Signaling Congestion Control Mechanisms for Supporting Machine-to-Machine Communications in 4G LTE Cellular Networks

Aiswarya Prasannakumar

A Thesis Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

Department of Computer Science and Engineering

June 2013

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

Aiswarya

(Signature)

(Aiswarya Prasannakumar)

CS11M08

(Roll No.)

Approval Sheet

This Thesis entitled Signalling Congestion Control Mechanisms for Supporting Machine-to-Machine Communications in 4G LTE Cellular Networks by Aiswarya Prasannakumar is approved for the degree of Master of Technology from IIT Hyderabad

(KIRAN KUCHI 15th July 2013)

(———-) Examiner
Dept. of Electrical Engineering
IIT Hyderabad

Kotaro Kataoka July 10th, 2013

(———-) Examiner
Dept. Computer Science and Engineering
IIT Hyderabad

B Bheemarjuna Reddy 10/7/13

(Dr Bheemarjuna Reddy Tamma) Adviser
Dept. of Computer Science and Engineering
IIT Hyderabad

C. Krishna Mohan 10/7/2013

(———) Chairman
Dept. of Computer Science and Engineering
IIT Hyderabad

Acknowledgements

Many individuals contributed in many different ways to the completion of this thesis. I am deeply grateful for their support, and thankful for the unique chances they offered me. I am greatly thankful to my superviser Dr. Bheemarjuna Reddy Tamma for his valuable guidance, constant encouragement and timely suggestions. I would like to make a special mention of the excellent facility provided by my institute IIT Hyderabad. I would like to thank Mukesh Kumar Giluka for the support and the guidance he provided. More than to anyone else, I owe to the love and support of my family and friends. My father P.N Prasannakumar, my mother V.K.Rajini, my sister Aswini, my brother Aswin, my husband Manu Mohan and my friends Saranya, Sushmitha, Sharu, Gandherva, Rahul and Prerana.

Abstract

Typically, in Machine-to-Machine (M2M) communication system, an M2M device (sensor, meter, etc.) captures an event (pressure, inventory level, etc.), which is relayed through a communication network (wireless, wired or hybrid, cellular) to an application (software program), that translates the captured event into meaningful information without any human intervention. Based on this concept, several noble applications have been developed such as environment monitoring, smart grid, e-healthcare, and fleet management which will affect various aspects of our life.

Because of ubiquitous coverage and global connectivity, cellular networks are playing a major role in the deployment of M2M applications. But, presently cellular networks are optimized for Human-to-Human (H2H) and Human-to-Machine (H2M) communications and in future also, these seem to be uninterrupted because of the percentage of revenue contributed by these applications. At the same time, characteristics of H2H/H2M are quite different from that of M2M. In comparison to H2H/H2M, M2M applications have low traffic volume, high uplink to downlink traffic ratio, larger density of devices in a particular geographical area, and limited mobility of devices. With these differences, supporting M2M applications in cellular networks such as 4G LTE networks is very challenging to the telecom network operators. One of major challenges is efficient allocation of radio resources between M2M applications and H2H/H2M applications. Another major challenge is signaling congestion due to presence of too many M2M devices in any given cell area and simultaneous contention by them for radio resources. Apart from this, some other issues are also exist for supporting M2M communication such as standardization issues, addressing and naming issues, and privacy and security issues.

In this thesis, we propose solutions to address this second challenge. When an M2M device tries to connect to network after switched on or it wants to uplink synchronized with the base station (eNodeB) or it wants to do handover to another eNodeB, the device goes through random access channel (RACH) procedure in 4G LTE cellular networks. Due to presence of too many M2M devices, there is a high probability for several M2M devices performing RACH procedure at the same time and causing congestion in the network. Several congestion handling methods have been proposed by 3GPP and others to solve this problem. Push based methods were proposed in which the random access attempt was done based on a probability which causes unnecessary delay if the load on the network is low. Pull based methods rely on the eNodeB to send a signal before the device can perform random access request. This is not feasible in case of M2M devices as there are a huge number of M2M devices trying to access the network. In this work, we have proposed a new congestion handling method called Numbering Scheme (NS) which spreads the access of M2M devices so as to avoid signaling congestion. This is a proactive method which does not wait till network overload occurs to spread the access. Simulation results show that NS perform better than other algorithms when the load on the network is low to medium.

Among various congestion handling method proposed in the literature, some are best suitable when there is less congestion in the network while some are efficient in high congestion scenarios. But, there exists is no single method which is able to efficiently handle signaling congestion in all scenarios. So, we have defined a congestion management function which will estimate the congestion in the network and based on the level of congestion, it will suggest most suitable congestion handling method to be applied. NS is also used as a congestion handling method in congestion management function. Simulation results show that congestion management function performs better than any single congestion handling function and it performs equally well in case of low and high congestion.

When the signaling congestion in the network becomes too high for the eNodeB to handle, it will bar some M2M devices from accessing the network. 3GPP has proposed, Access Class Barring (ACB) method in which the devices are divided into 10 low priority access classes. When congestion level is high, eNodeB will bar some access classes (AC) from accessing the network. This will cause all the devices in that class to be barred even if the device is not delay tolerant. To overcome this problem, 3GPP specified a new barring method called Extended Access barring (EAB). In this method, all the devices which can wait for the congestion to reduce are assigned a status called EAB. When congestion is high all devices configured for EAB will be barred from accessing the network. Once the congestion goes down eNodeB will stop EAB and all the barred devices will access the network. This causes a sudden spike in number of devices accessing the network. We propose a randomized approach called EAB spike removal to reduce this spike to prevent the network from being congested again. Simulation results have shown that this method reduces the congestion after EAB and increases the number of users succeeding in RACH procedure.

Contents

Chapter 1

Introduction

1.1 Introduction to M2M

Machine-to-machine (M2M) communication [4, 17] is an emerging communication concept where the goal of networking can be realized, fully or partially, with limited or no human intervention. The main motivation behind M2M communication is based on the observation [18] that after enabling communication between multiple machines through an underlying network, several applications were found which does not need any human participation. Due to this characteristic, M2M communication is becoming a market changing force for a wide variety of applications. Some of the M2M applications are smart grid, e-healthcare, smart homes, environmental monitoring, industrial automation, etc. According to the researchers [18, 19], by the end of 2014, 1.5 billion devices and by the end of 2020, 20 billion devices will be part of M2M communication. According to 3GPP specifications [20, 21, 22, 23], M2M communication can also be termed as Machine Type Communication (MTC). In the thesis, we have used the terms MTC and M2M interchangeably.

Because of ubiquitous coverage and global connectivity, cellular networks are playing a major role in the deployment of M2M applications. Cellular networks can be used in two different ways for supporting M2M communications:(a) an M2M device having embedded cellular radio sends data directly to the M2M server located on Internet through a cellular network. (b) an M2M device having embedded wi-Fi or ZigBee radio first sends data to an M2M gateway and then gateway forwards data (typically after aggregation) to the M2M server through a cellular network. Former is called as cellular M2M communication while later is called as capillary M2M communication [18]. In capillary M2M, gateway needs to have dual radios for supporting connection to cellular and Wi-Fi/ZigBee networks. In the thesis, we will deal with only cellular M2M communication scenario. In this thesis the terms device and node are used interchangeably.

Presently, cellular networks are optimized for Human-to- Human (H2H) and Human-to-Machine (H2M) communications and in future also, these seem to be uninterrupted because of the percentage of revenue contributed by these applications for telecom network operators. At the same time, characteristics of H2H/H2M is different from that of M2M. In comparison to H2H/H2M, M2M has low traffic volume, high uplink to downlink traffic ratio, larger density of devices in a particular geographical area, and limited mobility of devices. With these differences, supporting M2M in cellular networks is a big challenge to the telecom network operators. We have discussed these

challenges in chapter 2.

1.1.1 IoT, WSNs, M2M and CPS

The term Internet of Things (IOT) was first proposed [27] in 1999, which refers to the interconnection of each uniquely addressable object. Wireless sensor networks (WSN) [28], M2M and Cyber Physical Systems (CPS) [11] are progressive stages, through which the whole concept and aim of the IOT can be realized. WSN is the basic scenario of the IOT, which consists of spatially distributed sensors to monitor some environmental or physical condition and sending their sensed data to a particular location with the cooperation of each other. CPS [21] is a system featuring a tight combination of, and coordination between, the systems computational and physical elements. Basically, CPS focuses [18] on intelligentizing interaction, interactive applications, and even distributed real-time control. This correlation can be summarized as follows:

- WSNs, M2M and CPS are part of IOT
- WSN is the foundation stage of IOT
- WSN works as a supplement for M2M

1.1.2 M2M Applications

M2M applications can be broadly classified into six categories [27] :

- fleet management
- asset tracking
- building security
- Smart Grid
- E-healthcare

Contribution of these applications in present M2M market [7] are 51% , 18% ,14% , 9% , 6% and 2%, respectively. In this section, we will discuss some of these applications in detail.

Smart Grids

The concept of M2M communication can be utilised to use the power in electric grid efficiently. A smart capability electric grid or smart grid (SG) [13] will be knowing the requirements and capabilities of power consumers, distributors and providers. Figure 1.1 [25] shows a general model of smart grid. In the figure, the part of power generation to power consumption is same as legacy electric grid. But, in SG, to optimize the electrical power generation and/or distribution, home appliances and other electric power consuming devices send their power consumption and demand status periodically to a control centre through home area network (HAN), neighborhood area network (NAN) and wide area network (WAN). A HAN is established among all the home appliances and the smart meter in home. The home appliances send their power consumption and demand status data to smart meter using ZigBee or power line communication (PLC). Similarly, a NAN is established

among smart meters of the houses in an area and the concentrator or gateway. Smart meters send the collected measurements to concentrator using Wi-Fi. Now, the concentrator sends the collected data from smart meters to control centre using WAN. The control center receives data packets for processing and storage. This data is used to optimize the electrical power generation and/or distribution. For example, now, the SG knows the exact requirement of power in a particular area while in case of legacy electric grid, it will distribute a fixed amount of power which may be more or less for a particular area, which will incur extra maintenance cost in either of the cases. The M2M communication in SG must be private and secure since many of the autonomic functions that will run over it will be critical. If an attacker gets control of smart grid system then he can, not only black out in a particular area but also it can affect the privacy of a particular SG user. HAN, MAN and WAN are depicted in the figure.

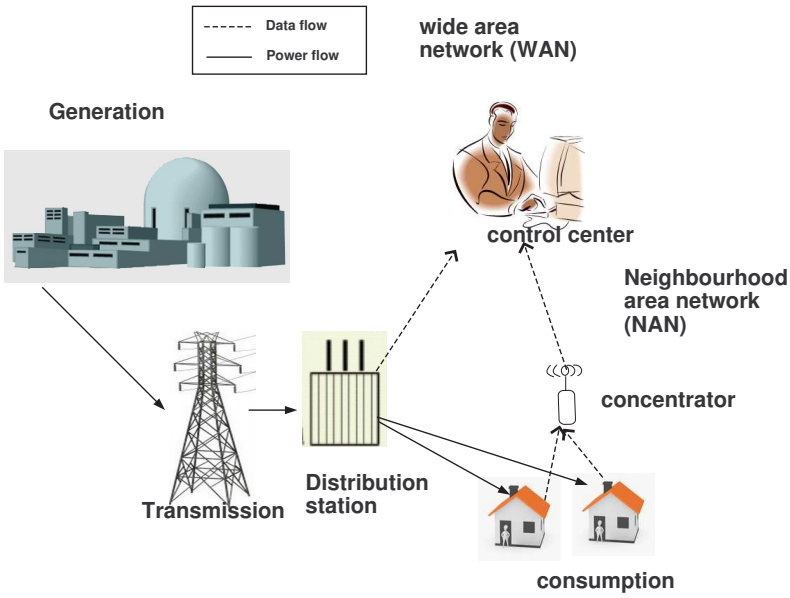

Figure 1.1: An example of Smart Grid system [25]

E-Healthcare

In E-Healthcare, body sensors are used to monitor the health condition of a patient for example, heartbeat, pulse rate, blood pressure and to inform about the monitored information to the patient as well as to the doctors in case of some abnormal condition. The M2M healthcare enables the monitoring of entire population in real-time. This provides the following advantages

1. Ambulance can be dispatched on time.

2. The patient can be monitored in home as if he is in hospital.

3. Monitoring particular symptoms on the population, the progression of a virus outbreak can be tracked.

Home Networking

A home network [26] is composed of various smaller home device sub-networks. Each sub-network will consists of an aggregator which will aggregate the data sent by the devices of the sub-network for example, electric appliances, laptop, printer etc and will send the aggregated data to a Internet gateway (router). Examples of such sub-networks are Zigbee sub-networks (electrical appliances, air conditioner (AC)), Wi-Fi sub-networks (laptop, printer, and media server), UWB sub-networks (HDTV, camcorder), smart grid sub-networks (smart meters, smart thermostat, smart switch), body area sub-network (smart phone, monitoring instrument, body sensors) and Bluetooth sub-network (music centre, portable audio player).

Home devices belonging to same application establish a subnetwork and each subnetwork has a sub-gateway (SGW) [18]. All SGWs are logically implemented on cognitive gateway. The cognitive gateway also consists of a home gateway (HGW). The HGW and SGW are logically separated. HGW manages the whole network and connects the home network to the outside world (e.g., the Internet). The network related functionalities are implemented in the HGW including access control, security management, QoS management, and multimedia conversion.

1.1.3 M2M Architecture

A typical architecture of M2M enables an M2M device to communicate with a M2M server situated at very far from it. The European Telecommunication Standards Institute (ETSI) [16] has proposed the architecture of M2M communication which consists of following components:

1. Device Domain or M2M area Domain: This component consists of numerous smart M2M devices equipped with sensors. These M2M devices are used for various purposes depending upon the application for which they are used. For example, in environment monitoring application, they will sense some environmental condition such as temperature, pressure, in asset tracking system, they will keep track of location of the vehicle on which the M2M sensor devices are planted. After this, they will send these sensed or measured data to a Gateway. The network established among M2M devices and the gateway is called M2M area network. The technologies used in the M2M area network are Zigbee, Bluetooth, Femtocell,and etc.

