Congestion Control for M2M Communications in LTE Networks

by

Suyang Duan

B.E., Zhejiang University, 2011

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

June 2013

Abstract

When incorporating machine-to-machine (M2M) communications into the Third Generation Partnership Project (3GPP) Long Term Evolution (LTE) networks, one of the challenges is the traffic overload due to a large number of machine type communication (MTC) devices with bursty traffic. One approach to tackle this problem is to use the access class barring (ACB) mechanism to regulate the opportunity of MTC devices to transmit request packets. In this thesis, we first present an analytical model to determine the expected total service time. For the ideal case that the LTE base station (eNodeB) has the information of the number of backlogged users, we determine the optimal value of the ACB factor, which can reduce congestion and access delay. For the practical scenario, we propose a heuristic algorithm to adaptively change the ACB factor without the knowledge of the number of backlogged users. Results show that the proposed heuristic algorithm achieves near optimal performance. We also study the scenario where the number of preambles dedicated to M2M traffic is not fixed and investigate whether dynamic resource allocation can reduce the average number of random access opportunities per MTC device. Simulation results show that the fixed resource allocation scheme can achieve as good performance as the dynamic scheme in reducing the number

of opportunities and thus dynamic resource allocation is not necessary.

Preface

A version of Chapter 2 has been submitted for publication. I was responsible for deriving the analytical model, proposing the algorithm and carrying out simulations. I was also responsible for studying the dynamic resource allocation scheme presented in Section 2.5. The submitted paper was originally prepared by me, and further revised by all the co-authors: Suyang Duan, Vahid Shah-Mansouri, and Vincent W.S. Wong.

Table of Contents

List of Figures

List of Acronyms

3GPP	The Third Generation Partnership Project
ACB	Access Class Barring
ACK	Acknowledgement
AGTI	Access Grant Time Interval
CN	Core Network
CSMA	Carrier Sensing Multiple Access
DC	Digital Camera
DFT	Deferred First Transmission
EAB	Extended Access Barring
eNodeB	E-UTRAN Node B, or Evolved Node B
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
GPRS	General Packet Radio Service

H2H	Human-to-Human
HDTV	High Definition Television
IEEE	The Institute of Electrical and Electronics Engineers
LTE	Long Term Evolution
MAC	Medium Access Control
MME	Mobility Management Entity
MTC	Machine Type Communication
M2M	Machine-to-Machine
PAN	Personal Area Network
PDCCH	Physical Downlink Control CHannel
PID	Proportional Integral Derivative
PLMN	Public Land Mobile Network
PRACH	Physical Random Access CHannel
PUSCH	Physical Uplink Shared CHannel
QoS	Quality of Service
RACH	Random Access CHannel

RAN	Radio Access Network
RB	Resource Block
RN	Radio Network
SGSN	Serving GPRS Support Node
SNR	Signal-to-Noise Ratio
UE	User Equipment

List of Symbols

C	Number of preambles that have been selected by more than one user
C_i	Number of preambles that have been selected by more than one user in time slot i
$\hat{C}$	Average number of preambles that have been selected by more than one user during the past h time slots
D_m	Indicator of the number of users that select the m^{th} preamble. $D_m = 0$ indicates idle preamble. $D_m = 1$ indicates successful transmission. $D_m = c$ indicates the m^{th} preamble has been selected by more than one user
$f(j-k, M-k)$	Number of ways to put $j-k$ different objects into $M-k$ different cells, so that each of these cells either has no object, or at least has two objects
I_A	Number of random access time slots within the activation time

I_X	Total service time, defined as the number of random access time slots required for all the MTC devices to transmit one packet each.
K_i	Number of successful transmissions at time slot i
M	Number of preambles
N	Number of MTC devices
N_i	Number of backlogged users at time slot i
N_i^a	Number of backlogged users who pass the ACB check at time slot i
p	ACB factor
$g(t)$	Arrival probability distribution function
$\mathbf{q_i}$	State vector for the i^{th} time slot, $\mathbf{q_i} = (q_{i,0}, q_{i,1}, \ldots, q_{i,N})$
$q_{i,n}$	Probability that during i^{th} time slot, there are n backlogged users in the system, $n = 0, 1, \ldots, N$
$\mathbf{R}$	$(N+1) \times (N+1)$ transmission probability matrix
r_{st}	Element of matrix $\mathbf{R}$, which is the probability that given s backlogged users in the system, $s - t$ users pass the ACB check and transmit packets successfully without collision

S	The set of events that at least one cell has exactly one object, $S = S_1 \cup S_2 \cup \cdots \cup S_{M-k}$.
S_c	Set of events where the c^{th} cell has exactly one object
T_A	Activation time
t_0	The time when system starts. It is also the time when activation of MTC devices begins
t_{I_A}	The time when activation stops, *i.e.*, no more new activation will take place after this time
W_i	Cumulative number of successful transmissions up to time slot i
$\mathcal{B}(\alpha, \beta)$	Beta distribution function with parameters α and β
λ_i	Number of new activations in the i^{th} time slot
μ	Mean of C
σ^2	Variance of C

Acknowledgements

I would like to thank my supervisor, Prof. Vincent Wong for his support and help in finishing my degree. I would like to thank Dr. Vahid Shah-Mansouri for his help in finishing the paper. I wish to thank Dr. Hu Jin for his help in discussing the idea and formulating the problem. I wish to thank Qingsan Zhu for his help on building the analytical model. I also wish to thank Enxin Yao, Bojiang Ma, Binglai Niu, Zehua Wang, Jun Zhu, Peiran Wu, Hao Ma, Shaobo Mao for all the help they provided. I would like to thank my parents for their efforts to bring me up.

Chapter 1

Introduction

This chapter begins with the introduction of machine-to-machine (M2M) communications and machine-type-communication (MTC) devices. The problems regarding incorporating M2M communications into the Third Generation Partnership Project (3GPP) Long Term Evolution (LTE) network are discussed, followed by a literature survey on the existing research of M2M communications as well as the motivations and the contributions of this thesis. The list of publications and the structure of the thesis are given at the end of this chapter.

1.1 M2M Communications

M2M system is a network which includes a large number of MTC devices that can communicate with little or no human intervention in order to accomplish specific tasks. M2M communications enable the implementation of the Internet of Things, in which ubiquitous connections can be established either on demand or periodically [1]. According to [2], there will be 12.5 billion MTC devices (excluding smart phones and tablets), in the world in 2020, up from 1.3 billion today. 3GPP is active in developing M2M-related

standards for LTE networks.

According to [3], an MTC device is a user equipment (UE) equipped for M2M communications. It can communicate through a public land mobile network (PLMN) with MTC servers and/or other MTC devices. Although local communications are also possible between MTC devices through a wireless personal area network (PAN), the major form of communications for an MTC device is to communicate via cellular networks, *e.g.*, evolved node B (eNodeB) in LTE networks.

M2M communications have a wide range of applications. Based on the type of communications between MTC devices and eNodeB, there are two different categories: data monitoring and data exchange. Data monitoring refers to one way data flow from MTC devices to eNodeB. The applications include vital sign monitoring in health care system, monitoring of the oil pipelines and on-demand charging transactions in e-commerce [1]. In this category, MTC devices are used as sensors to report data to the data center for processing. On the other hand, applications in the second category exchange data with eNodeB. MTC devices report data to eNodeB, and after raw data processing, eNodeB will provide feedback with processed data as well as instructions to be carried out. Fleet management and smart meters in smart grid are two major applications in this category [3]. In many cities of China, taxis are equipped with MTC devices to report the location of the car periodically or on demand. The taxi company can schedule a nearby taxi that is available to serve the passenger upon receiving requests. Locations of taxis that are updated and reported periodically can also be used to predict potential traffic jams. In

smart grid, real time pricing is used to reflect the current supply-demand level. Smart meters report the current load to the utility company and receive updated price, and sometimes schedule controllable appliances according to the price.

1.2 M2M Communications in LTE Networks

Using cellular network as the air interface for M2M communications has several advantages. The network coverage of the service provider makes it possible to deploy MTC devices in most urban and rural areas, and the backhaul of the LTE networks can provide seamless communication between MTC devices and MTC applications [4]. The well established cellular network infrastructure makes it unnecessary to deploy new base stations dedicated to M2M communications, and the service provider can better utilize its resource by dividing its under-utilized frequency bands for human-to-human (H2H) traffic and M2M traffic respectively to make more profit.

However, as cellular networks are optimized for H2H communications, there are several problems concerning MTC devices accessing cellular networks. One of the problems is efficiency. Compared to H2H communications which have high data rate, M2M communications usually feature low data rate as well as infrequent transmissions. The signalling overhead used in H2H communications to achieve synchronization and resolve contention can be much larger than the size of actual user data packet [5]. The problem is even worse for battery-powered MTC devices which consume most of their power on data transmission.

Another problem is the network congestion, including air interface congestion and core network (CN) congestion. Air interface congestion takes place when a large number of devices are attached to a single eNodeB. As described in [3], the number of MTC devices within a cell can be significantly large, *e.g.,* thousands of devices accessing a single base station. The system will suffer from severe congestion if these devices try to transmit to eNodeB within a short period of time. If congestion does not happen at the radio network (RN) and the data packets have been successfully received by eNodeB, packets from different eNodeBs arrive at the serving GPRS support node - mobility management entity (SGSN-MME), the gateway to the CN, and congestion can also take place there. In [6], it is discussed that MTC related signaling congestion and overload can be caused by: a) an external event triggering massive numbers of MTC devices to attach/connect all at once; b) recurring M2M applications that are synchronized to the exact (half/quarter) hour. Depending on the network infrastructure, these two can take place both at the RN and CN.

1.3 Related Work

Current research papers on M2M communications aim at three aspects: (a) how to reduce the influence of M2M traffic on H2H traffic when they are sharing the network resources, (b) how to improve the efficiency of transmission due to high signaling overhead, and (c) how to perform congestion control to reduce transmission delays, increase throughput and guarantee quality of service (QoS).

1.3.1 How to Reduce the Impact on H2H Traffic

As M2M traffic and H2H traffic share the limited network resources, different allocation schemes can potentially affect the performance of H2H traffic. Lee *et al.* in [7] compared the performance of two scenarios. The radio access resources are split into two sets. In the first scenario, each set of resources is assigned to one type of traffic. In the second scenario, one set of resources is assigned to H2H traffic and the other set is shared by H2H and M2M traffic. An analytical model is presented for throughput analysis with numerical results of the H2H throughput under different traffic models and rates. Results showed that when the arrival rate of H2H traffic is fixed and small, the first scheme outperforms the second. When H2H traffic is very large, the second scheme has a better performance under different M2M traffic rate. On the other hand, given a certain M2M traffic arrival rate, both schemes have similar performance under different H2H traffic rates.

In [8], a system model to estimate the performance of H2H traffic under the impact of M2M traffic is presented. Emulations are carried out to obtain results regarding different coding schemes, signal-to-noise ratios (SNRs) and building densities, *i.e.*, number of devices within an area. The work [8] focuses on the performance of LTE under different traffic models, specifically how many resource blocks (RBs) should be allocated for a user which can either be an H2H user with data rate of 1 $Mbit/s$ or an M2M user with a packet of 10 $kbytes$. An analytical Markovian model is proposed using the number of RBs as the state parameter, for different QoS classes within the system, and blocking probability of H2H users under different M2M traffic arrival rate using this model is

derived.

1.3.2 How to Increase the Transmission Efficiency

As the signalling overhead of a packet for M2M communications is usually much larger than that of the actual user data, the efficiency of data transmission tends to be very low. To solve this problem, some papers proposed schemes based on data aggregation so as to transmit multiple user data within a single packet. Wu *et al.* in [1] proposed an architecture for M2M communications that uses an aggregation point to serve as a relay between MTC devices and eNodeB. According to this architecture, the first hop from MTC devices to the aggregation point can be achieved using either IEEE 802.11, IEEE 802.15 or power line communications. Then packets are aggregated and forwarded to eNodeB via cellular networks. At the same time, direct communications are also allowed for some MTC devices to directly access eNodeB. A similar scheme is also proposed in [9] which uses wireless connection for both hops.

