

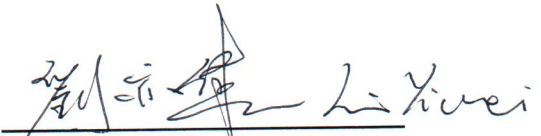


DOCTORATE IN APPLIED ELECTRONICS

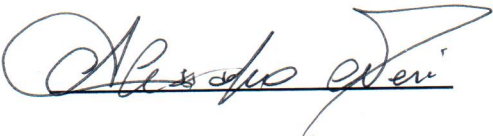
XXIX OF DOCTORATE CYCLE

**Advanced Mobile Access Routing
for Rail Traffic Management
System**

PhD student: **Yiwei Liu**



Tutor: **Prof. Alessandro Neri**



Coordinator: **Prof. Enrico Silva**





UNIVERSITÀ DEGLI STUDI ROMA TRE

Dipartimento di Ingegneria
Corso di Dottore in Elettronica Applicata

Tesi Di Dottore

Advanced Mobile Access Routing for Rail
Traffic Management System

Laureando

Yiwei Liu

Matricola 18522

Relatore

Prof. Alessandro Neri

Anno Accademico 2016/2017

Acknowledgment

First of all, I would like to thank my family, especially to my mother and my husband. Without their supporting, I can not finish my Ph.D study. Then I want to thank all my friends in Radiolabs. We are working together and I feel very happy to stay with them. Agostino Ruggeri and Andrea Coluccia are my best friends in Radiolabs, I have learned a lot from them, both on the coding skills and the Italian language. Further more, I would like to thank professor Neri and professor Carli, they taught me a lot on the theories and algorithms in my research topic and how to carry out researching work. Finally I want to give my gratitude to all colleges in Roma Tre, thank you for the three years companion and help.

Introduction

Railway transportation plays an important role in our daily lives. European Rail Traffic Management System (ERTMS)[1] is the current used railway control protocol, which is the most advanced standard for managing and controlling railway traffic. It has two main components, the European Train Control System (ETCS) and the GSM-R(Global System for Mobile Communications-Railways) for railway operations [2].

There are four main services carried by GSM-R: control information and signaling data exchanging between train and control center, voice communication between train driver and control center staff, regular voice communication between control center staff and railway maintenance workers, and emergency call between train driver and railway maintenance workers.

However, exploiting old GSM technology has reached its technological limits (*i.e.*, technological obsolescence). On one hand, there are adjacent channels interference among GSM-R and public mobile networks, and limitation of transmission capacity for GSM-R technology itself. On the other hand, extension to secondary lines and freight trains of safe and efficient traffic management systems, originally designed for high speed trains, requires cost effective solutions for both train localization and communications. For this reason, we propose two alternative solutions to support both signaling data exchanging and voice services. In particular, we use Multipath TCP (MPTCP) solution to sustain the control information and signaling exchanging service, whilst Multipath UDP (MPUDP) solution to support all kinds of voice services. It is worth to mention that both two solutions do not request the dedicate GSM-R networks. The traffics are always delivered over public cellular networks and satellite communication network.

In our new MPTCP solution, data packets delivery is supported by both the best

effort networks (*i.e.* Public Land Mobile Networks– PLMNs) and the QoS (Quality of Service) guaranteed network (*i.e.* satellite network). These two kinds of public networks are cohere by the MPTCP protocol, with the aim of providing handover procedures for seamless connectivity. In practice, once the communication between train and control center is built up, more than one subflows will involve in the data transmission. According to our emulated results, our MPTCP solution can well support the signaling data transmission with high reliability.

For the train control voice service, we use multiple UDP links to support the IP based voice packets transmission. UDP packets transported via different paths may have diverse channel delays and packet loss rate. Thus, receiver suffers from a heavy out of order problem. To solve both the out of order and packet losing problems, we introduce RaptorQ protocol in our MPUDP solution. RaptorQ can bring in the redundancy during encode procedure and can execute decode procedure regardless of the packets order. From our test results, we found that RaptorQ rise up packet loss tolerant, and settled the out of order problem, too.

Video camera starts to be used in the train positioning service in order to achieve a higher level of positioning accuracy. Meanwhile, with the increasing train speed, real time video monitor for supervising driving conditions, and video call service between driver and control center could be also required in the train control system.

