MASTER'S THESIS

An LTE Random Access Channel Model for Wireless Sensor Network Applications

Michael Burakov
Albin Eldstål-Damlin

Master of Science in Engineering Technology
Computer Science and Engineering

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

An LTE Random Access Channel Model for Wireless Sensor Network Applications

Michael Burakov
Albin Eldstål-Damlin

Dept. of Computer Science, Electrical and Space Engineering
Luleå University of Technology
Luleå, Sweden

April 6, 2012

Supervisor:
Evgeny Osipov

ABSTRACT

In this thesis we investigate the scheduling aspect of using 3GPP Long Term Evolution (LTE) as a radio transport in Wireless Sensor Network (WSN) applications.

A possible performance bottleneck in LTE transmission scheduling is identified where a typical WSN would interfere with traditional cellular network users. All users are alloted equal resources in a specialized signalling channel (PUCCH), which are to be used for scheduling requests. This indicates that a WSN with a high number of nodes and a low traffic intensity would skew that signalling channel in such a way that regular users would see increased delays and thus latency. We suggest a subtle change in the LTE scheduling request procedure to allow such low-intensity nodes to use Random Access (RACH) exclusively, foregoing their resources in the regular scheduling channel.

We use a probabilistic network-level model of Multichannel Slotted ALOHA[1] as the basis for our own mathematical model of LTE RACH. This is done by identifying the decision points of the RACH protocol and devising ways to model them using probability theory. several key changes to the base model are proposed to better suit our purpose.

Finally, the results are verified using a discrete-time event simulator. The operation and structure of the simulator are described in detail and its results compared and used to correct misalignments in the original model.

We conclude that the throughput and delay of the LTE Random Access Channel can be accurately modelled as a Slotted ALOHA network with multiple independent channels and suggest methods to model the two success/failure selection stages.

Contents

ACKNOWLEDGEMENTS

This thesis was made possible with the help from many. We would like to thank our supervisor Evgeny Osipov and Laurynas Riliskis at Luleå University of Technology, for their guidance and effort throughout the project, Mats Nordberg and Wang Min at Ericsson Research for insight, ideas and feedback, Mårten Ericson at Ericsson Research for keen insights on simulator design. Many thanks to the people at Luleå Academic Computer Society (LUDD) for a pleasant working environment as well as much needed relief during evenings of work.

Chapter 1

Introduction

1.1 Motivation

Wireless Sensor Networks is an active research area, and a lot of work is put into techniques to improve them. The current state of the art employs wireless technologies such as Bluetooth and Zigbee for short-range mesh-type communications.

This thesis investigates some possibilities to use next-generation cellular networks (4G LTE) for the wireless data transmission. This would allow more commodity hardware to be used in the sensor nodes, while also letting WSN designers take advantage of the centralized cellular network model and existing infrastructure.

In order to realize this, care must be taken to ensure that a WSN in a public cellular network does not adversely affect the regular users. We study what impact this would have and how it can be mitigated by design.

1.2 Problem Definition

1.2.1 WSN and LTE

When discussing Wireless Sensor Networks, one usually considers a mostly uplink-bound network pattern. The archetypical WSN contains sensor nodes which send either continuous or periodic data updates to some centralized aggregation point. For this reason, we have chosen to omit discussion of the LTE downlink, focusing exclusively on uplink transmissions and scheduling.

The main characteristic of such a network is that it contains a large number of nodes which do not necessarily consume a lot of resources. This is in contrast with a typical LTE network, where cellular phones (sustained bursts of voice data) and other mobile devices (web surfing traffic, sustained downloads) typically consume large swaths of bandwidth.

The main challenge in supporting an LTE WSN is thus to handle the large number of nodes, and to allow these nodes an energy-efficient signalling protocol for extended periods of operation.

What is LTE?

3GPP Long Term Evolution (LTE) is a radio standard geared towards the fourth generation of mobile devices (4G). It differentiates itself from older standards mainly by being a fully packet-based transport, with no special telephony features. It employs the classic star topology with a central base station known as the *Evolved Universal Terrestrial Radio Access Network Node B*, shortened to *E-UTRAN NodeB*, *eNodeB* or *eNB*.

As part of the design goal of an entirely packet-oriented radio protocol, LTE is designed to work as the transport and physical layers of a standard IP stack according to the OSI model.

A technical description of the relevant parts of LTE can be found in Chapter 2.

WSN vs Cellular Network

One crucial difference between LTE and a traditional WSN is that WSNs are usually envisioned as mesh-type networks, where nodes communicate directly with each other but destination nodes may be out of direct range of transmitters.

LTE on the other hand is a centralized star-topology setup, where each node only ever communicates with eNodeB and it, in turn, routes traffic where it needs to go.

This can be an advantage or a disadvantage. It is likely, for example, that the average distance to eNodeB is longer than the average distance to a neighbour, yielding longer transmission distances which therefore require more energy.

On the other hand, the direct connection to eNodeB does away with the need for relay nodes, which evens out the battery life of nodes in the network by eliminating special roles which would drain certain nodes faster than others.

Furthermore, LTE would assign IP addresses to nodes and therefore allow WSNs to span multiple cells seamlessly. Such a layout allows a single WSN to span a large area without, as in the mesh case, having to contend with regional energy depletion fragmenting the network.

Relying on an existing LTE cell may not always be feasible, due to limited coverage in remote areas, charging arrangements with carriers or other factors.

This thesis will not further discuss the pros and cons from such a high-level perspective, but rather focus on the technical details and feasibility. Some of these questions are raised again in Chapter 7 as grounds for future study.

1.2.2 First impression

Due to the low-traffic nature of a typical WSN and the resource allocation (plentiful data channels, scant control channels) in LTE, it is immediately apparent that the main

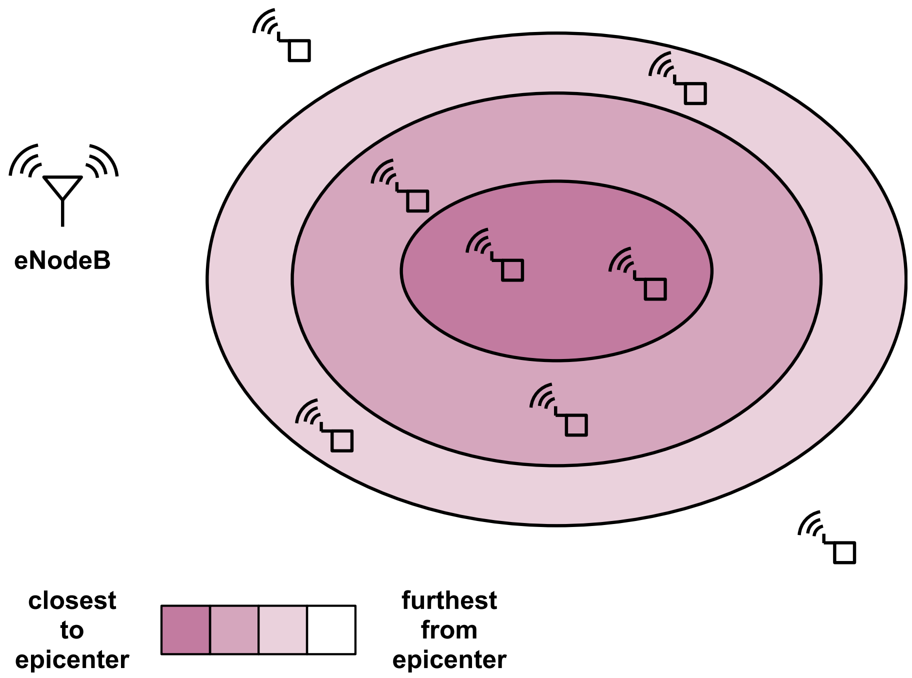


Figure 1.1: An example WSN around an event source

bottleneck would be control traffic and not data bandwidth. The technical details of this are described further in Chapter 2.

1.3 A WSN Scenario

Figure 1.1 shows an example of WSN nodes arranged around an event source, such as a volcano whose temperature is being monitored. When an event occurs (the temperature changes), nodes closer to the epicenter will need to transmit updates more frequently than those far away. Therefore, we may consider these nodes as divided into *zones*. Each zone has a number of nodes in it, and all nodes in the same zone transmit their data with a certain fixed frequency. Such a scenario will be further considered in Chapter 6.

1.4 Related Work

Jian Zhang et al. provide an overview of the reasoning behind convergence of Wireless Sensor Networks and Mobile Communication Networks[2]. It is noted that both sides of this equation can gain from such a merger; WSN can benefit from the greater robustness and signal integrity of modern cellular technology, Mobile Service providers gain a greater potential for applications wherever WSNs are and will become relevant. Some thoughts are presented with regards to *Air-Interface Convergence*, which is where this thesis delves deeper.

In September of 2011, the 3rd Generation Partnership Project (3GPP, responsible for LTE) concluded a study[3] on how to improve the LTE radio access network for machine-to-machine (M2M) communications such as WSNs. Among other things, the study suggests dividing Random Access resources between M2M nodes and human users. Simulation results are presented and a method is suggested to suppress Random Access from more latency-tolerant devices in situations where a cell is threatened by overload. This approach is also investigated by Lee et al. in [4].

Lien, Chen and Lin[5] describe the current state of M2M in 4G and evaluates performance by simulation. A grouping-based approach to resource management is proposed.

Lien, Liau, Kao and Chen[6] further suggest a way for multiple LTE cell base stations to cooperatively manage large numbers of M2M nodes by adapting 3GPP's Access Class Barring.

Marius Corici et al. discuss the evolution of other such resource reservation methods when applied to the LTE Core Network for the purpose of improving M2M support[7]. The article suggests three mechanisms changing the way users are handled internally in LTE, based on the user's expected traffic behavior.

Aleksandar Damnjanovic et al. investigate the usage of pico- and femtocells in LTE networks[8]. Such techniques would allow an LTE-based WSN to operate even in areas with no carrier coverage.

1.5 Contributions

- A simplified WSN scenario supporting heterogeneous zones;
- An analysis of LTE uplink scheduling;
- A suggested modification of LTE to reduce adverse impact of WSN on regular cellular users;
- A mathematical model of LTE RACH, with and without collision resolution.

1.6 Outline and Methodology

In order to determine whether LTE network can be adapted to WSN applications this thesis covers the following parts.

In Chapter 2, we first give a brief introduction to LTE with a short description of its functionality. There one can find information about the LTE stack, how LTE packets are composed into transport blocks and what channels are used to transmit it. We also describe a potential bottleneck, both with regards to WSN performance and interference with existing cellular users.

This is followed by presenting a mathematical model of the RACH channel, in Chapter 3. The proposed model is based on an existing S-ALOHA model with multiple channels, using Markov chains and probability theory. Performance evaluation of the model is present in the form of metrics such as throughput and delay.

Further, we introduce a simulator of the defined model in Chapter 4. The simulator is implemented in MATLAB and used to verify correctness of the RACH model by observing the same metrics. Differences in results of the model and simulator are identified, which prove incorrectness in the original model. Therefore, modifications to it are made according to Chapter 5.

Results of the final modified model are presented in Chapter 6. These results evaluate the model by comparison with the simulator in the form of various graphs.

Finally, suggestions of what more can be done within current topic are present in Chapter 7. A brief summary and concluding remarks are found in Chapter 8.

Chapter 2

A Brief Introduction to LTE

2.1 Terminology

Recall that LTE is used for cellular networks. As the name implies, the larger network is divided into operational *cells*, with a *base station* assigned to handle each. This base station is known as *eNodeB*. Also in the cell are the *User Equipment* (or UE) nodes, such as cellular phones, wireless sensor nodes, personal computers, etc. These are also commonly referred to as *terminals* or *nodes*.

While inter-cell communication is an important part of LTE, this thesis will only discuss communication between network users and their immediate base station.

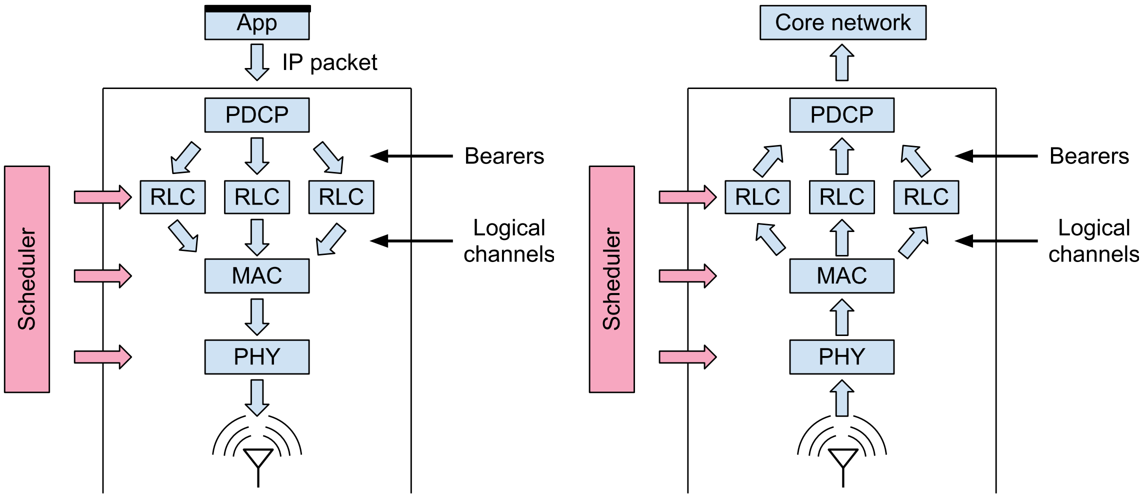
2.2 The LTE Stack

The LTE radio stack (figure 2.1) provides the Data Link and Physical layers of the OSI model, and is intended for use under an IP implementation.

It consists of four layers:

- PDCP handles optional ciphering and IP header compression
- RLC handles packet segmentation and retransmission (ARQ)
- MAC handles priorities, retransmission (HARQ) and scheduling
- PHY handles radio carrier modulation and checksumming

Each layer adds its own set of headers to its PDU (the data passed down the stack) when transmitting and strips it from the SDU (the data passed up the stack) when receiving. This nesting of headers is shown in figure 2.2.



(a) The radio stack of a transmitting terminal *(b) The radio stack of a receiving eNodeB*

Figure 2.1: The LTE stack when transmitting and receiving data

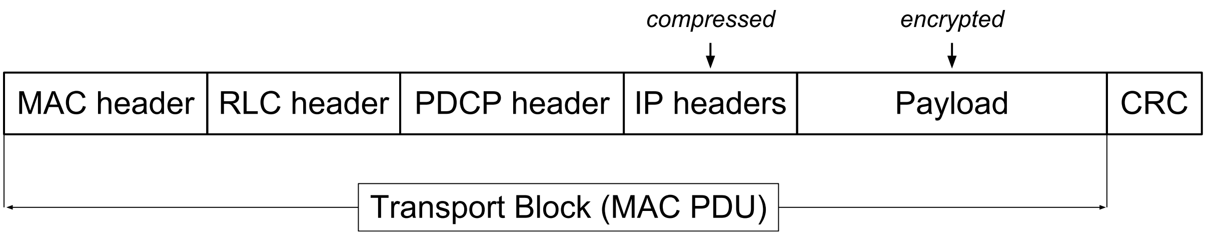


Figure 2.2: The header hierarchy of a transport block

2.2.1 Transmission

When a node X transmits an IP packet over LTE, the following sequence of events takes place in X's radio stack

- The PDCP layer performs optional compression of IP headers
- The PDCP Layer performs optional encryption of the IP payload and IP headers
- The PDCP Layer attaches its own header and sends the resulting PDU down the stack
- The RLC fragments the data based on recommendation from the scheduler
- The RLC adds its header to each of the fragmented PDUs and sends them down the stack

- The MAC concatenates RLC PDUs (possibly from multiple radio bearers/RLC instances) based on scheduler recommendation
- The MAC adds its header to its PDU and sends it down the stack. This PDU is also known as a Transport Block.
- The PHY calculates a CRC and adds it to the end of the MAC PDU

2.2.2 Reception

When the data arrives at the eNodeB, the following takes place in its radio stack

- The PHY calculates the CRC of the transport block
- The MAC determines whether the data can be fully decoded, or requests a retransmission if it cannot (see Section 2.3 below)
- The MAC splits up the various RLC PDUs based on the MAC header and passes them to their respective RLC instances
- The RLC re-concatenates the fragmented parts based on their RLC headers and passes them up the stack
- PDCP decrypts the IP headers and payload if they were encrypted
- PDCP decompresses IP headers if they were compressed
- PDCP passes the IP packet up through the rest of the stack, to be handled by the core network behind eNB.

