Master of Science Thesis in Communication Systems
Department of Electrical Engineering, Linköping University, 2017

A Study on Segmentation for Ultra-Reliable Low-Latency Communications

Linnea Faxén

Master of Science Thesis in Communication Systems

A Study on Segmentation for Ultra-Reliable Low-Latency Communications

Linnea Faxén

LiTH-ISY-EX--17/5039--SE

Supervisor: **Marcus Karlsson**
ISY, Linköpings universitet
Jonas M Olsson
Ericsson
Simon Sörman
Ericsson

Examiner: **Emil Björnson**
ISY, Linköpings universitet

Division of Communication Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Copyright © 2017 Linnea Faxén

Abstract

To enable wireless control of factories, such that sensor measurements can be sent wirelessly to an actuator, the probability to receive data correctly must be very high and the time it takes to the deliver the data from the sensor to the actuator must be very low. Earlier, these requirements have only been met by cables, but in the fifth generation mobile network this is one of the imagined use cases and work is undergoing to create a system capable of wireless control of factories. One of the problems in this scenario is when all data in a packet cannot be sent in one transmission while ensuring the very high probability of reception of the transmission. This thesis studies this problem in detail by proposing methods to cope with the problem and evaluating these methods in a simulator.

The thesis shows that splitting the data into multiple segments and transmitting each at an even higher probability of reception is a good candidate, especially when there is time for a retransmission. When there is only one transmission available, a better candidate is to send the same packet twice. Even if the first packet cannot achieve the very high probability of reception, the combination of the first and second packet might be able to.

Sammanfattning

För att möjliggöra trådlös kontroll av fabriker, till exempel trådlös sändning av data uppmätt av en sensor till ett ställdon som agerar på den emottagna signalen, så måste sannolikheten att ta emot datan korrekt vara väldigt hög och tiden det tar att leverera data från sensorn till ställdonet vara mycket kort. Tidigare har endast kablar klarat av dessa krav men i den femte generationens mobila nätverk är trådlös kontroll av fabriker ett av användningsområdena och arbete pågår för att skapa ett system som klarar av det. Ett av problemen i detta användningsområde är när all data i ett paket inte kan skickas i en sändning och klara av den väldigt höga sannolikheten för mottagning. Denna uppsats studerar detta problem i detalj och föreslår metoder för att hantera problemet samt utvärderar dessa metoder i en simulator.

Uppsatsen visar att delning av ett paket i flera segment och sändning av varje segment med en ännu högre sannolikhet för mottagning är en bra kandidat, speciellt när det finns tid för en omsändning. När det endast finns tid för en sändning verkar det bättre att skicka samma paket två gånger. Även om det första paketet inte kan uppnå den höga sannolikheten för mottagning så kan kanske kombinationen av det första och andra paketet göra det.

Acknowledgments

This thesis has been written at Ericsson Research in Linköping during the spring of 2017. I would like to thank Ericsson Research for having me as a thesis worker and giving me the opportunity to study interesting and challenging problems. A special thanks to my supervisors at Ericsson: Jonas Olsson who always had the time to discuss the thesis and was genuinely interested in my results, and Simon Sörman who helped with the initial setup and let me use his LaTex code for all images of base stations and hexagons throughout this thesis.

I would also like to thank my supervisor Marcus Karlsson and examiner Emil Björnson at Linköping University. Marcus proof-read the report already from an early stage which helped me improve it throughout the thesis work. Emil has done a great job of scrutinizing my report, which enabled me to improve it even further.

Linköping, June 2017
Linnea Faxén

Contents

Notation

Abbreviations

Abbreviation	Definition
3GPP	3rd Generation Partnership Project
5G	Fifth Generation Mobile Network
4G	Fourth Generation Mobile Network
3G	Third Generation Mobile Network
ACK	Acknowledgment
ARQ	Automatic Repeat Request
BLEP	Block Error Probability
CDF	Cumulative Distribution Function
CQI	Channel Quality Index
DL	Downlink
DRX	Discontinuous Reception
EMBB	Enhanced Mobile Broadband
FDD	Frequency Division Duplex
FDM	Frequency Division Multiplexing
FEC	Forward Error Correction
HARQ	Hybrid Automatic Repeat Request
IMT	International Mobile Telecommunications
IR	Incremental Redundancy
ITU	International Telecommunications Union
ITU-R	International Telecommunications Union Radiocommunication Sector
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
MMTC	Massive Machine Type Communication
NAK	Negative Acknowledgment
NR	New Radio
OFDM	Orthogonal Frequency Division Multiplexing
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
SNR	Signal to Noise Ratio
SINR	Signal to Interference and Noise Ratio
TDD	Time Division Duplex
TDM	Time Division Multiplexing
TR	Technical Report
TTI	Transmission Time Interval
UE	User Equipment
UL	Uplink
URLLC	Ultra-reliable low-latency communication

Defined parameters

Notation	Definition
R	Reliability
L	Latency
Y	Percentage of users in a cell that operate with target link reliability R under latency bound L.
$C(L, R, Y)$	URLLC system capacity for given L, R, and Y
$P_{\mathrm{e}_{\mathrm{tot}}}$	Total probability of error for a packet
P_{e_i}	Probability of error for segment i
P_{e_M}	Probability of error for data in buffer
Δ_{SINR}	Back-off, constant to withdraw from estimated SINR
N_{SINR}	History-size, number of slots to save SINR for
k	Fraction-factor, fraction of segment size to link adapt for

1

Introduction

This chapter initiates the thesis by describing the studied problem, stating the purpose, and formulating the questions the thesis aims to answer. Assumptions made and limitations that exist are also presented. At the end of the chapter, a short overview of the content of the thesis is given.

1.1 Background

> "We are just at the beginning of a transition into a fully connected Networked Society that will provide access to information and sharing of data *anywhere* and *anytime* for *anyone* and *anything* [1]."

Today, a large number of devices are connected, mobile phones connected to a mobile network and to the internet, as well as computers connected to the internet. In recent years, this connectivity has expanded into new devices such as tablets and smart watches. As E. Dahlman et al. states in [1], and many with them, this is just the beginning of a fully connected society. The connectivity raises a number of possibilities for new kinds of devices to connect in new places. These new kinds of devices can be household appliances, traffic control devices, sensors and much more. In addition, users of mobile phones and computers always demand a higher data rate.

To enable this increasing demand of data rate, access to communication in new areas, and different usage scenarios, a new mobile network is being developed and standardized. This mobile network is called Fifth Generation, or 5G. Exactly what capabilities a 5G mobile network will have and what requirements a mobile network must fulfill in order to be called 5G, is still being standardized. The standardization procedure is performed in standardization groups such as the 3rd Generation Partnership Project (3GPP) and the International Telecommunications Union (ITU). This work was ongoing when this thesis was written.

The ITU have not specified the requirements for a 5G mobile network yet, but they have released a report called "Framework and overall objectives of the future development of IMT for 2020 and beyond"[2]. This report describes the ITUs vision of the 5G society and is used as a framework to develop the requirements for 5G. In the report, the ITU have identified three usage scenarios to support a diverse range of applications which all will be part of the upcoming 5G standard. The scenarios are: enhanced mobile broadband (EMBB), massive machine type communication (MMTC) and ultra-reliable and low-latency communications (URLLC). High reliability in communications means that the probability of receiving and decoding a packet correctly is very high. Latency is the time from the moment a transmitter decides to transmit a packet until the moment a receiver has received and successfully decoded that packet. EMBB is directed towards mobile phones and will ensure higher data rates. MMTC covers connection of a massive number of machines (instead of mobile phones held by people as in EMBB) that have low demands on data rate, for example sensors that transmit data very seldom. URLLC communication is characterized by high demands on availability, latency and reliability. Possible scenarios are wireless control of factories and transportation safety [2].

In URLLC, to be able to meet the demand on high reliability, the usage of diversity in both frequency and space can be employed. Diversity is a method to improve a packet's reliability by transmitting the packet over multiple channels and combining the received packets into one more reliable packet. The reliability is improved since the different channels experience different levels of fading and interference, so that if one channel is heavily interfered the other channel hopefully has a better quality. In frequency diversity, the multiple channels are multiple frequency bands. Transmitting with multiple antennas utilizes the diversity in space since each antenna forms a different communication channel. Time diversity is difficult to exploit due to the targeted very low latency, otherwise the message could be repeated to achieve a higher reliability. In order to ensure lower latency, the Transmission Time Interval (TTI, duration of a transmission over the radio interface) can be shortened [3].

These improvements for reliability and latency are a good start but not enough to meet 99.999% reliability and 1 ms end to end latency which are the expected requirements for industrial applications [4, Ch.7, Sec.3]. The industrial application scenario is one of the most demanding since it requires both high reliability and very low latency while other URLLC scenarios trade off either reliability or latency.

Link adaptation is the process of adapting a data transmission to the channel quality. The channel quality in radio communications typically changes rapidly and significantly, and the link adaptation tries to adapt to these changes with the help of information about the channel quality provided by the user. These changes to the channel quality occur for a couple of reasons. Fading causes variations in the channel attenuation, frequency selective fading causes rapid and random variations while shadow fading and path loss significantly affect the average received signal strength [5, Ch.6, p.79]. In addition, interference from nearby users' transmissions impact the interference level at the receiver.

In earlier communication systems the goal of the link adaptation has been to deliver as much data as possible over the channel. For URLLC, the goal is to deliver as much data as possible while fulfilling the reliability and latency demands. To fulfill this, a more restrictive link adaptation is needed that assigns resources to meet the increased reliability, sometimes without retransmissions. However, the link adaptation should not over-assign resources, since that would support fewer users in the network.

One part of the link adaptation procedure is segmentation. As used in this thesis, segmentation means the process of splitting a packet into multiple smaller segments since the whole packet cannot be transmitted in a single transmission while subject to the error constraint. This means that, based on the current measured information about the channel quality, the data transmission of the whole packet would have a probability of error larger than the error constraint. For a reliability of 99.999% the error constraint is 10^{-5}. In URLLC, this means that we must adapt each segment's error constraint so that the whole packet achieves the targeted reliability. However, how to select the error constraint for each segment is nontrivial. In addition, the splitting of a packet in order to achieve a higher reliability might not be the best approach.

1.2 Purpose

This master's thesis proposes and evaluates methods to improve segmentation in Long Term Evolution Advanced (LTE-A) to better suit the needs of URLLC. (LTE-A is a fourth generation communication system.) The aim is to come up with guidelines for how segmentation should be implemented for URLLC and get insights into what URLLC benefits from, both when packets must be delivered in a single transmission and when the users have one retransmission available.

Segmentation is of interest to study, since segmentation must be handled in some way for URLLC and preferably in a way which meets its stringent requirements. In addition, it is hard to foresee what methods would yield the best results since decisions taken by the methods that are taken very seldom, (such as when to segment and how to segment in certain cases) also can have a large impact in URLLC. Therefore, a thorough study that simulates a whole network is needed to examine the problem.

1.3 Problem Formulation

The thesis aims to answer the following questions:

- How can segmentation be handled in URLLC?
- What way of segmentation yields best results with respect to URLLC capacity and resource efficiency?

URLLC capacity and resource efficiency will be used as criteria to evaluate the methods where URLLC capacity follows the definition set out by 3GPP in [6, Ch.13, Sec.2] and which will be described in Chapter 2.

1.4 Assumptions and Limitations

This thesis studies segmentation and evaluates the result by simulating a communication network. In this network, there is a multitude of parameters and settings that can be varied and that affect the results. To get a reasonable scope of the thesis, many of these parameters are not varied but set to the values recommended by 3GPP or the supervisors. All of these assumptions on the network are described in Section 3.1.

Another assumption is that if a packet is split into segments, the reception of each segment is modeled as independent from the reception of other segments. This is a simplification that is presented and argued for in Section 2.4.2.

The thesis does not study the connection procedure either, the simulations start collecting data once all users have established a connection with the base station, and the base station transmits packets to the users periodically. This is in order to only study the data transmission from base station to users.

In URLLC the target reliability has been proposed to be at $1 - 10^{-5}$ (99.999%). In order to get reliable results from a simulation simulating such a low probability of error, very long simulation times are needed. Therefore lower target reliabilities (higher probability of error), such as $1 - 10^{-3}$ and $1 - 10^{-4}$, are used instead.

1.5 Structure of the Report

To give the reader an overview of the master's thesis, the content of each chapter is summarized below.

Chapter 1: Presents the problem and questions to be answered.
Chapter 2: Provides relevant theoretical background for the thesis by describing LTE-A, standardization, URLLC and segmentation.
Chapter 3: Describes in detail how the work was carried out.
Chapter 4: Presents obtained results for single transmission.
Chapter 5: Presents obtained results for retransmission.
Chapter 6: Discusses achieved results, scrutinizes the used method and examines the thesis from a wider perspective.
Chapter 7: Presents answers to the stated questions and ideas for future work.

2

Theoretical Background

This chapter describes the theoretical background needed to understand the thesis. As such, it does not present any new contributions to the area, but merely describes the current standards and research, except for Section 2.4.2. In Section 2.4.2, it is described how the probability of receiving a packet has been modeled in this thesis.

The elements that will be covered are: LTE-A, standardization, particular features and demands of URLLC as well as segmentation. The chapter aims to provide a sufficient background to be able to understand the method, results and discussion of the thesis.

2.1 LTE-Advanced

From the initial agreements in 3GPP it is clear that 5G will resemble LTE-A in some aspects such as waveform and architecture [7]. Therefore, it makes sense to base a study on 5G on an LTE-A network. This section will describe procedures within LTE-A so that the concepts of URLLC can be understood. Should the reader wish for a thorough explanation of LTE-A, the book "LTE/LTE-Advanced for Mobile Broadband" by E. Dahlman et al. [5] is recommended.

2.1.1 Introduction

In a mobile network we have base stations that provide a number of user equipments (UEs) with data transmission services. A base station is placed on a site and covers all UEs within the cell area covered by that site. Multiple sites are placed next to each other to cover a larger area. The site itself contains one or multiple sectors. Sectors are usually modeled as hexagons [6, Sec.A.2.4]. In Figure 2.1, a site layout with 7 sites where each site has three sectors, is illustrated.

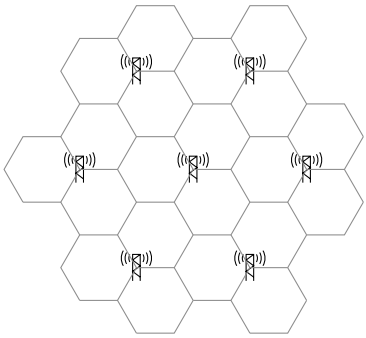


Figure 2.1: *An example of a site layout with 7 sites.*

Data transmitted from the base station to the UE is said to be in the downlink (DL) direction. The other direction, data from the UE to the base station is denoted by uplink (UL), which is illustrated in Figure 2.2 [5, Ch.1, p.6].

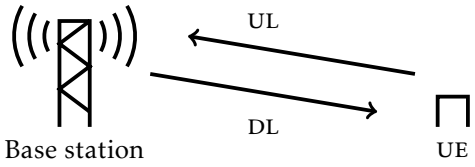


Figure 2.2: *Illustration of UL and DL.*

In order to receive DL transmissions while transmitting UL transmissions, the transmissions must be separated. To separate the UL and DL transmissions, they can be transmitted in the same frequency band but at different times. This is called Time Division Duplex (TDD). The other mode is Frequency Division Duplex (FDD) where the separation occurs in frequency instead of time, the UE transmits UL and DL at the same time but on different frequency bands.

Time in LTE-A is divided into radioframes, subframes, slots and Orthogonal Frequency Division Multiplexing (OFDM) symbols. OFDM is the modulation scheme used in LTE-A which is described in Section 2.1.2. The radioframe is 10 ms long and consist of 10 subframes of 1 ms each, as illustrated in Figure 2.3. Each subframe in turn consists of two slots of seven OFDM symbols each. A subframe is the smallest schedulable unit of time in LTE-A. This corresponds to the duration of a TTI [5, Ch.10, p.144].

Transmission Procedure

In LTE-A, when a UE has data to transmit to the base station (data in the UL direction), the UE requests scheduling from the base station and receives a grant that describes the resources on which the UE can transmit. The resources assigned to a user is time and frequency. Time in the form of subframes the user is allowed to transmit in, and frequency in the form of frequency bands on which the user is

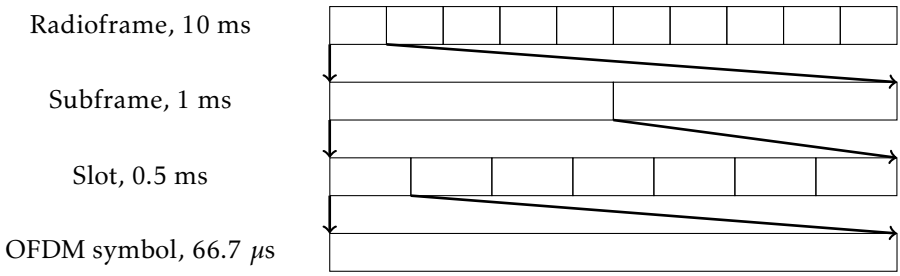


Figure 2.3: *Frame structure of LTE.*

allowed to transmits its data. The base station schedules which UEs are allowed to transmit and on what resources. In DL, the base station schedules which UEs to transmit to and on what resources.

Having been assigned resources, the user chooses the transmission parameters to use for its transmission. This process is called link adaptation and is closely related to scheduling. Thereafter, the data is transmitted through the air from the base station to the UE using OFDM (when transmitting in the DL direction). Both scheduling and link adaptation try to adapt the transmission to the varying radio link using information about the channel quality from the UE. However, due to the random nature of the variations in the channel quality, a perfect adaptation to the radio link is impossible. Packets might be lost due to a failed link adaptation, channel fading or interference from other users. When a packet is lost, a Hybrid Automatic Repeat Request (HARQ) procedure requests retransmissions of the packet. The following sections will describe each of these steps in greater detail, focusing on the DL.

2.1.2 OFDM

OFDM is a modulation scheme where the data is modulated twice. The digital data is first modulated using conventional modulation schemes such as Quadrature Phase Shift Keying (QPSK) or Quadrature Amplitude Modulation (QAM), chosen by the link adaptation. The modulated data stream is then split into N data streams, each modulated by a specific waveform. The waveform used for one of the N data streams is usually called a subcarrier. All of the modulated streams are then added together to form the baseband signal. In order to separate the OFDM symbols at the receiver, the OFDM symbols must be orthogonal. This is achieved by choosing subcarriers that are pairwise orthogonal over the duration of the OFDM symbol [8].

One of the benefits with OFDM transmission is the ability to transform a high rate data stream into multiple low rate data streams that can be transmitted in parallel. In addition, a frequency selective fading channel is split into multiple frequency flat fading channels which makes the transmission more robust against fading [8].

Another feature of OFDM is the tight packing of the subcarriers in the fre-

quency domain. However, a smaller subcarrier spacing also makes the OFDM transmission more sensitive to Doppler spread and other kinds of frequency inaccuracies. LTE-A uses a subcarrier spacing of 15 kHz [5, Ch.3, p.40].

In LTE-A, 12 subcarriers during one OFDM symbol are grouped into a resource-block which can be assigned to a user as a resource [5, Ch.9, p.129]. Consecutive resource-blocks in frequency are grouped into subbands. In this thesis, a subband corresponds to a resource-block (the group consists of only one resource-block).

2.1.3 Scheduling

In LTE-A, dynamic scheduling is the standard, where the scheduling decisions dynamically change from subframe to subframe and the scheduling information is transmitted to the UEs [5, Ch.13, p.272]. Another option is semi-persistent scheduling, where the scheduling decision is transmitted to the UE and the UE is notified that this scheduling assignment is valid for every n:th subframe. This can be useful to reduce overhead control signaling.

Dynamic Scheduling

Dynamic scheduling enables the scheduler to adapt the resources to the varying channels the UEs experience. From subframe to subframe the base station allocates different frequency bands or number of resources depending on the channel quality of the UE [5, Ch.6, p.79]. Depending on the UE's position, different frequency bands can vary greatly in quality and a scheduler that can use this to its advantage will achieve a higher system capacity. System capacity here refers to a higher total data rate provided on average from each base station site and per hertz of licensed spectrum, this measure of capacity is also called spectral efficiency [5, Ch.1, p.8]. To get an efficient resource utilization, the scheduler typically tries to allocate as few resources as possible per UE while fulfilling the UEs' requirements on quality-of-service (typically requirements on delay and reliability), thereby enabling service to more UEs in the system.

Scheduling Scheme

There are many different ways of choosing which user to schedule. One example is maximum rate scheduling where the user with the instantaneously best radio-link conditions is scheduled. This is beneficial from a system capacity perspective but users experiencing worse radio-link conditions during a longer time, for example due to a longer distance to the base station, might never be scheduled. Another scheduling strategy is the round-robin scheduler where the users take turns using the shared resources and the users are scheduled equally often. This schedules all users but leads to overall lower system performance compared to maximum rate scheduling. A scheduling strategy thus needs to be able to take advantage of the fast varying channel conditions while ensuring some throughput for all users.

The proportional fair scheduler does this by scheduling the user with relatively highest rate — comparing the users instantaneous rate to its own average

rate. A user k is selected according to

$$k = \arg\max_i \frac{R_i}{\overline{R_i}}, \tag{2.1}$$

where R_i is the instantaneous data rate of user i and $\overline{R_i}$ is the average data rate for user i [5, Ch.6, p.84]. The time period over which the rate is averaged must be chosen to make use of the fast channel variations and at the same time limit the long-term differences in service qualities. A too short time period will react strongly to fast channel variations but not take the long-term average into account.

The proportional fair scheduler schedules a user when the quality for that user is better than it is on average, ensuring users are scheduled when they have good quality and thus achieving a better system performance as compared to round-robin. In addition, since one user cannot have better quality on average at all times, it makes sure to schedule more users as compared to maximum rate when some users have a much worse quality than others.

The above described scheduling strategies assign all resources to one user at a time — separating users only in the time domain by Time Division Multiplexing (TDM). In LTE-A however, users are separated both in time with TDM and frequency with Frequency Division Multiplexing (FDM). This enables us to schedule more than one user at a time.

For small packets where all frequency bands at the base station are not needed to transmit the packet, multiple users' packets can be transmitted at the same time but over different frequency bands. A greedy-filling approach can be used where one user is chosen and assigned resources for its transmission until it can transmit its packet, then the second user is assigned resources until it can transmit its packet, and so on until the base station is either out of users or resources. The users are chosen according to the scheduling scheme, for example maximum rate, round-robin or proportional fair.

The scheduling strategy is not standardized by 3GPP [5, Ch.13, p.274]. An implementation of LTE-A might therefore use a different scheduling strategy than another implementation. The actual scheduling strategy might not be known to the public since it can be one of the factors that sets a network apart from its competitors.

2.1.4 Channel State Reporting

Channel state reports describe the channel conditions and are transmitted by the UE to the base station so that the information can be used for scheduling and link adaptation decisions. The exact content of the reports can be configured but usually the reports contain a Channel Quality Index (CQI). The CQI represents the highest modulation and coding scheme that can be used for DL transmission with a Block Error Probability (BLEP) of at most 10%. The base station uses the reported CQI as a recommendation on what modulation and coding scheme to use, but might choose another modulation and coding scheme based on information the UE does not have.

Channel state reports are either aperiodic or periodic. Aperiodic reports are only transmitted when requested while periodic channel state reports are configured to be delivered with a certain periodicity. This can be as often as once every 2 ms. [5, Ch.13, p.283]

The CQI is derived from the Signal to Noise Ratio (SNR), or the Signal to Noise and Interference Ratio (SINR) measured by the UE. The mapping of SNR, or SINR to CQI is performed by the UE and is implementation specific, thus not specified by 3GPP. Some implementations are based on SNR measurements while some are based on SINR measurements, for example. Usually the mapping is done by reading out the decoding block error probability on a curve of decoding error probability plotted against SNR, or SINR, given a certain format. These curves are obtained from theoretical models and simulations.

Should the base station wish to know the actual SINR-value or SNR-value, the same procedure can be reversed at the base station. Given a CQI that corresponds to a certain format, the curves can be studied to find a target BLEP of 10% which will yield the SNR, or SINR, experienced by the UE. Note that each such transformation does introduce modeling errors since the curves are simplifications [9].

Since the mapping of SNR, or SINR to CQI is implementation-specific in the UE, the base station does not know which implementation the UE uses and the base station uses its own implementation that can differ from the UE's implementation. However, all implementations of CQI mapping fulfill the definition that a transmission with the recommended parameters should yield a BLEP of at most 10%. Therefore, even if the base station and the UE use different implementations, the results should be the same.

Another problem with channel state reporting is that the CQI in the report describes what the channel was like at the time the UE measured SNR (or SINR) and mapped the SNR (or SINR) to a CQI value. The channel and interference may change from that moment until the moment a transmission is sent to the UE.

2.1.5 Link Adaptation

Link adaptation is the process of selecting the transport format for a transmission to adapt to the varying radio link quality. This is done by selecting modulation scheme, for example QPSK or 16-QAM, and channel coding rate. The selection is done based on what resources the UE has been allocated, the CQI of the channel and the requirements of the transmission. If the quality is good, a higher order modulation such as 16-QAM or even 64-QAM can be used, which has a higher information rate but also lower reliability. The code rate might also be increased, lowering the number of redundant bits used for error correction [5, Ch.6, p.81].

An estimate of the BLEP for a certain format is calculated from the CQI and a coding model. This is then compared to the requirement on the BLEP and if met, the format can be used for transmission.

Outer Loop

The quality measurement used by the link adaptation is an adjusted CQI-value. The reported CQI is converted to SNR, or SINR (as explained in Section 2.1.4) and adjusted by an outer loop. This is done in order to better adjust the channel estimate to the actual quality of the channel. The outer loop helps combat channel measurement errors and model errors. A typical implementation of an outer loop is described below, however an outer loop can be implemented in other ways as well.

At first, the outer loop has an initial offset value that is applied to the SINR. Then, based on the HARQ feedback from the UE in the form of acknowledgments (ACK) or negative acknowledgments (NAK) this offset is adjusted. Should the base station receive an ACK, the packet was received and decoded without error and our estimated channel quality was correct or pessimistic, which caused us to give more resources than necessary for this transmission. This causes a small change in the offset towards a higher SINR, a higher quality of the channel. On the other hand, if the base station receives a NAK it means the packet was received in error, possibly because we had an optimistic estimate of the channel quality. Therefore the offset is adjusted to a lower SINR, representing that the channel is worse than the CQI suggests. Eventually the offset converges to a value. Should the channel change, the feedback causes the offset to adapt to the new channel [10].

2.1.6 Retransmission

To protect the transmitted data from channel fading, receiver noise and unpredictable interference, LTE-A uses a combination of Forward Error Correction (FEC) and Automatic Repeat Request (ARQ). In FEC, parity bits are added to the information bits transmitted. These parity bits add redundancy to the transmission and can be used to correct errors. The other method, ARQ, detects if the received packet is in error or not. If the packet is received and decoded correctly the transmitter is sent an ACK. Should the packet on the other hand be in error, the transmitter is sent a NAK, demanding that the packet is retransmitted. The combination of FEC and ARQ used by LTE-A is called HARQ. In HARQ, the FEC is used to correct a subset of the errors and error detection is used to detect uncorrectable errors that require a retransmission [5, Ch.6, p.90].

Soft Combining

Requesting a retransmission gives the receiver a new chance of receiving and decoding the packet. However, the packet received in error contains information about the packet even though it is not enough information to decode the packet. In HARQ with soft combining, the packet received in error is stored in a buffer and combined with the retransmission(s) into a more reliable packet. Thereafter, error correction and error detection is run on the combined packet, issuing another retransmission if it is still in error.