2. Network Domain: It provides the network for transmission of M2M sensory data from M2M area domain to Application domain. The network technologies involved in this domain are xDSL, WLAN, satellite, GSM, GPRS, CDMA2000, worldwide interoperability for microwave access (WiMAX), LTE, LTE-A, and etc.

3. Application Domain: The back-end server is an important component for the whole M2M paradigm, which collects all sensory data delivered to it. Based on the information gathered by the server, after processing of these data, it will take some intelligent decision. For example, in case of intelligent transportation system, the server will extract all the information from various global position systems (GPS) data. Based on the information, server will decide the traffic condition in the city and send alert messages to the vehicles which are registered for this application so that they can follow the less congested traffic path.

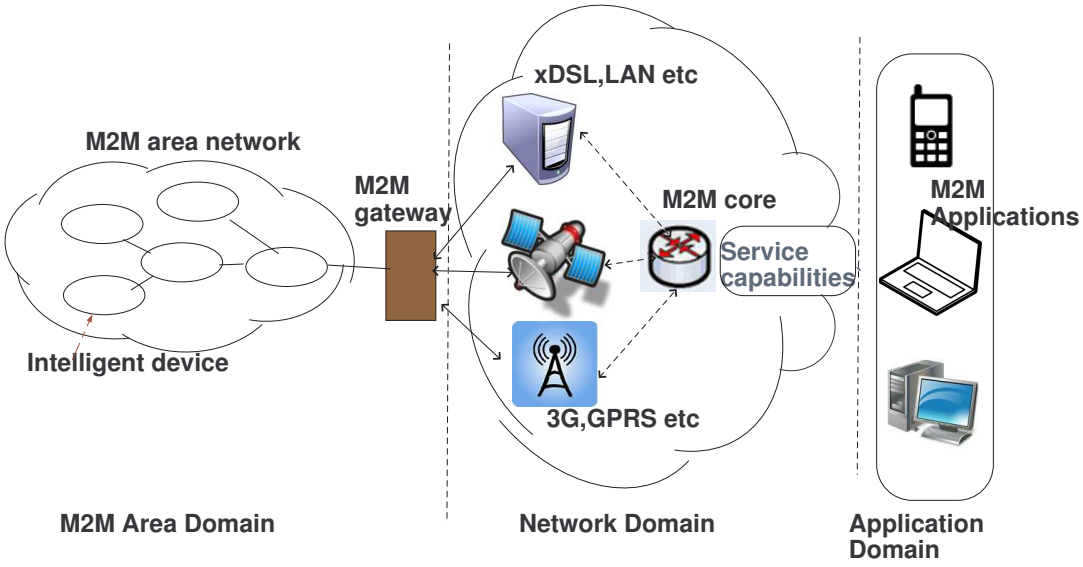

Figure 1.2: M2M Architecture

Based on the technologies used in M2M area network, ETSI has defined the term capillary M2M (short-range system) and cellular M2M (long range system). If the communication between the M2M devices and the M2M gateway is through WSN, then it is called capillary M2M and if the communication is cellular, then it is called cellular M2M. Capillary M2M and cellular M2M will likely coexist until (almost) full migration to the cellular system has been achieved.

1.2 Introduction to 4G LTE Cellular Networks

1.2.1 Overview

LTE is a standard for wireless communication of high-speed data for mobile phones and data terminals. 2G is the first digital mobiel systems which was for voice data. 2.5G used EDGE and GPRS for a data rate of 50 kbps. 3G was the first mobile system handling broadband data [**?**]. It used WCDMA and CDMA 2000 to get a data rate of 384 kbps. Then 3.5G came into picture with datarate of 5-30 Mbps. LTE provides a data rate of 100 Mbps. LTE was proposed by 3GPP. It can have a bandwidth of 1.25 MHz to 20 MHz. This is achieved using Orthogonal Frequency Division Multiplexing (OFDMA). This is used in downlink (sending data from the base station to the node). It uses single carrier FDMA for uplink (from the node to the base station). LTE-Advanced (LTE-A) provides a data rate of upto 300 Mbps (downlink) and 150 Mbps for uplink. LTE architecture is shown in figure 1.3. The components of LTE are:

- **UE**: User Equipment is the mobile handset.
- **MME**: Mobility Management Entity. It is a control-plane node. It is responsible for handling securit keys, handling of IDLE-ACTIVE transitions, handover and establishment and release

of a connection.

- **P-GW**:Packet Data Network Gateway (PDN-GW/P-GW) is used to connect to the Internet. It allocates IP address and ensures Quality of Service (QoS).
- **eNodeB**: It is known as evolved Node B. It communicates directly with UEs. It is similar to a base station in 2G. It makes the scheduling decisions for the UEs. It is also responsible for handover from one eNodeB to another.
- **S-GW**: Serving gateway acts as a mobility anchor when terminals move between eNodeBs.

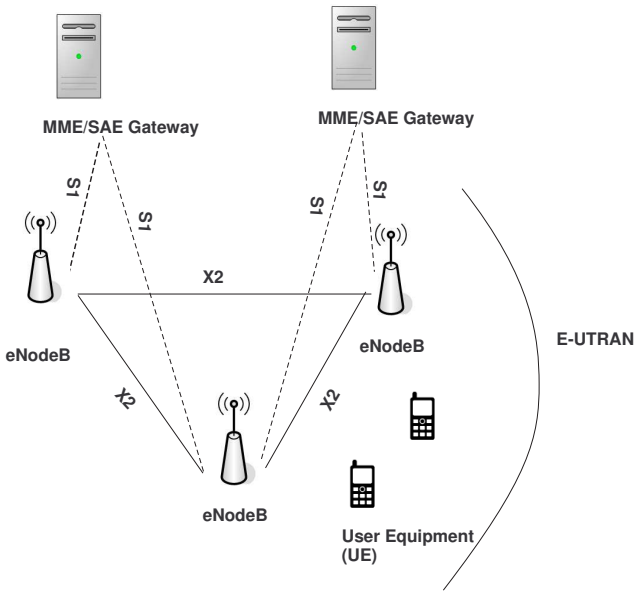

Figure 1.3: LTE Architecure

1.2.2 LTE Frame Format

LTE has two frame formats, Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD). In LTE TDD same frequency is used for uplink transmission as well as downlink transmission. So uplink and downlink slots come one after the other. Seperate frequencies are used for uplink and downlink in LTE FDD. Uplink and downlink transmissions are divided into frames of duration 10 ms. Each frame is further subdivided into ten subframes. Each subframe is of 1 ms. Each frame has 2 slots each of 0.5 ms each. So there are totally 20 slots in a frame as shown in Figure 1.4. The MAC scheduler schedules resource blocks once in every 1 ms. Bandwidth of LTE is about 20 MHz which will be around 100 resource blocks (RB). Resource blocks correspond to a timefrequency unit of 1 ms times 180 kHz [**?**]. Each RB has a bandwidth of 180 kHz with 12 sub carriers and seven OFDM symbols. A total of 84 resource elements are available in one RB. Total number of bits in an RB depends on the modulation scheme. In uplink SC-FDMA is used. In downlink OFDMA is used. Each frame carries the control information and data information in both uplink and downlink. The channels used in downlink transmission are PDCCH (Physical Downlink Control Channel) and

PDSCH (Physical Downlink Shared Channel). Similarly for uplink transmission PUCCH (Physical Uplink Control Channel) and PUSCH (Physical Uplink Shared Channel) channels are used. A UE can be allocated a minimum of one RB.

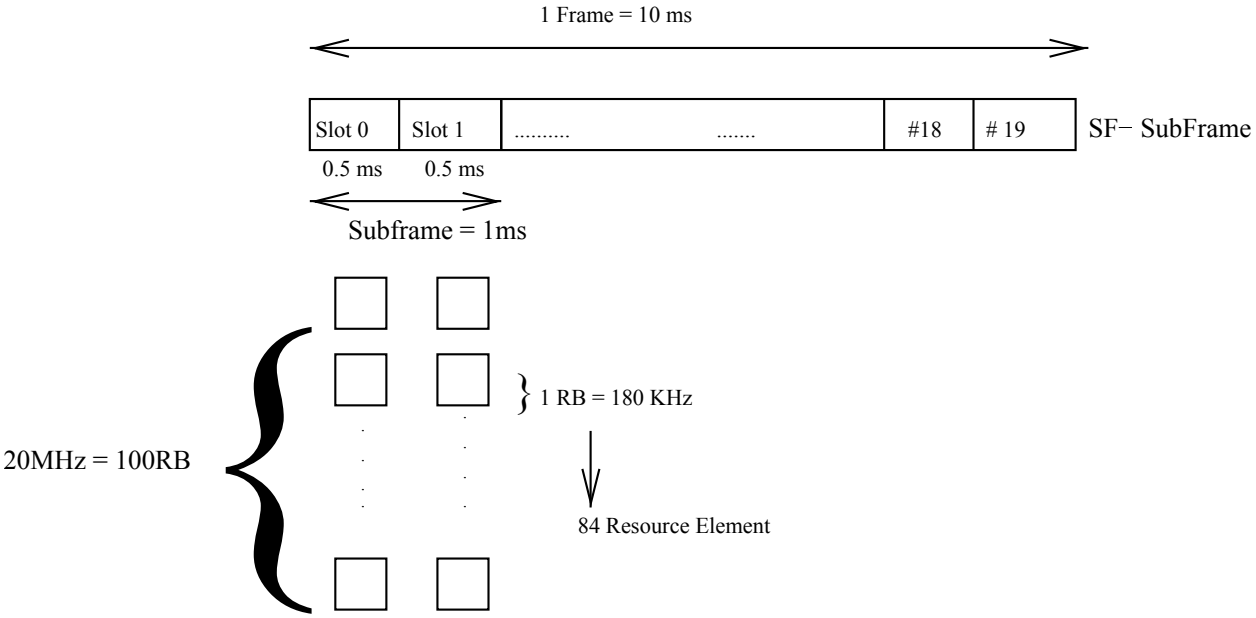

Figure 1.4: LTE Frame Format

1.2.3 PRACH and EAB

When a device is switched on, it will try to find out in which cell it currently is in and then will try to synchronize with the cell. The process of getting the cell information is known as cell search.

Cell Search

In cell search, a device will acquire the frame timing of the cell, that is, the start of the downlink frame. It needs the identity of the cell and should synchronize with the cell. The eNodeB broadcasts two synchronization signals for enabling this.
Primary Synchronization Symbol (PSS): From PSS device or UE can find the

1. Five millisecond timing of the cell
2. Position of Secondary Synchronization Symbol (SSS)
3. Cell identity within cell group

Secondary Synchronization Symbol (SSS): From SSS device or UE can find the

1. Frame timing
2. Cell identity group

MIB (Master Information Block)

To actually access the network the device will require system information which is boradcasted by the cell. System information consists of Master Information Block (MIB) and System Infomation Block (SIB). MIB is broadcasted in the BCH (Broadcast channel). It contains information which will enable the device to read the information provided in DL-SCH. It consists of information of the cell bandwidth, System Frame number, etc.

SIB (System Information Block)

This contains the detailed system information. SIB is transmitted in DL-SCH. There are several SIBs each containing different system information. SIB1 contains information of the location of other SIBs and what kind of restrictions the operator has put on the cell. SIB2 contains the information required for performing random access. Following are the parameters whose values are broadcasted using SIB2:

- *number of RA-Preambles*
- *sizeOfRA-PreamblesGroupA*
- *preambleTransmissionMax*
- *ra-ResponseWindowSize*
- *PreambleInitialReceivedTargetPower*
- *PowerRampingStep*
- *MaxHARQ-Msg3Tx*
- *MACContentionResolutionTimer*
- *rootSequenceIndex*
- *PRACHConfigurationIndex*
- *zeroCorrelationZoneConfig*
- *PRACHFreqOffset*

The parameter number of RA-Preambles gives the total number of contention based preambles which can be used by the devices. The sizeOfRA-PreamblesGroupA gives the maximum size of the message which can be considered as a group A message. Preambles can be further grouped into group A and B based on the size of the message sent. When a random access request fails a preamble is retransmitted. The preambleTransmissionMax is the maximum number of times a preamble can be retransmitted by the device. Once number of preamble attempts by the device reaches this maximum, it will report to the higher layers that random access request has failed. The parameter ra-responseWindowSize is the number of frames a device will check if it gets a response from eNodeB for its random access request. The parameter MaxHARQ-Msg3Tx is number of times HARQ (Hybrid ARQ) will happen for terminal identification message. rootSequenceIndex gives the index of the root sequence which is used in random access request. Other SIBs contain information

about cell reselection, name of home eNodeB, public warning messages, uplink cell bandwidth, neighbouring cell related information, parameters for uplink power control, etc.
SIBs are mapped to system informations (SIs). SIB1 is always mapped to SI-1 and the remaining can be grouped together and mapped to different SIs. Since SIB1 contains details of where to look for other SIBs this is sent more frequently than other SIBs. It is always sent in subframe 5.

Random Access

Once the device receives PSS, SSS, MIB and SIB, it will be downlink synchronized with the cell. But the eNodeB does not know about the existence of this particular device. If the device wants to request data from the eNodeB or if it wants to send some data, it needs to be uplink synchronized with the eNodeB. That is, it should know when the uplink frame starts. So, the device will start Random Access Procedure for connection setup. In this procedure, the device will send a random access preamble to eNodeB and then if the eNodeB is able to decode the message it will send a random access reponse. Then the device will send its identity to the eNodeB. Once the eNodeB receives this, it will acknowledge the device with a contention resolution message.
Random Access procedure is used in the following cases.