Up to now, our architectures only service for the data and voice transmission of railway control system, but in the future, they should bear the railway driving monitor or position reference video transmission and on board public services as well. In order to guarantee the transmitted video quality in control system as well as to achieve a high quality of experience for train passengers, train operators need some methods to monitor the real time transmitting video quality (VQ). In fact, the on board VQ evaluation should request as less reference information from the original video as possible.

Thus, we design a no reference VQ metric NR_{ALM} for IP based video transmission. It estimates the on board VQ based on only information from received video. Prediction VQ score is computed via pooling all spatial and temporal degradations by means of log-logistic model.

To summarize, We propose two communication architectures based on the best

effort networks and satellite network, they service for the data and voice transmission, respectively. We give out a no reference video quality assessment method, too. It helps the train operator to achieve a high QoE for the passengers. According to our test results, all of our achievements have outstanding performances.

Contents

- Acknowledgment** **ii**

- Introduction** **iii**

- Contents** **vi**

- List of Figures** **x**

- List of Tables** **xiv**

- 1 Introduction of European Rail Traffic Management System/European Train Control System** **1**
 - 1.1 European Rail Traffic Management System/European Train Control System overview 2
 - 1.2 ERTMS/ETCS levels 2
 - 1.3 EuroRadio protocol for ERTMS/ETCS safety and security 6
 - 1.4 Disadvantages of the current GSM-R system and its defects 7
 - 1.5 Equipment manufacturer for ERTMS and ETCS systems 10
 - 1.6 Costs and reduce for railway operation 11

- 2 Train control signaling data transmission solution based on Multipath TCP links** **13**
 - 2.1 Introduction 13
 - 2.2 Overview of multipath TCP protocol 18
 - 2.3 New architecture of train control communication system 22
 - 2.4 Paths adding and dropping strategies based on new architecture 24

2.5	Paths adding & dropping strategies evaluation on proposed architecture	33
2.5.1	Evaluation scenarios design	33
2.5.2	Adding and dropping strategies description	34
2.5.3	Testbed software and hardware setting	35
2.5.4	Obtained Results	36
2.6	Further studies on MPTCP performance	37
2.6.1	Static Experiments	38
2.6.2	Dynamic Experiments	43
2.7	Conclusions	45
3	Train control voice service supporting based on Multipath UDP links	48
3.1	Introduction	48
3.2	Studies on existed VoIP multipath transmission solutions	52
3.2.1	Multipath RTP (MP RTP) protocol	52
3.2.1.1	Architecture and stack	53
3.2.1.2	Advantages and disadvantages	53
3.2.2	Multipath TCP (MPTCP) protocol	54
3.2.2.1	Architecture and stack	55
3.2.2.2	Advantages and disadvantages	55
3.2.3	MPT-GRE in UDP solution	56
3.2.3.1	Architecture and stack	57
3.2.3.2	Advantages and disadvantages	57
3.3	Proposed new solution for train control voice service based on Multipath UDP links	59
3.3.1	Architecture for MPUDP solution	59
3.3.2	Stack of MPUDP solution	61
3.3.3	Advantages and disadvantages for MPUDP solution	63
3.3.4	Overview and related works of RaptorQ protocol	63
3.4	MP-UDP software architecture design	66
3.4.1	Flow Acquisition	67
3.4.2	Encode RaptorQ	68

3.4.3	MP Data Splitter	68
3.4.4	Decode RaptorQ	69
3.4.5	Flow delivery	69
3.4.6	Congestion Control	70
3.4.7	Path Scheduling	70
3.4.8	Logging	70
3.5	MP-UDP software behaviors evaluation with RaptorQ scheduler	71
3.5.1	Related works of VoIP quality assessment	71
3.5.2	Testbed hardware and software requirements	74
3.5.3	Path adding and dropping test for MP-UDP software	74
3.5.3.1	obtained results	75
3.5.4	Performance assessment for RaptorQ en/decode function	76
3.5.4.1	En/decode time consume with different symbol and sub- symbol sizes for various file dimensions with 100% over- head	77
3.5.4.2	Overhead requirement for different Packet Loss Rate (PLR<20%) with 100% successful en/decode	80
3.5.5	MP-UDP software performance evaluation with RaptorQ scheduler	87
3.5.5.1	Test design	87
3.5.5.2	Results and analysis based on two cellular networks . .	93
3.5.5.3	Results and analysis based on one cellular and one satel- lite network	104
3.5.6	Performance comparison among basic, balance and RaptorQ sched- ulers	113
3.5.6.1	Test design	113
3.5.6.2	Results analysis	114
3.5.6.3	Advantage and disadvantage summary for different sched- ulers	124
4	On Board media quality assessment method	127
4.1	State of art studies of video quality assessment methods	128