Soft combining is usually divided into two types: Chase combining and Incremental Redundancy (IR). In Chase combining, the retransmission consists of the

same set of coded bits as the original transmission. The receiver then uses maximum ratio combining on all received bits of the original transmission and the retransmission. Maximum ratio combining is a type of diversity combining that combines several signals, or in this case received bits in order to get a higher SNR for the signals or received bits. The combined packet is then sent to the decoder. This type of soft combining can be seen as additional repetition coding.

IR soft combining instead creates multiple sets, each set representing the same information bits but containing different parity bits. For each retransmission, another set is typically transmitted. Since the retransmission may contain additional parity bits, the code rate is generally lowered when the previous attempts are combined with the retransmission. IR is the basic scheme in LTE-A [5, Ch.6, p.91-92].

The sets in IR are usually generated by using a low-rate code and puncturing the output. With a rate 1/4-code we transmit three parity bits for each information bit. By transmitting only every third bit of the coded bits (we puncture the first and second bits) we get an effective code rate of 3/4. For the retransmission we puncture the second and third bits instead, transmitting only a third of the bits but different bits. This transmission also has a code rate of 3/4 but combined with the original transmission we now have 2/3 of the total bits coded with a rate 1/4-code which gives us a resulting code rate of 3/8. With a second retransmission we have transmitted all redundancy versions of the bits and achieve a code rate of 1/4. Any additional transmissions will not change the resulting code rate since all redundancy versions have been received.

Types of HARQ

HARQ protocols can be synchronous or asynchronous as well as adaptive or non-adaptive [5, Ch.12, p.250]. In an asynchronous HARQ protocol, the retransmission can occur at any time. Synchronous HARQ protocols, on the other hand, imply that retransmissions occur at a fixed time after the previous transmission. A non-adaptive HARQ protocol requires that the retransmission uses the exact same resources and transport format as the original transmission, while in an adaptive HARQ protocol, the resources and possibly the transport format can be changed between retransmissions. In IR, the number of coded bits and the used modulation scheme can be different for different retransmissions, thus changing the transport format [5, Ch.6, p.91].

LTE-A uses asynchronous adaptive HARQ protocols for the DL and synchronous HARQ protocols for the UL. The UL typically uses a non-adaptive protocol but the possibility exists to use adaptive as well [5, Ch.12, p.251].

HARQ Timing in DL

When the UE has received an encoded packet, the UE tries to decode the packet and sends its acknowledgment back to the base station. In LTE-A the UE transmits its response 4 subframes after it received the data on the DL. From a latency perspective, it would be better if the response was transmitted sooner but this would also require greater processing capacity at the UE.

The base station receives the acknowledgment from the UE and prepares a retransmission, if needed. The retransmission is transmitted 4 subframes after the base station received the acknowledgment. In other words, it is not until 8 subframes after the initial transmission to the UE, that the retransmission is transmitted to the UE in the DL case. To be able to receive new data while processing the earlier received data, the UE has 8 HARQ processes in FDD that can operate in parallel [5, Ch.12, p.255].

2.1.7 DL Data Transmission

To summarize the description of LTE-A, an example of a DL data transmission is described in this section.

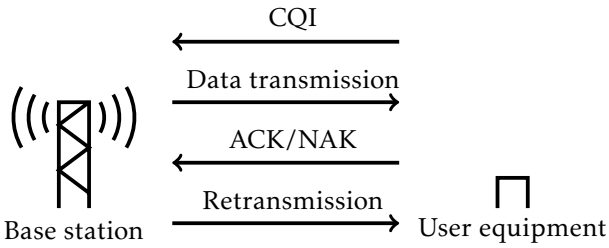


Figure 2.4: *Overview of procedures in a DL transmission.*

As seen in Figure 2.4, the UE periodically transmits channel state reports that contain a CQI-value in this example. This CQI-value is used by the base station to schedule the UE and through link adaptation choose a format for the upcoming data transmission. Thereafter, the data is transmitted using the chosen format. The UE tries to decode the received packet. The result of the decoding is transmitted as an acknowledgment (ACK or NAK) to the base station which, if needed, issues a retransmission.

2.2 Standardization

This section describes the procedure of standardizing a new mobile network as well as the actors and current status of the 5G standardization process. Thereafter, the agreements made so far in regards to URLLC are presented.

2.2.1 Procedure

The ITU develops and maintains recommendations such as the specifications for the radio interface for the different International Mobile Telecommunications (IMT) technologies. These IMT systems consist of IMT-2000 (Third Generation Mobile Network, 3G) and IMT-Advanced (Fourth Generation Network, 4G) at present. The specifications describe what is required of a system to be called, for example 4G [1]. These specifications are created in a consensus-based manner

where companies and organizations submit their proposals to the specification. One such organization is 3GPP whose technologies are widely deployed in the world with LTE-A being the most recent [5, Ch.1, p.8].

So far, ITU has released their vision of 5G and are working on specifying the requirements and evaluation criteria [11]. In the meantime, 3GPP is specifying its requirements, evaluation criteria and proposals of standard based on the vision. These proposals will then be submitted to ITU [5, Ch.1, p.8].

Initial agreements in 3GPP show similarities between 5G and LTE-A, especially for the EMBB scenario [7]. Only minor decisions have been made regarding URLLC and most of them are only to define the requirements and how to evaluate these requirements.

2.2.2 Agreements in 3GPP

To give the reader an overview of the current state of the standardization of URLLC, the agreements with regards to URLLC from the meetings held by 3GPP from the summer of 2016 to January of 2017 will be briefly discussed. The result is summarized in Table 2.1.

Table 2.1: *Summary of agreements concerning URLLC from latest 3GPP RAN1 meetings*

Meeting	Date	Agreements
RAN1#85	2016-05	Capacity should be used for evaluation
RAN1#86	2016-08	Evaluation method and scenarios decided, described in TR 38.802. URLLC capacity defined. Decided on options to consider in regards to multiplexing with EMBB and scheduling.
RAN1#86b	2016-10	NR should support dynamic resource sharing. Study TTI duration and latency based on one retransmission, utilizing HARQ. Single transmission can still be studied.
RAN1#87	2016-11	At least UL transmission scheme without grant is supported for URLLC. Asynchronous and adaptive HARQ is supported for DL.
NR1	2017-01	For an UL transmission scheme with/without grant, repetition of transmission is supported. Support of mini-slots agreed.

As can be seen in Table 2.1, during the RAN#86 meeting held by 3GPP, the evaluation method and scenarios were decided for URLLC. Options for scheduling of URLLC and multiplexing with EMBB were also suggested. Multiplexing with EMBB might be necessary in areas where there are both URLLC UEs and EMBB UEs, demanding different reliability and latency requirements.

To accommodate for URLLC's low latency, alternative scheduling might be needed. In LTE-A a UE requests scheduling from the base station and receives

a grant that describes the resources on which the UE can transmit. However, this procedure takes time and an alternative could be so-called grant-free transmission, which also was discussed at RAN#86 [12].

The same year, in October, meeting RAN#86b was held were it was agreed that New Radio (NR, the name of 3GPP's 5G technology) should support dynamic resource sharing between different latency and reliability requirements for EMBB and URLLC. This means that resources can be shared between different UEs from different types of scenarios.

It was also agreed to study using at least one HARQ retransmission. The overall HARQ signaling procedure's reliability should also be taken into account in this study. TTI and achievable latency when using one retransmission was also agreed to be studied further [13]. It was also noted that studying retransmissions does not preclude studying single transmission.

During the next meeting, RAN#87, some more decisions regarding resource sharing was made, for example that a URLLC transmission may occur in resources scheduled for ongoing EMBB traffic. Furthermore it was decided that URLLC should support at least one grant-free UL transmission scheme. That URLLC for DL should support asynchronous and adaptive HARQ was also decided [14].

Meeting NR1 decided that for UL transmission schemes, repetitions of a transmission should be supported. The meeting also discussed and decided on requirements for the control channel of URLLC. The notion of mini-slot was agreed and several use cases were agreed to be taken into account when designing these mini-slots. Mini-slots are a concept of transmitting data in a subframe that does not necessarily start where the subframe starts or ends where it ends, and can be smaller than a subframe [15]. Note that so far only the support of mini-slots has been agreed, it is not decided if or how they should be used.

2.3 URLLC

As described in the background of this thesis, 5G envisions a multitude of new scenarios that demand great changes to the mobile network from earlier generations. Even within URLLC there are many different scenarios and requirements. Some demand a reliability of $1 - 10^{-9}$ and a latency of 1 to 10 ms while others demand a lower reliability of $1 - 10^{-5}$ but with a very low latency of 1 ms [4, Ch.7, Sec.3].

This combination of low latency (1 ms) and high reliability ($1 - 10^{-5}$) is the most demanding case. It is derived from what 3GPP calls conventional industrial control applications [4, Ch.7, Sec.3]. In industrial control applications the end to end latency is typically measured between a sensor measuring data and a process logic controller that processes the collected data and instructs the actuators [16]. Either the sensor and process logic controller communicate to each other directly (device to device communication) or a base station handles the communication, relaying the information to the devices. By connecting the process logic controller via cable to the base station the end to end communication essentially becomes one transmission between the base station and sensor.

Wireless technologies for factory applications have received interest in recent years. The wireless technologies are interesting, both because of the predicted lower cost, and increased flexibility. For example, installation and maintenance of cables is costly and requires trained personnel. In addition, replacements might also be needed which stops production. The flexibility of wireless communication also makes it possible to realize different production deployments rapidly. However, the wireless technologies have so far been unable to meet the requirements [16].

2.3.1 Requirements

The general URLLC requirement according to 3GPP is that the reliability of a transmission of one packet of 32 bytes should be $1 - 10^{-5}$, with a user plane latency of 1 ms. User plane latency and reliability are defined in Definitions 2.1 and 2.2, and are both according to the current 3GPP agreements. Other requirements for URLLC might be added at a later time [17, Ch.7, Sec.9].

Definition 2.1. (Latency) User plane latency (L) is defined as the time it takes to successfully deliver an application layer packet/message from the radio protocol layer entry point to the radio protocol layer exit point via the radio interface in both UL and DL directions, where neither device nor base station reception is restricted by Discontinuous Reception (DRX, a mode in which the UE sleeps for certain periods).[17, Ch.7, Sec.5]

Definition 2.2. (Reliability) Reliability (R) is defined as the success probability of transmitting X bits within the user plane latency (L) at a certain channel quality. The time of L seconds corresponds to the user plane latency and includes transmission latency, processing latency, retransmission latency and queuing/scheduling latency (including scheduling request and grant reception if any)[6, Ch.13, Sec.2].

One way of illustrating the latency and reliability is to plot the Cumulative Distribution Function (CDF) of the latency as seen in Figure 2.5. The CDF shows the probability that the latency will be less than or equal to a certain value. Due to transmission errors, all packets might not be received. After a timeout they will be count as lost, which is shown in Figure 2.5 as "Lost Packets". Since the CDF shows the probability that a packet will be received within a certain latency or less, this probability can be seen as the reliability to receive a packet within that latency. For example, in Figure 2.5, R is the probability to receive a packet within 1 ms, so in order to meet the requirements, R should be larger than $1 - 10^{-5}$.

2.3.2 Scenarios

For system-level simulations of URLLC, two main scenarios have been identified by 3GPP [6, Sec.A.2.4]. The scenarios are Indoor Hotspot and Urban Macro. The Indoor Hotspot scenario considers a single floor inside a building that contains multiple rooms and base stations. This scenario is therefore useful to simulate, for example, a floor in a factory building. In the Urban Macro scenario, users are

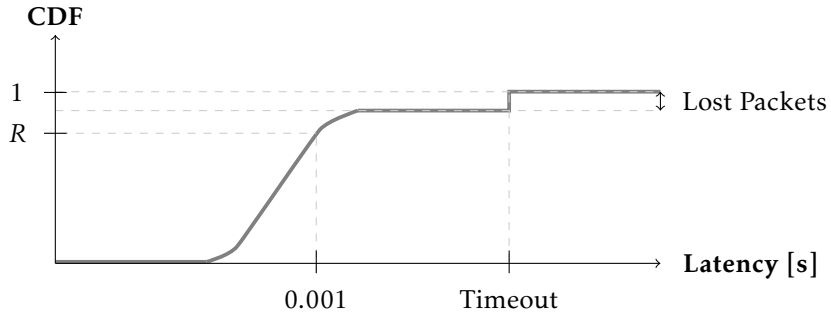


Figure 2.5: *Conceptual CDF of latency.*

assumed to be outdoors at street level or indoors in buildings while the fixed base stations are placed clearly above the surrounding buildings heights [17, Ch.6, Sec.1.4]. Users in Urban Macro can move at pedestrian speed (3 km/h) or slow car speed (30 km/h)[6, Sec.A.2.4].

According to [4, Ch.7, Sec.3], industrial control is usually placed in a geographically limited area but might also be deployed in wider areas (e.g. city-wide networks) where access to them might be limited to authorized users. This city-wide network corresponds to an Urban Macro scenario. Both the Indoor Hotspot scenario and the Urban Macro scenario include inter-cell interference.

2.3.3 Evaluation

3GPP has decided that URLLC capacity will be used as a performance metric for evaluation and feature selection [18]. The URLLC capacity describes how many UEs, or how much load the network can support. For URLLC, the number of UEs that can meet the requirements during a certain load is what is interesting. URLLC capacity is defined in Definition 2.3. Note that URLLC capacity is different from channel capacity, the maximum rate by which information can be transferred over a given communication channel, which is more commonly referred to as capacity in academic literature [5, Ch.2, p.15].

Definition 2.3. (URLLC capacity) URLLC system capacity, $C(L, R, Y)$, is defined as the maximum offered cell load under which Y% of users in a cell operate with target link reliability R under latency bound L. $X = (100 - Y)$% is the fraction of users in outage. A UE is in outage if the UE can not meet the latency L and link reliability R requirements [6, Ch.13,Sec.2]. URLLC capacity is measured in bits per second [bits/s].

This means that for a given value of Y, the URLLC capacity is the maximum offered cell load (number of UEs and packet arrival intensity for these UEs) that the network can support while fulfilling the latency and reliability demands of Y% of these UEs. This is illustrated in Figure 2.6, for $Y = 75$ it is clear that the URLLC capacity is 400 packets per second per user times the size of the packet in bits times the current number of users.

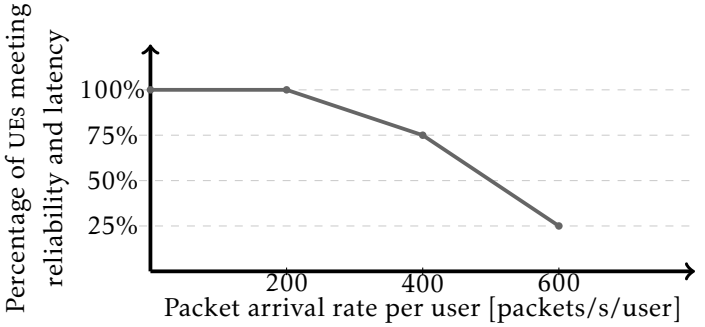


Figure 2.6: *Conceptual image of URLLC capacity.*

2.3.4 Numerology for URLLC

Numerology in this thesis refers to OFDM numerology, the configuration of subcarrier spacing and symbol duration for OFDM. Before the actual numerologies are discussed, the meaning of subframe and slot within this thesis must be defined. 3GPP has decided that the subframe duration is 1 ms also in NR. However, a subframe will no longer be the same as in LTE-A. In LTE-A, a subframe corresponds to the TTI and is the smallest schedulable unit of time. At each TTI, one transmission is sent over the radio link. For NR, multiple numerologies will be supported, where each numerology will have a different TTI. The exact nomenclature for NR has not been decided and so both the name subframe and slot can be found in 3GPP documents to represent a TTI, the context usually gives way to the exact meaning. In this thesis the term slot will be used to denote the TTI of a certain numerology.

The slot will, as the slot in LTE-A, contain 7 OFDM symbols while the duration of the slot will depend on the used numerology. This frame structure is illustrated in Figure 2.7. Note that the number of slots in a subframe depends on the slot duration [6, p. 8].

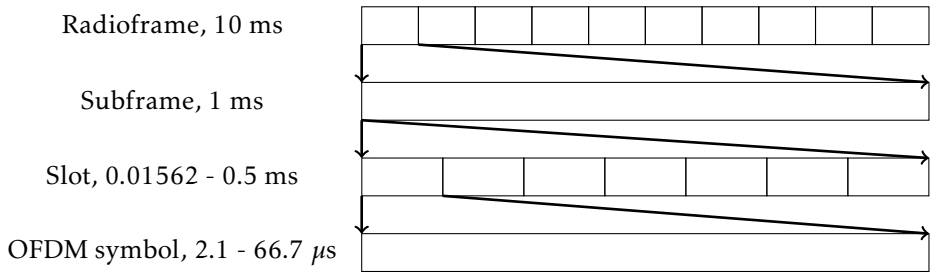


Figure 2.7: *5G frame structure.*

3GPP has decided that NR will support multiple numerologies in order to handle a wide range of frequency and deployment options. For normal cyclic prefix,

the numerologies are derived by scaling a basic subcarrier spacing

$$f_m = 2^m \cdot 15 \text{ kHz}, \tag{2.2}$$

where the base case, $m = 0$, corresponds to the subcarrier spacing used in LTE-A [7].

The duration of an OFDM symbol is inversely proportional to its subcarrier spacing. Therefore, changing the subcarrier spacing also changes the OFDM symbol duration. In NR, the number of OFDM symbols per slot will be kept equal between all numerologies. This means that the slot duration would shrink with increased subcarrier spacing [19]. The slot duration for normal cyclic prefix for numerology m is calculated as,

$$T_m = \frac{0.5}{2^m} \text{ ms}, \tag{2.3}$$

according to [7].

It has also been decided that scalable numerologies should allow subcarrier spacings from 15 kHz to 480 kHz, some of which are given in Table 2.2 [7]. Compared to the subframe duration of 1 ms in LTE-A, these numerologies can support a much lower latency.

Table 2.2: *Comparison of numerologies.*

	m = 0	m = 1	m = 2	m = 5
Subcarrier spacing	15 kHz	30 kHz	60 kHz	480 kHz
Slot duration	500 µs	250 µs	125 µs	15.62 µs

For URLLC, the subcarrier spacing has not been decided or recommended as of yet.

2.4 Segmentation

Should the link adaptation fail to select a format that transmits the data packet at target BLEP, there are three alternatives this thesis considers for the user. The first is to ask the scheduler for more resources and find a new format that hopefully can transmit all data at target BLEP. Another solution is to divide our data into smaller segments and transmit as much of the packet as possible at target BLEP. In the next TTI the user tries to transmit the rest, or at least another segment. The third solution is to not transmit anything in this TTI and hope for a better channel quality in the next TTI.

With greedy-filling, resources are allocated to a user until it succeeds with its link adaptation or the base station runs out of resources in a given TTI. If the base station runs out of resources in a TTI, the packet must be segmented or transmitted in the next TTI to try again. Not transmitting anything when there are available resources is a waste of resources but will also lead to less interference for users in neighboring cells.

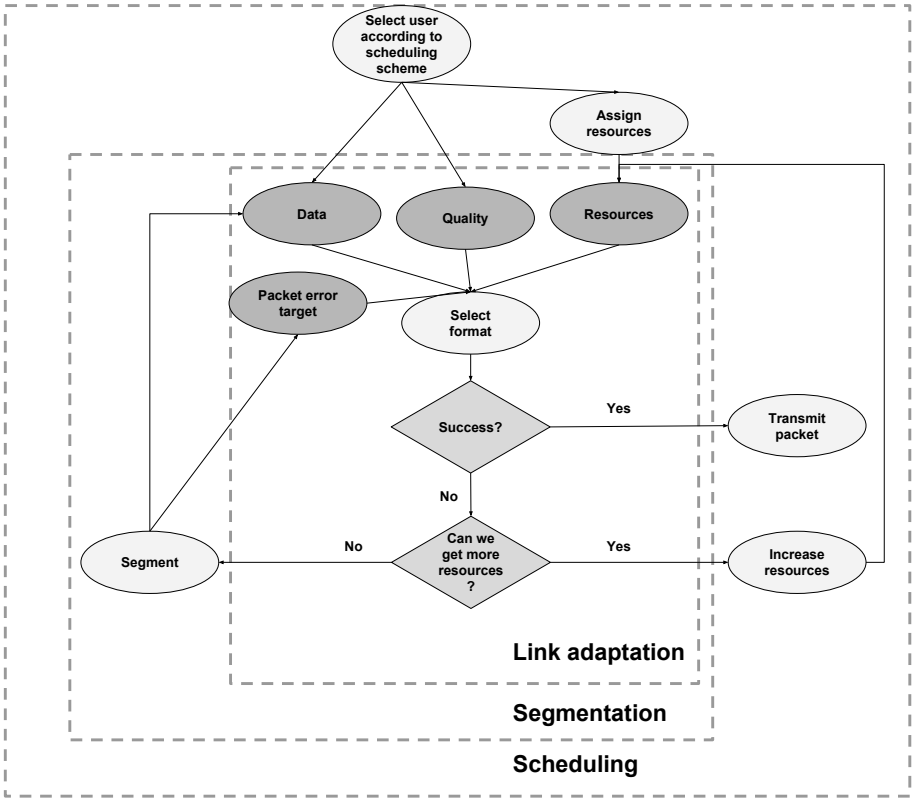


Figure 2.8: *Flowchart of scheduling and link adaptation, the dark gray ovals are variables that are changed by the processes in the light gray ovals. The diamonds represent processes that take decisions. The base station starts at the top of the figure when scheduling users.*

2.4.1 Segmentation in LTE-A and URLLC

LTE-A with greedy-filling assigns resources to the user until it succeeds or the base station is out of resources. When out of resources, the base station segments the UE's packet if needed. Each segment is sent at the target BLEP. This process is illustrated in Figure 2.8 to clarify the actions of the scheduler, link adaptation and segmentation.

In LTE-A the target BLEP is usually 10% since this has shown to generally yield a high throughput. 90% of the time the packets will succeed, and for the other 10%, HARQ can issue one or multiple retransmissions in order to retrieve the packet.

For URLLC, the packets should be delivered with a certain reliability which is generally higher than the reliability in LTE-A. Therefore, if the packet is split into multiple segments, each segment's target BLEP must be adjusted so that the BLEP

for the total packet is equal to the target BLEP. Since multiple segments are sent, each segment must be sent with an even lower BLEP since all segments must be received and decoded correctly in order for the packet to be decoded. Although segmentation requires higher reliability per segment, it is also easier to get that high reliability since there is less information in the segment.

2.4.2 Modeling Probability

In order to adjust each segment's target BLEP correctly, the probability of receiving the total packet based on the probability of receiving each segment must be modeled. This is simple if the probability of one segment being correctly received and decoded is independent of the other segments and there is an accurate model of the randomness. However it is not known if the segments can be regarded as independent. In this thesis it is assumed that the events are independent, which is a simplification. While a perfect model would be better, it might not be much better in this particular study since many other factors in the simulation might have a larger impact. First, the assumption of independent transmissions is explained and justified, thereafter the model of total packet error probability is presented.

Independent Transmissions

The use of independent transmissions is here motivated for a packet split into two segments. Let $P(A)$ denote the probability to receive and successfully decode segment A and $P(B)$ denote the probability to receive and successfully decode segment B. Furthermore, let the packet consist of two segments so that the probability $P(A \cap B)$ denotes the probability of receiving and successfully decoding the total packet. In order to find a bound on the probability of receiving and successfully decoding the total packet, rewrite it on the form

$$P(A \cap B) = P(B|A) \cdot P(A) \,. \tag{2.4}$$

The probability of receiving and successfully decoding both segment A and B is equal to the conditional probability of receiving and successfully decoding B given that A has been received and successfully decoded times the probability of receiving and successfully decoding A. Now consider the case that A and B are independent,

$$P(B|A) = P(B) \,. \tag{2.5}$$

Since the events are independent, the prior information that A has happened does not affect the probability of B. In the case of segmentation, one might argue that events A and B should be dependent. Should segment A be correctly received and successfully decoded, the channel quality is most probably good and will continue to be so during the next slot. At least it is more probable that the channel will continue to be good than to abruptly worsen in quality. It therefore makes sense that,

$$P(B|A) \geq P(B) \,. \tag{2.6}$$

Combining (2.4) and (2.6),

$$P(A \cap B) \geq P(A) \cdot P(B) \,, \tag{2.7}$$

we get a lower bound on the total probability. From (2.7) we see that using the same reliability for each segment as for the total packet would yield,

$$P(A \cap B) \geq (1 - 10^{-5}) \cdot (1 - 10^{-5}) \approx 0.99998 \,. \tag{2.8}$$

While, using the probability $\sqrt{1 - 10^{-5}}$ for each of the segments would yield,

$$P(A \cap B) \geq \sqrt{1 - 10^{-5}} \cdot \sqrt{1 - 10^{-5}} = 0.99999 \,. \tag{2.9}$$

From (2.8) it is clear that while this target reliability per segment might yield a large enough reliability, it can also be less than the targeted 0.99999. On the other hand, adjusting the segments' probability gives a reliability that will be at least 0.99999. The assumption of independent transmission of segments yields a lower bound on the probability of receiving and successfully decoding the entire packet. This is a simplification of the problem but will at least give a better approximation of the target reliability. Therefore this thesis assumes independent transmission of segments.

Another bound

The Fréchet inequalities give upper and lower bounds on the probability of a conjunction of events and is used here to argue for the modeling of error probabilities [20]. With these inequalities it can be shown that using the modeling of independent transmissions to calculate the target reliability for each segment, the lower bound will always be close to the targeted $1 - 10^{-5}$ while using the reliability $1 - 10^{-5}$ for each segment results in a lower bound below the targeted $1 - 10^{-5}$.

Let a packet be split into the segments $A_1, A_2, ...A_n$ so that $P(A_1)$ is the probability to receive and successfully decode segment number one and $P(A_1 \cap A_2 \cap ... \cap A_n)$ is the probability to receive and successfully decode the entire packet split into n segments. Then, the Fréchet inequalities for a conjunction of events is

$$\max(0, P(A_1) + P(A_2) + ... + P(A_n) - (n-1)) \leq P(A_1 \cap A_2 \cap ... \cap A_n), \text{and} \tag{2.10}$$

$$P(A_1 \cap A_2 \cap ... \cap A_n) \leq \min(P(A_1), P(A_2), ..., P(A_n)). \tag{2.11}$$

If $P(A_1) = P(A_2) = \cdots = P(A_n) = 1 - 10^{-5}$ is used for $n = 2$ and $n = 3$, the lower bound in (2.10) gets smaller than the required $1 - 10^{-5}$, as seen in Table 2.3.

On the other hand, if the target error reliabilities are chosen as $P(A_1) = P(A_2) = \cdots = P(A_n) = \sqrt[n]{1 - 10^{-5}}$, for $n = 2$ and $n = 3$, the lower bound is very close to the required 0.99999 as seen in Table 2.4.