- When a device is switched on
- For uplink synchronization
- During handover for uplink synchronization with new cell
- Downlink data arrival
- Positioning

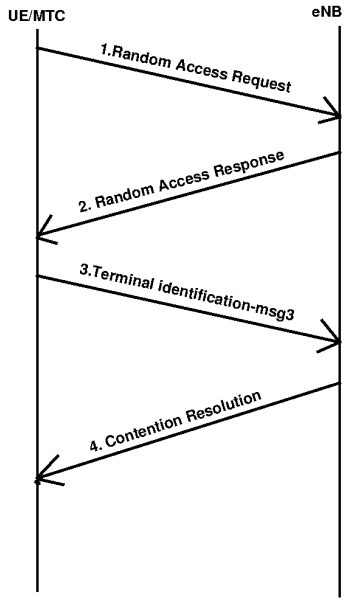

Figure 1.5: Random Access Procedure

Random access procedure is shown in the figure 1.5 *Random Access Preamble Selection:*
For sending random access request, the device should select a random access preamble. There are a maximum of 64 different random access preambles per cell. The device will select one of this at random. Random access preamble consists of a preamble sequence. A preamble sequence is derived from a root sequence. This root sequence is obtained by reading the root sequence index value sent by the eNodeB in SIB2. The root sequence is a zadoff-Chu sequence. The preamble sequences are generated from this root sequence by cyclically shifting the root sequence. The property of preambles made from cyclically shifted zadoff-Chu sequence is that they have zero cross-correlation property. So, ideally there is no interference between them. The eNodeB can obtain accurate timing estimate from them because of cyclic auto-correlation. We can create a maximum of 64 sequences from a root sequence. If the cell size is less than 1.5 km then we can use a single root sequence to make all the 64 preambles. For larger cells more than one root sequence is needed. The number of sequences which can be generated from a root sequence is also broadcasted by the eNodeB in SIB2 in zero correlation zone configuration. For example, suppose the root sequence index is 12 and the zero correlation zone configuration is 12. It means that we can have 7 preambles per root sequence. So we need a total of 10 root sequences. So the root sequence indices from 12-22 will be used by this particular cell.

Random Access Preamble Transmission:
This is the step where contention can happen. Random access preamble is sent in a special random access channel called Physical Random Access Channel (PRACH). This channel consists of 6 RBs. There are different preamble formats. In configuration 0, duration is 1 ms: 0.1 ms for the cyclic prefix, 0.8 ms for the zadoff-chu sequence and 0.1 ms is guard period. Other configurations can go upto 3 ms. The guard period is used because the device is not yet uplink synchronized with eNodeB. By keeping the guard period we can make sure that the random access request does not overlap with actual data transmission which happens after the PRACH slot. Different configurations are used depending upon the cell size. When a device has data to send, it will randomly select a preamble. The preamble can be selected from group A or group B based on the size of message which is to be sent. If the size is less than sizeOfRA-PreamblesGroupA parameter from SIB2 then the device will select a preamble from group A otherwise, it will select a preamble from group B. Then the device will wait for a PRACH slot and then send the selected preamble in that slot. It will set the PREAMBLE_TRANSMISSION_COUNTER to 1. The device will then wait for a response from the eNodeB in PDCCH (Physical Downlink Control Channel). The response window starts from three subframes after the end of preamble transmission till ra-ResponseWindowSize subframes.

Random Access Response:
If the eNodeB successfully receives the random access request it will send a response. But multiple devices can select the same preamble which can collide or some preambles can get lost as well. It is also possible that two devices select the same preamble at the same time and the eNodeB receives one of them before the other and processes the first one. This can happen because of the lack of uplink timing synchronization. Random access response can contain a backoff indication subheader. This happens when there is congestion in the core network. If the device receives a backoff it will set its backoff parameter value to the value given in it. Otherwise the backoff parameter value is set to 0 ms. If the response is not a backoff it will contain the following items:

- Index of the preamble recevied
- Timing advance for timing correction
- TC-RNTI temporary identity for the device with which the device should send the response from now on
- Scheduling grant using which the device can send message 3

Terminal Identification:
If the device does not receive the random access resposne within the ra-ResponseWindowSize then it will increment the PREAMBLE_TRANSMISSION_COUNTER by 1 and if it is greater than the *preambleTransmissionMax*, then it will inform the higher layers that it failed in the random access. Then the device will backoff by taking a random backoff value between 0 and the backoff parameter value. If the device receives a random access response with its preamble index then it will send the following information:

- C-RNTI of the device if it had connected to the network before
- Core network terminal identifier if this is the first time the device connects to this cell

It will be send in UL-SCH(Uplink shared channel) in the scheduling grant received in the previous step. This is possible because the device is now uplink synchronized with eNodeB. Once *message 3* is transmitted the device will start *mac-ContentionResolutionTimer*. In this message HARQ (Hybrid ARQ) is there.

Contention Resolution:
When multiple devices try to send using the same preamble and eNodeB accepts one of them then all the devices which used the same preamble index will think that it is the one which was accepted. For example, suppose two devices send preamble sequence index 3 and eNodeB received device 1's preamble request successfully, then the eNodeB will send a random access response with preamble sequence index as 3. Both the devices will accept this random access response because it contains the preamble sequence index which the device sent. Now device 1 and 2 will send message 3 and the eNodeB will accept only the correct one because only that device is properly synchronized with eNodeB. So the eNodeB will send a response containing device 1's information. The device 2 does not receive the information and it backs off. This step is called contention resolution. The eNodeB will send the response in PDCCH (Physical Downlink Control Channel). If the device receives the response containing its information then it will stop its mac-ContentionResolutionTimer as it has successfully completed its Random Access Procedure. If the device does not have a C-RNTI, it will set TC-RNTI value to C-RNTI and discard TC-RNTI. If the device does not receive the response before the mac-ContentionResolutionTimer expires or its information is not there in any of the messages then it will understand that its random access procedure has failed and then it will backoff by taking a random backoff value between 0 and the backoff parameter value.

There are two types of random access

- Contention free

- Contention based

Contention free Random Access: In contention free random access the device will do random access request with a pre-assigned random access preamble. This happens when the eNodeB already knows about the device and knows that device needs to do random access. It happens in cases like handover, downlink data arrival etc. In case of handover the previous eNodeB knows that the device needs to be handed over to the neighbouring eNodeB. Then it will provide the random access preamble to be used by the device with the new eNodeB. The new eNodeB also knows about this device from the old eNodeB. So when a device sends with this preamble the eNodeB identifies it as that particular device and it will provide timing advance for uplink synchronization. The device need not send its C-RNTI to the new eNodeB as it already knows those details. The only thing which is needed by the device before communicating with that eNodeB is the timing synchronization. That is obtained by doing random access procedure. So there is no need of steps 3 and 4 (terminal identification and contention resolution) as there is no contention. Random access procedure ends with first 2 steps. In case of downlink data arrival the eNodeB informs the device about the data arrival and then gives information for random access procedure. The device then does random access procedure to be uplink synchronized with the eNodeB. Once the synchronization is done then it will proceed to receive data. Figure 1.6 shows the different types of preambles.

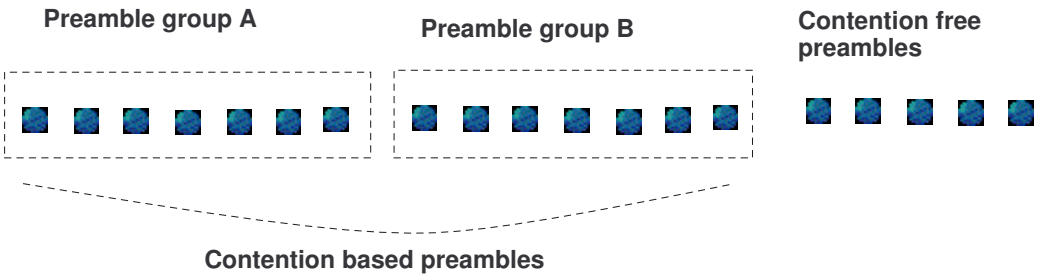

Figure 1.6: Types of preambles

Contention based Random Access:
Contention based random access is used when a device is switched on or when it is not uplink synchronized with the eNodeB. Here the devices will randomly select a preamble from the available set of preambles and then send it. Multiple devices can send using the same preamble sequence causing collision. When the load on the network is very high that is, a large number of devices are accessing the network at the same time, these collisions can be really high. Suppose the load on the network is very high and say 100 devices are trying to access the network at the same time. Even in the ideal case there will be many collisions. Even when the load on the network is low it is possible that two or more devices can choose the same preamble at the same time.

Extended Access Barring (EAB)

Sometimes congestion becomes so high that no device is able to access the network. When the congestion becomes too much for the eNodeB to handle then the eNodeB will start EAB. In EAB

device which are configured for EAB will be barred from accessing the network. Low priority devices or delay tolerant devices can be configured for EAB. The eNodeB will set a flag to inform that EAB has started and this will be broadcasted. When a device which is configured for EAB gets data to send it will check if EAB is on. If it is on then it will send data only after the EAB is turned off. This decreases the number of devices trying to do random access at the same time thereby decreasing congestion. When the number of devices accessing the network at the same time decreases, the success rate increases and more devices succeed in accessing the network. Once congestion becomes less, EAB will be turned off and then all the devices which were barred can access the network.

1.2.4 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 explains about issues and challenges in M2M communications. Chapter 3 gives the literature survey. Chapter 4 explains the proposed signaling congestion control mechanisms, Chapter 5 contains simulation setup and results and finally Chapter 6 contains the conclusions and future directions.

Chapter 2

M2M Issues and Challenges

Since M2M communication is using various existing technologies such as WSN and cellular networks, so the challenges inherent to these technologies will also be present in M2M. On the other hand, since the existing technologies and frameworks are designed for H2H or H2M communication, so to make them compatible for M2M also, we will have to face several challenges. In this chapter, we will discuss various technical and non-technical challenges:

2.0.5 Standardisation Issues

Since, the scope of M2M is very versatile, several standards have been under development depending on the application. We may get several standards even for a single application, for example, Smart Grid. But we do not know which standard will be good for long time. So, interoperability of the standards is required and it is one of the challenges. There are various standard organizations working on this issue such as 3GPP, ETSI, IEEE, and telecommunications industry association (TIA).

2.0.6 Addressing and Naming Issues

In M2M, several times it is required of device to device communication or server to device communication. In this case, a specific address of each device will be required. The rate of growth of M2M and the proliferation of M2M devices shows that IPv4 will not be sufficient to address all such devices. So, other option is IPv6 in which more than 10^{38} devices can be addressed which is enough to identify any object in this universe uniquely. But, the address size of IPv6 is of 128 bits i.e. 16 bytes and generally, size of payload in an M2M data packet is of 5-10 bytes. So, size of header and data is of same order which is very inefficient because it will enforce a large overhead and congestion in the network. At the same time, processing of these packets on an M2M node will incur more power consumption. 6LoWPAN [31] is working for integration of low-power IEEE 802.15.4 devices into IPv6 networks.

2.0.7 Networking Issues

There are several topics that come under this issue viz. congestion in the network due to M2M traffic, scheduling of LTE network to accommodate M2M, traffic characteristics of M2M, etc. M2M

uses existing cellular network and PLMN (public landline mobile network) network in its network domain. But, these networks are specially designed for H2H, H2M and M2H communications. So, it is a big challenge for network designers of M2M communication to make the existing communication networks accommodable for M2M. Traffic characteristics [13] of M2M are different from existing H2H communication. Since, the existing communication networks are mainly designed for H2H communication, so they should be given the first priority to use the network resources. Keeping this in mind, the network operators will have to face following challenges to deploy M2M [7]:

1. H2H will generate high volume traffic while M2M will generate low volume traffic. At the same time, the ratio of uplink to downlink traffic volume is more in M2M compared to H2H. So, network operators need careful spectrum allocation and management to avoid contention between low volume, uplink-heavy M2M traffic and high volume, downlink-heavy H2H traffic.

2. In most of the cases, H2H and M2M devices are located in the same geographical area which can cause congestion in the radio access network (RAN) part (device domain part). Therefore, careful network resource allocation will be required to handle the contention of M2M traffic and H2H traffic. On the other hand, M2M devices which belong to same application may be part of different device domains. So, they may not cause congestion in the RAN part but they will combinely cause congestion in the core network.

3. Priority of M2M traffic is less than H2H traffic but there are various real time M2M applications which need high priority over H2H traffic.

2.0.8 Security, Privacy and Safety

It is one of the critical challenge for M2M service provider. We can realize the seriousness of this challenge by following examples:

1. When we connect billions of devices then we basically provide billions of entry points for so many malicious attacks. So, if anybody, hacks the system once, say smart grid system, the hacker can access utility systems and therefore he can hack smart meters in a particular area. Consequently, he can disconnect the power supply of that particular area. Similarly, one of the implemented capabilities in smart grid systems is remote firmware upgrader. Using this concept, it is possible to install a malware function remotely by an attacker in the smart meters. This malware may not allow any other firmware upgrader to be installed remotely by some legal authorities. So, in this case, the service provider will have to go door to door to change the malware or firmware. These are the examples of security and denial of service issues.

2. If an attacker has an access to a smart meter, he can trace all the activities of a user for example, when he gets up, when he sleeps, when he goes out of the home and when he comes back etc. So, privacy of a user will be compromised.

3. Often, M2M devices spend most of the time unattended; and thus, it is easy to physically attack them.

4. Most of the communications are wireless, which makes eavesdropping extremely simple.

5. M2M devices are characterized by low capabilities in terms of both energy and computing resources; therefore, they cannot implement more complex cryptographic and other algorithms to ensure the security.

2.0.9 Other issues

1. Since, to deploy M2M architecture, billions of M2M devices and other components will be involved. Cost of the devices and the cost of deployment of the architecture will directly affect the consumer.

2. Various M2M applications are there in which M2M devices needed to be placed in such a remote area where frequent access for power replacement will not be possible. Further, 3G or 4G modules are very expensive in terms of power.

3. Some M2M devices may face extreme environmental condition e.g very low temperature or very high temperature. In such environmental condition, the devices may not work.

4. Reliability and performance of network is required when devices are communicating with each other to take decision.

5. When technology is available, then huge amount of devices will also be available. So, e-waste management challenges will also be there.

6. Policy issues will also be there. For example, who will be responsible if an attacker does some damage purposely.

Chapter 3

Literature Survey

We need efficient congestion control mechanisms so that introduction of M2M devices to cellular networks do not cause them to collapse. Following are the existing congestion control mechanisms from the literature.

3.1 Survey on Congestion Management Methods

When large number of devices try to access the network at the same time signaling congestion can occur. This is because we have 64 preambles in a PRACH slot and if many devices try to access the network they can choose the same preamble as another device and this will lead to collision which may result in failure of that preamble attempt. Such devices will then backoff and then try to access the network at a later time. This leads to delays and more congestion at a later stage. Several methods have been proposed to solve the problem of signaling congestion in LTE RAN. These methods can be classified as follows :

3.1.1 Push Based Solutions

In push based methods [8, 1], the UE or MTC device will initiate the RACH procedure. Push based methods are:

- p-persistent approach
- Pre-Backoff approach
- Backoff Indicator Adjustment approach
- Randomization Indicator
- Wait Timer Adjustment
- Maximum number of preamble transmission adjustment

p-persistent approach

In this approach, the UE will not send the random access preamble request message immediately but it will send with probability p. By doing this, contention can be avoided up to some extent. The problem with this method is that even when congestion is not there M2M device will send only data at a particular time with some probability. This causes unnecessary delay.