Using the idea of self-organized network, Tu *et al.* in [10] and Ho *et al.* in [11] studied the joint problem of massive access management and energy efficiency. To avoid massive access attempts, group-based communications are used. Within each group, a coordinator is selected so that all other MTC devices within the group send their packets to the coordinator, which then forwards the packets to the base station. The coordinator within each group is chosen based on optimum energy consumption so as to minimize the total energy consumption within each group, and consequently the total

energy consumption of the system.

Zhou *et al.* in [12] proposed a scheme that each MTC device does not transmit a packet immediately upon packet arrival, but instead waits for a number of packets and then aggregates these packets into a single packet and transmits. The collision probability can be reduced greatly with this method since the total transmission attempts are reduced, and yet the scheme may generate longer packet delays. A semi-Markov chain model is presented to study the tradeoff between latency and collision rate.

While MTC devices used in fleet management are mobile devices, some other MTC device applications have fixed locations, such as smart meters in smart grid. As mentioned in Section 1.2, the low efficiency problem is caused by large size of packet overhead compared to small user data size, and the overhead of packets is used for MTC devices to achieve synchronization with eNodeB as well as contention resolution. The round trip delay of MTC devices with fixed locations will not change with time, and there is a possibility for fixed-location MTC devices to skip steps for synchronization and proceed to transmit user data directly. In light of this, Ko *et at.* in [13] proposed a new random access scheme, which can skip steps during the signaling exchange period before actual user data is transmitted. During the random access process, an MTC device receives timing alignment instruction from eNodeB. Assume that this device has succeeded in at least one transmission earlier and knows its previous timing alignment value. If this value matches with the new instruction from eNodeB, then the MTC device will skip synchronization steps and proceed to user data transmission. Simulation results show

that the transmission efficiency is greatly improved and the scheme yields significantly shorter packet delays and lower collision probability.

1.3.3 Overload Control

Different solutions are proposed by 3GPP to alleviate the overload problem in [3]. These solutions are as follows.

1. Access class barring (ACB) scheme: eNodeB broadcasts an ACB factor between 0 and 1 via control signalling to individual UEs or UE groups. Every time an MTC device initiates a transmission, it randomly generates a number between 0 and 1. If this number is less than the ACB factor, it proceeds to transmission. Otherwise, it will go to backlogged status and wait for the next available time slot. In addition to normal ACB, extended access barring (EAB) is also proposed to selectively control access attempts from UEs that are considered more tolerant to delays. These UEs are configured for EAB and have lower priority in accessing the network compared to normal UEs in case of congestion.

2. Separate random access channel (RACH) resources for MTC devices: Interference between M2M traffic and H2H traffic may exist as these two are sharing the RACH resources. For LTE networks, RACH resources are mainly preambles, which can be split between them. In this case, M2M traffic and H2H communications can take place simultaneously at the same frequency band. It is also possible that separate RBs, the time-frequency blocks that provide random access opportunities,

are divided between them, so that H2H traffic and M2M traffic access the network at different time slots and/or different frequency bands.

3. Dynamic allocation of RACH resources: As the number of MTC devices can be large, and the traffic pattern of M2M may not be uniform, the service provider, instead of allocating resources to MTC devices in a fixed pattern, can dynamically allocate them when the network load is predictable, or in the case that the network is already suffering from congestion.

4. MTC specific backoff scheme: This solution is discussed in details in [14], which uses a dedicated backoff parameter for MTC devices. Compared to H2H traffic, M2M traffic is more tolerant to delay. The MTC backoff time is longer than normal UEs (*e.g.*, smart phones), to disperse random access attempts from MTC devices and reduce the impact on H2H traffic.

5. Slotted access: Different MTC devices are assigned to different access slots, and each device only attempts a random access during its dedicated slot. These access slots correspond to certain time period of system frames, and different MTC devices choose these slots based on their own ID.

6. Pull-based scheme: Instead of waiting for MTC devices to initiate a transmission, eNodeB can broadcast a paging message and enquire about certain information it needs. It can also initiate a transmission when eNodeB is aware that some devices may have data to transmit. Upon receiving the paging message, MTC devices may

choose to transmit immediately or backoff for a certain period according to the paging message.

Based on these basic solutions, a lot of papers have been trying to provide new solutions on how to perform congestion control. If access model of MTC devices is modeled as the slotted ALOHA scheme, then there is an optimal traffic load that will yield the maximum throughput. Assuming eNodeB knows the current traffic load which exceeds the optimal traffic load, the ACB factor can then be set as the ratio of the optimal traffic load over the current traffic load to reduce the number of random access attempts to the optimal value. This scheme is discussed in [15], which uses the channel statistical occupancy rate to estimate the traffic load by dividing the time into time slots and monitor the rate of busy slots over all the sampling time. The scheme can outperform slotted ALOHA for high traffic load, which is a common scenario for M2M communications.

Lien *et al.* in [16] discussed the problem of how ACB factor, *i.e.*, the probability for an MTC device to initiate a random access attempt, can be jointly calculated among several neighboring eNodeBs. The work [16] assumed that the coverage of different eNodeBs have overlaps, and MTC devices located in the overlapped areas can choose one of eNodeBs for access. The system model contains two steps. First, it provides a strategy for each MTC device to independently choose an eNodeB to access based on all the ACB factors broadcast by eNodeBs. Although a larger ACB factor means a higher probability for an MTC device to pass the ACB check and transmit, when all the MTC devices select

the same eNodeB, it may cause packet congestion, and the cumulative delays for these devices may not be reduced. Thus MTC devices can adopt mixed-strategy decision so that the access attempts towards eNodeB with the largest ACB factor will be dispersed to other eNodeBs. Then, given that eNodeBs have the information about the strategy these MTC devices adopt as well as the locations of all the MTC devices, they can try to divide these MTC devices into each cell as equally as possible and then determine the optimal ACB factor accordingly. Simulation results showed that the scheme can reduce average access delay compared to conventional ACB.

It can be seen that the ACB factor is of prime importance in the access class barring scheme. In [17], a congestion-aware admission control solution is proposed to obtain this ACB factor. Instead of estimating this probability based on the traffic of radio access network (RAN), this factor can be obtained based on the length of queue of packets at SGSN-MME. The system uses a proportional integrative derivative (PID) controller to adaptively change the reject probability, using the difference between the current queue length and the reference value as the input. Reject values are determined for different groups of devices that have different priority in accessing the network, and these values are delivered to eNodeBs to perform access reject at RAN accordingly. The PID gains are empirically obtained by simulations. Compared to a fixed factor ACB, the scheme can reduce the queue length and the number of dropped packets at MME. In other words, congestion level is decreased.

Taleb *et al.* in [18] proposed a bulk signaling handling scheme. When a large number of MTC devices are triggered simultaneously and initiate signaling transmissions to eNodeBs, congestion may take place at MME. In [18], these signaling packets are held at eNodeB until a certain number of signaling messages have arrived. By analyzing the structure of the signaling packet, a signaling packet aggregation scheme is proposed which can be used for controlling congestion and overload.

Using drift analysis, Wu *et al.* in [19] utilized the statistics of consecutive idle and collision slots to reduce access delay under bursty traffic situation. As the number of competing nodes in random access has great influence on system performance, a scheme is proposed to estimate the number of MTC devices that try to access eNodeB. Unlike fixed-step drift analysis, the proposed algorithm is fast-converging and robust in estimating the state information which can adaptively change the size of the step. It is also suitable for the bursty traffic case where a large number of MTC devices are activated and try to access one single eNodeB around the same time.

Sheu *et al.* in [20] proposed a self-adaptive scheme to schedule MTC devices which report periodically to eNodeB. The scheme is a combination of ACB, separate RACH resources, dynamical allocation of RACH resources, MTC specific backoff and pull-based schemes. In addition to these aspects, the scheme proposed that MTC devices inherit the same contention resource from the previous successful experience so as to avoid collisions caused by periodical reporting. If resource allocation has not been updated, these MTC devices will keep on using the same contention resource until the contention level at

eNodeB is stable and eNodeB reduces the resources allocated for MTC devices. Then each MTC device will recalculate which resource to access based on a rule known to all MTC devices so that rescheduled devices will not collide on new MTC resources.

When the number of MTC devices is large, an effective medium access control (MAC) protocol which can provide scalable solutions for M2M communications is of crucial importance. Liu *et al.* in [21] proposed a frame-based hybrid MAC scheme for M2M networks. In this scheme, a frame is divided into contention period and transmission period, and the length of both periods can be changed dynamically. MTC devices first contend for transmission during the contention period, and the transmission period will provide random access opportunities for devices that succeed in the contention. An optimization problem on how to set the length of both periods is formulated in [21] to maximize the system throughput, and the number of devices that can transmit during the transmission period.

As MTC devices contain a wide range of different applications which have different QoS requirements, congestion control schemes can be designed based on satisfying requirements of each QoS class and allocating resources among different classes. Lien *et al.* in [22] and Gotsis *et al.* in [23] used packet delays as the QoS requirement. The model uses time controlled feature of LTE network, *i.e.*, the network only allows MTC device access attempts within an allocated access grant time interval (AGTI). Different AGTIs are allocated to each class based on its access priority and traffic rate. The work [22] considered constant traffic arrival rate while the work [23] studied event-driven bursty

traffic and applied queuing model to each class. The scheme in [22] guarantees an average experienced delay for each class while [23] derives a bound for probabilistic packet delay. Kwon *et al.* in [24] studied the problem of minimizing resources allocated for MTC devices in a multicell system, which uses the outage-probability as the QoS requirement. The work [24] considered not only collisions caused by simultaneous transmissions of MTC devices within a cell, but also interference from MTC devices in neighboring cells.

The envisioned M2M communications can also be applied to home networks. In [25], Zhang *et al.* proposed an architecture of home area M2M networks, and studied QoS management in home M2M networks. In home M2M networks, multimedia devices, sensors, smart meters and smart phones can all be part of the network, among which multimedia devices, such as digital camera (DC) and high definition television (HDTV), can consume much network resources. The work [25] focused on the QoS of multimedia devices, studied three transmission standards for multimedia devices and outlined a cross-layer joint admission and rate control scheme, which can allocate radio bandwidth based on QoS demand intelligently in resource-constrained home area M2M networks.

1.4 Motivations and Contributions

In this thesis, our focus lies in alleviating congestion in RAN. We aim to manage random access attempts by the users to reduce the congestion in an overload condition instead of rejecting access at eNodeB or CN in LTE networks. In case of an emergency, it is crucial that all the information from every single MTC device is collected as soon as possible.

Therefore, we need to minimize the total amount of time it takes for all the MTC devices to finish user data transmissions. We consider the use of ACB scheme with an *adaptive* ACB factor. The contributions of this thesis are as follows:

- We first derive a detailed analytical model to determine the minimum time required to handle all the requests from the users. We obtain the expectation for the time required to handle the requests of all the MTC devices where new traffic arrivals follow a beta distribution.
- We propose an algorithm to dynamically adjust the ACB factor.
- The analytical model is validated by simulation results. Results also show that our proposed heuristic algorithm can achieve near optimal performance. Simulation results under different traffic models are presented which show the robustness of the proposed algorithm.

Our work differs from related works in different directions. In our work, we use the number of collisions in RAN to determine the ACB factor. This is different from [17] using the collision information in CN. As ACB is a method that performs congestion control at RAN, the congestion level at CN may not reflect the congestion level at RAN. If each eNodeB has a small traffic load while the number of eNodeBs is large, then congestion will only happen at CN due to large number of packets from different eNodeB. It is also possible that each eNodeB is suffering from heavy congestion while CN has few packet arrivals because congestion in RAN results in small number of successful transmissions.

