there is no need for individual paging messages. On the other hand, the centralized feature of the Pull based approaches allows improving the network utilization, regulating easily the network load. However, their disadvantage consists in the fact that signaling load is slightly higher due to numerous paging messages. Despite this disadvantage, it is more preferable to use Pull based approaches as a central network element (e.g., eNB) can easily control the network load at any time, ensuring a good QoS.

III. Traffic Scattering For Group Paging (TSFGP)

A. System Model:

In this study, we consider a group of M MTC devices uniformly distributed over N cells. Therefore, each cell will host, on average, M/N MTC devices. It is assumed that each eNB, i.e. base station, reserves R Random Access (RA) resources for the contention access. The RA resources are defined in terms of RA Opportunities (RAOs), which are equal to the number of frequency bands in the RA slot multiplied by the number of RA preambles. However, in our model, we suppose that there will be just one frequency band, and therefore the number of resources is equal to the number of preambles. After transmitting the paging message, addressed by the Group ID (GID), the MTC devices will start the RACH procedure with a probability p_{act}, instead of leaving them to start all at once. The objective behind using the probability p_{act} is to ensure that MTC devices starting the RACH procedure, at each RA slot, have access to the channel with a success probability that matches the network capacity. In other words, the objective is that the number of successful MTC devices at each RA slots is equal, at most, to the number of MTCs that can be acknowledged during the RAR window W_{RAR}. We recall that the number of RAR responses during a RAR window is equal to:

$$N_{ACK} = N_{RAR} \cdot W_{RAR} \quad (1)$$

where N_{RAR} is the maximum number of RARs per a response message.

B. Analysis of ordinary Group Paging:

The authors in [13] introduced an interesting analytical model for the group paging method, which will constitute the basis of our analysis. After receiving a group paging message, all members of the group will transmit their preambles at the first available RA slot. The number of successful and collided MTC devices after the first transmission of the preamble is:

$$M_{1,s} = \begin{cases} Me^{-\frac{M}{R}}p_1 & ;\ \text{if } Me^{-\frac{M}{R}}p_1 \leq N_{ACK} \\ N_{ACK} & ;\ \text{otherwise} \end{cases} \quad (2)$$

$$M_{1,c} = M - M_{1,s} \quad (3)$$

where M denotes the number of MTC devices, and p_1 is the preamble detection probability for the first preamble transmission. After finishing the RAR window, the collided MTC devices at the first RA slot, i.e. $M_{1,c}$, will do backoff and then retransmit the preamble when their backoff timers expire. As the backoff time follows a uniform distribution, the collided MTC devices will be uniformly distributed over the next slots during the backoff interval W_{BO}. Generally, the number of MTC devices transmitting the preamble for the second time in a RA slot is equal to the part of slots from the backoff interval that falls before this RA slot multiplied by the number of collided MTC devices. However, the first RA slot a that falls within the backoff window, as illustrated in Fig. 1, will be at:

$$x_a(i) = i + \left\lceil \frac{T_{RAR} + W_{RAR}}{T_{RA\_REP}} \right\rceil \quad (4)$$

where $x_a(i)$ is the order of the first RA slot within the

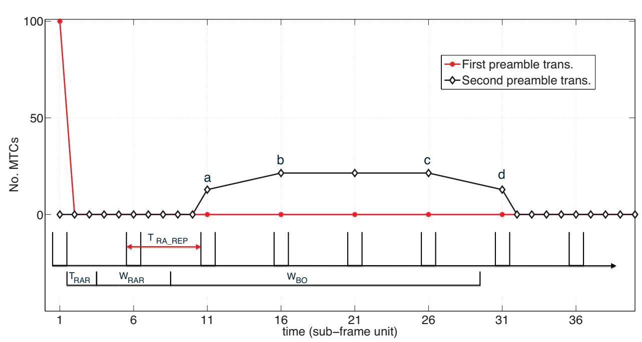

Fig. 1: Number of MTC devices at each RA slot for the first and second preamble transmissions for the considered parameters.