2.3 The MAC Layer

The MAC layer employs a retransmission scheme known as *HARQ with Soft Combining*. HARQ uses ACK/NAK responses to indicate whether a retransmission is necessary. Soft Recombining is a technique where each retransmission of a packet is re-encoded differently, such that each consecutive failure can still be added to what is known at the receiver. This way, several failed retransmissions can be recombined into one correct copy of the source packet. Soft Recombining increases retransmission robustness in, for example, environments with high frequency selectivity.

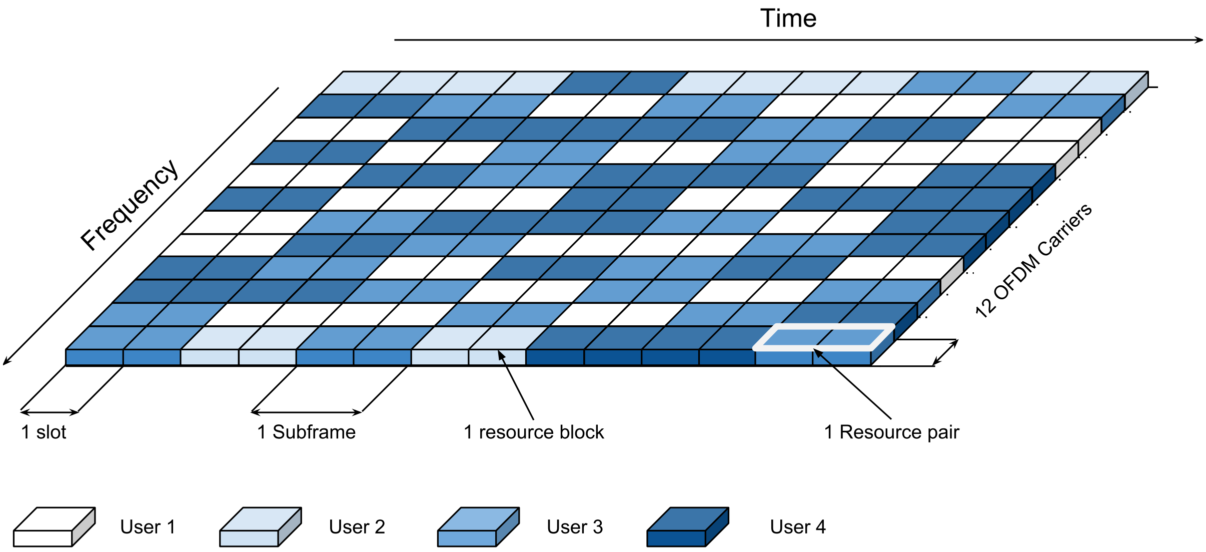
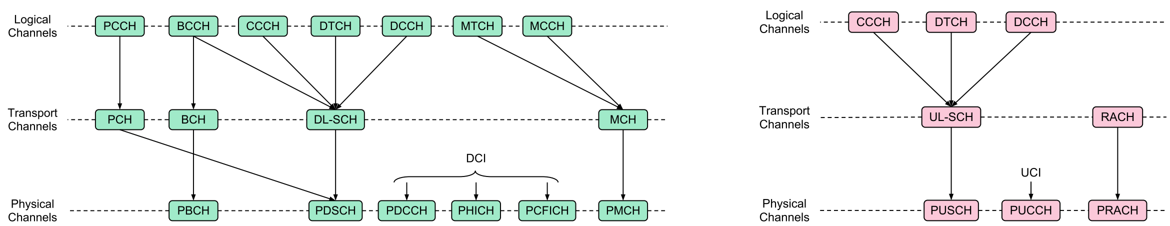


Figure 2.3: An illustration of resource scheduling in a data channel



(a) Downlink data transmissions *(b) Uplink data transmissions*

Figure 2.4: Channel mappings

2.4 The Physical Layer

2.4.1 Frequency Division

A service provider is typically allotted a *frequency band* where their signals are allowed. The first challenge of a radio protocol is to divide this band into useful parts. In LTE, a technique known as Orthogonal Frequency Division Multiplexing (OFDM) is employed[9]. In OFDM, the frequency band is divided into a number of carrier frequencies, which are orthogonal to each other. This means that a signal on carrier 1 will not interfere with a signal on carrier 2, etc. Using OFDM, multiple devices can communicate with eNodeB at the same time in the shared frequency band, without disrupting each other or interfering with neighbouring LTE cells (which use different frequency bands).

Various encoding schemes are used to transmit digital information on these carriers with a symbol resolution of 1, 2, 4 or 6 bits per transmitted symbol (using BPSK, QPSK, 16QAM or 64QAM, respectively). The encoding scheme used for any given channel at any given moment is selected based on channel quality and received power.

For example, if a symbol takes 0.1 second to transmit and the frequency band has been divided into 20 carriers, this yields a total maximum throughput of $\frac{6 \cdot 20}{0.1} = 1200$ bits per second using 64QAM.

2.4.2 Time Division

Time in LTE is divided into frames of 10 ms. Each frame consists of 10 subframes and each subframe consists of two slots. Each slot corresponds to 7 symbols.

Although the symbol length (which corresponds to the primitive data unit in the physical layer) is the smallest unit of time in LTE, no scheduling is ever done on a time period shorter than one subframe. Figure 2.3 illustrates the smallest scheduled unit, known as a *resource pair*. It consists of 1 subframe in time and 12 OFDM carriers in frequency. This adds up to $2 \cdot 7 \cdot 12 = 168$ symbols of data.

2.5 Channels

All traffic is divided into *channels*, which dictate the purpose of transmissions. While LTE supports both Time Division Duplexing (uplink and downlink interleaved over time) and Frequency Division Duplexing (uplink and downlink applied in separate frequency bands), we will focus on the case of FDD.

In an FDD cell the system frequency band is divided into two (not necessarily equal) parts, one for uplink channels and one for downlink channels. Within these parts, however, the actual channels are multiplexed both in time and frequency[10] (see figure 2.5).

There are 3 types of channels: *logical channels*, *transport channels* and *physical channels*. The MAC layer is responsible for mapping logical channels to transport channels, while the physical layer is responsible for mapping transport channels to physical ones. Channels are mapped to carrying channels in lower layers as illustrated in figures 2.4a and 2.4b.

2.5.1 Logical Channels

The MAC layer provides services to the RLC layer in the form of logical channels. Depending on what type of information a logical channel carries, they are divided into *control channels*, used for transmission of control information and configurations, and *traffic channels*, used for transmission of user data. The following logical channels are employed:

- *the Broadcast Control Channel* (BCCH) is used to transmit *system information*, which is broadcast to all terminals in the cell for configuration;

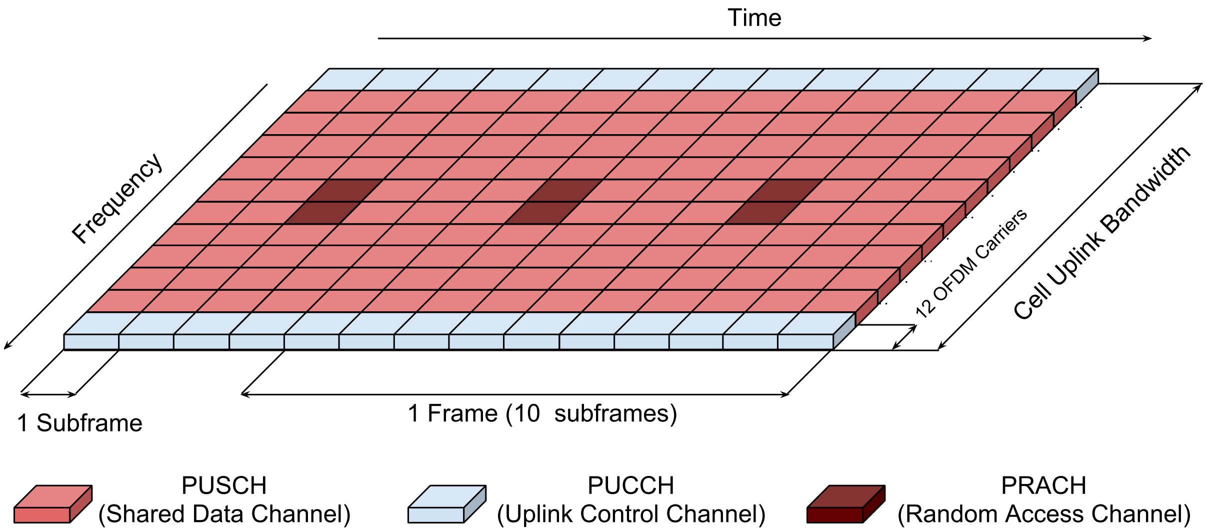


Figure 2.5: A simplified view of time/frequency multiplexing of physical uplink channels

- *the Paging Control Channel* (PCCH) is used for paging of the terminal, whose current location is unknown and therefore paging message needs to be transmitted in multiple cells;
- *the Common Control Channel* (CCCH) is used to transmit control information in conjunction with random access;
- *the Dedicated Control Channel* (DCCH) is used to transmit individual control information to/from a terminal;
- *the Multicast Control Channel* (MCCH) is used to transmit control information needed for reception of the *MTCH*;
- *the Dedicated Traffic Channel* (DTCH) is used to transmit all uplink and non-*MBSFN* (Multicast-Broadcast Single Frequency) downlink user data to/from a terminal;
- *the Multicast Traffic Channel* (MTCH) is used for downlink transmission of *MBMS* (Multimedia Broadcast/Multicast Service).

2.5.2 Transport Channels

The physical layer provides services to the MAC layer in the form of transport channels, which are defined by *how* and *with what characteristics* information are transmitted over the radio. All data transmitted over transport channels is divided into *TB* (Transport Blocks) by the MAC layer. The following transport channels are defined:

- *the Broadcast Channel* (BCH) is used to transmit parts of the BCCH system information, like *MIB* (Master Information Block), this channel has a fixed transport format;
- *the Paging Channel* (PCH) is used to transmit paging messages from PCCH logical channel, supports *DRX*;
- *the Downlink Shared Channel* (DL-SCH) is used to transmit downlink data and some parts of the *BCCH* system information not mapped to *BCH*.
- *the Multicast Channel* (MCH) is used for transmission of MBMS;
- *the Uplink Shared Channel* (UL-SCH) is similar to DL-SCH but used for transmission of uplink data;
- *the Random Access Channel* (RACH) used by the terminal to transmit random access preamble and some other configurations used in random access, it is only transport channel where transmission data aren't in the form of TB.

2.5.3 Physical Channels

The physical layer consists of the set of time-frequency resources used to transmit a particular transport channel, mapped to corresponding physical channel. Additionally, there are physical channels without corresponding transport channel, known as L1/L2 control channels. These are used for *DCI* (Downlink Control Information), which provides the terminal with necessary information for proper reception and decoding of downlink data, and *UCI* (Uplink Control Information), which provides scheduler and HARQ protocol with terminal status. The following physical channels are defined[11]:

- *the Physical Downlink Shared Channel* (PDSCH) is used for downlink unicast data transmission, even for transmission of paging messages;
- *the Physical Broadcast Channel* (PBCH) is used to transmit parts of the *SI*, needed by the terminal to access the network;
- *the Physical Multicast Channel* (PMCH) is used for MBSFN operation;
- *the Physical Downlink Control Channel* (PDCCH) is used to transmit downlink control information, namely scheduling decision needed for reception on the PDSCH and transmission on the PUSCH;
- *the Physical Hybrid-ARQ Indicator Channel* (PHICH) is used to transmit ACK or NAK, which indicates to the terminal whether TB should be retransmitted or not.
- *the Physical Control Format Indicator Channel* (PCFICH) is used to transmit information to terminals necessary to read PDCCHs;

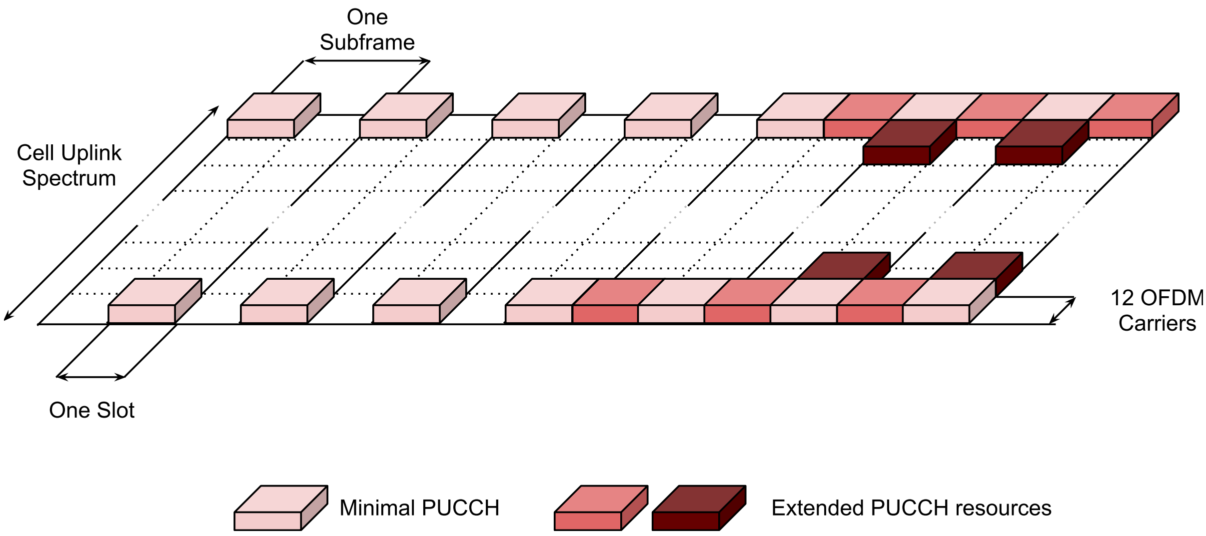


Figure 2.6: PUCCH scheduling

- *the Physical Uplink Shared Channel* (PUSCH) is similar to PDSCH but used for transmission of uplink data.
- *the Physical Uplink Control Channel* (PUCCH) is used by the terminal to transmit:
  - ACK, which indicates to the eNodeB whether TB should be retransmitted or not;
  - channel-state report aiding downlink channel-depending scheduling;
  - request for resources to transmit uplink data.
- *the Physical Random-Access Channel* (PRACH) is used by the terminal for random access.

2.6 Uplink Transmission And Scheduling

Our work hinges on uplink transmissions, and therefore we will focus on the PUCCH and PRACH control channels which are used to request scheduling for uplink transmissions.

