Jitter Management in Circuit Switched Voice over HSPA Networks

GE SONG

KTH Information and
Communication Technology

Master of Science Thesis
Stockholm, Sweden 2009

TRITA-ICT-EX-2009:1

Jitter Management in Circuit Switched Voice over HSPA Networks

Ge Song

Master Thesis Report

2009.04.03

Supervisor
Tomas Frankkila
Ericsson Research, Luleå

Examiner
Gerald Q. Maguire Jr.
Royal Institute of Technology (KTH)

Abstract

When a shared channel or packet switched network is used for transmission (e.g. WLAN, HSPA (Turbo-3G), LTE (4G)), it introduces variance in the delay of packets. This variance is called jitter. This jitter can lead to significant degradation of quality in real-time services if it is not properly handled.

High Speed Packet Access (HSPA) is an extension to the third Generation W-CDMA cellular network that provides significantly increased bandwidth and network capacity by introducing a High Speed-Downlink Shared Channel (HS-DSCH) for downlink and an Enhanced-Dedicated Channel (E-DCH) for uplink. Both HS-DSCH and E-DCH use re-transmissions in order to ensure a low block error rate, as a result jitter is induced in both channels. Moreover, HS-DSCH also uses channel dependent scheduling between users adding additional jitter.

Since HSPA uses IP and the voice service is provided by voice over IP (VoIP), jitter management is performed at the destination end-point. However, 3GPP has also specified transportation of circuit switched voice over HSPA (CSoHS), where jitter management needs to be performed separately, at both the entry point to the core network and in the receiving end-point as jitter is introduced both in the uplink and downlink.

This report studies CSoHS, with a focus on its delay and jitter characteristics. It introduces two schemes for jitter management: a fixed jitter buffer and an adaptive jitter buffer. These jitter buffer designs are evaluated mainly by looking at the jitter loss (i.e., the proportion of packets that have to be discarded because they exceed the maximum permitted jitter) and the buffering time. The results show that the adaptive jitter buffer can achieve better performance in balancing the trade-off between jitter loss and buffering delay when dealing with various network conditions. In contrast, the fix jitter buffer is not capable of tracking variations in the network conditions, as the performance of the fixed jitter buffer is determined solely by the configuration of the initial buffer level. The adaptive jitter buffer is able to consistently provide equal or better quality of service than the fixed jitter buffer.

Acknowledgement

I would like to acknowledge the contributions of the following groups and individuals to the development of my master thesis:

Stefan Håkansson and Tomas Frankkila from Ericsson Research for offering me such valuable opportunity to work on this thesis within Ericsson; and my sincere gratitude to Tomas Frankkila, my industrial supervisor for his expertise;

Ingemar Johansson and Jonas Lundberg for their guidance in helping me study the existing simulator and jitter buffer techniques.

Special thanks to my Chinese colleagues, Wang Min, Zuo yang and Xin Jun for their assistance in my work and life in Luleå.

Daniel Enström, my section manager for his support and administration during my work;

My KTH examiner, Gerald Q. "Chip" Maguire Jr. for his concern and examination of my thesis work;

My family and all my friends for their loving support along the way.

Table of Contents

Abstract i

Acknowledgement iii

List of Figures vi

List of Tables vii

List of Symbols and Abbreviations viii

1. Introduction 1

1.1 Circuit Switched vs. Packet Switched Voice Services 1

1.2 Problem Statement 1

1.2.1 Introduction to Jitter and Jitter Buffer 1

1.2.2 Distributed Jitter Buffers in CSoHS 2

1.3 Structure of the Thesis Project 3

1.4 Outline of this Thesis 4

2. Background 5

2.1 Overview of the Architecture 5

2.1.1 Nature of Speech 5

2.1.2 Network Impairments 7

2.2 Adaptive Multi-Rate Speech CODEC 8

2.2.1 Coding Modes 8

2.2.2 Silence Suppression 9

2.2.3 Error Concealment 9

2.3 Jitter Buffer Management 9

2.3.1 Types of Jitter Buffers 9

2.3.2 Jitter loss and Jitter Induced Concealment 11

2.3.3 Performance Requirements for JBM 11

2.4 HSPA 12

2.4.1 HSDPA 12

2.4.2 HSUPA 16

2.5 Circuit Switched over HSPA (CSoHS) 18

2.5.1 Delay Budget of CSoHS 18

2.5.2 JBM for CSoHS 21

3. Method 22
3.1 Simulation Environment 22
3.1.1 Simulator 22
3.1.2 Design of CSoHS JBMs 23
3.1.3 Delay & Error Profiles 24
3.2 Evaluation 27
3.2.1 Objective Evaluation 27
3.2.2 Subjective Evaluation 27

4. Analysis 29
4.1 Analysis of the Static JBM 29
4.1.1 UL Delay and Error Profiles from 3GPP 29
4.1.2 Synthetic UL Delay and Error Profiles 29
4.1.3 DL Delay and Error Profiles from 3GPP 32
4.2 Results of Adaptive JBM 33
4.2.1 UL Delay and Error Profiles from 3GPP 33
4.2.2 Synthetic UL Delay and Error Profiles 35
4.2.3 DL Delay and Error profiles from 3GPP 42
4.3 Comparison between Static and Adaptive JBMs 43
4.3.1 Discussion 44
4.4 End-to-End Aspects of CSoHS 45
4.4.1 Overall Jitter Loss 45
4.4.2 End-to-End Delay 45
4.5 Subjective Evaluation 46

5. Conclusions 47
5.1 Conclusion 47
5.2 Future work 49

References 50

A. HSPA Data Flow Illustration 52

B. Pseudo code of Designed JBM 54

C. Channel Delay and JBM Adaptation 58

List of Figures

Figure 1.1: Jitter and jitter buffer in end-to-end IP network.. 2
Figure 1.2: Distributed jitter management in CSoHS. 3
Figure 2.1: Pattern of speech signal... 6
Figure 2.2: Network impairment 1
Figure 2.3: Adaptation in silence period 1
Figure 2.4: Schedule users with favourable channel conditions [8].................................. 13
Figure 2.5: Multiple HARQ processes (6 assumed) [4] 13
Figure 2.6: Radio interface protocol architecture of HS-DSCH 1
Figure 2.7: Radio interface protocol architecture of E-DCH 1
Figure 3.1: Simulation chain 1
Figure 3.2: Channel delay of "HSUPA_PA3_45u"... 25
Figure 3.3: Channel delay of "HSUPA_PB3_45u"... 25
Figure 4.1: End-to-end delay of "HSUPA_PA3_45u" with adaptive JBM 33
Figure 4.2: End-to-end delay of "HSUPA_PB3_45u" with adaptive JBM........................ 34
Figure 4.3: Channel delay of "Low-load, drop timer=75ms"... 36
Figure 4.4: End-to-end delay for "Low-load, drop timer=75ms"...................................... 36
Figure 4.5: Channel delay of "Low-load, drop timer-200ms" 37
Figure 4.6: End-to-end delay for "Low-load, drop timer=200ms".................................... 37
Figure 4.7: Channel delay of "Over-load, drop timer=75ms" 38
Figure 4.8: End-to-end delay for "Over-load, drop timer=75ms" 38
Figure 4.9: Channel delay of "Over-load, drop timer=200ms" 39
Figure 4.10: End-to-end delay with initial buffer level=4.. 39
Figure 4.11: End-to-end delay with initial buffer delay=10 40
Figure 4.12: Channel delay of "Dynamic, drop timer=200ms" 40
Figure 4.13: End-to-end delay with initial buffer level =4... 40
Figure 4.14: End-to-end delay with initial buffer level=10 41
Figure 4.15: Jitter loss rates of adaptive and fixed JBMs... 1
Figure 4.16: End-to-end delay (ms) of adaptive and fixed JBMs 1
Figure 5.1: Delay comparison 1
Figure 5.2: Jitter loss comparison... 1
Figure C.1. Channel delay of "Medium-load, drop timer=75ms" 58
Figure C.2. End-to-end delay with adaptive JBM 58
Figure C.3. Channel delay of "Medium-load, drop timer=200ms" 59
Figure C.4. End-to-end delay with adaptive JBM 59
Figure C.5. Channel delay of "High-load, drop timer=100ms" 59
Figure C.6. End-to end delay with adaptive JBM 60
Figure C.7. Channel delay of "High-load, drop timer=200ms" 60
Figure C.8. End-to-end delay with adaptive JBM 60
Figure C.9. Channel delay of "HSDPA_PA3_100u_G1.65dB_55ms" 61
Figure C.10. End-to-end delay.. 61
Figure C.11. Channel delay of "HSDPA_PB3_45u_G0.09dB_55ms" 62
Figure C.12. End-to-end delay.. 62
Figure C.13. Channel delay of "HSDPA_PB3_45u_G0.09dB_155ms" 62
Figure C.14. End-to-end delay.. 63
Figure C.15. Channel delay of "HSDPA_PB3_100u_G0.09dB_95ms" 63
Figure C.16. End-to-end delay.. 64
Figure C.17. Channel delay of "HSDPA_PB3_100u_G0.09dB_155ms" 64
Figure C.18. End-to-end delay.. 64

List of Tables

Table 1.1:Structure of the Thesis Project 3
Table 1.2: Outline of the Thesis 4
Table 2.1: Delay budget of CSoHS 18
Table 3.1: Delay & error profiles from 3GPP 24
Table 3.2: Characteristics of 3GPP delay and error profiles 26
Table 3.3: Synthetic UL delay and error profiles 26
Table 4.1: Results of 3GPP E-DCH profiles with static JBM 29
Table 4.2: Test results of synthetic E-DCH profiles with static JBM (part 1) 30
Table 4.3: Results of synthetic E-DCH profiles with static JBM (part 2) 32
Table 4.4: Results of 3GPP DL delay and error profiles 32
Table 4.5: Results of 3GPP UL profiles with adaptive JBM 33
Table 4.6: Results of synthetic UL profiles with adaptive JBM (part 1) 35
Table 4.7: Results of synthetic UL profiles with adaptive JBM (part 2) 42
Table 4.8: Results of 3GPP DL profiles with adaptive JBM 43
Table 4.9: Jitter loss rates with adaptive JBM 45
Table 4.10: Overall delay of CSoHS with adaptive JBM 45
Table 4.11: Overall delay of CSoHS with fixed JBM 46
Table 5.1: Comparison between static and adaptive JBMs 47

List of Symbols and Abbreviations

AM	Acknowledge Mode
AMR	Adaptive Multi-Rate
BLER	Block Error Rate
CN	Core Network
CODEC	Coder/Decoder
CS	Circuit Switched
CSoHS	Circuit Switched over HSPA
DCH	Dedicated Channel
DL	Downlink
DRX	Discontinuous Reception
DTX	Discontinuous Transmission
E-DCH	Enhanced Dedicated Channel
EUL	Enhanced Uplink
HARQ	Hybrid Automatic Repeat reQuest
HS-DSCH	High Speed Downlink Shared Channel
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Access
HSUPA	High Speed Uplink Packet Access
IMS	IP Multimedia Subsystem
JBM	Jitter Buffer Management
LLR	Late Loss Rate
MAC	Medium Access Control
PDCP	Packet Data Convergence Protocol
PDU	Protocol Data Unit
RAN	Radio Access Network
RLC	Radio Link Control
RNC	Radio Node Controller
RTP	Real Time Protocol
RX	Receive
SDU	Service Data Unit
SID	Silence Indicator
SN	Sequence Number
TM	Transparent Mode
TNL	Transport Network Layer
TS	Time Stamp
TSN	Transmission Sequence Number
TTI	Transmission Time Interval
TX	Transmit
UE	User Equipment
UL	Uplink
UM	Unacknowledged Mode
UMD	Unacknowledged Mode Data
UTRAN	Universal Terrestrial Radio Access Network
VAD	Voice Activity Detection

1. Introduction

This chapter gives a more explicit description of the problem defined for this thesis project. It also describes what has been done during the project and outlines the structure of the report.

1.1 Circuit Switched vs. Packet Switched Voice Services

First of all, it is necessary to clarify two very basic concepts: circuit-switched and packet-switched voice service. In circuit switched (CS) voice services a dedicated point-to-point connection (i.e., a circuit) is established between nodes or terminals *before* the voice communication starts. The end-to-end delay will be constant during this connection. Each circuit is used exclusively by one user, until the circuit is released and a new connection is set up. Even if no actual communication is taking place via this dedicated circuit, its resource remains unavailable to others [1].

In contrast, a packet-switched voice service operates over a packet switched (PS) network. In such a network the encoded voice data is placed in packets, which are routed over a shared network. Each packet is labelled with its destination address. At each network node, packets may be queued or buffered while awaiting forwarding, resulting in variable delay and throughput that depends on the traffic load in the network [2].

1.2 Problem Statement

This section first gives a general introduction of jitter and the concept of a so-called jitter buffer, followed by a discussion of distributed jitter buffers in a circuit switched voice service implemented for use with High Speed Packet Access (HSPA) networks, called Circuit Switched over HSPA (CSoHS).

1.2.1 Introduction to Jitter and Jitter Buffer

When using a shared channel or a packet switched network for transmission (e.g. IP networks), the network introduces variation in the media delivery rate due to network congestion, packet queuing, different routes, etc. For voice services, this variation needs to be equalized before the decoder presents the encoded media to the user, otherwise it may give rise to severe quality degradations rending the service useless. Normally, this kind variation is handled by using a so-called jitter buffer, as shown in

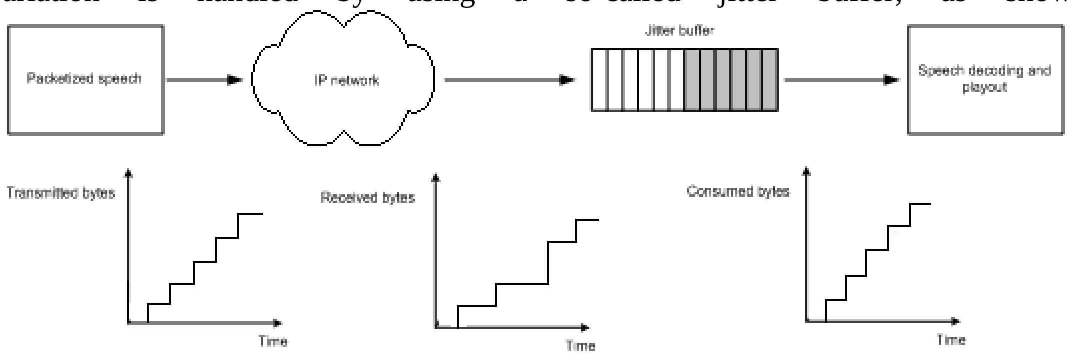

Figure 1.1. Note that more formally this is a “de-jitter buffer”, however, we will follow common usage and refer to it as a jitter buffer in the remainder of this thesis.

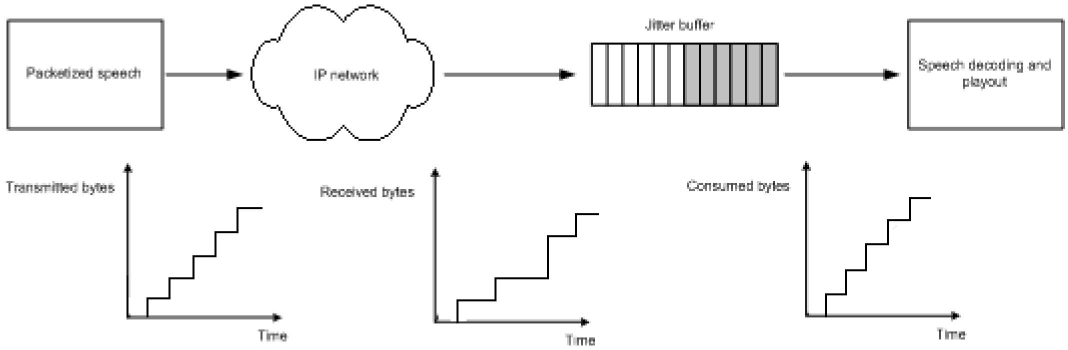

Figure 1.1: Jitter and jitter buffer in end-to-end IP network

The basic function of is this jitter buffer is to collect data, then deliver this data to the decoder at the expected (often constant) rate. Further details of the jitter buffer are discussed in Section 2.3.

1.2.2 Distributed Jitter Buffers in CSoHS

HSPA is a collection of mobile telephony protocols that extend and improve the performance of existing third-generation (3G) cellular telephony technologies. It includes both the uplink (UL), High Speed Uplink Packet Access (HSUPA) and downlink (DL), High Speed Downlink Packet Access (HSDPA) extensions. The Third Generation Project Partnership (3GPP) Release 7 specification introduced HSPA. Several benefits occur by running circuit switched voice over HSPA according to [20] and [21], these are:

- It helps save battery power in the user terminals. This is because in CSoHS it is possible to deliberately queue up data blocks in the transmitter, then send multiple data blocks at the same time. However, this increases both the delay and the jitter at the source.
- In normal 3G, the receiving side must always be ready to receive signals, while in HSPA or CSoHS, the User Equipment (UE) on the downlink knows that there will be a transmission only every “Hybrid Automatic Repeat request (HARQ) round trip time” – thus the HARQ processing at the receiver can be turned off when it is known to be idle.
- Less jitter needs to be handled by handling the UL and DL separately, than handling the jitter introduced by the whole route.

In CSoHS networks, the HSUPA (uplink) uses a fast retransmission scheme to ensure a low block error rate (BLER). The HSDPA (downlink) is a shared channel that also uses fast retransmission, as well as channel dependent scheduling between users. Therefore, additional jitter is introduced on both the uplink and downlink traffic. Additionally, because in CSoHS the transport network is a traditional circuit switched network, this requires that frames are delivered regularly and continuously, for example one frame every 20ms (the frame rate depends on the frame length of the speech CODEC scheme used; different CODECs may use different frame lengths, e.g. 10ms, 30ms). Uplink jitter therefore needs to be equalized before the frames enter the circuit-switched backbone network. This is done by implementing a jitter buffer at the Radio Node Controller (RNC) in the radio access network. Similarly, the jitter introduced on the downlink traffic is equalized by the jitter buffer in the receiving terminal as shown in Figure 1.2.

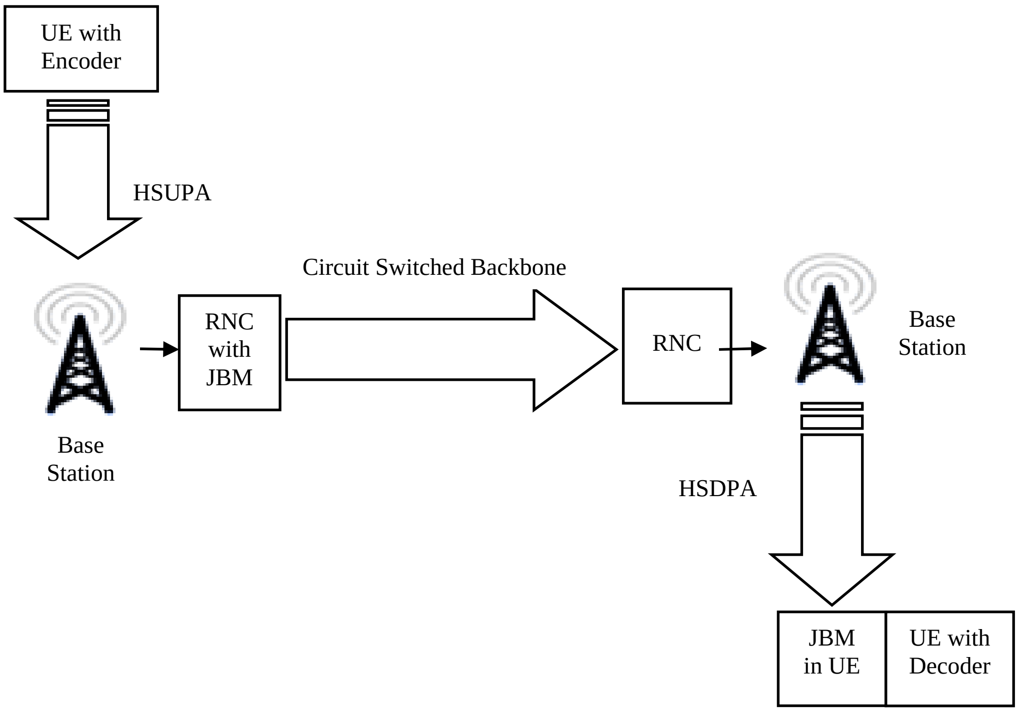

Figure 1.2: Distributed jitter management in CSoHS.
UE: User Equipment, RNC: Radio Network Controller, and JBM: Jitter Buffer Management

In the above picture, the UE is one of the mobile devices. The RNC is responsible for controlling multiple base stations. JBM is the jitter buffer management function. There is also a speech decoder in the core network that can only receive speech frames (or SID or NO_DATA) every 20ms. Thus the radio network must deliver one frame exactly every 20ms to the core network (CN). This CODEC in CN decodes the speech to G.711 PCM (either A-lay PCM or my-law PCM). This CODEC implementation is exactly the same as in legacy CS networks (W-CDMA or GSM) and therefore has no jitter buffer.

1.3 Structure of the Thesis Project

The goal of this thesis project was to implement and evaluate distributed jitter buffers; i.e., separately for the uplink and downlink of CSoHS. In order to achieve this goal, the thesis project was carried out in three steps: a literature study, practical implementation, and an evaluation of the results – as summarized in Table 1.1.

Table 1.1:Structure of the Thesis Project

Literature study	Gain knowledge of Jitter buffer, HSPA, CSoHS, etc.
Practical implementation	Design a jitter buffer management function on the HSPA Radio Link Control (RLC) layer and integrate it into the existing simulator.
Evaluation of results	Simulate different delay and error profiles and analyze the results to evaluate the performance of the jitter buffer management function.

1.4 Outline of this Thesis

This thesis presents the results of all three steps described in section 1.3. The thesis is divided into five main chapters as shown in Table 1.2.

Table 1.2: Outline of the Thesis

Title of the Chapter	Content of the Chapter
Introduction	• Statement of the problem • Introduction • Overview of the thesis
Background	• Results of the literature study; Presenting the required background knowledge and relevant prior work
Method	• Output of practical implementation Demonstration of the JBM design and simulation environment • Presentation of the criteria for evaluation
Analysis	• Results of the evaluation Discussion of the simulation results
Conclusion	• Comments and conclusions • Future work

2. Background

This chapter presents the results of the literature study including background knowledge and relevant works that have previously been done. It consists of four sections.