In both situations, the scheme in [17] may fail. The work [15] uses the channel statistical occupancy rate to estimate the traffic load and determine the ACB factor. However, this occupancy rate may not be accurate when collision happens. If two users collide using the same period of time and neither succeeds, no throughput is realized, and the system will treat this period as non-occupied. This does not reflect the real congestion level at RAN. Thus it is more accurate to determine ACB factor based on RAN congestion level. Also, we formulate our problem based on a multi-channel random access model. This is different from the conventional model used in single channel random access [19]. In LTE networks, there are 64 preambles available for random access within each cell. Simultaneous transmissions are possible, which is a multi-channel problem instead of a single one. Our work is also different in our beta distribution traffic model instead of conventional Poisson distribution, the latter of which is more suitable for traffic pattern with exponential inter-arrival time rather than a limited time bursty traffic.

1.5 List of Publications

The following publications have been completed based on the work during the MASc study.

- Vahid Shah-Mansouri, Suyang Duan, Ling-Hua Chang, Vincent W.S. Wong, and Jwo-Yuh Wu, "Compressive Sensing based Asynchronous Random Access for Wireless Networks, in *Proc. of IEEE Wireless Communications and Networking Con-*

ference (WCNC), Shanghai, China, April 2013.

- Suyang Duan, Vahid Shah-Mansouri, and Vincent W.S. Wong, "Dynamic Access Class Barring for M2M Communications in LTE Networks," submitted to *IEEE Global Communications Conference (GLOBECOM)*, Atlanta, GA, Dec. 2013.

1.6 Structure of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we present the system model. An introduction on LTE random access procedures is first discussed, followed by the analytical model to determine the total service time. We propose a heuristic algorithm to update the transmission probability p so as to reduce the total service time without full system information. Numerical results of the analytical model and simulation results are presented to show that the algorithm can achieve near-optimal results. Simulation results on different traffic models are presented to show the robustness of our algorithm. We also discuss how to reduce the average number of random access opportunities per MTC device with dynamic resource allocation. The thesis is concluded in Chapter 3 with conclusions and future work.

Chapter 2

Dynamic Access Class Barring for M2M Communications in LTE Networks

In this chapter, we propose an adaptive ACB scheme for congestion control for M2M communications in LTE networks. We first summarize the procedures of random access process in LTE networks in Section 2.1. In Section 2.2, we present our analytical model to estimate the total service time, the time for all MTC devices associated with eNodeB to finish transmitting one single packet from each device. In Section 2.3, we propose a heuristic algorithm to adaptively change the ACB factor, p, so as to reduce the total service time. Numerical results are presented in Section 2.4. In Section 2.5, we study how to reduce the average number of random access opportunities per MTC device by dynamic resource allocation. A summary is given in Section 2.6.

2.1 Random Access Procedures in LTE Networks

In this section, we summarize the random access procedure in LTE networks. In LTE networks, user data is transmitted through Physical Uplink Shared CHannel (PUSCH) via scheduled transmissions. Asynchronous devices acquire synchronization with eNodeB and reserve uplink channel using RACH. RACHs are repeated in the system with a certain period. Each node requiring an uplink channel transmits a preamble in a RACH. There are two types of access in a RACH. The first type is contention-based, which is used for regular users. The second type is contention-free, which provides low latency service for users with high priority (*e.g.*, handover). In this chapter, we only focus on the contention-based random access, which consists of the following steps [26].

- Step 1: Preamble transmission;
- Step 2: Random access response;
- Step 3: Layer 2/Layer 3 (L2/L3) message;
- Step 4: Contention resolution message.

In Step 1, each UE randomly selects a sequence called preamble from a pool known both to UEs and eNodeB. Transmission of this sequence serves as a request for a dedicated time-frequency resource block in the upcoming scheduling transmission in Step 3. As UEs only transmit the sequence without indicating their own IDs in the request, when two UEs select the same preamble, eNodeB will receive the same sequence. In Step 2,

eNodeB acknowledges all the preambles it has successfully received, conveying a timing alignment instruction so that subsequent transmission can be synchronized. In Step 3, UEs begin using PUSCH to transmit their IDs upon receiving the acknowledgement (ACK). If two UEs have selected the same preamble in Step 1, both will be instructed to transmit their IDs within the same time-frequency slot in Step 3. In this case, collision will happen. In Step 4, contention resolution message will be broadcast with the ID of UEs successfully decoded by eNodeB. If a collision happens while eNodeB still manages to decode the message in Step 3, it will inform the UE whose Step 3 message is decoded and this successful UE will send an ACK. Unacknowledged UEs remain silent until the next RACH.

In an LTE cell, 64 preambles are available for random access, among which some are reserved for contention-free access. When MTC devices access the LTE network, they have to share the remaining preambles for contention-based access with H2H UEs (*e.g.*, smart phones). In our model, we assume that separate resources are allocated to M2M traffic and H2H traffic. Hence, we only consider how MTC devices compete for dedicated preambles among themselves. Note that random access can only take place within certain time-frequency blocks specified by eNodeB, *i.e.*, Physical Random Access CHannel (PRACH), which is the physical layer mapping of RACH. For example, when PRACH configuration index is set to 6, RACH will occur every 5 ms within a bandwidth of 180 kHz with a duration ranging from 1 ms to 3 ms [26, 27]. In this chapter, we only consider transmissions within the random access channels. Note that here the term

channel refers to a time-frequency RB. It does not refer to the medium that electromagnetic waves travel. In the following analysis, we will use the terms channel and RB interchangeably. Specifically, a PRACH is a time-frequency RB where random access attempts from MTC devices take place, which appear periodically.

2.2 System Model

Consider N MTC devices which have previously registered with eNodeB. These devices have just recovered from an emergency, *e.g.*, a power black out, and all of them try to re-establish synchronization with eNodeB. As these devices are not synchronized, they will not be activated all at once, but within a limited time T_A, denoted as the activation time. Each MTC device is activated at time t with probability density function $g(t)$ for $0 \leq t \leq T_A$. $g(t)$ follows a beta distribution with parameters $\alpha = 3, \beta = 4$ as in [27]

$$g(t) = \frac{t^{\alpha-1}(T_A - t)^{\beta-1}}{T_A^{\alpha+\beta-1}\mathcal{B}(\alpha, \beta)}, \tag{2.1}$$

where $\mathcal{B}(\alpha, \beta)$ is the beta distribution function [28].

Assume there are I_A random access channels within the activation time. The duration of the random access channel is shorter than the interval between two random access channels. We divide the activation time into I_A discrete slots where slot i begins with i^{th} random access channel, as shown in Fig. 2.1. The length of each time slot is equal to the interval between two random access channels. The i^{th} time slot starts at t_{i-1} and ends at t_i. The first time slot starts from $t_0 = 0$. The last one ends at $t_{I_A} = T_A$. All MTC

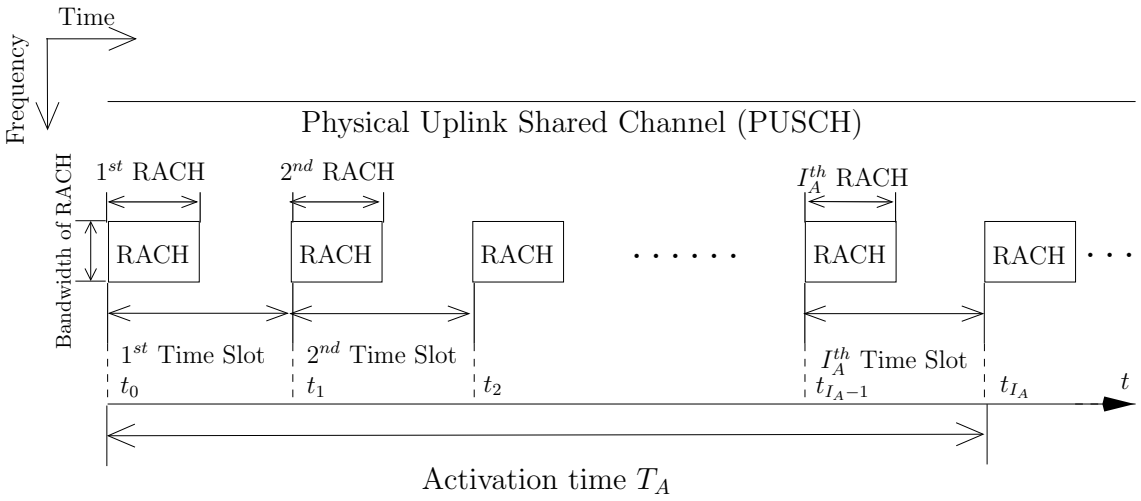

Figure 2.1: Random access time slots.

devices which have been activated within slot i, *i.e.*, they are activated within $[t_{i-1}, t_i]$, choose the random access channel at the beginning of the next slot for their first access trial. According to [27], the expected number of new activations (arrivals) during time slot i, $\lambda_i, i = 1, 2, \ldots, I_A$, is subject to the distribution of activation traffic $g(t)$ and the total number of devices N as

$$\lambda_i = N \int_{t_{i-1}}^{t_i} g(t)\mathrm{d}t, \quad i = 1, 2, \ldots, I_A. \tag{2.2}$$

As a method to alleviate the congestion, eNodeB broadcasts an ACB factor p as part of the system information before each random access opportunity using Physical Downlink Control CHannel (PDCCH). In each random access channel, an MTC device which has not yet connected to the network generates a random number between 0 and 1. If this number is less than p, then the request packet will be sent. Otherwise, the MTC device stays silent and waits for the next random access channel, in which both newly

activated users in the next slot and the backlogged users will perform ACB check before transmission. If more than one MTC device selects the same preamble, then a collision will happen at eNodeB. We assume that when a collision happens, eNodeB will not be able to decode the collided Step 2 message, and thus none of the collided MTC devices succeeds in this access channel. Whenever a user fails in one random access channel, it will try to send the sequence during the following channel after ACB check. This scheme uses the deferred first transmission (DFT), where new arrivals are treated as backlogged users.

We are interested in estimating the total time it takes for eNodeB to collect all users' data. After the call is initiated through RACH, the user data is transmitted without contention on PUSCH via scheduled transmission and the time it takes is constant. Therefore, the dominant part is the time for all the MTC devices to successfully transmit Step 1 preamble sequences, which we denote as the total service time. In total, it takes the system I_X random access channels before all the requests are successfully transmitted. As I_X is a random variable, we determine its expectation, $\mathbb{E}[I_X]$.

For the i^{th} random access channel (*i.e.*, i^{th} time slot), we introduce a $1 \times (N+1)$ state vector $\mathbf{q_i} = (q_{i,0}, q_{i,1}, \ldots, q_{i,N})$, which represents the probability distribution of the number of backlogged users in the system at time slot i. The element $q_{i,n}$ denotes the probability that there are n backlogged users right after the random access channel of slot i. By definition, $\sum_{n=0}^{N} q_{i,n} = 1$, for $i = 0, 1, \ldots$. At the first random access channel starting at time $t_0 = 0$, we have $q_{0,0} = 1$ and $q_{0,n} = 0$, for $n = 1, 2, \ldots, N$.