backoff interval W_{BO}, related to the preamble transmission at the RA slot i, T_{RAR} is the processing delay at the eNB, and T_{RA_REP} is the interval between two consecutive RA slots. The proportion of MTC devices whose backoff timers expire and retransmit their preambles at the RA slot a is equal to the time of the RA slot a, in a sub-frame unit, minus the duration before the start of the RAR window (normalized by W_{BO}):

$$\alpha_a = \frac{\left\lceil \frac{T_{RAR}+W_{RAR}}{T_{RA\_REP}} \right\rceil T_{RA\_REP} - (T_{RAR} + W_{RAR})}{W_{BO}} \quad (5)$$

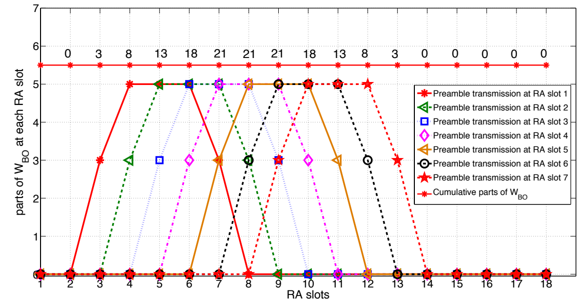

Fig. 2: The cumulative parts of W_{BO} for each RA slot for MTC devices transmitting their preambles for the second time, where $W_{BO} = 21$ and $T_{RA\_REP} = 5$.

Ragarding the RA slots from b to c, they will be at:

$$x_{bc}(i) = i + \left\lceil \frac{T_{RAR} + W_{RAR}}{T_{RA\_REP}} \right\rceil + k, \quad k = 1, 2, ..., K_{max} \quad (6)$$

where $K_{max} = \lfloor (W_{BO} - \alpha_a W_{BO})/T_{RA\_REP} \rfloor$, and the proportion of MTC devices that retransmit their preambles at these RA slots is equal to:

$$\alpha_{bc} = \frac{T_{RA\_REP}}{W_{BO}} \quad (7)$$

The rest of MTC devices will retransmit at the RA slot d. This RA slot will be just after the last one of the slots bc, i.e.:

$$x_d(i) = i + \left\lceil \frac{T_{RAR} + W_{RAR}}{T_{RA\_REP}} \right\rceil + K_{max} + 1 \quad (8)$$

and the proportion of MTC devices is:

$$\alpha_d = 1 - \alpha_a - \alpha_{bc} K_{max} \quad (9)$$

Therefore, the number of MTC devices retransmitting their preambles at the RA slots a and d are $M_{1,c}\alpha_a$ and $M_{1,c}\alpha_d$, respectively, while it is equal to $M_{1,c}\alpha_{bc}$ for the RA slots between b and c. Now, if we assume that, at each RA slot, the same number of MTC devices arrives, then each RA slot will generate the graphic form illustrated in Fig. 1, and therefore the number of collided MTC devices will be the sum of the contribution of each RA slot (Fig. 2). We see clearly from Fig. 2 that when the number of arrivals at each RA slot is the same, the system converges to a situation whereby the number of MTC devices retransmitting their preambles is constant. This means that the number of successful MTC devices at each RA slot will be constant too.

C. Analytical Model

The key idea behind our proposed TSFGP scheme is to scatter the access attempts of MTC devices to the network over a group paging interval instead of letting them start the RACH procedure all at the same time. In general, The number of MTC devices at each RA slot can be expressed as [13]:

$$M_i = \sum_{n=1}^{N_{PT_{max}}} M_i[n]$$

where $N_{PT_{max}}$ is the maximum number of preamble transmissions, and the number of successful MTCs is given by [13]:

$$M_{i,s}[n] = \begin{cases} M_i[n]e^{-\frac{M_i}{R}}p_n & ;\ \text{if } \eta_i \leq N_{ACK} \\ \frac{M_i[n]e^{-\frac{M_i}{R}}p_n}{\eta_i} N_{ACK} & ;\ \text{otherwise} \end{cases}$$

where $\eta_i = \sum_{n=1}^{N_{PT_{max}}} M_i[n] e^{-\frac{M_i}{R}} p_n$. As the network can send a maximum of N_{ACK} responses to MTC devices, we are interested in the case whereby the total number of successful MTCs at each RA slot is inferior or equal to N_{ACK}. In other words, we search for the case that $\eta_i \leq N_{ACK}$, and therefore