- Section 2.1 gives an overview of the architecture and presents some necessary basic concepts;
- Section 2.2 gives an introduction of Adaptive Multi-Rate speech CODEC;
- Section 2.3 explains jitter buffer management (JBM) techniques in detail;
- Section 2.4 describes HSPA networks with a focus on those features that influence delay and jitter;
- Section 2.5 presents relevant prior work regarding CSoHS; including its delay and jitter characteristics, and some notes regarding the design of a JBM function.

2.1 Overview of the Architecture

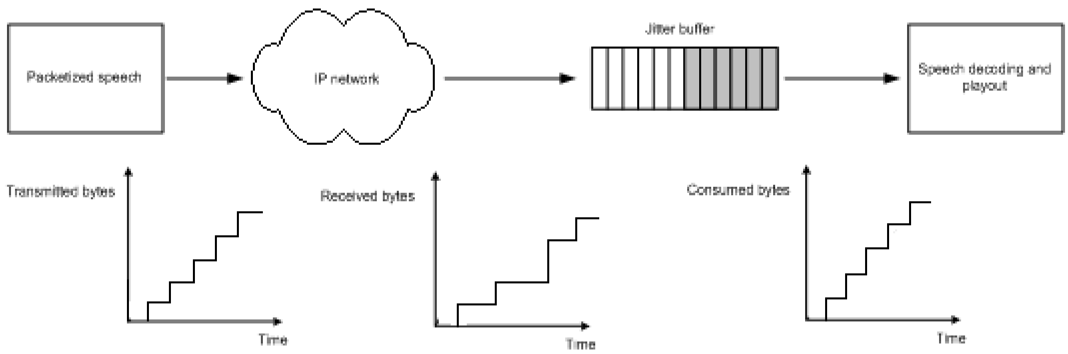

Figure 1.1 on page 2 illustrated the general architecture of a VoIP service. The voice signal is encoded into frames by an encoder. One or more frames are encapsulated into a packet (e.g. an RTP packet) which is transmitted across the network. At the receiving end, the delay jitter is equalized by the jitter buffer, then frames are delivered to the decoder at the expected rate. Finally, a voice signal is played out after the frames are decoded. The following sub-sections present a basic explanation of the concepts that are used in this general architecture.

2.1.1 Nature of Speech

This thesis focuses on conversational voice service. Figure 2.1 shows an example of what a typical speech signal looks like.

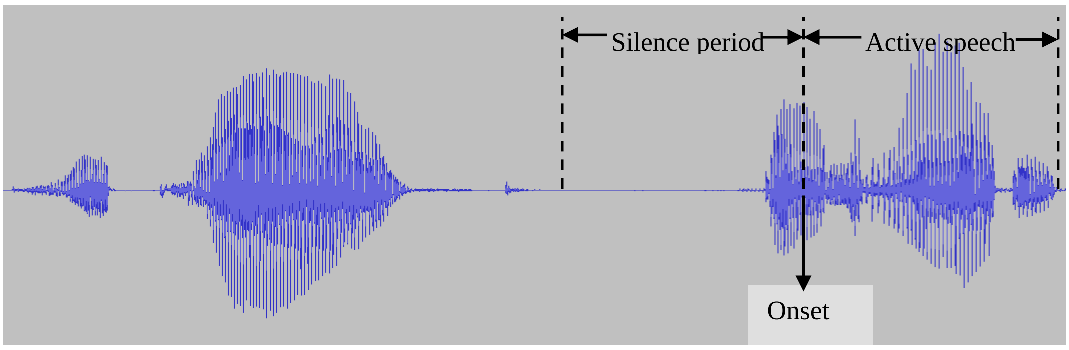

Figure 2.1: Pattern of speech signal

The x-axis is the time and the y-axis is the amplitude of the speech samples. As shown in the figure, the speech signal contains a mixture of active and silence periods.

- An *active period* contains the actual speech that comes from microphone. It may contain a mixture of speech and background noise.
- Each *silence period* is a pause in-between active periods and may or may not contain background noise (i.e., sounds from the surrounding environment).

An important concept is a *talk-spurt*. A talk-spurt is a period of continuous active speech between two silence periods. The beginning of a talk-spurt is referred to as the *onset* of the talk-spurt.

2.1.2 Network Impairments

In packet switched networks, packets experience various delays during transmission across the network; due to different routes being selected for different packets, different amounts of queuing at each of the routers along the path, shared transmission resources, etc. Some packets may even be lost. Figure 2.2 shows the impairments introduce by the network.

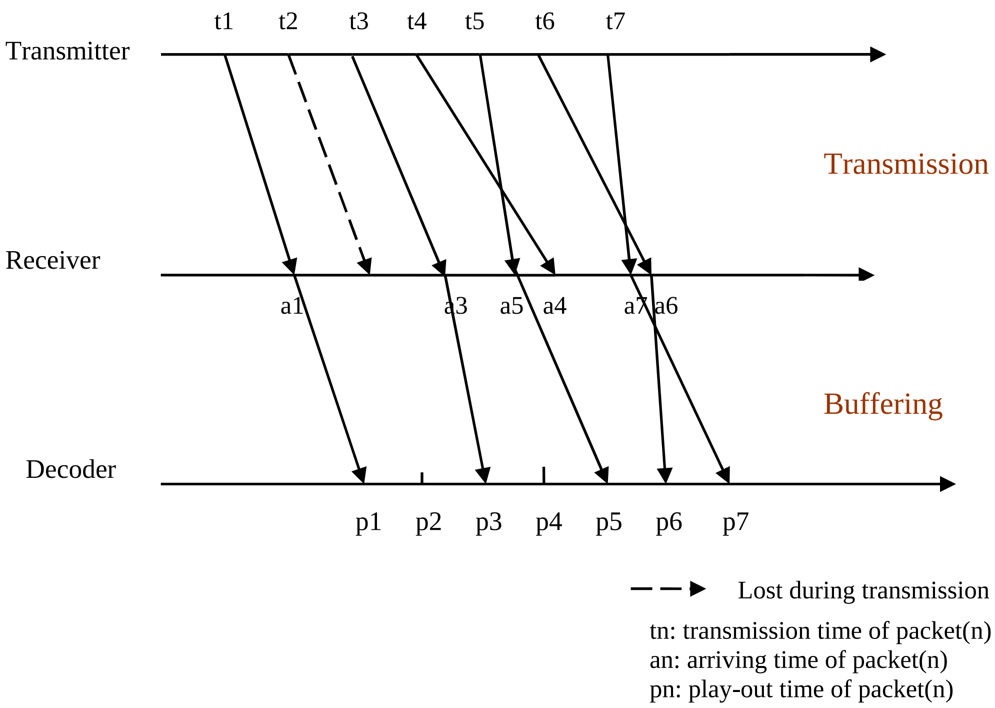

Figure 2.2: Network impairment

In the scenario illustrated above, jitter was introduces during network transmission. As a result packets arrive at the receiving end at irregular intervals. Packet 2 was lost during transmission, while packets 4 and 5 and packets 6 and 7 arrived out of sequence. The voice quality would be degraded if these received packets were decoded and played immediately without any pre-processing (such as packet re-ordering). This is why the jitter buffer is needed. Packets are stored in the jitter buffer for some time in order to reorder them, so that they can be delivered in-sequence to the decoder at regular intervals. Note that despite its successful arrival at the destination node, packet 4 is discarded because it arrives later than its expected play-out time. As there are no corresponding packets to be played at time p2 and p4, the gap may be covered by error concealment, which will be discussed in a later section.

2.2 Adaptive Multi-Rate Speech CODEC

The Adaptive Multi-Rate (AMR) CODEC is an audio data compression scheme optimized for speech coding and was originally designed for circuit-switched mobile radio systems. It has been adopted as the standard speech CODEC by 3GPP, both for GSM and 3G. The AMR speech coder consists of a multi-rate speech coder, a source controlled rate scheme including a voice activity detector and a comfort noise generation system, and an error concealment mechanism to combat the effects of transmission errors and lost packets [17].

However, due to its flexibility and robustness, it is also suitable for real-time speech communication services over packet-switched networks [3]. AMR is the standard CODEC for the Multimedia Telephony Service for IMS (MTSI). MTSI, also referred to as Multimedia Telephony, is a standard IMS (IP Multimedia Subsystem) telephony service that has been specified in 3GPP Release 7 [9].

2.2.1 Coding Modes

The sampling frequency of narrow band AMR is 8kHz, which results in 8000 samples per second. One AMR frame is 20ms long and therefore contains 160 samples. AMR supports 8 speech coding modes as shown in table 2.1. It uses link adaptation to select one of these eight different bit rates based on link conditions [3].

Table 2.1: AMR coding modes

Mode	12.2	10.2	7.95	7.40	6.70	5.90	5.15	4.75	AMR_SID
Bit rate (Kbits/s)	12.2	10.2	7.95	7.40	6.70	5.90	5.15	4.75	1.80

2.2.2 Silence Suppression

Silence suppression is a technique to reduce the bandwidth required during silence periods or background noise periods. AMR supports voice activity detection (VAD) and generation of comfort noise parameters during silence periods. The operation of sending only comfort noise parameters at regular intervals during silence periods is called discontinuous transmission (DTX). DTX was originally designed for circuit-switched cellular systems to reduce the interference level (giving a better carrier to interference ratio (C/I) for other users) and to save battery power. The CODEC can reduce the number of transmitted bits and frames to a minimum during silence periods. The AMR frames containing comfort noise parameters are called silence indicator (SID) frames [13].

2.2.3 Error Concealment

Frames may be lost due to transmission errors. Some action should be taken in these cases, both for lost speech frames and for lost SID frames. Error concealment actions can also be used in the case of speech packets lost in the transport network. In order to mask the effect of isolated lost frames, the speech decoder should be informed, so that error concealment shall be initiated. Concealment is generally done by using a set of prediction parameters to synthesize the missing speech. Insertion of speech signal independent silence frames is not allowed as stated in [3]. For subsequent lost frames, a muting technique can be used to indicate to the listener that transmission has been interrupted [17]. More explicit description of error concealment can be found in [18].

2.3 Jitter Buffer Management

The necessity of using the jitter buffer has been discussed in Section 1.2. This section gives a more explicit presentation of different jitter buffer techniques.

2.3.1 Types of Jitter Buffers

There are basically two types of jitter buffer: static and adaptive.

2.3.1.1 Static Jitter Buffer

A static (or fixed) jitter buffer simply collects frames then delivers frames to the speech decoder at the expected time intervals to ensure a smooth play-out rate. A static jitter buffer does not react to changes in network conditions. Thus static jitter buffer exhibits a constant end-to-end delay during the whole length of a communication session.

2.3.1.2 Adaptive Jitter Buffer

Just opposite of the static jitter buffer, an adaptive jitter buffer may change the end-to-end delay during a session in order to optimize the trade-off between buffering delay and buffer induced frame losses. Generally, the buffering time can be modified at two different ways, during talk-spurts and in silent periods. The algorithm will estimate the needed buffering time continuously and update it when possible [10]. We will consider each of these alternatives below.

- Update during silence periods

  The main method to adjust the buffering time is to change the length of silent periods as shown in Figure 2.3.

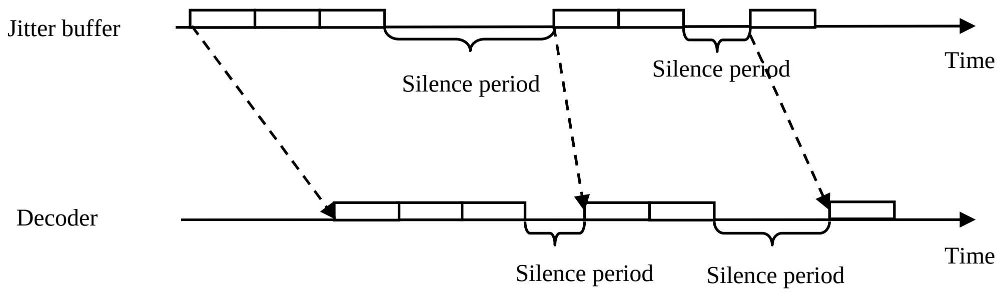

Figure 2.3: Adaptation in silence period

The jitter buffer is always set to an initial buffer level measured in an integral number of packets, which means that the jitter buffer will only start delivering packets to the decoder once it collects this number of packets. If the jitter buffer detects a silent period, a new initial buffering level will be calculated and applied at the beginning of the next talk-spurt. In this fashion the adjustments can be large enough to adapt to large changes in the network conditions.

Note that in this approach the receiver is using the silence periods to catch up the sender (i.e., to reduce the end-to-end delay). As a result the end-to-end delay will **not** continue to grow over the duration of a session (as long as there are sufficient silence periods).

- Update during talk-spurts

  However, adaptation only during silence might not be sufficient if the delay jitter increases abruptly during a talk-spurt – as the above algorithm has to wait until the next spurt. Therefore, it is also desirable to change the buffering time *during* a talk-spurt. A simple method is to simply add a gap (via a dummy frame or NO_DATA) and let the error concealment mechanism try to conceal the gap. A more advanced way is so-called time-scaling based upon interpolation or decimation of speech frames [10]. If packets arrive slower than they are consumed then the buffering time has to be increased to avoid buffer under-run. In this case, interpolation could be applied. Interpolation produces a longer frame, hence the play-out duration for the frame will be extended, which will increase the following frame's buffering time. If on the other hand, packets arrive faster than they are consumed, then the jitter buffer has to play out packets faster to avoid buffer overflow. Using decimation the frame length is shortened and buffering time for following packets will be reduced. Time-scaling can often be done on the decoded speech frames (although it is also possible with some CODECs to perform the time-scaling of the encoded frame). Note that the changes should **not** be done too often nor should the changes be too large, since this could result in unnatural sounding speech and/or unsatisfactory speech quality.

  Note that interpolation and decimation may be needed even if there is no jitter, as the sampling clocks of the source and destination may not have exactly the same rate.

2.3.2 Jitter loss and Jitter Induced Concealment

Sometimes packets are successfully transmitted to the receiver side, but may be discarded by the JBM because of:

- Buffer overflow or intentional packet dropping when reducing the buffer's depth during adaptation and
- Packets arriving at the jitter buffer after its scheduled play-out time, also known as late loss.

In this these we have assumed that the jitter buffer always has enough buffer capacity to store packets, hence no speech frames need to be discarded during adaptation because of overflow. Thus, jitter loss is only due to late loss. In order not to significantly reduce the speech quality, the amount of JBM induced frame loss should be kept below a certain value.

$$Jitter\_loss\_rate = \frac{JBM\_induced\_frame\_losses}{Number\_of\_transmitted\_frames} \qquad \text{(Eq.1)}$$

It was recommended in [9] that the jitter loss rate should be kept below 1% over the entire communication session. Additionally, the jitter loss rate is calculated only for speech frames because the loss of SID frames is known to cause very little degradation in comparison to losing a speech frame.

Sometimes the JBM has to insert dummy (or NO_DATA) frames in order to cover gaps. This may happen in the following cases:

- Buffer under-run because the jitter buffer is empty and has no frame to deliver to the decoder when it is requested to do so or
- The expected packet has not arrived at the jitter buffer (possibly because it was lost in transmission or experienced too long delay).

These JBM introduced dummy frames are sent to the decoder to activate error concealment.

2.3.3 Performance Requirements for JBM

In order not to significantly degrade the voice service, there are some basic requirements that any JBM has to achieve. As suggested in [9], these performance requirements are:

1. The JBM shall minimize the buffering time at all times - while still limiting jitter loss;
2. If the jitter loss limit cannot be met, then it is always preferred to increase the buffering time in order to reduce the jitter loss; and
3. If sample-based time scaling is used (time-scaling performed after the speech decoder), then artefacts caused by time scaling shall be kept to a minimum.

These requirements were originally proposed in [9] for JBM in Multimedia Telephony. However, they will also be used as guidelines for our JBM design.

2.4 HSPA

As stated previously HSPA consists of two standards: High Speed Downlink Packet Access (HSDPA) and High Speed Uplink Packet Access (HSUPA).

2.4.1 HSDPA

2.4.1.1 General Features

In 3GPP's WCDMA Release 5, HSDPA introduces a new transport channel, the High Speed Downlink Shared Channel (HS-DSCH). This provides a greatly enhanced system capacity and much higher user data rates for the downlink (i.e., transmissions from the radio access network's base station to the mobile terminal). The theoretical peak data rates can be up-to 14.4Mbit/s. Generally, HSDPA has the following features ([4] and [8]):

- Shared channel and multi-code transmission

  Shared channel transmission means that some channel (spreading) codes and the transmission power are a common resource and can be dynamically shared between users in the time and code domains. This results in more efficient use of the available codes and transmission power.

- Higher-order modulation

  3GPP's WCDMA Release 99 uses Quadrature Phase Shift Keying (QPSK) modulation for downlink transmission. In addition to QPSK, HSDPA can also use 16 Quadrature Amplitude Modulation (16QAM) to provide higher data rates.

- Fast link adaptation

  The radio channel conditions experienced by different downlink communication links vary significantly. Each user terminal that uses high-speed services transmits regular channel quality reports to the base station. Fast link adaptation adjusts the transmission parameters based upon the instantaneous radio conditions reported by the terminal and (when channel conditions permit) this enables the use of high-order modulation for communication with a terminal that currently has good communication conditions.

- Shorter Transmission Time Interval (TTI)

  In HSDPA, the TTI is reduced to 2ms for the downlink as compared to 10ms, 20ms, or 40 ms used in 3GPP's WCDMA Release 99. This reduces the round-trip time between the UE and the base station and improves the tracking of instantaneous channel variations, which in turn can be utilized for link adaption and fast scheduling.

- Channel dependent scheduling

  Channel dependent scheduling is a major source of jitter in HSDPA. This feature insures that the shared channel transmission is utilized by the users with the most favourable channel conditions at any given moment, as shown in Figure 2.4.

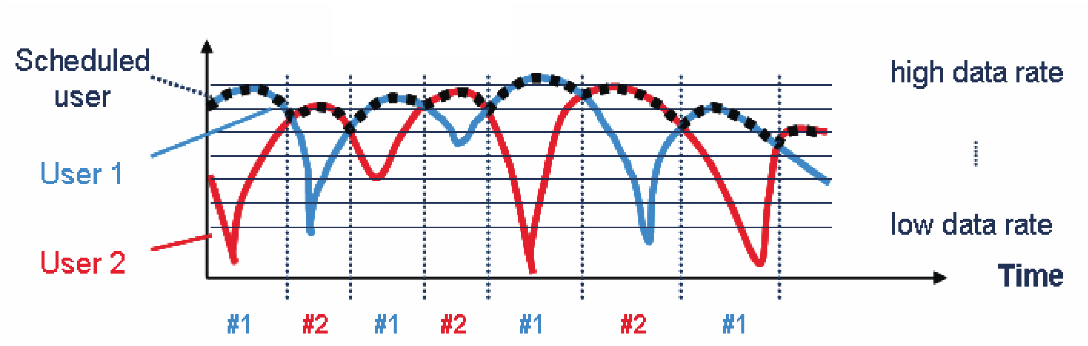

Figure 2.4: Schedule users with favourable channel conditions [8]

The scheduler estimates the instantaneous radio conditions of the downlink channel. Each UE that uses HSDPA services transmits regular channel quality report to the scheduler in the base station. For each TTI, the scheduler decides which user the HS-DSCH should be allocated to. In addition, the scheduler can also take traffic priority into account. Usually, retransmissions are prioritized over scheduling of new data. Another prioritization is that real-time media and streaming services can be given higher priority than best-effort data traffic.

- Fast Hybrid Automatic Repeat reQuest (HARQ) with soft combining

Fast Hybrid Automatic Repeat reQuest (HARQ) with soft combining is another major source of jitter in HSDPA. The UE can rapidly request the retransmission of missing data and can combine information from the original transmission with the later retransmission before decoding the signal (called soft-combining). There is one HARQ entity per user and each entity consists of multiple HARQ processes (up to 8) to allow for continuous transmission to a single UE. A negative acknowledgement (NACK) reply is sent when data is missing at the receiving end. An acknowledgement (ACK) reply is sent when data is received correctly. The HARQ protocol is shown in Figure 2.5.

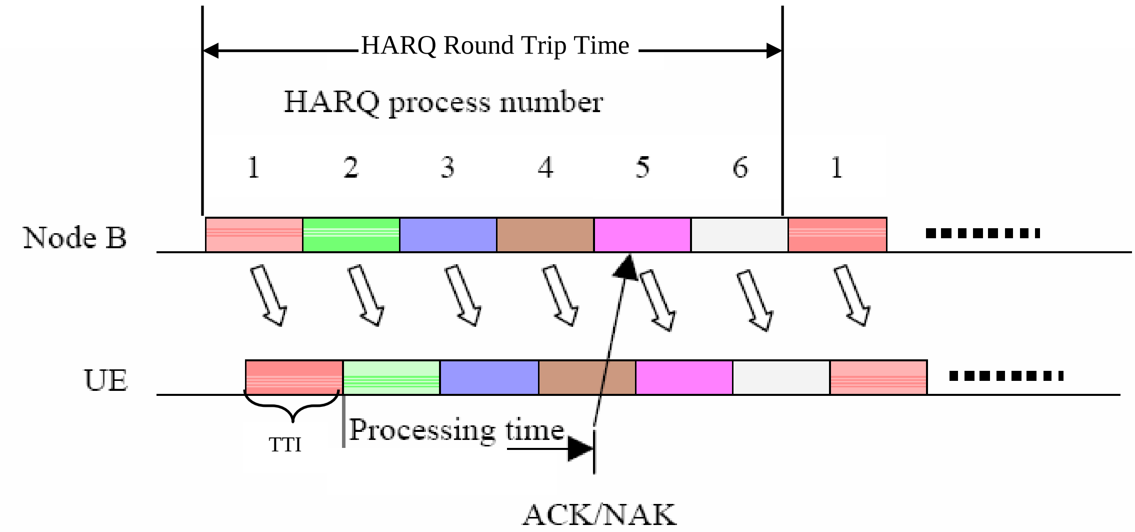

Figure 2.5: Multiple HARQ processes (6 assumed) [4]

$$HARQ\_round\_trip\_time = TTI \times HARQ\_process\_number \qquad \text{(Eq.2)}$$

Previously, retransmissions were handled by the Radio Node Controller (RNC), but in HSDPA this functionality has been moved to the base station (Node B), which resides closer to the air interface, hence the retransmission latency is reduced.