Pre-Backoff approach

In this approach, when a device has data to send it will read the RAR PDUs (Random Access Response Protocol Data Units) which were meant from some other devices and then will check the backoffs of those devices and determine the state of the network. The device will then backoff based on the backoff provided in those RAR PDUs. This happens even before the first attempt is made to send the preamble. This will spread the initial access attempt as well [10]. This method is only applied for devices which are delay tolerant in nature. If the device does not see any backoff in the RAR PDU checked it assumes that there is no congestion in the network at this point of time and will access the network immediately without backing off. After the backoff is done it will try to access the network.

Backoff Indicator Adjustment approach

In this approach, the backoff time for MTC devices is kept quite long so that more UE/MTC devices can be served before a particular device start resending the preamble request. The backoff time can be quite long and it can be longer than the maximum value provided using backoff index [9]. The maximum backoff value is now 960 ms. We should keep backoff which is more than that i.e. several seconds. One way of implementing this is to include an additional backoff indicator in the RAR PDU along with the existing Backoff Indicator (BI) in the MAC header. This additional field can be read only by M2M devices which are configured for the long backoff i.e. delay tolerant devices. We need both these backoff fields because the legacy devices do not know about long backoff and they can use original BI field and the delay tolerant devices will use the new field. The eNodeB does not know if it is replying to a delay tolerant device or a normal device. So eNodeB puts both backoff indicators [10]. A disadvantage of this approach is that it can cause delay for sending M2M device's data even when the network is not in a congested state.

Randomization Indicator

In this method, the RAN or CN (Core Network) will broadcast a randomization indicator. If the indicator is set, the M2M device which is restricted will backoff. The device accesses the network only after the backoff timer expires [11]. RAN needs to broadcast an additional parameter in this method. This will cause additional overhead.

Wait Timer Adjustment

In this method, we add a new parameter called wait timer. It is the amount of time the device has to wait after it fails to receive RAR, msg3 and 4. The device does not send any information to the eNodeB at this time [1].

Maximum number of preamble transmission adjustment

In this method, the maximum number of preamble transmission is adjusted to see if success rate of random access procedure can be increased by increasing the preambleTransmissionMax [5] parameter.

3.1.2 Pull based Solutions

In pull based methods, eNodeB will initiate the RACH procedure. Since the eNodeB is controlling the access of devices congestion will not occur.

Contention free random access procedure

Contention free approach is one of the pull based methods which is generally used at the time of handover, positioning etc. In this approach, the eNB will assign a contention free preamble to UE [1]. UE will send the random access request with this contention free preamble. These methods can only be used in special cases like handover, downlink data arrival etc. and not during normal random access procedure.

Paging

A device will only access the network when the paging message sent by the eNodeB contains its identifier. In this method, extra control channel resources are required for paging. This is not feasible when there are large number of M2M devices in a cell.

3.1.3 Dedicated Resource allocation solutions

In this concept, the random access slots are separated between M2M and UE devices. In this way UEs are not affected by the M2M traffic.

Slotted Access

In this method, each random access opportunity is defined as an access slot [8]. Each M2M device will access the network only in its assigned slot. The device is associated with the access slot with the help of its identifier.

Virtual Resource Allocation

M2M devices are divided in to many classes based on the type of service and its priority. A random access slot is termed as a virtual resource. Different amount of virtual resources are pre-assigned to different type of M2M device classes based on it priority. Emergency and scheduled traffic have the same virtual resource in this scheme. Different number of slots are allocated to different classes of devices. For example, low priority traffic will have less slots assigned to it than high priority traffic [2].

Self-Adaptive Persistent Contention Scheme

This scheme proposes that if a device does successful RACH procedure then probability of success in next RACH procedure will be increased if the previous preamble is used again. This scheme consists of two phases - contention phase and compact phase. In this scheme separate slots for M2M and UEs are there. In contention phase after a contention is successful, M2M device will memorize the successful preamble sequence, contention frame used and contention order to use it in the next access attempt [3]. In compact phase, when eNodeB realizes that the M2M devices are stable it reduces the number of M2M specific random access resources.

Separated PRACH slot allocation approach

In separated PRACH slot allocation approach, PRACH resources are allocated either to UE devices or M2M devices [1]. One method is to seperate the set of available preambles into two subsets where one is given to UE and another to M2M devices. Another method is to give one set of preambles only for UEs and the next set for both M2M and UE. If the number of UE and MTC devices coming is bursty this method can cause delay when load on the network is less.

3.1.4 Code Expanded Random Access

In this method users will send preamble in multiple random access subframes creating access codewords that are used for contention. So with the same number of preambles and subframes the amount of contention resources is increased thereby allowing more users to do random access at the same time [12].

3.1.5 Access Barring methods

In access barring methods certain devices are barred from accessing the network. In case of high load on the network, eNodeB will decide to bar devices which have lower priority and are more delay tolerant.

Access Class Barring (ACB)

In access class barring approach, MTC devices are classified into different access classes (AC).There are a total of 16 access classes. Class 15 is for PLMN staff, class 14 for emergency services, class 13 for public utilities like water,gas etc, class 12 for security services and class 11 is for PLMN use. Normal devices come under classes 0 to 9 [14]. Single or multiple access classes are barred in case of a possibility of congestion. The number of classes to be barred are decided on the basis of the amount of congestion in the network at that point of time. When ACB is in effect this information is broadcasted in SIB2 [15]. Any device which wants to access will check if it is barred by reading this SIB and then will send data if it is not barred. In access class barring the MME (Mobility Management Entity) will send OVERLOAD_START signal. This signal is used for informing the eNodeB that there is congestion in the network and the eNodeB is asked to perform access barring to reduce congestion. On receiving the signal the eNodeB will broadcast the access barring time as well as barring factor. The device will then select a random number between zero and 1. If the generated value is greater than barring factor, the device will try to access the network [13].

Otherwise it will be barred for (0.7 + 0.6* rand)*BarringTime. Another method was proposed to differentiate M2M devices and UEs. A lower threshold is kept for starting access barring for M2M devices. So ACB will start for M2M devices first and if the congestion is not reduced even after that then only UEs get barred. So the MME will send an OVERLOAD_START signal for M2M devices alone. Once eNodeB receives this message, it will send barring messages for M2M devices alone.

Extended Access Barring (EAB)

When the load on the network becomes too high eNodeB can bar some devices. These devices are normally delay tolerant. EAB is an overload protection mechanism by which devices which are configured for EAB are barred from accessing the network. When the MME/eNodeB finds that congestion is there in the network it will initiate barring. This barring information can be set either as a single flag or a separate flag for each access class. This method is derived from the legacy ACB. The difference is that in legacy ACB when a class is barred everyone in that class is barred from accessing the network. But in case of EAB only those devices which are configured for EAB will be barred. For example, if access class 3 is barred by EAB all the devices which are configured for EAB in access class 3 will be barred, all the others will be able to access the network. If all EAB configured devices are barred then the access will be as shown in Figure 3.1. When a device tries to access the network the following actions are done:

1. It checks the "targeted subcategory" to see if it belongs to the group which are targeted for potential barring
2. It will check if its access class is authorized
3. If it is not authorized then it will be barred for some time and after that it will again check if it is barred [32]

There are 3 categories of UEs which are configured for EAB [34].

- Category A: UEs that are configured for EAB
- Category B: UEs configured for EAB which are not in their HPLMN or equivalent PLMN
- Category C: UEs configured for EAB which are neither in the most preferred PLMN or in their HPLMN or equivalent PLMN

One bit per access class is configured for category A for more granular control and one bit for categories B and C [33].

EAB information update: EAB information can be broadcasted in existing SIBs or a new SIB can be defined for it. If we use an existing SIB for EAB information the information update procedure is subject to SI modification period [35]. If the EAB has to be fast enough to react to the changes in the network this will be a bottleneck. If we allow the EAB information to change during the SI period the legacy UEs will be affected as it uses HARQ with soft combining. If we use a new SIB the SI modification period will not cause problems as legacy UEs will ignore this broadcast. For fast EAB update we can perform paging. We can either perform paging for EAB update at every

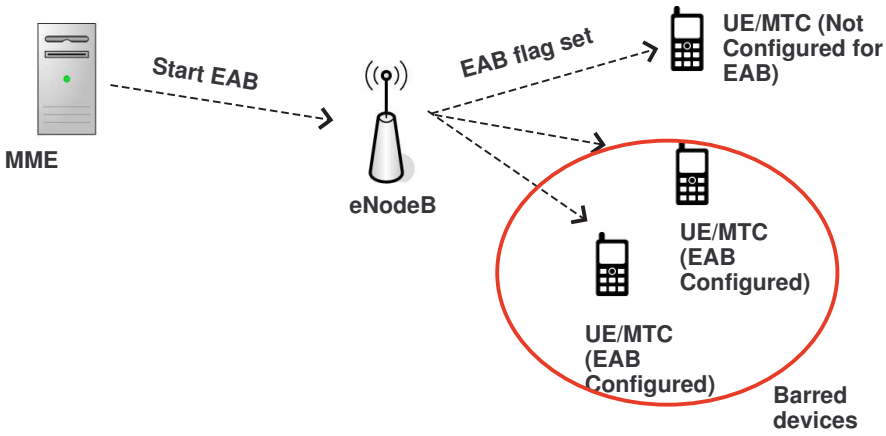

Figure 3.1: Extended Access Barring

paging opportunity or at a particular paging opportunity. In the second method we use an additional paging opportunity for EAB update [36]. All the devices will listen to this paging occasion to get the EAB update. The problem with this approach is that the node needs to wake up at this time to listen to this update [37]. In the first method it checks any paging update for the information.

Problems with EAB When the EAB is turned on, all the devices configured for EAB will be barred from accessing the network. When EAB is turned off all the devices who are barred will come up and try to access the network resulting in a sudden peak in usage. This causes congestion in the network. So we need to find some mechanisms to distribute access of the barred devices once the EAB is over.

1. Randomization Approach: In this method each device which was barred will take a random time and then backoff for that time. Once the backoff is over the device will try to access the network. In this way the sudden spike caused due to barred devices from accessing the network can be avoided. The problem with this approach is that we just take a random backoff and spread the access. If the number of devices getting unbarred are less then this method will cause unneccesary delay in accessing the network. But, if the number of devices getting unbarrred after EAB is tunred off is more then there will still be some congestion because the spread in access is not enough.

2. Unbarring one class at a time: In this approach when congestion goes down the eNodeB will unbar one class at a time [15]. In this way the number of devices accessing the network per slot will decrease thereby reducing the spike. The problem with this approach is that we might cause unnecessary delay for devices which belong to the class which was unbarred last. It is also possible that there might be more devices from one class and if all of them are unbarred together it might cause some congestion.

Randomization Approach

In this method each device which was barred will take a random time and then backoff for that time. Once the backoff is over the device will try to access the network. In this way the sudden spike caused due to barred devices from accessing the network can be avoided. The problem with this approach is that we just take a random backoff and spread the access. If the number of devices getting unbarred are less then this method will cause unneccesary delay in accessing the network. But, if the number of devices getting unbarrred after EAB is tunred off is more then there will still be some congestion because the spread in access is not enough.

Unbarring one class at a time

In this approach when congestion goes down the eNodeB will unbar one class at a time [15]. In this way the number of devices accessing the network per slot will decrease thereby reducing the spike. The problem with this approach is that we might cause unnecessary delay for devices which belong to the class which was unbarred last. It is also possible that there might be more devices from one class and if all of them are unbarred together it might cause some congestion.

Chapter 4

Proposed Work

4.1 PRACH solution

Various methods were discussed to handle the congestion in previous chapter. Some methods avoids congestion up to some extent but as soon as number of devices accessing the network increases, their performance degrade. In our proposed work, we classify the signaling congestion level into three categories: (i) *No Congestion Scenario*, when there is very less congestion or no congestion. (ii) *Moderate congestion scenario*, when there is a good amount of congestion (iii) Access barring or *Extreme congestion scenario*, when network is unable to handle the congestion and we need to bar devices. In this thesis, we propose NS to address signaling congestion in LTE due to M2M.

4.1.1 Numbering Scheme

We propose a new randomized access dispersion based congestion handling method which is more suitable than existing congestion handling methods for *No Congestion Scenario* and *Moderate congestion scenario*. In the proposed method, Numbering Scheme (NS), when a M2M device does successful RACH procedure, the eNB assigns a number between 0 to N. If a device is assigned a number K then on its next network access, it will send the random access request on K^{th} PRACH slot from the time of activation of the device. In M2M communication, in general, devices do not access the network continuously, for example, periodic monitoring. So each time M2M devices access the network, RACH procedure is needed to be done for uplink synchronization with eNB. If the device fails in its K^{th} PRACH slot, then it will backoff and will send the random access request on $2K^{th}$ PRACH slot. If the device fails even after that it will no longer wait for the next K^{th} PRACH slot. It will follow the normal backoff scheme. Here, the value of N is chosen by the eNB in such a way that simultaneous access of a number of devices is spread across some PRACH slots to reduce contention. At the same time average access delay is also controlled. An example of numbering scheme is shown in figure 4.1. Here when the device with number 3 becomes active, it will wait for 3 PRACH slots and then it will try to access the network.