$$M_{i,s}[n] = M_i[n] e^{-\frac{M_i}{R}} p_n \tag{10}$$

Let M_1 denote the number of arrivals at each RA slot, which represents the value $M_i[1]$, and thus we have:

$$\begin{aligned} M_{i,S}[1] &= M_i[1] e^{-\frac{M_i}{R}} p_1 = M_1 e^{-\frac{M_i}{R}} p_1 \\ M_{i,C}[1] &= M_1 - M_{i,S}[1] = M_1(1 - e^{-\frac{M_i}{R}} p_1) \end{aligned}$$

However, we see clearly from Fig. 2, that when the number of successful MTC devices is stable, the cumulative parts of W_{BO} becomes equal to W_{BO}, and therefore the total MTC devices transmitting their preambles for the $(n+1)^{th}$ time, i.e. $M_i[n+1]$, is equal to the collided MTC devices transmitting their preambles for the $(n)^{th}$ time, i.e. $M_{i,C}[n]$, and thus $M_{i,C}[n] = M_i[n+1]$. For $n = 2$, we have:

$$\begin{aligned} & M_i[2] = M_{i,C}[1] \\ & M_{i,S}[2] = M_i[2] e^{-\frac{M_i}{R}} p_2 = M_1(1 - e^{-\frac{M_i}{R}} p_1) e^{-\frac{M_i}{R}} p_2 \\ & M_{i,C}[2] = M_i[2] - M_{i,S}[2] = M_1(1 - e^{-\frac{M_i}{R}} p_1)(1 - e^{-\frac{M_i}{R}} p_2) \\ & \qquad = M_1 \prod_{k=1}^{2} (1 - e^{-\frac{M_i}{R}} p_k) \end{aligned}$$

By induction, we can find that

$$\begin{aligned} & M_i[n] = M_{i,C}[n-1] \\ & M_{i,S}[n] = M_1 \prod_{k=1}^{n-1} (1 - e^{-\frac{M_i}{R}} p_k) e^{-\frac{M_i}{R}} p_n \\ & M_i[n+1] = M_{i,C}[n] = M_1 \prod_{k=1}^{n} (1 - e^{-\frac{M_i}{R}} p_k) \end{aligned}$$

or

$$M_i[n] = M_{i,C}[n-1] = M_1 \prod_{k=1}^{n-1} (1 - e^{-\frac{M_i}{R}} p_k) \tag{11}$$

Therefore, the total number of MTC devices at each RA slot, in the stable state, is equal to:

$$M_i = \sum_{n=1}^{N_{PT_{max}}} M_i[n] = M_1 \sum_{n=1}^{N_{PT_{max}}} \prod_{k=1}^{n-1} (1 - e^{-\frac{M_i}{R}} p_k) \tag{12}$$

However, as demonstrated in the Appendix, Equation (12) can be written as follows:

$$M_i = M_1 \sum_{m=0}^{N_{PT_{max}-1}} \alpha_m e^{-\frac{m M_i}{R}} \tag{13}$$

where α_m is given the Appendix. It is worth noting that the exponential function e^x can be approximated by the following equation [14]:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + ... \tag{14}$$

Applying this approximation to Equation 13 and reformulating it, we obtain:

$$\left(\sum_{m=0}^{N_{PT_{max}-1}} m^2 \alpha_m \right) M_i^2 - 2 \left(\frac{R^2}{M_1} + R \sum_{m=0}^{N_{PT_{max}-1}} m \alpha_m \right) M_i + 2R^2 \sum_{m=0}^{N_{PT_{max}-1}} \alpha_m = 0 \tag{15}$$

which is a second order equation for M_i that can be easily solved. After obtaining the value of M_i by Equation 15, we calculate the number of successful MTC devices by Equation 10. Fig. 3 shows the results of Equation 15 when varying the value M_1. From this figure, we see that there is a little difference between the actual and approximated values regarding the total number of arrivals. But, this difference is negligible when focusing on the number of successful MTC devices. This difference can be explained by the fact that we use the second order approximation for the value e^x instead of its actual value. It is worth noting that the relationship between the number of new arrivals and the successful MTC devices in the stable state is linear when $M_1 \leq N_{ACK}$. Based on these results, we propose to activate at each RA slot a maximum of N_{ACK} MTC devices, instead of letting all MTC devices activate all at the same time. In general, if there are (M/N) MTC devices with network access attempts scattered over I_{max} RA slots, then there are, on average, $(M/N)/I_{max}$ MTC devices in each RA slot, where I_{max} is the number of RA slots within the group paging interval. It is equal to [13]:

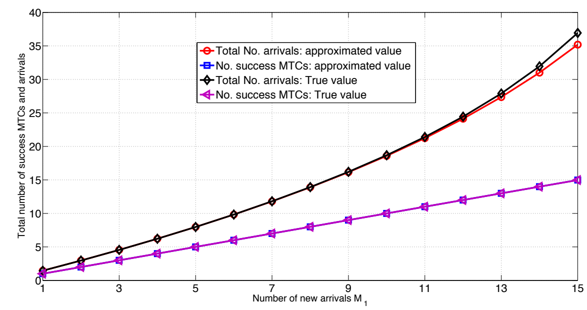

Fig. 3: The actual and approximated values of the number of successful MTCs and the total number of arrivals in the stable state as a function of the number of new arrivals M_1, where $N\_ACK = 15$.

$$I_{max} = 1 + (N_{PT_{max}} - 1) \left\lceil \frac{T_{RAR} + W_{RAR} + T_{BO}}{T_{RA\_REP}} \right\rceil$$

In order to make sure that there will be M_1 new arrivals at each RA slot, we scatter the access attempts of MTC devices over virtual RA slots $I_{V_{max}}$, where $I_{V_{max}} = \left\lceil \frac{(M/N)}{M_1} \right\rceil$. Now, each member of the group randomly generates an integer value from within the range $[1, I_{V_{max}}]$, where this value represents the RA slot in which the MTC device will start the RACH procedure. If the generated value falls within the interval $[1, I_{max}]$, then this device will start the RACH procedure within this RA slot, otherwise it turns into the idle state. The objective of this procedure is to determine directly whether a MTC device will proceed the RACH procedure, instead of letting it to check that at each RA slot, e.g. similar to the p-persistent mechanism. It should be noted that the performance of our proposed TSFGP scheme can be further improved by increasing the group paging interval, i.e. increasing I_{max}. Note that increasing I_{max} for the GP method does not have an impact as all MTC devices start the RACH procedure at the first available RA slot and the number of preamble transmissions is fixed.

IV. Performance Evaluation

Computer simulations were carried out by a C++ simulator, in order to verify the performance of the proposed method TFSGP. The parameters of RACH procedure specified by Table 6.2.2.1 in [2], and also the control-plane latency analysis specified in Table $B.1.1.1-1$ in [15] are used in the simulations. Regarding the number of MTC devices, it will vary in the range $[10, 1000]$.

A. Performance Metrics

In order to evaluate the performance of the proposed scheme and compare it against that of the GP mechanism, we consider the following performance metrics: $i)$ the success, collision, and drop probabilities; $ii)$ the average access delay; and $iii)$ the average number of preamble transmissions. The success probability is defined as the probability that the RACH procedure has been successfully completed within the maximum number of preamble transmissions. It is normalized by the total number of MTC devices; i.e., devices in both active and idle modes. The collision probability is the ratio of the number of collided RAOs to the total number of reserved RAOs. For the sake of a fair comparison with the ordinary GP mechanism, in our simulation, the group paging interval used in our proposed scheme is set equal to that of the GP method. As there will be M_1 MTC devices to be activated at each RA slot, there will be a part of MTC devices that will remain in idle mode if $(M/N) > I_{max}M_1$. Therefore, the drop probability is equal to:

$$P_d = \begin{cases} \frac{(M/N)-I_{max}M_1}{(M/N)} & ;\ \text{if}\,(M/N) > I_{max}M_1 \\ 0 & ;\ \text{otherwise} \end{cases} \tag{16}$$

Regarding the average access delay, it is defined as the aggregate access delays experienced by all MTC devices that successfully completed the RACH procedure averaged by the total number of these successful MTC devices. The average number of preamble transmissions is equal to the total number of preamble transmissions of all MTC devices that successfully completed the RACH procedure divided by the total number of successful MTC devices.