In HSDPA the HARQ, together with channel dependent scheduling determines the delay jitter of transport blocks. A drop timer defines the maximum delay. This value will be configured by the RNC, then delivered to Node B so that the scheduler can schedule its transport blocks according to this value. Any transport blocks that experience a longer delay than this drop timer are considered to arrive too late and will be discarded (generating a loss).

2.4.1.2 Architecture

HS-DSCH is a new transport channel which provides a service at the physical layer to the MAC layer. Therefore a new functional entity of the MAC layer called MAC-hs was introduced and the physical layer was updated with new functionalities as well. The radio interface protocol architecture is shown in Figure 2.6. The new MAC-hs entity was placed in Node B as this is close to the UTRAN access point in order to achieve the desired signalling speed.

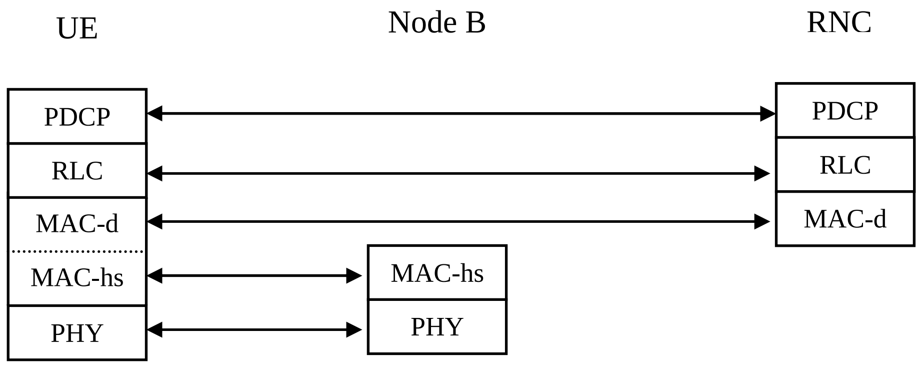

Figure 2.6: Radio interface protocol architecture of HS-DSCH

Each layer provides certain services with a number of functions. Here we shall only discuss those functions with close relevance to our work. A detailed description can be found in [12].

2.4.1.2.1 MAC Functions

The MAC layer comprises several MAC entities, including MAC-hs and MAC-d, as shown in the above figure. These MAC entities manage the following functions [12]:

- HARQ

  In HSDPA, the MAC-hs (in HSUPA this will be the MAC-e/MAC-es) is responsible for establishing the HARQ entity and perform HARQ.

- In-sequence delivery and assembly/disassembly of higher layer protocols data units (PDUs).

  In HSDPA the transmitting MAC-hs (in HSUPA this will be the MAC-es/MAC-e) entity assembles payload of the MAC-hs PDUs (or MAC-es PDUs in HSUPA) from the MAC-d PDUs, then adds a MAC-hs header. The receiving MAC-hs (MAC-es) entity is responsible for reordering of the received data blocks according to the transmission sequence number (TSN) included in the MAC-hs (or MAC-es) header, then disassembling the data block into MAC-d PDUs and delivering them in sequence to the higher layers. (A. shows details of the PDUs and Service Data Units (SDUs).)

This functionality facilitates our work because the JBM is implemented on the RLC layer and does **not** have to reorder the received RLC PDUs from the MAC layer, as they have already been re-ordered. However, it should be noted that this reordering by the MAC layer **increases** the delay when the PDUs are not successfully received in order, as the MAC layer will buffer the out of order PDUs and wait for the missing PDU. As noted earlier this will increase jitter.

2.4.1.2.2 RLC Functions

The RLC layer can operate in three different modes [12]:

1. Acknowledged Mode (AM)

   This mode is typically used for data (web) traffic. In AM, upper layer PDUs are transmitted with guaranteed delivery to the peer entity. This is achieved by RLC retransmissions. If the HARQ functionality fails, then the data will be retransmitted by the RLC. However, the RLC retransmission will only be required in very rare circumstances, for example during handover. Note that in HSDPA only hard handover is supported. A hard handover means that the connection between the UE and the Node B is broken before the connection to the new Node B is established. Without RLC based retransmission hard handover might cause data loss [16].

2. Unacknowledged Mode (UM)

   In UM, upper layer PDUs are transmitted without guaranteed delivery to the peer entity. In other words, RLC retransmission is **not** used in this mode. UM is the normal mode for real-time media since RLC retransmissions add quite a lot of jitter.

3. Transparent Mode (TM)

   In TM, upper layer PDUs are transmitted without adding any protocol information, possibly including segmentation/reassembly functionality. If segmentation has been configured and a RLC SDU is larger than the RLC PDU size used by the lower layer for that TTI, the transmitting TM RLC entity segments RLC SDUs to fit the RLC PDUs size without adding RLC headers. All the RLC PDUs carrying one RLC SDU are sent in the same TTI, and no segment from another RLC SDU are sent in this TTI. If segmentation has not been configured, then more than one RLC SDU can be sent in one TTI by placing one RLC SDU in one RLC PDU. All RLC PDUs in one TTI must be of equal length [22].

In this thesis we assume that only RLC UM is used because the RLC retransmission functionality under AM may add excessive delay and the lack of fast retransmission in TM would lead to too many lost frames.

2.4.1.2.3 PDCP Functions

The most relevant functions for this work that the Packet Data Convergence Protocol (PDCP) can provide are [12]:

1. Header compression and decompression of IP data streams (e.g. TCP/IP header, RTP/UDP/IP header).
2. PDCP AMR Data PDU

In order to enable CSoHS a new type of PDCP PDU is defined: AMR Data PDU. The header of the PDCP AMR Data PDU is of one octet length, where the first 3bits distinguish AMR frame types and the other 5 bits provides the PDCP PDU with an AMR counter as timestamp.

2.4.2 HSUPA

The improvements of the downlink were driven mainly by data (web) traffic. However, it was discovered that fast feedback for the uplink was also important in order to adapt the uplink bit rate to high rates. Hence, as a complement to HSDPA, 3GPP's WCDMA Release 6 introduced HSUPA, also known as Enhanced Uplink (EUL), which added a new transport channel called the Enhanced Dedicated Channel (E-DCH), with a peak data rate of up to 5.8Mbit/s.

2.4.2.1 General Features

Similarly to the HS-DSCH, E-DCH transmission is based on the following basic principles ([4] and [8]):

- Shorter TTI

  HSUPA uses 2ms or 10ms TTI instead of 10ms, 20ms, or 40ms in as in the earlier 3GPP WCDMA Release 99. The shorter TTI reduces overall latency and enables the other features to adapt rapidly.

- Fast scheduling

  Unlike the downlink, the common resource shared among terminals for the uplink is the amount of tolerable interference, which is related to the total received power at the base station. The amount of this common uplink resource used by a terminal depends on the data rate that is being used. Normally, a higher data rate requires greater transmission power, hence consuming more of this uplink resource. The overall target of the uplink scheduler is to rapidly reallocate this common resource between UEs, with a larger fraction of this resource being assigned to users that momentarily require higher data rates, while keeping the system's operation stable by avoiding sudden interference peaks.

  In addition, channel dependent scheduling can be also optionally used as on the DL. However, this was not considered in this thesis.

- HARQ with soft combing

  This is similar to the HARQ used for HSDPA. The base station can rapidly request retransmission of erroneously received data and combine them with previously successfully received information. In case of 10ms TTI, 4 HARQ processes are configured; while in the case of 2ms TTI, 8 HARQ processes are configured as specified in [13].

  If channel dependent scheduling is *not* applied in HSUPA, then HARQ is the major source of delay jitter. Similarly to HSDPA, a drop timer is also configured in HSUPA by the RNC, this determines the maximum number of HARQ retransmissions.

2.4.2.2 Architecture

Similar to HS-DSCH, E-DCH is a new transport channel. Hence, a new MAC entity, MAC-e was added in Node B, to handle HARQ retransmissions, scheduling, etc. Another new MAC entity, MAC-es was added to the RNC to perform reordering and combining data from different Node Bs in case of soft handover. Compared to hard handover, soft handover allows the UE to be connected to multiple Node Bs in parallel[16]. Thus, soft handover avoids the data losses that may occur for hard handover. Figure 2.7 shows the radio interface protocol architecture of E-DCH.

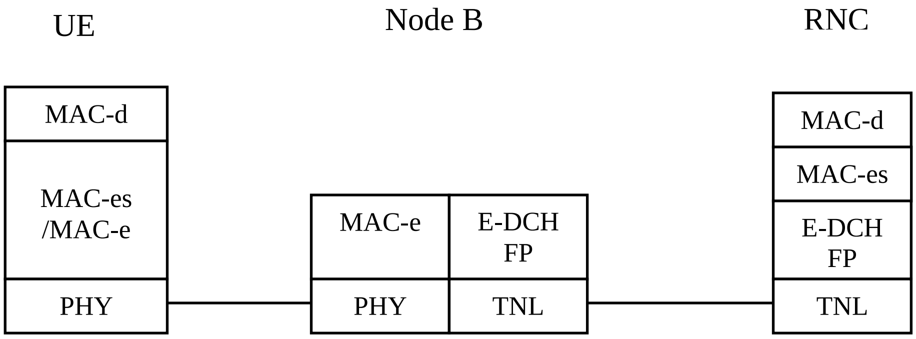

Figure 2.7: Radio interface protocol architecture of E-DCH

2.5 Circuit Switched over HSPA (CSoHS)

The motivation for running CSoHS has been explained in Chapter 1. . This section summarizes some relevant prior work regarding CSoHS, especially the delay budget and some notable differences in JBM design compared with a VoIP system.

2.5.1 Delay Budget of CSoHS

In CSoHS, jitter is caused mainly by HARQ for both UL and DL; while channel dependent scheduling is another major source of jitter – but in this thesis we will only consider this for the DL. The maximum delay and frame error rate after HARQ are controlled by the RNC. The UL and DL scheduling parameters are set by different RNCs independently and each connection will have its own jitter buffer.

In order not to degrade the quality of service, a delay budget was proposed in [7]. The allocation of this budget to different potential sources of delay is shown in

Table 2.1. Note that the sum of all of the parts of the delay budget sets a bound on the maximum delay. Each of these components will be explained in the following sections.

Table 2.1: Delay budget of CSoHS

	Uplink delay components	RAN/CN processing	Online transmission	Downlink delay components
Speech encoding	35 ms	-	-	35 ms
Air interface	50 ms	-	-	26 ms
Speech decoding	5 ms	-	-	5 ms
Scheduling	-	-	-	80 ms
Sum	90 ms	40 ms	10 ms	146 ms

2.5.1.1 UL Delay

In CSoHS, we consider 10ms TTI and 2ms TTI separately.

- 10 ms TTI

  Maximum of 1 retransmission with a residual BLER<1%

  As discussed in section 2.4.2, 4 HARQ processes are configured, according to (Eq.2), the HARQ round trip time is: *40ms = 10ms TTI × 4*

  The resulting radio interface delay is: *50 ms = 10ms TTI + 40ms jitter*

- 2ms TTI

  The typical assumption is that there will be a maximum of 3 retransmissions with a residual BLER < 1%. A maximum of one or two retransmissions could also be used. The actual maximum number of retransmissions is a configuration parameter under the RNC’s control.

With 8 HARQ processed configured, the HARQ round trip time is: *16ms = 2ms TTI × 8*

The resulting radio interface delays are:

18ms = 2ms TTI + 16ms jitter × 1 (with a maximum of 1 retransmission)

34ms = 2ms TTI + 16ms jitter × 2 (with a maximum of 2 retransmissions)

50ms = 2ms TTI + 16ms jitter × 3 (with a maximum of 3 retransmissions)

So it can be concluded from that the radio interface delay for CSoHS over E-DCH is expected to range from 18ms to 50ms - depending on the network settings. The RNC is responsible of setting the operating parameters (TTI and drop timer). The RNC should also take its total jitter buffer capacity into account as it must receive transmissions from multiple UEs. When setting these parameters a maximum delay by the RNC of 50ms should be observed in order to assure good quality of service.

2.5.1.2 DL Delay

When sending circuit-switched voice over HSDPA, only a 2ms TTI is used. Assuming a maximum of 2 retransmissions with a target residual BLER < 1% and 6 HARQ processes configured, the radio interface delay could be:

- *14ms – with 1* retransmission
- *26ms – with 2* retransmissions

As noted earlier another source of jitter is channel dependent scheduling. The scheduling delay budget is a trade-off between capacity and delay. A longer maximum scheduling time implies somewhat greater capacity. The jitter buffer in the UE needs to compensate for the delay variance introduced by scheduling and HARQ.

A typical HSDPA voice scheduling delay budget would be 50ms to 80ms. However, a scheduling delay of up to 150ms could be considered if increased capacity is more important in the operator’s network; although this may degrade the quality experienced by a user. The drop timer configured by the RNC is delivered to the Node B, thus the scheduler will schedule the DL packet based upon this value. The operator can choose the scheduler delay budget according to their own preference (i.e., shorter delay or greater capacity). However, a maximum delay needs to be defined in order to determine the maximum jitter buffer size in the UE and to avoid exceeding the overall end-to-end delay.

As the 150ms scheduling delay is considered to be too long, 80ms is used. So the maximum DL delay is: 106ms = 26ms air interface + 80ms scheduling.

2.5.1.3 End-to-End Delay

Besides the delay of the air-interface, there are many other factors influencing the end-to-end delay. In the proposed delay budget, 30ms RAN/CN processing delay and 10ms transmission delay were assumed. Moreover, the speech encoding and decoding delays are 35ms and 5ms respectively. Therefore the maximum end-to-end delay of CSoHS is:

276ms = 90ms UL + 30 ms RAN/CN processing + 10ms transmission on lines + 146ms DL

The quality of service requirement in 3GPP's Technical Specification 22.105 for real-time conversational voice recommends a preferred mouth-to-ear delay <150ms and a maximum delay of 400ms with a speech frame erasure rate <3%. Thus the proposed overall delay of 276ms is considered to be acceptable (although it is almost twice as high as would be desirable) and it leaves *no room for delay anywhere else* in the mouth-to-ear path.

2.5.2 JBM for CSoHS

Numerous studies have been conducted of JBM for VoIP. According to [10], the principles are also applicable to JBM for CSoHS, but with some differences:

1. JBM for VoIP utilizes a time stamp and sequence number in the RTP packet header, while circuit-switched speech frames do not carry such timing information.
2. In VoIP, the way to detect a talk-spurt onset is to check the marker bit in the RTP header, while for CSoHS the RLC has to detect this onset. However, this is easily done by checking the size of the transport block that the RLC receives.

In order to utilize VoIP JBM designs for CSoHS, both time stamp and sequence number information needs to be provided to the JBM just as in RTP.

- To emulate a RTP time stamp, a new PDCP AMR Data PDU was defined where the last 5 bits in the header form a field called the AMR counter [10]. This field is used as a (relative) time stamp.
- The sequence number in the RLC UMD frame is used as is to emulate a RTP sequence number.

At the transmitting side, one AMR frame is provided to the PDCP layer every 20ms, and the AMR counter increments with each AMR frame. NO_DATA frames are generated during DTX if there is no SID_frame. However, if the AMR frame is of type NO_DATA, then no PDCP PDU will be generated. Thus only SID frames will be transmitted during the silence period. During non-silence periods, one PDCP PDU will be passed to RLC layer every 20ms.

At the receiving side, the JBM will forward an AMR frame every 20ms synchronously to the AMR decoder. If JBM detects a silence period or a lost packet based on the AMR counter and the RLC sequence number, it will locally generate a NO_DATA or Speech lost packet and deliver this to the speech decoder to cause the decoder to activate error concealment.

3. Method

This chapter describes how the implementation and evaluation were carried out. The chapter introduces the simulation environment that was used. Following this a demonstration of the JBM design is given along with the criteria to be used for the evaluation.

3.1 Simulation Environment

This section gives a brief introduction to the simulator, and explains the design of the JBMs and the delay and error profiles.

3.1.1 Simulator

We used an existing simulator which was previously used for VoIP simulations. This simulator is mainly implemented in C++. Moreover, there are already existed different types of JBMs implemented in simulator. (Note that this is **not** a HSPA simulator, but is a VoIP simulator that is being adapted to study delay and jitter which are determined by the delay and error profiles.) Unfortunately, the details of the simulator could not be further revealed due to its confidentiality.

The main issue when using this simulator was that the JBMs were integrated in the speech decoder. However, the JBM for CSoHS is required to be implemented separately from the speech decoder on a lower layer. Therefore the existing JBMs are disabled and new JBMs were implemented separately from the decoder. Furthermore, special care was taken to avoid taking advantage of any mechanisms available in IP, UDP, or RTP that would **not** be available in CSoHS. The simulation chain is shown in Figure 3.1.

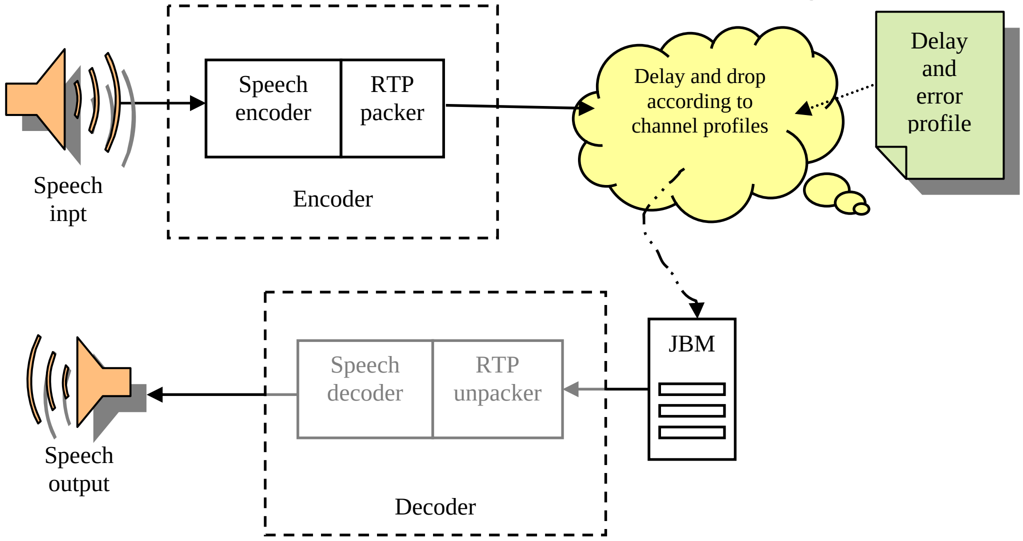

Figure 3.1: Simulation chain

The simulator can simulate communication in both directions. In this thesis only one direction was used, as shown in the figure. The speech CODEC used is AMR12.2 with DTX enabled. Only one AMR frame is contained in an RTP packet to simulate how speech frames are packetized into transmission blocks in CSoHS.

First of all, it has to be verified that the simulator works properly. This is done by running the c-code of AMR 12.2 obtained from 3GPP TS 26.073 with DTX enabled and running the simulator with JBM enabled – on the same audio file, but without any delay or error profiles. The result is that the two generated speech files are virtually identical except for some delay difference due to the lack of synchronization of the encoder and decoder and the JBM initialization. Thus it is concluded that the simulator performed as expected.

3.1.2 Design of CSoHS JBMs

Two types of jitter buffers were implemented: a static JBM, which as noted earlier does not change the end-to-end delay during the session; and an adaptive (or semi-static) JBM, which adapts its buffering depth at the beginning of a talk-spurt according to the network's condition. The implementation of both JBMs was done in C++. Unfortunately, time scaling was not implemented nor tested due to the limited time period for this thesis project.

As it is necessary to implement the JBM on the RLC layer, an important issue is the AMR counter mentioned in Section 2.4.1 as it contains timing information that is not accessible for a circuit-switched voice stream. Hence only the sequence number extracted from the RLC header could be utilized. Based on this limitation, a static JBM is designed conforming to the following principles:

1. The initial buffer level is set according to the drop timer configured by RNC and is an integer number of packets.

$$initial\_buffer\_level = \left\lceil \frac{drop\_timer}{packet\_duration} \right\rceil \qquad \text{(Eq.3)}$$

   The result will be the closest higher integer, if the drop timer is not evenly divisible.

2. The jitter buffer starts to output packets to the AMR decoder once the buffer depth reaches the initial buffer level.
3. The AMR decoder requires one frame every 20ms. Thereby a NO_DATA packet will be generated and delivered to the decoder whenever there is a sequence gap or buffer under run.
4. As the overall jitter loss rate needs to be limited below 1%, the JBMs for both the uplink and downlink should each (separately) maintain a jitter loss rate under 0.5%.

For the adaptive JBM, there are additional issues:

1. Adaptation is achieved during a silence period according to buffering times of the most recent packets.
2. The adaptation algorithm is statistical. The new target buffer level is derived by calculating the largest variation among the buffering times of the most recent 200 packets.

3. The buffer level is increased by inserting NO_DATA packets ahead of new talk-spurt and decreased by removing NO_DATA or SID packets between two talk-spurts.

   The pseudo code of the both types of JBM can be found in Appendix B.

3.1.3 Delay & Error Profiles

A delay or error profile is a simple ASCII or text file giving information about the network delay and packet loss. For the simulator the format was:

66 ..... *(66ms delay)*
50
-1 ..... *(Packet loss)*
18
34

The value in each line indicates the network delay of the packet in millisecond, while a negative value means a packet loss.

The delay and error profiles can either be recorded from measurements in real systems or can be generated from simulations. Using delay and error profiles in combination with the simulation framework enables complete repeatability.

3.1.3.1 Delay and Error Profiles from 3GPP

There are a number of profiles that have been used in earlier 3GPP projects [6]. These profiles are shown in Table 3.1.

Table 3.1: Delay & error profiles from 3GPP

HSUPA	HSDPA	
HSUPA_PA3_45u	HSDPA_PA3_45u_G1.65dB_55ms	
HSUPA_PB3_45u	HSDPA_PA3_45u_G1.65dB_95ms	
	HSDPA_PA3_45u_G1.65dB_100ms	
	HSDPA_PA3_45u_G1.65dB_155ms	
	HSDPA_PA3_45u_G1.65dB_215ms	
	HSDPA_PA3_100u_G1.65dB_55ms	Low load
	HSDPA_PA3_100u_G1.65dB_95ms	
	HSDPA_PA3_100u_G1.65dB_100ms	
	HSDPA_PA3_100u_G1.65dB_155ms	
	HSDPA_PA3_100u_G1.65dB_215ms	
	HSDPA_PB3_45u_G0.09dB_55ms	Medium load
	HSDPA_PB3_45u_G0.09dB_95ms	
	HSDPA_PB3_45u_G0.09dB_100ms	
	HSDPA_PB3_45u_G0.09dB_155ms	
	HSDPA_PB3_45u_G0.09dB_215ms	
	HSDPA_PB3_100u_G0.09dB_95ms	High load
	HSDPA_PB3_100u_G0.09dB_100ms	
	HSDPA_PB3_100u_G0.09dB_155ms	
	HSDPA_PB3_100u_G0.09dB_215ms	

Detailed explanation about these profiles and how they are generated can be found in [6]. Obviously, having only two profiles for the uplink is far from sufficient to evaluate the JBM performance in a wide variety of conditions. Moreover, these two profiles exhibit quite similar properties as shown in the following figures.