Table 2.3: *Fréchet bounds on the resulting packet error probability when using* $P(A_1) = P(A_2) = \cdots = P(A_n) = 1 - 10^{-5}$ *for two and three segments.*

Number of Segments, n	Lower bound	Upper bound
2	0.99998000000	0.99999000000
3	0.99997000000	0.99999000000

Table 2.4: *Fréchet bounds on the resulting packet error probability when using* $P(A_1) = P(A_2) = \cdots = P(A_n) = \sqrt[n]{1 - 10^{-5}}$ *for two and three segments.*

Number of Segments, n	Lower bound	Upper bound
2	0.99998999998	0.99999499999
3	0.99998999997	0.99999666666

The modeling of independent transmissions might therefore be incorrect (the segments may depend on each other) but since the model yields a lower bound close to the targeted reliability it models it better than using the targeted reliability for each segment which yields a lower bound below the targeted reliability. By studying the bounds, other models that have a lower bound close to the targeted reliability and perhaps a lower upper bound, also closer to the targeted reliability, can be found. However, the model of independent transmissions was deemed good enough in this thesis.

Definition of Packet Error Probability

Having explained the assumption of independent transmissions, the assumption is used to define the probability of error for a packet with regards to the probability of error of each segment. Assume that a packet is segmented into M segments. Let $P_{\mathrm{e}_{\mathrm{tot}}}$ be the total probability of error for the packet and P_{e_i} be the probability of error for segment i. Assuming independent transmissions, the total reliability of a packet is

$$(1 - P_{\mathrm{e}_{\mathrm{tot}}}) = (1 - P_{\mathrm{e}_1}) \cdot (1 - P_{\mathrm{e}_2}) \cdot \ldots \cdot (1 - P_{\mathrm{e}_M}) \,. \tag{2.12}$$

Rearranging (2.12), the total packet error probability is

$$P_{\mathrm{e}_{\mathrm{tot}}} = 1 - (1 - P_{\mathrm{e}_1}) \cdot (1 - P_{\mathrm{e}_2}) \cdot \ldots \cdot (1 - P_{\mathrm{e}_M}) \,. \tag{2.13}$$

The notation presented here is used in Chapter 3 to describe how the different methods model the probability of error for each segment.

2.4.3 Previous Studies

Segmentation has been studied earlier, for older mobile networks. The primary goal has been to improve the resource efficiency by always transmitting data if there is data to transmit. If all data in a packet cannot be sent at target BLEP, only a segment is sent. In the next slot the rest of the packet can hopefully be sent.

If many users have data to transmit, this ensures that all resources are used to quickly transmit the data.

To the writer's knowledge there has not been any studies into segmentation where the segment's target BLEP is adjusted in order to meet the target BLEP of the resulting packet. In LTE-A, additional checks such as retransmission protocols make sure to correct errors after the transmission and ensure a much lower resulting BLEP than the one used in the transmission. However, all these checks introduce latency which must be kept low in URLLC. Therefore, in order to meet the stringent requirements on both latency and reliability, the reliability must be achieved already with the transmissions themselves, not relying on higher layers.

An alternative to segmenting the packet is to repeat the packet in several slots in order to meet the reliability. There are some proposals on such schemes in 3GPP but so far, none of them have been agreed to be used. One proposal is to use a rateless HARQ [21], or aggressive continuous transmission as it is called in another proposal [22]. The idea is to transmit retransmissions of the packet in each slot, not waiting for feedback before choosing to transmit. When an ACK is received, the retransmissions are stopped. The cost is possible resource waste since a retransmission is sent before it is known if it is needed.

3

Method

The goal of this thesis is to come up with guidelines for the implementation of segmentation in URLLC and get insights into what URLLC benefits from. This chapter describes what has been done in order to achieve this goal.

Methods were first designed, then implemented in a simulator, and finally evaluated through simulation. The initial results from the simulator were used to adjust errors in the methods' implementation and improve the methods. These improved methods were then run in longer simulations to get results about the methods' performance.

The simulations were run both in a single transmission and a retransmission scenario. In the single transmission scenario, no retransmissions are available. On the other hand, in the retransmission scenario, the base station can afford one retransmission. The retransmission can be afforded due to fast HARQ, a special variant of HARQ. For fast HARQ, in order to transmit feedback faster, some resources that could otherwise be used for DL transmission are used for UL transmission of acknowledgment.

In this chapter, the simulator and the chosen scenario for the thesis is described. Thereafter, the system model of the network used is presented. Finally, the proposed methods are presented for both single transmission and retransmission.

3.1 Simulations

An internal simulator at Ericsson Research was used for the simulations. The simulator can simulate a complete LTE-A network and contains additions for 5G, or at least what Ericsson believe 5G will contain.

3.1.1 Scenario

The scenario used in the thesis is suggested by 3GPP for evaluating URLLC [6, Ch.13,Sec.2], but the scenario is used with some modifications. One of the modifications is a smaller cell radius in order to achieve a reasonably high SINR to all users.

Using a larger number of antennas, especially at the base station, would increase the SINR to the users and enable the use of a larger cell radius. In the scenario suggested by 3GPP, it is stated that up to 256 antenna elements at the base station and eight at the user can be used [6, p. 57]. The antenna mapping in the simulations in this thesis maps one antenna element to one antenna port. However, an increase in number of antenna elements does also increase the simulation time which already is long. Due to the very low target reliability, in order for a number of error events to occur the number of packets simulated becomes very large. Therefore, only 16 antenna elements are used at the base station in this thesis, and two antenna elements at the user. In order to get reliable statistics, a lower target reliability of $1 - 10^{-3}$ was also used in the beginning. This enables shorter simulation times to see the effects of the methods and potential bugs and errors. Thereafter, the target reliability of $1 - 10^{-4}$ was used.

Another modification of the 3GPP scenario is that no EMBB traffic is run simultaneously since the focus is on studying the segmentation of URLLC traffic and not the multiplexing between URLLC and EMBB. Also, only a small network of one site with three sectors, as illustrated in Figure 3.1, was simulated. In [6, Sec.A.2.4], it is suggested to use 57 sectors to simulate a network.

The sectors in the site are scheduled by the same base station, however the schedules for the sectors are not coordinated. Instead, the sectors are scheduled independently and cause interference to each other. To schedule sectors served by the same base station in a coordinated manner is possible, but to coordinate schedules between different base stations is hard. By scheduling the sectors uncoordinatedly, a smaller network can be used and the network experiences a similar effect as to having a larger network with uncoordinated base stations. The users in the scenario move within a sector but the users never change sector.

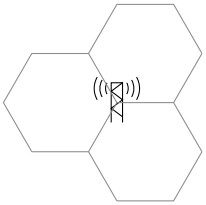


Figure 3.1: *Cell layout in scenario.*

The scenario is an Urban Macro scenario where 80% of the users are considered to be indoors and moving at a speed of 3 km/h while the other 20% are outdoors, moving at 30 km/h. The traffic is unidirectional, packets are only sent in one direction, in this case the DL direction. The packets arrive periodically for each user, but each user starts its packet arrival at different times. The packet

size is 50 bytes. For carrier frequency, 4 GHz is used, as recommended by 3GPP. The parameters used in the scenario are summarized in Table 3.1.

Table 3.1: *Simulation assumptions for URLLC*

Name of parameter	**Value**
Layout	Single layer, hexagonal grid
Carrier frequency	4 GHz
Simulation bandwidth	20 MHz per component carrier. For FDD, simulation bandwidth is split equally between UL and DL.
Duplex	FDD
Direction	DL
Cell radius	120 m
User distribution	Randomly and uniformly distributed over sector at least 35 meters from base station. 20% outdoor in cars at 30 km/h, 80% indoor at 3 km/h.
Traffic model	Periodic arrival
No. base station antenna ports	16
No. user antenna ports	2
Packet size	50 bytes
Channel model	3D-Uma [23]

Load

To study the URLLC capacity and resource efficiency of the methods, the load was varied by adjusting the periodicity of the packet arrival. For a larger period, the time between packets is longer and therefore the load is lower.

For different target reliabilities, different periodicity was needed to find a fitting load. A fitting load is one where the methods barely meet the latency and reliability demands. Slight deviations from this load can then be used to see which methods fail first and why the methods fail.

Simulation Time

For the target error probability of 10^{-3}, the simulation time was set so that for the longest simulated period, 100 error events per user would occur during simulation. Each simulation was run for a couple of different periods of packet arrival but each simulation was run an equally long simulation time for the different periods. By ensuring 100 error events for the longest period of packet arrival, there were even more error events for the shorter periods of packet arrival.

For the target error probability of 10^{-4}, only 10 error events per user were simulated. A longer simulation time would be better, but at the end of the thesis there was not enough time for very long simulations.

Let N_e be the number of occurred error events, T_{sim} be the duration of the simulation in seconds and T_{packet} the time period between packet arrivals in seconds. The number of occurred error events is roughly the number of packets arriving per second, times the number of seconds the simulation is run, times the probability of error. With this, the simulation time can be calculated as

$$N_e = \frac{T_{sim} P_{e_{tot}}}{T_{packet}} \Leftrightarrow T_{sim} = \frac{N_e T_{packet}}{P_{e_{tot}}}. \tag{3.1}$$

In addition to the simulation time and number of users, each simulation was also run for 10 different random seeds to get a more reliable result. The simulator uses the random seed as input to its pseudo-random number generator which is used for example to randomize the time instant a user starts sending packets and the position of the UE.

3.1.2 System Model

This section will briefly describe the system model by describing how different LTE-A mechanisms are implemented for the simulations. The model is summarized in Table 3.2.

Table 3.2: *Summary of system model*

Name of mechanism	**Type**
Scheduling scheme	Delay- and segment-prioritized, Greedy-filling
Outer loop	Turned off, SINR history and back-off used instead
Retransmission protocol	Turned off or fast HARQ using IR, non-adaptive
Numerology	30 kHz subcarrier spacing, 0.25 ms slot duration
Modulation scheme	QPSK
Control channel	Ideal
FEC	Convolutional

Scheduling Scheme

For scheduling, a delay- and segment-prioritized scheduling scheme is used instead of the proportional-fair scheme used in LTE-A. The user with most segments is scheduled first. If no user has segments, the user with the oldest packet is scheduled first. In the case of a tie between number of segments, the user with the oldest packet goes first. In the case of a tie between age of packet, the user with the highest quality is scheduled first.

In this way, old packets which are in need of scheduling are scheduled and if a packet has been segmented it is also prioritized. This is natural since if one segment is sent, and not the others, the first one was a waste of resources, and such cases should be avoided. In a tie due to delay, a user with higher channel quality can hopefully be scheduled with a fraction of the total resources in the

first slot. The other packet might need many resources in several slots in order to get its packet over and therefore it makes sense to let the higher quality packet pass first.

The scheduler also uses a greedy-filling approach, assigning resources to one user until the user can transmit its packet before assigning resources to the next.

The existing delay-prioritizing scheduler in the simulator was modified to prioritize segments and use the users' quality in the case of a tie on delay- and segment-priority.

Outer Loop

The outer loop in the link adaptation in LTE-A is not used. Since the base station gets very few NAKs it would take a very long time until the loop converges. In addition, some of the methods change the target error probability for their segments. This would require a more complex outer loop that is aware of these changes and can adapt to them quickly.

Instead, a constant back-off and a history of SINR values is used. The positive back-off is subtracted from the SINR derived from the CQI in order to get a pessimistic channel quality estimate. This subtraction is performed in dB-scale as both the back-off and the measured SINR value are measured in dB. The resulting pessimistic channel quality estimate is stored in a history of SINR. Each UE stores its own SINR values and then the worst SINR in its history is used for link adaptation. The number of SINRs to store can be adjusted and is called history-size, or N_{SINR}. The back-off is measured in dB and is denoted by Δ_{SINR}. This method is needed in order to achieve our strict demands but it leads to a waste of resources.

Figure 3.2 illustrates a CDF of the DL SINR for a typical UE in the simulations. The UE in Figure 3.2 experiences an SINR that is 5 dB or lower in roughly 10% of its transmissions.

Should the SINR from the CQI report be 5 dB but the actual SINR experienced by the UE be higher, located to the right of 5 dB in Figure 3.2, the UE gets a pessimistic channel quality estimate. The quality is better than what the UE believes it to be and so more resources than necessary will be spent. However, should the actual SINR be lower than the reported SINR, located to the left of 5 dB in Figure 3.2, the UE believes the channel quality to be better than what it is. This leads to an incorrect estimate of the probability of error for the packet and can thus lead to the UE not segmenting at all even though it would need to in order to meet the required probability of error.

In order to reach a high URLLC capacity, it is better to waste resources than miss packets. However, a too large Δ_{SINR} or N_{SINR} can give rise to other phenomena. For example, if the situation always is better than what is reported, methods that do not try to follow the target error probability but takes more chances will be able to support more UEs and will most of the time also meet the demands. Therefore, these parameters have to be tuned to each scenario.

The parameters Δ_{SINR} and N_{SINR} are especially important in the single transmission scenario. In the retransmission scenario, should a packet fail due to misinformation, such as when a UE's estimate describes a much better channel

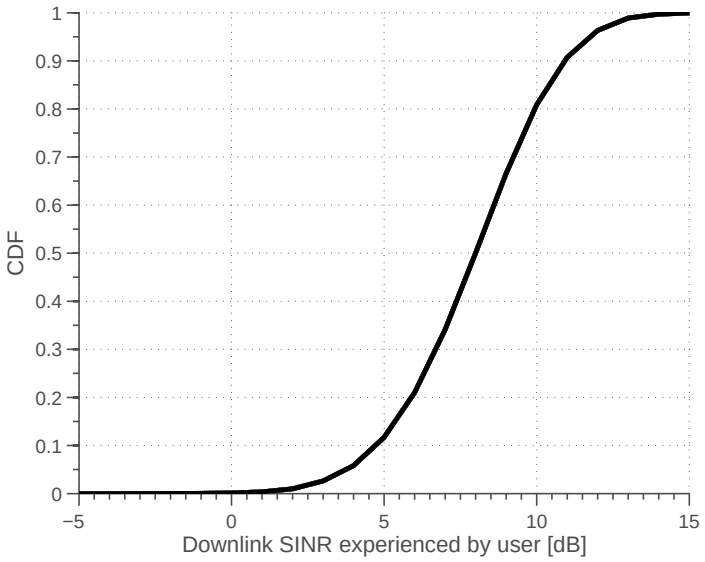


Figure 3.2: *DL SINR for a UE in the network.*

quality than the actual channel quality, a retransmission can often be enough to deliver it. However, should the base station split the packet into multiple segments and allocate too few resources for multiple segments, there is not enough time to retransmit all segments within the latency bound.

In addition, the value of N_{SINR} might need to be adjusted to the periodicity of the arriving packets. When packets are arriving often, a smaller N_{SINR} might be better since this means only the more recent subframes are used to select a worst SINR. With longer period between the packets, a larger N_{SINR} is needed to at least cover a couple of sent packets and their SINRs.

It is outside the scope of this thesis to find optimal settings for all these parameters but some values for them have been tested and are presented in Chapters 4 and 5. Shorter simulations were run for a number of different setups, the ones that could deliver packets within the latency and reliability demands for 80-100% of the users were then run in longer simulations.

Numerology

As described in Section 2.3.4, different numerologies will be supported by 5G. In the simulations, the numerology with a subcarrier spacing of 30 kHz and a slot duration of 0.25 ms was used. This gives us 4 slots within the latency bound.

HARQ

In the single transmission scenario, no HARQ is used. This is since there only is 4 slots within the latency bound while HARQ transmits a retransmission 8 slots after the initial transmission. An alternative is the fast HARQ which is not part of any standard yet, but a proposal. In fast HARQ, resources are spent on decoding the packet and transmitting the feedback earlier resulting in a retransmission two slots after the original transmission. Denoting each slot with a box, the fast HARQ is illustrated in Figure 3.3.

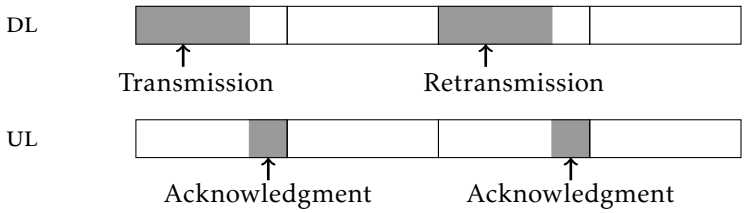


Figure 3.3: *Fast HARQ.*

Modulation Scheme

In the simulations, only the modulation scheme QPSK is used. This is to limit the number of parameters that affect the simulations and focus on the result of the different methods. QPSK is the most reliable modulation scheme of the available modulation schemes in LTE-A but offers a low data rate, thus it seems reasonable to use for URLLC where reliability is most important.

Control Channel

The control channel for both UL and DL is assumed to be ideal. The acknowledgments sent from the UE, as well as the CQI reports are thus always received and decoded correctly. Note that the CQI reports' actual content still is an estimated SINR. A non-reliable control channel would be interesting to study but was deemed outside the scope of this thesis.

Target Reliability

This thesis often refers to a packet not achieving the target reliability. Remember that this is only an estimate. With the help of the channel estimate and a coding model in the base station, the base station can derive a probability of correct reception of this packet using the current format and resources. This probability is seen as the packets reliability. At decoding however, it is not certain that the probability is the same — the channel varies with time and the channel estimate at the base station is outdated. In addition, interference from other UEs might be hard to predict. Therefore, a packet achieving target reliability in link adaptation might not do so at decoding, the situation can be better or worse than anticipated.

Code Rate

The code rate is the number of information bits divided by the total number of bits in a packet. The code rate describes how many information bits are sent per transmitted bit. This parameter is used by the coding model to estimate the error probability for the packet. When using HARQ with IR, the retransmission is combined with the original transmission, usually adding new parity bits and thus lowering the code rate. A lower code rate means a lower probability of error. Therefore, in order for the resulting probability of error to be 10^{-5}, the code rate for the initial transmission should take the number of retransmissions into account by calculating its predicted error probability using the total bits from both the initial transmission and the retransmission(s). This leads to a higher code rate and probability of error for the first transmission, however with the retransmissions the resulting probability of error is lowered.

Coding

The FEC code used in the simulations is a convolutional code. 3GPP has not decided on which code to use for URLLC in NR as of yet, but convolutional coding is a good candidate. For example, convolutional codes do not experience an error floor, which Turbo codes that are used in LTE-A may experience for certain configurations. Plotting achieved BLEP versus SINR for FEC codes, the BLEP decreases with increasing SINR. An error floor is when there is a point in SINR after which the BLEP curve does not fall as quickly as before. This makes codes that experience an error floor less efficient when the BLEP should reach very low levels [3].

In addition, convolutional code decoding has a shorter delay than the iterative decoder typically used for Turbo decoding. This is due both to the lower decoding complexity of the convolutional decoder and to the fact that a convolutional decoder can decode the data while it is being received [3].

3.1.3 Evaluation

URLLC capacity and resource efficiency are used to evaluate the suggested methods. URLLC capacity was defined in Definition 2.3 as the maximum offered cell load under which Y% of users in a cell operate with target link reliability R under latency bound L. A UE is considered successful if the probability to receive a packet within the latency bound, L, is larger than or equal to the target reliability R. This probability can be read out from a CDF of latency for all packets for all seeds for one user. Y is the number of successful UEs divided by the total number of UEs. By plotting Y for different offered cell loads and methods, the URLLC capacity for each method for a certain value of Y can be read out from the plot.

The exact value of the URLLC capacity is not the most interesting in this study, instead it is which method achieves the highest URLLC capacity that is interesting. Often it is enough to compare the value of Y, the percentage of successful users, between methods. Consider two methods simulated for a moderate load where the first method achieves the requirements for 96% of its users and the second

method only achieves the requirements for 90% of its users. Then, the URLLC capacity for $Y = 96$ would be the simulated moderate load for the first method while the second method must offer a lower load (a lower URLLC capacity) to increase the number of successful users. This holds as long as the number of successful users does decrease for a higher load, which most often is the case.

A high value of Y, perhaps $Y \geq 90$ is interesting since URLLC should support a high availability, the URLLC service should be available to the users at almost all times.

In this thesis, a user is counted as successful only if the number of packets delivered within the latency bound can meet the reliability demand when averaged over all seeds. The simulation is run for different 10 seeds and then the result of all seeds are summarized into one result so that the number of packets received within the latency bound is averaged across all 10 seeds. Since the seed sets the position of the UE, this means that the user has to be able to deliver most of its packets from a number of different positions and therefore with a number of different channel qualities.

Another way to measure the URLLC capacity would be to regard each seed as a realization of a UE and count also them as UEs. Then, instead of 75 users averaged over 10 seeds there would be 750 users. This might yield a higher URLLC capacity since a few bad seeds will count only as a few bad seeds and not drag down the average of a user.

3.2 Timing

To study segmentation for URLLC this thesis proposes a couple of methods and evaluates them both in single transmission and when there is time for one retransmission. This section describes some of the effects of timing in these two cases.

3.2.1 Single Transmission

In the single transmission scenario, the user cannot wait for an acknowledgment and has to make do with one transmission. If the user cannot increase the number of used resources for its transmission further and still cannot transmit its whole packet while achieving the target reliability, the user must segment its packet or wait for the next slot. However, with a slot duration of 0.25 ms, only four slots fit within the latency.

With a target latency of 1 ms, the packet must be delivered within that time. The latency is measured from end to end, from the time the base station has the data packet until the UE has decoded the packet. An overview of this is given in Figure 3.4.

Since there always is some scheduling latency, at most three slots can be used within the latency bound. Should the base station be loaded heavily, a user might not even be scheduled in the same slot its data arrived, increasing the scheduling latency to slightly more than one slot. In this case, only two slots are left to segment into.

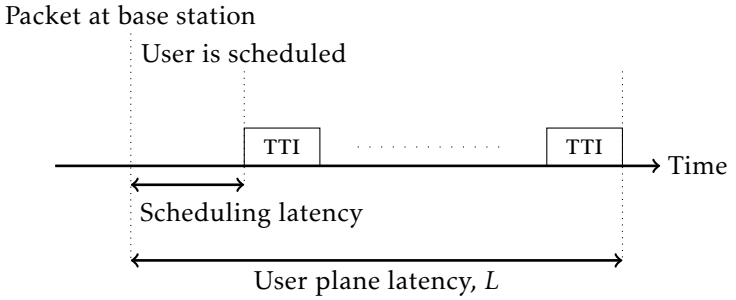


Figure 3.4: *Overview of time.*

3.2.2 Retransmission

Section 3.1.2 describes fast HARQ and how this gives the base station the possibility to transmit a retransmission based on the UE's acknowledgment at the cost of fewer resources to transmit in. Some methods benefit more than others from a retransmission. Consider segmenting a packet into three slots, then the only retransmission opportunity in the third slot is already occupied by another segment. A packet segmented into two slots can afford losing the first segment and retransmitting it in the third slot, but there is not enough time to retransmit the second segment. In addition, new methods are possible when retransmissions are available. Therefore, this thesis suggests some additional methods for the retransmission scenario.

Retransmissions are also very useful when sending whole packets. If the channel estimate shows a much better quality than what the channel actually has, too few resources will be assigned to the transmission and the whole packet is sent at once, probably resulting in a lost packet. This can be recovered by sending a NAK and issuing a retransmission.

All proposed methods calculate the predicted error probability for a transmission from the resulting code rate, after possible retransmissions, as described in Subsection 3.1.2. However, for the methods that split a packet into multiple segments, the code rate is calculated as in single transmission for these segments. This is since all segments cannot be retransmitted, there is only time for one retransmission.

3.3 Proposed Methods

This thesis considers five methods to evaluate segmentation for URLLC in single transmission. The methods are called: Baseline, Two is Enough, Estimated, Never, and Forced. These methods, except for Baseline, were designed in collaboration with the supervisors for this thesis. For retransmission, these methods are evaluated together with two additional methods: Delayed forced and Dare. The descriptions of the methods will first describe the methods in single transmission and then briefly describe how they act when there is a retransmission.

3.3.1 Baseline

This method was the one implemented in the simulator at the beginning of the thesis. It is very simple and similar to how segmentation works in LTE-A. When a user cannot send its whole packet at target reliability, the user sends as much as possible of its packet at this target reliability. In the next TTI, the user sends as much as possible, again at target reliability

$$P_{\mathrm{e}_i} = P_{\mathrm{e}_{\mathrm{tot}}} \,. \tag{3.2}$$

Baseline works but uses the wrong target error probability for its segments and has no control over how many segments it sends. This method is used only as a comparison.

Baseline with Retransmission

As described in Section 3.2.2, if the packet is segmented into two segments the method can benefit from the retransmission. Since this method has no control of the number of segments, but will in most cases use a small number of segments, it will benefit from retransmission.

3.3.2 Two is Enough

In Two is Enough, if the user must segment its packet, it is assumed that two segments is enough. The two segments use adjusted error probabilities so that the total error probability is correct if two segments is enough. Should the user need more segments, these are sent at the packet target error probability

$$\begin{aligned} P_{\mathrm{e}_1} = P_{\mathrm{e}_2} &= 1 - \sqrt{1 - P_{\mathrm{e}_{\mathrm{tot}}}} \,, \\ P_{\mathrm{e}_3} = \ldots = P_{\mathrm{e}_M} &= P_{\mathrm{e}_{\mathrm{tot}}} \,. \end{aligned} \tag{3.3}$$

Two is Enough uses a slightly more correct target error probability than Baseline and should therefore perform slightly better. It is a simple, controlled segmentation that might cover most cases. When two segments actually is enough, this method uses the correct error probability.

Two is Enough with Retransmission

Two is enough assumes it can deliver the whole packet in two segments and will in those cases benefit from the retransmission, however just as Baseline it does not control the actual number of used segments.

3.3.3 Estimated

This method takes all segments into account when adapting the target error probability for its segments. The idea is to estimate the number of needed segments and adjust the size and target probability of each at every slot. Estimation is very

simple — assume that the quality in the next slot is at least as good as the quality in the current slot. This makes sense since the scheduler prioritizes users with segmented packets, thus assigning them more resources in upcoming slots. The user will not get more resources only if it is already using all of the base station's resources or another user has more segments. However, the situation can worsen due to interference from other users as well.

In addition to estimating the number of segments, Estimated keeps track of the number of used segments as to not overstep the latency budget. The method must strike a balance between using correct error probability and transmitting at wrong error probability in order to make it in time. After all, a packet sent at a higher error probability has a higher probability of being received than one not sent.

Structure of Method

The method is divided into an outer loop and an inner loop. The inner loop performs the actual estimation of the number of segments to be used. The outer loop is illustrated in Figure 3.5 and is used to keep track of if the current slot is the last slot and if there is need for segmentation. If the user can transmit all data in its buffer it should simply transmit all data in its buffer. The buffer either contains a whole packet, or the rest of the packet, if the user has started to segment the packet. In the outer loop, the user first checks if this is the last slot it can transmit in, in that case it transmits all data in its buffer and allows a higher probability of error for that transmission. This is in order to be able to transmit the last segment of a packet and not let the earlier transmitted segments go to waste. Thereafter, it tries to send all in its buffer at once. If that is not possible, it uses the inner loop to estimate how many segments it will need.

Target probability of error

As described in Section 2.4.2, due to the assumption of independent transmission of segments, the total reliability of a packet is expressed as

$$(1 - P_{\mathrm{e}_{\mathrm{tot}}}) = (1 - P_{\mathrm{e}_1}) \cdot (1 - P_{\mathrm{e}_2}) \cdot \ldots \cdot (1 - P_{\mathrm{e}_M}) \,. \tag{3.4}$$

To calculate the target error probability for a segment, the number of segments N is fixed and it is assumed that each segment will use the same target error probability

$$(1 - P_{\mathrm{e}_{\mathrm{tot}}}) = (1 - P_{\mathrm{e}_i})^N \Rightarrow P_{\mathrm{e}_i} = 1 - \sqrt[N]{1 - P_{\mathrm{e}_{\mathrm{tot}}}} \,. \tag{3.5}$$

(3.5) can be seen in Figure 3.6 and is used to calculate the target probability of error for a segment, first for $N = 2$, then $N = 3$, and so on. When estimating the initial number of segments, $P_{\mathrm{e}_{\mathrm{target}}}$ in Figure 3.6 is equal to the target error probability of the whole packet, $P_{\mathrm{e}_{\mathrm{tot}}}$.