When $i > I_A$, no more new activation takes place. The probability that there is no backlogged user at $i = I_A$ may be zero. As i increases in the system, $q_{i,0}$ starts growing and approaches 1 eventually. Let $\hat{i}$ denote the smallest $i > I_A$ such that the probability of zero backlogged user in the system is non-zero

$$\hat{i} = \min_{i=0,1,2,\ldots} \{i\} \text{ subject to } q_{i,0} > 0,\ i > I_A. \tag{2.3}$$

For $i > \hat{i}$, $q_{i-1,0}$ and $q_{i,0}$ denote the probability that there is no backlogged user in the system at the beginning and at the end of random access channel i, respectively. For $i > \hat{i}$, $q_{i,0}$ denotes the probability that the system has finished all transmission requests upon completion of random access channel i (*i.e.*, at channel i or before that). The probability that the system finishes all transmissions at random access channel i is $(q_{i,0} - q_{i-1,0})$. The expectation of I_X is

$$\mathbb{E}[I_X] = \sum_{i=1}^{\infty} i(q_{i,0} - q_{i-1,0}). \tag{2.4}$$

As $q_{i,0} = 0$, for $i = 1, 2, \ldots, \hat{i} - 1$, equation (2.4) becomes

$$\mathbb{E}[I_X] = \hat{i} q_{\hat{i},0} + \sum_{i=\hat{i}+1}^{\infty} i(q_{i,0} - q_{i-1,0}). \tag{2.5}$$

The next step is to determine how $q_{i,0}$ evolves with time (*i.e.*, as i increases). We consider the evolution of $\mathbf{q_i} = (q_{i,0}, q_{i,1}, \ldots, q_{i,N})$ over time. In total, there are M preambles available in the system. We denote the number of backlogged users at the i^{th} random access opportunity as N_i, the number of users who pass the ACB check and transmit their preamble as N_i^a, where $N_i^a \leq N_i$, and the number of successful preamble transmissions during that random access channel as K_i. First, we determine the probability of exactly

$K_i = k$ ($k \leq M$) successful preamble transmissions when there are $N_i = n$ backlogged users during the current time slot i, $\mathbb{P}(K_i = k \mid N_i = n)$. This probability consists of three parts:

1. Among n backlogged users, there are $N_i^a = j$ users who pass the ACB check and transmit their preambles, $\mathbb{P}(N_i^a = j \mid N_i = n)$.

2. Among j transmitted preambles, k preambles succeed.

3. The remaining $j - k$ preambles collide.

The first part can be obtained as

$$\mathbb{P}(N_i^a = j \mid N_i = n) = \binom{n}{j} p^j (1-p)^{n-j}. \tag{2.6}$$

An analogy of the second and third parts would be to place j different objects into M different cells, on condition that there are exactly k cells that have one object in each of them, and the remaining cells have either no object, or at least two objects. The number of ways of putting j different objects into M different cells is M^j. First, we choose k objects and k cells, and put one object in each cell. The number of different combinations is $\binom{j}{k}\binom{M}{k}k!$. Then, we put the remaining $j - k$ objects into $M - k$ different cells so that each of these $M - K$ cells either has no object or at least two objects in it. We refer to the number of different ways as $f(j-k, M-k)$. If $M = k$, then there is no cell to put any objects, so that $f(j-k, 0) = 0$. When $j = k$, we have $f(0,0) = 1$. We denote $S_c,\ c = 1, 2, \ldots, M-k$ as the set of events, where the c^{th} cell has exactly one object. Then, the set $S = S_1 \cup S_2 \cup \cdots \cup S_{M-k}$ includes all the cases that at least one cell has

exactly one object. Using the principle of inclusion and exclusion [29], the cardinality of this set is

$$\begin{aligned}|S| &= |S_1 \cup S_2 \cup \cdots \cup S_{M-k}| \\ &= (-1)^0 \sum_{c=1}^{M-k} |S_c| + (-1)^1 \sum_{c=1}^{M-k} \sum_{l \neq c} |S_c \cap S_l| \\ &\quad + (-1)^2 \sum_{c=1}^{M-k} \sum_{l \neq c} \sum_{\substack{r \neq c \\ r \neq l}} |S_c \cap S_l \cap S_r| \\ &\quad + \cdots + (-1)^{M-k-1} |S_1 \cap S_2 \cap \cdots \cap S_{M-k}|, \end{aligned} \tag{2.7}$$

in which

$$\sum_{c=1}^{M-k} |S_c| = \binom{M-k}{1} \binom{j-k}{1} 1!(M-k-1)^{j-k-1},$$

$$\sum_{c=1}^{M-k} \sum_{l \neq c} |S_j \cap S_l| = \binom{M-k}{2} \binom{j-k}{2} 2!(M-k-2)^{j-k-2}.$$

We define $u \triangleq \min(M-k, n-k)$. The last term of this series is

$$|S_1 \cap S_2 \cap \ldots \cap S_{M-k}| = \binom{M-k}{u} \binom{j-k}{u} u!(M-k-u)^{j-k-u}. \tag{2.8}$$

Therefore,

$$|S| = \sum_{c=1}^{u} (-1)^{c-1} \binom{M-k}{c} \binom{j-k}{c} c!(M-k-c)^{j-k-c}.$$

Our goal is to determine the total number of cases where no cell has exactly one object

in it, which is the cardinality of the set $\overline{S}$.

$$\begin{aligned}
|\overline{S}| &= (M-k)^{j-k} - |S| \\
&= (M-k)^{j-k} + \sum_{c=1}^{u} (-1)^c \binom{M-k}{c} \binom{j-k}{c} c!(M-k-c)^{j-k-c} \\
&= \sum_{c=0}^{u} (-1)^c \binom{M-k}{c} \binom{j-k}{c} c!(M-k-c)^{j-k-c} \\
&= f(j-k, M-k). \qquad (2.9)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{P}(K_i = k \mid N_i = n) \\
&= \sum_{j=0}^{n} Pr(N_i^a = j \mid N_i = n) \frac{\binom{j}{k}\binom{M}{k} k! f(j-k, M-k)}{M^j} \\
&= \sum_{j=0}^{n} \binom{n}{j} p^j (1-p)^{n-j} \binom{j}{k} \binom{M}{k} k! \\
&\quad \times \frac{\sum_{c=0}^{u} (-1)^c \binom{M-k}{c} \binom{j-k}{c} c!(M-k-c)^{j-k-c}}{M^j}. \qquad (2.10)
\end{aligned}$$

We introduce an $(N+1) \times (N+1)$ transmission probability matrix,

$$\mathbf{R} = \begin{pmatrix} r_{0,0} & r_{0,1} & \cdots & r_{0,N} \\ r_{1,0} & r_{1,1} & \cdots & r_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N,0} & r_{N,1} & \cdots & r_{N,N} \end{pmatrix}, \qquad (2.11)$$

where $r_{s,t} = 0$, for $t > s$. When $t \leq s$, $r_{st} = \mathbb{P}(K_i = s - t \mid N_i = s)$, which is the probability that given s backlogged users in the system, $s - t$ users pass the ACB check and transmit successfully without collision. In other words, $r_{s,t}$ is the probability that the number of backlogged users changes from s to t. Note that $r_{0,0} = 1$.

For time slot $i > I_A$, there is no new activation in the system. In this case, matrix $\mathbf{R}$ can relate vectors $\mathbf{q_{i+1}}$ and $\mathbf{q_i}$ as $\mathbf{q_{i+1}} = \mathbf{q_i R}$. For time slot $i = 1, \ldots, I_A$, we need to take into account the new arrivals when relating $\mathbf{q_{i+1}}$ and $\mathbf{q_i}$. Given that z MTC devices have been activated until the beginning of time slot i, we have $q_{i,n} = 0$ for $z < n \leq N$

$$\mathbf{q_i} = (q_{i,0}, q_{i,1}, q_{i,2}, \ldots, q_{i,z}, 0, 0, \ldots, 0). \tag{2.12}$$

The vector $\mathbf{q_i}$ shows the probability distribution of the number of backlogged users after completion of the $(i-1)^{th}$ random access channel. If the number of newly activated devices in time slot i is λ_i, we define vector $\mathbf{q'_i}$ by shifting $\mathbf{q_i}$ to the right λ_i units as

$$\mathbf{q'_i} \triangleq (\underbrace{0, 0, \ldots, 0}_{\lambda_i}, q_{i,1}, q_{i,2}, \ldots, q_{i,z}, 0, 0, \ldots, 0). \tag{2.13}$$

The vector $\mathbf{q'_i}$ represents the probability distribution of the number of backlogged users by the end of time slot i (*i.e.*, right before the start of random access channel $i+1$). Therefore, we can compute $\mathbf{q'_i}$ by $\mathbf{q_{i+1}} = \mathbf{q'_i R}$. As we know how $\mathbf{q_i}$ evolves with time for both $i > I_A$ and $i \leq I_A$, the state vector of each time slot can be derived starting from $i = 1$. Consequently, using equation (2.5), the total service time can thus be estimated.

The ACB parameter p plays an important rule in the performance of contention control in a random access channel. Therefore, it is of interest to find the optimal p. If $N_i^a = j$ users among $N_i = n$ backlogged ones pass the ACB check, each of them will choose from M preambles with equal probability, $\frac{1}{M}$. Consider preamble m and let $D_m = 0, 1, c$, respectively denote the cases where the preamble m is selected by none of the users, by exactly one user, and by more than one user. The probability that only one

user selects preamble m is

$$\mathbb{P}(D_m = 1 \mid N_i^a = j) = \binom{j}{1} \frac{1}{M} \Big(1 - \frac{1}{M}\Big)^{j-1}. \tag{2.14}$$

As each preamble is independent of others, the expected number of successful transmissions in time slot i is

$$\begin{aligned} \mathbb{E}\left[K_i \mid N_i^a = j\right] &= \sum_{m=1}^{M} \mathbb{P}(D_m = 1 \mid N_i^a = j) \\ &= M \binom{j}{1} \frac{1}{M} \Big(1 - \frac{1}{M}\Big)^{j-1}. \end{aligned} \tag{2.15}$$

Therefore,

$$\begin{aligned} &\mathbb{E}\left[K_i \mid N_i = n\right] \\ &= \sum_{j=1}^{n} \mathbb{P}(N_i^a = j \mid N_i = n) M \binom{j}{1} \frac{1}{M} \Big(1 - \frac{1}{M}\Big)^{j-1} \\ &= \sum_{j=1}^{n} \binom{n}{j} p^j (1-p)^{n-j} \binom{j}{1} \Big(1 - \frac{1}{M}\Big)^{j-1} \\ &= np\Big(1 - \frac{p}{M}\Big)^{n-1}. \end{aligned} \tag{2.16}$$

The minimum total service time can be achieved if we maximize the number of successful transmissions during each time slot. In other words, the maximum system throughput corresponds to the minimum total service time. By taking the derivative of (2.18) with respect to p, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}p} \mathbb{E}(K_i \mid N_i = n) = n\Big(1 - \frac{p}{M}\Big)^{n-2} \Big(1 - \frac{np}{M}\Big). \tag{2.17}$$

When $M \geq n$, we have $\frac{\mathrm{d}}{\mathrm{d}p} \mathbb{E}(K_i \mid N_i = n) \geq 0$. The maximum throughput is achieved when $p = 1$, *i.e.*, when the preamble number is larger than the number of request packets

waiting to be transmitted, ACB factor should be set to 1. In other words, no ACB check will be performed and packets will be transmitted upon activation. When $M < n$, let $\frac{\mathrm{d}}{\mathrm{d}p}\mathbb{E}(K_i \mid N_i = n) = 0$, then $p = \frac{M}{n}$. Therefore, we have

$$p^* = \min\left(1, \frac{M}{n}\right). \tag{2.18}$$

If the ACB factor can be dynamically changed during each time slot according to equation 2.18, then the minimum total service time can be achieved. We refer to this scheme as the optimal p scenario. Based on equations 2.18 and 2.10, we can determine the transmission probability matrix R, and then $\mathbf{q_i}$ for each time slots. I_X can thus be derived using equation 2.5. On the other hand, I_X can also be determined using simulations. Analytical and simulation results of I_X will be presented in Section 2.4. Note that in reality, eNodeB cannot obtain information about the number of backlogged MTC devices in the system. Thus the optimal p scenario can only serve as a reference for the theoretical minimum total service time. We will propose a heuristic algorithm in the following section, which can adaptively update p to reduce the total service time. This algorithm is based on the information available to eNodeB, and can be realized in real environments.