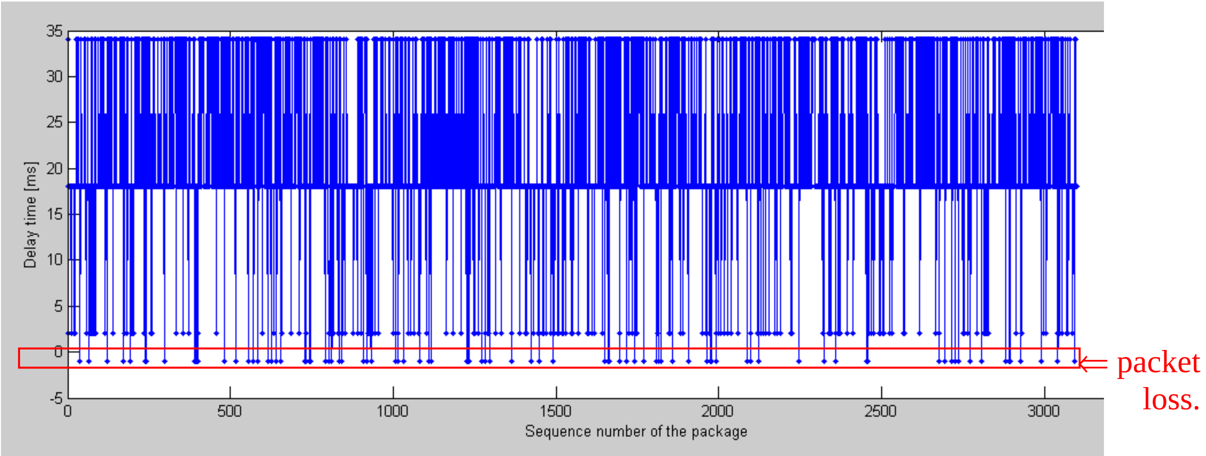

Figure 3.2: Channel delay of “HSUPA_PA3_45u”

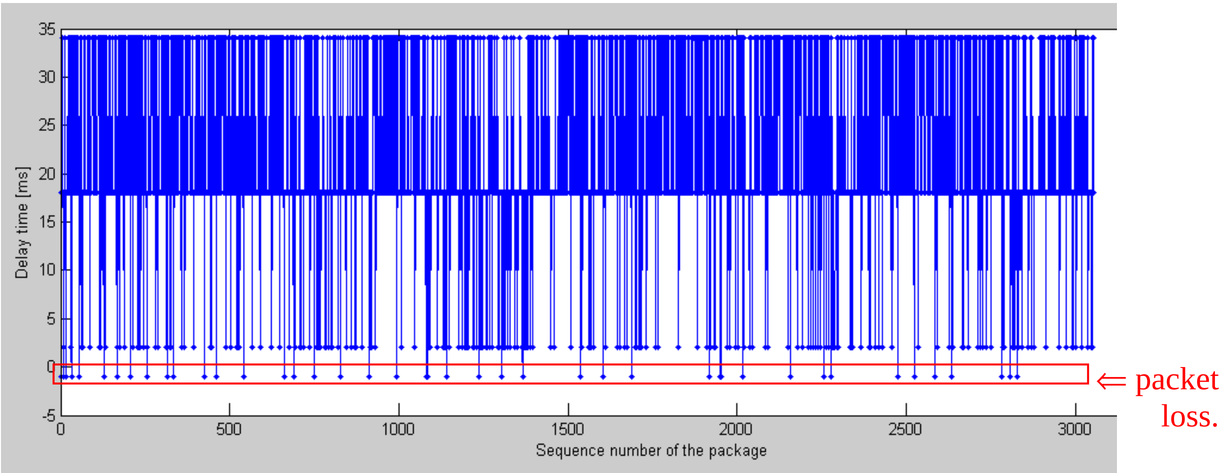

Figure 3.3: Channel delay of “HSUPA_PB3_45u”

As the figures show, a maximum jitter of only 34ms is not challenging enough to test the JBMs’ performance or to compare the adaptive and static JBMs.

Initially, the number of HSDPA profiles seemed to be sufficient. However, after studying these profiles, it is observed that it is unnecessary to simulate all of them because some of them show the same or rather similar characteristics. Therefore, a set of five of the downlink profiles were selected (shaded in dark blue in Table 3.1) and categorized into different loads:

Low-load	“HSDPA_PA3_100u_G1.65dB_55ms”
Medium-load	“HSDPA_PB3_45u_G0.09dB_55ms” and “HSDPA_PB3_45u_G0.09dB_155ms”
High-load:	“HSDPA_PB3_100u_G0.09dB_95ms” and “HSDPA_PB3_100u_G0.09dB_155ms”.

The characteristic of each delay and error profile are presented in Table 3.2.

Table 3.2: Characteristics of 3GPP delay and error profiles

File name	HSUPA_PA3_45u	HSUPA_PB3_45u	HSDPA_PA3_100u_G1.65dB_55ms	HSDPA_PB3_45u_G0.09dB_55ms	HSDPA_PB3_45u_G0.09dB_155ms	HSDPA_PB3_100u_G0.09dB_95ms	HSDPA_PB3_100u_G0.09dB_155ms
Number of entries	3098	3054	2899	2899	2898	2898	2898
Packet loss	99	47	0	0	0	69	0
PLR	3.20%	1.5%	0	0	0	2.38%	0
Mean delay (ms)	21.77	22.02	10.66	10.60	10.59	22.18	27.83
Max delay (ms)	34.1	34.1	16	54.67	64	91.33	126
Min delay (ms)	2	2	6	2	2	2	2

3.1.3.2 Synthetic Delay and Error Profiles

As discussed in the previous section, the UL delay and error profiles from the earlier 3GPP work were judged to be insufficient for this project. Thereby, additional channel profiles representing different loads are created using Matlab scripts in order to test how the implemented JBMs react to various network conditions and to assess the advantages of the adaptive JBM over the static JBM. As long as channel dependent scheduling is not used in the uplink, only HARQ is taken into account in order to decide upon the delay value of each packet. In HSUPA either 10ms TTI or 2ms TTI could be used. In this project it was deliberately decided to utilize only with 2ms TTI since a shorter TTI allows reduced delays. As discussed in section 2.5.1, the HARQ round trip time is 16ms with 2ms TTI. The new generated channel profiles are shown in Table 3.3. There are 3000 entries for each delay and error profile.

Table 3.3: Synthetic UL delay and error profiles

Drop timer	Load			
	Low load	Medium load	High load	Overload
75 ms (max spike 66ms)	0.2% up to Maximum spike 66ms	PLR=0.51% 9.93% up to maximum spike 66ms	Not simulated	PLR=4.5% 27.2% up to maximum spike 66ms
100 ms (max spike 98ms)	Not simulated	Not simulated	PLR=0.44% 31.23% spikes up to or beyond 66ms	Not simulated
200 ms (max spike 192ms)	0.27% up to Maximum spike 66ms	PLR=0 12.2% spikes up to or beyond 66ms	PLR=0.036% 23.67% spike up to or beyond 66ms	PLR=0.073% 32.8% spikes up to or beyond 66ms

For low load, jitter spikes mostly correspond to 1 to 2 retransmissions (18ms or 34ms). In the generated data there are examples with no spikes and rare spikes up to 66ms (corresponding to 4 retransmissions). In these profiles no packet experiences longer than 75ms delay.

For medium load, jitter spikes mostly correspond to 1 to 3 retransmissions (18ms, 24ms, or 50ms), there are quite frequent jitter spikes of up to 66ms. Approximately 0.5% of all packets experience delays longer than 75ms.

For high load, there are frequent jitter spikes of up to 66 ms (4 retransmissions), some spikes are even up to the drop timer value. About 2% of packets are expected to have delays longer than 75ms.

For the over load situation, there are frequent jitter spikes up to the drop timer value. In this setting approximately 5% of packets can be delayed longer than 75ms.

Among these synthetic HSUPA channel profiles, the ones colored in blue are likely to be more interesting as they represent extreme conditions, hence the are most likely to result in greater differences between the JBMs.

It is important to note that the UL channels are generated simply to test how the designed JBMs react to variation of network condition; as the behavior does **not** take the proposed delay budget explained in section 2.5.1 into account.

3.2 Evaluation

The performance of the designed JBMs is evaluated both objectively and subjectively.

3.2.1 Objective Evaluation

The objective evaluation will be accomplished by logging and analyzing the necessary information from the simulation chain including:

Decoding time	This verifies that packets are delivered to decoder every 20ms.
Jitter loss	Since CSoHS has two JBMs, The jitter loss rate should be kept below 0.5% for UL and DL separately to keep the overall jitter loss rate under 1% (other distributions are also possible, e.g. 0.6% for UL and 0.4% for DL).
End-to-end delay	The end-to-end delay is used to observe how the semi-static JBM adapts its buffer depth.

Moreover, a comparison is made between the adaptive and static JBMs to understand the difference between the adaptive one over the static one.

3.2.2 Subjective Evaluation

A subjective evaluation was base on a simple listening test. Every generated sound file is listened to and the quality is informally judged. In particular, the speech files generated from the same delay and error profile, but by different types of JBMs were compared to see how the JBMs' performance impacted the speech quality.

A major difference between the objective and subjective evaluation methods is that the objective evaluation only considers the losses introduced by the JBM. When listening to the files, the voice quality depends on the sum of all kinds of losses (channel loss, jitter loss, buffer under-run, etc.)

- The devices used for listening were a Sennheiser HD 545 reference headset connected to the computer via a Roland Corp. EDIROL USB Audio Capture UA-25 (this is a 24 bit 96kHz audio interface).

4. Analysis

This chapter presents and discusses of the all simulation results with both static and adaptive JBMs. As the number of uplink channel profiles is considered to be adequate and covers a variety of circumstances, the results of downlink simulation are present as well (although they imply the same conclusion). A comparison is made between static and adaptive JBMs with the focus on jitter loss control and buffering delay. Finally some comments are made based upon the results of the subjective listening test.

4.1 Analysis of the Static JBM

As described above, the performance of the static JBM is judged in terms of jitter loss and end-to-end delay.

4.1.1 UL Delay and Error Profiles from 3GPP

Initially, two delay and error profiles from 3GPP were tested. The results with these profiles are shown in Table 4.1.

Table 4.1: Results of 3GPP E-DCH profiles with static JBM

Channel condition	HSUPA_PA3_45u	HSUPA_PB3_45u
Initial JBM level	2	2
Transmitted packets	2758	2758
Received packets	2671	2713
Received speech frames	2638	2681
Lost packets	87	45
Packet loss rate	3.15%	1.63%
End-to-end delay of fixed JBM [ms]	58.13	98.13
Jitter loss rate of fixed JBM	0.22%	0%

The initial JBM buffer level is set based upon the maximum (expected) delay value. (As noted previously this was known to be 34ms for these profiles, hence using this value means that the jitter loss should be very low.) It might be noted that the overall delay for the channel when using the profile “HSUPA_PB3_45u” was 98.13ms. This seems to be too long. It has this value because the first two packets transmitted are consecutively lost during transmission. However, the JBM does not start initialization until it receives an initial packet; unfortunately this is the third transmitted packet. If one eliminates these first two packets from the analysis, then the end-to-end delay would be only 58.13ms.

4.1.2 Synthetic UL Delay and Error Profiles

As explained in section 3.1.3, additional delay and error profiles were generated in order to test the JBM’s performance under various network conditions. Those channel profiles shaded in blue in Table 3.3 are judged to be more interesting because they are extreme conditions. The test results with the static JBM are shown in Table 4.2. The dynamic channel model was created by concatenating together several delay and error profiles from different loads. The initial buffer level was set according to the drop timer. The jitter buffer starts to extract packets once the collected number of packets reaches this initial level.

Table 4.2: Test results of synthetic E-DCH profiles with static JBM (part 1)

Channel condition	Low load		Over load			Dynamic	
Drop timer [ms]	75	200	75	200		200	
Initial JBM level	4	10	4	4	10	4	10
Label	L75	L200	O75_4	O200_4	O200_10	D4	D10
Transmitted packets	2758	2758	2758	2758		13777	
Received packets	2758	2758	2633	2756		13774	
Received speech frames	2723	2723	2600	2721		13589	
Lost packets	0	0	125	2		3	
Packet loss rate	0	0	4.5%	0.073%		0.021%	
End-to-end delay of fixed JBM [ms]	100.13	218.13	130.13	146.13	250.13	98.13	234.13
Jitter loss rate of fixed JBM	0	0	0	0.81%	0	0.7%	0

As these results show, the end-to-end delay is highly dependent on the initial buffer level. A larger initial buffer level results in longer delay because the static JBM does not adapt to the network conditions. For example with an initial JBM buffer level of 4 this corresponds to a delay of 80ms (=4*20ms) and an initial JBM buffer level of 10 corresponds to a delay of 200ms (10*20ms). Thus we see that in the case of a low load that the additional end-to-end delay was less than 20 ms (i.e., less than one 20 ms audio frame) longer than the delay due to the initial JBM buffering.

The jitter loss appears to be nicely controlled - if the initial buffer level is set according to the drop timer. This is because that the drop timer defines the maximum transmission delay, thus the jitter buffer is always capable of handling all of the spikes – since they are limited by the drop timer to be below this bound. However, in two special cases, “overload, drop timer=200ms” and “dynamic”, where the largest spike is up to 194ms, very high jitter loss rates occur when the initial buffer level is set to 4 packets. This can be easily explained because such a small initial buffer level is not able to catch up with these larger spikes. Thus a clear result of this testing is that the initial JBM buffer level must be greater than or equal to the drop timer.

Although the delay and error profiles of extreme cases should be sufficient to verify the JBM’s performance, the other synthetic profiles were also simulated and the results are shown in

Table 4.3.

Table 4.3: Results of synthetic E-DCH profiles with static JBM (part 2)

Channel condition	Medium load		High load	
Drop timer [ms]	75	200	100	200
Initial JBM level	4	10	5	10
Transmitted packets	2758	2758	2758	2758
Received packets	2744	2758	2746	2757
Received speech frames	2709	2723	2711	2722
Lost packets	14	0	12	1
Packet loss rate	0.51%	0	0.44%	0.036%
End-to-end delay of fixed JBM [ms]	82.13	234.13	150.13	266.13
Jitter loss rate of fixed JBM	0.036%	0	0	0

The results are similar to the previous ones. As before, the end-to-end delay depends on how large the initial buffer level is set. The jitter loss is well controlled since the initial buffer level is set based upon the drop timer.

4.1.3 DL Delay and Error Profiles from 3GPP

The simulation results using the 3GPP DL delay and error profiles are shown in **Error! Reference source not found.**. The results are again similar to the previous results. The initial buffer level was decided according to the drop timer in all cases, so that the jitter loss is very well controlled.

Table 4.4: Results of 3GPP DL delay and error profiles

Channel condition	Low load	Medium load		High load	
Drop timer [ms]	55	55	155	95	155
File name	HSDPA_PA3_100u_G1.65dB_55ms	HSDPA_PB3_45u_G0.09dB_55ms	HSDPA_PB3_45u_G0.09dB_155ms	HSDPA_PB3_100u_G0.09dB_95ms	HSDPA_PB3_100u_G0.09dB_155ms
Initial JBM level	3	3	8	5	8
Transmitted packets	2758	2758	2758	2758	2758
Received packets	2758	2758	2758	2697	2758
Received speech frames	2723	2723	2723	2662	2723
Lost packets	0	0	0	61	0
Packet loss rate	0	0	0	2.2%	0
End-to-end delay of fixed JBM [ms]	72.13	62.13	174.13	132.81	184.13
Jitter loss rate of fixed JBM	0	0.07%	0	0	0

4.2 Results of Adaptive JBM

The simulations with the adaptive JBM were done using the identical delay and error profiles as with the static JBM.

4.2.1 UL Delay and Error Profiles from 3GPP

Table 4.5 shows the results of using the 3GPP UL delay and error profiles. As can be observed the jitter loss rates remain below 0.5% as required and the end-to-end delay is around 50ms with both delay and error profiles. Hence the results with both profiles are acceptable.

Table 4.5: Results of 3GPP UL profiles with adaptive JBM

Channel condition	HSUPA_PA3_45u	HSUPA_PB3_45u
Initial JBM level	2	2
Transmitted packets	2758	2758
Received packets	2671	2713
Received speech frames	2638	2681
Lost packets	87	45
Packet loss rate	3.15%	1.63%
Late speech frames	0	7
Jitter loss rate	0	0.26%
Average buffering time [ms]	27.48	31.81
Average end-to-end delay [ms]	49.25	53.94

The following figures show how the end-to-end delay varies during the communication session. These figures graphically show the JBM's adaptation.

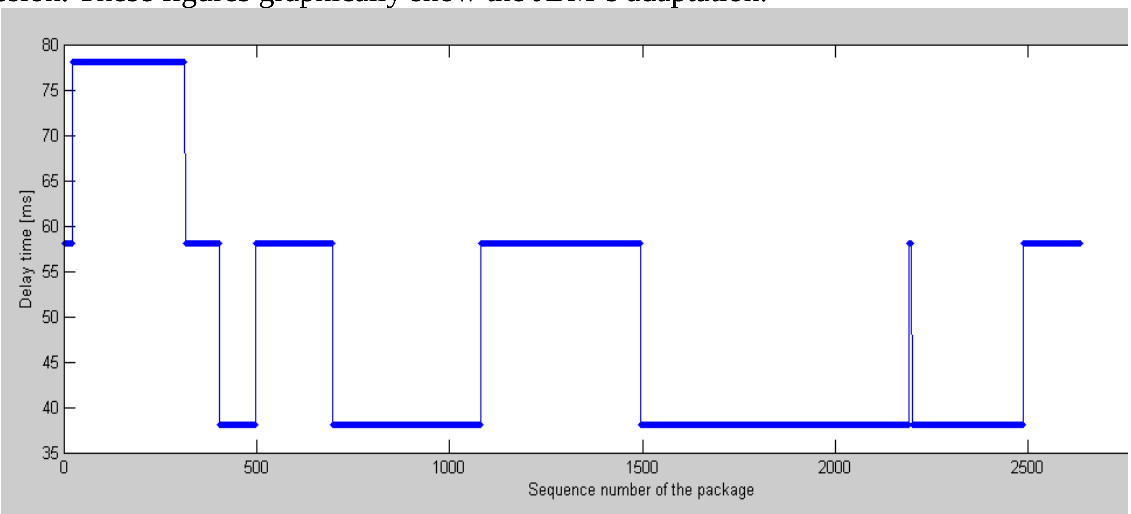

Figure 4.1: End-to-end delay of "HSUPA_PA3_45u" with adaptive JBM

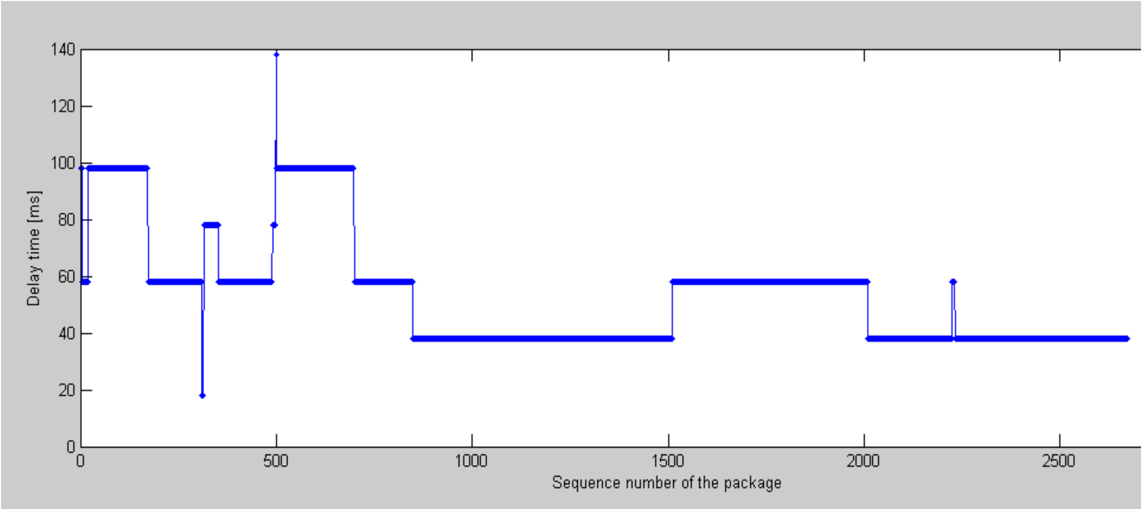

Figure 4.2: End-to-end delay of “HSUPA_PB3_45u” with adaptive JBM

4.2.2 Synthetic UL Delay and Error Profiles

Table 4.6 shows the performance of the adaptive JBM for the same profiles as were shown in Table 4.2 for the static JBM. Generally, the results are reasonable and show that the implemented adaptive JBM performs properly under different network conditions because:

- The jitter loss rates are always maintained below 0.5% for each channel.
- No matter what initial buffer level is set, the adaptive JBM ends up with close values of average buffering time and average end-to-end delay for the same channel profile. This indicates that the adaptation works properly because JBM manages to adapt to the network conditions as they vary over time.

Table 4.6: Results of synthetic UL profiles with adaptive JBM (part 1)

Channel condition	Low load		Over load			Dynamic	
Drop timer [ms]	75	200	75	200		200	
Initial JBM level	4	10	4	4	10	4	10
Label	L75	L200	O75_4	O200_4	O200_10	D4	D10
Transmitted packets	2758	2758	2758	2758		13777	
Received packets	2758	2758	2633	2756		13774	
Received speech frames	2723	2723	2600	2721		13589	
Lost packets	0	0	125	2		3	
Packet loss rate	0	0	4.5%	0.073%		0.021%	
Late speech frames	2	5	0	7	11	28	30
Jitter loss rate	0.073%	0.18%	0.038%	0.26%	0.40%	0.21%	0.22%
Average buffering time [ms]	73.22	69.09	86.82	144.83	135.07	94.33	96.68
Average end-to-end delay [ms]	94.80	90.18	135.22	197.37	187.44	130.94	133.28

The following figures in the following sections show the delay and jitter of each channel and how the jitter buffer adapts.