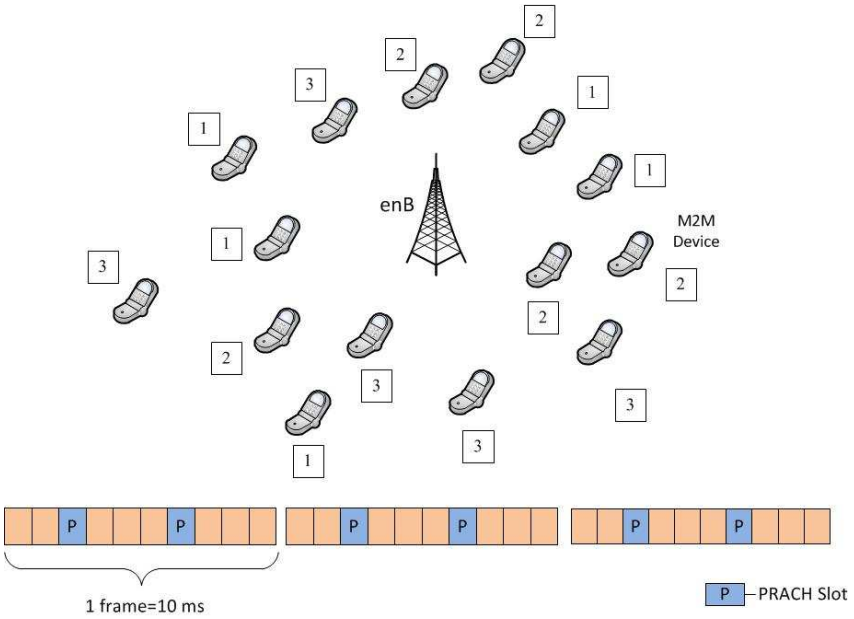

Figure 4.1: Numbering Scheme

4.1.2 Congestion Management Function

To find out the suitability of NS for *No Congestion Scenario* or *Moderate Congestion Scenario* over other methods, we propose Best congestion handling method selection algorithm. In this algorithm, performance of different congestion handling methods like p-persistent approach (Section 3.1.1), slotted access (section 3.1.3) and backoff indicator adjustment approach (Section 3.1.1) are estimated for a particular condition of congestion in the network. The method which sustains the network for longer amount of time even if congestion increases, is most suitable for that level of congestion in the network.
After successful preamble transmission, each device informs eNB through MSG3 about how many times it backed off before succeeding. Based on this information, in Best congestion handling method selection algorithm, the eNB calculates the average backoff b per device at a particular time instant. If value of b is greater than a threshold value (λ) then there is a possibility of high or low congestion depending upon the value of λ. Now, one of the congestion handling methods are applied. After applying the method, average success rate in a PRACH slot will increase and b will decrease. But, as the number of devices accessing the network increases, average success rate decreases and b increases. Here, we measure the time t_1 elapsed to reach the success rate again same as before applying the method. Similarly, measure the time (t_2) taken to reach the average backoff equal to λ. Here, we run the algorithm with different congestion handling method. The method with maximum t_1 and t_2 value is most suitable congestion handling method.

We propose an algorithm by which the eNB can suggest the most suitable congestion handling method to be used by all M2M devices for RACH procedure based on the condition of congestion in the network. With the help of algorithm 1, most suitable congestion handling method can be found out for both *No Congestion Scenario* and *Moderate Congestion scenario*. In the next Chapter,

Algorithm 1 Best Congestion Handling Method Selection

Input: Congestion handling methods, Start time, λ
Output: t_1, t_2

1: $N = 1000$ {Number of devices present in the cell}
2: **if** *current_time* $<$ *start_time* **then**
3: wait
4: **else**
5: find b {Average backoff calculated by eNB after receiving MSG3 by all the devices}
6: **if** $b > \lambda$ **then**
7: find S_1 {Average success rate before applying the algorithm}
8: Apply the algorithm
9: find S_2 and b_2 after next PRACH slot. {S_2 is average success rate after applying the algorithm and b_2 is average backoff calculated by eNB after receiving MSG3 by all the devices}
10: start timer t_1 and t_2
11: **if** $S_1 \geq S_2$ *or* $b_2 > \lambda$ **then**
12: Return $t_1 = -1$ and $t_2 = -1$
13: **else**
14: $N = N + 1000$
15: find S_2 and b_2 after next PRACH slot.
16: **if** $S_1 \geq S_2$ *and timer* t_1 *is not stopped* **then**
17: stop the timer t_1
18: **end if**
19: **if** $b_2 \geq \lambda$ *and timer* t_2 *is not stopped* **then**
20: stop the timer t_2
21: **end if**
22: **if** both timer stopped **then**
23: return t_1 and t_2
24: **else**
25: go to step 14
26: **end if**
27: **end if**
28: **else**
29: $N = N + 1000$
30: go to step 5
31: **end if**
32: **end if**

experimental results are analyzed to decide the suitable methods for *No Congestion Scenario* and *Moderate Congestion Scenario*.

Congestion Management Function is the main algorithm in which eNB decides which congestion estimation method is to be chosen based on the level of congestion in the network. In algorithm 2, we have used λ_1 and λ_2 as thresholds of *No Congestion Scenario* and *Moderate Congestion Scenario*. So, if the number of times backoff done by a device before successful preamble transmission is less than λ_1 then the device falls under *No Congestion Scenario* but if it is greater than λ_1 and less than λ_2 then the device falls under moderate congestion scenario otherwise it falls under extreme congestion scenario.

At a time instant, to estimate that the congestion scenario the network is in, we first need to calculate number of devices falling under different scenarios. In the algorithm, to denote number of devices falling under *No Congestion Scenario, Moderate Congestion Scenario* and *Extreme Congestion Sce-*

Algorithm 2 Congestion Management Function

Input: $tolerance\_factor$

1: start timer
2: $devScen1 = 0$, $devScen2 = 0$,$devScen3 = 0$; {number of devices existing in No Congestion Scenario,2 and 3}
3: $prevScen = NONE$;
4: **if** new device successfully connects to eNB **then**
5: **if** $newdevice.backoff \leq \lambda_1$ **then**
6: $devScen1 = devScen1 + 1$;
7: **end if**
8: **if** $\lambda_1 < newdevice.backoff \leq \lambda_2$ **then**
9: $devScen2 = devScen2 + 1$;
10: **end if**
11: **if** $newdevice.backoff > \lambda_2$ **then**
12: $devScen3 = devScen3 + 1$;
13: **end if**
14: **end if**
15: **if** timer expired **then**
16: $congScen$ =CongestionEstimationSubroutine ($devScen1$,$devScen2$,$devScen3$,$prevScen$, $tolerance\_factor$)
17: **if** $congScen == scenario1$ and $prevScen \neq scenario1$ **then**
18: $prevScen = scenario1$
19: broadcast the information to M2M devices to start congestion handling method for *No Congestion Scenario*
20: **else if** $congScen == scenario2$ and $prevScen \neq scenario2$ **then**
21: $prevScen = scenario2$
22: broadcast the information to M2M devices to start congestion handling method for *Moderate Congestion Scenario*
23: **else**
24: $prevScen = scenario3$
25: broadcast the information to M2M devices to start EAB
26: **end if**
27: go to step 1
28: **else**
29: go to step 3
30: **end if**

nario, we used variables $devScen1$, $devScen2$ and $devScen3$ respectively. The eNB calls *Congestion Estimation Subroutine* and passes these values as parameters. Algorithm 3 estimates the congestion scenario of the network. To do this, it finds that value of which variable among $devScen1$, $devScen2$ and $devScen3$ is $tolerance\_factor$% greater than other two. The $tolerance\_factor$ is passed as a parameter from the main algorithm to this subroutine. If that variable is $devScen1$ then the subroutine returns *No Congestion Scenario* and if it is $devScen2$ then subroutine returns *Moderate Congestion Scenario* otherwise *Extreme Congestion Scenario*. The main algorithm passes a parameter called *prevScen* i.e. previous scenario to the congestion estimation subroutine. This parameter denotes the scenario under which network is falling at the time of calling of the subroutine by the main algorithm. If no variable among $devScen1$, $devScen2$ and $devScen3$ has sufficient value then subroutine returns previous scenario if the previous scenario is *No Congestion Scenario* or *Moderate Congestion Scenario* but if previous scenario is *Extreme Congestion Scenario* then it returns *Moderate Congestion Scenario* because in *Extreme Congestion Scenario*, eNB uses EAB as a congestion handing method which bars the devices from accessing the network. So, if the difference between the values of $devScen3$ and λ_2 is not much, then it is better to return *Moderate Congestion Scenario* in place of returning *Extreme Congestion Scenario*.

Algorithm 3 Congestion Estimation Subroutine

Input: $devScen1$,$devScen2$,$devScen3$,$prevScen$,$tolerance\_factor$
Output: Congestion Scenario

1: $x = tolerance\_factor$
2: **if** $devScen1 > (100 + x)devScen2/100$ and $devScen1 > (100 + x)devScen3/100$ **then**
3: $scenario = scenario1$
4: return $scenario$
5: **else if** $devScen2 > (100 + x)devScen1/100$ and $devScen2 > (100 + x)devScen3/100$ **then**
6: $scenario = scenario2$
7: return $scenario$
8: **else if** $devScen3 > (100 + x)devScen1/100$ and $devScen3 > (100 + x)devScen2/100$ **then**
9: $scenario = scenario3$
10: return $scenario$
11: **else**
12: **if** $prevScen == scenario3$ **then**
13: $scenario = scenario2$
14: return $scenario$
15: **else**
16: $scenario = prevScen$
17: return $scenario$
18: **end if**
19: **end if**

If the subroutine returns *No Congestion Scenario* to main algorithm then eNB chooses congestion handling method for no congestion (decided by Best Congestion handling method selection algorithm) and broadcasts this information to all M2M devices in the network. Similarly, if it returns *Moderate Congestion Scenario* then eNB chooses congestion handling method for moderate congestion and if it is *Extreme Congestion Scenario* then eNB chooses EAB.

4.2 EAB Extensions

When there is high congestion in the network, the eNB need to start EAB. The duration of EAB depends on level of congestion in the network. So, EAB will be continued until number of devices accessing the network per second (say b) is coming below a threshold. This threshold is actually the capacity of network to support number of devices accessing the network per second (say e) without any congestion. So, minimum $(b-e)$ devices per second should be barred from accessing the network so that congestion can be handled. But, to unbar these devices we need to wait for the congestion to come down to a considerable level so that EAB can be removed.

Since, it is possible that number of devices configured for EAB is so big that eNB need not bar all EAB configured devices because with fewer number of barred devices also we can achieve the barring rate of $(b-e)$ devices per second (case 1). Similarly, it also possible that this number is so less that even after barring all EAB configured devices, the barring rate can not be reached to $(b-e)$ devices per second (case 2). So to address this problem, we have proposed two terms $EAB1$ and $EAB2$. The term $EAB1$ is a status which is assigned by eNB to all EAB configured devices. So in the earlier case, some or all devices having $EAB1$ status can be barred. For the later case, those devices in the network will be assigned a status $EAB2$ which have following properties: (1) Slightly less delay tolerant than $EAB1$ devices. (2) Priority of data generated by devices are less. So all or some devices having $EAB2$ status will be barred when barring of all $EAB1$ devices is not sufficient to achieve the barring rate of $(b-e)$ devices per second.

Let, k% of total devices have status of $EAB1$. So, out of b devices $bk/100$ devices will have $EAB1$ status. If value of $bk/100$ is greater than $b-e$ then only c percent of total $EAB1$ devices are needed to be barred. The value of c can be calculated as follow:

$$c = (b-e)/bk * 10000 \tag{4.1}$$

Let, d% of total devices are having $EAB2$ status. So out of b devices, $bd/100$ devices will have $EAB2$ status. If value of $bk/100$ is less than $b-e$, it means that barring of all $EAB1$ devices is not sufficient. So, in this case the eNB will start barring $EAB2$ devices. Out of $bd/100$ devices, eNB will bar $b-e-bk/100$ devices. The percentage f of total $EAB2$ devices to be barred, can be calculated as follow:

$$f = ((100-k-100e/b)/d) * 100 \tag{4.2}$$

Algorithm 4 explains whole procedure of calling $EAB1$ and $EAB2$ devices.

Algorithm 4 EAB Algorithm

Input: b, c, d, e, f, k

1: **if** $b-e < bk/100$ **then**
2: Allow all $EAB1$ devices to access the network with probablity $1-c/100$
3: **else**
4: **if** $bk/100 < b-e < bk/100 + bd/100$ **then**
5: Bar all $EAB1$ devices
6: Allow all $EAB2$ devices to access the network with probablity $1-f/100$
7: **else**
8: Bar all $EAB1$ and $EAB2$ devices
9: **end if**
10: **end if**

After applying the algorithm, eNB will wait for the congestion to come down. Duration of EAB will depend on number of devices trying to access the network before and during barring. When congestion comes down below a considerable level, the eNB will remove EAB so that all $EAB1$ and $EAB2$ devices can access the network. But just after removing EAB, along with some unbarred devices all barred devices will try to access the network at the same time. So again situation of congestion can occur. To avoid this post EAB situation, we have proposed an approach in which the eNB will ensure that just after removing EAB number of devices accessing (new, barred, unbarred) the network is same as number of devices that can be supported by eNB in a PRACH slot.
First we have calculated minimum number of PRACH slots required by eNB to spread accesses. For this we have estimated maximum number of devices barred from accessing the network during EAB. Since, b is the average number of devices accessing the network per second. As congestion will come down during EAB, value of b will also decrease. Let, m number of devices are unsuccessful to connect to network due to congestion just before EAB start. So, these m devices will again try to access the network during EAB. Since, $k\%$ of total devices have EAB1 status and $d\%$ of total devices have EAB2 status. If all EAB1 and EAB2 devices are barred from accessing the network then out of m, $m(k+d)/100$ devices will be barred and rest $m - m(k+d)/100$ will be allowed to access the network during EAB.
Now, let s number of devices tried to access first time during EAB. Similar to the case of m, in this case $s(k+d)/100$ devices will be barred and remaining $s - s(k+d)/100$ devices will be allowed to access the network during EAB. So, maximum number of devices barred (say t) during EAB will be $(m+s)(k+d)/100$ i.e.

$$t \leq (m+s)(k+d)/100 \tag{4.3}$$

Just after removing EAB barred devices will attempt to access the network from the first PRACH slot. Let there are average p devices attempt to access the network in each PRACH slot other than the barred devices. These p devices will contain (1) devices which were not barred during EAB but still unsuccessful (2) fresh devices which are attempting first time in the current PRACH slot (3) devices which started accessing the network first time after EAB but were unsuccessful in previous PRACH slots. So, number of devices accessing the network in first PRACH slot just after removing EAB is $t + p$.
Let v is the maximum number of devices supported by the eNB in one PRACH slot. So, in second PRACH slot maximum number of devices accessing the network is $t + 2p - v$ where $t + p - v$ devices were unsuccessful in last PRACH slot and p is average number of new devices accessing the network. Similarly, in nth PRACH slot maximum number of devices accessing the network will be $t + np - (n-1)v$. Let, till nth PRACH slot all barred devices have been connected to the network and network is able to handle the access rate of devices. So we can write the following equation :

$$t + np - (n-1)v \leq v \tag{4.4}$$

After solving we can write the equation as follows:

$$n \geq t/(v-p) \tag{4.5}$$

It means that if only v devices are allowed to access the network per PRACH slot then till the nth

PRACH slot all barred devices will access the network successfully and possibility of congestion after EAB can be avoided.
If there is only one PRACH slot per frame then time (say g) needed to complete n PRACH slots is $10n$ milliseconds. Now if all barred devices choose a number between 1 and g then average t/g devices will choose same number. Let the number chosen by those t/g devices is x and the nearest PRACH slot before and after xth millisecond is at yth and zth millisecond respectively. Now devices which chosen a number between $(y+1)$ and z access the network in the PRACH slot at zth millisecond then total number of devices (barred devices during EAB) accessing the network in this PRACH slot will be $10t/g$. After simplification, this value will be $v-p$. Since, the average number of new devices accessing the network is p. So total average number of devices (new, barred and unbarred during EAB) accessing the network in a PRACH slot will be v which is the maximum number of devices that eNB can support in a PRACH slot. With this approach, the eNB can spread the access sufficiently and congestion after EAB can be ignored.