4.2	Proposed Approach	130
4.2.1	Global temporal analysis	132
4.2.2	Local temporal analysis	133
4.2.3	Spatial analysis	133
4.3	Results and Discussion	135
4.4	Conclusion	137
	Conclusion and Future Development	138
	Bibliography	140

List of Figures

1.1	Railway control systems in Europe	3
1.2	ERTMS/ETCS level 0	4
1.3	ERTMS/ETCS level 1	4
1.4	ERTMS/ETCS level 2	5
1.5	ERTMS/ETCS level 3	6
1.6	Structure of the radio communication system [11]	7
1.7	GSM-R system for railways [13]	8
2.1	Comparison of Standard TCP and MPTCP Protocol Stacks: (a)Regular TCP stack, (b) Multi-Path TCP stack	19
2.2	Three-way handshake with options and keys	19
2.3	Network architecture of the integrated railway system, comprised of multiple PLMNs, and satellite network connectivity links (dotted lines).	22
2.4	Public cellular network coverage along the railway track. Uncovered areas are called as radio holes.	23
2.5	Adding subflow strategy for the proposed MPTCP-based network architecture.	29
2.6	Dropping subflow strategy for the proposed MPTCP-based network architecture.	30
2.7	The management of QoS subflow status.	32
2.8	Test scenario with overlapping QoS link and best effort links (cellular network 1, 2 and 3).	35
2.9	Illustration for testbed.	36

2.10	Test scenario for city and harsh environments. In the city environment, three cellular networks exist with delay $d = 150$ ms, and bandwidth $B = 2048$ kb/s. They are illustrated by dashed line, gray and grid ellipses. The city environment is also covered by satellite network with higher delay and lower bandwidth (<i>i.e.</i> , $d^* = 500$ ms, $B^* = 512$ kb/s). In the harsh environment, there is a radio hole, and it is only covered by one cellular network (dashed circle) and satellite network (big solid line circle). The cellular network has delay $d = 150$ ms, and bandwidth $B = 2048$ kb/s; while the satellite network has higher delay and lower bandwidth, $d^* = 500$ ms, $B^* = 512$ kb/s. For easy of reference, the train moves from left to right along the railway track, and the time corresponding to the transition between two adjacent areas is also reported.	37
2.11	Throughput [bit/s] experienced for different connection links <i>i.e.</i> , satellite network (<i>black line</i>), cellular network 1, 2 and 3, corresponding to <i>red</i> , <i>green</i> , and <i>blue lines</i> , respectively. Light purple bars show the total throughput during the entire connection period.	38
2.12	Static experiments. Average receiving bitrate vs. transmission rate for different scenarios. The behavior is independent from the particular scenario.	40
2.13	Static experiments. Average delay vs. transmission rate for different scenarios.	40
2.14	Static experiments. Average delay STD vs. transmission rate for different scenarios.	41
2.15	Static experiments. Average jitter vs. transmission rate for different scenarios.	41
2.16	Dynamic experiments. Average receiving bitrate vs. transmission rate for different tests.	45
2.17	Dynamic experiments. Average delay vs. transmission rate for different tests.	45
2.18	Dynamic experiments. Average delay STD vs. transmission rate for different tests.	46
2.19	Dynamic experiments. Average jitter vs. transmission rate for different tests.	46
3.1	VoIP Standard implementation	50
3.2	VoIP normal implementation	53