4.2.2.1 Low-load, drop timer=75ms

Figure 4.3 shows the channel delay. In the case of a low load channel, there is no packet loss and only very rare jitter spikes of up to 66ms.

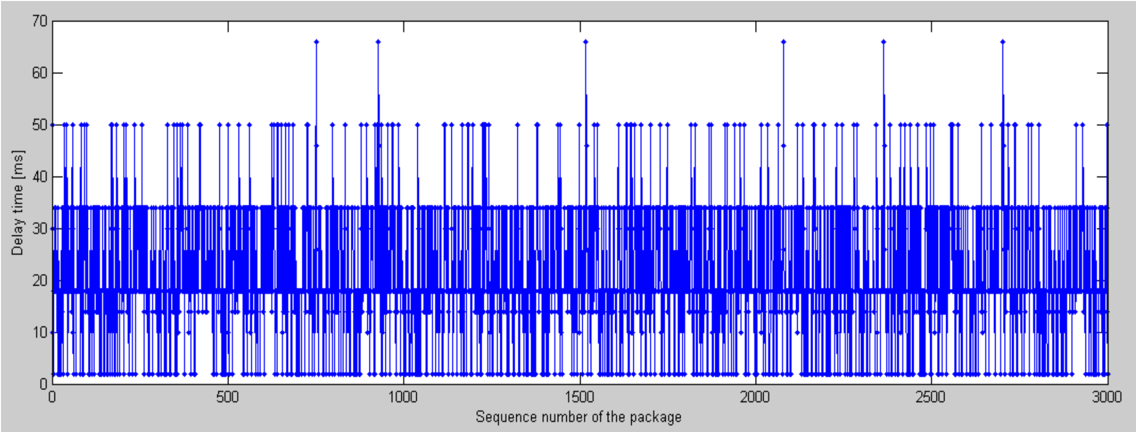

Figure 4.3: Channel delay of “Low-load, drop timer=75ms”

Figure 4.4 shows how the end-to-end delay changes during the whole communication, which indicates the JBM adaptation.

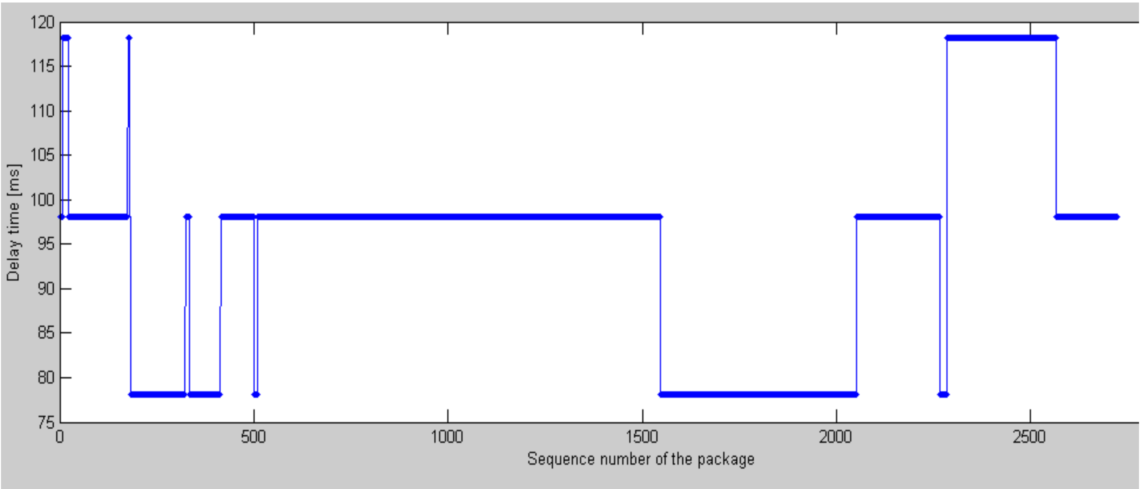

Figure 4.4: End-to-end delay for “Low-load, drop timer=75ms”

4.2.2.2 Low load, drop timer 200ms

Figure 4.5 show the channel delay. For a low load channel, the packet loss rate (PLR) at 75ms is zero as discussed in section 3.1.3. Thus there is no spike larger than 66ms, although the drop timer is 200ms.

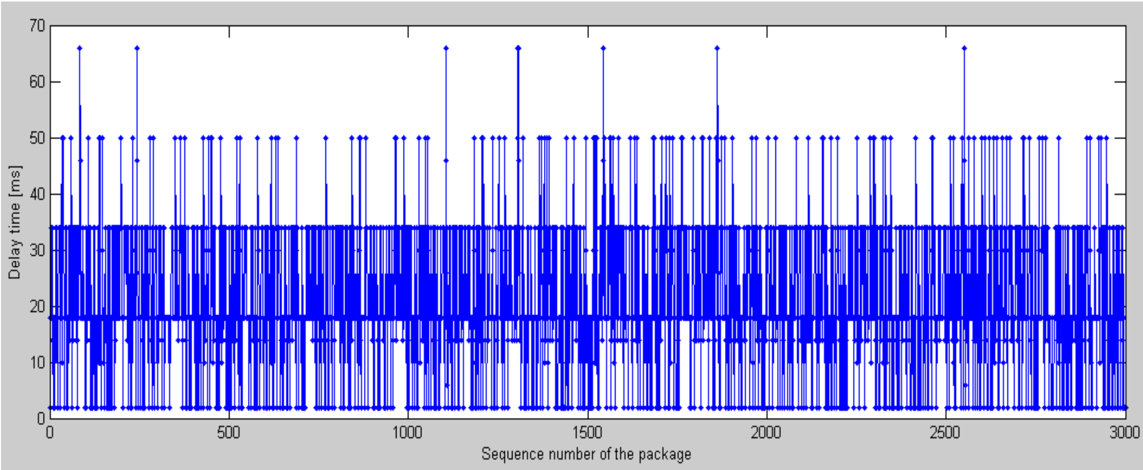

Figure 4.5: Channel delay of "Low-load, drop timer-200ms"

Figure 4.6 shows the change in the end-to-end delay for this case. Obviously, the initial buffer level of 10 packets is much more buffering than needed, so the JBM adapts down quickly and maintains a lower level.

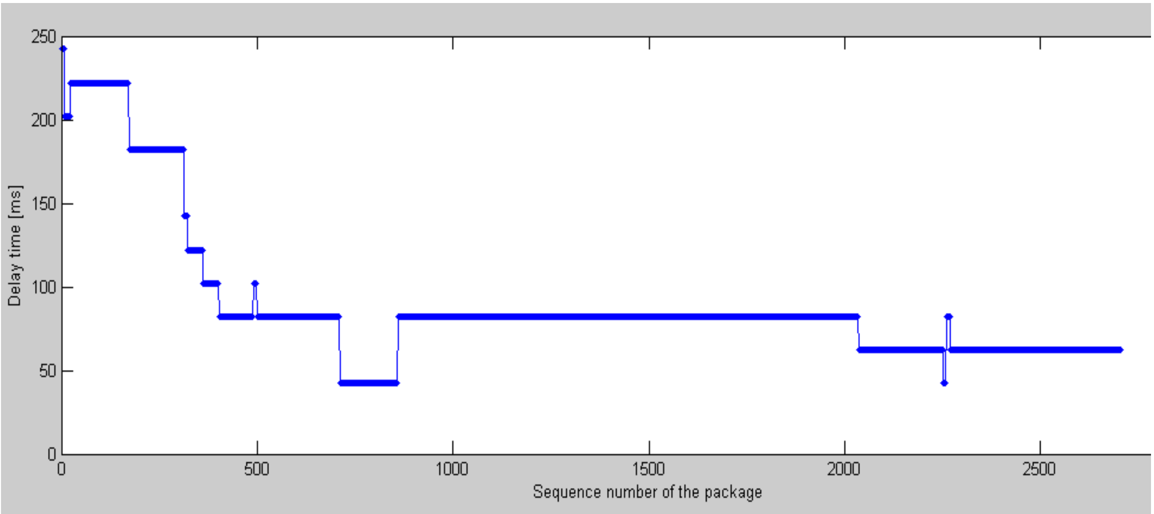

Figure 4.6: End-to-end delay for "Low-load, drop timer=200ms"

4.2.2.3 Over-load, drop timer 75ms

Figure 4.7 shows the channel delay for the case of over-load with a drop timer of 75ms. A value of -1 indicates a packet loss. For this channel, most spikes reach 66ms and there are quite a lot of losses (PLR=4.5%) due to the small value of the drop timer. Note that these packets are dropped by the RLC and not by the receiving side.

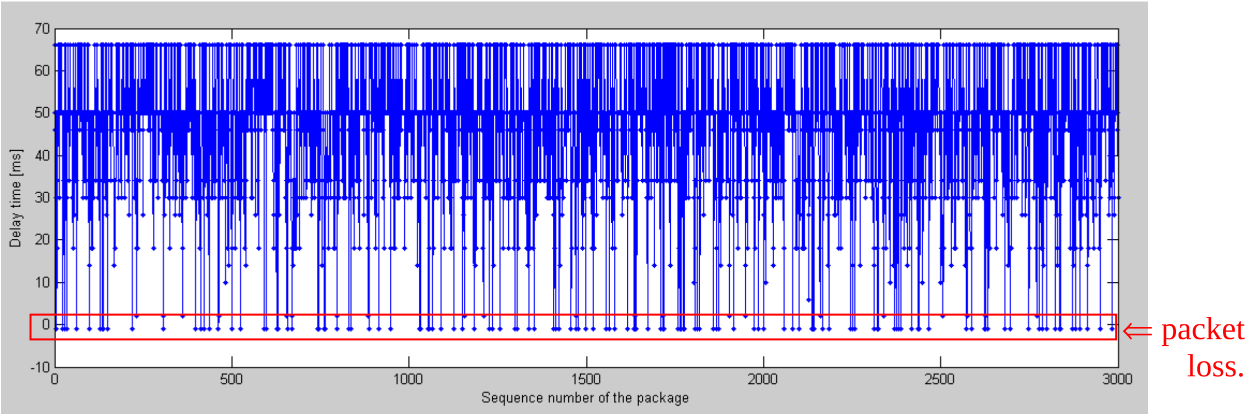

Figure 4.7: Channel delay of "Over-load, drop timer=75ms"

Figure 4.8 shows how the end-to-end delay varies.

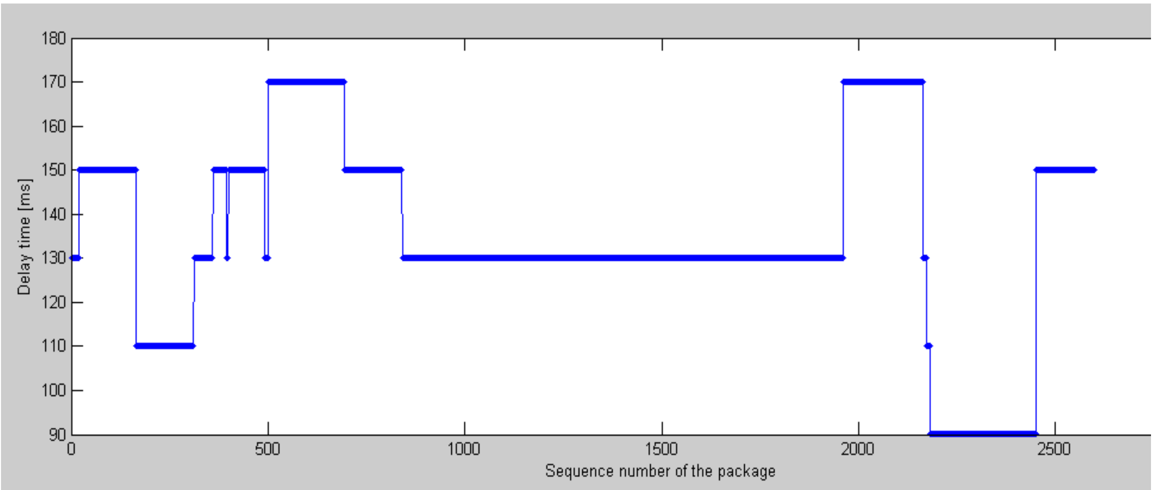

Figure 4.8: End-to-end delay for "Over-load, drop timer=75ms"

4.2.2.4 Over-load, drop timer 200ms

Figure 4.9 shows the channel delay. As the figure shows, the channel shows sharp variation with a number of spikes larger than 100ms or even up to the drop timer.

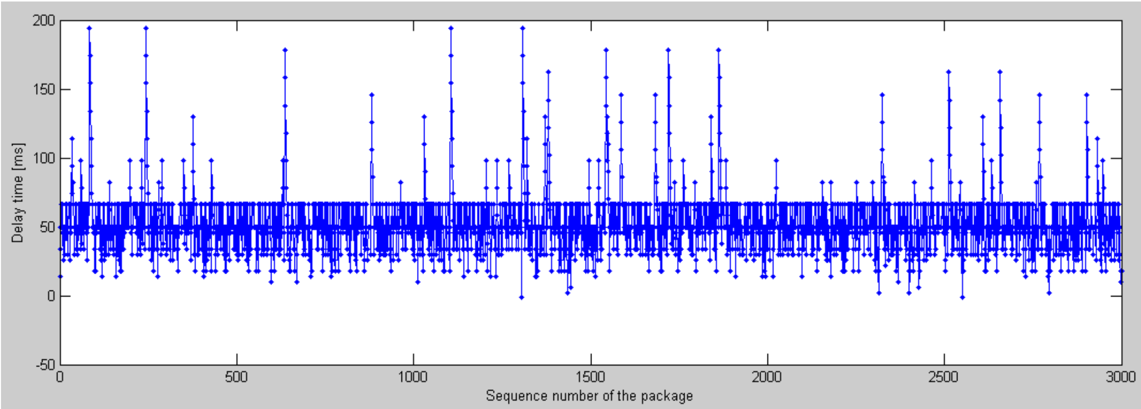

Figure 4.9: Channel delay of “Over-load, drop timer=200ms”

Figure 4.10 shows the change of end-to-end delay with initial JBM level being set to 4 packets. An initial buffer level of 4 packets is not capable of handling such larger spikes in the over loaded channel. So the JBM increases its buffer depth in order to avoid too much jitter loss.

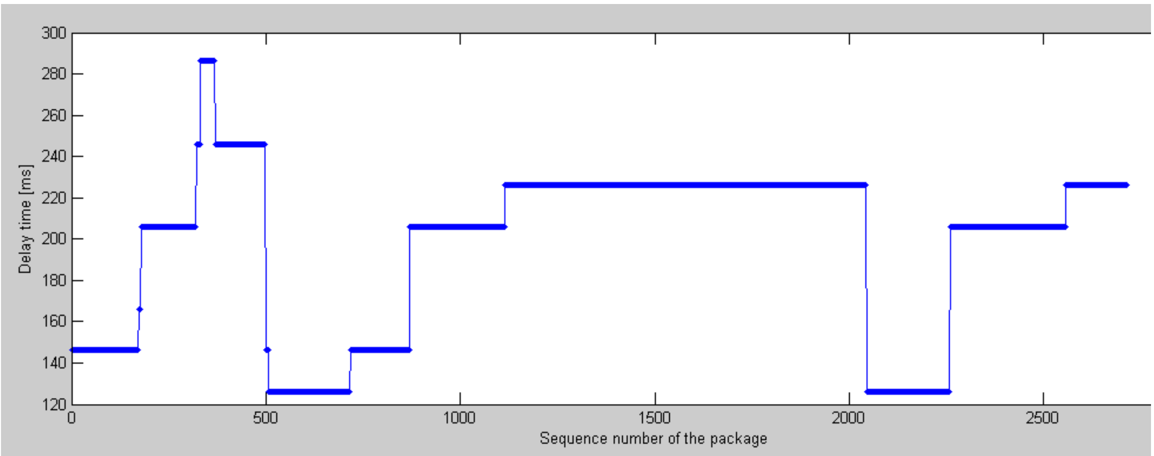

Figure 4.10: End-to-end delay with initial buffer level=4

Figure 4.11 shows the change of end-to-end delay when initial buffer level is set to 10 packets. It can be seen that the JBM is able to adapt downwards sometimes when the channel is less aggressive to reduce unnecessary latency.

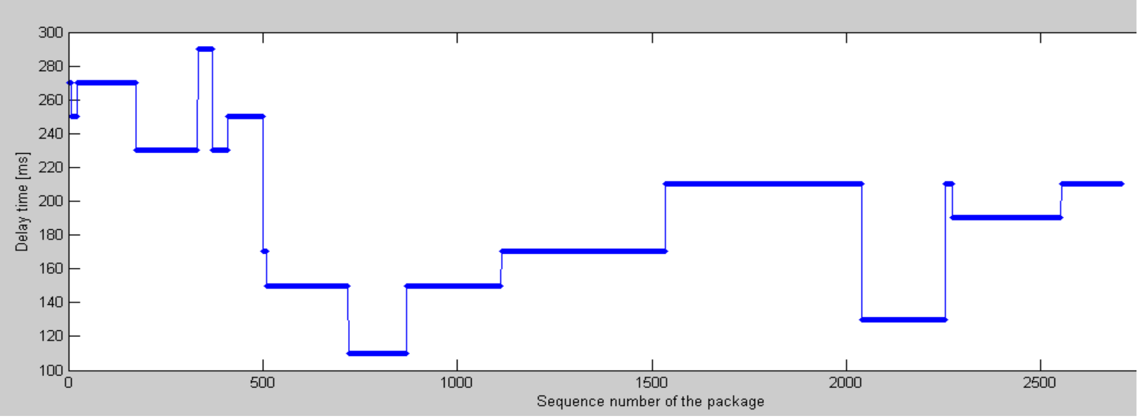

Figure 4.11: End-to-end delay with initial buffer delay=10

4.2.2.5 Dynamic channel, drop timer 200ms

Figure 4.12 shows the channel delay. Concatenated by several delay and error profiles from different load, the dynamic channel shows various features in different time periods.

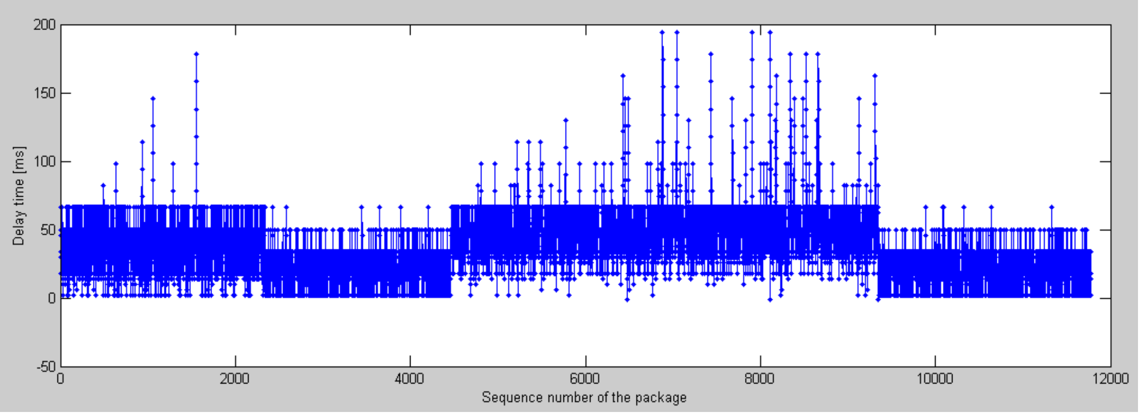

Figure 4.12: Channel delay of "Dynamic, drop timer=200ms"

Figure 4.13 shows how the end-to-end delay varies when the initial buffer level is set to 4 packets.

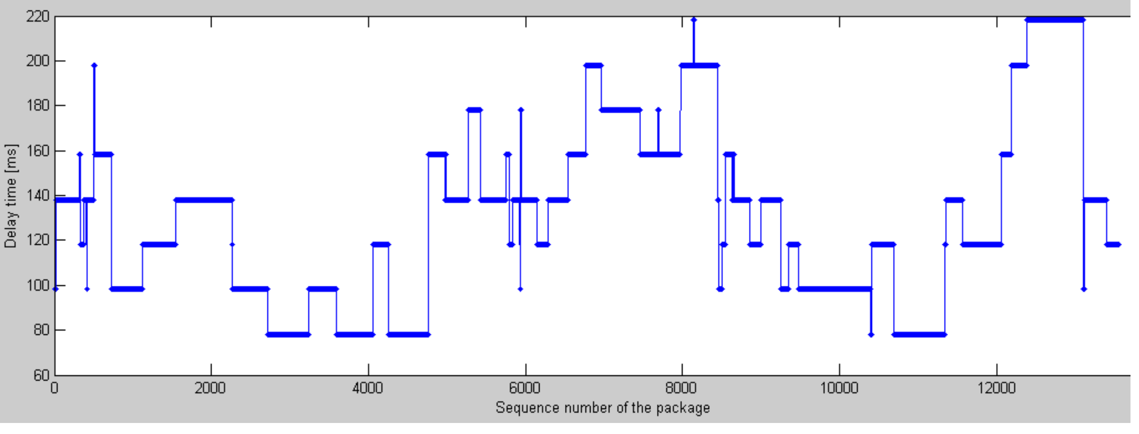

Figure 4.13: End-to-end delay with initial buffer level =4

Figure 4.14 shows how end-to-end delay varies when the initial buffer level is set to 10 packets. As the dynamic channel begins with the largest spikes only up to 66ms, the JBM decreases its buffer level from 10 packets rapidly at the beginning.