In the upcoming slots, the number of segments needed to transmit the data left in the buffer is estimated. The idea is to estimate the number of needed segments in each slot, thus updating the estimate and improving its performance.

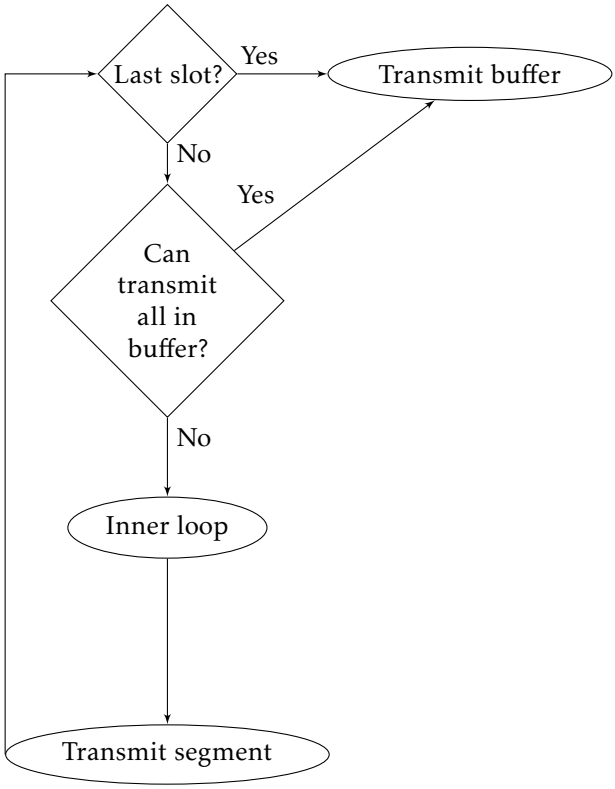


Figure 3.5: *Outer loop of Estimated.*

Even though the first slot estimated that three segments would be needed, the second slot might have received more resources and can transmit all in its buffer. In this case, the user should transmit all in its buffer since it is beneficial to make use of the available resources. It could also happen that the first slot estimates that two slots would be enough but due to interference, the quality in slot two has worsened and three segments are needed in total. By estimating the number of needed segments in each slot, the estimate of the number of segments improves.

In order to calculate the error probability to use for the segments in upcoming slots, the remaining error probability is first calculated. The remaining error probability is the target error probability of the data in the buffer given what has been transmitted and the total error probability for the packet. With knowledge of the earlier segment's used probability of error and the targeted error probability for the resulting packet, the remaining error probability can be calculated as

$$P_{\mathrm{e_M}} = 1 - \frac{1 - P_{\mathrm{e_{tot}}}}{(1 - P_{\mathrm{e}_1}) \cdot (1 - P_{\mathrm{e}_2}) \dots (1 - P_{\mathrm{e}_{M-1}})}\,, \tag{3.6}$$

where M is the current segment, 1 to $M-1$ are the earlier transmitted segments, and $P_{\mathrm{e}_1}, P_{\mathrm{e}_2}, \dots P_{\mathrm{e}_{M-1}}$ are the error probabilities that have been used for segments

1 to $M-1$. In order for the total probability of error to be correct, (3.4) must hold. Therefore, should segment M be split into more segments, the probability of receiving all sub-segments of M must be equal to the probability of receiving M. Let $P_{e_{Mi}}$ be the probability to receive sub-segment i of segment M, and N be the number of sub-segments segment M is split into. Then the probability of receiving M is

$$(1 - P_{e_M}) = (1 - P_{e_{Mi}})^N \Rightarrow P_{e_{Mi}} = 1 - \sqrt[N]{1 - P_{e_M}}\,. \tag{3.7}$$

(3.7) can be seen in Figure 3.6 and is used to calculate the target probability of error for a segment when segments of the packet have been transmitted earlier. When estimating the upcoming number of segments, $P_{e_{target}}$ in Figure 3.6 is equal to the remaining error probability, P_{e_M}.

Inner loop

The inner loop estimates the number of segments to use for transmission, which decides the target error probability and desired size of the segment. Two segments are enough if half of the packet can be transmitted at the error probability given in (3.3), for example. When a user has sent a segment and returns to the inner loop to select a format for the next segment, the inner loop is used to estimate the number of segments needed to transmit the rest of the packet.

To estimate the number of needed segments, the inner loop starts with the guess that two segments will be enough. The outer loop has already tried to send everything in the buffer and it is now up to the inner loop to estimate how many segments that will be needed. Should two segments not get at least half of the packet over at correct error probability, three segments and one third of the packet is tested. Thereafter four, and so on. The number of segments tested is increased until it would overstep the latency budget.

In order to more aggressively make use of each slot, a factor called fraction-factor, denoted by k is used. Instead of at least one half of the packet (in the case of two segments) as the limit on a successful segment, k times one half of the packet is used. Through simulation it was shown that $k = 0.85$ worked well. This means that as long as 85% of half of the packet is transmitted, it is a valid segment. In order to succeed with the total packet, the user must get more resources in the upcoming slots.

Let N denote the number of segments tested and $N_{\max}$ be the total number of segments allowed within the latency bound. The error probability, P_{e_i} denotes the target error probability of segment i and $P_{e_{target}}$ denotes the current target error probability of the data in the buffer. The size of the current segment is denoted by S_i and the size of the data in the buffer is S_{tot}. With this notation, the inner loop is illustrated in Figure 3.6.

Note the lower right bubble in Figure 3.6. This bubble ensures that the user always transmits something, even though it is less than the desired $k\frac{1}{N}S_{tot}$ bits. It is only allowed to transmit less than $k\frac{1}{N}S_{tot}$ bits when all possible N up to $N_{\max}$ have been exhausted. The target error probability used is the one derived

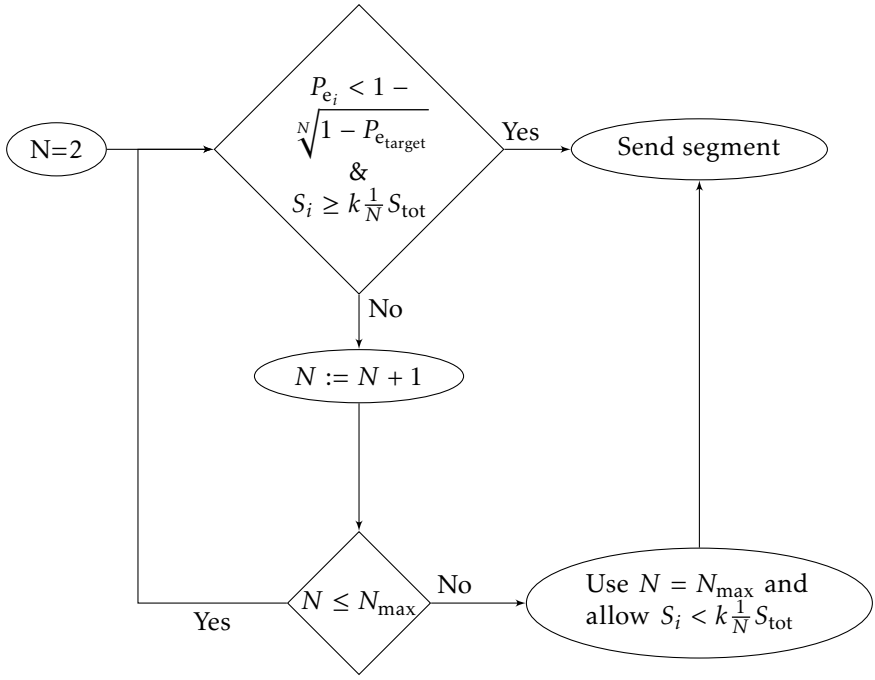


Figure 3.6: *Inner loop of Estimated.*

with N_{max}. Without the check in this bubble, the method might skip a slot, not transmitting anything in it and hoping to transmit more in the upcoming slot. To always transmit something, no matter how small, maximizes the resource usage. It also helps the method to achieve the latency, making use of each slot available. However, it does rely on the fact that more resources will be available in the next slot in order to transmit the fraction of $\frac{1}{N}S_{tot}$ that was not possible to transmit in the last slot in addition to the next $\frac{1}{N}S_{tot}$ fraction.

Another alternative would be to only transmit if more than $\frac{1}{N}$ of the data in the buffer can be sent. That would minimize the number of wasted segments, should the amount of resources not be increased and the packet be lost. However, this might decrease the total number of delivered packets since many segments that do not achieve $\frac{1}{N}$ of the data in its buffer do get increased resources due to the scheduling scheme.

A tradeoff has to be made and in the suggested method it has been chosen to always transmit. In a scenario with more slots, due to a higher latency or a different numerology the other option might be preferred.

Example

To illustrate how the error probabilities are calculated, this section describes an example that is illustrated in Table 3.3. A user has a packet it wants to transmit

but cannot ask for more resources and cannot transmit all of it at target error probability. It uses the numerology of 30 kHz and has three slots during which it can transmit its packet. In the first slot, $N = 2$ is tested which gives an error probability of: $P_{e_1} = 1 - \sqrt{1 - 10^{-5}} = 0.5 \cdot 10^{-5}$, for the segment. A format is chosen and checked against the coding model. At least half of the packet times k can be sent at error probability $0.5 \cdot 10^{-5}$ according to the coding model and therefore the format is used for transmission and the used error probability noted.

In slot number 2, the remaining error probability is $P_{e_M} = 1 - \frac{1-10^{-5}}{(1-0.5 \cdot 10^{-5})} = 0.5 \cdot 10^{-5}$. A format is chosen but it cannot deliver all that is left in the buffer at error probability P_{e_M} according to the coding model, in this case. In the first slot we estimated that this would be the last slot, however the situation has worsened. Since three slots can fit within the latency bound and one slot has already been used, $N = 2$ can be tested as well. With $N = 2$ and $P_{e_M} = 0.5 \cdot 10^{-5}$, the error probability is: $P_{e_2} = 1 - \sqrt{1 - 0.5 \cdot 10^{-5}} = 0.25 \cdot 10^{-5}$. A format for this error probability is chosen and delivers more than half of what is left in the buffer times k according to the coding model. Therefore, the format is used for transmission.

Thereafter, time moves forward to slot number 3, the last slot within the latency. The remaining error probability is $P_{e_M} = 1 - \frac{1-10^{-5}}{(1-0.5 \cdot 10^{-5}) \cdot (1-0.25 \cdot 10^{-5})} = 0.25 \cdot 10^{-5}$. A format is chosen and can deliver all that is left in the buffer at the error probability P_{e_M} so it is used for transmission. The resulting reliability for the packet that successfully transmitted in all three slots is: $(1 - P_{e_{tot}}) = (1 - P_{e_1}) \cdot (1 - P_{e_2}) \cdot (1 - P_{e_3}) = (1 - 0.5 \cdot 10^{-5}) \cdot (1 - 0.25 \cdot 10^{-5}) \cdot (1 - 0.25 \cdot 10^{-5}) \approx 0.99999$. Thus, the packet was segmented and achieved both the latency and reliability demands, in theory. If the format cannot meet the error probability requirement, the format that gives the highest reliability would be chosen and transmitted anyway since this is the last slot.

Table 3.3: *An example of used error probabilities for the method Estimated*

Slot 1, $P_{e_{target}} = 10^{-5}$	N	P_{e_1}	P_{e_i} and S_i ok?
	1	10^{-5}	No
	2	$0.5 \cdot 10^{-5}$	Yes
Slot 2, $P_{e_{target}} = 0.5 \cdot 10^{-5}$	N	P_{e_2}	
	1	$0.5 \cdot 10^{-5}$	No
	2	$0.25 \cdot 10^{-5}$	Yes
Slot 3, $P_{e_{target}} = 0.25 \cdot 10^{-5}$	N	P_{e_3}	
	1	$0.25 \cdot 10^{-5}$	Yes

Summary

Estimated uses a more correct target error probability and controls the number of segments with regards to the latency. This method should be an improvement to both Baseline and Two is Enough.

Estimated with Retransmission

Estimated segments its packet into multiple segments and behaves similar to Baseline and Two is Enough in retransmission. In some cases it gets extra reliability from the possible retransmission, and in others it does not.

Code Rate

As long as Estimated does not segment, the code rate for calculation of expected BLEP is calculated with respect to the number of retransmissions available in the retransmission scenario. However, if the packet is split into multiple segments, the code rate of each segment is calculated as if there are no available retransmissions for that segment. This is a simplification since if the packet is split into two segments, the first segment could make use of a retransmission and therefore should adjust the code rate for calculation of expected BLEP accordingly. But, Estimated can also change the number of segments used should the quality of the channel get worse, changing its estimate from two segments to three. If the first segment assumed a retransmission was available and adjusted its code rate for calculation of expected BLEP, but a retransmission is not available anymore since a third segment must be transmitted in the third slot, the first segment will use an incorrect target error probability. Therefore, the code rate calculation only assumes available retransmissions as long as the packet is not split into segments. This is a simplification but in order for Estimated to be able to adjust the number of segments, it would be difficult to do it in another way.

3.3.4 Never

This method never performs segmentation. If the packet does not meet the target reliability and there are no more resources, the packet is sent anyway as long as the expected BLEP is not 1. The coding model returns an expected BLEP of 1 if the decoder would not be able to decode a packet with the allocated resources and estimated quality, for example when the allocated resources and modulation yields fewer bits to transmit than the number of information bits (this would result in a coderate higher than one). The idea is that this is a worst-case scenario to compare with to see if segmentation gives any improvement. Note that a user has a few slots before the latency bound, therefore Never can try to send the whole packet in each slot hoping that the quality will improve.

In this method, if Never cannot ask for more resources and cannot achieve the target reliability, as long as it achieves a BLEP lower than 1, it transmits its packet. If the BLEP of the packet would be 1, the method will try again in the next slot.

Never with Retransmission

Never always sends the whole packet and thus always benefits from the possible retransmission. However, there might not be enough resources to retransmit all lost packets. If many users' packets are not received since Never does not segment, there might not be enough resources to send a retransmission for each lost

packet. In addition, if a user suffers from a very bad quality, one retransmission might not be enough.

3.3.5 Forced

The method Forced does not segment a packet, instead it forces a retransmission directly after the transmission, as illustrated in Figure 3.7. This can be thought of as a form of repetition code, if using Chase combining. However, this method uses IR to add extra protection. If both packets are received, they are combined in the decoder to increase the probability of correct decoding.

The retransmission uses as many subbands as the original transmission, as long as the expected BLEP from the coding model is not one. In that case, the subbands are increased. In other words, the retransmission is a copy of the first transmission if the copy is expected to be possible to decode, also with a very low probability of correct decoding. Therefore, the retransmission might not either fulfill the expected BLEP, but combined in the UE the two transmissions might be enough to receive and decode the packet correctly. Often the expected BLEP is lower than the actual BLEP.

The method is named Forced since it forces a retransmission directly after transmission of the packet, before the actual acknowledgment is received. In this way, the method makes use of the very useful HARQ protocol and still meets the latency bound. This is not segmentation in the sense that it segments the packet but it might work as well or even better.

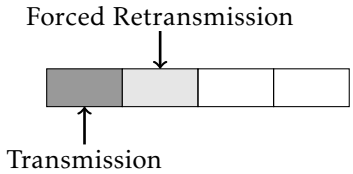


Figure 3.7: *The method Forced forces a retransmission directly after its initial transmission.*

Code Rate

Since Forced effectively gets a retransmission when it forces a retransmission, the code rate used to predict the error probability of the packet for the initial and the forced transmission should be adjusted. If the user is in need of segmentation, it re-selects its format for the inital transmission, now calculating the code rate as if it has a retransmission, increasing the number of total bits and lowering the code rate for calculation of predicted error probability. Then it transmits at its actual code rate which is higher than the resulting code rate from the combination of the initial transmission and the forced retransmission. The forced transmission also calculates the predicted probability of error from the lower code rate.

Forced in Retransmission

When there is time for a retransmission, Forced can transmit the same packet up to three times, as seen in Figure 3.8. Should the link adaptation for the first packet not achieve the target reliability, it will directly issue a retransmission in the second slot. Thereafter, if the acknowledgment was a NAK for the first transmission a retransmission is issued in the third slot. This can lead to unnecessary transmissions since it might have been enough with a second retransmission. However, this also adds extra reliability that might be beneficial.

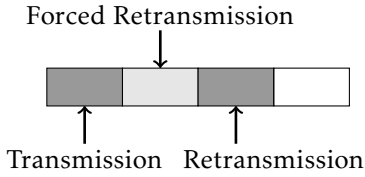


Figure 3.8: *The method Forced in retransmission can both force a retransmission and make use of the retransmission from fast HARQ.*

3.3.6 Delayed Forced

This method is only evaluated in the retransmission scenario. Instead of forcing a retransmission on the first transmission, it would be better to wait for the response of the first transmission and issue a retransmission only if needed. If the retransmission's predicted error probability does not meet its target reliability, a retransmission is forced directly in the fourth slot, as can be seen in Figure 3.9. Note however that this makes use of all our slots and is only possible when the scheduling latency is negligible or the latency demand can be slightly loosened. This method is still interesting to compare with Forced to see if the extra reliability Forced provides is needed or if Delayed forced can support a higher URLLC capacity.

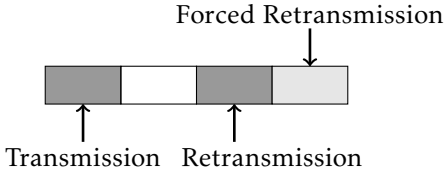


Figure 3.9: *The method Delayed forced transmits in slot 1, makes use of the retransmission from fast HARQ in slot three and forces a retransmission in slot four.*

3.3.7 Dare

This method is only evaluated in the retransmission scenario. The idea is to use a much higher probability of error for the first transmission, 10^{-1} instead of 10^{-5}. In 90% of the transmissions the packets will be received correctly. The other packets get a retransmission that uses the correct target reliability ($1 - 10^{-5}$). This might save resources and be reliable enough. However, due to an implementation miss, Dare used the high probability of error also for its retransmission in the simulations. This method was inspired by the Intel proposal "URLLC link adaptation aspects" [24].

Dare adjusts the code rate with respect to the number of retransmissions for its retransmission in order to get a correct resulting error probability.

3.3.8 Summary

The proposed methods are quite many and therefore summarized in Table 3.4 and this section. The column *Segments* is marked with an "x" if the method segments its packets. The methods that are tested only in the retransmission scenario are marked in the column *Retransmission only*. The methods are many in number since when a couple of them were implemented and functional it was relatively simple to implement more. There are also many interesting aspects to investigate.

Single Transmission

In the Single Transmission scenario, there is no time for a retransmission. The proposed methods are: Baseline, Two is Enough, Estimated, Never, and Forced. The first three methods are similar in that they segment the packet in each slot if necessary. What differs is the target error probability and number of segments used.

Never is used as a worst-case scenario and is meant to show how much can be gained through segmentation. Forced does not segment but issues a retransmission without waiting for an acknowledgment if it cannot meet the target error probability. Baseline, Two is Enough and Never are mostly used as comparisons. What is of greatest interest is how Estimated and Forced performs. Should Forced outperform Estimated then it might be better to issue retransmissions than segmenting overall.

Retransmission

In the retransmission scenario, there is time for a retransmission based on a HARQ feedback from the UE. The proposed methods are: Baseline, Two is Enough, Estimated, Never, Forced, Delayed forced, and Dare. The summary given for Single transmission holds here as well, except that Never might perform better compared to the other methods when there is a retransmission available. Estimated and Forced are still interesting to compare with each other to see if segmentation is a valid option. The method Delayed forced should be able to support a higher

load than Forced but might not be as reliable as Forced. Dare is a wild guess that has to be evaluated.

Table 3.4: *Overview of proposed methods*

Name	Segments	Retransmission only	Argument
Baseline	x		Baseline
Two is Enough	x		Simple segmentation
Estimated	x		True segmentation
Never			Worst-case
Forced			Alternative to segmentation
Delayed forced		x	Variant of forced
Dare		x	Alternative to segmentation

4

Results with Single Transmission

This chapter presents the obtained results for single transmission. In this scenario, the methods were simulated for two target reliabilities, $1 - 10^{-3}$ and $1 - 10^{-4}$, referred to as moderate and high reliability. The results for the two different reliabilities are presented in this chapter.

4.1 Results for Moderate Reliability

To evaluate the methods in the single transmission scenario, the methods were first simulated for a lower target probability of error, $P_{e_{tot}} = 10^{-3}$, compared to the requirement on URLLC of an error probability of 10^{-5}. For this probability of error, three settings were simulated to see how the size of the SINR history, N_{SINR}, and the back-off, Δ_{SINR} parameter affected the methods. The parameters for the settings for the single transmission scenario are summarized in Table 4.1. The setting Basic is used as a reference.

Table 4.1: *Simulation parameters for single transmission, moderate reliability.*

Setting	$P_{e_{tot}}$	Number of users	Δ_{SINR} [dB]	N_{SINR}
Basic	10^{-3}	75	1	20
Long history	10^{-3}	75	1	1000
Large Δ_{SINR}	10^{-3}	75	2	20

For each setting, the offered cell load was varied by adjusting the period of the arriving packets in order to see for which offered cell load the different methods could fulfill the requirements on latency and reliability for all users. The period and corresponding frequency for the simulated offered cell loads are presented in

Table 4.2 and are measured per user. The values of packet arrival rate are derived from the period of packet arrival. The period of packet arrival was varied and tested in the simulator. Values were chosen to observe both many and few users achieving the requirements.

Table 4.2: *Period of packet arrival and corresponding rate of packet arrivals per user for single transmission, moderate reliability.*

Load	Lowest	Low	Medium	High	Highest
Period [s]	0.00307	0.00232	0.00157	0.00132	0.00117
Rate [packets/s]	326	431	637	758	855

The URLLC capacity with these three settings is plotted and analyzed in Section 4.1.1. Thereafter, Section 4.1.2 analyzes the setting *Basic*.

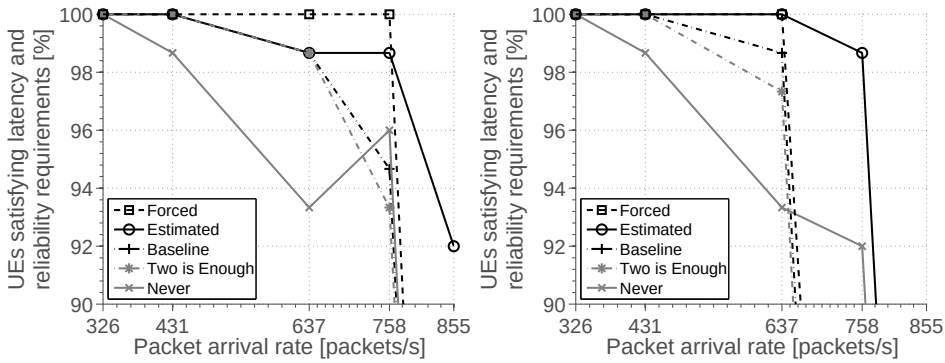
4.1.1 Comparison of URLLC capacity due to History-Size and Back-Off

The Basic setting is compared to a setting with larger N_{SINR} and to another setting with larger Δ_{SINR} in this section. An increase in Δ_{SINR} or N_{SINR} leads to a more pessimistic channel quality estimate and assignment of more resources for each transmission. This improves URLLC capacity when there are plenty of resources but the estimate is overly positive. If the channel estimate is overly positive for a transmission, the transmission will be assigned too few resources for the actual channel quality, and the packet might not be correctly decoded. It would be beneficial to use more resources to increase the probability of delivering the packet. However, a large increase in Δ_{SINR} or N_{SINR} might decrease the number of available resources so that users start to fail because they run out of resources instead.

An increase in Δ_{SINR} makes the channel quality estimates of all users more pessimistic and an increase in N_{SINR} increases the number of saved SINR values for a user. Since it is the lowest SINR value among the saved values that is used in the link adaptation, an increased number of saved values leads to a more pessimistic channel quality estimate. An old, low SINR value will be chosen over the more recent higher SINR values. Exactly how pessimistic the channel quality estimate is depends on the content of the saved history for each user. A user with high quality will save many high-quality SINR and use the lowest of these, for example.

To clearly see the effect of short and long SINR history, the URLLC capacity for the basic setting with $\Delta_{\text{SINR}} = 1$ and $N_{\text{SINR}} = 20$ is plotted beside the URLLC capacity for the setting with $\Delta_{\text{SINR}} = 1$ and $N_{\text{SINR}} = 1000$ in Figure 4.1.

With the basic setting in Figure 4.1a, Forced achieves 100% successful users for all packet arrival rates but the highest. Estimated, Baseline and Two is Enough achieve 100% successful users for the two lower packet arrival rates while Never only achieves it for the lowest packet arrival rate. Never also decreases its percentage of successful users for a lower packet arrival rate in Figure 4.1a. This is studied and explained in the section *A note on Never*.



(a) $\Delta_{SINR} = 1$, $N_{SINR} = 20$ *(b)* $\Delta_{SINR} = 1$, $N_{SINR} = 1000$

Figure 4.1: *Comparison of URLLC capacity due to N_{SINR}, limited to the interval 90-100%.*

For a longer SINR history in Figure 4.1b, Estimated and Forced achieve 100% successful users for quite high packet arrival rate. Two is Enough and Baseline achieve 100% successful users for the two lower packet arrival rates and Never only achieves 100% successful users for the lowest packet arrival rate.

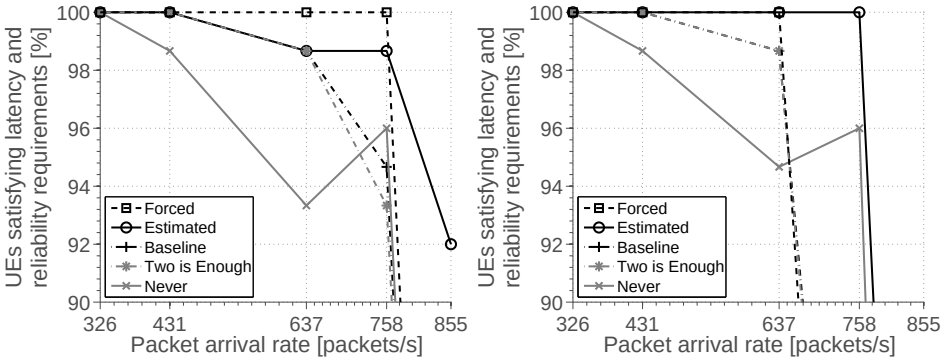
From Figure 4.1, it can be seen that for a packet arrival rate of 758 packets/s, all methods improve their percentage of successful users with a shorter history. Decreasing the packet arrival rate to 637 packets/s, it is clear that Estimated is improved with a longer history and able to reach 100% successful users. For lower packet arrival rates, both of the settings yield similar results. Also, the plot is limited in the interval from 90-100% but it can be seen that the lines drop drastically downward to the right of 758 packets/s in Figure 4.1a. The packet arrival rate is too high to yield a high percentage of successful users for a packet arrival rate larger than 758 packets/s. The effects of high packet arrival rates is more closely studied in Section 4.1.2.