3.3	Architecture for MP RTP	54
3.4	Protocol stack for MP RTP solution	54
3.5	Architecture for MP RTP solution	55
3.6	Protocol stack for MP TCP solution	55
3.7	Architecture for MP T-GRE in UDP solution	57
3.8	Protocol stack for MP T-GRE in UDP solution	58
3.9	The architecture of proposed solution.	61
3.10	Data sending and receiving example for MP UDP software.	62
3.11	Architecture for MP UDP solution with of RaptorQ scheduler	62
3.12	Protocol stack for MP UDP solution with of RaptorQ scheduler	62
3.13	MP UDP system components	67
3.14	R value versus Mouth to ear delay	72
3.15	Test scenario	75
3.16	Test results.	76
3.17	Delay budget	78
3.18	Set up for test 1	79
3.19	Set up for test 2	81
3.20	Receiving time vs. packet loss rate	82
3.21	overhead vs. Packet loss rate	82
3.22	Test setup for scheduler RaptorQ	93
3.23	Receiving time for different codecs	95
3.24	Receiving time for different overheads	96
3.25	Average delay for different codecs	97
3.26	Average delay according to different codec types	98
3.27	Average delay for different overheads	99
3.28	Average delay according for different overhead rates	99
3.29	Average jitter according to different codec types	100
3.30	Average jitter for different overheads	101
3.31	Average jitter for different codec types with diverse overhead rates	102
3.32	Packet loss rate for five types of codec transmissions with 0% overhead for three different kinds of channel packet loss rate setting	102

3.33	Average transmission time for different overhead rates and codec types . . .	106
3.34	Average delay for different overhead rates	107
3.35	Average delay all kinds of overhead rate.	108
3.36	Average delay for different overhead rates and packet loss rate	109
3.37	Average delay time with different different codec types	109
3.38	Jitter with different overhead rates	110
3.39	Jitter with different packet loss rate and codec types	110
3.40	Jitter with different codec types	111
3.41	Packet loss rate with different overhead rates	112
3.42	Receiving bandwidth with two cellular networks	119
3.43	Receiving bandwidth with one cellular and one satellite networks	120
3.44	Average jitter for two cellular networks	121
3.45	Average jitter for one cellular and one satellite networks	122
3.46	Average loss rate for two cellular networks	122
3.47	Average loss rate for one cellular and one satellite networks	123
3.48	Average out of order rate for two cellular networks	124
3.49	Average out of order rate for one cellular and one satellite networks	125
4.1	Two main artifacts appear in the frame, they are broken blocks and repeat lines.	131
4.2	E_l and E_r are the vertical edges on the left and on the right boundary of the block, and with A_c , A_l and A_r are the average values inside the current block and of the left and right adjacent blocks.	134

List of Tables

2.1	Synthetic Scenarios for PLMN Outage Performance Evaluation	25
2.2	Parameters used in the simulations.	38
2.3	Static Experiments Setting	39
2.4	Dynamic Experiments Setting	44
3.1	APU model for the MOS of some codecs [99]	73
3.2	Network parameters	75
3.3	Results of test1, with symbol size 512 byte	79
3.4	Results of test1, with symbol size 256 byte	80
3.5	Packet loss rate 1%	83
3.6	Packet loss rate 3%	83
3.7	Packet loss rate 5%	84
3.8	Packet loss rate 10%	84
3.9	Packet loss rate 15%	85
3.10	Packet loss rate 20%	85
3.11	Receiving time vs. packet loss rate	85
3.12	overhead vs. Packet loss rate	86
3.13	The networks delay and jitter parameters, we use C as Cellular network and S as satellite network from now on.	87
3.14	Test design (part 1)	88
3.15	Test design (part 2)	89
3.16	Test design (part 3)	90
3.17	Test design (part 4)	91

3.18	Test design (part 5): From test 1 to test 60 with two cellular networks, from test 60 to test 135 with one cellular network and one satellite network. These tests are focus on the different packet loss rate and codec types.	92
3.19	The packet loss rate for five codec types with different channel loss rates and the overhead rates	103
3.20	The packet loss rates for one cellular link and one satellite link	112
3.21	The transmission packets length, packet rate and bandwidth for each codec	114
3.22	transmission test list (part1).	115
3.23	transmission test list (part2).	116
3.24	transmission test list for two main kinds of scenarios with different codec types, schedulers and channel loss rates (part3).	117
4.1	Performance comparison for CIF and 4CIF video sequences.	136

Chapter 1

Introduction of European Rail Traffic Management System/European Train Control System

Railway transportation not only plays an important role in the construction of all countries, but also plays an important role in people's daily lives. In other words people's daily life is inseparable from the railway transportation. Some people go to work by train, some people go for their journey by train, and some other people use train to transport materials. Train appears in everyone's live and we are very familiar with it. Nevertheless, few people know what is the European Rail Traffic Management System (ERTMS) [3] and which kind of communication networks supports the transmission of controlling information. So, in this thesis, we will give some information about ERTMS and present a few new transmission supporting solutions for both signaling and voice services.