4.2.1 System Model

In this section, we have calculated values of some parameters assumed in previous section. These parameters are m, s.

Calculation of m

Since, m devices are unsuccessful to access the network due to congestion just before EAB. When a device is unsuccessful in one PRACH slot then it will backoff and will attempt again. Let i is the number of PRACH slot after which the unsuccessful device will try to do RACH procedure. So, if a device tried to access the network first time at xth PRACH slot but not succeed then it will again try to access the network at $(x+i+1$th PRACH slot. Now, let j is the maximum number of backoff a device does if RACH procedure is unsuccessful. So, now we can say that device will try to access the network last time at $x+(i+1)j$th PRACH slot if it is unsuccessful in all backoffs. If the device is unsuccessful in this PRACH slot alsothen it will stop backoff and will report to higher layers about its unsuccess. So, it can be said that device was active from xth PRACH slot to $x+(i+1)j$th PRACH slot.
Let $x+(i+1)j-1$th PRACH slot is the last PRACH slot after which EAB started. So, any device which does RACH procedure first time in any PRACH slot between xth and $x+(i+1)j-1$th PRACH slot (including both) and remain unsuccessful then these devices can be assumed as active and will try to access the network during EAB also. Since, b devices per second is the average access rate of network. The value of b will increase or decrease depending on the congestion in the network. Assume that value of b before EAB is a devices per second. So, in a PRACH slot average $a/100$ devices will access the network if only one PRACH slot per frame is there. In the worst case, if all devices are unsuccessful in each PRACH slot between xth and $x+(i+1)j-1$th PRACH slot then maximum value of m will be

$$a/100 + a/100 + \cdots + \text{upto}(i+1)j\text{terms}$$

Therefore, we can write that

$$m \leq (i+1)ja/100 \tag{4.6}$$

Calculation of s

Since, s is the number of devices who are attempting to access the network first time during EAB. As calculated in previous section that out of s, $s(k+d)/100$ devices are barred from accessing the network due to EAB. Similarly, out of m, $m(k+d)/100$ devices are barred from accessing the network. So, minimum number of devices who are allowed to access the network during EAB is $(m+s)(1-(k+d)/100)$. Number of devices who succeed during EAB will be known to eNB. Let, this number is l. So, we can write that

$$(m+s)(1-(k+d)/100) \geq l$$

After simplification we can say that

$$s \geq l/(1-(k+d)/100) - m \tag{4.7}$$

After comparing equation 5 and 6, we can write

$$s \leq l/(1-(k+d)/100) - (i+1)ja/100 \tag{4.8}$$

Chapter 5

Simulation Setup and Performance Results

5.1 Simulation Setup

Matlab has been used for simulation of congestion in RACH procedure.All simulation parameters are taken from the 3GPP Specfication 37.868 [8]. Simulation parameters are given in Table 5.1.

Table 5.1: Simulation Parameters

Parameters	**Values**
Number of preambles	54
Number of MTC devices	1000 to 30000
Number of preamble retransmissions	10
HARQ retransmission probability	10%
Preamble detection probability	1- $1/e^i$ where i is the i^{th} preamble transmission
No Of RACH oppurtunities per frame	1,2
λ_1, λ_2	3,8
Simulation Time	10s,60s
BackOff Indicator	20ms

5.2 Selecting n value for NS

We need to find the best value of n for NS which will minimize the delay and increase the number of users succeeding in each slot. When the value of n is very less then spreading will be less. This is useful when the amount of congestion in the network is less. So we need to find the value of n for which the delay in successfully accessing the network is less but the number of users succeeding is more. Table **??** shows the average access delay and number of users succeeding for different values of n. We can see that when $n = 8$ the number of users succeeding is high and the delay is tolerable. When we increase the value of n the number of users succeeding goes up marginally and later on decreases. But the delay is high for higher values of n. So we take n as 8.

No. of users	No. of successful users				Average access delay			
	n =4	n =8	n =12	n =16	n =4	n =8	n =12	n =16
10000	10000	10000	9828	9906	9.90	13.84	15.23	18.08
20000	11943	12057	12054	12108	565.50	518.65	645.40	689.59
25000	6491	6594	6562	6545	691.93	647.97	700.34	799.80
30000	5891	5936	5884	5869	707.46	687.01	754.60	826.06

Table 5.2: Different values of n for uniform distribution

No. of users	No. of successful users				Average access delay			
	n =4	n =8	n =12	n =16	n =4	n =8	n =12	n =16
10000	7031	7126	7193	7291	235.24	229.27	222.14	215.88
20000	4571	6471	6556	6621	595.26	564.50	653.10	688.99
25000	3822	5924	6054	6126	530.36	542.28	549.64	684.85
30000	3697	4919	5032	5037	569.34	566.89	692.43	725.18

Table 5.3: Different values of n for beta distribution

5.3 Results of Congestion Management Function

In this section, based on the simulation results we have analyzed the performances of Best congestion handling method selection algorithm and Congestion Management Function (main algorithm). Best congestion handling method selection algorithm takes congestion handling method and λ as input and returns t_1 and t_2 as output. Here, λ denotes whether the congestion scenario is *No Congestion Scenario* or *Moderate Congestion Scenario*. For *No Congestion Scenario*, we have assumed the value of λ is 3 and for *Moderate Congestion Scenario*, it is 8. We run the Best congestion handling method selection algorithm for various congestion handling methods and note the value of t_1 and t_2 as shown in Table 5.3. For *No Congestion Scenario*, the value of t_1 and t_2 is maximum when congestion handling method is numbering scheme and for scenario 2, it is p-persistent. As discussed in previous chapter, in a particular congestion scenario, the congestion handling method having maximum value of t_1 and t_2 is the most efficient method for that scenario. So from the Table 5.3, we can conclude that numbering scheme is best for *No Congestion Scenario* and p-persistent is best for *Moderate Congestion Scenario*. Further, these results are used by Congestion Management Function. As discussed in previous chapter, when eNB run Congestion Management Function at a particular time instant, it first estimates the condition of congestion in the network. If condition of congestion is *No Congestion Scenario* then eNB chooses numbering scheme as congestion handling scheme as concluded from Table 5.3. Similarly, if it is *Moderate Congestion Scenario* then eNB chooses p-persistent and if congestion level is *Extreme Congestion Scenario* then eNB chooses EAB. So by using this algorithm, eNB alleviates the effect the congestion significantly. To evaluate the performance of Congestion Management Function, we use following parameters: (i) number of successful users (ii) average access delay.

Average access delay is defined as the difference between the time when a device is ready to start RACH procedure and its RACH procedure finishes successfully. To compare the performance of Congestion Management Function with other congestion handling methods, we plot a number of successful user versus number of user graph as shown in Figure 5.1. In the figure, we can see that

Table 5.4: Results of Best congestion handling method selection algorithm

Schemes	**No Congestion**		**Moderate congestion**	
	$t1$	$t2$	$t1$	$t2$
NS	195	195	60	65
Slotted Access	45	55	25	45
BackOff Indicator Adjustment	40	60	30	45
p-persistent	160	210	80	70

some of the methods like backoff indicator adjustment and base case perform better when number of users in the network are less but their performances start degrading when number of users in the network increases. While for some other method like p-persistent, its performance is good when number of users are high in the network. This is because in case of very low congestion in the network almost every PRACH attempt will succeed because the number of active devices is very less. So sending with some probability will increase the delay. But, the Congestion Management Function performs better in both low congestion and high congestion.

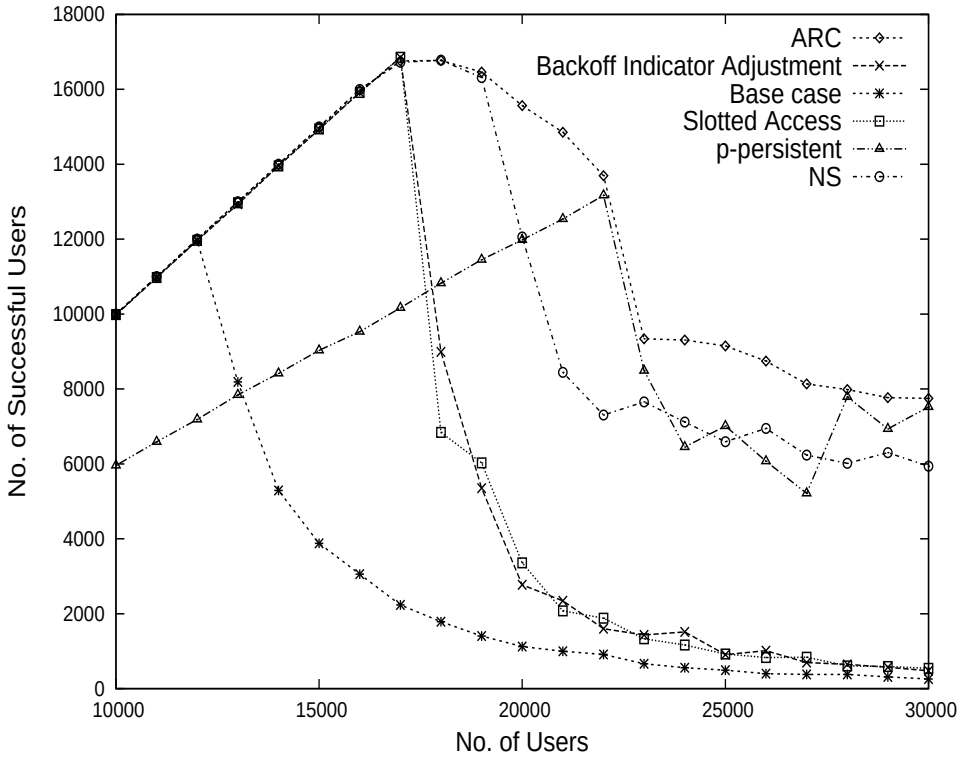

Figure 5.1: Number of successful users for uniform distribution

In Figure 5.2, we plot average access delay versus number of users graph. Here, performance of Congestion Management Function is very good in comparison to backoff indicator adjustment, base case and slotted access while in case of p-persistent, performance of Congestion Management Function is comparable for lesser number of user while it is higher in case of higher number of devices

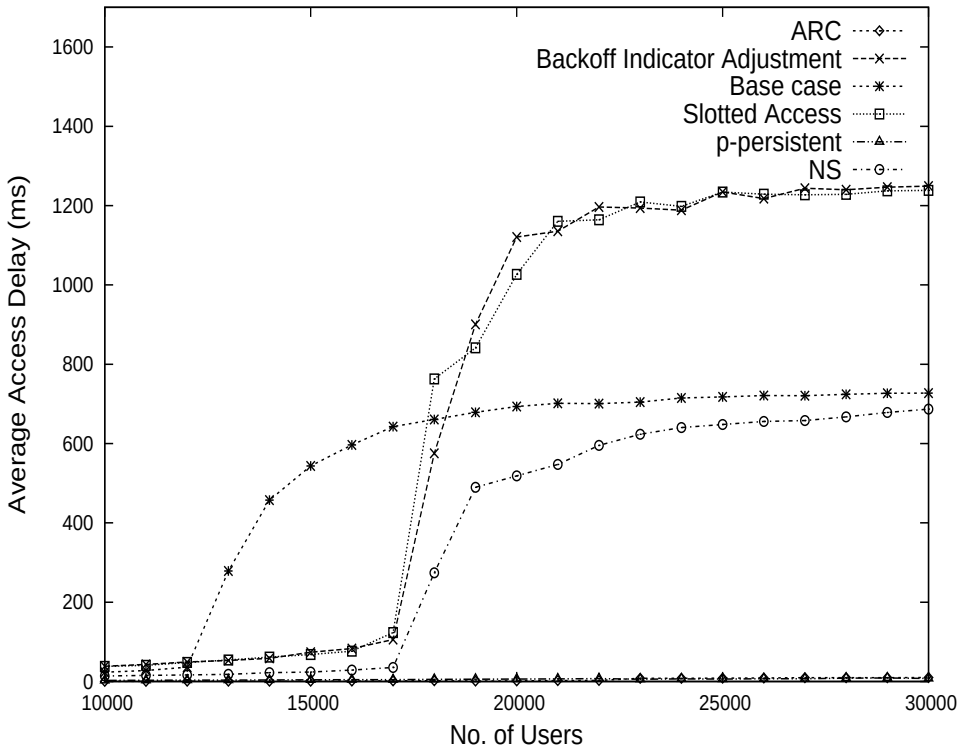

Figure 5.2: Average access delay Per User for uniform distribution

for uniform distribution. In case of very high congestion also, Congestion Management Function performs well because in this case, the algorithm chooses EAB as congestion handling method. Since, EAB allows more devices which are not barred to succeed, so average access delay remains low. In Figure 5.3, we plot average backoff per user for different methods and we can see that the average backoff is less for Congestion Management Function for uniform distribution.
In case of beta distribution we can see that Congestion Management Function performs better than other solutions 5.4. In case of access delay the congestion management function has a higher value than p-persistent and numbering 5.5. Congestion Management function has lesser backoff than any other congestion handling function 5.3.