B. Results

Fig. 4 plots the success, drop, and collision probabilities for the GP and TSFGP methods. Although there is a group of non-activated MTC devices in case of our proposed scheme, TSFGP outperforms the GP method. In fact, we observe that the success probability of the GP method decreases to about 20% when $(M/N) = 1000$, while it remains higher than 70% in case of TSFGP. Regarding the collision probability, two important observations are noticed. The first one is the large gain achieved by TSFGP in comparison to GP. The second one is that the collision probability of TSFGP increases as the number of MTC devices increases, and then remains stagnant when (M/N) exceeds a certain value. This is attributable to the fact that the drop probability starts increasing when (M/N) exceeds $I_{max}M_1 = 55 * 15 = 825$, as shown in Fig. 4. Accordingly, the collision probability remains constant after this point as the network does not permit the activation of further MTC devices (i.e., more than $I_{max}M_1$) during the GP interval. Fig. 5 shows the average access delay for both GP and TSFGP methods and that is for different numbers of MTC devices. It is apparent that TSFGP considerably decreases the average access delay compared to the GP method. Furthermore, we remark that the average delay experienced in TSFGP increases as the number of MTC devices (M/N) increases, and then decreases when (M/N) exceeds a certain value. The same behavior is also observed for the average number of preamble transmissions shown in Fig. 6. This behavior is particularly due to the fact that the MTC devices activated at the first RA slot have the ability to transmit the preambles $N_{PT_{max}}$ times, while the ones activated in the last RA slot transmit the preamble just once. Fig. 6 shows the average preamble transmissions. From the figure, we observe that when $M/N = 650$, the average preamble transmissions is about 2.4. If we know that the average time to transmit the preamble just once is about $1 + (T_{RAR} + W_{RAR} + W_{BO}/2)/T_{RA\_REP} = 1 + (2 + 5 + 21/2)/5 = 4.5$ RA slots, then to retransmit the preamble 2.4 times, this device should be activated before $(4.5 \times 2.4 \simeq 11)$ RA slots, i.e. before about 11 RA slots from the end of the GP interval. For $M/N = 650$ MTC devices, the last device activated will be at the $\lceil 650/15 \rceil = 44^{th}$ RA slot, i.e. before 11 RA slots from the end of the GP interval if we know that $I_{max} = 55$. As a result, when M/N exceeds this value, the average access delay and the average preamble transmissions will decrease, as the MTC devices activated in

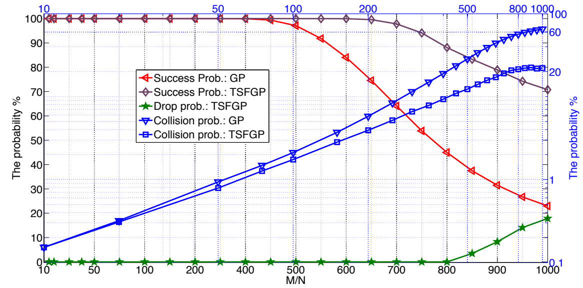

Fig. 4: The success, drop, and collision probabilities in case of both GP and TSFGP.

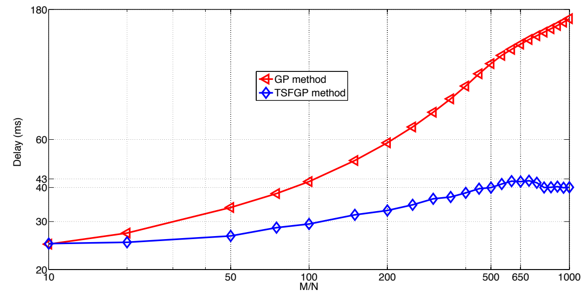

Fig. 5: The average access delay in case of both GP and TSFGP.