1.1 European Rail Traffic Management System/European Train Control System overview

Europe has the most sophisticated and intensive rail links in the world. Twenty three different signaling systems in 15 countries coexisted in Europe. Signal systems are shown in Figure 1.1. For this reason, international train have to mount with several on board equipment. In fact, systems switching time consuming will decrease railway line transportation capacities [4]. In order to achieve the interoperability, and to allow a free train circulation in different European railway networks [5], the European Commission (EU) and International Union of Railways (IUR) together with seven biggest railway signaling companies (Alstom, Adtranz, Siemens, Alcatel, Ansaldo, Westing House, Invensys) gave birth to a new specification standardization train control signaling system, so called European Rail Traffic Management System–ERTMS.

Today, more than 2700 km of high-speed lines in Europe (including Switzerland) are operated with European Train Control System in commercial service. Moreover, in the long term, some countries such as Luxembourg, Sweden, Switzerland and Denmark are willing to implement ERTMS nationwide, too [6].

1.2 ERTMS/ETCS levels

ERTMS system is mainly separated into two parts: the first part is the European Train Control System (ETCS), which is an Automatic Train Protection (ATP) system. ETCS transfers information to both drivers and controllers in real time, which improves flexibility of train management. The second part is GSM-R (Global System for Mobile Communications - Railways) the GSM mobile communications standard for railway operations. This network provides the railway-specific voice and data services. The assembly are included both on-board and infrastructure equipment [3]. ETCS, which supervises the train movement, is comprised of the signaling, controlling and train protection systems. It can be applied at four different levels, which define whether

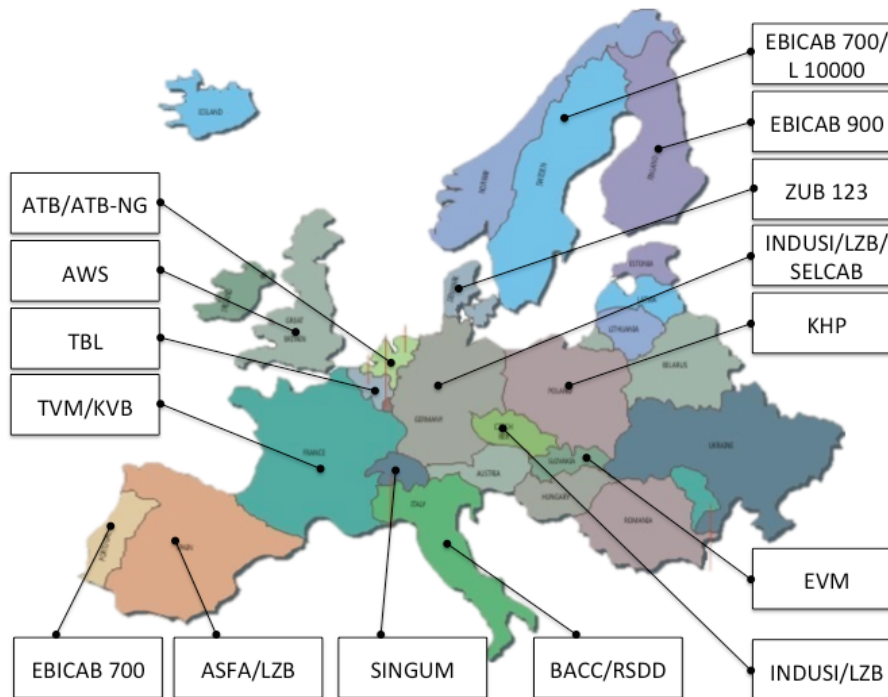


Figure 1.1: Railway control systems in Europe

trackside signals are still used and how the data is transmitted between train and infrastructure[7].

Level 0: ETCS-compliant locomotives or rolling stock interact with lineside equipment that is non-ETCS compliant [1], shown in Figure1.2. It means that, train equipped with ERTMS/ETCS can run on railway lines without ERTMS/ETCS system (national signal system) or lines with ERTMS/ETCS system [8]. The on board equipment monitor the maximum speed depending on the type of train. The train driver should follow the trackside signals.

Level 1: ETCS is installed on lineside (possibly superimposed with legacy systems) and on board; data transmission from track to train via ETCS balises, shown in Figure1.3.