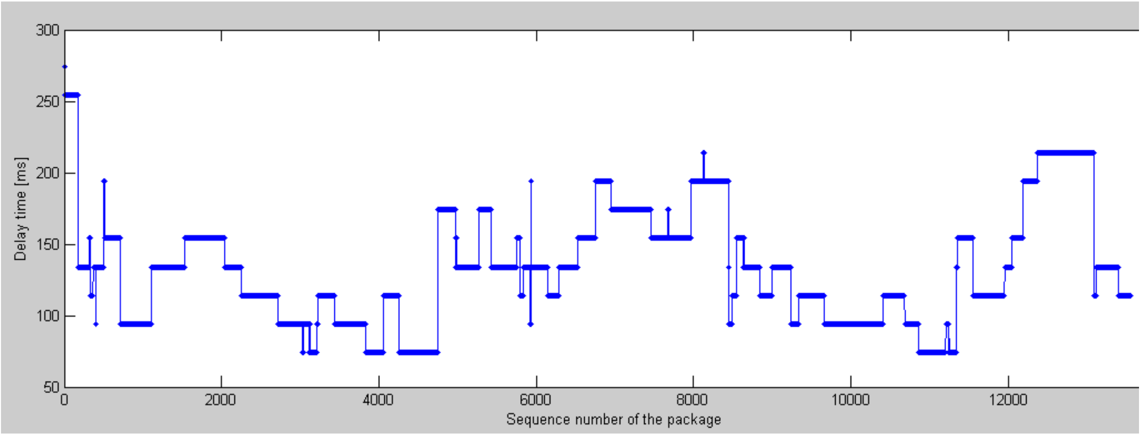

Figure 4.14: End-to-end delay with initial buffer level=10

Although the previous simulation results could be considered adequate to verify the adaptive JBM's performance, the rest of the synthetic delay and error profiles are simulated as well for completeness. These results are shown in Table 4.7.

Table 4.7: Results of synthetic UL profiles with adaptive JBM (part 2)

Channel condition	Medium load		High load	
Drop timer [ms]	75	200	100	200
Initial JBM level	4	10	5	10
Transmitted packets	2758	2758	2758	2758
Received packets	2744	2758	2746	2757
Received speech frames	2709	2723	2711	2722
Lost packets	14	0	12	1
Packet loss rate	0.51%	0	0.44%	0.036%
Late speech frames	0	9	0	7
Jitter loss rate	0	0.33%	0	0.26%
Average buffering time [ms]	57.91	87.34	67.57	91.12
Average end-to-end delay [ms]	94.48	124.94	117.89	138.41

The figures showing the channel delay and how the end-to-end delay varies can be found in Appendix C.

4.2.3 DL Delay and Error profiles from 3GPP

As channel dependent scheduling is used in the downlink, the delay values can be much more random than in the uplink (figures for DL delay and buffer adaptation are shown in Appendix C.). The simulation results are collected in Table 4.8.

The results are satisfactory since jitter loss is limited to under 0.5% for each channel and the end-to-end delay is also acceptable.

Table 4.8: Results of 3GPP DL profiles with adaptive JBM

Channel condition	Low load	Medium load		High load	
Drop timer [ms]	55	55	155	95	155
Initial JBM level	3	3	8	5	8
Transmitted packets	2758	2758	2758	2758	2758
Received packets	2758	2758	2758	2697	2758
Received speech frames	2723	2723	2723	2662	2723
Lost packets	0	0	0	61	0
Packet loss rate	0	0	0	2.2%	0
Late speech frames	0	7	0	4	4
Jitter loss rate	0.037%	0.26%	0.15%	0.15%	0. 22%
Average buffering time [ms]	28.72	61.91	70.38	70.03	94.77
Average end-to-end delay [ms]	34.68	71.14	80.99	90.85	118.55

4.3 Comparison between Static and Adaptive JBMs

The jitter loss and the buffering time (or our approximation to end-to-end delay) are essential metrics to judge each JBM's performance. Thus a comparison is made between the adaptive and the fixed JBMs based on these two performance metrics. For this comparison, only the results from Table 4.2 are used as they represent extreme cases and they stress the JBMs.

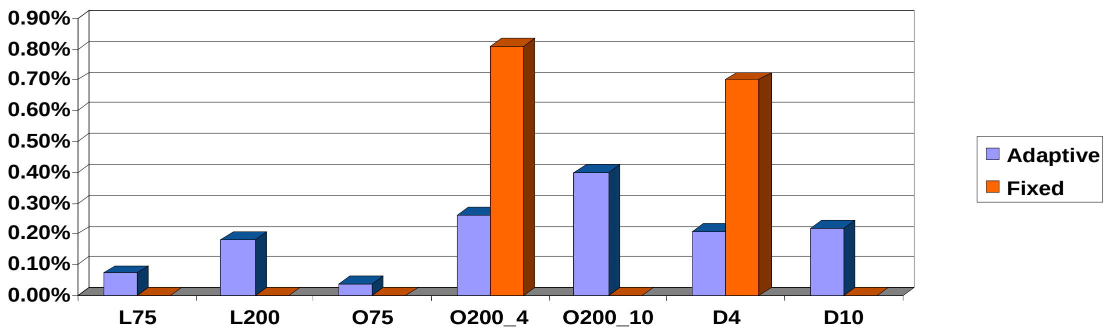

Figure 4.15: Jitter loss rates of adaptive and fixed JBMs

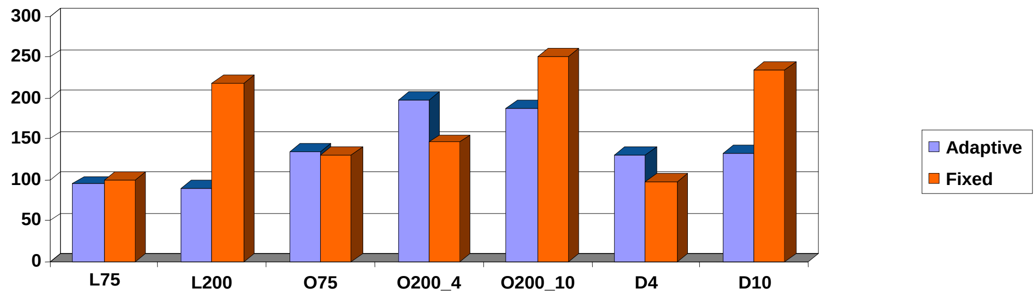

Figure 4.16: End-to-end delay (ms) of adaptive and fixed JBMs

As Figure 4.15 shows, the adaptive JBM shows more stable performance in jitter loss control under all kinds of network conditions. It is stated in [9] that limiting the jitter loss takes priority over minimizing delay. For most channel conditions, the jitter loss of the fixed JBM is zero because the initial buffer level was set according to drop timer which was enough to catch the largest spike. However, for channels “over load with 200ms drop timer” and “dynamic” where large spikes (up to 194ms) occur and when the initial buffer level is set to 4 packets, then the fixed JBM shows much worse performance with respect to jitter loss since it does not adapt to varying channel condition, while the adaptive JBM is able to keep jitter loss low by compensating with a longer delay. However, this test shows how adaptive JBM can outperform the fixed JBM with respect to these variations in channel conditions. However, in real system the initial level of fixed JBM would be decided by the drop timer to prevent such high jitter loss. Thus in practice there would be no significant different with respect to packet loss for these two JBMs.

Based on the performance shown in Figure 4.16, it can be concluded that the adaptive JBM is able to avoid unnecessary delay when the initial buffer level is set too high for the channel. In contrast, the delay resulting from the fixed JBM is determined soley by the initial jitter buffer level. As the figure shows, whenever the initial buffer level is set to 10, the adaptive JBM can adapt downwards by allowing some jitter loss while the fixed JBM simply maintains a long end-to-end delay.

4.3.1 Discussion

Based on the comparison between the adaptive and static JBMs, the advantages of the adaptive JBM over the static JBM can be stated as follows:

- The adaptive JBM’s target level for each talk-spurt does not always have to be the same as the drop time, but can adapt downwards under better channel conditions. This enables the use of a longer drop timer for all UEs, so that the UEs with poor radio conditions will still experience lower BLER because more retransmissions will be allowed for these UEs. On the other hand, those UEs that have good channel conditions will experience shorter delays when using an adaptive JBM.
- The adaptive JBM provides consistent performance while balancing the trade-off between jitter losses and buffering delay under different kinds of channel conditions. Usually the adaptive JBM tries to minimize latency. However, as noted in [9] it is more important to limit jitter loss. Fortunately, adaptive JBM is able to

increase its target level once the jitter loss becomes too high. Therefore, the jitter loss rate can be maintained under some limit by compromising with added delay.

4.4 End-to-End Aspects of CSoHS

As the UL and DL have been discussed separately, it is interesting and necessary to look at the whole behavior of CSoHS. However, the focus remains on jitter loss and the end-to-end delay.

4.4.1 Overall Jitter Loss

Table 4.9 summarizes all the jitter loss rates of adaptive JBM for the tested channels in ascending order (by loss rate). It is quite clear from this table that the jitter loss rate is below 0.5% for all kinds of UL or DL channels separately. Even in the worst cases, the jitter loss rates for UL and DL are 0.40% and 0.26% separately. So the overall worst jitter loss rate is: *0.40% UL + 0.26% DL = 0.66% (< 1%)*, which meets the performance requirement.

On the other hand, the jitter loss of the fixed JBM can be always guaranteed if the initial level is set properly according to the drop timer.

Table 4.9: Jitter loss rates with adaptive JBM

UL	0.038%	0.073	0.18%	0.21%	0.22%	0.26%	0.33%	0.40%
DL	0.037%	0.15%	0.22%	0.26%				

4.4.2 End-to-End Delay

As mentioned previously, the synthetic delay and error profiles were designed to stress the JBM and to see how the designed JBMs responded to changes in network conditions. These delay and error profiles were created using quite simple methods and are not necessarily representative of the channel conditions that might be experience in reality.

However, the channel profiles from 3GPP were generated in a more advanced way and are believed to be much closer to the conditions that will occur in real systems. Thus only those results based upon the 3GPP delay and error profiles are discussed further in this section. With two UL and five DL channels, there can be 10 possible combinations as shown in Table 4.10. (with plus 40ms UL coding + 30ms RAN/CN processing + 10ms online transmission + 40ms DL coding).

Table 4.10: Overall delay of CSoHS with adaptive JBM

UL (ms)	DL (ms)				
	34.68	**71.14**	**80.99**	**90.85**	**118.55**
49.25	203.93	240.39	250.24	260.10	287.80
53.94	208.62	245.08	254.93	264.79	292.49

There are two delay values (287.80ms and 292.49ms) which are slightly longer than the delay budget of 276ms - discussed in Section 2.5.1, but it is though that these values should still be acceptable. For all the other cases, the overall delay is within the delay budget.

For the fixed JBM, the overall CSoHS delay results are shown in Table 4.10.

Table 4.11: Overall delay of CSoHS with fixed JBM

UL (ms)	DL (ms)				
	72.13	**62.13**	**174.13**	**132.81**	**184.13**
58.13	250.26	240.26	352.26	310.94	362.26
98.13	290.26	280.26	392.26	350.94	402.26

The results seem to be a little unsatisfactory since the overall delay exceeds the expected delay budget of 276ms in most cases. This is because the initial buffer level of the fixed JBM is set by the drop timer in order to limit the jitter loss. Therefore I believe that an adaptive JBM is more applicable in a real implementation.

4.5 Subjective Evaluation

For subjective evaluation, only informal listening tests were used. All speech files generated from simulations were listened to in order to find out how the JBM's performance impacts the voice quality, especially for comparisons between adaptive JBM and static JBM. The results indicate that the adaptive JBM is able to provide consistent voice quality under different network conditions, while the fixed JBM's performance depends on the network condition and the initial buffer level.

Furthermore, the adaptive JBM provides equal or better voice quality than the fixed JBM for the same channel profile. If the channel delay showed small variations, then the adaptive JBM provided performance equal to the fixed JBM; while in those extreme cases where the channel introduced larger delay variations, then the adaptive JBM outperformed the fixed JBM.

5. Conclusions

The thesis concludes with some conclusions and proposals for future work.

5.1 Conclusion

In this thesis project, two types of jitter buffer for jitter management in CSoHS were implemented and evaluated; a fixed (static) JBM which does not react to changes in network conditions and an adaptive (semi-static) JBM which adapts its buffer depth at the beginning of talk-spurts. This adaptation is achieved by looking at the buffering times of the most recent frames and calculating the largest difference. The performance of these two approaches was evaluated using both objective and subjective means. The objective evaluation focused on jitter loss control and buffering time; while the subjective evaluation was a series of informal listening tests.

A well designed jitter buffer management approach should be able to compromise properly between jitter loss and delay as needed. While for streaming media, the one-way delay is considered not very important, as rather long (~400ms) delay does not significantly degrade the quality of service. This means a longer delay can be tolerable in order to achieve low jitter loss. However, for real-time conversational voice service, both jitter loss and delay can degrade the quality of service, which makes the JBM design more difficult. The simulation results showed that the adaptive JBM can achieve satisfactory and more consistent performance while balancing the trade-off between jitter losses and buffering delay when dealing with various network conditions. Table 5.1 shows a comparison between static and adaptive JBM in terms of jitter loss and overall delay. The results in each row come from the same channel profile, with the same initial buffer level.

Table 5.1: Comparison between static and adaptive JBMs

Overall delay (ms)		Jitter loss rate	
Static JBM	Adaptive JBM	Static JBM	Adaptive JBM
58.13	49.25	0.22%	0
98.13	53.94	0	0.26%
100.13	94.80	0	0.073%
218.13	90.18	0	0.18%
130.13	135.22	0	0.038%
146.13	197.37	0.81%	0.26%
250.13	187.44	0	0.40%
98.13	130.94	0.7%	0.21%
234.13	133.28	0	0.22%
82.13	94.48	0.036%	0
234.13	124.94	0	0.33%

150.13	117.89	0	0%
266.13	138.41	0	0.36%
72.13	34.68	0	0.037%
62.13	71.14	0.07%	0.26%
174.13	80.99	0	0.15%
132.81	90.85	0	0.25%
184.13	118.55	0	0.22%

The comparison shows that normally the adaptive JBM performs better or equally well as the fixed JBM. However, there are a couple of exceptions (shaded in green) where the fixed JBM achieves slightly shorter delay as well as lower jitter loss rate.

Moreover, the adaptive JBM outperformed the fixed JBM in some extreme cases:

- When the initial buffer level was set too high for the channel, the adaptive JBM is able to decrease the buffer depth, thereby reducing the end-to-end delay. It does this by shortening the length of silence period, which in turn reduces the unnecessary end-to-end delay. For instance in case of low-load, 200ms drop timer, initial buffer level 10, the difference in delay is shown in Figure 5.1.

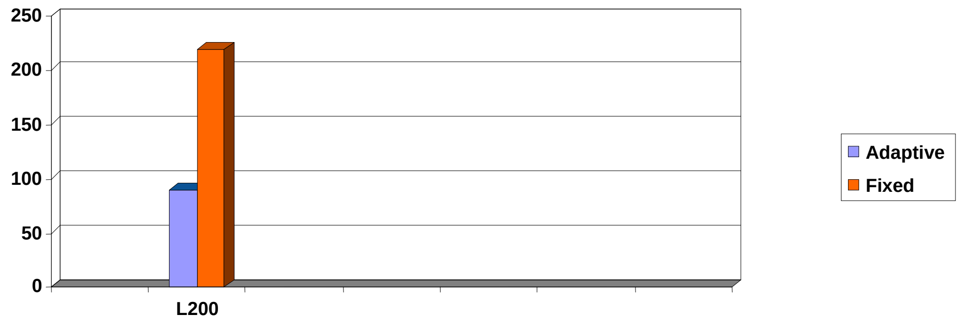

Figure 5.1: Delay comparison

- When the initial buffer level is too low for the channel, the adaptive JBM approach rapidly increases the buffer depth by extending the length of the silence period in order to limit jitter loss. For instance in the case of over-load, 200ms drop timer, initial buffer level 4, the comparison is shown in Figure 5.2.

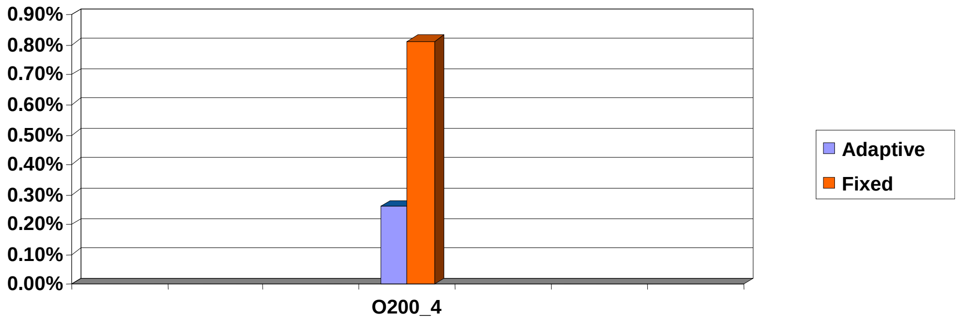

Figure 5.2: Jitter loss comparison

However, the jitter loss could also be limited when using fixed JBM if the initial buffer level is properly set according to the drop timer value.

Unfortunately, the fixed JBM's performance is highly dependent on the initial buffer level and it is not able to balance the trade-off between the jitter loss and buffering delay. Thus if the initial jitter buffer depth is much larger than the existing jitter or vice versa, then the adaptive JBM manages to adapt correctly to the current jitter characteristics. Under the same conditions, the fixed JBM either leads to excessive delay or excessive jitter losses. Therefore these simulations results show that it is potentially more suitable to apply adaptive JBMs; although it is suggested by 3GPP that fixed JBMs can be used on the uplink of CSoHS.

5.2 Future work

First of all, time scaling would be very interesting to implement as it can be optionally used in downlink. This would enable adaptation during a talk-spurt, which is a complement to adaptation only during silence period. This can help further improve the performance of the adaptive jitter buffer.

Moreover, in this thesis work the adaptation algorithm used statistics based on the buffering time of the most recent 200 packets. However, a different history length and other distributions of jitter loss rate (i.e, not dividing the UL and DL loss rate evenly) should also be studied. Much work has been done on adaptation algorithms for jitter buffers in VoIP, for example seven algorithms were introduced in [14]. It might also be interesting to apply these different algorithms and test their performance in HSPA networks.

Additionally, additional delay and error profiles that better conform to the proposed delay budget could be expected to provide a more accurate conclusion about how the JBM's performance will impact the end-to-end delay of CSoHS.

References

[1] Wikipedia “circuit switching”, last accessed 2009.03.22. URL, http://en.wikipedia.org/wiki/Circuit_switching

[2] Wikipedia “packet switching”, last accessed 2009.03.22. URL, http://en.wikipedia.org/wiki/Packet_switched

[3] Wikipedia “Adaptive multi-rate compression”, last accessed 2009.03.22. URL, http://en.wikipedia.org/wiki/Adaptive_multi-rate_compression

[4] Per Beming, Erik Dahlman, Wolfgang Granzow, Stefan Parkvall, and Pontus Wallentin, “WCDMA High-Speed Downlink Shared Channel Technical Description (Release 5)”, Ericsson Internal Technical Report, March 2003

[5] Stefan Parkvall, Johan Torsner and Mats Sågfors, “WCDMA Enhanced Uplink- Technical Description (Rel 6)”, Ericsson Internal Technical Report, December 2004

[6] 3GPP technical document S4-060192, RAN 1, “LS on RAB and the error-delay-profile for the performance characterization of VoIP over HSDPA/EUL”, 3GPP, May 2006

[7] 3GPP Tdoc R2-080292, Nokia Corporation, Nokia Siemens Networks, Qualcomm Europe, Huawei, “Delay and loss rate for CS over HSPA”, 3GPP, January 2008

[8] Ericsson AB, White Paper, “Basic concepts of HSPA”, February 2007

[9] 3GPP Technical Specification 26.114 V7.5.0, “IP Multimedia Subsystem (IMS); Multimedia Telephony; Media handling and interaction”, 3GPP, June 2008

[10] 3GPP Tdoc R2-074018, Nokia Siemens Networks, Nokia, “CS-HSPA De-jitter Buffer and Adaptation Unit”, 3GPP technical document, October 2007

[11] Henrik Rasmussen and Thomas Lund, “Adaptive Jitter Buffering in a Voice over IP Application”, Masters thesis, University of Karlskrona/ Ronneby, May 2000

[12] 3GPP Technical Specification 25.301 V8.3.0, “Radio Interface Protocol Architecture”, September 2008

[13] Rohde & Schwarz, “High Speed Uplink Packet Access (HSUPA) White Paper”, January 2006

[14] J. Sjoberg, M. Westerlund, A. Lakaniemi, Q. Xie, “RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs”, IETF RFC 4867, April 2007

[15] Pin Hu, “The Impact of Adaptive Playout Buffer Algorithm on Perceived Speech Quality Transported over IP Networks”, Master thesis, THE UNIVERSITY OF PLYMOUTH September 2003

[16] Wikipedia “Handoff”, last accessed 2009.03.22. URL, http://en.wikipedia.org/wiki/Handoff

[17] 3GPP Technical Specification 26.071 V8.0.0, “Mandatory speech CODEC speech processing functions; AMR speech CODEC; General description”, 3GPP, December 2008

[18] 3GPP Technical Specification 26.091 V8.0.0, “Mandatory speech CODEC speech processing functions; AMR speech CODEC; Error concealment of lost frames”, 3GPP, December 2008

[19] Roland Corporation, "UA-25 24 bit/96 kHz USB Audio Capture", web page, last accessed March 23, 2009 http://www.roland.com/products/en/UA-25/

[20] Huawei, “Support CS services on HSPA”, 3GPP technical document R2-073974, Shanghai, China, October 2007

[21] NSN, Nokia, “Supporting CS over HSPA”, 3GPP technical document R2-073487, Athens, Greece, August 2007

[22] 3GPP Technical Specification 25.322 V8.3.0, “Radio Link Control (RLC) Protocol Specification”, September 2008

A. HSPA Data Flow Illustration

The PDU (Protocol Data Unit) and SDU (Service Data Unit) refers to the data delivered between layers. The PDU at one layer is the SDU of the lower layer. The SDU at one layer is the 'payload' of this layer's PDU.