5.4 Results of EAB Extensions

The simulation parameters are chosen according to 3GPP specifications [8]. The detailed simulation parameters are given in Table 5.5. We have simulated the basic RACH setup which uses EAB as access barring method. This is taken as the base case. After that we have simulated EAB1 and EAB2 setup. In this case, we have used same amount of devices for EAB1 as that for basic EAB setup. Additionally we have added 20% of the other devices as EAB2 configured devices. We have used a distribution similar to beta distribution for creating congestion in the network. This is an extreme scenario in which access rate is more than 2700 devices per second. We use 54 preambles per RACH occasion. The other 10 are assumed to be used for contention free random access. We

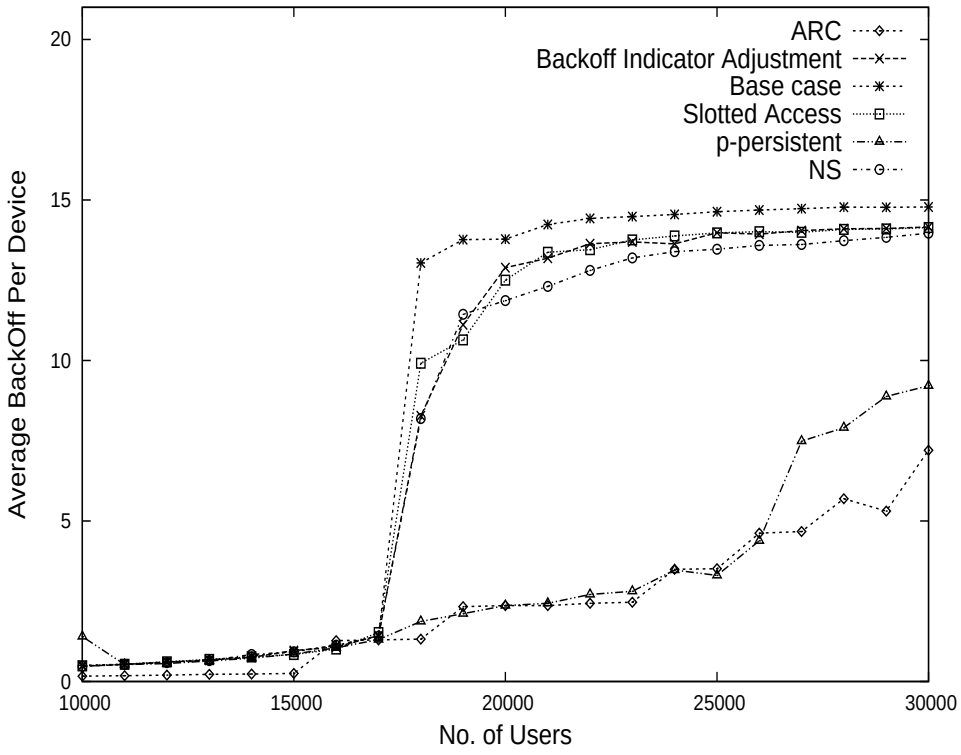

Figure 5.3: Average BackOff Per User for different algorithms for uniform distribution

have assumed 60% of M2M devices as EAB configured devices.

We have assumed single RACH occasion (PRACH slot) per frame. So a RACH occasion happens once in every 10 ms. Every user which gets data to send will wait till the next PRACH slot and send the data. The distribution is such that there is a region where the number of devices accessing the network will be considerably less than what the network can handle. This happens before and after the high congestion region. So every user who is barred from sending data when EAB was on gets a chance to unbar and try on the low congestion region. This is the normal pattern which happens in an actual setup. Normally there will be sudden peaks in usage in the network due to synchronous access of multiple devices. Then this peak will reduce and the usage goes back to normal and below normal depending upon the devices in the network.

When the number of users is less EAB2 will perform like normal EAB. As the number of users increases more congestion happens and more users try to access the network. Our approach helps us to increase the success probability for each device accessing the network. We have assumed that each device does one access attempt (including the backoffs and retries). If the device fails to access the network after PREAMBLE_TRANSMISSION_MAX attempts then it is considered as failed and it does not try to access the network again. If a device succeeds in accessing the network it will not try again. It is assumed to be connected to the network throughout the simulation after the successful RACH procedure.

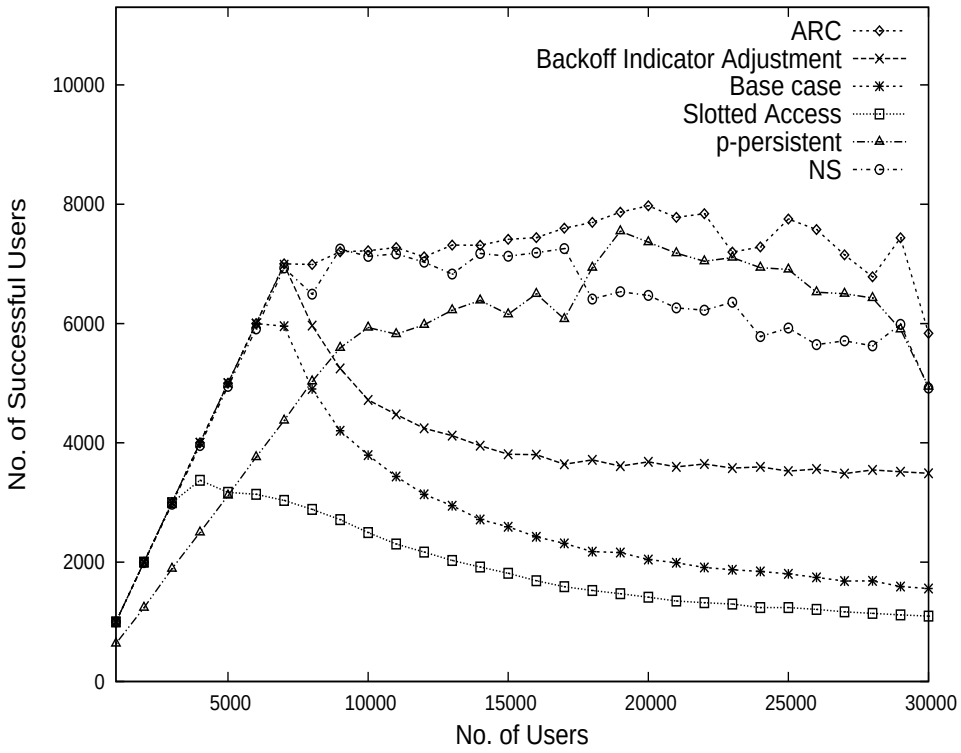

Figure 5.4: Number of successful users for beta distribution

5.4.1 Analysis

When the congestion in the network increases initially both normal EAB procedure and the proposed procedure works almost in the same way. But when the congestion becomes more than what the eNodeB can handle even with EAB, then the proposed solution will give better results in terms of number of users succeeding and access delay. This is because of the fact that number of retries becomes less. When more devices are barred, those devices that are unbarred will be able to access the network as the number of devices accessing the network will be below the threshold which the eNodeB can handle. These devices will succeed and when EAB is turned off the other devices will have fewer devices to compete with resulting in more devices succeeding once the EAB is turned off. This decreases the access delay and number of retries.

When EAB is turned off there is a sudden increase in number of devices accessing the network. This will cause congestion. When we use randomization to remove these spikes the access is spread. This increases the chance of a device succeeding at that time. The randomization algorithm estimates the amount of spread required. So if the number of devices trying to access the network after EAB is less then amount of spread will be less. When congestion is very less this algorithm gives similar results to that of EAB. When congestion in the network increases number of devices trying to access the network after EAB also goes up. This will reduce the number of users succeeding at that time. Almost all devices accessing the network at the time when EAB gets over will back off. This will have a spiraling effect because these devices will try to access the network again after back off. These devices will cause more congestion and the system might get really congested because of this. When

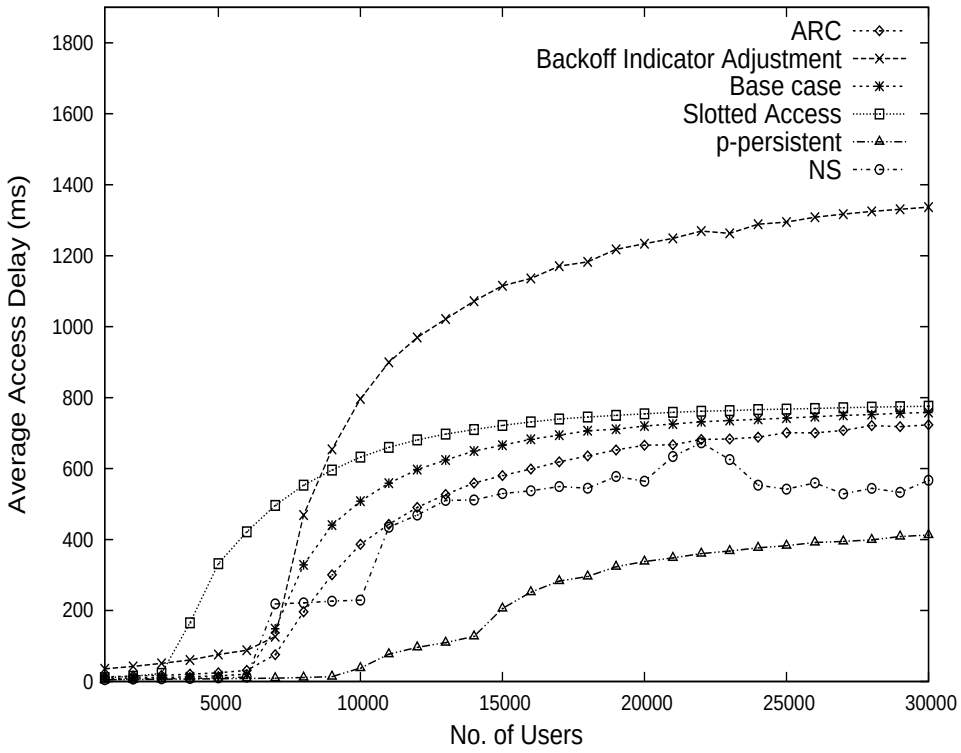

Figure 5.5: Average access delay Per User for beta distribution

randomization is applied less number of devices accesses the network immediately after EAB. Most of these devices will succeed because congestion is not there. When the congestion is more we can see that Figure 5.9 randomization performs better than normal EAB. More users succeed in case of randomization when compared to normal EAB. It also has the lesser number of backoffs and less delay when compared to normal EAB.

Average access delay for EAB1 and EAB2 is lesser than that of other methods. The reason is that with EAB spike removal method the access is spread further and delay will be higher than just EAB1 and EAB2 as shown in Figure 5.8. Average backoff per device for EAB1 and EAB2 method is lesser than that of normal delay because the congestion is controlled much better using this method and backoffs are also minimized. The best value of average backoff is for EAB spike removal because in this case the number of devices backing off because of the sudden access happening at the end of EAB is also reduced as shown in Figure 5.7.

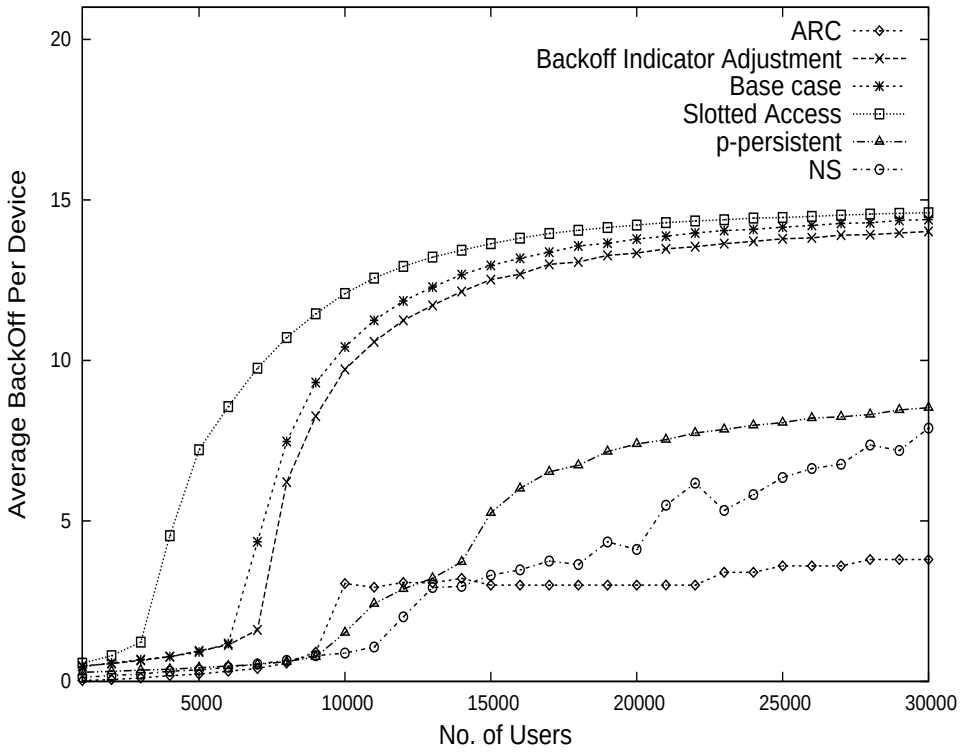

Figure 5.6: Average BackOff Per User for different algorithms for beta distribution

Table 5.5: Simulation Parameters for EAB

Parameters	**Values**
Number of devices a system can support per PRACH slot (v)	54
Number of MTC devices	1000 to 30000
Maximum number of preamble retransmissions(j)	10
HARQ retransmission probability	10%
Percent of devices having EAB1 status(k)	30
No Of RACH oppurtunities per frame	1
Percent of devices with EAB2 status (d)	20
Simulation Time	10s
ra-ResponseWindow(i)	5