The basic setting with $\Delta_{\text{SINR}} = 1$ and $N_{\text{SINR}} = 20$ is plotted in Figure 4.2a, beside the setting $\Delta_{\text{SINR}} = 2$ and $N_{\text{SINR}} = 20$ in Figure 4.2b. Comparing Figure 4.2 to Figure 4.1, it is clear that a larger Δ_{SINR} yields similar results to a larger N_{SINR}. Estimated gets an even better percentage of successful users, reaching 100% successful users for packet arrival rate 758 instead of 637 packets/s, otherwise the results are very similar.

4.1.2 Short History

In order to study the different methods' performance, the Basic setting will be studied in greater detail in this section. In Figure 4.3, the URLLC capacity for the Basic setting is plotted without limiting it to any interval.

The method Forced is the only method to achieve 100% successful users for the higher packet arrival rates with packet arrival rates 637 and 758 packets/s.



(a) $\Delta_{SINR} = 1$, $N_{SINR} = 20$ *(b)* $\Delta_{SINR} = 2$, $N_{SINR} = 20$

Figure 4.2: *Comparison of URLLC capacity due to Δ_{SINR}, limited to the interval 90-100%.*

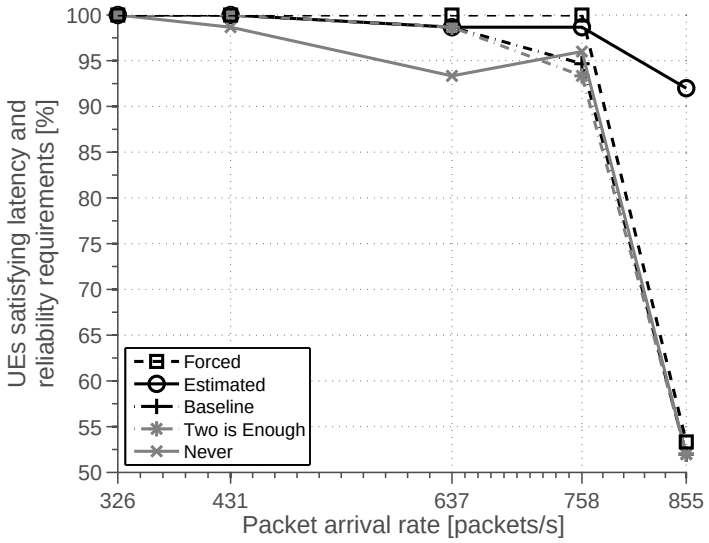


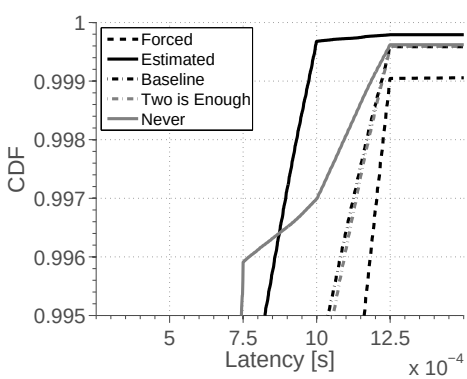
Figure 4.3: *URLLC capacity for Basic setting, covering very high packet arrival rate.*

No method achieves 100% successful users for the highest packet arrival rate of 855 packets/s.

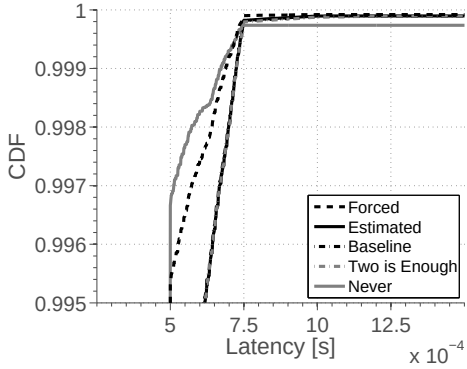
All methods improve with a lower packet arrival rate, except Never that decreases in percentage of successful users for the packet arrival rate 637 packets/s. As mentioned, this is studied and explained in the section *A note on Never*. From Figure 4.3 it is also clear that for a very high packet arrival rate, all methods except Estimated drastically worsen.

A CDF of the delay for all users with very high packet arrival rate (855 packets/s) and medium packet arrival rate (637 packets/s) is depicted in Figure 4.4. In Figure 4.4b the CDF of delay with medium packet arrival rate is above the required reliability of 0.999 and within the latency bound of 1 ms. However, from Figure 4.3 it is clear that does not mean 100% successful users is achieved. For example, the method Never does not achieve 100% successful users for the packet arrival rate of 637 packets/s in Figure 4.3, even though its CDF of delay in Figure 4.4b seems to meet the demands. In Figure 4.4a, it is clear that all methods except Estimated fail the latency bound of 1 ms. For a high packet arrival rate, it seems like the latency is the deciding factor.

Baseline and Two is Enough (if three segments are used) use a different target error probability for their segments compared to Estimated, however in Figure 4.4a they achieve a high resulting reliability close to Estimated, although too late. Estimated seems to perform better mostly due to the method's ability to keep track of the number of slots available and to allow a higher probability of error in the third slot, while Baseline and Two is Enough wait until the fourth slot to schedule their last users.



(a) High packet arrival rate, 855 packets/s



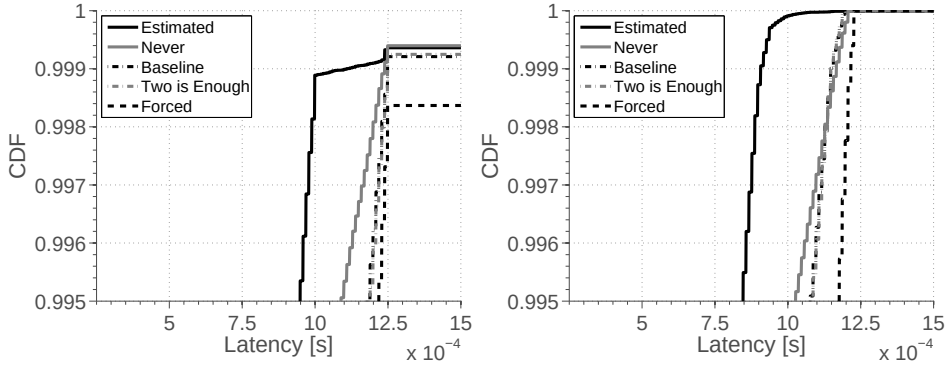
(b) Medium packet arrival rate, 637 packets/s

Figure 4.4: *CDF of delay over all packets from all users, with high and medium packet arrival rate.*

High Packet Arrival Rate

The fact that latency is the limiting factor for high packet arrival rate, can also be seen in Figure 4.5, where two users that fail the demands for most methods with the packet arrival rate of 855 packets/s are plotted. It is the latency demand that is not met while the reliability demand is met, though too late. In Figure 4.5a, the method Forced also fails the reliability demand. User 26 succeeds for the method Estimated as seen in Figure 4.5b. Estimated does achieve quite a high percentage

of successful users for high packet arrival rates, which can be seen in Figure 4.3, and User 26 is one of the users that succeed.



(a) User 1 *(b) User 26*

Figure 4.5: *CDF of delay for two users that fail the requirements with high packet arrival rate, 855 packets/s.*

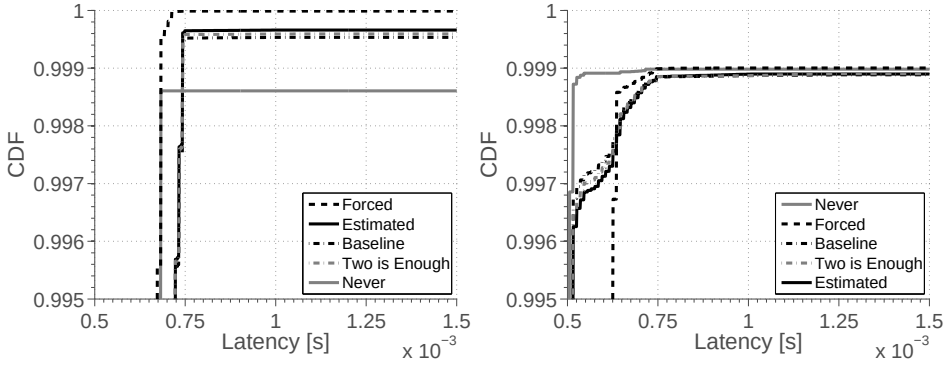
Medium Packet Arrival Rate

For a lower packet arrival rate, 637 packets/s, all methods but Forced have users that cannot achieve the requirements on reliability and latency. In Figure 4.6 two users that do not meet the demands for at least one method, are plotted. The users' CDF of delay is seen in Figure 4.6a and 4.6b, which clearly show that it is the reliability demand that is not met. In Figure 4.6a, only the method Never is below the targeted reliability of 0.999. But for the second user, in Figure 4.6b, also Estimated, Baseline and Two is Enough fall below 0.999. Forced barely makes it.

Since Never always tries to transmit, no matter if it achieves the target reliability, it is not surprising that it fails due to wrong target reliability, missing 0.999. However, the estimating methods Estimated, Two is Enough and Baseline try to segment in order to achieve the correct target reliability. To understand why the estimating methods do not achieve the demands for user 64, the two users are compared in Figure 4.7 and Figure 4.8.

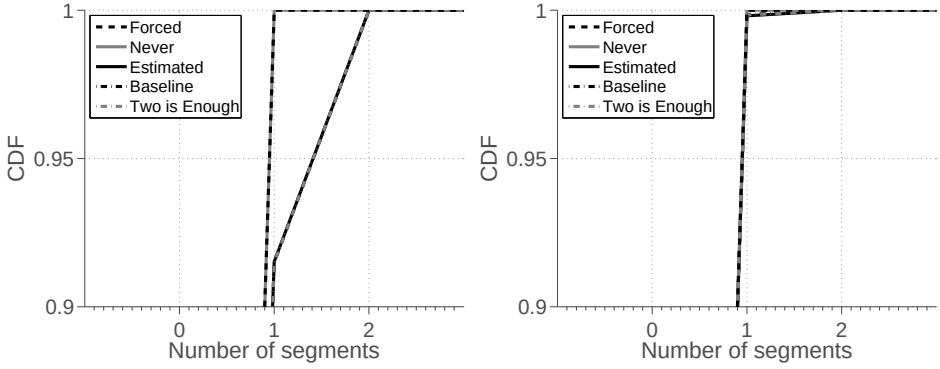
In Figure 4.7 the CDF of the number of segments for the transmitted packets is seen. It is clear from Figure 4.7b that the estimating methods do not segment very often for user 64, while the number of packets split into two segments is greater for user 33 in Figure 4.7a. Since User 64 does not meet the reliability, it should segment more to try to reach it. The question then becomes, why does User 64 not segment more?

In Figure 4.8, the difference in DL SINR (the difference between the actual SINR at the UE and the SINR estimated by the base station) is plotted. When the user has an estimated channel quality that is greater than the actual channel quality, the difference in DL SINR is negative. When the difference in DL SINR is



(a) User 33 *(b) User 64*

Figure 4.6: *CDF of delay for two users that fail the requirements with medium packet arrival rate, 637 packets/s.*

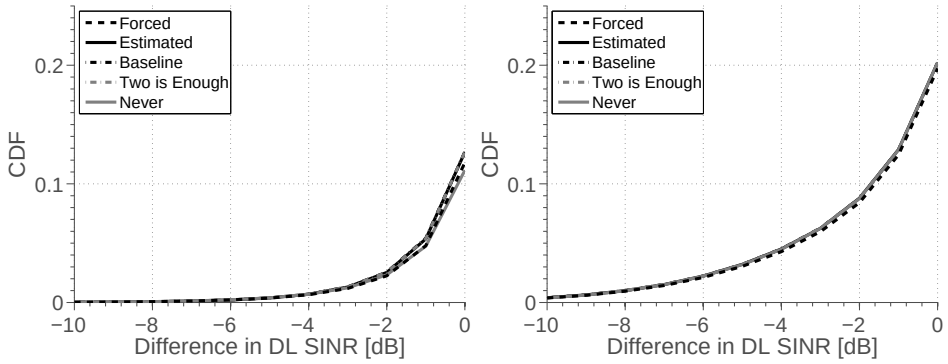


(a) User 33 *(b) User 64*

Figure 4.7: *CDF of number of segments for two users that fail the requirements with medium packet arrival rate, 637 packets/s.*

negative (less than 0 dB), the user believes the channel quality to be better than what it actually is, and thus assigns too few resources or does not realize it needs to segment in order to reach the target reliability.

User 64 has a probability of 20% to have an SINR difference lower than 0 dB while User 33 only has a probability of 10% to have an SINR difference lower than 0 dB. The difference is not very big, but enough to cause more lost packets and a fewer number of segmented packets due to misinformation. User 64 in Figure 4.8b will more often than User 33 in Figure 4.8a have an optimistic channel quality estimate, thus assigning less resources and not segment its packets since the channel quality seems good. A more pessimistic channel quality estimate is



(a) User 33 *(b) User 64*

Figure 4.8: *CDF of difference in DL SINR for two users that fail the requirements with medium packet arrival rate, 637 packets/s.*

needed, and as seen in Figure 4.1 and 4.2, Estimated's percentage of successful users is improved with a larger N_{SINR} or Δ_{SINR}.

A note on Never

In Figure 4.3, it was noted that all methods except Never increase their percentage of successful users with lower packet arrival rate. Never instead decreases in percentage of successful users for the packet arrival rate of 637 packets/s. There are three users that meet the demands for packet arrival 758 and 431 packets/s but not for the packet arrival 637 packets/s. One of these three users is plotted in Figure 4.9.

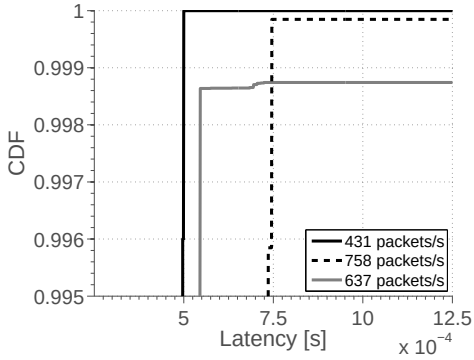


Figure 4.9: *CDF of delay for a problematic user using the method Never with different packet arrival rates.*

In Figure 4.9, Never falls below the targeted 0.999 with the medium packet

arrival rate of 637 packets/s, while it is well above the target for a lower and higher packet arrival rate. However, the delay-curve is further to the left for the medium packet arrival rate than for the high packet arrival rate. This means that more packets arrive earlier since more resources are available. The user transmits in Slot 2 if there are too few resources in Slot 1 to schedule all users with a packet. These occurrences have decreased for Never now that the packet arrival rate is less. However, the reliability target is not met.

I believe that this is due to the now relatively shorter history at this packet arrival rate. The period between packets have increased, while SINR is still stored at every slot. Therefore the user has history about fewer packets to its disposal. Never benefits from a larger Δ_{SINR} since it then assigns extra resources and the few times it does not seem to meet the target reliability, should the Δ_{SINR} be large enough, it might be met anyway. Never also benefits from a larger N_{SINR} for high packet arrival rates, as seen in Figure 4.1b and 4.2b, where only Estimated and Never are the methods that achieve a percentage of successful users larger than 90 for the packet arrival of 855 packets/s. This is since Never uses fewer resources when compared to the other methods that all add resources in order to perform segmentation.

The percentage of successful users of Never increases for the next packet arrival rate of 431 packets/s, as seen in Figure 4.9. This is since the packet arrival is now much lower and the number of segmentations overall is small. In Figure 4.9 one can see that almost all packets are delivered in the first slot for this user with the low packet arrival rate.

4.1.3 Resource Efficiency

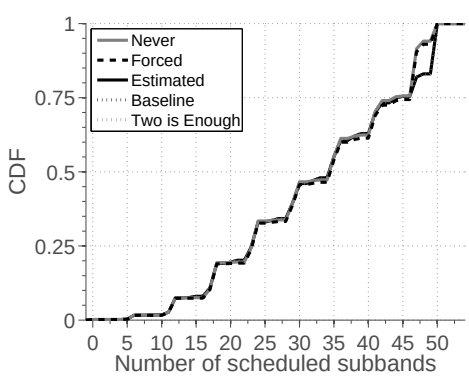
In Figure 4.10, the CDF of the number of scheduled subbands for DL in the network can be seen, for different packet arrival rates. The number of subbands assigned to a user is incremented in steps, usually of step-size two or three subbands. Therefore, a user cannot be assigned any number of subbands from one to fifty. This causes the total number of subbands to also get a step-like behaviour, as seen in Figure 4.10.

With high packet arrival rate, as depicted in Figure 4.10a, for 50% of the transmissions, 35 or fewer subbands are scheduled which is more than half of the total subbands, since the total number of subbands is 50. In Figure 4.10b, for 50% of the transmissions, 23 or fewer subbands are scheduled. This gives an idea of how tightly scheduled the subbands are with different packet arrival rates.

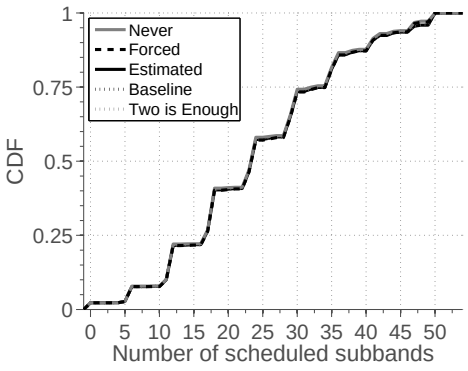
In Figure 4.11, a zoomed in version of Figure 4.10 is depicted. From this figure, the differences between the methods are distinguishable.

Never lies above the other methods in Figure 4.11a. For a given number of subbands the probability of the method Never to schedule that many subbands or fewer is greater than the other methods. In other words, Never uses fewer subbands.

Forced on the other hand, can be seen below the other methods in Figure 4.11a, using more subbands than them until the number of subbands is slightly above 45. However, the difference in number of subbands between Forced and

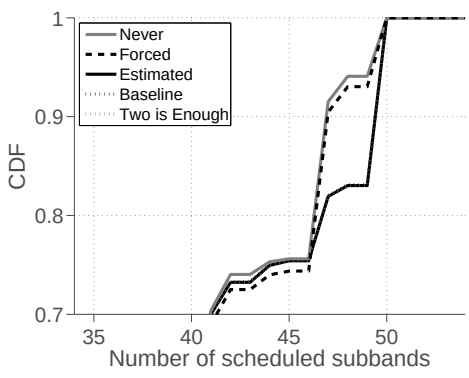


(a) High packet arrival rate, 855 packets/s

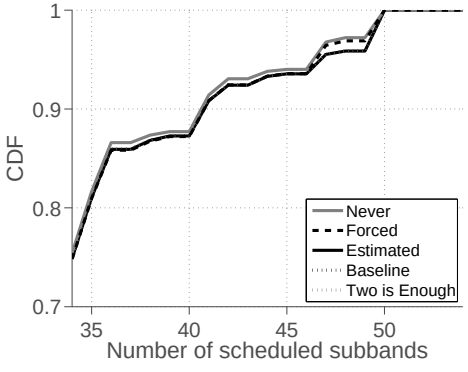


(b) Medium packet arrival rate, 637 packets/s

Figure 4.10: *Number of scheduled subbands in DL with high and medium packet arrival rate.*



(a) High packet arrival rate, 855 packets/s



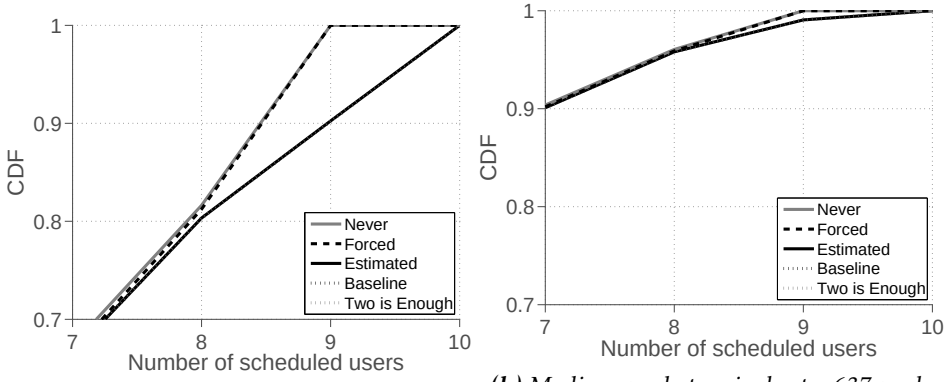
(b) Medium packet arrival rate, 637 packets/s

Figure 4.11: *Closeup of number of scheduled subbands in DL with high and medium packet arrival rate.*

the other methods is very small at this point and therefore the methods can be thought of as using almost the same number of subbands. At 46 scheduled subbands, the CDF makes a high "jump" up towards 47 subbands at 90%. Therefore, Forced only has 10% probability to schedule more than 47 subbands. The segmenting methods however, are closer to 18% probability to schedule more than 47 subbands.

In Figure 4.11b, it can be seen that for the medium packet arrival rate, Never also uses the least resources. A similar high "jump" to that in Figure 4.11a can

also be seen in Figure 4.11b, where Forced very seldom schedules more than 47 subbands.



(a) *High packet arrival rate, 855 packets/s*

(b) *Medium packet arrival rate, 637 packets/s*

Figure 4.12: *Number of scheduled users in DL with high and medium packet arrival rate.*

Since the segmenting methods can schedule a segment for a user which does not require as many subbands as scheduling a whole packet, these methods more often utilize all subbands compared to Forced and Never. The methods Forced and Never can also schedule all subbands if the users demand many subbands for their packets. However, if 47 subbands is enough to schedule all users but one, these last three subbands might not be sufficient to schedule the last user. On the other hand, three subbands might be enough to schedule a segment. The segmenting methods are better at squeezing in users and thus scheduling more subbands and users. In Figure 4.12 it is seen that the segmenting methods can schedule more users than the methods Forced and Never. Especially with high packet arrival rate, in Figure 4.12a, the methods Estimated, Baseline and Two is Enough schedule 10 users in almost 10% of all transmissions while Never and Forced only schedule 9 users.

4.2 Results for High Reliability

The settings used for high reliability are similar to those used for moderate reliability. However, the choice of the parameters Δ_{SINR} and N_{SINR} affected the methods to a greater extent when simulated with a high reliability. Therefore, quite many values for the parameters were simulated, as seen in Table 4.3, but with quite few number of different packet arrival rates, as seen in Table 4.4.

The methods Baseline and Two is Enough are simple methods for segmentation while Estimated is a more advanced method for segmentation that works similarly to Baseline and Two is Enough but actually estimates the number of

needed segments. Since Baseline and Two is Enough were outperformed by the method Estimated when simulating with a moderate reliability, they were not simulated for a high reliability. The methods that were simulated for high reliability are: Estimated, Never and Forced. Even though Never also was outperformed in moderate reliability it is a good baseline to compare with.

Table 4.3: *Simulation parameters for single transmission, high reliability.*

Setting	$P_{e_{tot}}$	Number of users	Δ_{SINR} [dB]	N_{SINR}
1	10^{-4}	75	1	20
2	10^{-4}	75	1	1000
3	10^{-4}	75	2	20
4	10^{-4}	75	2	1000
5	10^{-4}	75	3	20
6	10^{-4}	75	3	1000

For each setting, the offered cell load was varied by adjusting the period of the arriving packets in order to see for which offered cell load the different methods could reach 100% successful users. The period and corresponding frequency for the simulated offered cell loads are presented in Table 4.4 and are measured per user.

Table 4.4: *Period of packet arrival and corresponding rate of packet arrivals per user for single transmission, high reliability.*

Load	Low	Medium	High
Period [s]	0.00507	0.00207	0.00117
Rate [packets/s]	197	483	855

4.2.1 Comparison of URLLC capacity due to History-Size and Back-Off

The selection of the parameters Δ_{SINR} and N_{SINR} affect the methods' performance, especially for Estimated. For one setting, Estimated is the worst method, even outperformed by Never, as seen in Figure 4.13, while for another setting, Estimated not only beats Never but also is the only method able to achieve 100% successful users, as seen in Figure 4.14. The other two methods have one user that is very close to meeting the demands but fails the reliability for the low packet arrival rate, therefore they only achieve 98.7% successful users.

Estimated's varying performance due to the parameters Δ_{SINR} and N_{SINR} can clearly be seen in Figure 4.15 where the setting $\Delta_{SINR} = 3$, $N_{SINR} = 20$ achieves 100% successful users while $\Delta_{SINR} = 3$, $N_{SINR} = 1000$ only achieves 95% successful users for the low packet arrival rate of 197 packets/s.

Forced on the other hand, does not vary its performance as much, as seen in Figure 4.16 where the largest difference between two settings is 1.33% for the

lowest packet arrival rate of 197 packets/s.