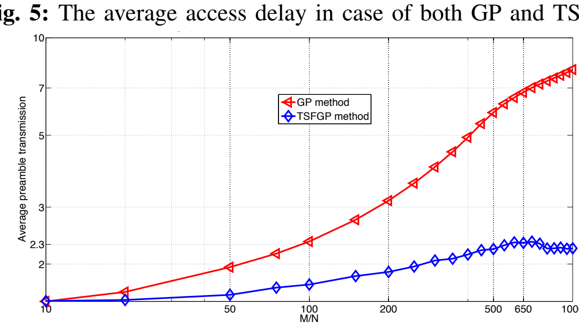

Fig. 6: The average Preamble transmission in case of both GP and TSFGP.

the last 11 RA slots do not have the possibility to transmit the preamble 2.4 times, until reaching a stable case, i.e. when the number of activated MTC devices at the last RA slot in the GP interval is equal to M_1.

V. Conclusion

In this paper, as an improvement to the GP mechanism, we proposed the TSFGP scheme, whereby, instead of simultaneous network access attempts of a group of MTC devices, the devices are instructed to access the network at times scattered over a predetermined interval. Although there is always a part of MTC devices that are not immediately activated, TSFGP highly outperforms the GP method and that is in terms of the success and collision probabilities. Regarding the latter, it is largely decreased by TSFGP to about three times when $M/N = 1000$. Besides, TSFGP reduces the number of preamble transmissions which represents an important achievement in terms of energy conservation. Furthermore, it must be admitted that the performance of TSFGP can be further improved by increasing the group paging interval, i.e. I_{max}. This defines one of the authors' future research directions on the topic of this paper.

VI. Appendix

In this section, we try to rewrite Equation 12. First of all, we have

$$W_i = \frac{M_i}{M_1} = \sum_{n=1}^{N_{PT_{max}}} W_i[n] = \sum_{n=1}^{N_{PT_{max}}} \prod_{k=1}^{n-1} (1 - e^{-\frac{M_i}{R}} p_k)$$

When varying n frome 1 to $N_{PT_{max}}$, we obtain

$$W_i[1] = 1$$

$$W_i[2] = 1 - e^{-\frac{M_i}{R}} p_1$$

$$\begin{aligned} W_i[3] &= (1-e^{-\frac{M_i}{R}} p_1)(1-e^{-\frac{M_i}{R}} p_2) \\ &= 1 - (p_1 + p_2) e^{-\frac{M_i}{R}} + p_1 p_2 e^{-\frac{2M_i}{R}} \end{aligned}$$

$$\begin{aligned} W_i[4] &= (1 - e^{-\frac{M_i}{R}} p_1)(1 - e^{-\frac{M_i}{R}} p_2)(1 - e^{-\frac{M_i}{R}} p_3) \\ &= 1 - (p_1 + p_2 + p_3) e^{-\frac{M_i}{R}} + (p_1 p_2 + p_1 p_3 \\ &\quad + p_2 p_3) e^{-\frac{2M_i}{R}} - p_1 p_2 p_3 e^{-\frac{3M_i}{R}} \end{aligned}$$

$$\begin{aligned} W_i[5] &= (1-e^{-\frac{M_i}{R}} p_1)(1-e^{-\frac{M_i}{R}} p_2)(1-e^{-\frac{M_i}{R}} p_3)(1-e^{-\frac{M_i}{R}} p_4) \\ &= 1 - (p_1 + p_2 + p_3 + p_4) e^{-\frac{M_i}{R}} + (p_1 p_2 + p_1 p_3 + \\ &\quad p_1 p_4 + p_2 p_3 + p_2 p_4 + p_3 p_4) e^{-\frac{2M_i}{R}} - (p_1 p_2 p_3 + \\ &\quad p_1 p_2 p_4 + p_1 p_3 p_4 + p_2 p_3 p_4) e^{-\frac{3M_i}{R}} + p_1 p_2 p_3 p_4 e^{-\frac{4M_i}{R}} \end{aligned}$$

...