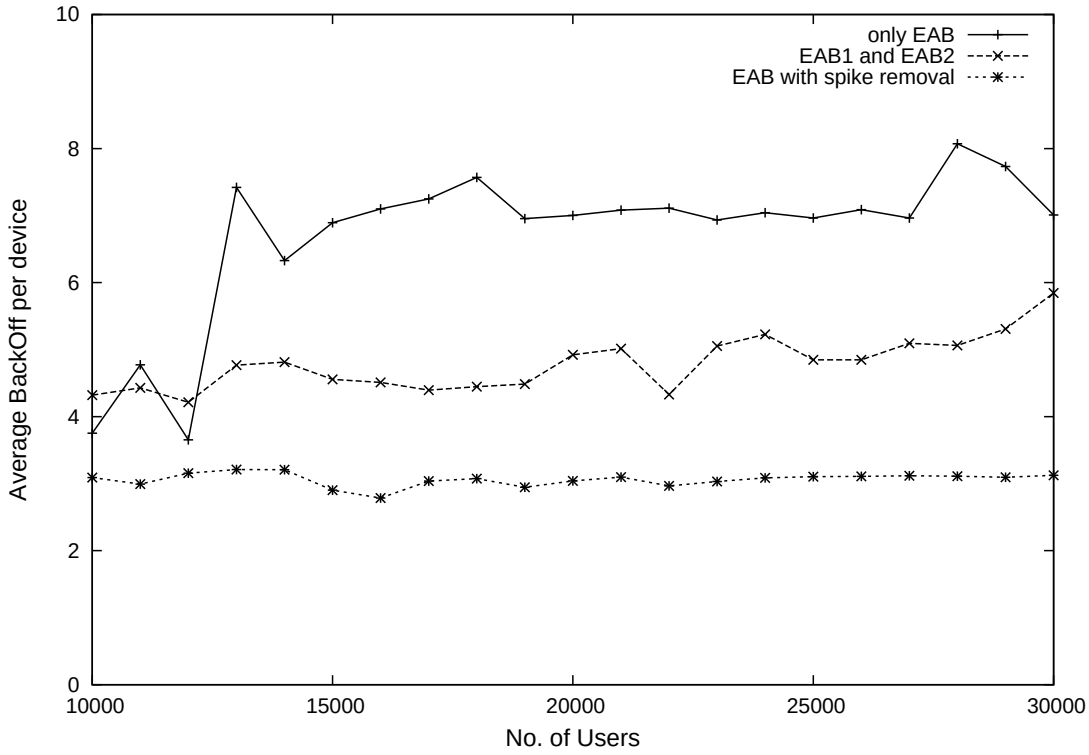

Figure 5.7: No of Users vs Average Backoff Per device for EAB

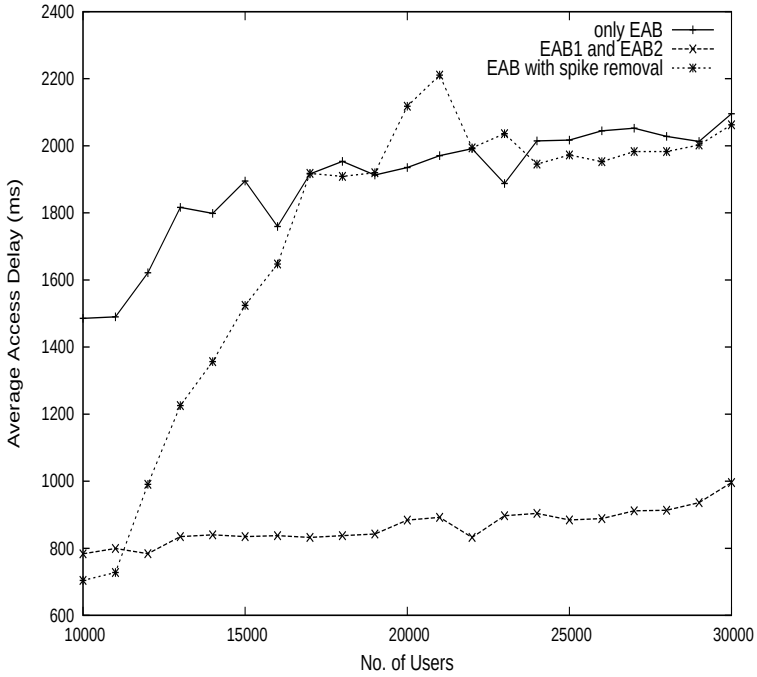

Figure 5.8: No of Users vs Average Access Delay for EAB

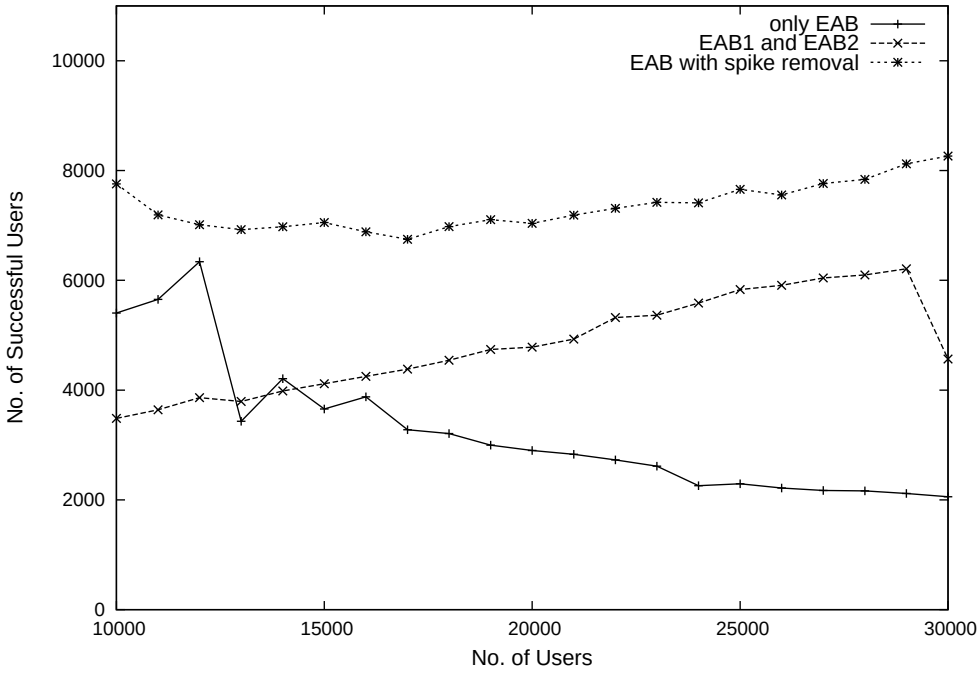

Figure 5.9: No of Users vs No of Successful Users for EAB

Chapter 6

Conclusions and Future Directions

We studied the performance of various RACH congestion handling mechanisms.We saw that each method works best with some particular congestion state of the system and does not perform very well when the load in the system is different. We proposed a novel congestion handling method, numbering scheme, which performs very well when the load on the network is low to medium. We classified the level of congestion in the network to three scenarios.(i) *No Congestion Scenario*,(ii) *Medium Congestion Scenario* and (iii) *Extreme Congestion Scenario*. We also proposed a method by which we can find out which algorithm performs better in a given congestion state of the system. Simulation results have shown as NS performs better in the first scenario and p-persistent approach performs better in the second scenario.

A novel congestion management function was proposed which will estimate the current load in the cell and then decide which congestion handling mechanism is to be used by MTC devices based on that.To simulate different types of congestion in a cell we used uniform random distribution as well as beta distribution. Uniform distribution shows the condition of the network when the load on the network is increasing gradually. Beta distribution shows the condition of the network when number of devices accessing the network increases suddenly. Simulation results showed that this algorithm performs better than any single congestion handling method and is able to perform well in lightly as well as heavily loaded condition of the cell in both distributions.

When the load of the system becomes too high for eNB to handle it will perform EAB mechanism to do admission control. We proposed an extension to EAB so that it can perform well even when the load on the network is too high even after barring all EAB configured devices. We also proposed a randomization method which helps in reducing the the sudden peak in network access caused when EAB is switched off. This peak in usage is caused because all barred devices becomes unbarred when EAB is turned off and they will try to access the network at the same time. Simulation results have shown that randomization method helps in spreading this access so that the eNodeB is able to handle the number of devices trying to attach to it. This reduces the delay in access as well.

Though our EAB extension is performing better than normal EAB, the central challenge for future is to speed up the processing step and get a more accurate method for predicting the load on the network. Another challenge is to look at the load on the network some time in past and then predict the current network level to use the optimal congestion handling method for the level of congestion.

References

[1] Ming-yuan Cheng,Guan-yu Lin, and Hung-yu Wei. Overload Control for Machine-Type-Communications in LTE-Advanced System . *J. Power Sources* 57, (2012) 38–44.

[2] Jen-Po Cheng , Chia-han Lee, and Tzu-Ming Lin, C. S. Adjiman. Prioritized Random Access with dynamic access barring for RAN overload in 3GPP LTE-A networks. *J. Power Sources* 138, (2011) 368–372.

[3] Shiann-Tsong Sheu, Chun-Hsiang Chiu, Yen-Chieh Cheng, and Yen-Chieh ChengKai-Hua Kuo. Self-Adaptive Persistent Contention Scheme for Scheduling Based Machine Type Communications in LTE System. 2nd edition. John Wiley & Sons, 2012.

[4] D. S. Watson, M. A. Piette, O. Sezgen, and N. Motegi. Machine to machine (M2M) technology in demand responsive commercial buildings *in Proc. Of 2004 ACEEE Summer Study on Energy Efficiency in Buildings 2004.*

[5] Ming-yuan Cheng,Guan-yu Lin, Hung-yu Wei, and Chia-Chun Hsu. Performance Evaluation of Radio Access Network Overloading from Machine Type Communications in LTE-A Networks. 2nd edition. John Wiley & Sons, 2012.

[6] Ki-Dong Lee,Sang Kim, and Byung Yi. Throughput Comparison of Random Access Methods for M2M Service over LTE Networks. 2nd edition. John Wiley & Sons, 2011.

[7] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jeffrey Pang, and Jia Wang. A First Look at Cellular Machine-to-Machine Traffic Large Scale Measurement and Characterization. 2nd edition. SIGMETRICS, 2012.

[8] 3GPP TR 37.868 V0.5.1, Study on RAN Improvements for MachineType Communications. 2010.

[9] 3GPP TS 36.321 V10.0.0, Evolved universal terrestrial radio access (E-UTRA) medium access control (MAC) protocol specification. 2012.

[10] 3GPP TSG RAN WG2#73BS R2-112863, Backoff enhancements for RAN overload control. 2011.

[11] 3GPP TSG SA WG2 Meeting#79E TD S2-103125, Back-off time randomization for overload control. 2010.

[12] Nuno K. Pratas, Henning Thomsen, Cedomir Stefanovic, and Petar Popovski. Code-Expanded Random access for machine-type communications . 2nd edition. John Wiley & Sons, 2012 1681–1686.

[13] 3GPP TSG SA WG2 Meeting#77 TD S2-100xxx, MTC Signal Congestion control. 2010.

[14] 3GPP TSG-RAN WSG2 Meeting#74 TD R2-113030, Extended Access Barring for MTC devices. 2011.

[15] Umesh Phuyal, Ali T Koc, Mo-Han Fong, and Rath Vannithamby. Controlling Access Overload and Signalling Congestion in M2M Networks. 2nd edition. Asilomar, 2012 591–595.

[16] A. A. Iordanidis. Mathematical Modeling of Catalytic Fixed Bed Reactors. Ph.D. thesis, University of Twente, the Netherlands 2002.

[17] S. Dye. Machine-to-machine (M2M) communications *Article (CrossRef Link)*,

[18] Min Chen, Jiafu Wan, and Fang Li. Machine-to-Machine Communications:Architectures, Standards and Applications *Transactions on Internet and Information Systems 6, no. 2 2012.*

[19] V. Galeti, I. Boji, M. Kuek, G. Jei, S. Dei, and D. Huljeni. Basic principles of Machine-to-Machine communication and its impact on telecommunications industry, MIPRO, 2011.

[20] 3GPP TS 22.368 V11.3.0, Service requirements for Machine Type Communications (MTC) Stage 1. Sept 2011.

[21] Draft ETSI TS 102 689 V0.4.1 (2009-mm), Machine-to-Machine communications (M2M); Service requirements 2009.

[22] 3GPP TR 22.868 V8.0.0, *"Study on Facilitating Machine to Machine Communication in 3GPP Systems,"*,. March 2007.

[23] 3GPP TR 33.812 V9.2.0, *"Feasibility study on the security aspects of remote provisioning and change of subscription for Machine to Machine (M2M) equipment,"*,. June 2010.

[24] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Lwaski, and Y. Nozaki. Toward intelligent machine-to-machine communications in smart grid, *IEEE Communications Magazine 49, (2011) 60–65.*

[25] Dusit Niyato, Lu Xiao, and Ping Wang. Machine-to-machine communications for home energy management system in smart grid, *IEEE Communications Magazine (2011).*

[26] Y. Zhang, R. Yu, S. L. Xie, W. Q. Yao, Y. Xiao and M. Guizani Home M2M networks: architectures, standards, and QoS improvement, *IEEE Communications Magazine 49, (2011) 45–52.*

[27] http://en.wikipedia.org/wiki/Internet of Things

[28] http://en.wikipedia.org/wiki/Wireless_sensor_network

[29] http://en.wikipedia.org/wiki/Cyber Physical Systems

[30] http://www.etsi.org/WebSite/homepage.aspx

[31] J. Hui, D. Culler, and S. Chakrabarti. 6LoWPAN: Incorporating IEEE 802.15.4 Into the IP Architecture ,. Internet Protocol for Smart Objects (IPSO) Alliance, White Paper 3, 2009.

[32] 3GPP TSG GERAN#49 GP-110277, Realizing Extended Access Barring. 2011.

[33] 3GPP TS 22.011 V11.2.0, Digital cellular telecommunications system (Phase 2+);Universal Mobile Telecommunications System (UMTS);LTE;Service accessibility,. 2011-12.

[34] 3GPP TSG RAN WG2#75 R2-115215, Consider some issues of EAB,. 2011.

[35] 3GPP TSG RAN WG2#77 R2-120195, EAB information update procedure,. 2012.

[36] 3GPP TSG RAN WG2#77 R2-121229, Fast EAB update mechanism,. 2012.

[37] Anna Larmo, and Riikka Susitaival. RAN overload control for Machine Type Communcations in LTE. *GC'12 Workshop 138, (2012) 1626–1631.*