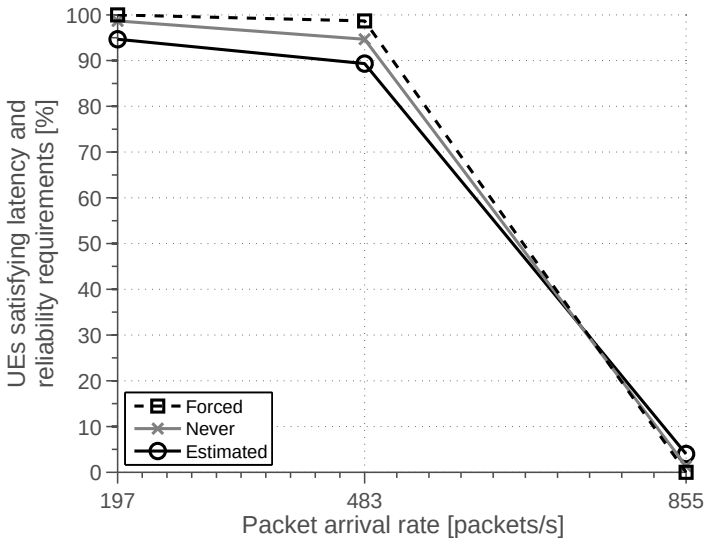


Figure 4.13: *URLLC capacity for the setting $\Delta_{SINR} = 3$ and $N_{SINR} = 1000$, Estimated is outperformed not only by Forced but also by Never.*

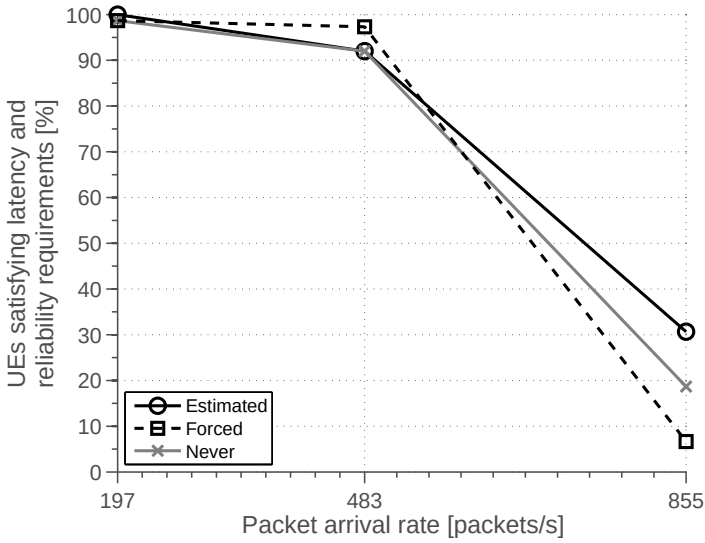


Figure 4.14: *URLLC capacity for the setting $\Delta_{SINR} = 2$ and $N_{SINR} = 20$, Estimated is the only method to achieve 100% successful users, while Forced and Never have one user that fails the requirements.*

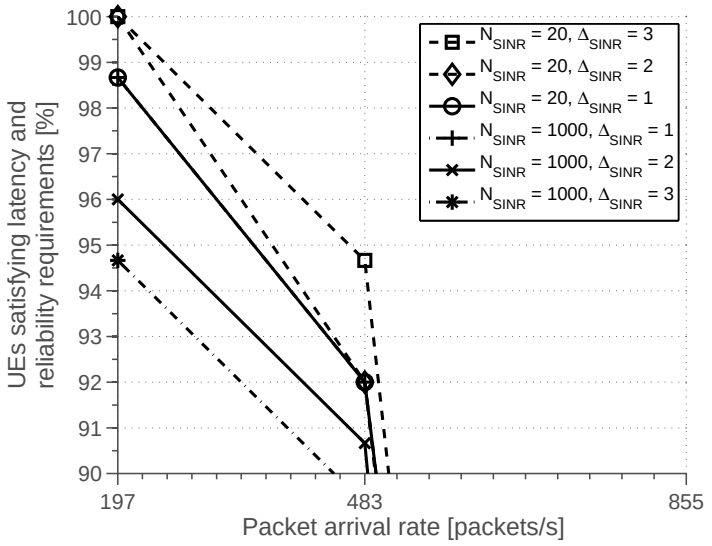


Figure 4.15: *URLLC capacity of the method Estimated for all simulated settings, the results vary with the settings and a larger list-size decreases the percentage of UEs that satisfy the latency and reliability requirements.*

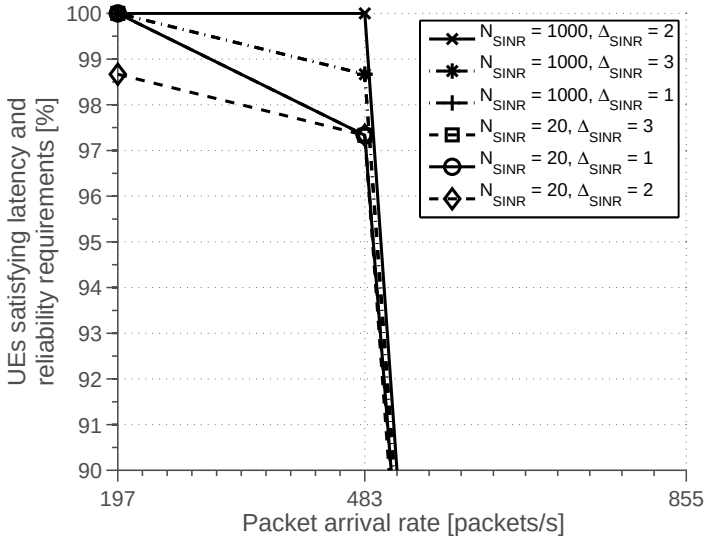


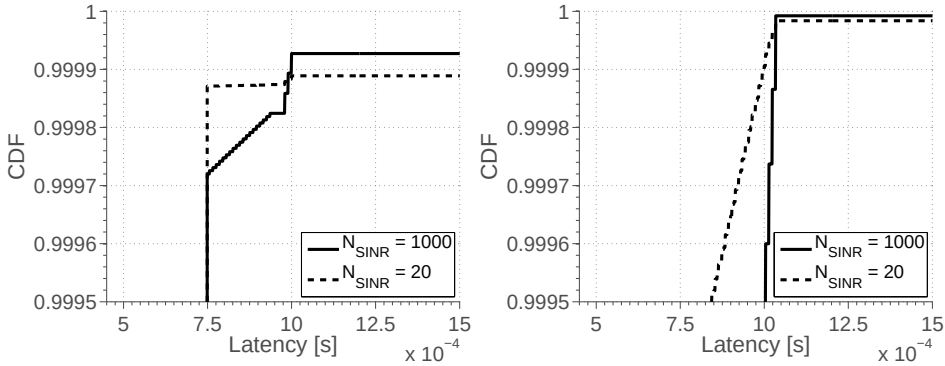
Figure 4.16: *URLLC capacity of the method Forced for all simulated settings, the results are similar but a larger list-size increases the percentage of UEs that satisfy the latency and reliability requirements.*

Another observation is that Estimated's percentage of successful users decreases with a large N_{SINR} while Forced's percentage of successful users increases with a large N_{SINR}. The difference in performance for the method Estimated due to a large or small N_{SINR} decreases with smaller Δ_{SINR}, in fact for $\Delta_{\text{SINR}} = 1$, the achieved percentage of successful users for both $N_{\text{SINR}} = 20$ and $N_{\text{SINR}} = 1000$ is equal. On the other hand, for $\Delta_{\text{SINR}} = 3$ the highest percentage of successful users across all settings is achieved with $N_{\text{SINR}} = 20$ but the lowest percentage of successful users across all settings is achieved with $N_{\text{SINR}} = 1000$.

4.2.2 Study of Estimated

As seen in Figure 4.15, Estimated increases its URLLC capacity with a smaller N_{SINR} and larger Δ_{SINR}. Both of these parameters give the user a more pessimistic view of the channel quality estimate when increased, which makes the user assign more resources. In order to understand how these parameters affect Estimated's URLLC capacity, the users that do not meet the requirements for different parameter setups are studied.

With a back-off of $\Delta_{\text{SINR}} = 1$, it was seen that the achieved percentage of successful users was the same for both $N_{\text{SINR}} = 20$ and $N_{\text{SINR}} = 1000$. Therefore, the same number of users fail for both of the tested values of N_{SINR}. However, studying the failing users for the settings $N_{\text{SINR}} = 20$ and $N_{\text{SINR}} = 1000$ shows that both settings have six users that cannot meet the requirements of which only three are in common. The CDF of two users that do not meet the requirements for one setting of N_{SINR} are plotted in Figure 4.17 for both $N_{\text{SINR}} = 20$ and $N_{\text{SINR}} = 1000$. It is clear that for $N_{\text{SINR}} = 20$, User 19 meets the requirements but User 15 does not, but when $N_{\text{SINR}} = 1000$ it is User 19 that cannot meet the requirements but User 15 manages to.



(a) User 15 *(b) User 19*

Figure 4.17: *CDF of delay for two problematic users with medium packet arrival rate, 483 packets/s, with* $\Delta_{SINR} = 1$.

For User 15 in Figure 4.17a, the setting $N_{\text{SINR}} = 20$ does not manage to meet

the targeted reliability. On the other hand, User 19 in Figure 4.17b for the setting $N_{\text{SINR}} = 1000$ does not manage to meet the required latency. Studying all users that fail the requirements for medium packet arrival rate, $\Delta_{\text{SINR}} = 1$, it is clear that for $N_{\text{SINR}} = 20$ the failing users fail since they cannot meet the targeted reliability. For $N_{\text{SINR}} = 1000$, most failing users fail due to not meeting the required latency and some due to missed reliability. It seems like $N_{\text{SINR}} = 20$ leaves some users with an optimistic estimate of the channel quality, thus assigning too few resources and missing the reliability. When increasing to $N_{\text{SINR}} = 1000$ most users instead fail due to a too large latency, indicating a too pessimistic channel quality estimate that consumes resources for all users such that other users cannot get as many resources as they need until it is too late.

Increasing Δ_{SINR} to 3, all failing users for $N_{\text{SINR}} = 1000$ fail due to missed latency. On the other hand, $N_{\text{SINR}} = 20$ only has four users that cannot meet the requirements, two of which do so due to delay and two due to reliability. This parameter setup seems to strike a balance between missing due to reliability or latency and achieves the highest percentage of successful users of the tested setups, as seen in Figure 4.15.

Estimated compared to Never

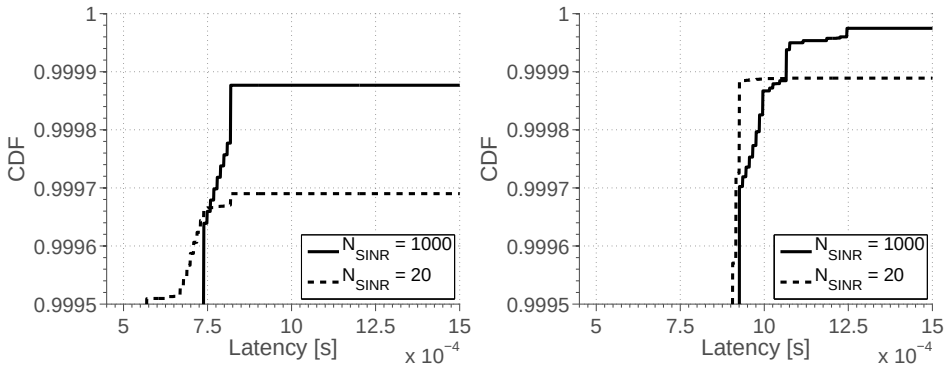
With $N_{\text{SINR}} = 20$ and $\Delta_{\text{SINR}} = 1, 2, 3$, Estimated is at least as good as Never. However, with $N_{\text{SINR}} = 1000$, Never beats Estimated for medium packet arrival rate. When the history-size is large, all users are already over-assigning resources for their transmissions and therefore, segmentation is rarely needed. In Estimated's case, when the estimate shows a need to segment, Estimated assigns subbands in several slots to the task and runs out of subbands for some users in some cases as compared to Never which does not assign any extra when it seems needed.

4.2.3 Study of Forced

The percentage of UEs that satisfy the latency and reliability requirements for Forced increases with a larger N_{SINR} as seen in Figure 4.16, for medium packet arrival rate. This indicates that Forced rarely runs out of resources and a larger N_{SINR} and Δ_{SINR} only help the method to assign more resources to all users so that also the more problematic ones succeed. When increasing the packet arrival to a high packet arrival rate also Forced runs out of resources but that is due to the very high packet arrival rate.

Studying the users that fail the requirements, it turns out that the few users that do fail often do due to a missed reliability. Increasing N_{SINR} improves the reliability of these users and no other users start to fail due to missed latency since there are enough resources.

The parameter setup $N_{\text{SINR}} = 1000$ and $\Delta_{\text{SINR}} = 2$ achieves 100% successful users also for medium packet arrival rate while the setups $N_{\text{SINR}} = 1000$, $\Delta_{\text{SINR}} = 1$ and $N_{\text{SINR}} = 1000$, $\Delta_{\text{SINR}} = 3$ have one user each that fails the requirements. These users' CDF of delay is depicted in Figure 4.18. As can be seen in Figure 4.18a, for the setup $N_{\text{SINR}} = 1000$ and $\Delta_{\text{SINR}} = 1$, the user is User 1 and it fails the



(a) User 1, $\Delta_{SINR} = 1$ *(b) User 64, $\Delta_{SINR} = 3$*

Figure 4.18: *CDF of delay for two problematic users with medium packet arrival rate, 483 packets/s. User 1 cannot meet the reliability for neither $N_{SINR} = 20$ nor $N_{SINR} = 1000$ while User 64 cannot meet the reliability for $N_{SINR} = 20$ and misses the latency with $N_{SINR} = 1000$.*

reliability demand. For the setup $N_{\text{SINR}} = 1000$ and $\Delta_{\text{SINR}} = 3$, the user is User 64 and it fails the latency demand, as seen in Figure 4.18b. Choosing $\Delta_{\text{SINR}} = 2$ seems to strike a balance between over-assigning resources in order to meet the reliability and keeping enough resources to schedule all users in time.

4.2.4 Comparison of Estimated and Forced

How can Forced achieve such a high percentage of UEs that satisfy the latency and reliability requirements? For medium packet arrival rate, 483 packets/s, not a single parameter setup where Estimated beats Forced has been found. In fact, even the best parameter setup for Estimated with the packet arrival of 483 packets/s does not beat any of the tested parameter setups for Forced. Since Estimated estimates the number of needed segments it should be able to use few resources for users that need segmentation but have good quality and many resources for users that need segmentation and have a bad quality. However, this all relies on the fact that Estimated can make a good estimate, based on the information available. Often, this information is inaccurate since time has passed from the moment the channel was measured until the UE transmits its packet. During this time, the UE can change its position, which changes the channel quality. Also, other UEs might have started to transmit, causing more interference. One way to combat the inaccuracies is to use the parameters N_{SINR} and Δ_{SINR} in order to get a more pessimistic channel quality estimate and 'realize' that users are in need of segmentation. However, this might not be enough.

To better see how Forced can beat Estimated, the setup $N_{\text{SINR}} = 20$, $\Delta_{\text{SINR}} = 1$ is studied. In this setup, for medium packet arrival rate, Estimated has 4 users that fail the reliability demand while Forced achieves the reliability demand, as

seen in Figure 4.19 where one of these user's CDF of delay is plotted.

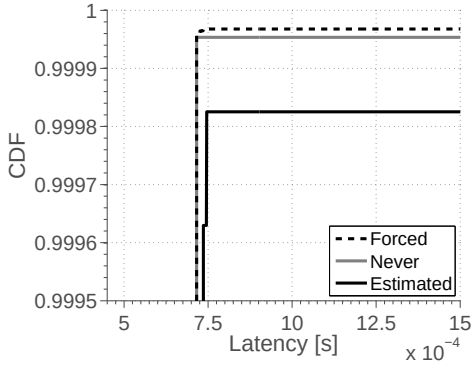


Figure 4.19: *CDF of delay for User 32 with medium packet arrival rate with* $N_{SINR} = 20$, $\Delta_{SINR} = 1$. *The method Estimated clearly falls below the reliability demand while the methods Never and Forced meet the demand.*

Studying this user more closely, it can be seen that its actual SINR at the UE is slightly larger for the methods Forced and Never as compared to Estimated, and the probability to have a negative SINR difference is slightly lower for the methods Forced and Never as compared to Estimated. A negative SINR difference means that the base station estimates a better channel quality for the UE than the actual channel quality. A higher actual SINR improves the SINR difference since the difference is calculated as the actual SINR minus the SINR used in the link adaptation and the difference should be positive. Therefore, since Forced has a higher actual SINR, it increased the probability of receiving the packets correctly and forcing a retransmission when needed.

But, how can Forced have a better actual SINR? The total number of scheduled subbands in each TTI is very similar across all methods. However, Estimated schedules more subbands than Forced and Never for a few number of users, such as for User 50 shown in Figure 4.20.

Estimated schedules multiple subbands for a user during consecutive slots in order to meet the reliability demand while Forced only issues a retransmission, hoping that will be enough to meet the reliability. Since Estimated estimates the number of needed subbands and can use up to three slots to deliver its packet, Estimated should be able to achieve a higher reliability than Forced. The ability to achieve a higher reliability for a user is generally very good, but also costly as seen in Figure 4.20. Forced costs less resources and is ofter reliable enough, as seen in Figure 4.21 where both Forced and Estimated manage to meet the reliability demand.

Estimated costs more resources for problematic users which causes more interference to other users, when compared to Forced. More interference causes a lower actual SINR at other users which makes it harder for them to receive their packets. In addition, a lower SINR forces more users to segment since their chan-

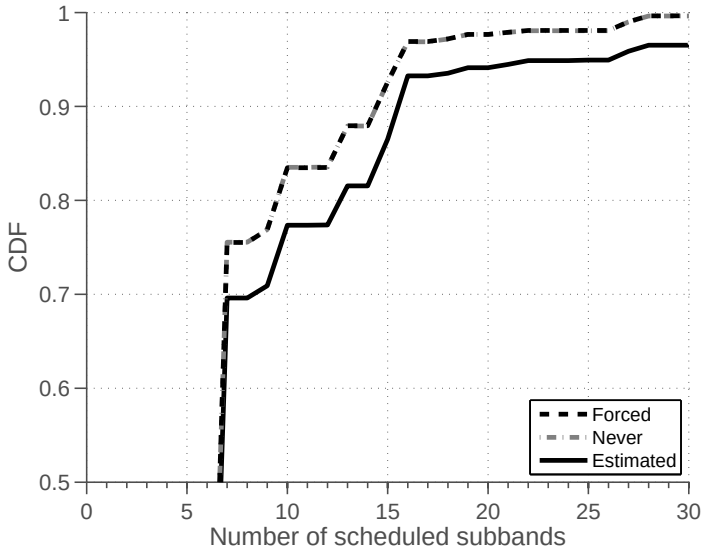


Figure 4.20: *CDF of the number of scheduled subbands for User 50 with medium packet arrival rate and $N_{SINR} = 20$, $\Delta_{SINR} = 1$. The method Estimated has a higher probability to schedule more subbands.*

nel quality now has worsened. With $N_{\text{SINR}} = 20$, $\Delta_{\text{SINR}} = 1$ and medium packet arrival rate Forced forces a retransmission for 3% of its transmissions while Estimated sends two segments in 5% of its transmissions. This increase in number of packets that segments causes further interference to other users for the method Estimated, compared to the method Forced.

In essence, in order for our methods to achieve a high URLLC capacity, N_{SINR} and Δ_{SINR} are needed to operate at a more correct DL SINR for some users, while other users will use more resources than needed. Estimated's strength is that the method can adapt the number of segments and resources for each segment to the channel quality, but for a user with very bad channel quality this will cost many resources. This increased number of used subbands, compared to Forced, in turn causes greater interference among other users, so that in order to deliver their packets at a correct target reliability an even larger N_{SINR} and/or Δ_{SINR} are/is needed. Since Forced does not cause this extra interference, it can still deliver packets to the other users.

4.2.5 Increased Latency Bound

In Section 4.2.2 it was noted that for the setting $\Delta_{\text{SINR}} = 3$ and $N_{\text{SINR}} = 1000$, the users that fail the requirements for the method Estimated all fail the latency requirement. With this large back-off and history-size, all users cost more resources and even though Estimated allows a higher error probability in the third slot,

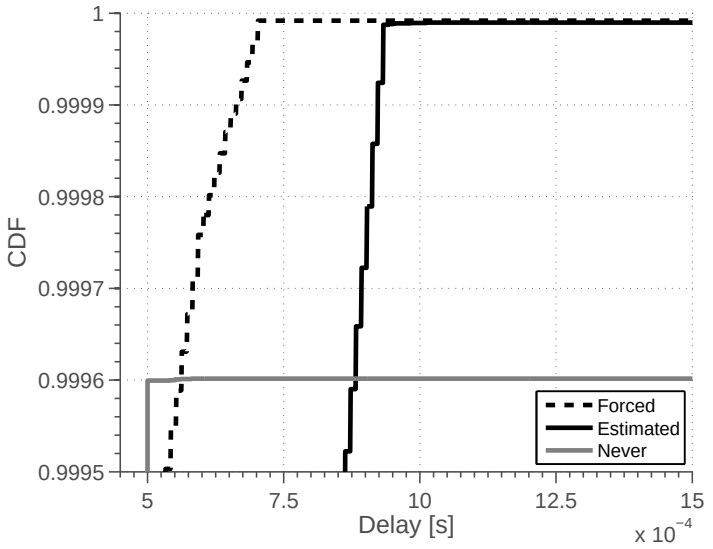
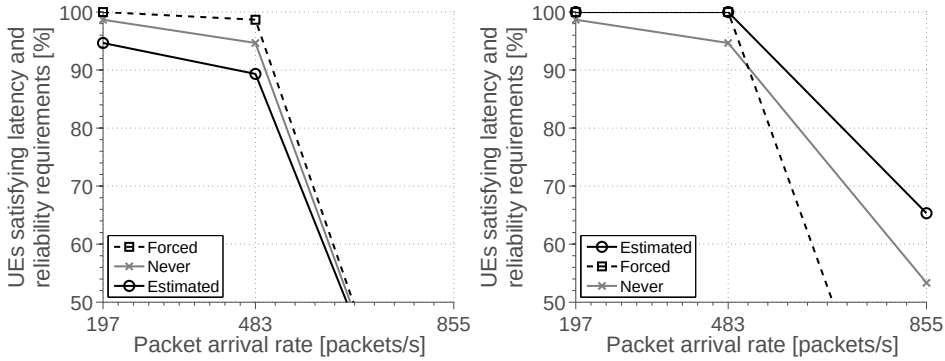


Figure 4.21: *CDF of the delay for User 50 with medium packet arrival rate with $N_{SINR} = 20$, $\Delta_{SINR} = 1$. Both Forced and Estimated meet the reliability and latency.*

some users are forced to be scheduled in the fourth slot, thus arriving at the UE after 1 ms. If the latency bound is increased from 1 ms to 1.25 ms the URLLC capacity is improved for Estimated since the users scheduled in the fourth slot now also count as successful users, as seen in Figure 4.22b where Estimated achieves 100% successful users for both low and medium packet arrival rate and a reasonable percentage of successful users also for high packet arrival rate compared to not achieving a 100% successful users even for low packet arrival rate in Figure 4.22a.

Forced aims to schedule its users in the first two slots but may be forced to schedule the users in later slots if the packet arrival rate is high and there are not enough resources to schedule all users. In this scenario, for the high packet arrival rate of 855 packets/s, the packet arrival rate is so high that all users are not scheduled even within 4 slots for Forced. Therefore, the percentage of successful users is very low for high packet arrival rate, also with an increased latency bound, as seen in Figure 4.22b. However, Forced does improve its percentage of successful users for medium packet arrival rate from 98.7 to 100 with an increased latency bound.

Never, on the other hand, uses the least resources of the three methods. In addition, with $\Delta_{\text{SINR}} = 3$ and $N_{\text{SINR}} = 1000$, all users are already assigned extra resources so it might be enough to do nothing extra for segmentation in most cases. Never schedules a number of users in the 4th slot when the packet arrival rate is high, therefore the URLLC capacity is increased when increasing the latency



(a) *Latency bound 1 ms* ***(b)*** *Latency bound 1.25 ms*

Figure 4.22: *Percentage of successful users for $\Delta_{SINR} = 3$ and $N_{SINR} = 1000$ with the original latency of 1 ms and an increased latency bound of 1.25 ms which increases the URLLC capacity for all methods in general and Estimated in particular.*

bound also for Never.

Estimated is greatly improved with an extended latency bound for the setting $\Delta_{\text{SINR}} = 3$ and $N_{\text{SINR}} = 1000$ as seen in Figure 4.22. The only times the fourth slot is used is if a user has not been scheduled yet or could not be scheduled in slot three since the expected BLEP was 1. One way to improve Estimated also for 1 ms latency bound is to forbid transmissions in the fourth slot since they will arrive outside of the latency bound. Then, these packets will never be received, however there will be more resources left over for other packets to use. Hopefully these resources can ensure that other packets are delivered within three slots. The method was allowed to transmit in the fourth slot in this thesis in order to see how many packets end up in the fourth slot.

Estimated can use one, two or three slots, while Forced can use one or two slots and Never one slot for one packet. Therefore, it is not surprising that Estimated is the method to improve most by extending the latency bound, it is enough for a user to miss one or two slots if segmenting to transmit in the fourth slot. In addition Estimated costs many subbands and runs out of these faster for a higher packet arrival rate than Forced and Never, causing more users to not be scheduled in a slot.

5

Results with Retransmission

This chapter presents the obtained results for retransmission. In this scenario, the methods were simulated for two target reliabilities, $1-10^{-3}$ and $1-10^{-4}$, referred to as moderate and high reliability. The results for the two different reliabilities are presented in this chapter.

5.1 Results for Moderate Reliability

The methods were run with two settings in the retransmission scenario with moderate reliability ($P_{e_{tot}} = 10^{-3}$): one where N_{SINR} was set to 20 and one where N_{SINR} was set to 1000, as can be seen in Table 5.1. In the Retransmission scenario, all methods from Single Transmission are tested together with two more methods, Dare and Delayed Forced. With retransmissions available which can handle most cases when the user has an optimistic estimate of the channel quality by retransmitting the lost packet, the parameter Δ_{SINR} can be lower and most methods are still able to achieve 100% successful users.

Table 5.1: *Simulation parameters for retransmission, moderate reliability.*

Setting	$P_{e_{tot}}$	Number of users	Δ_{SINR} [dB]	N_{SINR}
Basic	10^{-3}	75	0.5	20
Long history	10^{-3}	75	0.5	1000

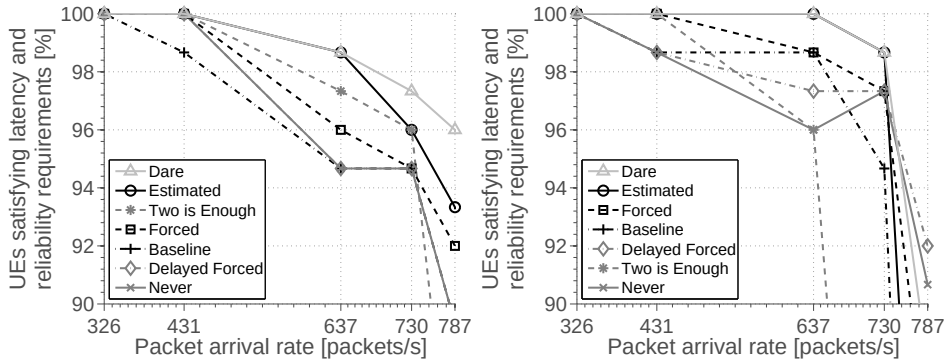
The offered cell loads used are similar to the Single Transmission scenario and presented in Table 5.2.

Table 5.2: *Period of packet arrival and corresponding rate of packet arrivals per user for retransmission, moderate reliability.*

Load	Lowest	Low	Medium	High	Higher	Highest
Period [s]	0.00307	0.00232	0.00157	0.00137	0.00127	0.00117
Rate [packets/s]	326	431	637	730	787	855

5.1.1 Comparison of History-sizes

The methods Dare and Estimated achieve a high percentage of successful users also for high packet arrival rate, as can be seen in Figure 5.1. Increasing N_{SINR} further increases the achieved percentage of successful users for both methods, except for the high packet arrival rate of 787 packets/s. Forced also performs well with both N_{SINR}s, but achieves a lower percentage of successful users than Dare and Estimated. Two is Enough beats Forced when N_{SINR} is 20, as seen in Figure 5.1a, but is among the worst methods with a larger N_{SINR}, depicted in Figure 5.1b.



(a) $N_{SINR} = 20$ ***(b)*** $N_{SINR} = 1000$

Figure 5.1: *Comparison of URLLC capacity due to N_{SINR}, limited to the interval 90-100%.*

As discussed in Section 4.1.1, as long as the URLLC capacity improves with a larger N_{SINR}, the users have enough resources. It seems like Two is Enough is one of the methods to most rapidly get out of resources in the Retransmission scenario as its percentage of successful users worsens for many of the different packet arrival rates with a larger N_{SINR}. One reason for this is that Two is Enough does not take chances like Estimated should the user be in its last slot to transmit. In addition, since Two is Enough uses a tougher target reliability, it assigns more resources to its segments, as compared to Baseline. This causes it to run out of resources quicker, especially when retransmissions are needed.