Now, summing up the similar terms, we obtain:

$$\begin{aligned} W_i &= W_i[1] + W_i[2] + W_i[3] + W_i[4] + W_i[5] + ... \\ &= \sum_{t=1}^{N_{PT_{max}}-0} (-1)^0 1 + \sum_{t=1}^{N_{PT_{max}}-1} (-1)^1 \sum_{k_1=1}^{t} p_{k_1} e^{\frac{-M}{R}} + \\ &\quad \sum_{t=1}^{N_{PT_{max}}-2} (-1)^2 \sum_{k_1=1}^{t} \sum_{k_2=k_1+1}^{t+1} p_{k_1} p_{k_2} e^{\frac{-2M}{R}} + \\ &\quad \sum_{t=1}^{N_{PT_{max}}-3} (-1)^3 \sum_{k_1=1}^{t} \sum_{k_2=k_1+1}^{t+1} \sum_{k_3=k_2+1}^{t+2} p_{k_1} p_{k_2} p_{k_3} e^{\frac{-3M}{R}} + ... \end{aligned} \tag{17}$$

From Equation 17, we can conclude that:

$$\begin{aligned} W_i = &\sum_{m=0}^{N_{PT_{max}}-1} \sum_{t=1}^{N_{PT_{max}}-m} (-1)^m \times \\ &\underbrace{\sum_{k_1=1}^{t} \sum_{k_2=k_1+1}^{t+1} ... \sum_{k_m=k_{m-1}+1}^{t+m-1}}_{\text{m times}} p_{k_1} ... p_{k_m} e^{-\frac{mM_i}{R}} \end{aligned} \tag{18}$$

Let α_m be equal to:

$$\alpha_m = \sum_{t=1}^{N_{PT_{max}}-m} (-1)^m \underbrace{\sum_{k_1=1}^{t} ... \sum_{k_m=k_{m-1}+1}^{t+m-1}}_{\text{m times}} p_{k_1} ... p_{k_m} \tag{19}$$

Therefore, we have:

$$W_i = \frac{M_i}{M_1} = \sum_{m=0}^{N_{PT_{max}}-1} \alpha_m e^{-\frac{mM_i}{R}} \tag{20}$$

which is equal to Equation 13.

References

[1] L. Changwei, "Telco development trends and operator strategies," *WinWin Magazine*, no. 13, pp. 19–22, July 2012.

[2] 3GPP TR 37.868 V11.0.0, "Study on RAN improvement for Machine-type-Communications," September 2012.

[3] O. Arouk, A. Ksentini, Y. Hadjadj-Aoul, and T. Taleb, "On improving the group paging method for machine-type-communications," in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 484–489.

[4] A. Ksentini, T. Taleb, X. Ge, and H. Honglin, "Congestion-aware MTC device triggering," in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 294–298.

[5] T. Taleb and A. Kunz, "Machine type communications in 3GPP networks: potential, challenges, and solutions," *Communications Magazine, IEEE*, vol. 50, no. 3, pp. 178–184, March 2012.

[6] T. Taleb and A. Ksentini, "On alleviating mtc overload in eps," *Ad Hoc Networks*, vol. 18, no. 0, pp. 24 – 39, 2014.

[7] M.-Y. Cheng, G.-Y. Lin, H.-Y. Wei, and A.-C. Hsu, "Overload control for machine-type-communications in lte-advanced system," *Communications Magazine, IEEE*, vol. 50, no. 6, pp. 38–45, June 2012.

[8] 3GPP TS 36.331 V11.5.0, "Radio Resource Control (RRC)," Sep. 2013.

[9] A. Ksentini, Y. Hadjadj-Aoul, and T. Taleb, "Cellular-based machine-to-machine: overload control," *Network, IEEE*, vol. 26, no. 6, pp. 54–60, November 2012.

[10] O. Arouk and A. Ksentini, "Multi-Channel Slotted Aloha Optimization for Machine-Type-Communication," in *Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, 2014, pp. 119–125.

[11] 3GPP R2-104870, "Pull Based RAN overload control," August 2010.

[12] 3GPP R2-104873, "Comparing Push and Pull based Approaches for MTC," August 2010.

[13] C.-H. Wei, R.-G. Cheng, and S.-L. Tsao, "Performance analysis of group paging for machine-type communications in lte networks," *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 7, pp. 3371–3382, September 2013.

[14] wikipedia. Exponential function. [Online]. Available: http://en.wikipedia.org/wiki/Exponential_function

[15] 3GPP TR 36.912, "Feasibility study for Further Advancements for E-UTRA (LTE-Advanced)," September 2012.