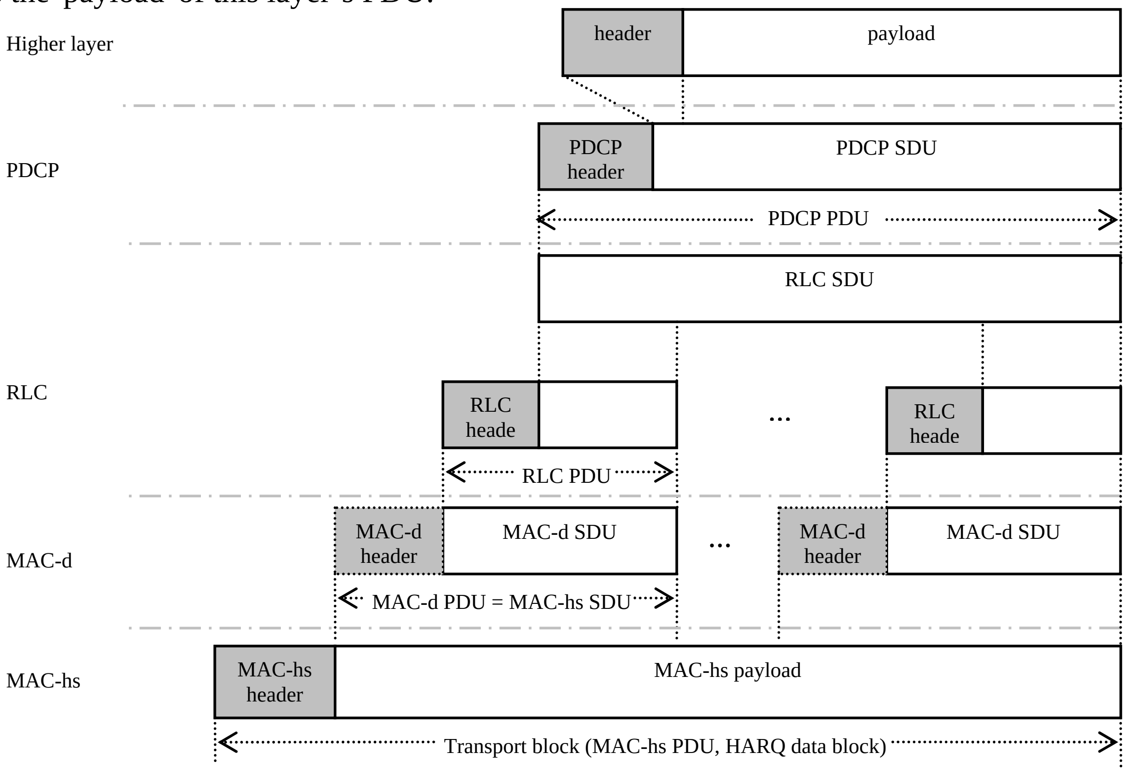

Figure A.1: Data flow of HS-DSCH

The PDCP can optionally perform header compression. Each PDCP PDU is equivalent to an RLC SDU. The RLC SDUs are segmented into smaller blocks. An RLC PDU is comprised of a data segment and the RLC header. The RLC PDU is equivalent to a MAC-d SDU. If logical channel multiplexing is performed on MAC-d, a header is added to form a MAC-d PDU. Otherwise no MAC-d header is needed. A MAC-d PDU is equivalent to a MAC-hs SDU. A number of MAC-d PDUs of possibly different sizes are assembled into one MAC-hs PDU which consists of the MAC-hs payload and a MAC-hs header. A MAC-hs PDU is identical with a Transport Block.

For HSUPA the data flow structure is more or less the same except that MAC-hs is replaced by MAC-es and a number of MAC-es PDUs are concatenated into one MAC-e SDU, then with a MAC-e header added to form the MAC-e PDUs, which is identical with a Transport Block. Figure A.2 shows the protocol data flow.

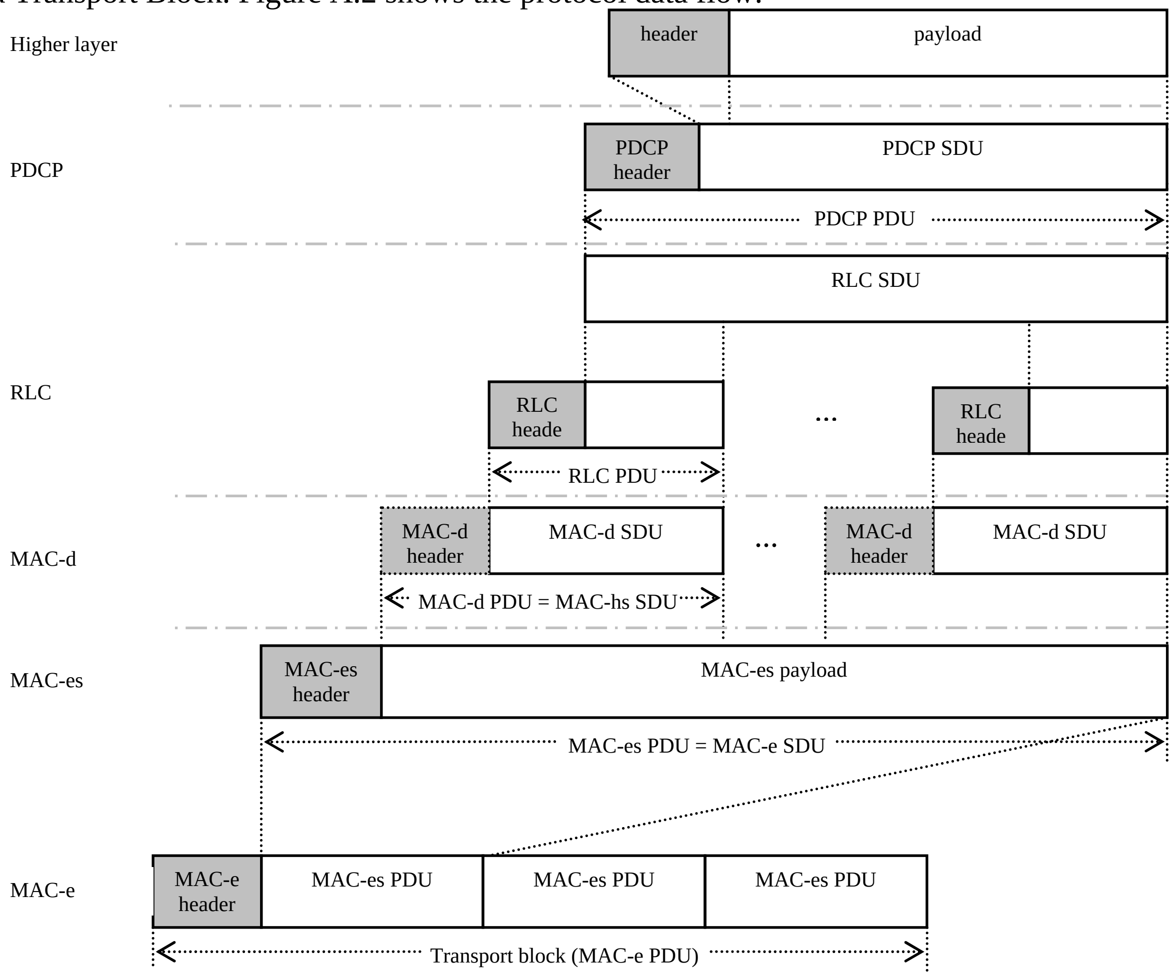

Figure A.2: Data flow of E-DCH

B. Pseudo code of Designed JBM

The pseudo code basically consists of two main functionalities:

- Reception: receives the packet from a network interface and stores them in the jitter buffer. This rouete is called whenever a packet arrives at the JBM.
- Decoding: extracts a packet to forward to the decoder every 20ms. It is called every 20ms after the jitter buffer reaches the initial buffer level.

Moreover, some variables and constants are explained below.

Table B.1: Variables in the pseudo code

Variable	Description
new_buffer_level	Buffer level calculated for new talk-spurt. Measured in number of packets
buf_max	Maximum buffering delay among the most recent packets
buf_min	Minimum buffering delay among the most recent packets
currentSN	Sequence number of the current receiving packet
requiredSN	Sequence number of the packet demanded by decoder
buffer_size	Number of packets in jitter buffer
<vector>	A vector that stores buffering delay of each packet

Constants:

frame_length = 20ms

B.1. Static JBM

Reception

```
{
  If (currentSN < requiredSN)
  {
    late loss, disgard packet;
  }
  Else
  {
    add packet to jitter buffer;
    buffer_size++;
  }
}
```

Decoding

```
{
  If (buffer_size==0)
  {
    //buffer underrun
    generate NO_DATA;
  }
  Else
  {
    if (sequence gap)
    {
      //demanded packet is missing
      generate NO_DATA;
    }
    Else
    {
      //regular delivery
      deliver the oldest packet to decoder;
      //update buffer size and next demanded packet
      buffer_size--;
      requiredSN++;
    }
  }
}
```

B.2. Adaptive JBM

Reception

```
{
  If (SID_first)
  {
    //the first SID after speech
    enter DTX;
  }
  If (speech_onset)
  {
    //beginning of new talk-spurt;
    start rebuffering;
    //calaulate new buffer level for new talk-spurt
    If (<vector>.size < 200)
      new_buffer_level=int((buf_max-buf_min)/frame_length);
    else
    {
```

```
        copy the last 200 values in <vector> to <vector_1>;
        sort <vector_1>;
        a=<vector_1>.last;
        b=<vector_1>.second;
        new_buffer_level=int((a-b)/frame_length);
      }
      <vector>.clear;
    }
    If (currentSN < requiredSN)
    {
      late loss, disgard packet;
    }
    Else
    {
      add packet to jitter buffer;
      buffer_size++;
    }
    If (rebuffering && (buffer_size > new_buffer_level))
    {
      //rebuffering completed
      start delivering speech packet of new talk-spurt to decoder;
    }
  }
```

Decoding

```
  {
    If (buffer_size==0)
    {
      //buffer underrun
      generate NO_DATA;
    }
    Else
    {
      if (sequence gap)
      {
        //demanded packet is missing
        generate NO_DATA;
      }
      Else
      {
        If (rebuffering && (buffer_size<=new_buffer_level))
        {
          //keep rebuffering ongoing
          genarate NO_DATA;
```

```
        }
        Else
        {
          //regular delivery
          deliver the oldest packet to decoder;
          store buffer delay to <vector> for statistics;
          //update buffer size and next demanded packet
          buffer_size--;
          requiredSN++;
        }
      }
    }
}
```

C. Channel Delay and JBM Adaptation

C.1. Synthetic UL Channels:

Medium-load, drop timer 75ms

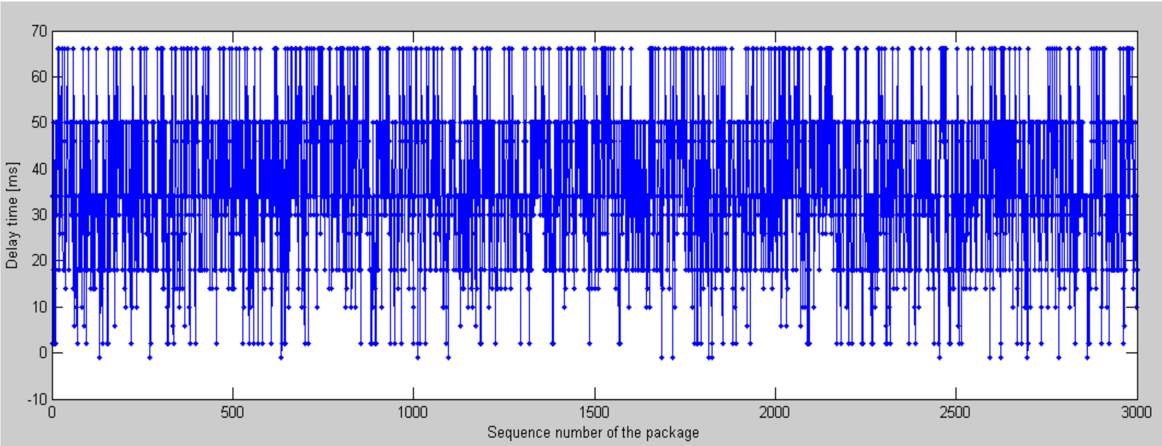

Figure C.1. Channel delay of "Medium-load, drop timer=75ms"

For the medium-load channel, there are a number of spikes up to the maximum allowed delay (66ms) and several losses as well.

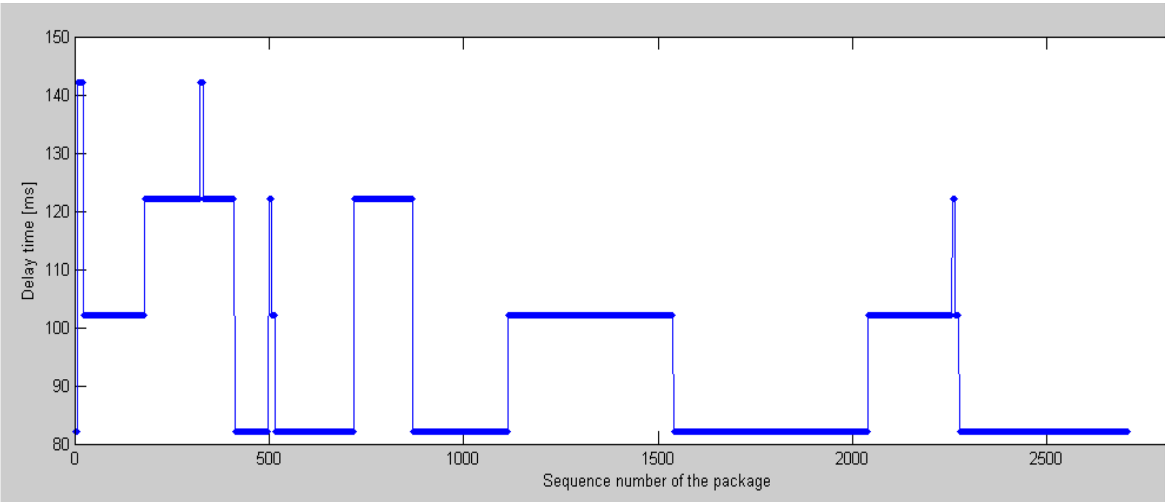

Figure C.2. End-to-end delay with adaptive JBM

Medium-load, drop time 200ms

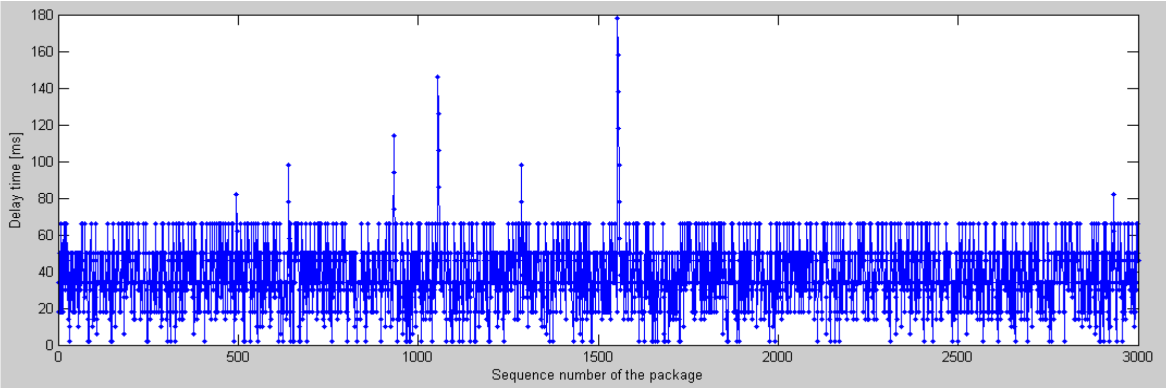

Figure C.3. Channel delay of "Medium-load, drop timer=200ms"

For the medium-load, there are very rare spikes beyond 66ms.

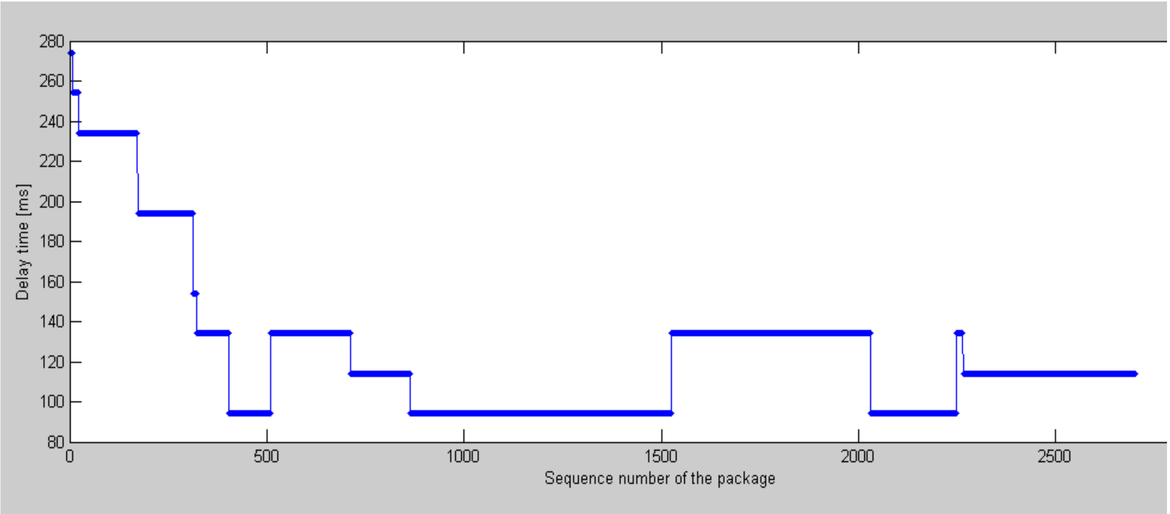

Figure C.4. End-to-end delay with adaptive JBM

The initial buffer level is set to 10 packets according to the drop timer. But it adapts down gradually since the channel is not that challenging.

High-load, drop timer 100ms

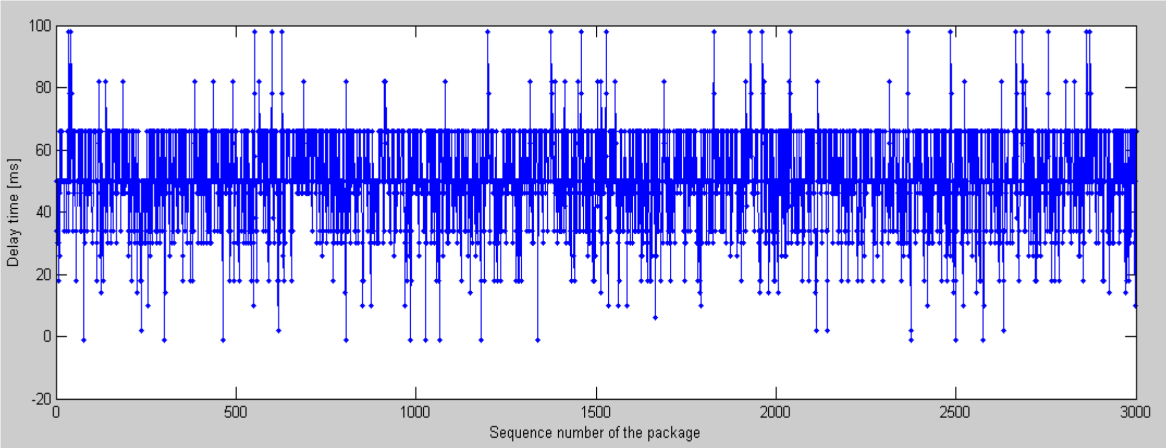

Figure C.5. Channel delay of "High-load, drop timer=100ms"

For the high-load channel, there are many spikes larger than 66ms, and some even reach the drop timer.

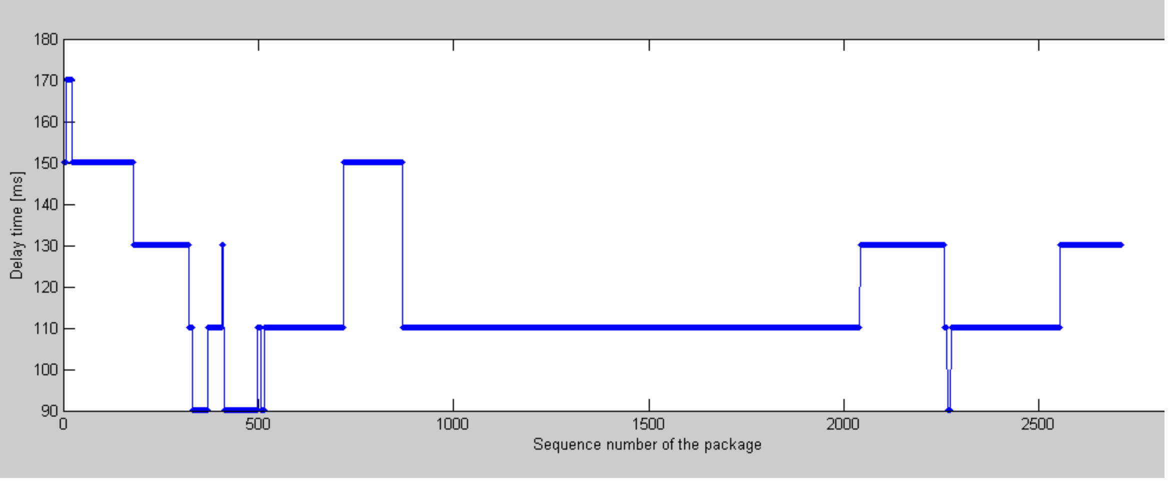

Figure C.6. End-to end delay with adaptive JBM

High-load, drop timer 200ms

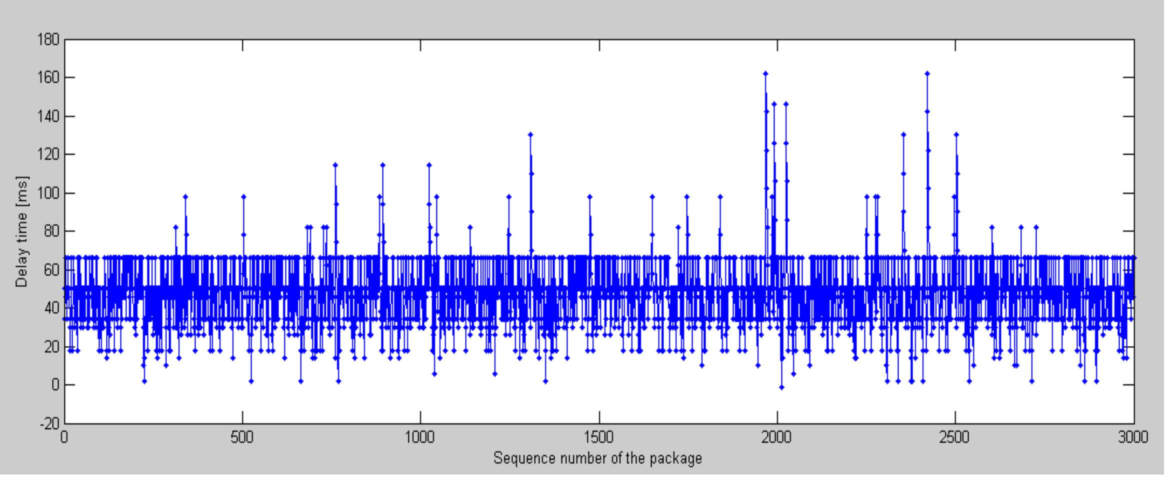

Figure C.7. Channel delay of "High-load, drop timer=200ms"

Larger spikes are introduced when the drop timer is set to 200ms. There are several spikes even beyond 100ms.