2.6.1 PUCCH

The Physical Uplink Control Channel is used for terminals to provide status information such as channel quality indication. It is also used to request scheduling for uplink transmissions. The channel constantly occupies a variable number of resource blocks at

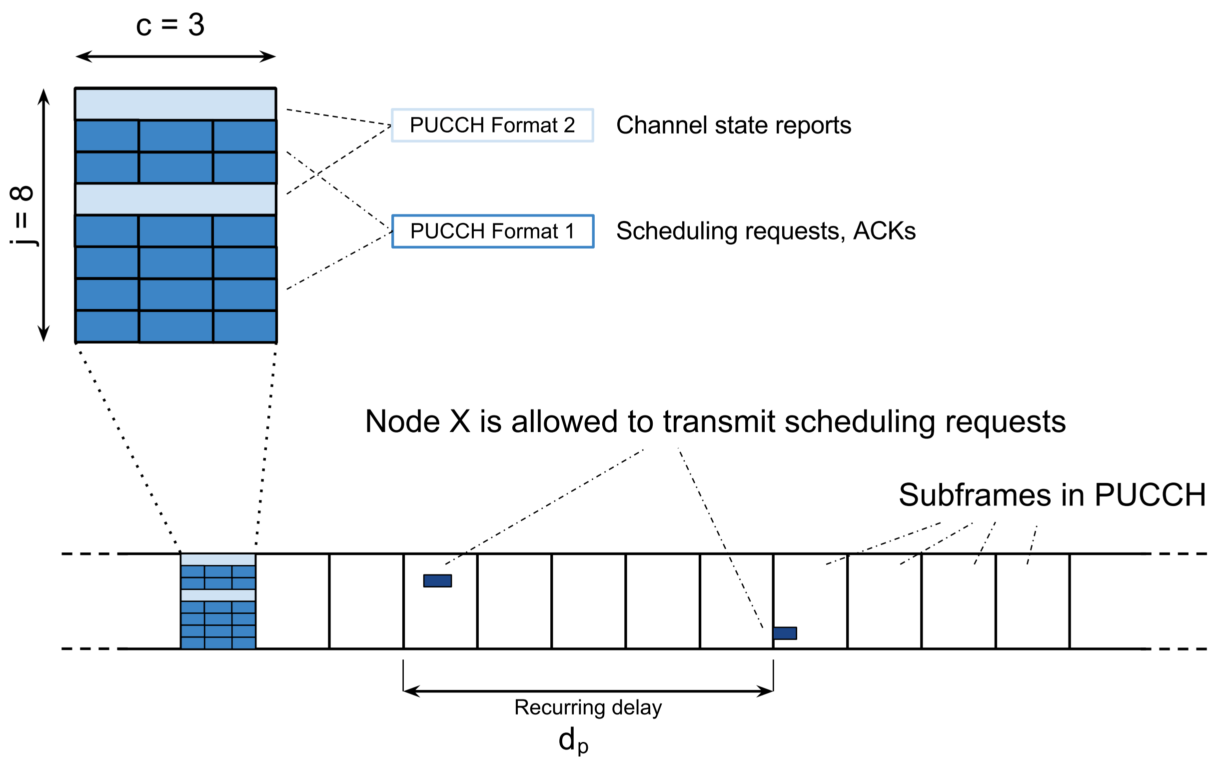


Figure 2.7: The delay of PUCCH

the top and bottom frequencies of the uplink spectrum (see figure 2.6). All connected and active nodes are scheduled in PUCCH in a round-robin fashion, regardless of their uplink transmissions. The pattern is illustrated in figure 2.7. This round-robin scheduling causes a delay d_p between PUCCH opportunities for a single node, which is proportional to the number of other active nodes in the cell. While connected the PUCCH resource allocated to a node is unavailable to all others, even if it goes unused. This means the resource will be wasted for the most part, if the connection time is too long and the terminal has only small amounts of data to send. It also means that this (connected but inactive) node will cause an increased PUCCH delay for all other nodes in the cell.

PUCCH Message Formats

On PUCCH, data is transmitted in one of three formats (named, appropriately, PUCCH formats 1, 2 and 3).

- F1 represents two bits of data, and can be used either as a scheduling request flag or as an ACK/NAK for received downlink data.
- F2 contains a channel status report, sent from the terminal to eNB so that the scheduler has information about channel quality, frequency selectivity, etc.

- F3 carries acknowledgements for three or more simultaneous downlink receptions on multiple component carriers[1].

Formats 1 and 2 are encoded using BPSK or QPSK, while F3 uses the same range of coding schemes as in PDSCH.

Since a subframe can carry more data than the two bits of an F1 message, PUCCH multiplexes messages into each subframe. This is done by two mechanisms, analogous to time and coding multiplexing. First, a set of $j \leq 12$ coding sequences (based on one sequence specific to the cell) is used to create j separate parts of the subframe. F1 messages are further multiplexed by c *cover sequences* creating a total of $j \cdot c$ *mini resources*. Typically, $c = 3$ cover sequences are used.

It's not always the case that all 12 coding sequences can be used (to increase diversity and reduce inter-cell interference, the number may be reduced), and we call the number used j. A subframe in PUCCH can thus be split into $j \cdot 3$ mini resources. Each such mini resource can hold a single F1 message. No terminal may send two messages (regardless of type) in the same subframe.

F2 messages can be mixed with F1 messages in the same PUCCH (though they are usually kept in separate time/frequency parts of PUCCH) and F2 messages occupy three mini resources, i.e. three times as many as an F1 message.

Figure 2.7 shows an example of F1 and F2 messages arranged in a PUCCH subframe.

2.6.2 PRACH

The Random Access Channel (RACH, carried in the physical PRACH) is used for unscheduled requests to transmit. This allows nodes which are connected to the network but not yet scheduled in PUCCH a chance to perform a first transmission.

A node X, to be able to receive or transmit data, needs to complete the following Random Access procedure:

1. X sends a Random Access Preamble in a predefined RA time slot on PRACH channel.

2. eNodeB sends a response containing a temporary identity, timing correction and a scheduling grant for subsequent steps.

3. X responds (on UL-SCH, using grant from step 2) with its own unique identifier.

4. eNodeB sends a final collision resolution message.

When these four steps have been completed, X is connected and has received a scheduling grant allowing it to transmit uplink data. Performing Random Access will unconditionally schedule node X in PUCCH in the future.

[1]Cells with multiple component carriers are not discussed in this thesis. Refer to [12] for more information.

PRACH is periodically recurring in short periods on a set number of frequencies in the uplink band, as illustrated in figure 2.5. We will refer to each of the single contiguous occurrences of PRACH as a *RACH opportunity*. Any number of nodes may use the same RACH opportunity, but each can use it for only one request.

RACH is governed by contention rather than a central scheduler. To mitigate the cost of collisions, the concept of *preambles* is employed.

RACH Preambles

A preamble can be thought of as a single slot of a RACH opportunity. A node which wishes to perform a RA request selects one of the defined preambles, partially based on the size of the pending data and partially at random.

There are a total of up to 64 different RA preambles defined for an LTE cell, and these are split into two subsets; the connecting node selects a preamble at random from one subset or the other depending on the size of the data it wishes to transmit. All 64 preambles are orthogonal and therefore will not interfere with other requests in the air. eNodeB can be configured to receive as many as 64 scheduling requests in the same RACH opportunity.

Preambles are differentiated using both time and coding, such that all 64 preambles can be occupied in the same RACH opportunity without colliding. The preamble chosen for a given transmission is thus used in the physical layer to determine transmission timing and to encode the transmitted request in a way orthogonal to all others.

A request which chose a preamble not used by any other is denoted as an *acquired* request, meaning its signal can be discerned by eNodeB for further processing.

Collision Resolution

If two or more nodes select the same preamble (and thus cause a collision in PRACH), there is a collision resolution scheme in place at eNodeB to recover at most one of the colliding requests. In the ideal case, two nodes which selected the same preamble will not fully collide (yielding no useful transmission), but rather be indistinguishable from each other. Step 4 in the Random Access Procedure above is intended to select one unique node out of all those who selected the same preamble.

When dealing with collision resolution in this thesis, we assume the ideal case where n colliding messages yield 1 acquisition and $n-1$ failures.

A request which was recovered as the "winner" of a collision is also denoted as *acquired*.

Servers

Due to timing constraints (responses must be sent within a fixed number of subframes) eNodeB has a limited capacity for handling the acquired requests during a single RACH opportunity. We characterize this limitation by limiting the number of *servers* which can decode successfully acquired RACH requests during one RACH opportunity. Combining

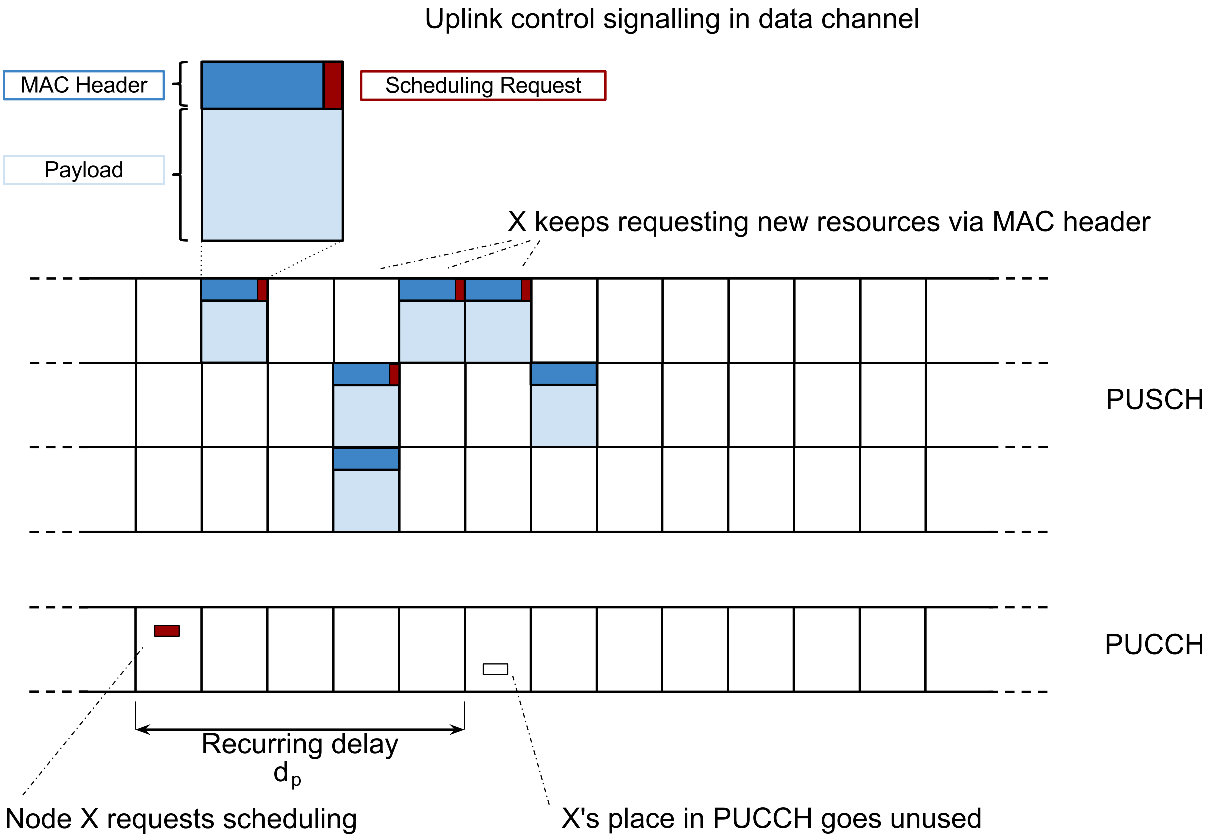


Figure 2.8: Semi-persistent scheduling requests in PUSCH

this limit of L servers with the limited number of K preambles available, the theoretical maximum number of successful RACH attempts in one opportunity is $min(K, L)$.

Any request which loses collision arbitration or fails to be assigned to an available server will be rejected and the corresponding node must retry at a later RACH opportunity.

2.6.3 PUSCH

LTE supports *Semi-persistent Scheduling*, which allows scheduling requests to be piggy-backed on data transmissions in the Uplink Shared Channel (PUSCH). In practice, this means that once scheduling has been successfully requested, a larger block of data can be transmitted contiguously without incurring further usage in the smaller control channels. For RACH this would ease the network load (if RACH was otherwise to be used for the subsequent transmissions), for PUCCH this would leave a PUCCH slot unused and therefore reduce the physical interference with other nodes using PUCCH.

Semi-persistent scheduling is implemented by a one-bit flag in the MAC header[13] of uplink data transmissions, as illustrated in figure 2.8

2.7 Summary

The LTE uplink has two channels used for scheduling, PRACH and PUCCH. These are used for the initial registration of a new terminal (PRACH random access) and then for every subsequent scheduling request. Typically, PRACH is used only for the initial Random Access and any later requests are made using PUCCH. This Random Access is the only time any collisions between terminals can occur, since PRACH is entirely contention-based and no scheduling is done before-hand.

Any connected terminal has a fixed recurring resource allotment in PUCCH, even if it is not used. After scheduling has been successfully requested (using either PRACH or PUCCH), the terminal receives reserved resources in the shared data channel PUSCH.

2.8 Reflection

Since PUCCH is shared between all *connected* terminals, regardless of uplink transmissions, it is easily saturated by an increasing number of terminals. The period d_p between two PUCCH resources for the same terminal (figure 2.7) depends on the total number of terminals currently connected to the eNodeB. The more terminals are connected the longer d_p will become for all of them, meaning new requests will be delayed.

In the case of WSNs, the number of terminals connected is potentially very high, even though the terminals don't necessarily transmit uplink very often. In such a network, the sensor nodes would waste lots of resources in PUCCH and therefore delay regular users with no added benefit. For this reason other operating modes should be considered, which allow the sensor nodes in an LTE network without the excessive scheduling overhead.

Our suggestion to make LTE more suitable for WSN use (and reduce the adverse impact on regular users) is to allow sensor nodes to use random access exclusively, without being scheduled in PUCCH. LTE currently does not allow such a connection pattern. This would be beneficial in the case where sensor nodes have a low transmission frequency and therefore would waste a lot of resources in PUCCH by their sheer numbers.

Chapter 3

LTE-WSN Model

3.1 Introducing a Mathematical RACH Model

RACH in LTE is a channel used to initiate a random access procedure, as described in Chapter 2. We proposed to also use this channel instead of PUCCH to send requests for uplink data transmission. To further investigate the feasibility of this, we need a mathematical model of RACH behavior. At closer inspection one can find modelling of RACH similar to that of multichannel slotted ALOHA, where preambles of RACH, which divide transmissions orthogonally, can be seen as a number of independent S-ALOHA channels. For this reason we base our model on multichannel slotted ALOHA model proposed by Zhao Liu and Magda El Zarki [1]. A number of modifications was made to suit the model to RACH of LTE. One of the major changes is the implementation of *collision resolution*. We also extend the model to include our notion of *servers*. These subjects are described in closer detail in Chapter 2.

This chapter provides a step-by-step description of how we arrive at a model and what parameters and metrics are used.

3.2 Model Definition

The model describes a number of participating nodes N in the network. At the beginning of each RACH opportunity, each node starts in one of two following modes: *I-mode* (Idle), where nodes transmit a new message with probability p_N or *B-mode* (Backlogged), where nodes retransmit old message with probability p_R.

All transmitting nodes from I- and B-mode each choose one of K available preambles. Some number of nodes will collide (i.e. choose the same preamble as another node) and, if collision resolution on eNodeB is disabled, return to B-mode. In case of enabled collision resolution, one node for each collided preamble will be successfully acquired by eNodeB together with those nodes which selected unique preamble.