2.3 A Heuristic Algorithm to Update p

In this section, we present a heuristic algorithm to adaptively update the ACB factor p. In a real system, eNodeB cannot acquire information regarding the number of backlogged

users in the system. The information is limited to the number of successful transmissions and the number of preambles selected by more than one user during each time slot, as well as the total number of M2M devices that have registered in the system, N. There is an inherent tradeoff in choosing ACB factor p. When p is too large, there will be a lot of preambles transmitted in the air, and there will be collisions on most of the preambles. On the other hand, when p is too small, very few users will be able to pass ACB check and transmit their preambles, resulting in fewer collisions but under-utilization of network resources.

At time slot i, assume that there are $N_i = n$ MTC devices trying to access eNodeB, which select with equal probability among all M preambles. In the following analysis, we assume that $n \geq M$. When $n < M$, no ACB check will be performed and all the MTC devices will attempt random access upon activation. The probability that preamble m is selected by a user in a time slot is p/M. The probability that no user chooses preamble m is

$$\mathbb{P}(D_m = 0 \mid N_i = n) = \left(1 - \frac{p}{M}\right)^n. \tag{2.19}$$

We can also obtain the probability that preamble m is selected by exactly one user as

$$\mathbb{P}(D_m = 1 \mid N_i = n) = \binom{n}{1} \frac{p}{M} \left(1 - \frac{p}{M}\right)^{n-1}. \tag{2.20}$$

Therefore, the probability that preamble m is selected by more than one user $\mathbb{P}(D_m = c \mid N_i = n) = 1 - \mathbb{P}(D_v = 0 \mid N_i = n) - \mathbb{P}(D_v = 1 \mid N_i = n)$. The expected number of

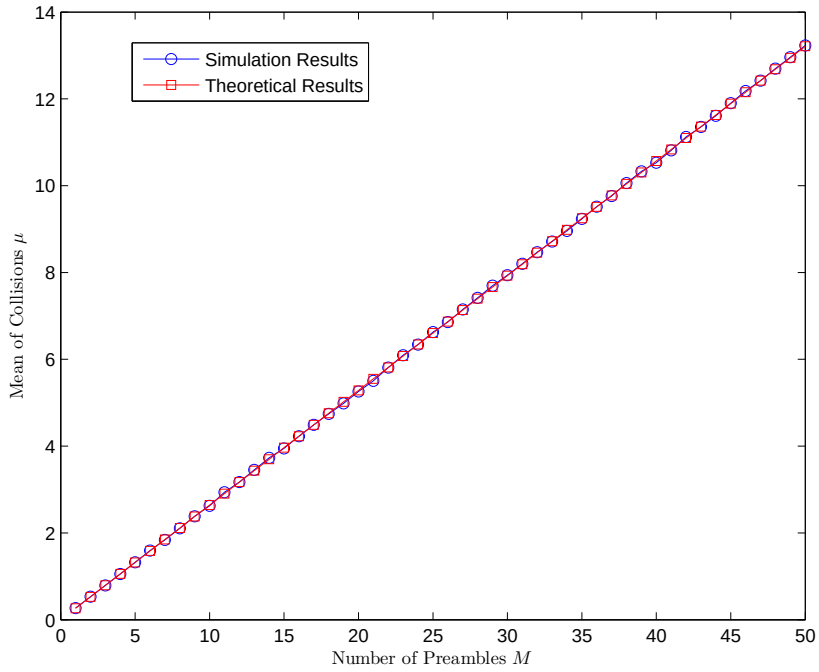

Figure 2.2: Simulation and theoretical values of μ with $N = 1000$.

preambles with collision, $\mathbb{E}[C]$, is thus

$$\begin{aligned}
\mathbb{E}[C] &= \sum_{m=1}^{M} \mathbb{P}(D_m = c \mid N_i = n) \\
&= M\mathbb{P}(D_m = c \mid N_i = n) \\
&= M(1 - \mathbb{P}(D_m = 0 \mid N_i = n) - \mathbb{P}(D_m = 1 \mid N_i = n)) \\
&= M\left(1 - \left(1 - \frac{p}{M}\right)^n - \binom{n}{1}\frac{p}{M}\left(1 - \frac{p}{M}\right)^{n-1}\right).
\end{aligned}$$

If p is equal to the optimal value, $p^* = \frac{M}{n}$, we obtain

$$\mathbb{E}[C] = M\left(1 - \left(1 - \frac{1}{n}\right)^n - \left(1 - \frac{1}{n}\right)^{n-1}\right). \tag{2.21}$$

When n is approaching infinity, $\lim_{n\to\infty} \mathbb{E}[C] = M(1 - 2e^{-1})$. For every preamble, the

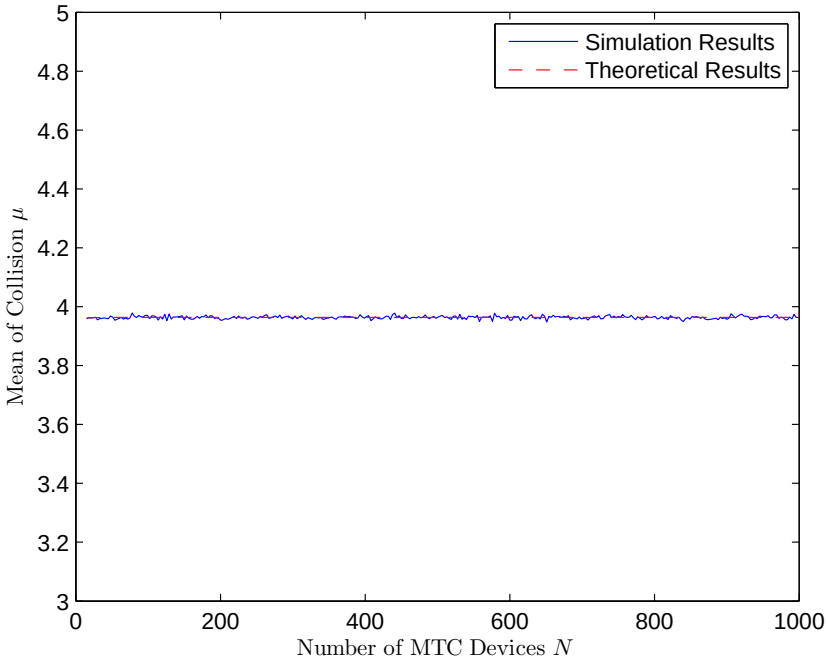

Figure 2.3: Simulation and theoretical values of μ with $M = 15$.

probability that a collision happens is $(1 - 2e^{-1})$. If we assume that the preambles are independent from each other, then the total number of preambles that have collided, C, follows a binomial distribution $C \sim Binomial(M, (1-2e^{-1}))$ with mean $\mu = M(1-2e^{-1})$ and variance $\sigma^2 = M(1 - 2e^{-1}) \times 2e^{-1}$. In Figs. 2.2 and 2.3 we present the simulation results of μ against its theoretical approximation $\mu = M(1 - 2e^{-1})$ for different number of preambles and different number of users. It can be seen that simulation results match the theoretical values.

In Figs. 2.4 and 2.5, simulation results of the variance σ^2 are plotted with its theoretical approximation for different number of preambles and users, respectively. It can be seen in Fig. 2.4 that for a fixed number of users ($N = 1000$), there is a slight difference

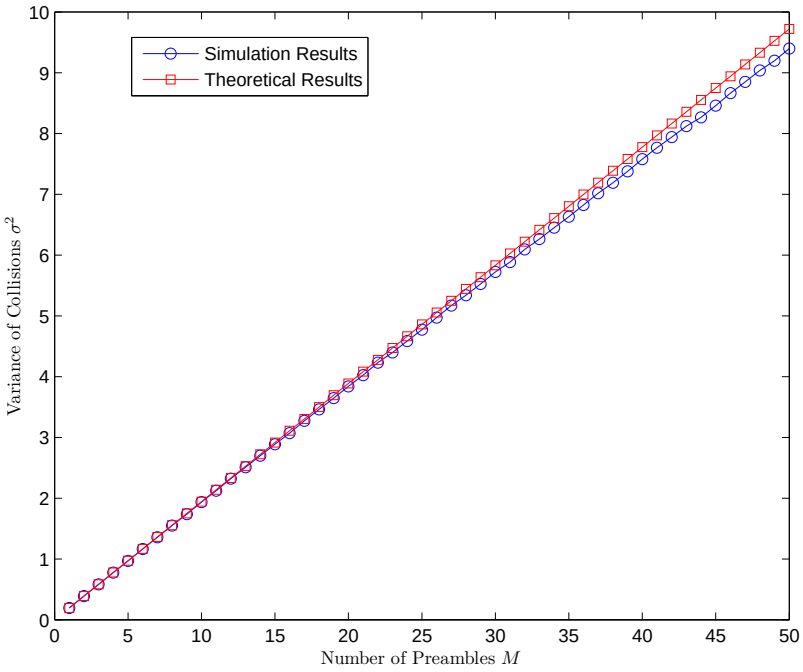

Figure 2.4: Simulation and theoretical values of σ^2 with $N = 1000$.

between simulation results and theoretical values when the number of preambles grows large. On the other hand, Fig. 2.5 shows that when the number of users increase while the number of preambles is fixed ($M = 15$), the simulation results approach the theoretical approximation. These four figures show that our approximation on the mean, μ, is accurate while on the variance, σ^2, can be a good approximation.

We denote the number of preambles that have been chosen by more than one device during time slot i as C_i. To avoid the effect of random fluctuations of C_i, we introduce $\hat{C}$, which is the average of C_i during the past h time slots, as an indicator of the collision status of the system. h can be varied based on the activation time. For each time slot i, $\widehat{C} = \frac{1}{h}(C_{i-1} + C_{i-2} + \cdots + C_{i-h})$. We use $\mu \pm \sigma$ as the threshold to determine collision

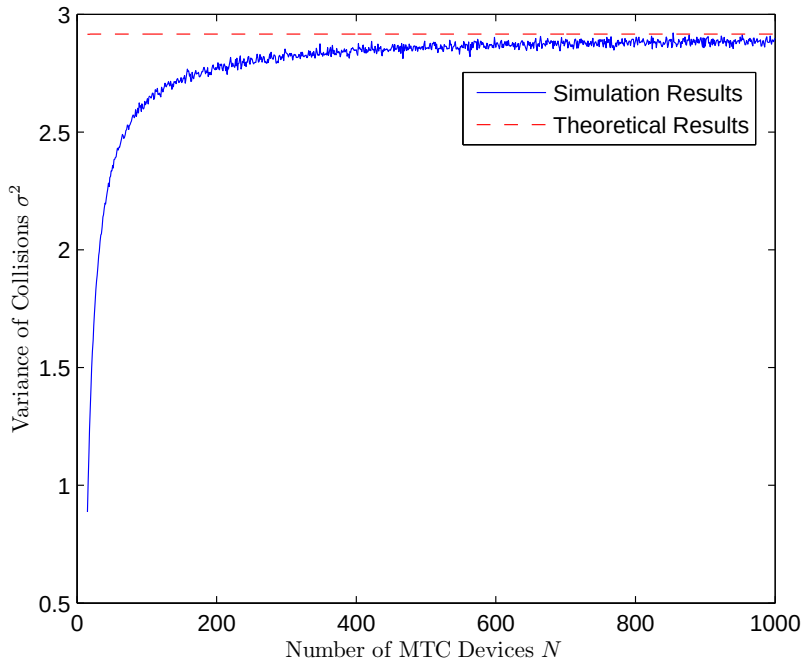

Figure 2.5: Simulation and theoretical values of σ with $M = 15$.