Delayed Forced and Never are among the worst methods, except for large N_{SINR} and high packet arrival rate, as seen in Figure 5.1b. Delayed Forced very

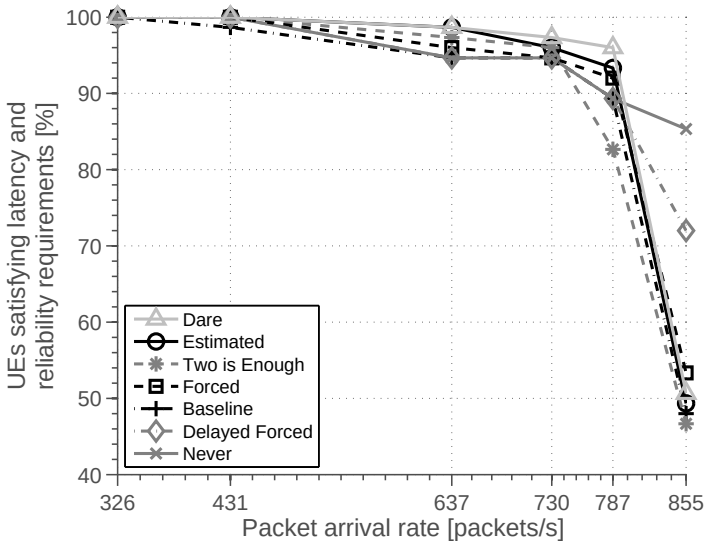
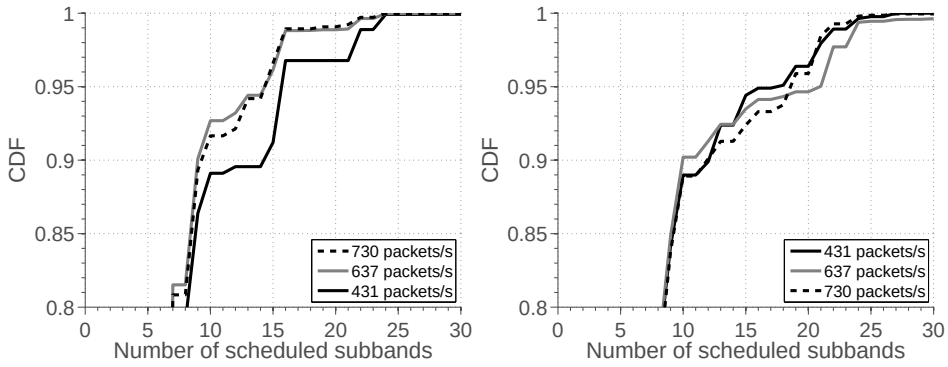


Figure 5.2: *URLLC capacity for $N_{SINR} = 20$, also showing high packet arrival rate.*

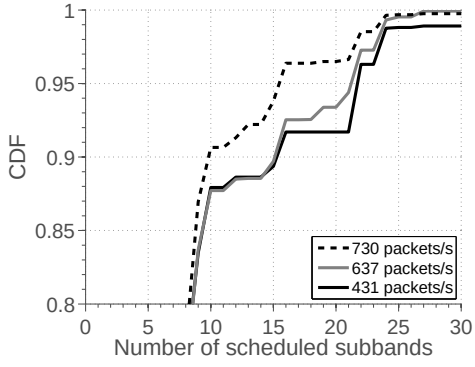
seldom gets use of its forced transmission due to scheduling latency and is therefore very similar to Never. Never simply retransmits a packet if it is lost, not having any of the extra protection mechanisms the other methods do. Therefore it is not surprising that it is among the worst. However, this also costs very few resources for the users. Therefore, Never can support a very high packet arrival rate while the other methods do not have time to schedule all users, as can be seen in Figure 5.2, where all methods except Never and Delayed Forced drastically worsen for the highest packet arrival rate.

Never decreases in percentage of successful users for a lower packet arrival rate with the setting $N_{\text{SINR}} = 1000$ from packet arrival 730 to 637 packets/s in Figure 5.1b. The difference is only 1.3%, which corresponds to one less user achieving the requirements. Studying the users more closely, it is in fact two users, Users 33 and 50, that achieve the requirements for packet arrival rate 730 but not for 637 packets/s, while there is also one user, User 74, that does not achieve the requirements for the packet arrival 730 packets/s, but does for 637 packets/s.

When the packet arrival rate lessens, User 74 gets more resources which greatly improves its delay and reliability, as seen in Figure 5.3a. However, this causes more interference to other users. User 33 does not get more resources for the decreased packet arrival rate, as seen in Figure 5.3a where the curves for 431 and 637 packets/s are very similar. For User 50, the 637 packets/s curve is sometimes below even the 730 packets/s curve in Figure 5.3c. A lower curve for a given number of subbands equals a lower probability to schedule more than that many subbands. In summary, one user gets more resources while other users do not get



(a) User 33 *(b) User 50*



(c) User 74

Figure 5.3: *CDF of number of scheduled subbands for three different users. User 74 increases the number of subbands dramatically from low to medium packet arrival rate while User 33 is unchanged and User 50 loses some subbands.*

more resources or even get less which can cause more users to fail than the number of users that get increased resources. However, this seems to happen rarely and only affects a few users.

5.1.2 Timing

In the retransmission scenario, packets can be counted when delivered on the first try or on the retransmission, 0.5 ms later. As described in Section 3.1.2, the fast HARQ used in the retransmission scenario allows a retransmission two slots after the original transmission. Users are scheduled at each slot so if a user arrives slightly before the slot starts, its delay will be close to the slot duration which is the duration of the transmission itself, 0.25 ms. However, if a user's packet arrives at the start of a slot, it arrives too late to be scheduled in this slot and is

scheduled in the next, resulting in a total delay of the packet closer to 0.5 ms. The last packets from the first slot, arriving at 0.45 ms, can be seen in Figure 5.4 as a sharp rise, especially for Dare. There are many packets with a much lower delay not seen in Figure 5.4 since the Figure only shows the CDF from 0.995 to 1.

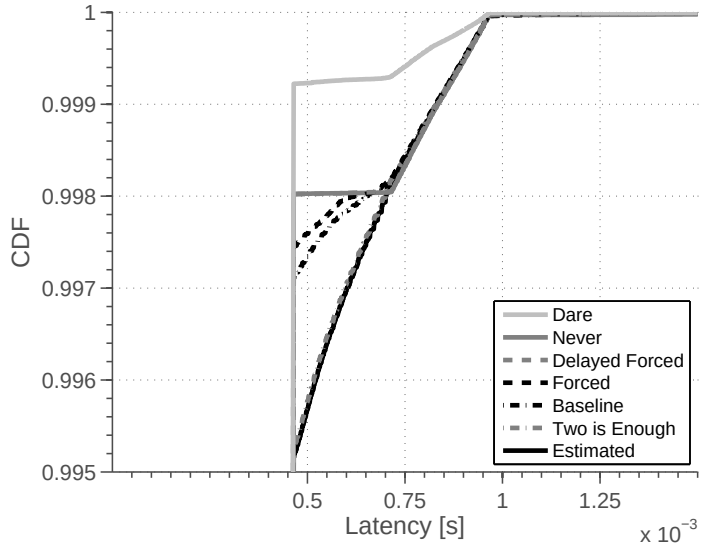


Figure 5.4: *CDF of delay with low packet arrival rate to illustrate when different slots are received.*

During the second slot, the forced retransmission for the method Forced, and the estimating methods' segment number two is sent. In Figure 5.4, one can see that the only methods increasing the probability of delivered packets within the latency 0.45 ms to 0.7 ms are the methods Estimated, Baseline, Two is Enough and Forced. The third slot contains the retransmission of the content of the first slot or segment three if the segmenting methods decided to split into three segments. Since all methods make use of retransmissions, all methods' probability of delivering a packet within the latency increase from 0.7 ms to 0.95 ms.

For higher packet arrival rate, a user might not get its first slot scheduled within 0.2 ms and then this pattern is not visible. For example, Dare might not schedule a user until 0.3 ms after the packet arrived and then this will increase the probability of receiving a packet at the time 0.55 ms which corresponds to slot 2, even tough it was transmitted in slot 1.

5.1.3 Comparison of Estimated and Forced

From Figure 5.1, it is clear that Estimated outperforms Forced. The difference is not great, remember that this simulation was run with 75 users so that if one user fails for one method but not the other, this gives a difference of 1.33% of

successful users. With the packet arrival of 730 packets/s, for $N_{\text{SINR}} = 20$, there is one user which Estimated, but not Forced can support. The user is User 74 and the CDF of the user's delay is seen in Figure 5.5.

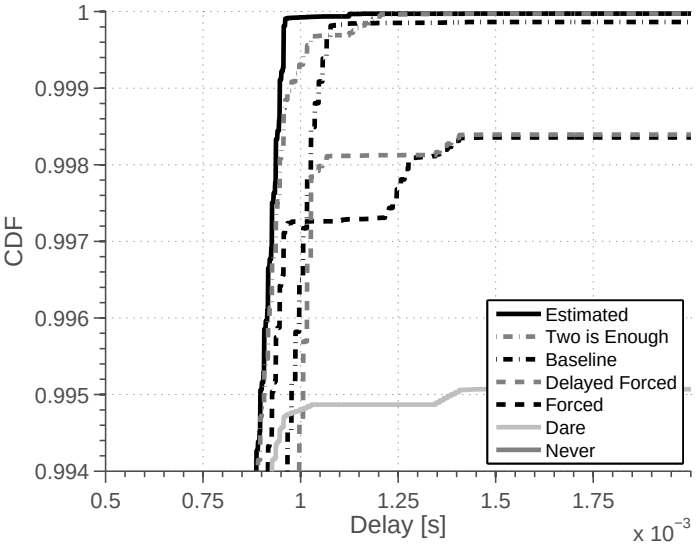


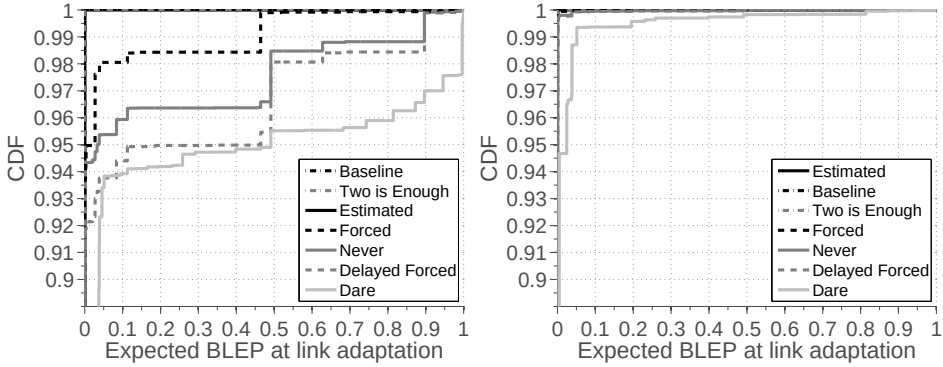
Figure 5.5: *CDF of delay for User 74.*

From Figure 5.5 it is clear that the estimating methods are close to the demand. Baseline is a little too late but Estimated and Two is Enough meet both the latency and reliability demand. The other methods, however, fall short of the reliability. Never achieves such a moderate reliability that it is not included in the plot.

All methods are aware that User 74 is a difficult user as can be seen in the high expected BLEP. The expected BLEP in Figure 5.6a, is very high for Forced, Delayed Forced, Never and Dare as compared to a user all methods can deliver to in Figure 5.6b. For example, in Figure 5.6a, the method Delayed Forced has a probability of 98% to have a BLEP lower than or equal to 0.5. In 2% of the transmissions it has a larger BLEP. On the other hand, for a user whose requirements are met, the probability of having a transmission with a BLEP larger than 0.5 is less than 0.5% as seen in Figure 5.6b.

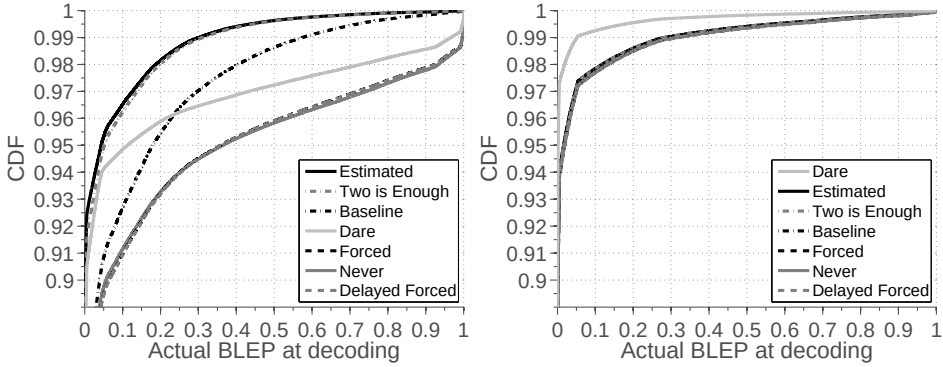
To be able to meet the reliability demand, the methods must achieve a much higher probability of a lower expected BLEP, such as Estimated, Two is Enough and Baseline do. They are barely visible in Figure 5.6, rising directly from 0 to 1 for a very low BLEP. During the studied period for User 74, the packet arrival rate is quite high but it is clear from Figure 5.5 that Forced, Delayed Forced Dare and Never cannot meet the reliability. This even though they have a low expected BLEP, that is to say they expect a high probability of error but their efforts to reduce it are not sufficient.

The actual BLEP at the decoder in Figure 5.7 shows a situation similar to the



(a) User 74 *(b) User 1*

Figure 5.6: *Expected BLEP at link adaptation for all transmissions (both original transmission and possible retransmissions) for two users.*



(a) User 74 *(b) User 1*

Figure 5.7: *Actual BLEP at decoder for all transmissions (both original transmission and possible retransmissions) for two users.*

expected in Figure 5.6, but even worse for all methods except Dare.

The estimating methods lower the expected BLEP by splitting the packet into segments while the other methods hope that a retransmission is enough. Forced also forces its retransmission. However it seems like these methods are not enough. Only segmenting into smaller segments and allocating many resources to these segments makes it possible to deliver to this user. By using an adaptive retransmission that increases the resource allocation, the other methods might also be able to deliver to this user. In addition, as seen in Section 4.1.3, Estimated can utilize all subbands and schedule more users than the other methods by scheduling a few number of subbands for a segmented packet.

5.1.4 Dare

Recall that Dare outperformed all other methods in Figure 5.1, also for a different N_{SINR}. Looking back at Figure 5.4, it is also clear that Dare achieves a very high probability of delivering packets within the latency already during the first slot. Dare outperforms the other methods since it manages to deliver more packets in the first slot and therefore issues less retransmissions that cost resources and interferes with other users. But how can it deliver more packets with its first transmission?

Dare aims for a lower target reliability with its first transmission, $1 - 10^{-1}$ instead of $1 - 10^{-3}$. Hence, one would think it would also achieve a lower target reliability than the other methods, and therefore lose more packets. However, Dare does not use any of the extra protection that for example Forced and Estimated does. Forced issues a forced retransmission and a retransmission, where the retransmission only is based on the result of the original transmission. This can cause unnecessary forced retransmissions that only causes interference to other users. Estimated can estimate up to three slots and in each slot assign resources in order to meet the target reliability.

In some cases, this extra protection is not necessary since the information is wrong. The situation for the UE seems much worse than it actually is and less resources would have sufficed. In these cases, the extra protection only adds extra interference, making it harder to deliver other users' packets.

Only transmitting the packet and retransmitting the packet if it was not decoded, as in the method Dare, was enough protection for most UEs in the simulations for low reliability. The decreased interference increases the SINR for the method Dare and improves the DL SINR difference since the UEs experience less interference from other UEs that change the channel quality from the time the channel quality was measured until the time of transmission.

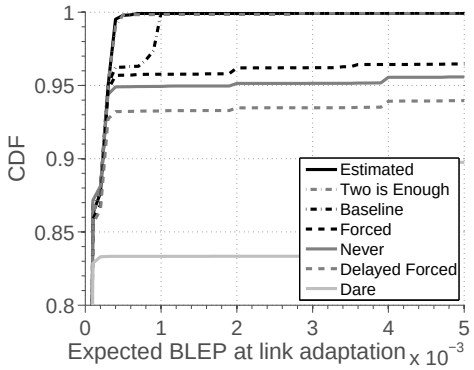


Figure 5.8: *Expected BLEP at link adaptation for all transmissions (both original transmission and possible retransmissions) for all methods for user 19 with the packet arrival rate 637 packets/s.*

In addition, many UEs have a good channel quality. Therefore, also the fewest

number of resources allocated to these UEs yields a high reliability, often much higher than $1 - 10^{-1}$. This can be seen in Figure 5.8 where the expected BLEP is plotted for all methods for one UE with one packet arrival rate. For this user, Dare has the highest expected BLEP, but still has a probability of 80% to have an expected BLEP lower than or equal to 10^{-3}.

5.2 Results for High Reliability

The settings used for retransmission are the same for moderate reliability and high reliability (except for the reliability target) and presented in Table 5.3. Most of the packet arrival rates used in the simulations are the same for both high and moderate reliability, though in high reliability an even lower lowest packet arrival rate was used, as seen in Table 5.4.

As for high reliability in single transmission, the methods Baseline and Two is Enough were not simulated for a high reliability in retransmission since they were outperformed by Estimated when simulated for moderate reliability in retransmission. In addition, the method Delayed Forced very seldom had time to make use of its delayed retransmission causing it to only be a worse version of Forced. Therefore, the methods simulated for high reliability are Estimated, Never, Forced and Dare.

Table 5.3: *Simulation parameters for retransmission, high reliability.*

Setting	$P_{e_{tot}}$	Number of users	Δ_{SINR} [dB]	N_{SINR}
Basic	10^{-4}	75	0.5	20
Long history	10^{-4}	75	0.5	1000

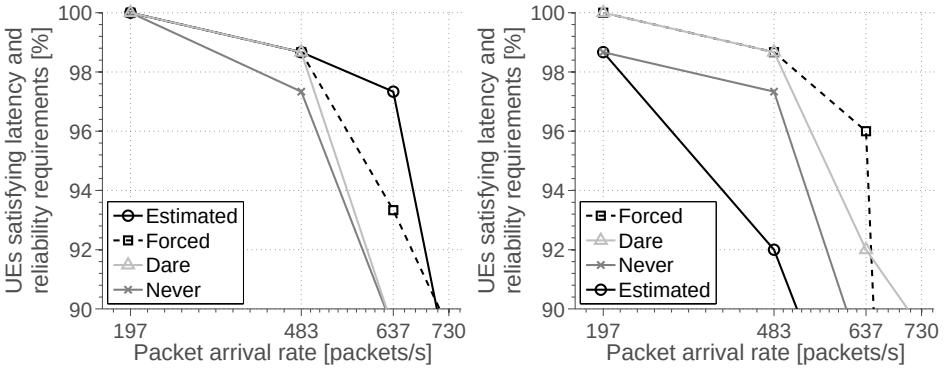
Table 5.4: *Period of packet arrival and corresponding rate of packet arrivals per user for retransmission, high reliability.*

Load	Lowest	Low	Medium	High	Higher	Highest
Period [s]	0.00507	0.00207	0.00157	0.00137	0.00127	0.00117
Rate [packets/s]	197	483	637	730	787	855

5.2.1 Comparison of History-sizes

The URLLC capacity for the retransmission scenario in high reliability, limited to 90-100% successful UEs, is plotted in Figure 5.9. From Figure 5.9 it is clear that Estimated achieves a higher URLLC capacity than Forced for the smaller history-size, $N_{SINR} = 20$. However, for the longer history-size of $N_{SINR} = 100$, Forced is the method to achieve the highest URLLC capacity while Estimated is the worst method, beaten also by Never. Never achieves the lowest URLLC capacity for $N_{SINR} = 20$ and the next-to-lowest URLLC capacity for $N_{SINR} = 1000$, beating Estimated. The method Dare that outperformed all other methods in moderate

reliability is now among the worse methods, getting the next-to-lowest URLLC capacity in $N_{\text{SINR}} = 20$ and next-to-best in $N_{\text{SINR}} = 1000$, achieving a URLLC capacity slightly higher than Never but either Estimated or Forced outperforms Dare, depending on the value of N_{SINR}.



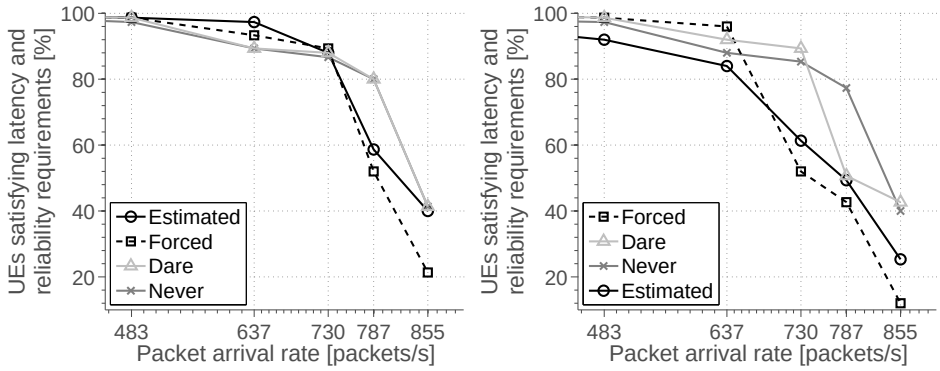
(a) $N_{SINR} = 20$ ***(b)*** $N_{SINR} = 1000$

Figure 5.9: *Comparison of URLLC capacity due to N_{SINR}, limited to the interval 90-100%.*

With $N_{\text{SINR}} = 1000$, Estimated has many users that cannot meet the latency bound, similar to the single transmission scenario in high reliability where Estimated ran out of resources when $N_{\text{SINR}} = 1000$. Forced on the other hand, mostly fails due to not meeting the targeted reliability, thus improving its performance with an increased history-size. Estimated can deliver packets successfully to these users by allocating them more subbands and using more segments, as compared to Forced, similar to retransmission in moderate reliability and single transmission in high reliability, however, these users do actually need the extra resources. When increasing the history-size to $N_{\text{SINR}} = 1000$, Estimated can deliver to some problematic users that need the extra resources, similar to Forced. However, Estimated also starts failing users since it cannot meet their latency bound.

Extending the vertical scale in the figure from 90-100% to 10-100% it can be seen that there are packet arrival rates where Dare and Never have a larger percentage of successful users than the other methods. Never and Dare use fewer resources than the other methods which makes them able to try to deliver packets for more users, even though they might fail the reliability for some users. Never was seen to use few resources in retransmission also in moderate reliability in Figure 5.2.

Dare drastically decreased in number of successful users in retransmission for moderate reliability together with the other methods in Figure 5.2. In Figure 5.10, Dare achieves as low a number of successful users for the highest packet arrival rate as in Figure 5.2, however the methods Forced and Estimated achieve an even lower number of successful users for high reliability (Figure 5.2) as compared



(a) $N_{SINR} = 20$ ***(b)*** $N_{SINR} = 1000$

Figure 5.10: *Comparison of URLLC capacity due to* N_{SINR}.

to moderate reliability (Figure 5.10). To achieve the reliability of $1 - 10^{-4}$ the other methods now spend more resources as compared to when the reliability was $1 - 10^{-3}$, while Dare spends the same amount.

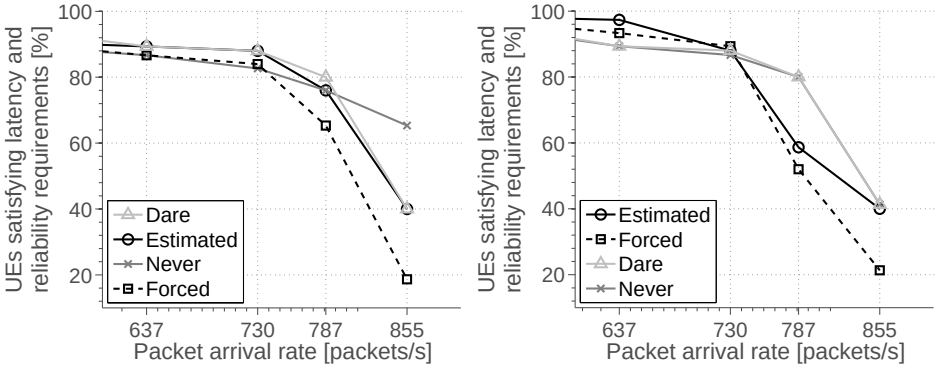
5.2.2 Dare

From the simulations for retransmission in moderate reliability it was seen that Dare outperformed all other methods, depicted in Figure 5.1. How can it be beaten in high reliability? In Figure 5.11 the URLLC capacity for both of the simulations for the targeted reliability of $1 - 10^{-3}$ and $1 - 10^{-4}$ are plotted, counting a UE successful only if it achieves a reliability of $1 - 10^{-4}$. In Figure 5.11a, the methods tried to achieve a reliability of $1 - 10^{-3}$, but plotted is only the percentage of UEs that fulfill the reliability of $1 - 10^{-4}$, for the methods also simulated in high reliability. In Figure 5.11b, the methods tried to achieve a reliability of $1 - 10^{-4}$ and the percentage of UEs that fulfill the reliability of $1 - 10^{-4}$ are plotted. In Figure 5.11, it can be seen that both Estimated and Forced improve their URLLC capacity from Figure 5.11a to Figure 5.11b for a lower packet arrival rate. When targeting $1 - 10^{-4}$, more UEs achieve $1 - 10^{-4}$. For a higher packet arrival rate, both methods run out of resources faster with a higher targeted reliability. However, quite many UEs in Figure 5.11a also achieve the reliability of $1 - 10^{-4}$, indicating a waste of resources since $1 - 10^{-3}$ would have sufficed and cost less resources.

Dare, on the other hand, achieves the same capacity in Figure 5.11a and Figure 5.11b. As described in the Section 3.3.7, Dare was meant to use a lower target reliability of $1 - 10^{-1}$ for its first transmission and a higher reliability for its retransmission. However, due to an implementation miss the number of used subbands is not increased for the retransmission, causing Dare to use the same number of subbands for its initial transmission and its retransmission. In moderate reliability, this was enough to deliver the packets to most users.

Most retransmissions are needed when the expected BLEP is lower than the target BLEP but wrong, the UE expects the packet to be delivered but the channel quality was worse than estimated and thus the packet lost. In these cases, since the expected BLEP is lower than the target BLEP, the resources should not be increased for the retransmission. When a retransmission is needed and the expected BLEP is higher than the target BLEP, the number of subbands should however increase. Due to the implementation miss, the number of subbands do not increase but when the two transmissions are combined in the decoder, the two transmissions are often enough to be able to decode the packet.

When increasing the reliability to $1 - 10^{-4}$ from $1 - 10^{-3}$, the number of subbands used for Dare is the same but the reliability demand stricter and thus more users fail. If Dare increased the number of used subbands for its retransmission, its URLLC capacity might improve.



(a) *Target BLEP during simulation, $P_{e_{tot}} = 10^{-3}$, URLLC capacity plotted with target $P_{e_{tot}} = 10^{-4}$*

(b) *Target BLEP during simulation, $P_{e_{tot}} = 10^{-4}$, URLLC capacity plotted with target $P_{e_{tot}} = 10^{-4}$*

Figure 5.11: *Comparison of URLLC capacity for different simulated $P_{e_{tot}}$.*

6 Discussion

This chapter discusses the achieved results, the method used to obtain the results, and looks at the thesis from a wider perspective.

6.1 Results

In Chapter 4 and 5, the results of the study were presented. In this section, the major results and some noteworthy discoveries are discussed.

6.1.1 Single Transmission

The results presented in Section 4.1 fit well with the theory in most aspects. For example, the network can achieve a higher URLLC capacity by adding extra protection such as segmentation, as in the method Estimated, or forced retransmissions, as in the method Forced, compared to just transmitting as in the method Never.

However, the effect of not having an outer loop and the values of the parameters Δ_{SINR} and N_{SINR} had a greater effect than anticipated. As seen in Figure 4.1, the setting of N_{SINR} can improve a method to reach a higher URLLC capacity, as for the method Estimated or greatly reduce the number of users that fulfill the requirements for higher packet arrival rate, as for the methods Forced, Baseline and Two is Enough. The parameters must be chosen to strike a balance between not achieving the reliability due to misinformation (too small Δ_{SINR} or N_{SINR}, causing the user to assign too few resources) and not achieving the latency due to resource shortage (too large Δ_{SINR} or N_{SINR}, causing the user to assign too many resources).