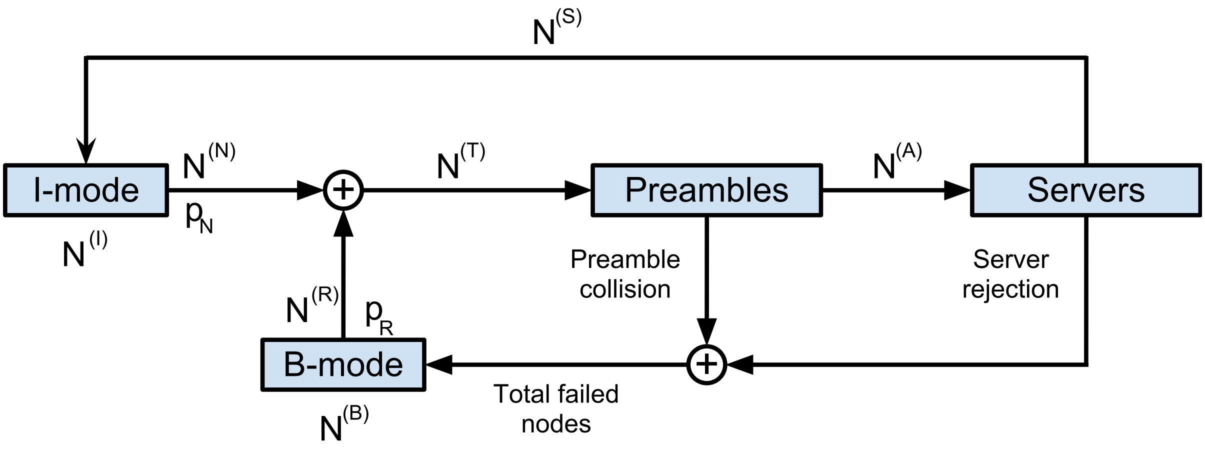


Figure 3.1: A visual illustration of the theoretical RACH model

Once acquired, just a limited number L messages can be successfully decoded by eNodeB on the same RACH opportunity (due to hardware and time constraints). This can be visualized as a set number of L parallel servers serving the acquired nodes with no queuing. All messages (at most L) which are served by eNodeB in this secondary selection stage are considered successful, i.e. have been successfully scheduled for transmission on eNodeB and the corresponding nodes return to I-mode. The remaining messages are rejected and the corresponding nodes are returned to B-mode.

It is important to note that the model is seen from the network perspective, i.e. it is possible to calculate the probability that particular number of users being in particular state, but not possible to track nodes separately (this would require a node based model, see [14]).

Figure 3.1 helps to visualize the model and table 3.1 summarizes all values used and their descriptions.

3.3 Mathematical Relations

From the following properties of the defined model, useful mathematical relations can be derived.

A static number of nodes are in circulation between I- and B-mode, i.e. the total number of nodes does not change. Therefore, the following statement holds true:

$$N = N_k^{(I)} + N_k^{(B)}. \tag{3.1}$$

Further, due to transmission and retransmission probabilities:

$$N_k^{(N)} = N_k^{(I)} \cdot p_N,$$

Name	Description
N	The total number of nodes in the network
$N_k^{(I)}$	The number of nodes in I-mode
$N_k^{(N)}$	The number of nodes performing a new transmission
$N_k^{(B)}$	The number of nodes in B-mode
$N_k^{(R)}$	The number of nodes performing a retransmission
$N_k^{(T)}$	The total number of transmitting nodes
$N_k^{(A)}$	The number of acquired nodes
$N_k^{(S)}$	The number of nodes which successfully requested scheduling
p_N	The probability that a node in I-mode transmits
p_R	The probability that a node in B-mode retransmits
K	The total number of available preambles
L	The total number of servers available on eNodeB

Table 3.1: Describes the values used in suggested RACH model, where k is index of RACH opportunity.

$$N_k^{(R)} = N_k^{(B)} \cdot p_R.$$

The total number of transmitting users during RACH opportunity is:

$$N_k^{(T)} = N_k^{(N)} + N_k^{(R)}. \tag{3.2}$$

Due to preamble selection of transmitting nodes and server selection of acquired nodes, the following must hold true:

$$N_k^{(S)} \leq N_k^{(A)} \leq N_k^{(T)}.$$

Due to node's transition from mode to mode properties:

$$N_{k+1}^{(B)} = N_k^{(B)} - N_k^{(R)} + failed\_nodes_k,$$

and

$$failed\_nodes_k = N_k^{(N)} + N_k^{(R)} - N_k^{(S)},$$

we can derive the following:

$$N_{k+1}^{(B)} = N_k^{(B)} - N_k^{(R)} + N_k^{(N)} + N_k^{(R)} - N_k^{(S)}$$

$$N_{k+1}^{(B)} - N_k^{(B)} = N_k^{(N)} - N_k^{(S)}. \tag{3.3}$$

3.4 Metrics

Values of system throughput and average delay are needed to evaluate performance of this mathematical model and helps to compare results with simulator implemented for model verification. See Chapter 4 for detailed description of how simulator works.

3.4.1 Throughput

While $N_k^{(S)}$ is the number of successful transmissions during one RACH opportunity, an average throughput can be defined as a mean value of $N_k^{(S)}$ over each RACH opportunity:

$$S = E[N^{(S)}].$$

Further, the throughput is normalized as S_0, which makes it easier to compare different sets of parameters. The throughput value is normalized against the maximum possible throughput, which in turn is limited by either the number of preambles or the number of servers:

$$S_0 = \frac{S}{S_{max}} = \frac{S}{min(K, L)}.$$

3.4.2 Delay

An average delay, D is the estimated time taken for a newly generated message to be successfully scheduled by eNodeB. It is important to note that after a message being scheduled there is some more time to wait and then perform the transmission itself, which is not included in D. The delay is estimated using the same method as Liu et al. in [1]:

$$D = \frac{N}{S} - \frac{1}{p_N}, \tag{3.4}$$

Appendix A describes the calculation in detail.

3.5 Analysis

The throughput, as defined above, can be further described using probability theory as

$$\begin{aligned} S &= E_{N^{(T)}}[E_{N^{(S)}|N^{(T)}}[N^{(S)}]] \\ &= \sum_{n=0}^{N} \left(\sum_{s=0}^{Min(n,K,L)} s \cdot f_{N^{(S)}|N^{(T)}=n}(s) \right) \cdot f_{N^{(T)}}(n). \end{aligned} \tag{3.5}$$

$E_{X|Y}[X]$ notation, in the equation 3.5, denotes the expected value of the variable X with respect to conditional distribution $f_{X|Y}$, lets substitute it with Z for now. $E_Y[Z]$ in order denotes the expected value of the variable Z with respect to conditional distribution f_Y.

$f_{N^{(S)}|N^{(T)}=n}(s) \stackrel{\text{def}}{=} Pr(N^{(S)} = s|N^{(T)} = n)$ notation denotes the conditional distribution of successfully transmitted nodes given the total number of transmitting nodes in a RACH opportunity. This can be also described as a function of the probability that precisely s nodes succeed given a number n transmitting nodes.

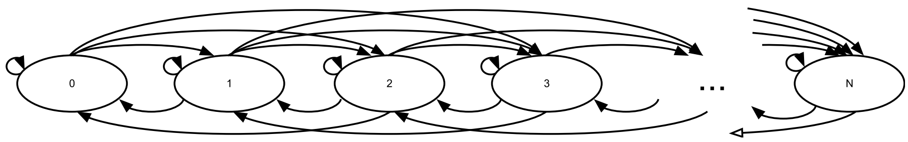


Figure 3.2: An illustration of the Markov Chain representation of the RACH model. Examples of state transitions given by arrows.

$f_{N^{(T)}}(n) \stackrel{\text{def}}{=} Pr(N^{(T)} = n)$ notation denotes the equilibrium distribution of the total number of transmitting nodes in a RACH opportunity. This can be also seen as a function of the probability that precisely n nodes transmit. Furthermore, it can be expressed in terms of equilibrium distribution of *Markov chain*.

3.5.1 Markov Chain Definition

The model described above can be represented as a Markov Chain (figure 3.2), where each state i corresponds to $N_k^{(B)} = i$, i.e. exactly i nodes are backlogged. This Markov Chain has a natural limit to the number of states, as the number of backlogged messages cannot exceed the total number of nodes N. Furthermore, from any state i, all states of larger denomination $i+1, i+2, ..., N$ are within a reach, since any number of new messages may arrive at the same time.

However, there is a limit to how many steps *"back"* the state machine can take in one slot, i.e. node's transition from B-mode to I-mode. As we can recall from the model definition, node can transition back to I-mode only in case of successful transmission. Moreover, the number of nodes successfully transmitted in one RACH opportunity can not exceed $Min(K, L)$ value, as such nodes must succeed in both selection stages.

The one-step transition probability matrix P of this Markov Chain can be constructed. The matrix is of dimensions $(N+1) \times (N+1)$. Each entry of the matrix represent the probability of the network transitioning from one state to another, which can be described as

$$p_{ij} = Pr(N_{k+1}^{(B)} = j | N_k^{(B)} = i), \tag{3.6}$$

where i and j are source and destination states respectively.

The derivation of the following one-step transition probability is presented in Appendix B:

$$p_{ij} = \sum_{n=0}^{N} \sum_{s=0}^{Min(K,n)} \binom{i}{n-j+i-s} \cdot p_R^{(n-j+i-s)} (1-p_R)^{j-n+s} \times \binom{N-i}{j-i+s} p_N^{(j-i+s)} (1-p_N)^{N-j-s} \times f_{N^{(S)}|N^{(T)}=n}(s). \tag{3.7}$$

As this is a Markov Chain, it has the Markovian property that the sum of all transition probabilities from a state i is equal to 1, i.e. the sum of probabilities in each column of the matrix P is equal to 1.

3.5.2 Equilibrium Distribution

With defined probability matrix P lets return to $f_{N^{(T)}}(n)$, which can be described according to equilibrium distribution of Markov chain as

$$\begin{aligned} f_{N^{(T)}}(n) &= \sum_{i=0}^{N} Pr(N^{(T)} = n | N^{(B)} = i) \cdot \pi_i \\ &= \sum_{i=0}^{N} \left(\sum_{m=Max(0,n-i)}^{Min(n,N-i)} \binom{i}{n-m} \cdot p_R^{n-m} \cdot (1-p_R)^{i-n+m} \right. \\ &\qquad \left. \times \binom{N-i}{m} \cdot p_N^m \cdot (1-p_N)^{N-i-m} \right) \cdot \pi_i. \end{aligned} \tag{3.8}$$

π is a stationary distribution and satisfies the set of linear equations:

$$\begin{cases} \pi = \pi P, \\ \sum\limits_{k=0}^{N} \pi_k = 1. \end{cases} \tag{3.9}$$

With a known probability matrix P the above system of equations can be solved for π, as a number of equations in P equals to the number of solutions which is π.

Equilibrium distribution $f_{N^{(T)}}$, as already been mentioned, defines the number of transmitting nodes in a RACH opportunity after the system has stabilized.

3.5.3 Blocking Probabilities

Due to preambles and servers mechanisms, it is not always the case that a transmitted message arrives successfully. Therefore, it is necessary to first find the probability of getting through preamble selection stage and then of server acceptance. The whole selection process is defined by $f_{N^{(S)}|N^{(T)}=n}(s)$, which is conditional distribution of successfully transmitted nodes (see the beginning of this section).

Due to the partition condition:

$$\sum_{a=0}^{Min(n,K)} Pr(N^{(A)} = a | N^{(T)} = n) = 1,$$

the conditional distribution can be divided in two parts:

$$\begin{aligned} f_{N^{(S)}|N^{(T)}=n}(s) &\stackrel{\text{def}}{=} Pr(N^{(S)}|N^{(T)}=n) \\ &= \sum_{a=0}^{Min(n,K)} Pr(N^{(S)}=s|N^{(A)}=a, N^{(T)}=n) \times Pr(N^{(A)}=a|N^{(T)}=n) \\ &= \sum_{a=0}^{Min(n,K)} f_{N^{(S)}|N^{(A)}=a,N^{(T)}=n}(s) \times f_{N^{(A)}|N^{(T)}=n}(a). \end{aligned} \tag{3.10}$$

Preamble Selection

In the equation 3.10, $f_{N^{(A)}|N^{(T)}=n}(a)$ is a conditional distribution of acquired messages given n transmissions in a single RACH opportunity. This can be also described as a probability that exactly a messages get acquired given that n were transmitted.

As already been mentioned before, there is slightly different algorithm to determine which nodes get acquired. The one is *with* collision resolution and the other is *without*. The mathematical derivations will be different because of differences in functionality.

While collision resolution is disabled, the distribution can be described as (derivation is present in Appendix C.1.2)

$$f_{N^{(A)}|N^{(T)}=n}(a) = \begin{cases} \dfrac{\binom{K}{a} \sum\limits_{i=0}^{Min(\lfloor \frac{n-a}{2} \rfloor, K-a)} \binom{K-a}{i}\binom{n-a-i-1}{n-a-2i}}{\binom{n+K-1}{n}} & \text{if } 0 \le a \le min(n,K), \\ 0 & \text{otherwise.} \end{cases} \tag{3.11}$$

While collision resolution is enabled, the distribution takes a different shape (the derivation is present in Appendix C.1.1):

$$f_{N^{(A)}|N^{(T)}=n}(a) = \begin{cases} \frac{\binom{K}{a}\binom{n-1}{n-a}}{\binom{n+K-1}{n}} & \text{if } 0 \le a \le min(n,K), \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

Server Acceptance

In the equation 3.10, $f_{N^{(S)}|N^{(A)}=a,N^{(T)}=n}(s)$ is a conditional distribution of successfully transmitted nodes given a acquired and n transmitted nodes in a RACH opportunity. In other words, the probability that exactly s nodes successfully transmit their messages given a acquired and n transmitting nodes.

Following equation describes this distribution (the derivation present in D):

$$f_{N^{(S)}|N^{(A)}=a,N^{(T)}=n}(s) = \begin{cases} 1 & \text{if } s = min(a,L), \\ 0 & \text{otherwise.} \end{cases} \tag{3.13}$$

3.6 Summary

An equation 3.11 or 3.12 (depending on collision resolution is enabled or not) for preamble selection and an equation 3.13 for server acceptance can be put together in equation 3.10 to describe a conditional distribution of successfully transmitted nodes $f_{N^{(S)}|N^{(T)}=n}(s)$.

While conditional distribution is known, it is now possible to determine one-step transition probability matrix P by resolving equation 3.7. This in turn allows to determine a stationary distribution π by using equation 3.9.

Further, the equilibrium distribution $f_{N^{(T)}}(n)$ can be determined by putting respective values of stationary distribution π into equation 3.8. Finally, conditional distribution $f_{N^{(S)}|N^{(T)}=n}(s)$ and equilibrium distribution $f_{N^{(T)}}(n)$ can be substituted in equation 3.5 to find the system throughput.

The system throughput and the average delay, which can be found by equation 3.4, are in turn used to evaluate the performance of the proposed mathematical model. This allows to compare results with a simulator whose functionality is described in Chapter 4.

CHAPTER 4

Verification by Simulation

4.1 Event Simulation

In order to verify our mathematical RACH model, a second data set is needed. We acquire this by implementing a simulation of the behaviour of RACH. The key points are, in order of execution:

1. Packet generation at nodes
2. Selection of transmitting nodes
3. Preamble selection for each node
4. Detection of preamble collisions
5. Resolution of preamble collisions
6. Server processing of acquired nodes

This process precisely adheres to the modelled process of RACH shown in figure 3.1. Implicit in stages 5 and 6 are mode transitions of single nodes; failures place the affected node in B-mode and acceptance from a server places the node in I-mode for the next RACH cycle.

4.2 Structure of the RACH Simulator

The stages outlined above are executed in order, in a cyclic fashion. At the end of each cycle, a set of output values have been recorded and the internal state has been prepared for the next cycle.

Node №	p_N	p_R	Mode	Arr.	Tx	Preamble №	Acq.	Succ.	Tx count	Pkt buffer
1	0.02	0.10	1	0	1	8	1	1	0	0
2	0.01	0.10	0	1	1	8	0	0	1	1
3	0.02	0.20	1	0	1	12	1	0	23	3
4	0.01	0.10	0	0	0	0	0	0	1	1

Table 4.1: An example state table with 4 nodes

Name	Description
Node №	Node ID
p_N	Transmission probability $0 \ldots 1$
p_R	Retransmission probability
Mode	Mode. 1 for B-mode, 0 for I-mode
Arr.	Arrival. 1 if a packet arrived this interval
Tx	Transmission flag. 0 if node waits, 1 if it attempts to transmit
Preamble №	Preamble selected. $1 \ldots K$ or 0 for no preamble
Acq.	Acquired flag. 1 if this node successfully selected a preamble this interval
Succ.	Success flag. 1 if this node was successfully handled by a server
Tx count	Current number of transmission attempts
Pkt buffer	Current number of buffered packets

Table 4.2: The columns of the running state table

4.2.1 Running State

The state of the simulation is kept in an $N \times 11$ matrix (Table 4.1) where each row describes a node and the columns are used as described in Table 4.2.