status of the current system. If $\widehat{C} > \mu + \sigma$, then it indicates that the system is having too many collisions, and the transmission probability is decreased. When $\widehat{C} < \mu - \sigma$, p is increased as it implies that the system resource is under-utilized. That is,

$$\text{if } \widehat{C} > \mu + \sigma, \text{ then } p := \nu_1 p, \ 0 < \nu_1 < 1 \tag{2.22}$$

$$\text{if } \widehat{C} < \mu + \sigma, \text{ then } p := \min(\nu_2 p, 1), \ \nu_2 > 1 \tag{2.23}$$

We use these two expressions to update p. The algorithm is shown in Algorithm 1. The initial value of p is 1. p keeps being updated until all the packets have been successfully transmitted. In this algorithm, ν_1 and ν_2 are design parameters used to adaptively adjust p. They are obtained via simulations. We use W_i to denote the cumulative number of successful transmissions up to time slot i. During each time slot, W_i is compared with

Algorithm 1 Algorithm for Adaptively Updating p

Inputs: μ, σ, N, design parameters ν_1, ν_2

Set $i := 1$, $p := 1$, $\widehat{C} := \mu$, $W_0 := 0$

while cumulative successful transmission $W_i < N$ **do**

    Monitor the number of successful transmission, K_i

    Monitor the number of preambles that are chosen by more than one user, C_i

    Update $W_i := W_{i-1} + K_i$

    if $i > h$ **then**

        Update $\widehat{C} := \frac{1}{h}(C_{i-1} + C_{i-2} + \cdots + C_{i-h})$

    end if

    if $\widehat{C} > \mu + \sigma$ **then**

        $p := \nu_1 p$

    else if $\widehat{C} < \mu + \sigma$ **then**

        $p := \min(\nu_2 p, 1)$

    end if

    Time slot $i := i + 1$

end while

N to see if all the packets have been successfully transmitted (Step 3). During time slot i, the number of successful transmissions, K_i, and the number of preambles selected by more than one user, C_i, are monitored by eNodeB (Steps 4 and 5). K_i is used to update W_i (Step 6). The initial value of $\widehat{C}$ is set to be μ. During the first h time slots, $\widehat{C}$ remains unchanged. Starting from the $(h+1)^{th}$ time slot, $\widehat{C}$ will be updated as in Step 8. This value is compared with thresholds to see if p needs updating (Step 10 to Step 13). If p does not exceed $\mu + \sigma$ or go below $\mu - \sigma$, then it will remain unchanged.

2.4 Numerical Results

In this section, we present the numerical results of the analysis and simulation results. We first determine the two parameters, ν_1, ν_2, that need tuning in the algorithm. We aim to find a set of parameters that can yield a performance as close to the optimal as possible. We first plot the optimal curve of the total service time versus the number of preambles. Then for each combination of ν_1 and ν_2, we plot the simulation result and calculate the distance between the estimation curve and the optimal curve, which is defined as the sum of the distances between all the data points on the estimation curve and their corresponding points on the optimal one. Since $\nu_1 < 1$, we vary it from 0.1 to 0.9 with a step of 0.1. The range of ν_2 can be more difficult to determine as $\nu_2 > 1$. We start from large values and large steps and gradually reduce the range and the size of the step. We locate the optimal value of ν_2 between 1 and 2. We vary it from 1.1 to 1.9 with a step of 0.1. Among all 81 combinations of ν_1 and ν_2, the minimum distance,

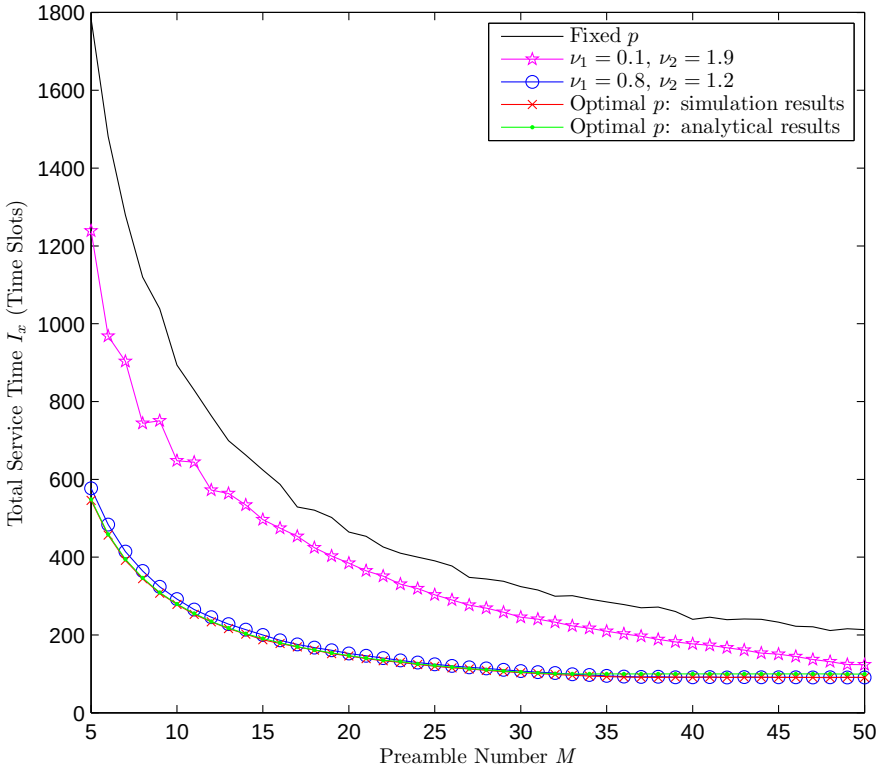

Figure 2.6: The total service time vs preamble number M with $N = 1000$ and $I_A = 100$ under beta distribution.

i.e., the best estimation, occurs at $\nu_1 = 0.8$, $\nu_2 = 1.2$, while the largest distance and the worst estimation happens at $\nu_1 = 0.1$ and $\nu_2 = 1.9$.

As the indicator of the congestion level in the system, $\hat{C}$ is the average of C_i in the past h time slots. h can be changed based on the length of activation time. In our simulations, h is chosen to be equal to 3.

In Fig. 2.6, we present the results of the total service time as the number of preambles M varies from 5 up to 50. The number of users N is equal to 1000, and the activation time I_X is 100. Analytical results and simulation results of the optimal p scenario match,

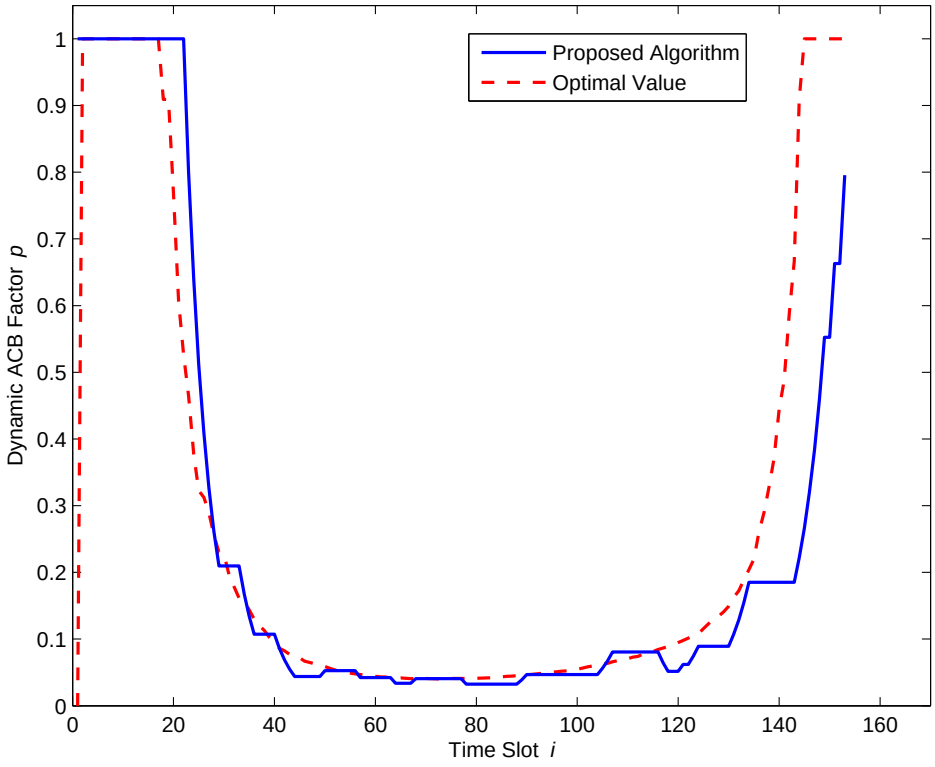

Figure 2.7: The dynamic ACB factor p vs number of time slots with $N = 1000$, $I_A = 100$, $M = 20$.

which validate our analytical model in Section 2.2. The best and the worst estimation scenarios are included to show the performance of our algorithm. As a reference, we also present the simulation results of a fixed ACB factor scenario, where p is a constant and set to be $\frac{M}{N}$. With increasing number of preambles allocated to M2M traffic, the total service time can be reduced as expected. As we can see, the performance of our heuristic algorithm is close to the optimal p scenario and much better than the fixed p scenario.

Fig. 2.7 shows how the dynamic ACB factor p in our proposed algorithm varies over time. In our algorithm, p is initially set to be 1. The algorithm will decrease p once it

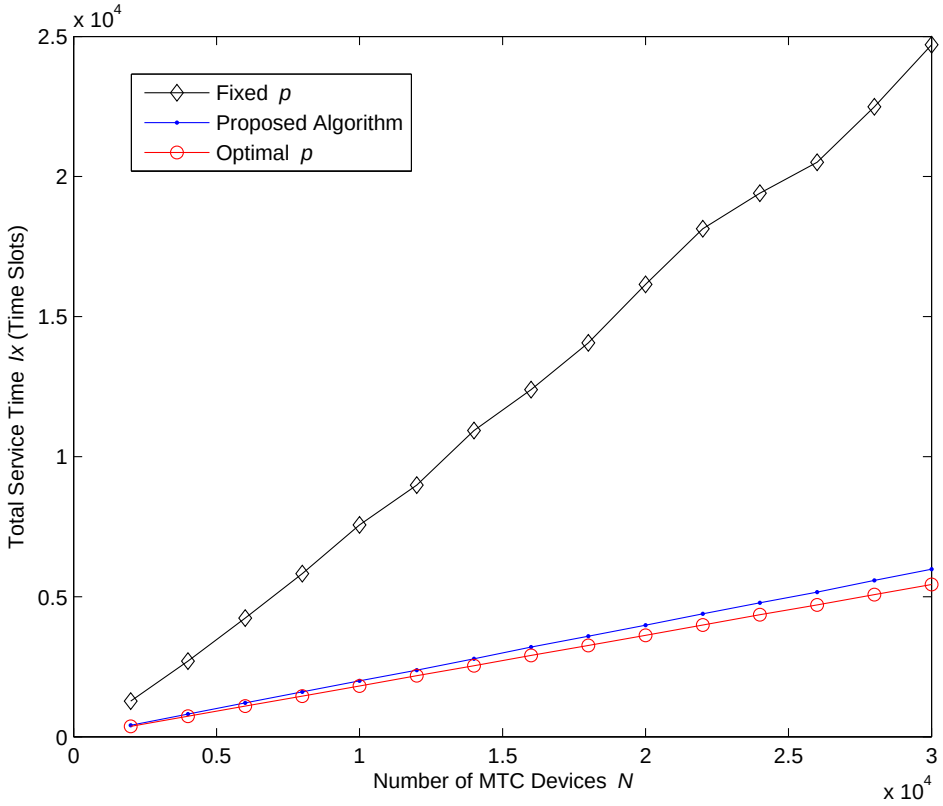

Figure 2.8: The total service time vs number of MTC devices N with $M = 15$, $I_A = 100$.

exceeds the optimal value, and increase p if it is less than the optimal value, but it will not go beyond 1 as p is a transmission probability. As can be seen in the figure, p in the proposed scheme fluctuate around the optimal value, which is the reason that the proposed algorithm can achieve near optimal results.