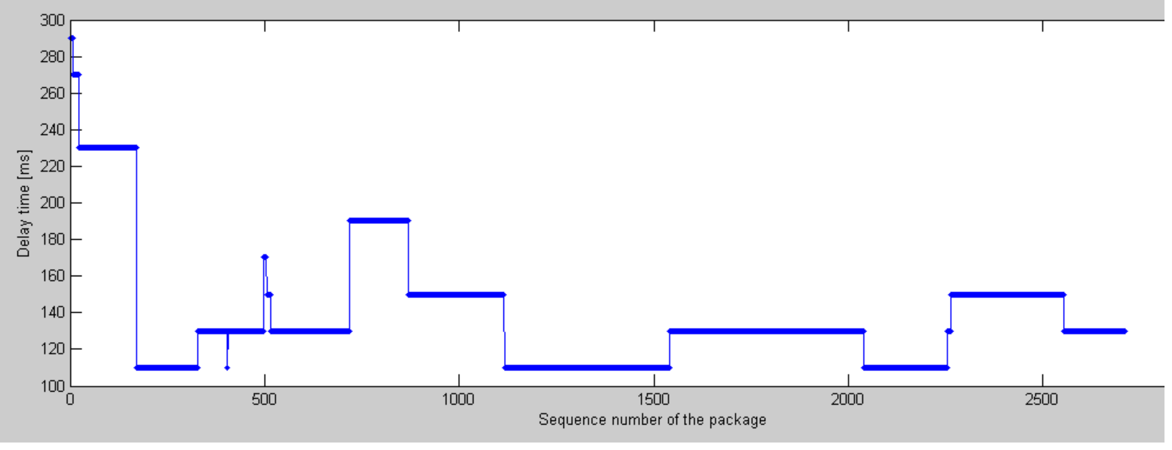

Figure C.8. End-to-end delay with adaptive JBM

Then initial buffer level of 10 packets is too larger even for the high-load channel. Thus the JBM adapts down quickly to minimize the delay.

C.2. 3GPP DL Channels

Low-load, drop timer=55ms. "HSDPA_PA3_100u_G1.65dB_55ms"

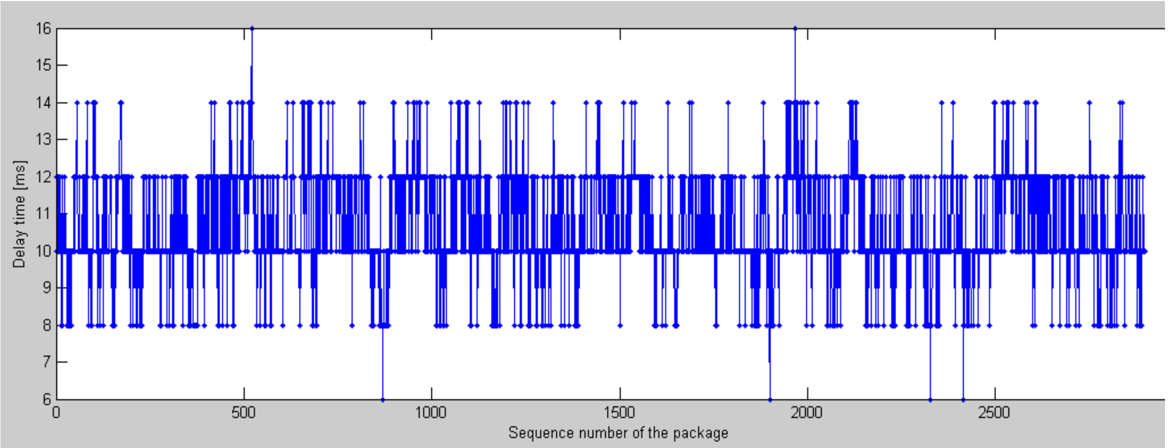

Figure C.9. Channel delay of "HSDPA_PA3_100u_G1.65dB_55ms"

This channel seems perfectly good with the maximum delay of only 16ms.

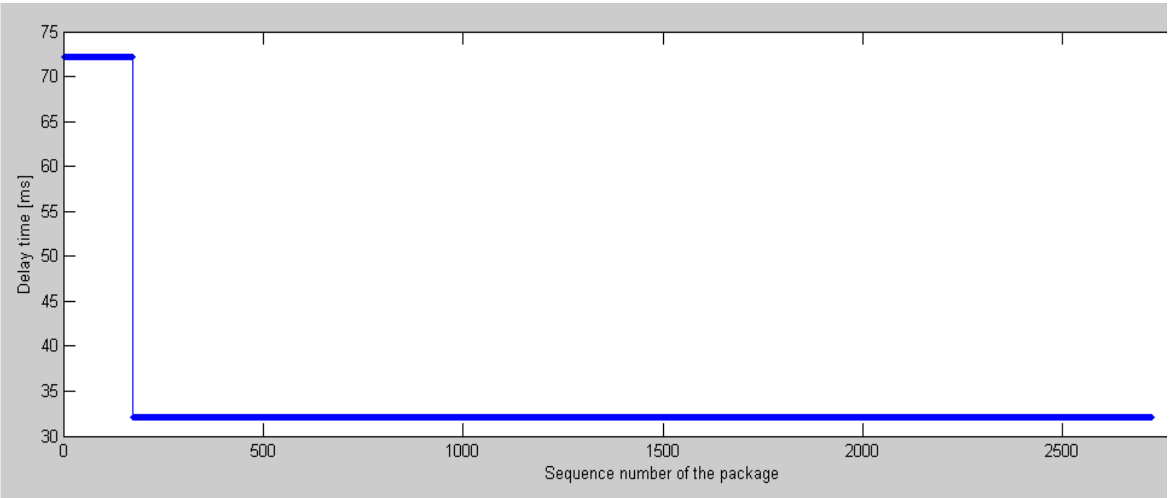

Figure C.10. End-to-end delay

The initial buffer level is set to 3 packets according to the drop timer and adapts down to fit the channel condition.

Medium-load, drop timer=55ms. "HSDPA_PB3_45u_G0.09dB_55ms"

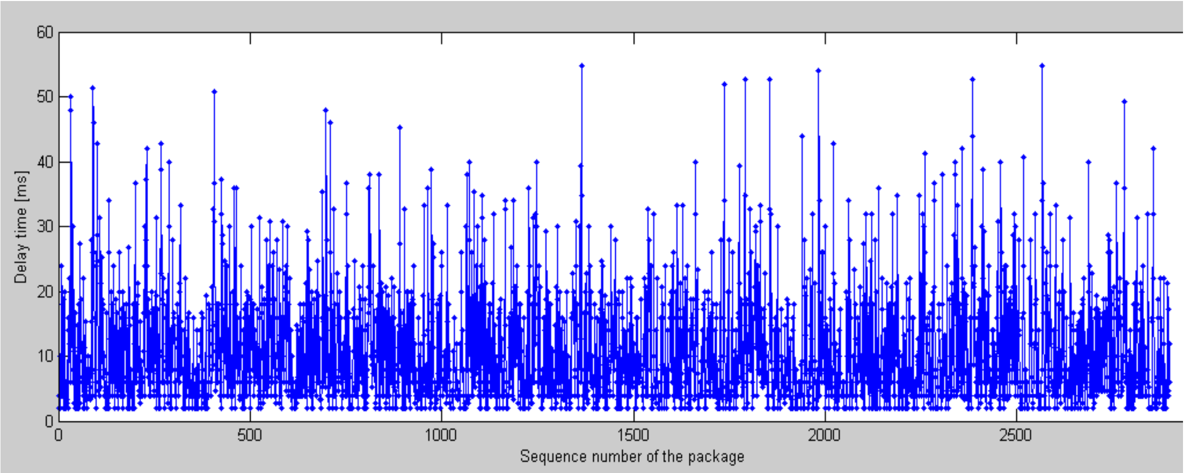

Figure C.11. Channel delay of "HSDPA_PB3_45u_G0.09dB_55ms"

Since channel dependent scheduling is used in HSDPA, the delay values are much more random than in HSUPA.

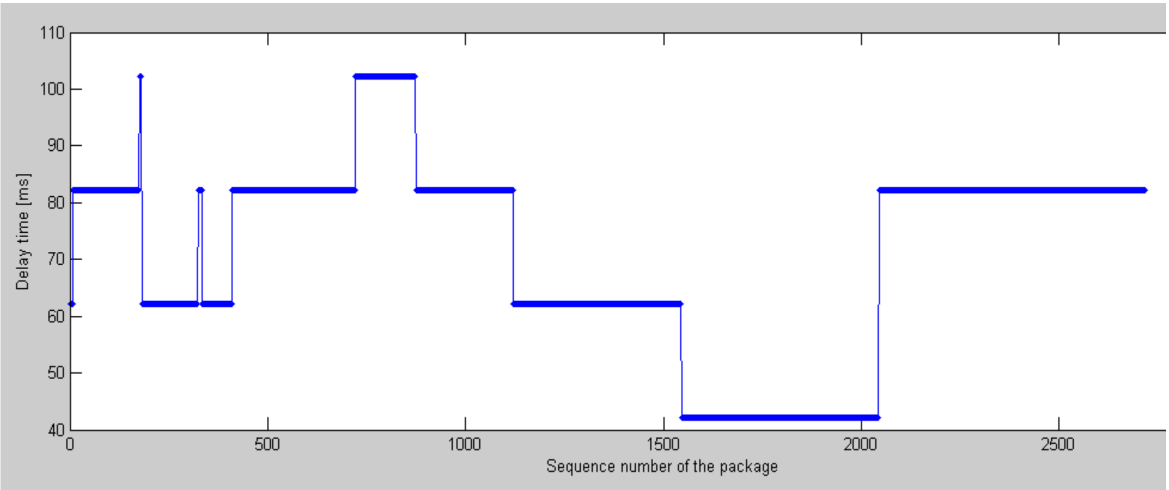

Figure C.12. End-to-end delay

Medium-load, drop timer=155ms. "HSDPA_PB3_45u_G0.09dB_155ms"

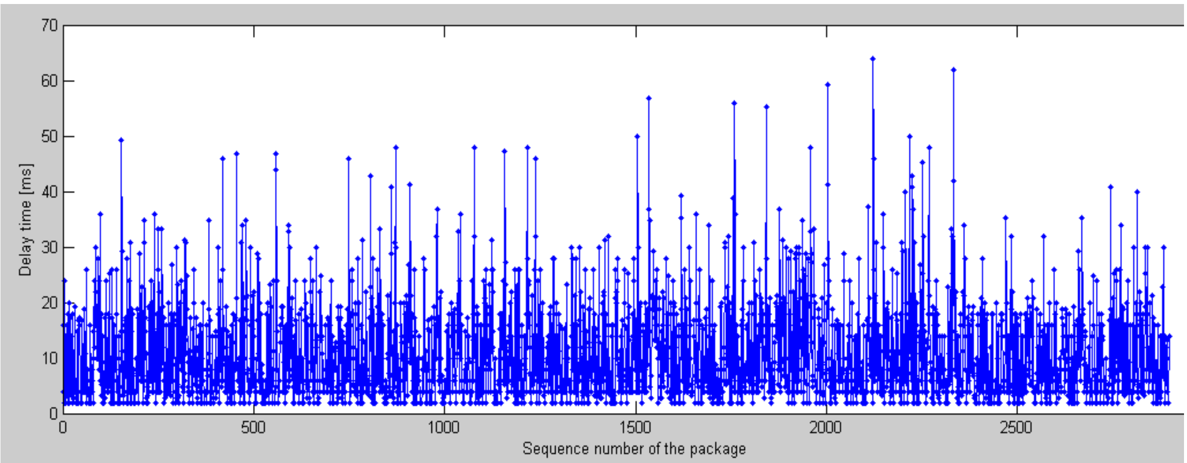

Figure C.13. Channel delay of "HSDPA_PB3_45u_G0.09dB_155ms"

For this medium-load HSDPA channel, very rare spikes can reach beyond 50ms.

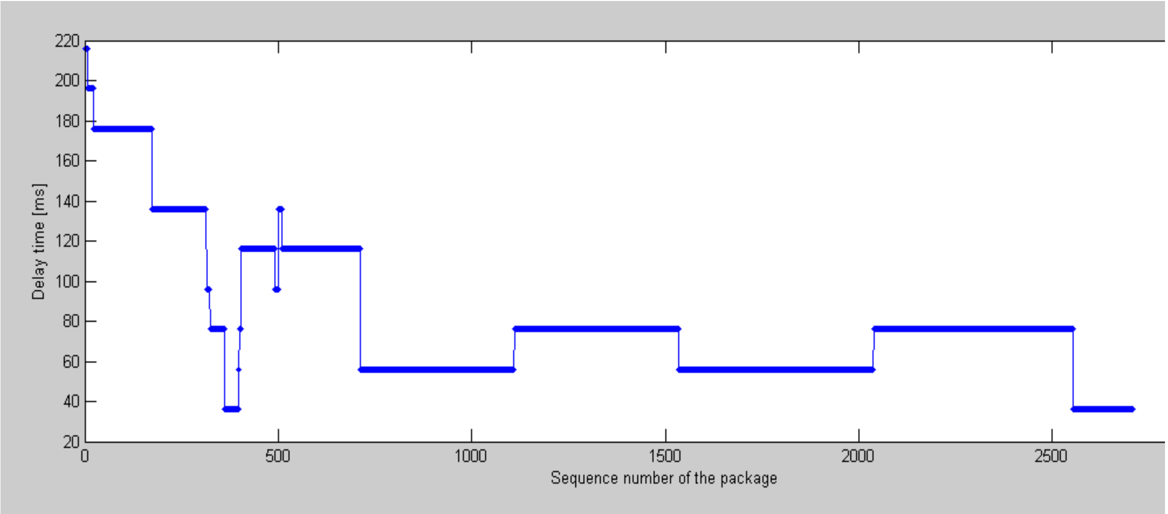

Figure C.14. End-to-end delay

Since the drop timer is 155ms, the initial buffer level is set to 8 packets, which is still too large for the channel. Hence the JBM is able to adapts downwards to reduce delay.

High-load, drop timer=95ms. "HSDPA_PB3_100u_G0.09dB_95ms"

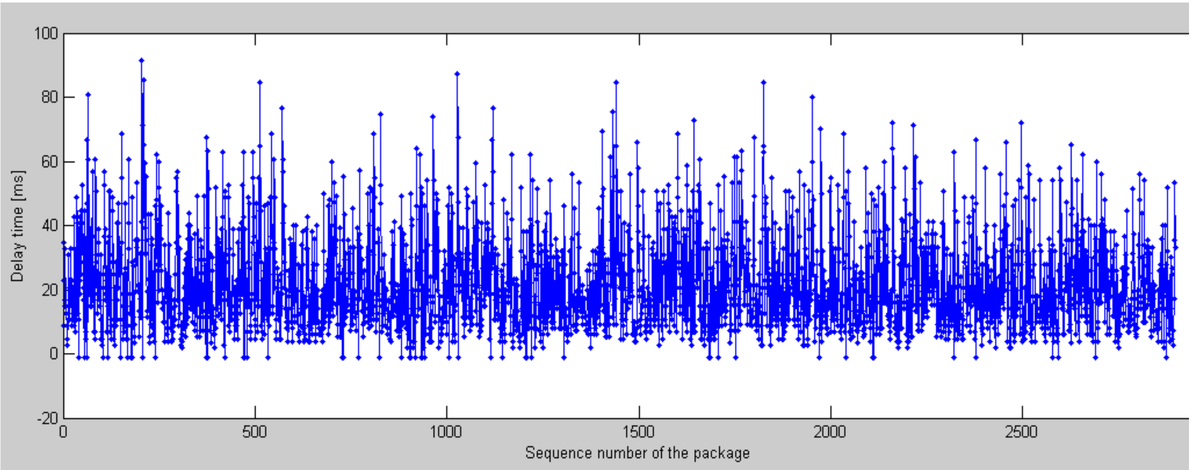

Figure C.15. Channel delay of "HSDPA_PB3_100u_G0.09dB_95ms"

For this high-load HSDPA channel, there are many larger spikes with some even close to the drop timer. Besides, a number of losses are introduced.

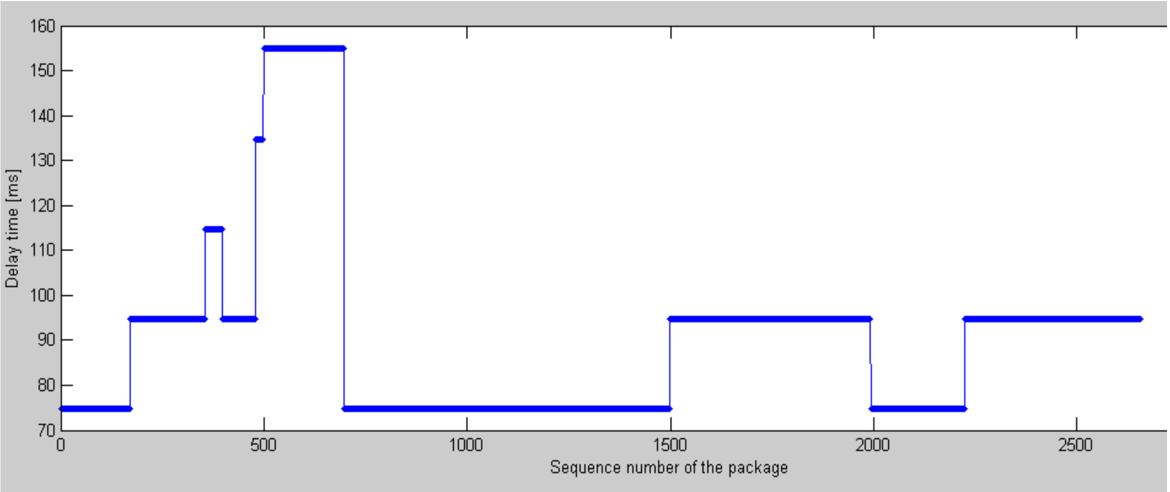

Figure C.16. **End-to-end delay**

The delay suddenly increases at about the 500th packet probably because of jitter loss protection.

High-load, drop timer=155ms. “HSDPA_PB3_100u_G0.09dB_155ms”

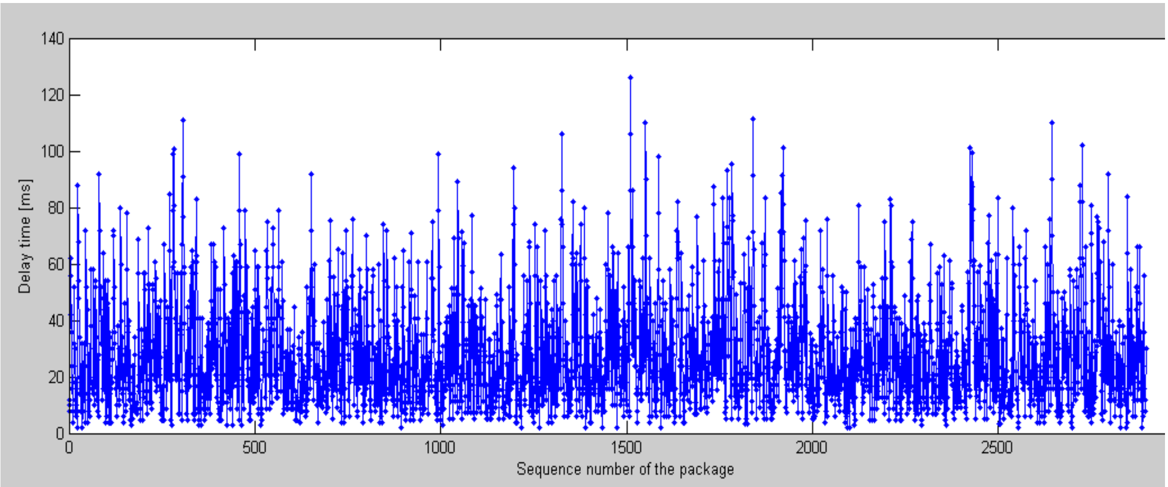

Figure C.17. **Channel delay of “HSDPA_PB3_100u_G0.09dB_155ms”**

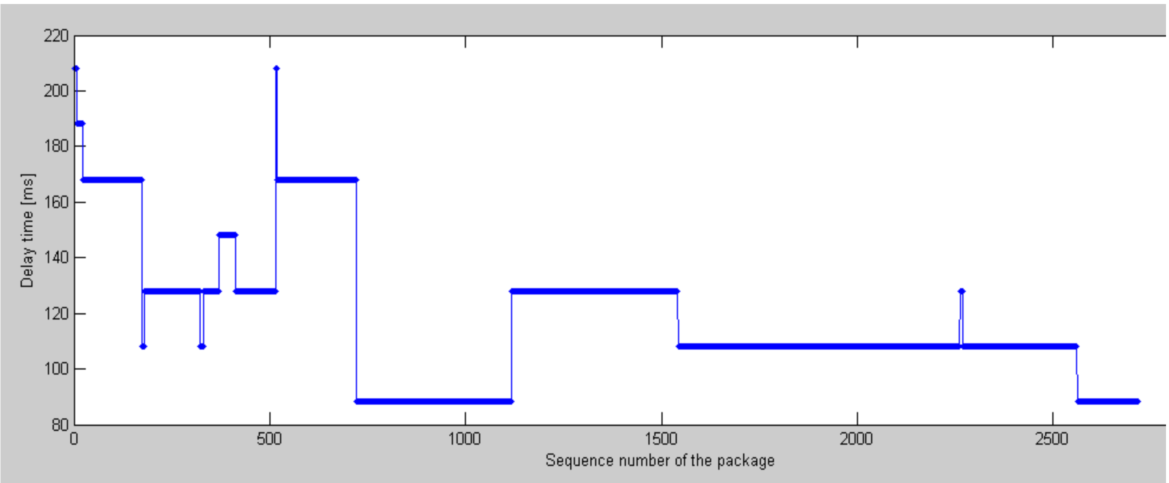

Figure C.18. **End-to-end delay**

The initial buffer level of 8 packets is still over enough for the high-load channel. Hence the JBM adapts down quickly to a proper level.

COS/CCS 2009-03

www.kth.se