For performance, the vast majority of the calculations performed during a simulation cycle are done using vector operations on entire columns of the state matrix. As a result of this design, the state matrix is re-sorted in its entirety several times during each cycle. This still yields acceptable performance; a network with 5000 nodes can be simulated in real-time on a single 3.0GHz processor core.

4.3 Metrics

During execution, a number of values are logged for further processing, plotting, etc. These values are outlined in Table 4.3. Most of these values are saved at each time interval and are thus stored as vectors of values. Such outputs are of a length equal to T.

The D output is different in this respect, as its length depends on the number of successful transmissions during the simulation.

Name	Description
Node №	Node ID
T	The time, in seconds, of each time interval
N_B	The number of nodes in B-mode at the start of the interval
N_I	The number of nodes in I-mode at the start of the interval
N_N	The number of new transmissions ($N^{(N)}$)
N_R	The number of retransmissions ($N^{(R)}$)
N_P	The number of preamble failures
N_A	The number of acquired nodes ($N^{(A)}$)
N_F	The number of server rejections
N_S	The number of successful transmissions ($N^{(S)}$)
D	Delay and transmission count of each successful message

Table 4.3: The various metrics measured by the RACH simulator

4.4 Execution

4.4.1 Stages of an Execution Cycle

Stage 1: Packet Generation

Packets are generated at nodes which are in I-mode. This is done according to each node's packet arrival probability p_N with a uniform random distribution. The arrival time (i.e. the sequential number of the current time interval) is recorded in the affected node's buffer for each arrived packet.

Stage 2: Selection of transmitting nodes

Any node in I-mode which has seen a packet arrival will attempt transmission. Nodes in B-mode are selected for transmission based on p_R, using a uniform random distribution. All transmitting nodes have their retransmission counter increased.

Stage 3: Preamble Selection

Each transmitting node selects an integer in the range $[1...K]$ and uses it for a preamble. The selection is done at random with a uniform distribution. Non-transmitting nodes are given the invalid preamble number 0.

Stage 4: Detection of Preamble Collisions

The nodes are sorted based on their selected preamble, using the current retransmission count as a secondary sorting key. The number of nodes using each preamble are counted. If the count exceeds 1, there is a collision on this preamble.

Stage 5: Resolution of Preamble Collisions

At this stage, behaviour differs if collision resolution is enabled. In such a simulation, the top node using each preamble is acquired and the rest fail and are returned to B-mode. Since the nodes were secondarily sorted by retransmission count the node with the most previous attempts prevails.

If collision resolution is not enabled, all nodes using colliding preambles are failed and returned to B-mode. Any node using a unique preamble is marked as acquired.

Stage 6: Server Assignment

The final limiting stage allows only a number L of the acquired nodes to succeed. Since the nodes are already sorted in order of retransmission count, the ones with the largest count will prevail.

4.4.2 Mode Transitions

At the end of such a cycle, a number of nodes have transitioned from I-mode to B-mode (via failures in stages 5 or 6), from I-mode to I-mode (by successful transmission), from B-mode to I-mode (via successful retransmission) and from B-mode to B-mode (via failed retransmission).

4.4.3 Test Conditions

To verify the various calculations performed in a cycle, the following failure conditions are evaluated. Failure to satisfy any one condition aborts the simulation, as it would indicate a critical implementation error.

Pre-check

- $B + I = N$, All nodes are still accounted for.

Stage 1: Packet Generation

- All packet arrivals were to nodes in I-mode

Stage 2: Selection of transmitting nodes

- All transmitting nodes have a pending packet

Stage 3: Preamble selection

- All chosen preambles are valid $[1, K]$

Stage 5: Resolution of preamble collisions

- All winners have a unique preamble
- If collisions are to be resolved, every chosen preamble has an associated winner

Stage 6: Server Assignment

- All winners were previously acquired
- All winners were previously transmitting

4.5 Support for Packet Buffering

To align the simulator further with reality and allow comparison to existing simulators, the simulator was amended to allow buffering of multiple packets at each node.

This affects two parts of the simulator: Packet generation and Processing of successful packets.

In this buffered mode, all nodes experience packet arrivals (as opposed to just nodes in I-mode.) Packets arrive using the same uniform random distribution, applied to each node's p_N at the start of each time interval.

Any node in I-mode remains in I-mode until there is a packet in the buffer, at which point it immediately transmits.

The packet buffer of each node is implemented as a fixed-size (length B) vector storing arrival time of each packet. These buffers are stored together in a single $N \times B$ matrix encompassing all the nodes of the network.

Any packet arriving at a node with a full buffer is recorded in a counter for that node and then immediately dropped. Dropped packets are counted for each node and reported with the rest of the output.

At the end of the cycle, successfully transmitted packets (at most one per node) are removed from the front of their respective buffers and the remaining buffer data is shifted forwards.

Successfully transmitted packets are logged in the output as two values describing the delay (in time intervals since first transmission attempt) and the number of transmission attempts leading up to the success.

Chapter 5

Modification of the LTE-WSN Model

5.1 Preliminary Simulation Results

Initially, comparing the Markov model to simulated results indicated a large mismatch between predicted and observed values, as shown in figure 5.1 (where $K = 16$, $L = 16$, $p_N = 0.61$, $p_R = 1.0$).

Further investigation shows that the estimation of the number of transmitting users coincides very well, while the estimated number of acquired nodes does not. This can also be seen in figure 5.1. While it illustrates throughput, $L = 16$ is not a limiting factor and thus $N^{(A)}$ has the exact same behavior.

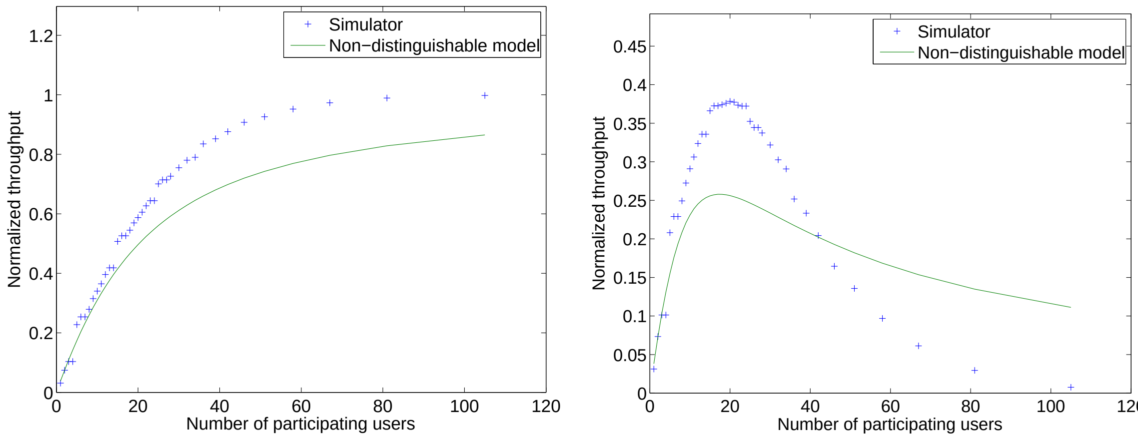
The reason for this mismatch in $E[N^{(A)}]$ is in an assumption in [1] where the channel selection problem is mapped to the common *balls and urns* paradigm. The method derived explicitly uses *distinguishable urns* and *indistinguishable balls*.

This assumption leads to an incorrect probability distribution $f_{N^{(A)}|N^{(T)}}(a)$ due to a subtle difference in the way permutations are counted.

5.2 Distinguishable Messages

Table 5.1a shows the possible ways to distribute 2 *indistinguishable* messages over 3 *distinguishable preambles* P1, P2 and P3. Note that we only care about *how many* messages are using each preamble, not which specific message(s). There are a total of 6 permutations, 3 of which have $a = 1$ and 3 of which have $a = 2$. With this distribution, $f_{N^{(A)}|N^{(T)}}(1) = 3/6 = 0.5$.

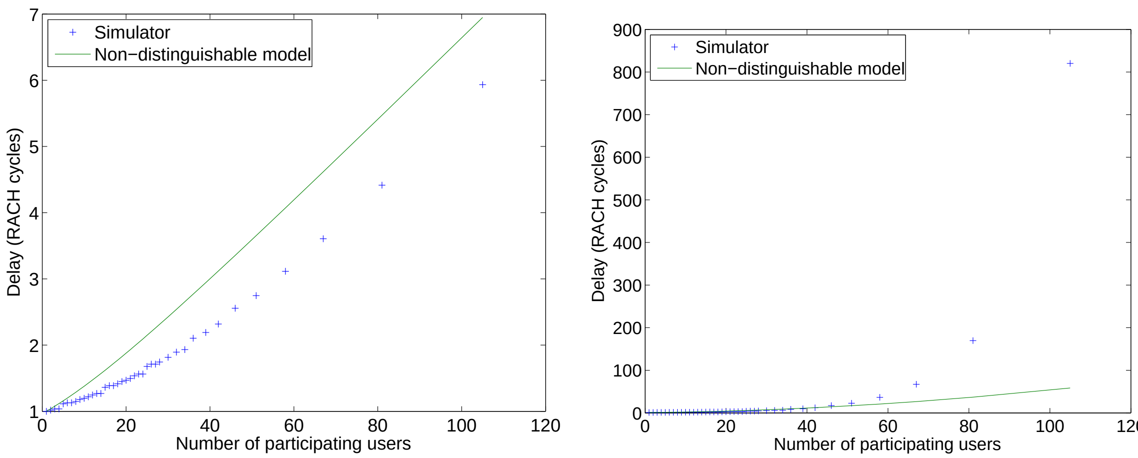
With *distinguishable messages* the specific messages also matter, as illustrated in table 5.1b. Again, 2 messages are distributed over 3 preambles. This yields a total of 9 permutations, 3 of which have $a = 1$ and 6 of which have $a = 2$. With this distribution,



(a) *With collision resolution*

(b) *Without collision resolution*

Figure 5.1: Predicted throughput compared to observed throughput



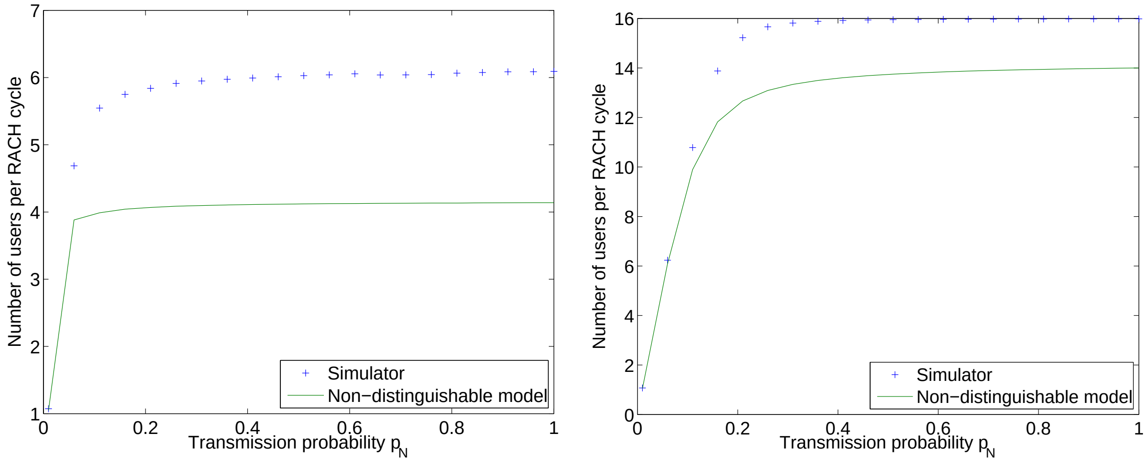
(a) *With collision resolution*

(b) *Without collision resolution*

Figure 5.2: Predicted average delay compared to average observed delay

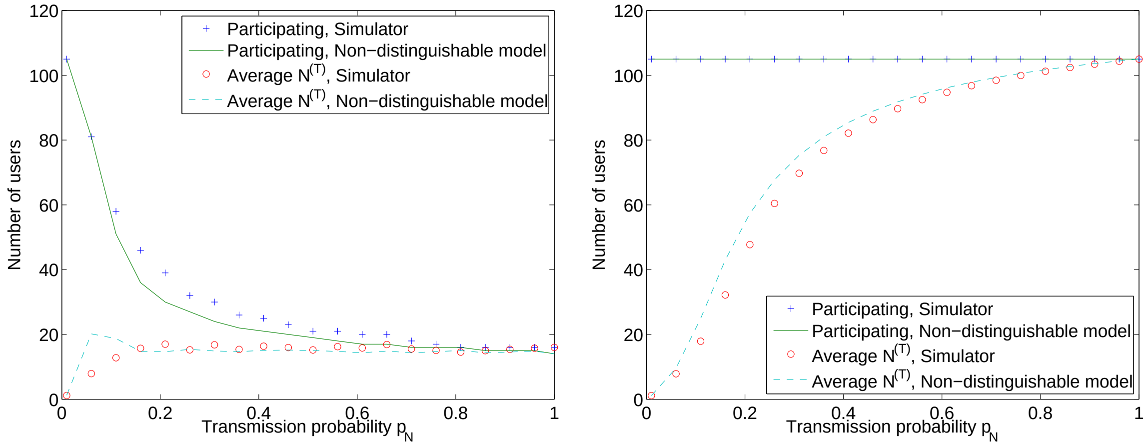
$f_{N^{(A)}|N^{(T)}}(1) = 3/9 = 0.33$.

The source of the error is that the old model assumes that all the cases (rows) in table 5.1a occur with the same probability. When nodes select preambles with a uniform probability distribution, however, this is not the case. It is clear from both theory and observed simulation results that the model must count the messages as *distinguishable*.



(a) Without collision resolution   *(b) With collision resolution*

Figure 5.3: Peak throughput for various transmission probabilities



(a) Without collision resolution   *(b) With collision resolution*

Figure 5.4: N and $N^{(T)}$ at peak throughput for each transmission probability

5.2.1 Modifications

With this new assumption of *distinguishable messages*, Appendix C.2.2 shows how we can redefine the conditional distribution for $N^{(A)}$ with collision resolution disabled as

# of Msgs using P1	# of Msgs using P2	# of Msgs using P3	a
0	0	2	1
0	2	0	1
2	0	0	1
0	1	1	2
1	1	0	2
1	0	1	2

(a) Cases with indistinguishable messages

Msgs using P1	Msgs using P2	Msgs using P3	a
0	0	1,2	1
0	1,2	0	1
1,2	0	0	1
0	1	2	2
1	2	0	2
1	0	2	2
0	2	1	2
2	1	0	2
2	0	1	2

(b) Cases with distinguishable messages

Table 5.1: The difference between distinguishable and indistinguishable messages

$$\begin{aligned} W(K,a,n) &= W(K-1,a,n) + \\ &+ n \cdot W(K-1,a-1,n-1) + \\ &+ \sum_{i=2}^{n-a} \binom{n}{i} \cdot W(K-1,a,n-i) \\ f_{N^{(A)}|N^{(T)}}(a) &= \frac{W(K,a,n)}{K^n}. \end{aligned} \tag{5.1}$$

Analogously, as shown in Appendix C.2.1, we arrive at the distribution with distinguishable messages and collision resolution enabled:

$$\begin{aligned} V(a,n) &= \sum_{i=1}^{n-(a-1)} \binom{n}{i} \cdot V(a-1,n-i) \\ f_{N^{(A)}|N^{(T)}}(a) &= \frac{\binom{K}{a} \cdot V(a,n)}{K^n}. \end{aligned} \tag{5.2}$$

CHAPTER 6

Results

6.1 Adapting the WSN scenario

To tie our model to a realistic WSN deployment scenario we define a network divided into *zones* around an event source, as illustrated in figure 1.1. Each zone samples the environment at a set frequency and thus each node in the zone generates traffic at that frequency. A network consists of a number of zones, each with a number of nodes and a transmission frequency common to all those nodes.