In reality, the number of M2M devices within a single cell can be significantly large. We vary the number of devices from 1000 up to 30000 in Fig. 2.8. Results show that our estimation can still achieve near optimal performance. Compared to the ACB with fixed factor, it yields much better performance in terms of reducing the total service time.

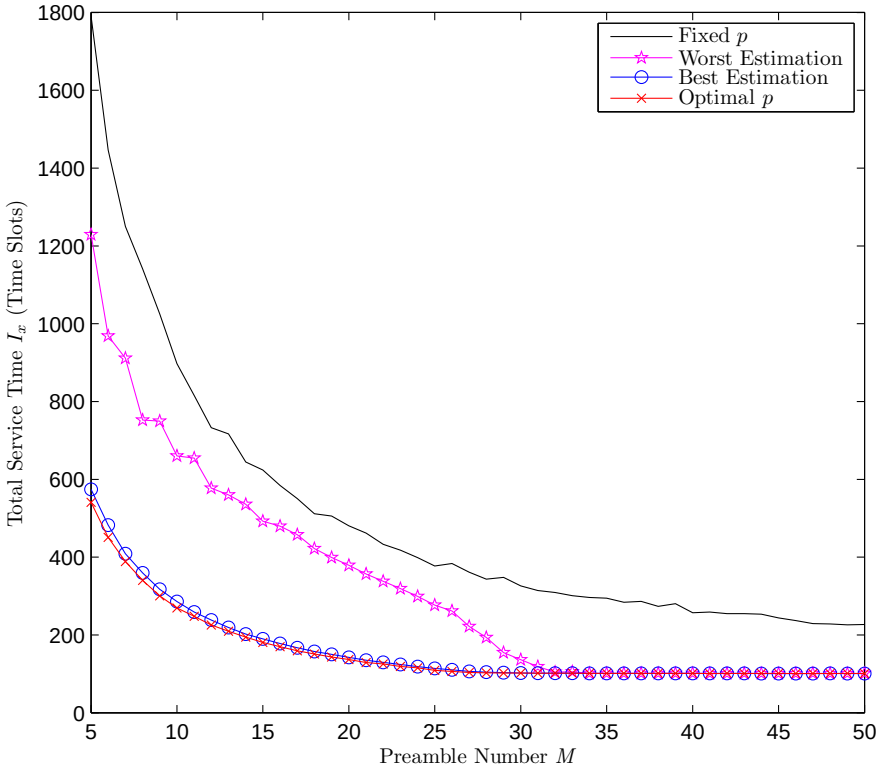

Figure 2.9: The total service time vs preamble number M with $N = 1000$ and $I_A = 100$ under uniform distribution.

This shows the scaling behavior of our algorithm.

As our model is not dependent on the activation model, the same parameters are applied to uniform distribution traffic model, *i.e.*, the activations of all the users are uniformly distributed within the activation time. This is also proposed in 3GPP standards [3]. The results are shown in Fig. 2.9. As can be seen, the algorithm works well under different traffic models.

2.5 Dynamic Allocating Preambles for M2M Traffic

In LTE networks, each cell has a limited number of 64 preambles for random access. These resources are shared by H2H UEs and MTC devices. We refer to one preamble in one RACH as one random access opportunity. For the fixed resource allocation scheme studied in previous sections, during each time slot, there are M random access opportunities allocated for M2M communications. The system allocates M opportunities in one time slot for I_X time slots, and the average number of opportunities per MTC device is $\frac{1}{N}MI_X$. Instead of dedicating a fixed number of preambles to M2M traffic, the system can possibly change the number of preambles available to MTC devices with time. We denote the number of preambles allocated for M2M traffic in time slot i as M_i. If the resources are dynamically allocated, the average number of opportunities per MTC device is then $\frac{1}{N}\sum_{i=1}^{I_X} M_i$. In this section, we discuss whether this value can be reduced by dynamic allocating preambles for M2M traffic.

The probability of a preamble being selected by exactly one user is derived in equation (2.20). When p is optimal, *i.e.*, $p = \frac{M}{n}$, we have

$$\begin{aligned}\mathbb{P}(D_m = 1 \mid N_i = n) =& \binom{n}{1}\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1} \\ =& \left(1-\frac{1}{n}\right)^{n-1}. \end{aligned} \tag{2.24}$$

When n approaches infinity, $\lim_{n\to\infty}\mathbb{P}(D_m = 1 \mid N_i = n) = e^{-1}$. For a total of M preambles, the expected number of successful transmissions is thus $\frac{M}{e}$. In other words, for a successful transmission, the system will have to provide e random access

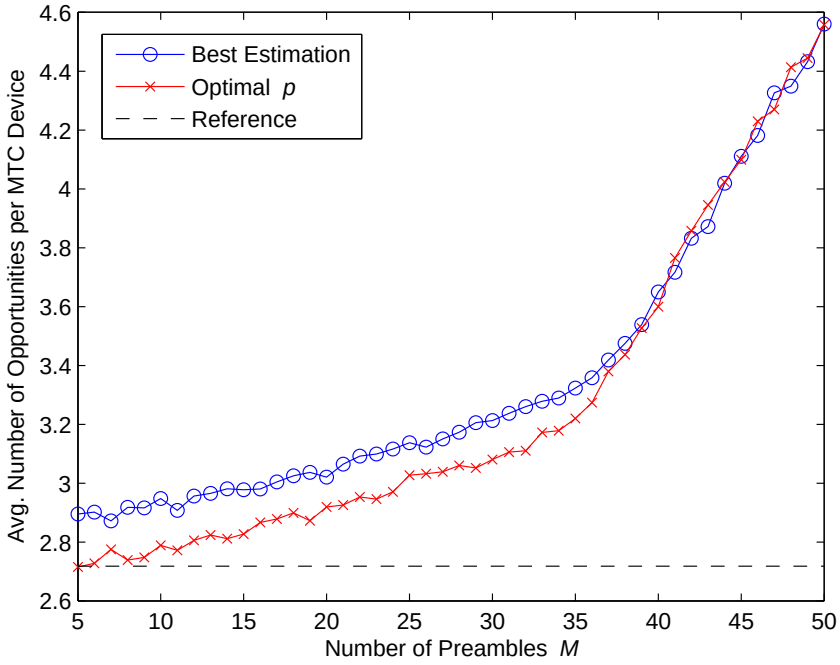

Figure 2.10: The average number of opportunities per MTC device vs the number of preambles with beta distribution activation model.

opportunities to realize this transmission during random access process where collisions exist (unlike scheduling transmission where one opportunity can realize one successful transmission). This value can be a reference line. We plotted the average number of opportunities per MTC device versus the number of preambles for the aforementioned fixed resource allocation schemes in Figs. 2.10 and 2.11.

There are three interesting results that can be obtained from these two figures:

1. In both figures, the average number of opportunities per MTC device go linearly up when M grows large (in Fig. 2.10, when $M > 37$, in Fig. 2.11, when $M > 27$). Further inspection shows that the slope of these two linear parts is proportional to

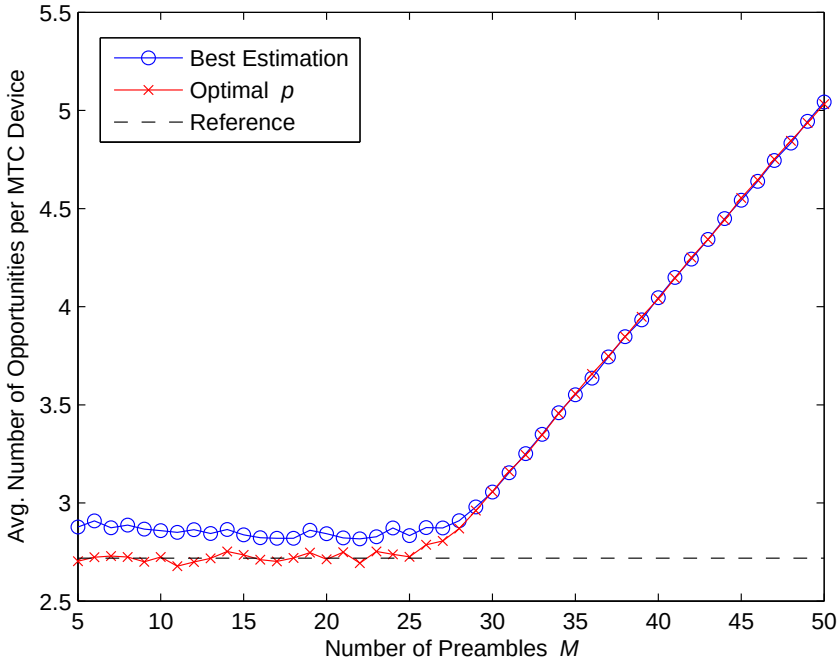

Figure 2.11: The average number of opportunities per MTC device vs the number of preambles with uniform distribution activation model.

the value of activation time. This means that the system is allocating too many preambles for M2M traffic and the system will stop as soon as all the devices have been activated. Within this range, if we increase the number of preambles for M2M traffic, the total service time will not be reduced as $I_X \approx I_A$, and the average number of opportunities per MTC device will be increased as shown in both figures. To reduce the average number of opportunities as well as achieve the minimum total service time, we should choose the start point, *i.e.*, $M = 37$ for beta distribution and $M = 27$ for uniform distribution. These two are both fixed resource allocation schemes.

2. The average number of opportunities increases with the number of preambles in beta distribution. This means that in order to reduce the average number of opportunities per MTC device, if there exists a dynamic resource allocation scheme, it will always choose the smallest number of preambles during every time slot, which becomes a fixed allocation scenario. This is a tradeoff between the average number of opportunities and the total service time, as with increasing number of preambles, the total service time decreases while the average number of opportunities increases. Based on QoS requirements and the resources available, the service provider can strike a balance between these two.

3. The average number of opportunities per MTC device remains roughly unchanged when M is small for uniform distribution. Within this range, changing the number of preambles will only result in different total service time. Still there is no need in a dynamic resource allocation scheme if we aim to reduce the average number of opportunities per MTC device. The number of preambles allocated for M2M traffic can be selected based on QoS requirements according to Fig. 2.9.

We assume that eNodeB knows the current number of backlogged MTC devices in the system. Under this assumption, we design a simple rule for eNodeB to update the number of preambles allocated for M2M traffic. The idea of dynamic resource allocation is to increase the number of preambles available for random access when the number of backlogged users is large and vice versa. In light of this, we make the number of

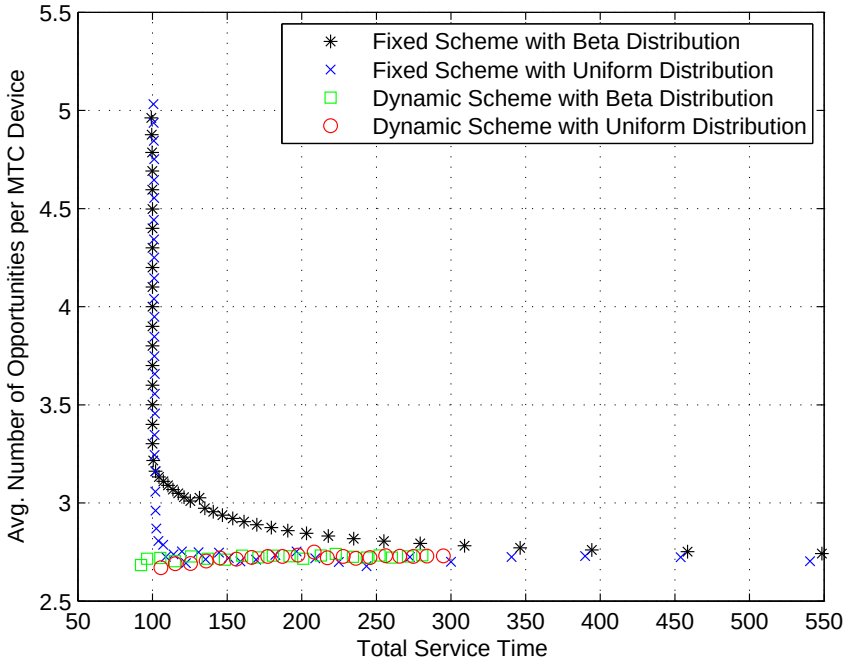

Figure 2.12: The average number of opportunities per MTC device vs total service time with uniform distribution activation model.

preambles, M_i, proportional to the number of backlogged devices, n:

$$M_i = \left\lceil \frac{n}{b} \right\rceil \tag{2.25}$$

in which b is a parameter that ranges from 1 to 20. The range of this parameters is estimated based on the number of MTC devices as well as activation time. The current range can ensure that M_i does not go beyond 64, which is limit of preambles available within a cell while providing as many preambles as possible. The number of preambles M_i is updated for each time slot. Similar to the optimal p scenario discussed in Section 2.2, this dynamic scheme cannot be realized in real environments as eNodeB has no information about the number of backlogged users. However, it can serve as a reference

on how good dynamic resource allocation schemes in general can possibly be.