The methods Estimated, Two is Enough and Baseline schedule more subbands and thus use more resources than the methods Never and Forced. However, they

also utilize the resources more efficiently by being able to make use of all available subbands, letting a user use the few remaining ones in one slot and then assigning the user more subbands in the upcoming slot.

Moderate Reliability

For moderate reliability, Forced can achieve a higher URLLC capacity than Estimated when N_{SINR} is small as seen in Figure 4.1a, however Estimated can achieve a reasonable percentage of successful users within the interval 90-100% also for quite high packet arrival rate while Forced drops down to a percentage of successful users of 50%, as seen in Figure 4.3. Estimated and Forced outperform the other methods, as seen in Figure 4.1 and Figure 4.2.

High Reliability

Increasing the reliability, Estimated is still the best method for very high packet arrival rate but now only achieves 30% successful users compared to 10% successful users for Forced as seen in Figure 4.14. In moderate reliability Estimated achieved a much higher percentage, 90% successful users while the other methods only could achieve 50% successful users for very high packet arrival rate as seen in Figure 4.3.

The parameters had an even greater effect in high reliability where for some settings Estimated got outperformed also by Never, depicted in Figure 4.13. Forced achieves a higher URLLC capacity than Estimated for almost all settings of parameters and especially for medium packet arrival rate. To over-assign resources with a large N_{SINR} and Δ_{SINR} and then use a method that uses a simple segmentation that adds the same amount of reliability to all as in the method Forced seems to work a long way.

For Estimated, N_{SINR} and Δ_{SINR} are needed to adjust the estimates from some users that have optimistic views on their channel quality. This gives the method a chance to allocate enough resources also to these users but to over-assign resources to users that already had a good channel quality estimate or even a pessimistic channel quality estimate that became even further pessimistic with the use of N_{SINR} and Δ_{SINR}. This over-assignment then causes additional interference to other users, thus needing even larger N_{SINR} and Δ_{SINR} in order to allocate enough resources since the actual SINR decreases but our estimate remains roughly the same. In this manner, Estimated is struck harder than Forced from over-assignments since Forced only assigns a retransmission while Estimated can spend several slots assigning multiple subbands in order to reach the target reliability.

As Estimated makes use of several slots and its over-assignment can lead to a resource shortage and delayed packets, the method benefits from an extended latency bound as seen in Figure 4.22. By forbidding the transmission of segments in the fourth slot, this can be slightly improved.

Estimated in High Load

The fact that Estimated can achieve such a high URLLC capacity while many other methods do not for high packet arrival rate seems to be due to its risk-taking in the last slot. Estimated is the only method that keeps track of the age of the packet and how many slots it can use within the latency, allowing a *higher probability of error* if the slot is the *last within the latency bound*. Other methods might wait until the fourth slot to transmit their packet, which due to scheduling latency is outside of the latency bound. This can be seen in Figure 4.3 where also Two is Enough and Baseline drastically decrease in the number of successful users. Those methods have a similar resource usage in number of subbands and number of scheduled users but one difference is that Estimated takes a risk in the third slot. The other methods do not and get many delayed packets, arriving just outside of the latency bound.

Effect of Following More Correct Error Probability

Another conclusion is that even though Estimated more closely follows a correct, very low error probability, the method does not get a higher URLLC capacity than the method Forced which does not follow a correct error probability. Even though the estimation follows a correct BLEP better, the actual BLEP at decoding is different due to the unpredictable interference from other users. If the channel knowledge was more accurate, or the user had more information about the situation of the UE, all methods would improve their URLLC capacity.

6.1.2 Retransmission

For moderate reliability, in Figure 5.1 it was seen that Dare achieves a high URLLC capacity and Estimated becomes as good with a larger N_{SINR}. Forced is not far behind Estimated in achieved URLLC capacity, especially when using a larger N_{SINR}, as seen in Figure 5.1.

For high reliability, Dare does not achieve a very high URLLC capacity, as seen in Figure 5.9. Instead, for a smaller N_{SINR} Estimated achieves the highest URLLC capacity and for a larger N_{SINR} Forced achieves the highest URLLC capacity. With a smaller N_{SINR} Estimated can achieve a high percentage of successful users also for higher packet arrival rates, depicted in Figure 5.1.

In the retransmission scenario some users fail due to few resources, leading to scheduling in later slots and retransmissions that arrive outside of the latency bound. There are also users that fail due to not meeting the reliability demand. Some of the users that struggle meeting the reliability demand can meet it using the method Estimated and sometimes Two is Enough.

Estimated only takes retransmissions into account when it is not segmenting. An improvement to Estimated could be take retransmissions into account also when segmenting. If three segments seems to be needed, the method can compare using three segments with the alternative to only use two segments and use a retransmission for the first segment. The alternative that achieves the highest expected reliability or uses the fewest resources can then be chosen.

Dare

It was better than expected to transmit at a lower target reliability in both the first transmission and the retransmission as Dare does for a simulated moderate reliability. For a high reliability, the method was worse compared to the other methods. With a high reliability, two transmissions with low target reliability is not always enough to deliver the packets and the methods Forced and Estimated that spend more resources on problematic users achieve a higher URLLC capacity. If the targeted reliability of the retransmission was adjusted to $1 - 10^{-4}$, and the number of subbands increased for the retransmission, Dare might improve. This was the idea from the beginning but a bug was discovered in the implementation towards the end.

6.1.3 Comparison of Estimated and Forced

In the single transmission scenario with moderate reliability, Forced achieved a higher URLLC capacity for a smaller N_{SINR} but Estimated became as good as Forced with a larger N_{SINR}, as seen in Figure 4.1. For high reliability in single transmission, Figures 4.15 and 4.16 show how Forced outperforms Estimated across all tested parameter settings.

In the retransmission scenario with moderate reliability Estimated is beaten by Dare but achieves a higher URLLC capacity than Forced for both tested parameter settings, as seen in Figure 5.1. For high reliability, Estimated beats Forced with a smaller N_{SINR} but is beaten by Forced with a larger N_{SINR}.

How can Estimated be better than Forced in the retransmission scenario but worse than Forced in the single transmission scenario? In both scenarios, Estimated does not succeed for some users due to missing the reliability, except in high reliability with a large N_{SINR} where it instead is the latency. Focusing on a smaller N_{SINR}, Estimated's problem in single transmission is that it can adapt how much to segment for a user based on the provided information but the information is not very reliable. This can lead to an optimistic segmentation, sending only a fraction of the total packet in the second slot and most in the first slot, but if the channel quality is worse than estimated, the first slot might be lost. It could also lead to a pessimistic segmentation, over-assigning resources and sending a larger fraction in the second slot, causing unnecessary interference to other users. Forced, on the other hand, only detects *if* segmentation is needed and does not estimate *how much* segmentation that is needed, issuing a forced retransmission when it detects that segmentation is needed. Often, this rough estimate is enough to achieve a high URLLC capacity.

In the retransmission scenario, if Estimated performs an optimistic segmentation and loses a segment in the first slot, there is time to retransmit that segment in the third slot. Therefore, Estimated is able to deliver packets successfully to more users in the retransmission scenario, compared to the single transmission scenario. The number of segments a packet is split into when segmenting is mostly two, and therefore Estimated can benefit from the retransmission also when segmenting. Estimated can also deliver to more problematic users than

Forced by segmenting into several slots, which was needed for some users in the retransmission scenario.

6.1.4 Comparison of Single Transmission and Retransmission

The thesis aim is to evaluate methods for segmentation in single transmission and in retransmission, not to evaluate whether single transmission or retransmission is best suited for URLLC. However, it is an interesting question and the results allow for some discussion on it.

Comparing the achieved URLLC capacity between retransmission and single transmission in high reliability, it is clear that a similar number of successful users is achieved for lower packet arrival rates. For the method Forced, the settings with large back-off and history-size achieve a slightly higher number of successful users in single transmission than in retransmission, and the settings with smaller back-off and history-size achieve a slightly lower number of successful users in single transmission than in retransmission. However, the settings with large back-off and history-size run in high reliability, single transmission have not been run in retransmission. For the method Estimated, the number of successful users is larger in the retransmission scenario for all tested settings.

Since the results are similar, it is hard to see a clear winner, even though retransmission seems slightly better, especially for Estimated. In addition, in these simulations, the retransmission scenario does not use the last two OFDM symbols in DL in order to use the fast HARQ. The retransmission scenario has fewer available resources than the single transmission scenario. In another fast HARQ setup, these two symbols could also be used for DL by transmitting the acknowledgment on a different frequency band, but the fast HARQ acknowledgment would then only be based on the result of the first five OFDM symbols and not the entire packet.

Also, if using a smaller slot duration of 0.125 ms, eight slots would fit within the latency, where a medium-speed HARQ can be used which issues a retransmission four slots after the original transmission. The medium-speed HARQ is not as fast as fast HARQ but faster than normal HARQ, enabled by simply transmitting acknowledgments earlier but not sacrificing OFDM symbols to do so. This would also increase the number of resources available since no OFDM symbols are spent on transmitting a fast HARQ.

In addition, this thesis has shown that one of the difficulties in link-adaptation and segmentation is the channel knowledge. Retransmissions offer more certain information than CQI values, an attempt can be made to transmit a packet and based on the acknowledgment the base station knows if it was delivered or not and can try again. Therefore, retransmissions are very interesting in order to enable URLLC.

6.1.5 System Model

This section discusses some of the mechanisms used in the simulation and what advantages and disadvantages different methods have in regards to these mecha-

nisms.

Outer Loop

An alternative to the parameters N_{SINR} and Δ_{SINR} would be an outer loop. However, since Estimated changes the target reliability for its transmissions, this would require a different kind of outer loop that can be signaled the current target reliability and adjust the back-off with regard to this. Forced on the other hand would function with a normal outer loop.

Control channel

In the simulations, an ideal control channel has been used in both UL and DL. This means that the acknowledgments (ACKs and NAKs) from the UE always are received and decoded correctly. In addition, the scheduling grants and the CQI values sent to the UE are also always received and decoded correctly. If one were to introduce errors in the control channels some messages such as: acknowledgments, scheduling grants, and CQI values may be missed or misinterpreted. However, the probability to misinterpret CQI values or scheduling grants is very low.

If a non-ideal control channel is used, the methods might have to aim for an even higher reliability than the target reliability. Then, the number of retransmissions can be decreased, and therefore the number of missed acknowledgments are also decreased. Storing the received SINR from the CQI in some way might also be beneficial. If a CQI value is missed, the history can be used to perhaps interpolate a SINR value instead of using the older SINR value.

6.2 Method

The method of this thesis has been to design, implement, simulate and evaluate a number of proposed methods for segmentation. One limitation of the study is that the targeted reliability of 10^{-5} never was simulated. This would require 10 to 100 times longer simulation time in order to get reliable results and was left out due to this. However, both the reliabilities 10^{-3} and 10^{-4} were simulated. 10^{-3} is a much smaller error probability than 10^{-1} that is used in LTE-A today, and 10^{-4} is even smaller. Therefore, these values should give a good indication of how the methods would work when simulated with an error probability of 10^{-5}, especially in the change from error probability 10^{-3} to 10^{-4}. The effects observed when decreasing the error probability to 10^{-4} from 10^{-3} are probably even more noticeable in a simulation with target error probability 10^{-5}. In addition, 10^{-5} is the suggested target for certain scenarios. In other scenarios other target error probabilities are also of interest.

A longer simulation time would also have been beneficial. Now 100 error events per user for 10^{-3} and only 10 error events per user for 10^{-4} was used. However with these number of error events, the simulation time was already roughly 40 hours for three different packet arrival rates, all methods and one

setting of history-size and back-off. In order to have time to find fitting load and study different parameters and loads this was deemed as long enough.

Another limitation is the assumption of independent transmissions. The transmissions are probably not independent but it is a channel modeling problem in its own right that requires many measurements to model how the transmissions depend on each other and then model their dependence. This assumption was thought of as a good enough model of the transmissions. If it is not, the method Estimated targets the wrong target reliability for its segments. This could either lead to the method aiming for a higher total reliability for its packets, spending unnecessary resources on it, or aiming for a lower total reliability for its packets, which would yield a lower URLLC capacity since the packets are delivered at a too moderate reliability.

Also, the simulator does not simulate URLLC, since URLLC in NR has not been entirely defined yet. Instead it simulates an LTE-A mobile network with additions for URLLC. The results achieved in this study might therefore not entirely apply to URLLC in its final form. The methods should still be comparable to each other and the insights from this should still be valuable and applicable to URLLC. This is also the same simulator used by the researchers at Ericsson to research URLLC, and therefore it is reasonable to believe that it simulates URLLC appropriately.

There are many parameters that can be changed or tuned in this thesis as well. One limitation is to only have studied the behavior of some of these. One parameter that would be interesting to study is the size of the packets. Now only 400 bits was simulated but 3GPP has recommended simulating packets of size 256, 400 and 1600 bits. It would also be interesting to study a shorter TTI duration, such as 0.125 ms. Then 8 slots would fit within the latency, which opens up for more possibilities for the methods. Estimated could segment into as many as seven slots instead of the current three and Delayed Forced could afford its forced retransmission, or perhaps multiple forced retransmissions. At the same time, perhaps too many parameters, methods and scenarios have been studied. It would have been better to focus on a select few and analyze those in detail.

6.3 The Thesis from a Wider Perspective

5G is more than an enhanced mobile network, as mentioned in the introduction by E. Dahlman et al. [1] and the ITU [2] both of whom envision a networked society. In this society, not only are mobile phones connected but traffic sensors, robots in factories and household appliances. This thesis has studied a small part of the enormous and ambitious project that is 5G. Since 5G is such a big project, it needs to consist of a multitude of small projects that study different aspects.

The results of the thesis show that if a user's estimate of the channel indicates that it cannot fit all of its packet within one slot and achieve its targeted reliability, something should be done. Exactly what depends on the scenario, such as if there are retransmissions available and how heavily loaded the network is.

These results can be used as guidelines for the continued work of specifying, researching and implementing URLLC. In addition, the study can be used to iden-

tify new methods or areas to study to further increase the performance of URLLC.

The main purpose of the study has been to increase the URLLC capacity by evaluating different methods. This is an important measure for URLLC since if one base station can support more users, this will result in a cheaper network to build and run, by reducing the total number of needed base stations.

7

Conclusion

This chapter concludes the thesis by looking back at its purpose to see if it has been fulfilled, and by giving answers to the questions stated in the problem formulation in Section 1.3. After that, ideas for future work within the area of URLLC are proposed.

7.1 Answers to the Problem Formulation

The goal of the thesis is to come up with guidelines for how segmentation should be implemented for URLLC and get insights into what URLLC benefits from, both when packets must be delivered in a single transmission and when the users have one retransmission available. Insights have been gained from the simulations, which are presented in Chapters 4 and 5, discussed in Chapter 6, and summarized in this chapter. The answer to the second question of the problem formulation, *"What way of segmentation yields best results with respect to capacity and resource efficiency?"*can be seen as a guideline for implementation of segmentation in URLLC.

- How can segmentation be handled in URLLC?

  One way of handling segmentation is to transmit anyway and hope for the best. This was studied by the method Never. Another way is to split the packet into segments and adjust each segment's error probability so that the total error probability of the packet is correct. This was studied by the method Estimated and partially by the methods Two is Enough and Baseline. A third way of handling segmentation is to not segment, but force a retransmission, sending the same packet twice without waiting for an acknowledgment from the UE. A fourth way of handling segmentation, which

only works in a scenario where the user has time to wait for and use a retransmission, is to transmit at different reliability targets, as seen in the method Dare.

- What way of segmentation yields best results with respect to capacity and resource efficiency?

  No method clearly outperforms all others over all scenarios. To just transmit is very rarely a good idea so to actually do something about segmentation seems to be preferable but what to do depends on the scenario. Firstly, if there are no retransmissions available, Estimated and Forced are the methods that achieve the highest URLLC capacity. If the packet arrival rate is high, Estimated is to prefer. However, this is due to Estimated taking a risk in the third slot. If Forced also took a similar risk, the method might improve for higher packet arrival rates.

  For lower packet arrival rates, Forced and Estimated are both able to achieve a high URLLC capacity but when simulated with a higher reliability, Estimated cannot achieve a higher URLLC capacity than Forced. This seems to be due to the fact that the method uses too many resources on users that do not need it and therefore runs out of resources and causes unnecessary interference to other users. Forced, on the other hand, assigns more resources to the same users that Estimated do but not as much. However, it turns out to be just enough resources in most cases. Therefore, Forced is to prefer. However, if the information about the users' quality could be more accurate, Estimated might perform better than Forced.

  If there is time for a retransmission, both Estimated and Forced achieve a high URLLC capacity for both low and high reliability and the method to achieve the highest URLLC capacity depends on the history-size. With a small history-size Estimated is best while with a large history-size Forced is better.

  For moderate reliability, Dare outperforms all other methods, however, for a higher reliability Estimated and Forced achieve a higher URLLC capacity. If Dare would increase the number of subbands for the retransmission in order to meet the target reliability, it might improve and achieve a higher URLLC capacity also for higher reliability. In its current state, it is beaten by the methods Forced and Estimated but there is a lot of potential for the method.

  Estimated can achieve a high URLLC capacity for a small history-size, also for higher packet arrival rates. Also, a smaller history-size means less wasted resources and thus a more resource efficient method. This makes Estimated the preferred method in retransmission. However, Estimated only beats Forced by a couple of users in retransmission with high reliability for a small history-size. Therefore, even if Estimated seems like the best method in retransmission, Forced is also a very good candidate. Since Estimated is able to fulfill the requirements for some problematic UEs with low SINR,

but Forced cannot, another alternative would be to use the method Forced for the majority of the UEs and Estimated for UEs with low SINR.

7.2 Future Work

This study has explored some aspects of link adaptation and segmentation in URLLC. One of the surprises was the effect of the history-size and back-off parameters. The values of these two parameters greatly affected the result. Therefore, it would be interesting to study the CQI and acknowledgments sent from the UE to the base station and how this is used in the outer loop. In addition, it would be interesting to study what more information the UE could send to the base station in order to improve the link adaptation.

One alternative would be to update the existing outer loop so that it can be notified when the link adaptation uses a different reliability target, such as Estimated and Dare do. This could then be compared to using the history-size and back-off parameters in order to see which method is better. Since it is an outer loop, it still suffers from very few NAK and will converge very slowly.

Instead of setting the history-size and back-off parameters for each simulation, the parameters could be set by the link adaptation algorithm itself. This might be hard to do properly, but is probably better than doing it by hand for every simulation. From the results it was clear that for the methods Estimated and Forced, the URLLC capacity increased with larger history-size as long as the packet arrival rate was moderate. If the base station starts running out of resources it will not be able to meet the latency demand for a large number of users since they are not scheduled at all or too late. In this case, it is better to use less resources per user and let some fail due to reliability but most users succeed. However, we are mostly interested in the packet arrival rates that can yield 100% successful users, which can only be achieved for a lower packet arrival rate. For a lower packet arrival rate, the base station should increase the history-size in order to meet the reliability demand.

The base station should be able to check if it can schedule all users' packets within the first slot of their arrival, and if not, decrease its history-size, thereby using less resources and become able to schedule more users. On the other hand, if it has resources left over, it can increase the history-size in order to get a more reliable transmission. The history-size should in addition be limited to some interval, perhaps no smaller than 20 and no larger than 1000. The back-off can be fixed to a value depending on the targeted reliability.

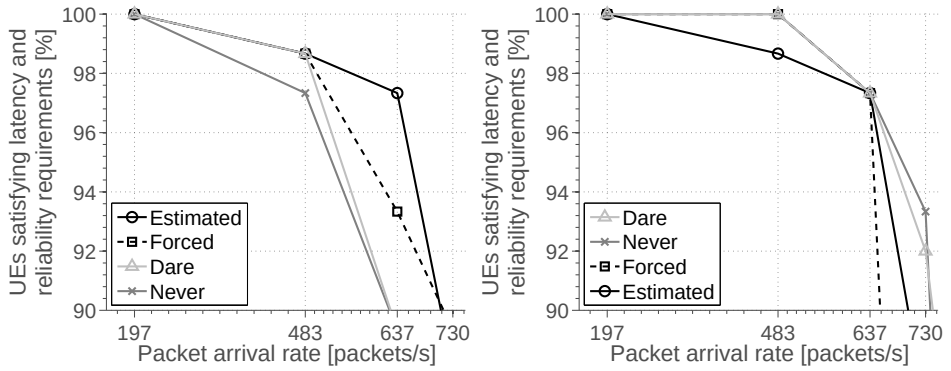
Also, taking the lowest SINR in the stored SINR history is a very rough implementation. Instead, the history could be used to get the lowest SINR and a probability for how often the SINR is this low. If the probability to have such a low SINR is very low, perhaps the average SINR or an SINR slightly larger than the lowest SINR can be used instead.

A third option, that might be combined with the modified outer loop, is to report more information from the UE to the base station. For example, the UE could estimate how much higher SINR that is needed to decode the packet or how

much lower SINR that also would suffice to decode the packet. This value tells the base station how far off its estimate was for this particular packet. ACKs and NAKs are essentially the same feedback as this SINR estimate, but instead of using one bit, two bits could be used to describe how close to decoding the packet the UE was instead of a simple Yes or No.

Towards the end of the thesis, an implementation miss was discovered. The number of subbands were not increased for retransmissions. This especially affects the method Dare, but also the method Forced. These methods would both behave differently if they increased the number of used subbands in the retransmission in order to meet the reliability target. It would be interesting to see how these methods behaved and if they would be able to achieve a higher URLLC capacity when the retransmissions can increase the number of used subbands, especially for the method Dare.

The implementation miss was corrected and one simulation run to get some results for Dare in retransmission when the number of subbands are increased for the retransmission. The simulation was run with target reliability $1 - 10^{-4}$. This simulation indicates that the methods Dare and Never greatly improve their capacity when the number of subbands are increased for the retransmission in the retransmission scenario, as seen in Figure 7.1. This indicates that when feedback can be received from a retransmission it is beneficial to only use that feedback since it is more correct than the reported SINR. However, a more thorough study is needed to investigate further.



(a) *Non-adaptive HARQ* ***(b)*** *Adaptive HARQ*

Figure 7.1: *Early results with increased number of subbands for retransmissions*

Another method that would be interesting to test would be a *Dare Delayed Forced in an eight slot scenario*. With eight slots, a medium-speed HARQ can be used which enables a retransmission 4 slots after the original transmission. Dare can deliver to 90% of the users. The other 10% get a retransmission and since the retransmission is in the fourth slot there is also time for one or several forced retransmissions. Often, many users have a good quality and are satisfied with

one transmission. The problematic users are discovered with the retransmission and if they have a very bad expected BLEP, extra protection in the form of forced retransmissions can be added.

List of Figures

List of Tables

Bibliography

[1] E. Dahlman *et al.*, "5G wireless access: Requirements and realization," *IEEE Commun. Mag.*, vol. 52, pp. 42 – 47, Dec. 2014. Cited on pages 1, 13, and 87.

[2] ITU-R, "Framework and overall objectives of the future development of IMT for 2020 and beyond," Recommendation ITU-R M.2083, ITU, Sept. 2015. Cited on pages 2 and 87.

[3] N. Johansson *et al.*, "Radio access for ultra-reliable and low-latency 5Gcommunications," in *2015 IEEE International Conf. on Communication Workshop (ICCW)*, pp. 1184 – 1189, 2015. Cited on pages 2 and 32.

[4] 3GPP, "Technical specification group services and system aspects; Feasibility study on new services and markets technology enablers for critical communications; Stage 1," TR 22.862, Rel-14 V.14.1.0, 3GPP, Sept. 2016. Cited on pages 2, 15, and 17.

[5] E. Dahlman *et al.*, *4G: LTE/LTE-Advanced for Mobile Broadband*. Elsevier Science, 2011. Cited on pages 2, 5, 6, 8, 9, 10, 11, 12, 13, 14, and 17.

[6] 3GPP, "Technical specification group radio access network; Study on new radio (NR) access technology physical layer aspects," TR 38.802, Rel-14 V.1.1.0, 3GPP, Feb. 2017. Cited on pages 3, 5, 16, 17, 18, and 26.

[7] G. Americas, "Wireless technology evolution towards 5G: 3GPP release 13 to release 15 and beyond," white paper, 5G Americas, Feb. 2017. Cited on pages 5, 14, and 19.

[8] L. Ahlin *et al.*, *Principles of Wireless Communication*. Studentlitteratur, 2015. Cited on page 7.

[9] X. Chen *et al.*, "A novel CQI calculation scheme in LTE/LTE-A systems," in *2011 International Conf. on Wireless Communications and Signal Processing (WCSP)*, pp. 1–5, Nov. 2011. Cited on page 10.

[10] A. Durán *et al.*, "Self-optimization algorithm for outer loop link adaptation in LTE," *IEEE Commun. Lett.*, vol. 19, pp. 2005–2008, Nov. 2015. Cited on page 11.

[11] ITU-R, "Workplan, timeline, process and deliverables for the future development of IMT." Available: `http://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2020/Documents/Anticipated-Time-Schedule.pdf`, 2015. [Online 2017-05-11]. Cited on page 14.

[12] S. Nagata, "Final report of 3GPP TSG RAN WG1 #86," Chairman's notes R1-1608562, v.1.0.0, 3GPP, Aug. 2016. Cited on page 15.

[13] S. Nagata, "Final report of 3GPP TSG RAN WG1 #86bis," Chairman's notes R1-1611081, v.1.0.0, 3GPP, Nov. 2016. Cited on page 15.

[14] S. Nagata, "Final report of 3GPP TSG RAN WG1 #87," Chairman's notes R1-1701552, v.1.0.0, 3GPP, Jan. 2017. Cited on page 15.

[15] S. Nagata, "Final report of 3GPP TSG RAN WG1 #AH1_NR," Chairman's notes R1-1701553, v.1.0.0, 3GPP, Feb. 2017. Cited on page 15.

[16] B. Holfeld *et al.*, "Wireless communication for factory automation: an opportunity for LTE and 5G systems," *IEEE Commun. Mag.*, vol. 54, pp. 36–43, June 2016. Cited on pages 15 and 16.

[17] 3GPP, "Technical specification group radio access network; Study on scenarios and requirements for next generation access technologies;," TR 38.913, Rel-14 v.14.1.0, 3GPP, Dec. 2016. Cited on pages 16 and 17.

[18] S. Nagata, "Final report of 3GPP TSG RAN WG1 #85," Chairman's notes R1-166056, v.1.0.0, 3GPP, May 2016. Cited on page 17.

[19] A. A. Zaidi *et al.*, "Waveform and numerology to support 5G services and requirements," *IEEE Commun. Mag.*, vol. 54, pp. 90–98, Nov. 2016. Cited on page 19.

[20] M. Fréchet, "Généralisation du théorème des probabilités totales," *Fundamenta Mathematicae*, vol. 25, no. 1, pp. 379–387, 1935. Cited on page 22.

[21] Q. Incorporated, "HARQ design for URLLC," Proposal R1-1612079, Qualcomm Incorporated, Nov. 2016. Cited on page 24.

[22] A.-L. S. B. Nokia, "Discussion on HARQ support for URLLC," Proposal R1-1612246, Nokia, Alcatel-Lucent Shanghai Bell, Nov. 2016. Cited on page 24.

[23] 3GPP, "Technical specification group radio access network; Study on 3D channel model for LTE," TR 36.873 Rel-12 v.12.4.0, 3GPP, Mar. 2017. Cited on page 27.

[24] I. Corporation, "URLLC link adaptation aspects," Proposal R1-1612584, Intel Corporation, Nov. 2016. Cited on page 44.