As described in Section 3, our RACH model requires packet arrivals to be described by the *transmission probability* p_N. This is the probability that a given node sees a packet arrival in a given RACH opportunity.

Such a p_N can be seen as a frequency in RACH periods rather than Hz, and converting between the two only requires knowledge of the time between two RACH periods. LTE allows configuration of this value between 1 and 20 ms. As an example in an LTE cell with a RACH period of 20 ms, a zone where transmissions occur at 5 Hz would have a $p_N = 5 \cdot 20 \cdot 10^{-3} = 0.1$.

Further, since the model only works with a single p_N which is uniform across all N nodes, such a collective value needs to be calculated for an arbitrary set of zones. This is simply a mean, weighted by the number of users in each zone. For example consider a WSN with x zones $1 \leq z \leq x$, each with N_z users and a transmission probability of $p_{N,z}$. Such a network could be simplified as a single mass of nodes where

$$N = N_1 + N_2 + \ldots + N_x$$

$$p_N = \frac{p_{N,1} \cdot N_1 + p_{N,2} \cdot N_2 + \ldots + p_{N,x} \cdot N_x}{N_1 + N_2 + \ldots + N_x}$$

Name	Description	Value
N	Number of nodes	1-103
K	Number of preambles	16
L	Number of servers	16
p_R	Retransmission probability	1.0

(a) Numerical parameters used

	Portion of nodes	Transmission probability p_N
Zone 1	10%	1.0
Zone 2	30%	0.7
Zone 3	60%	0.5
Total	100%	0.61

(b) The zones used

Table 6.1: A summary of parameters used in tests

6.2 Parameters

Unless otherwise noted, the results shown in this chapter are based on the parameters shown in Tables 6.1a and 6.1b.

All averages from the simulator are taken over 10 minutes with a RACH period of 20ms (yielding $3 \cdot 10^4$ RACH opportunities).

To yield the most informative datasets possible, the density of sampled values is higher at lower values of N. This allows us to better capture the peak and drop-off features, while not wasting too much computation on the stabilized tail of the curve. The values used for N are generated based on the time consumption of our model in a single pilot run, and range from 1 to 103.

The number of preambles and servers are tricky. As described in Chapter 2, we have as much as 64 preambles to operate with. However, we limit this number to 16 for WSN nodes with purpose of mitigating collision problems for other, none-WSN, participants of the LTE network. Further, it was decided that using as much servers as preambles is most optimal, though we tried different sets of these parameters and did not notice any abnormal deviations.

To link the model with WSN we consider the following zone network. *Zone 1* consists of 10% of all participating nodes N, which is 10 for $N = 103$. These nodes transmit with probability 1.0, which is every RACH opportunity. In *Zone 2* we have 20%, which is 31 nodes and they transmit with probability 0.7. Finally, *Zone 3* has 60%, 62 nodes, transmitting with the least probability - 0.5. As described in section 6.1 we can derive an average probability of all arriving to eNodeB messages, which is 0.61. This value is further used by the mathematical model and the simulator to derive throughput and delay values.

Retransmission probability is kept high relative to transmission probability, otherwise nodes will be stuck in *B-mode* for considerably long time. For this reason we set retransmission probability to 1.0, which is as high as transmission probability in *Zone 1*.

6.2.1 Scaling of Parameters

As the model is heavily reliant on calculating the number of permutations in several steps, calculation time can be quite long. Furthermore, some temporary values in mid-

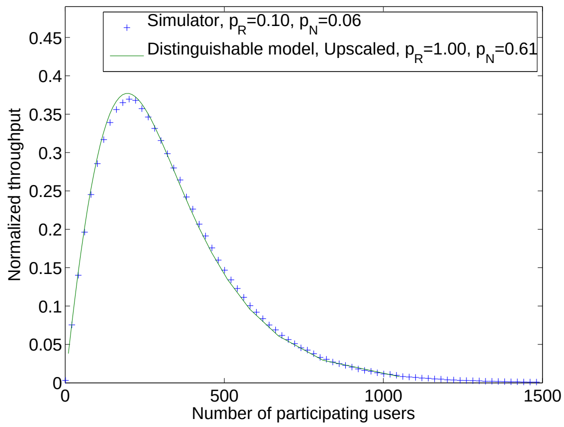


Figure 6.1: Illustration of up-scaling model results

calculation would exceed the range of an IEEE754 *double* even for $N = 130$. These limitations prevent the model from being applied practically for larger numbers of users. In use, however, one is typically interested in N as large as several thousand users or more.

In order to remedy this problem and cut down on calculation time, we scale the total number of users down and increase the transmission and retransmission probabilities to match. Our assumption is that such results would scale up to the desired number of users. To verify that this is correct, we compare a scaled-down modelled prediction with a full-scale simulation.

Figure 6.1 demonstrates our comparison. The simulator was run with full-scale parameters $K = 16$, $L = 16$, $p_R = 0.1$, $p_N = 0.061$, $N_{max} = 1500$. We predict this result by applying our model to scaled parameters $K = 16$, $L = 16$, $p_R = 1.0$, $p_N = 0.61$, $N_{max} = 103$.

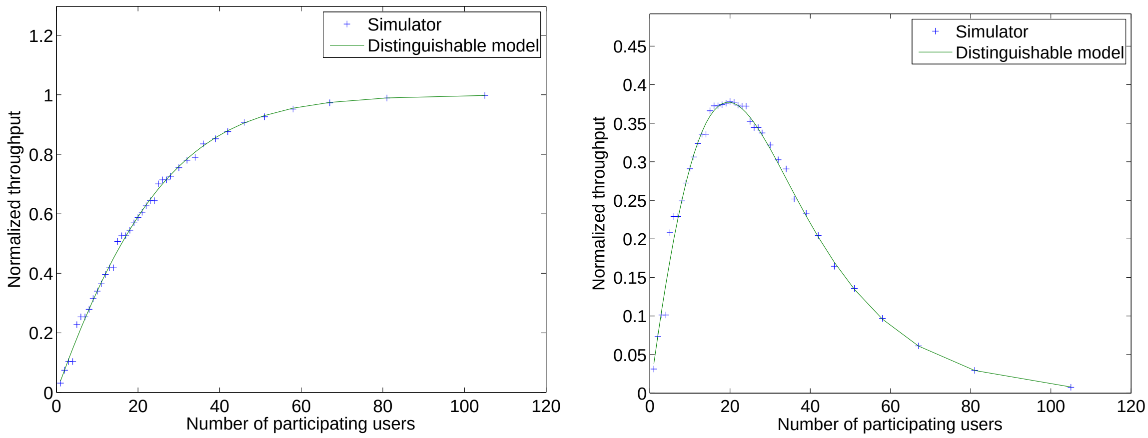
The figure shows the modelled throughput overlaid on the simulator's. The fit is good enough to indicate that this is a viable method of extrapolating the model's results. Such scaling then allows our model to be used for more realistic user counts (ranging in the thousands rather than the hundreds for a single LTE cell).

Using this technique, one can use the parameters listed above ($N_{max} = 103$, $p_N = 0.61$, $p_R = 1.00$) to model the behaviour of a much larger network such as $N_{max} = 10300$, $p_N = 0.0061$, $p_R = 0.01$.

6.3 Metrics

Throughput is measured in successful messages per RACH opportunity. When normalized, this is done against the theoretical maximum throughput $min(K, L)$.

Delay is measured in number of RACH cycles passed between message arrival and



(a) With collision resolution *(b) Without collision resolution*

Figure 6.2: Average throughput, model compared to simulator

successful transmission. An instantly successful transmission has a delay of 1 cycle.

6.4 Viability of the Markov Model

As discussed in section 5.2, the original model proposed in [1] does not agree with our observed results. Most of the result figures presented here have corresponding figures in that chapter.

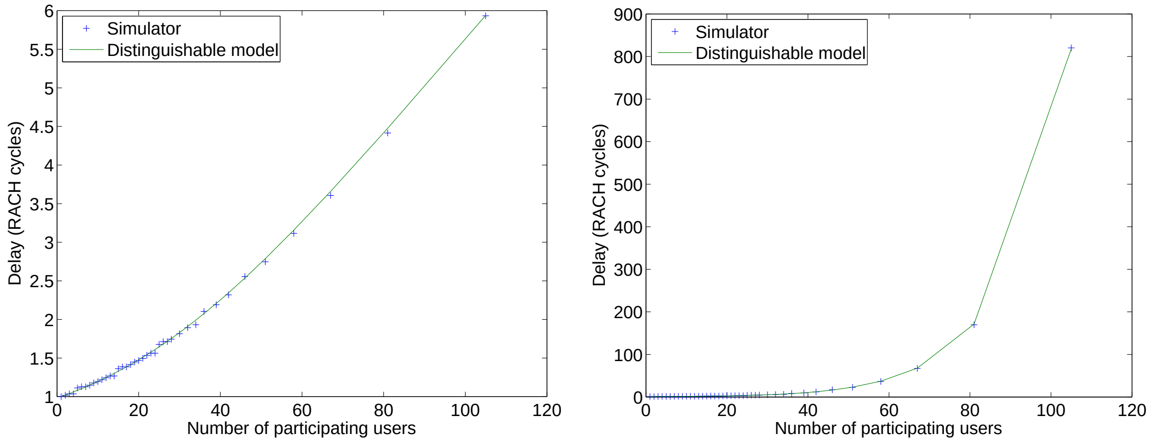
6.4.1 Throughput

Figure 6.2 shows the most basic metric of our model, the normalized throughput for a varying number of participating nodes. We see a very good correlation between our modified Markov model and the observed simulation results, both with and without collision resolution.

Another observation from figure 6.2b is that the peak normalized throughput with no collision resolution is roughly 37%, which corresponds to that of S-ALOHA[15]. With the normalization used, this means that a RACH system with 16 preambles (=16 separate S-ALOHA channels) has a peak throughput 16 times that of a single slotted ALOHA channel. The peak throughput thus appears to scale linearly with the number of preambles offered.

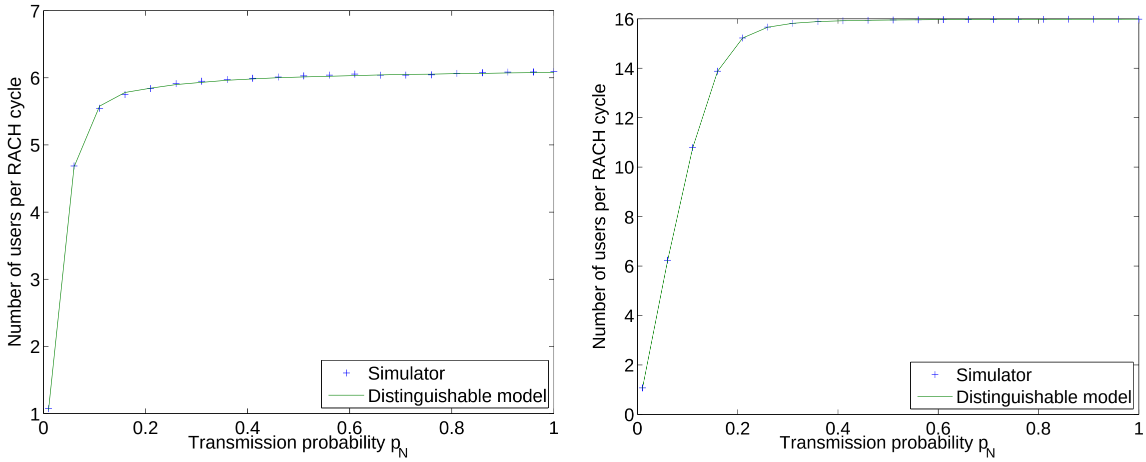
6.4.2 Delay

Figures 6.3a and 6.3b show the modelled transmission delay with and without collision resolution, respectively. Much like the throughput, our modified model closely matches



(a) With collision resolution *(b) Without collision resolution*

Figure 6.3: Average delay, model compared to simulator



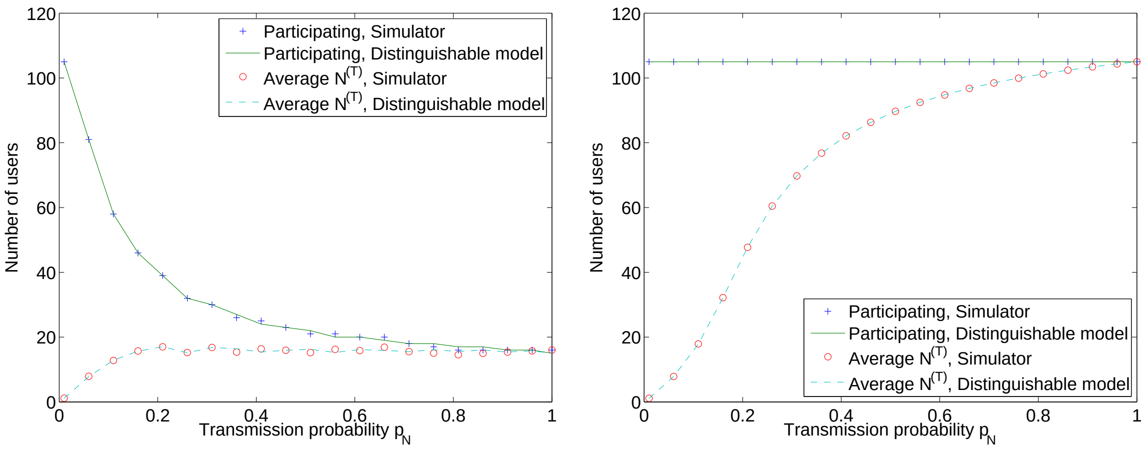
(a) Without collision resolution *(b) With collision resolution*

Figure 6.4: Peak throughput

the observed by simulator results. This also proves that delay derivation, appendix A, is viable, given a correctly modelled throughput.