We plot the average number of opportunities per MTC device versus the total service time in different settings in Fig. 2.12. We include the results of the fixed resource allocation scheme under beta distribution and uniform distribution. Each data point represents a scenario of a fixed number of preambles, ranging from 5 to 50. Simulation results under two traffic models are presented for the dynamic resource allocation scheme, in which each data point represents a different parameter b.

As can be seen in Fig. 2.12, if we aim to reduce only the average number of opportunities per MTC device, or the total service time, the dynamic allocation scheme does not have better performance than the fixed scheme. For beta distribution traffic arrival model, the fixed resource allocation scheme can achieve either the minimum average number of opportunities per MTC device or the minimum total service time, but not both at the same time. For uniform distribution, the fixed resource allocation scheme can achieve both at the same time as shown in the figure. The x value for this optimal point is I_A and the y value is e. According to previous analysis, I_A is the minimum total service time while e is the minimum average number of opportunities per MTC device. A good dynamic resource allocation scheme is only necessary if we aim to achieve both the minimum average number of opportunities per MTC device and the minimum total service time for beta distribution. Other than this, the fixed resource allocation scheme can achieve good performance in reducing the average number of opportunities per MTC device so as to save resources allocated for M2M communications.

2.6 Summary

In this chapter, we considered an overloaded M2M communication system. We presented how ACB factor can be dynamically updated to reduce the total service time. We started with the analytical model of an optimal case where eNodeB knows the number of backlogged users. Then, we proposed a heuristic algorithm where eNodeB updates the ACB factor adaptively based on the number of preambles with collision in previous time slots. Simulation results showed that our algorithm can achieve near optimal performance compared to the optimal case, and can greatly reduce the total service time compared to the scenario of fixed ACB factor. Then we studied the possibility of dynamic resource allocation for M2M communications. With simulation results, we showed that the fixed resource allocation scheme can minimize the average number of random access opportunities per MTC device and dynamic resource allocation schemes are not necessary.

Chapter 3

Conclusions and Future Work

3.1 Conclusions

In this thesis, we discussed a congestion control scheme for the bursty traffic scenario of M2M communications in LTE networks. We introduced a situation that in an emergency, eNodeB may wish to obtain message from all the MTC devices that are involved as soon as possible. As packets from these devices will arrive in a bursty pattern, congestion at RAN is unavoidable. A good scheme to perform congestion control is thus desirable.

As the envisioned M2M communication scenario uses cellular network as the radio access network, we first gave an introduction on the random access procedures in LTE networks. Then we modeled the system as a multi-packet reception system, and derived the transmission probability matrix. The matrix was used to track how the state vector evolves with time, and to obtain the expected minimum total service time, assuming that eNodeB is aware of the number of backlogged users in the system. Then we considered a more realistic scenario where eNodeB has no information regarding the number of backlogged users. We proposed a heuristic algorithm to adaptively update the transmission probability p, which yielded near optimal performance and huge reduction in the total

service time compared to the fixed p scenario. As the algorithm is independent of the packet arrival model, we used the same parameters on a different traffic model and still obtained close-to-optimal performance, which showed the robustness of our algorithm.

As the random access opportunities are limited resources for LTE networks, we investigated the problem of reducing the average number of random access opportunities per MTC device by dynamic allocating different number of preambles during each time slots. With analysis based on the results of simulations, we provided some insight into the system, and showed that dynamic allocation will not be able to reduce the average number of opportunities per MTC device. Fixed allocation schemes can achieve the minimum number of random access opportunities.

3.2 Future Work

In our system model, we only considered the case where each MTC device has one packet to transmit. In real systems, each MTC device may generate more than one packet, and instead of reducing the time for all the packets to be successfully transmitted, the target can be to reduce the time of successfully receiving a certain percentage of all the packets. Also, we used a p-persistent model as our access scheme. For future work, we can also consider binary backoff scheme, or even incorporating carrier sensing multiple access (CSMA) into M2M communications.

The original ACB scheme is two-fold. First it divides all the MTC devices into different QoS classes, and within each class, an ACB check will be performed before

a random access attempt can be transmitted. In our model, we treat all the MTC devices as one access class, and optimize congestion control under this assumption. If we can differentiate the QoS requirements of delay-tolerant and delay-sensitive devices and perform the ACB check separately, we will solve both congestion control problem and the QoS problem at the same time. By rejecting random access attempts from a certain class and setting different ACB factors for each class, more system resources will be released so that delay-sensitive devices will quickly finish transmissions to leave room for the devices that are barred. This aspect of joint consideration may well have interesting results on how system can quickly recover from a bursty load of traffic.

According to [3], the number of activations during each time slot under beta or uniform distribution traffic model is a constant instead of a random variable if the number of MTC devices and the length of activation time are given. This is different from traditional traffic model where during each time slot, the number of arrivals is a random variable. New traffic model may be considered as a possible extension, as long as the traffic model makes sense for M2M communications.

Bibliography

[1] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2M: From mobile to embedded Internet," *IEEE Comm. Magazine*, vol. 49, no. 4, pp. 36–43, Apr. 2011.

[2] Machina Research Sector Report, "Machine-to-Machine (M2M) communication in consumer electronics 2012-22," Feb. 2013.

[3] 3GPP, "Study on RAN Improvements for machine-type communications," 3rd Generation Partnership Project (3GPP), TR 37.868 V11.0.0, Oct. 2011.

[4] S.-Y. Lien, K.-C. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 66–74, Apr. 2011.

[5] A. Gotsis, A. Lioumpas, and A. Alexiou, "M2M scheduling over LTE: Challenges and new perspectives," *IEEE Vehicular Technology Magazine*, vol. 7, no. 3, pp. 34–39, Sep. 2012.

[6] 3GPP, "System improvements for machine-type communications," 3rd Generation Partnership Project (3GPP), TR 23.888 V11.0.0, Sep. 2012.

[7] K.-D. Lee, S. Kim, and B. Yi, "Throughput comparison of random access methods for M2M service over LTE networks," in *Proc. of IEEE GLOBECOM Workshops*, Houston, TX, Dec. 2011.

[8] C. Ide, B. Dusza, M. Putzke, C. Muller, and C. Wietfeld, "Influence of M2M communication on the physical resource utilization of LTE," in *Proc. of IEEE Wireless Telecommunications Symposium (WTS)*, London, United Kingdom, Apr. 2012.

[9] A. Bartoli, J. Hernandez-Serrano, M. Soriano, M. Dohler, A. Kountouris, and D. Barthel, "Secure lossless aggregation for smart grid M2M networks," in *Proc. of IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Gaithersburg, MD, Oct. 2010.

[10] C.-Y. Tu, C.-Y. Ho, and C.-Y. Huang, "Energy-efficient algorithms and evaluations for massive access management in cellular based machine to machine communications," in *Proc. of IEEE Vehicular Technology Conference (VTC-Fall)*, San Francisco, CA, Sep. 2011.

[11] C.-Y. Ho and C.-Y. Huang, "Energy-saving massive access control and resource allocation schemes for M2M communications in OFDMA cellular networks," *IEEE Wireless Communications Letters*, vol. 1, no. 3, pp. 209–212, Apr. 2012.

[12] K. Zhou and N. Nikaein, "Packet aggregation for machine type communications in LTE with random access channel," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Shanghai, China, Apr. 2013.

[13] K. S. Ko, M. J. Kim, K. Y. Bae, D. K. Sung, J. H. Kim, and J. Y. Ahn, "A novel random access for fixed-location machine-to-machine communications in OFDMA based systems," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1428–1431, Jul. 2012.

[14] 3GPP, "RACH overload solutions," 3rd Generation Partnership Project (3GPP), TSG RAN WG2 #70bis R2-103742, Jun. 2010.

[15] G. Wang, X. Zhong, S. Mei, and J. Wang, "An adaptive medium access control mechanism for cellular based machine to machine (M2M) communication," in *Proc. of IEEE International Conference on Wireless Information Technology and Systems (ICWITS)*, Hawaii, HI, Aug. 2010.

[16] S.-Y. Lien, T.-H. Liau, C.-Y. Kao, and K.-C. Chen, "Cooperative access class barring for machine-to-machine communications," *IEEE Trans. on Wireless Communications*, vol. 11, no. 1, pp. 27–32, Jan. 2012.

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

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

[19] H. Wu, C. Zhu, R. La, X. Liu, and Y. Zhang, "Fast adaptive S-ALOHA scheme for event-driven machine-to-machine communications," in *Proc. of IEEE Vehicular Technology Conference (VTC-Fall)*, Quebec City, Canada, Sep. 2012.

[20] S.-T. Sheu, C.-H. Chiu, Y.-C. Cheng, and K.-H. Kuo, "Self-adaptive persistent contention scheme for scheduling based machine type communications in LTE system," in *Proc. of International Conference on Selected Topics in Mobile and Wireless Networking (iCOST)*, Avignon, France, Jul. 2012.

[21] Y. Liu, C. Yuen, J. Chen, and X. Cao, "A scalable hybrid MAC protocol for massive M2M networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Shanghai, China, Apr. 2013.

[22] S.-Y. Lien and K.-C. Chen, "Massive access management for QoS guarantees in 3GPP machine-to-machine communications," *IEEE Communications Letters*, vol. 15, no. 3, pp. 311–313, Mar. 2011.

[23] A. G. Gotsis, A. S. Lioumpas, and A. Alexiou, "Analytical modelling and performance evaluation of realistic time-controlled M2M scheduling over LTE cellular networks," *Trans. on Emerging Telecommunications Technologies*, 2013.

[24] T. Kwon and J.-W. Choi, "Multi-group random access resource allocation for M2M devices in multicell systems," *IEEE Communications Letters*, vol. 16, no. 6, pp. 834–837, Jun. 2012.

[25] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home M2M networks: Architectures, standards, and QoS improvement," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 44–52, Apr. 2011.

[26] S. Sesia, I. Toufik, and M. Baker, *LTE–The UMTS Long Term Evolution: From Theory to Practice, 2nd edition.* Wiley, 2011.

[27] 3GPP, "[70bis#11]-LTE: MTC LTE simulations," 3rd Generation Partnership Project (3GPP), TSG RAN WG2 #71 R2-104663, Aug. 2010.

[28] A. K. Gupta and S. Nadarajah, *Handbook of Beta Distribution and Its Applications.* CRC Press, 2004.

[29] J. Riordan, *Introduction to Combinatorial Analysis.* John Wiley, 1959.