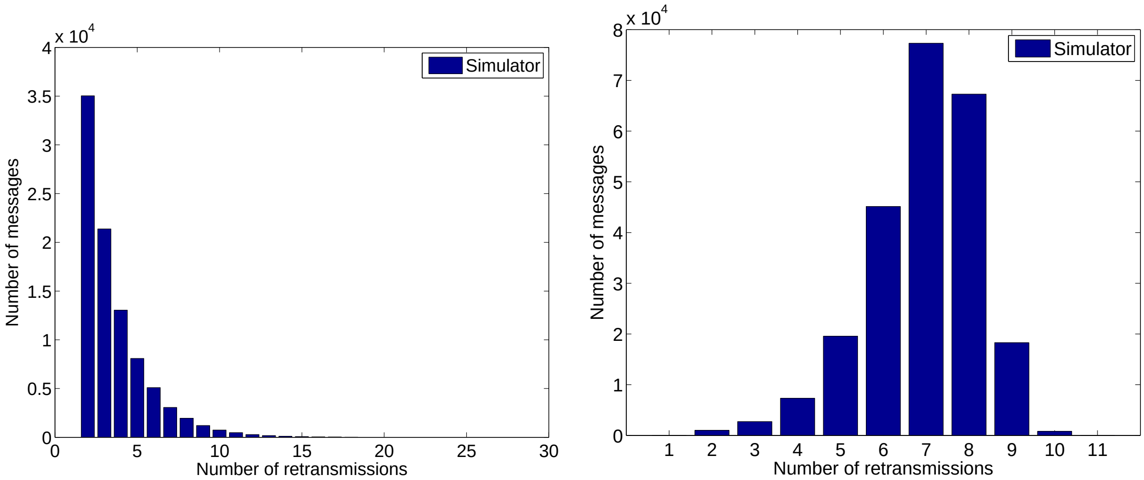
6.4.3 Peak Throughput

In order to gauge the maximum supported number of users in a network, we locate the throughput peak. This peak is found for varying p_N, identifying the best number of participating users. The throughput at this peak is illustrated by figures 6.4 and 6.4 with CR disabled and enabled, respectively. This correlation between transmission probability



(a) Without collision resolution *(b) With collision resolution*

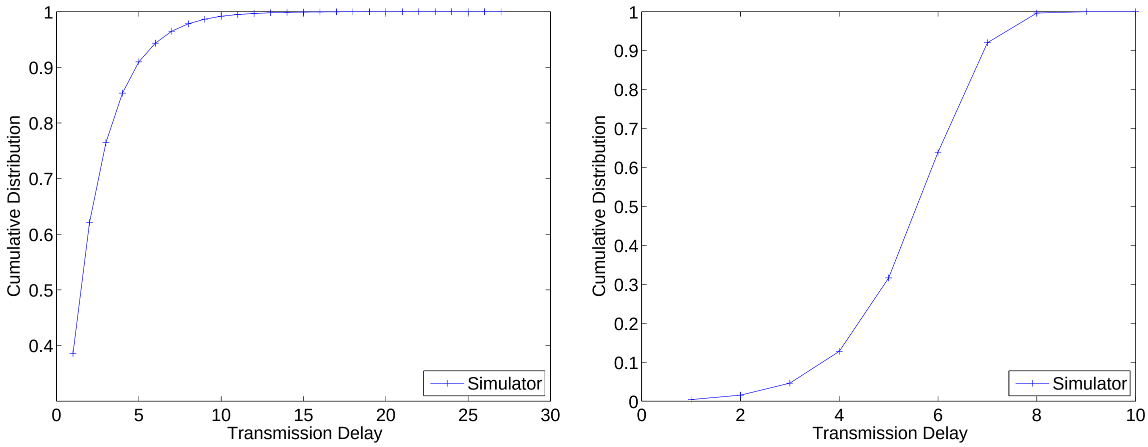
Figure 6.5: N and $N^{(T)}$ at peak throughput



(a) Without collision resolution *(b) With collision resolution*

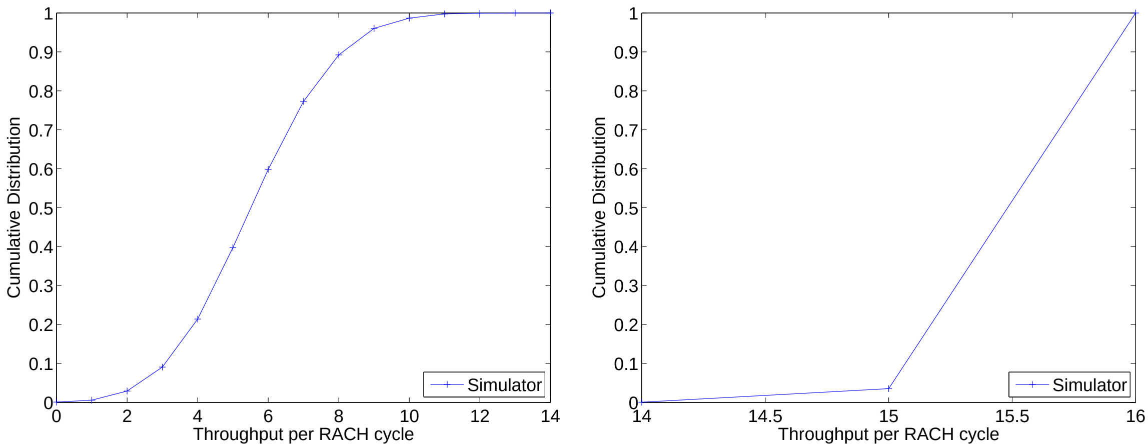
Figure 6.6: Retransmission counts over 10 minutes

and number of supported (participating) users is shown in figures 6.5a and 6.5b with and without collision resolution, respectively. We observe a stabilization of $N^{(T)}$, where a certain optimal number of transmitting users yields the maximum throughput. It may be advantageous to tweak p_R in order to closer approximate this number of transmitting users. That way, as more users end up in B-mode as backlogged, their retransmission probability would decrease and potentially allow a maximum throughput and thus the best opportunity to move nodes back to I-mode.



(a) *Without collision resolution* (b) *With collision resolution*

Figure 6.7: CDF of the transmission delay



(a) *Without collision resolution* (b) *With collision resolution*

Figure 6.8: CDF of the instantaneous throughput

6.5 Simulator Data

The simulator itself can be useful in providing certain data which the mathematical model cannot, namely retransmission counts and observed per-packet transmission delays. For consistency these values are illustrated for the simulation run which resulted in the peak throughput, i.e. $N = 105$ with CR and $N = 20$ without.

Figure 6.6 shows the number of retransmissions performed before successful transmission in the described scenario. This illustrates, out of all successful transmissions during the 10 minute interval simulated, how common each specific retransmission count was.

We see a larger spread of values when resolution is disabled, indicating a larger amount of jitter.

The Cumulative Distribution Function (CDF) is shown for transmission delay in figure 6.7. The early growth of this function indicates that the lower values of delay are the most common in our test scenario. Again the spread is larger when resolution is disabled, but the two grow at similar rates. The corresponding function for retransmission count is identical since our $p_R = 1.0$ dictates that every "delayed" RACH cycle is also one when a transmission attempt takes place.

Finally, the CDF of throughput (figure 6.8) illustrates the distribution of the number of successful transmissions in each RACH cycle of the simulation run. As expected, when CR is enabled the throughput is almost fully maximized.

6.6 Accuracy of Results

The model we produced for this work has proven quite accurate. It is important to note, however, that this is a probabilistic model compared to a simulator whose internal operation uses an almost perfectly uniform random distribution. As such, the close-fitting curves demonstrate only that the mathematical model is very good at predicting how our *idealized* network would behave, i.e. that there are no differing assumptions in the model design when compared to the simulator. This, however, only has bearing on the operation of a real-world WSN to the extent that our assumptions do.

6.7 Conclusion

Without collision resolution, the correlation between bandwidth and user count follows a peaking curve; at some global maximum, adding more transmitting nodes only serves to cause further collisions. We note that the peak average throughput is on par with S-ALOHA.

Our ideal model of collision resolution yields a strictly increasing throughput, as expected. Since a collision on a preamble yields exactly one acquired message regardless of how many nodes were involved in the collision, adding more transmitting nodes cannot *reduce* throughput. The normalized average throughput therefore asymptotically approaches 1.0

If multiple preambles are viewed as analogous to multiple S-ALOHA channels, we observe an increase in available bandwidth which is linear to the number of preambles.

The model can be used to determine an optimal number of transmitting users $N^{(T)}$ which yields the highest throughput in a system, and thus also the optimal number of participating users. This way, a WSN deployment can be optimized for the available network resources.

Chapter 7

Future Work

7.1 Mathematics

Chapter 3 discusses the *equilibrium distribution* π of our Markov chain. In our work, we rely on calculating the full single-transition probability matrix P and then solving the system of equations in equation 3.9 to find π. Determining the concrete matrix P is a very calculation-intensive task, and the number of elements grows with N^2. We do not, however need the actual values of P to estimate throughput. A method of calculating π given our derived properties of p_{ij} might be possible, and would simplify the practical application of our model. One example of this analytical use of π can be found in [14], where it allows an elegant derivation of the system throughput.

7.2 Further Study

7.2.1 Comparison

This thesis only deals with RACH as a means of scheduling. A comparison with the standard LTE method (RACH for first random access, PUCCH for subsequent requests) is necessary to compare our solution to the only currently possible case.

7.2.2 Collision Resolution

Our model is modified to include collision resolution for the preamble selection stage. The method used is simplified to where exactly one message can be recovered from any collision. In a realistic system, the probability of recovery is less than 1 and depends on the received power of the signal. This in turn depends on the environment, the distance between eNodeB and node and most importantly on the transmission power ramping which is performed at the node on each failed transmission attempt.

7.2.3 Alternate Methods

The problem we describe (a WSN starving regular users in PUCCH) could be solved in other manners than that investigated. For example, RACH scheduling scheduling resources could be reserved for WSN use as suggested in [3]. It would also be possible to assign PUCCH resources for the entire WSN rather than on a per-node basis and then let the WSN itself manage those (fewer) PUCCH resources by contention, cooperative scheduling or some other means.

7.2.4 Adaptive Load Mitigation

As described in section 6.4.3, our model can be used with a given number of users, preambles, servers and transmission probability to find an optimal number of transmitting users. If p_N is known, this optimal $N^{(T)}$ in turn can be used to determine the most efficient retransmission probability p_R. A system could even potentially determine this dynamically to reduce the impact of high-contention traffic spikes. Such a system may have other advantages such as wider throughput peaks.

7.2.5 Energy Efficiency

This thesis does not discuss energy efficiency, though it is one of the strictest requirements and limitations of a WSN. An investigation of the energy efficiency of a WSN using LTE when compared to, for example, a ZigBee mesh network is necessary to determine the real feasibility of LTE as a WSN transport.

CHAPTER 8

Conclusion

In this thesis, we presented a scenario model for a simplified sampling WSN around an event source. Nodes are placed in concentric *zones* around the source and said to transmit periodically. It is then shown that this scenario can be simplified to a single set of nodes with uniform behaviour.

LTE was studied for possible problem areas with regards to WSN integration. We identified a part of the design of PUCCH (uplink control) which could cause a WSN to interfere with the perceived quality of service for regular users. Large numbers of WSN nodes registered in PUCCH would increase the time between opportunities for regular users to request scheduling, thereby increasing delay in the network.

A solution is suggested, based on allowing WSN nodes to bypass PUCCH, instead using the Random Access procedure (RACH) to request scheduling when needed. To see the effects of this, a mathematical model of RACH is presented, based on an existing Slotted ALOHA model. We added support for relevant parts of the contention based RACH and verified our model by implementing a discrete-time event simulator.

A significant flaw was discovered in the original model and corrected by modifying the assumptions made. We verified the new model using the same RACH simulator.

We see a normalized throughput of roughly 37% for RACH, which indicates that it scales much like a set of independent S-ALOHA channels.

Finally, we demonstrated that the mathematical model is scalable to predict the behavior of thousands of sensor nodes in the same LTE cell, in line with more realistic demand scenarios.

Appendix A

Average Transmission Delay

An average transmission delay can be derived according to the interactive response time law[16]. Parameters used in the derivation are presented in table A.1.

The cycle time, which is time between two following one after another transmissions of a newly generated message, can be expressed as

$$(I + D),$$

then the number of messages sent per one time slot will be

$$\frac{T}{I + D}$$

and, for N number of users, system throughput can be expressed as the total number of messages sent per total time T

$$S = \frac{N \cdot \frac{T}{I+D}}{T} = \frac{N}{I + D}.$$

Therefore, the average transmission delay is

$$D = \frac{N}{S} - I,$$

I	The time node spends in the idle state, measured in RACH opportunities.
D	The time it takes for a newly generated message to get successfully scheduled by eNodeB.
T	The total time when measurements were taken.
N	The total number of participating nodes in the system.
S	The throughput of the system.
p_N	The rate with which a new message are generated.

Table A.1: Parameters used to derive an average transmission delay.

where idle time I is a waiting time between the old transmission have successfully performed and a new transmission occurs

$$I = \frac{1}{p_N}.$$

The final formula of the average transmission delay will be

$$D = \frac{N}{S} - \frac{1}{p_N}. \tag{A.1}$$

Appendix B

One-Step Transition Probability

One-step transition probability, p_{ij} is a probability for a system, defined in Chapter 3, to transit from state i to j in Markov chain.

Due to Markov chain and probability properties, the following condition can be derived:

$$\sum_{n=0}^{N} \sum_{s=0}^{Min(K,n)} Pr(N_k^{(S)} = s, N_k^{(T)} = n | N_k^{(B)} = i) = 1,$$

which includes all possible cases for a given number of backlogged nodes. Given this property and definition of p_{ij}, equation 3.6, we have the following:

$$p_{ij} = \sum_{n=0}^{N} \sum_{s=0}^{Min(K,n)} Pr(N_{k+1}^{(B)} = j, N_k^{(S)} = s, N_k^{(T)} = n | N_k^{(B)} = i). \tag{B.1}$$

According to equation 3.3 and given $N_k^{(B)} = i$ and $N_k^{(S)} = s$, we can derive the following expression:

$$\begin{aligned} N_{k+1}^{(B)} = j \Leftrightarrow N_k^{(N)} &= N_{k+1}^{(B)} - N_k^{(B)} + N_k^{(S)} \\ &= j - i + s. \end{aligned}$$

By applying it on equation B.1, we have:

$$\begin{aligned} p_{ij} &= \sum_{n=0}^{N} \sum_{s=0}^{Min(K,n)} Pr(N_k^{(N)} = j - i + s, N_k^{(S)} = s, N_k^{(T)} = n | N_k^{(B)} = i) \\ &= \sum_{n=0}^{N} \sum_{s=0}^{Min(K,n)} Pr(N_k^{(S)} = s | N_k^{(T)} = n, N_k^{(N)} = j - i + s, N_k^{(B)} = i) \\ &\quad \times Pr(N_k^{(T)} = n | N_k^{(N)} = j - i + s, N_k^{(B)} = i) \\ &\quad \times Pr(N_k^{(N)} = j - i + s | N_k^{(B)} = i) \end{aligned} \tag{B.2}$$

According to equation 3.2 and given $N_k^{(N)} = j - i + s$:

$$\begin{aligned} N_k^{(T)} = n \Leftrightarrow N_k^{(R)} &= N_k^{(T)} - N_k^{(N)} \\ &= n - j + i - s. \end{aligned} \tag{B.3}$$

Similarly, according to equations 3.1 and given $N_k^{(B)} = i$:

$$\begin{aligned} N_k^{(B)} = i \Leftrightarrow N_k^{(I)} &= N - N_k^{(B)} \\ &= N - i \end{aligned} \tag{B.4}$$

Further, the equation B.2 can be broke into three parts. Each part, in turn, can be modified according to equations B.3 and B.4.

The first part can be represented as

$$Pr(N_k^{(S)} = s | N_k^{(T)} = n, N_k^{(N)} = j - i + s, N_k^{(B)} = i) = Pr(N_k^{(S)} = s | N_k^{(T)} = n).$$

Due to equation B.3, the second part can be represented as

$$Pr(N_k^{(T)} = n | N_k^{(N)} = j - i + s, N_k^{(B)} = i) = Pr(N_k^{(R)} = n - j + i - s | N_k^{(B)} = i).$$

Due to equation B.4, the last part is represented as

$$Pr(N_k^{(N)} = j - i + s | N_k^{(B)} = i) = Pr(N_k^{(N)} = j - i + s | N_k^{(I)} = N - i).$$

Recombining these parts we have:

$$\begin{aligned} p_{ij} = \sum_{n=0}^{N} \sum_{s=0}^{Min(K,n)} & Pr(N_k^{(R)} = n - j + i - s | N_k^{(B)} = i) \\ & \times Pr(N_k^{(N)} = j - i + s | N_k^{(I)} = N - i) \\ & \times Pr(N_k^{(S)} = s | N_k^{(T)} = n). \end{aligned} \tag{B.5}$$

$Pr(N_k^{(R)} = n - j + i - s | N_k^{(B)} = i)$ is the probability that $N_k^{(R)}$ nodes retransmits given $N_k^{(B)}$ backlogged nodes, which can be expressed as

$$Pr(N_k^{(R)} = n - j + i - s | N_k^{(B)} = i) = \binom{i}{n - j + i - s} \cdot p_R^{n-j+i-s} \cdot (1 - p_R)^{j-n+s}.$$

$Pr(N_k^{(N)} = j - i + s | N_k^{(I)} = N - i)$ is the probability that $N_k^{(N)}$ nodes transmits given $N_k^{(I)}$ nodes in I-mode, this can be expressed as

$$Pr(N_k^{(N)} = j - i + s | N_k^{(I)} = N - i) = \binom{N - i}{j - i + s} \cdot p_N^{j-i-s} \cdot (1 - p_N)^{N-j-s}.$$

Finally, $Pr(N_k^{(S)} = s | N_k^{(T)} = n)$ is the probability that $N_k^{(S)}$ nodes succeed given $N_k^{(T)}$ transmitting nodes, which is per definition

$$Pr(N_k^{(S)} = s | N_k^{(T)} = n) \stackrel{\text{def}}{=} f_{N^{(S)} | N^{(T)} = n}(s)$$

APPENDIX C

Preamble Distribution

As outlined in Chapters 3 and 5, a method is needed to describe the way a set of transmitting nodes select their respective preambles and how any collisions are handled by the system.

The problem of describing nodes selecting preambles for their messages is equivalent to the more common notation of distributing balls in urns, and the combinatorial methods used in this appendix are based on that observation.

Further, since no node can have more than one pending message at any given time, the two terms are used interchangeably. A node being *acquired* is equivalent to its message being acquired.

C.1 Indistinguishable Messages

The mathematical model of RACH access is based on a notion that all participating nodes are treated similarly, i.e. the difference between two nodes cannot be seen in the model. For this reason, the problem of nodes choosing a preamble can be mapped to a problem of distributing n *indistinguishable messages* over K *distinguishable preambles*.

Further, the model mathematical model can either use collision resolution or let it be disabled, there is a need to consider two slightly different cases.

C.1.1 With collision resolution

While collision resolution is enabled, the number of acquired nodes is equal to the number of preambles chosen by at least one.

$f_{N^{(A)}|N^{(T)}=n}(a)$ represents a probability that all n messages are distributed over a preambles as described above, i.e. the ratio between permutations with $N^{(A)} = a$ and all possible permutations.

To find the number of ways we can distribute all n messages over a preambles and leave $K - a$ preambles unused we use the following method:

1. First we need to choose a from K preambles. This can be done in $\binom{K}{a}$ ways.
2. Once this is done, we assign one message each to these a preambles, to guarantee that they are used by at least one. That leaves $n-a$ unassigned messages.
3. We then need to distribute these remaining $n-a$ messages over the same a preambles, which already had one message each. This can be done in

$$\binom{a+n-a-1}{n-a} = \binom{n-1}{n-a}$$

different ways.

4. Multiplying the two factors we get total number of permutations we can distribute all messages so that exactly a preambles of the total K are occupied by n messages:

$$\binom{K}{a} \cdot \binom{n-1}{n-a} \tag{C.1}$$

The total number of ways we can distribute n indistinguishable messages over K distinguishable preambles is

$$\binom{n+K-1}{n} \tag{C.2}$$

Further, $f_{N^{(A)}|N^{(T)}=n}(a)$ can be represented as the ratio between equations C.1 and C.2:

$$f_{N^{(A)}|N^{(T)}=n}(a) = \frac{\binom{K}{a}\binom{n-1}{n-a}}{\binom{n+K-1}{n}}$$

However, this equation is valid only when the value of a does not exceed the total number of neither transmitted messages nor preambles. In such a case there would be no possible way to distribute the messages according to the criteria.

Thus, finally, the probability of n messages being distributed over K preambles so that exactly a preambles are in use and all others $K-a$ are unused can be described as:

$$f_{N^{(A)}|N^{(T)}=n}(a) = \begin{cases} \frac{\binom{K}{a}\binom{n-1}{n-a}}{\binom{n+K-1}{n}} & \text{if } 0 \leq a \leq min(n,K), \\ 0 & \text{otherwise.} \end{cases} \tag{C.3}$$

C.1.2 Without collision resolution

When collision resolution is disabled, the situation is a little different as we are interested in preambles which are used by exactly one message each and all other preambles fall out of interest.

Though the distribution $f_{N^{(A)}|N^{(T)}=n}(s)$ is different from the one with collision resolution, the method to find it is still the same. We need to find a ratio between the number of permutations satisfying distribution described above and all possible permutations.

1. We start exactly as the in case with collision resolution by choosing a from K preambles, which is possible to do in $\binom{K}{a}$ ways.

2. Once a preambles are chosen we can now assign a preambles to these so that each of the a preambles is used by exactly one message. The messages are indistinguishable and therefore it can be done just in one way.

3. We now have $n-a$ messages left to distribute over $K-a$ preambles so that each of these preambles contain either zero or at least two. Let us define a variable i, where $n-a$ remaining messages are distributed over i out of $K-a$ remaining preambles with each of the i preambles used by at least two messages. The remaining $K-a-i$ preambles will be left unused. This distribution can be described as

   - First, the following constraints for i can be observed:
   $$\left.\begin{array}{l} 0 \leq i \leq K-a \\ 0 \leq 2i \leq n-a \end{array}\right\} \Leftrightarrow 0 \leq i \leq Min(\lfloor\frac{n-a}{2}\rfloor, K-a). \tag{C.4}$$
   - All possible permutations to choose i out of $K-a$ preambles are
   $$\binom{K-a}{i} \tag{C.5}$$
   Then to assign two messages to each preamble, as all i preambles should be in use by at least two messages, can be done in only one way as messages are indistinguishable.
   - Further, to distribute $n-a-2i$ remaining messages over i preambles is possible in following number of ways:
   $$\binom{(n-a-2i)+i-1}{n-a-2i} = \binom{n-a-i-1}{n-a-2i} \tag{C.6}$$

   Combining equations C.4, C.5 and C.6, we get the following:
   $$\begin{cases} \binom{K-a}{i}\binom{n-a-i-1}{n-a-2i} & \text{if } 0 \leq i \leq Min(\lfloor\frac{n-2}{2}\rfloor, K-a), \\ 0 & \text{otherwise.} \end{cases} \tag{C.7}$$

4. By applying the combinational rules of sum and product onto C.7 and then combining it with the first point, we have the following:
$$\begin{cases} \binom{K}{a} \displaystyle\sum_{i=0}^{Min(\lfloor\frac{n-a}{2}\rfloor, K-a)} \binom{K-a}{i}\binom{n-a-i-1}{n-a-2i} & \text{if } 0 \leq a \leq Min(n, K), \\ 0 & \text{otherwise.} \end{cases} \tag{C.8}$$

Equation C.8 describes the number of permutations satisfying conditions of our problem with collision resolution disabled.

The total number of all possible permutations for indistinguishable messages and distinguishable preambles remains the same as with case of collision resolution is enabled, equation C.2.

Finally, the ratio between *allowed* permutations, equation C.8, and all possible permutations, equation C.2, gives us the probability of n messages being distributed over K preambles so that exactly a preambles are uniquely chosen:

$$f_{N^{(A)}|N^{(T)}=n}(a) = \begin{cases} \dfrac{\binom{K}{a} \sum_{i=0}^{Min(\lfloor \frac{n-a}{2} \rfloor, K-a)} \binom{K-a}{i}\binom{n-a-i-1}{n-a-2i}}{\binom{n+K-1}{n}} & \text{if } 0 \le a \le Min(n,K), \\ 0 & \text{otherwise.} \end{cases} \tag{C.9}$$

C.2 Distinguishable Messages

As outlined in Section 5.2, a method is required to determine the number of acquired nodes based on the assumption of *distinguishable preambles* and *distinguishable messages*. This is done by redefining the distribution $f_{N^{(A)}|N^{(T)}}(a)$. The problem can further be divided into a case with collision resolution and a case without.

In both cases, we approach the probability description by counting possible permutations of distributing the n messages across K preambles.

In common for both methods is the total number K^n ways to distribute n distinguishable messages over K distinguishable preambles.

It is also notable that this problem can be reduced to the well-known *birthday problem*, though that is not discussed here.

C.2.1 With collision resolution

When resolution is enabled, the total number of acquired nodes is equal to the number of preambles used by at least one message. We start by noting that there are $\binom{K}{a}$ different sets of a preambles in which this could occur. We are then interested in the total number of possible permutations of n messages over these a preambles such that all a are in use. We describe this number by the recursive function $V(a, n)$,

$$\begin{aligned} V(a,n) = \binom{n}{1} \cdot V(a-1, n-1) + \binom{n}{2} \cdot V(a-1, n-2) + \ldots \\ + \ldots + \binom{n}{n-(a-1)} V(a-1, a-1) \end{aligned} \tag{C.10}$$

where each invocation deals with a single preamble (the first among a) and each term describes one possible number of messages selecting this preamble. For example, in term

2

$$\binom{n}{2} \cdot V(a-1, n-2)$$

there are $\binom{n}{2}$ ways for 2 out of n messages to select this particular preamble. For each of those possibilities, there are then $(a-1)$ preambles and $(n-2)$ messages left to distribute and that can be done in $V(a-1, n-2)$ ways. The series ends naturally at $n-(a-1)$ since there need to be at least $(a-1)$ messages left in order to have at least one for each of the remaining $(a-1)$ preambles.

Equation C.10 can be simplified as

$$V(a, n) = \sum_{i=1}^{n-(a-1)} \binom{n}{i} \cdot V(a-1, n-i) \tag{C.11}$$

Further examination of this function and its purpose reveal a few base cases, namely

$$V(0, 0) = 1 \tag{C.12}$$

$$V(0, n > 0) = 0 \tag{C.13}$$

$$V(a, n < a) = 0. \tag{C.14}$$

Equation C.12 declares exactly one way to acquire zero messages using zero preambles. Equations C.13 and C.14 reflect that there can be no messages left after we run out of preambles and that we cannot acquire more messages than those transmitted, respectively.

Combining equation C.11 with the number of ways a preambles can be selected yields the total number of ways to distribute n messages over K preambles such that exactly a are in use:

$$\binom{K}{a} \cdot V(a, n).$$

Further, the ratio between this and the total number of possible ways to distribute the messages yields our probability distribution:

$$f_{N^{(A)}|N^{(T)}}(a) = \frac{\binom{K}{a} \cdot V(a, n)}{K^n} \tag{C.15}$$

C.2.2 Without collision resolution

In the case of no collision resolution, the number of acquired nodes is equal to the number of preambles which were chosen by exactly one node.

We seek the total number of ways to distribute n messages over K preambles such that exactly a preambles are used by a single message the others by either zero or at least two. We denote this number W(K, a, n).

This number of combinations can be split up in a function much like the resolution-enabled case above, where each invocation handles the first of the available preambles and defers the rest of them via recursion.

The first possibility for a single preambles is that it takes no messages. In that case, there will be one less preamble but just as many messages n required to leave exactly a messages acquired.

$$W(K-1, a, n) \tag{C.16}$$

The second possibility is that exactly one message is chosen. There are $\binom{n}{1} = n$ ways to do so. That leaves $(n-1)$ messages left to distribute over $(K-1)$ preambles acquiring $(a-1)$.

$$n \cdot W(K-1, a-1, n-1) \tag{C.17}$$

The third and final possibility is that 2 or more messages select the preamble. This is possible for any number of messages up to $(n-a)$ since there need to be at least a messages left to be acquired.

$$\binom{n}{2} \cdot W(K-1, a, n-2) + \binom{n}{3} \cdot W(K-1, a, n-3) + \ldots$$
$$\ldots + \binom{n}{(n-a)} \cdot W(K-1, a, a) = \sum_{i=2}^{n-a} \binom{n}{i} \cdot W(K-1, a, n-i) \tag{C.18}$$

Putting equations C.16, C.17, C.18 together yields a sum of all possible combinations and thus a full definition of W:

$$\begin{aligned} W(K, a, n) = W(K-1, a, n)+ \\ + n \cdot W(K-1, a-1, n-1)+ \\ + \sum_{i=2}^{n-a} \binom{n}{i} \cdot W(K-1, a, n-i) \end{aligned} \tag{C.19}$$

The base cases of this function are found as follows:

$$W(K, 0, 0) = 1 \tag{C.20}$$

$$W(K, a > K, n) = 0 \tag{C.21}$$

$$W(K, a, n < a) = 0 \tag{C.22}$$

Equation C.20 states that if all messages have been assigned and the required number have been acquired, the remaining preambles can be left unused. Equation C.21 marks the impossibility of acquiring more messages than the remaining number of preambles. Finally, equation C.22 states that we cannot acquire more messages than those transmitted.

The ratio between equation C.19 and the total number of possible ways to distribute n messages over K preambles yields our probability distribution

$$f_{N^{(A)}|N^{(T)}}(a) = \frac{W(K,a,n)}{K^n} \tag{C.23}$$

Appendix D

Server Acceptance

A *server* in our model represents the ability of eNodeB to handle a single incoming request during the RACH period it arrived. Any extra requests are rejected and the corresponding node must try again at a later RACH opportunity. We assume an ideal implementation where each server will accept one acquired message if at all available.

To fit into the rest of the mathematical framework, we need to formulate a probability distribution $f_{N^{(S)}|N^{(A)}=a,N^{(T)}=n}(s)$. This is the probability that, given exactly n transmitted and a acquired requests, exactly s are successful.

First, we note that with a set a and regardless of n, there is only one possible value for s. This s has a probability of 1, while all other s values have probabilities of 0.

For $a \leq L$:

$$f_{N^{(S)}|N^{(A)}=a,N^{(T)}=n}(s) = \begin{cases} 1, & \text{if } s = a \\ 0, & \text{otherwise .} \end{cases} \tag{D.1}$$

For $a > L$:

$$f_{N^{(S)}|N^{(A)}=a,N^{(T)}=n}(s) = \begin{cases} 1, & \text{if } s = L \\ 0, & \text{otherwise .} \end{cases} \tag{D.2}$$

Equations D.1 and D.2 can further be combined and simplified as

$$f_{N^{(S)}|N^{(A)}=a,N^{(T)}=n}(s) = \begin{cases} 1, & \text{if } s = min(a, L) \\ 0, & \text{otherwise .} \end{cases} \tag{D.3}$$

Bibliography

[1] Z. Liu and M. El Zarki, "Performance analysis of DS-CDMA with slotted aloha random access for packet PCNs," *Wirel. Netw.*, vol. 1, pp. 1–16, February 1995. [Online]. Available: http://dx.doi.org/10.1007/BF01196254

[2] J. Zhang, L. Shan, H. Hu, and Y. Yang, "Mobile cellular networks and wireless sensor networks: Toward convergence," *Communications Magazine, IEEE*, vol. 50, no. 3, pp. 164 –169, march 2012.

[3] 3rd Generation Partnership Project Technical Specification Group Radio Access Network, "Study on RAN Improvements for Machine-type Communications TR-37.868," september 2011.

[4] K.-D. Lee, S. Kim, and B. Yi, "Throughput comparison of random access methods for m2m service over lte networks," in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, dec. 2011, pp. 373 –377.

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

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

[7] M. Corici, J. Fiedler, T. Magedanz, and D. Vingarzan, "Evolution of the resource reservation mechanisms for machine type communication over mobile broadband evolved packet core architecture," in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, dec. 2011, pp. 718 –722.

[8] A. Damnjanovic, J. Montojo, J. Cho, H. Ji, J. Yang, and P. Zong, "Ue's role in lte advanced heterogeneous networks," *Communications Magazine, IEEE*, vol. 50, no. 2, pp. 164 –176, february 2012.

[9] 3rd Generation Partnership Project, "TS 36.201: E-UTRA LTE physical layer; General description."

[10] 3rd Generation Partnership Project, “TS 36.212: E-UTRA Multiplexing and channel coding.”

[11] 3rd Generation Partnership Project, “TS 36.211: E-UTRA LTE physical channels and modulation.”

[12] E. Dahlman, S. Parkvall, and J. Sköld, *4G LTE / LTE-Advanced for Mobile Broadband*, 2011.

[13] 3rd Generation Partnership Project, “TS 36.321: E-UTRA Medium Access Control (MAC) protocol specification.”

[14] G. Bianchi, “Performance analysis of the ieee 802.11 distributed coordination function,” *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535 –547, mar 2000.

[15] L. G. Roberts, “Aloha packet system with and without slots and capture,” *SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28–42, Apr. 1975. [Online]. Available: http://doi.acm.org/10.1145/1024916.1024920

[16] R. Jain, *The Art of Computer Systems Performance Analysis*, april 1991.