Level 1 is a signaling system that can be superimposed on the existing signaling system, which are the national signaling and track-release system. The train is controlled by existing signal system + balise + track circuit+ euroloop or GSM-R. The on board computer supervises and estimates the maximum speed and the braking curve continuously. Balises (or Eurobalise) are magnetic coupled transponders, which do not

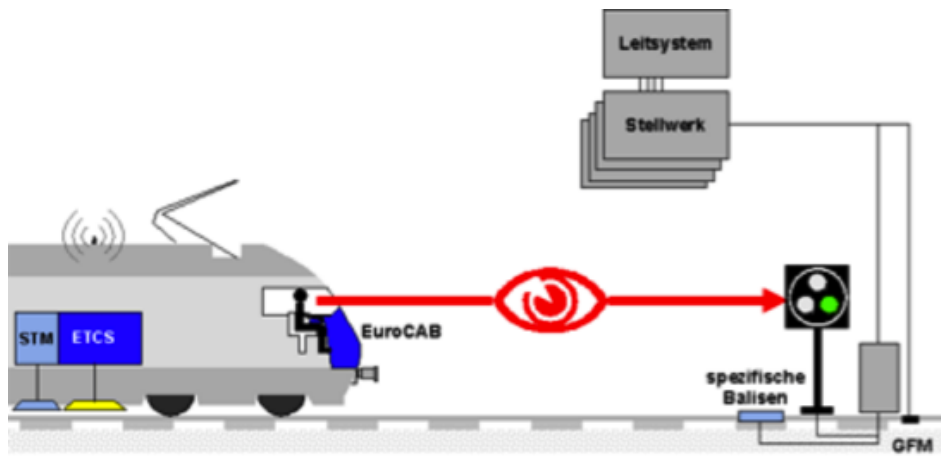


Figure 1.2: ERTMS/ETCS level 0

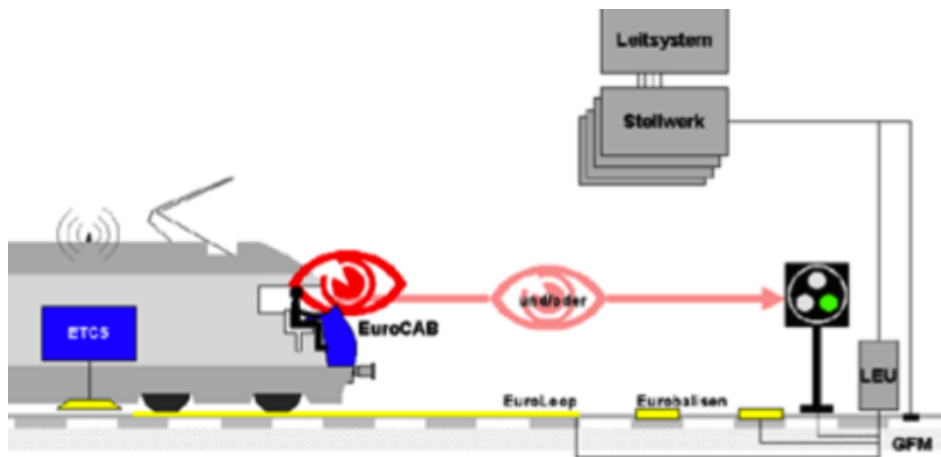


Figure 1.3: ERTMS/ETCS level 1

require a constant energy source. Balises are set with known interval along the tracks, and the information is provided when the train antenna unit passes or stands over the cooperated balise. The on-board antenna unit provides power to the up-link balises by generating a magnetic field. This field is produced in a transmit loop of the antenna unit, and induces a voltage in a reception loop of the balise. The Euroloop is a straightforward extension of the Eurobalise subsystem. Euroloop performs the function of a semi-continuous transmission system allowing transmission of information from a section of the track to the train[9].

Level 2: ETCS data transmission is continuous; the currently used data carrier is GSM-R. Level 2 is a typical safety-critical hybrid system, which guards velocity, acceleration, and distance evolving continuously. The computer can dynamically control

the power, brakes and affect in a continuous-time [10]. This digital radio-based system is combined by track circuit + balise + GSM-R. All the train report their position and direction to the Radio Block Center (RBC) in a certain time interval automatically. Train movement is continually monitored by RBC. The movement authority, speed, location and access route information are transmitted to the train via GSM-R. The Eurobalises are used at this level as passive positioning beacons or "electronic milestones". Between the two positioning signal point, the train detects their position depend on sensors. Balises works as a distance reference to correct error, shown in Figure1.4.

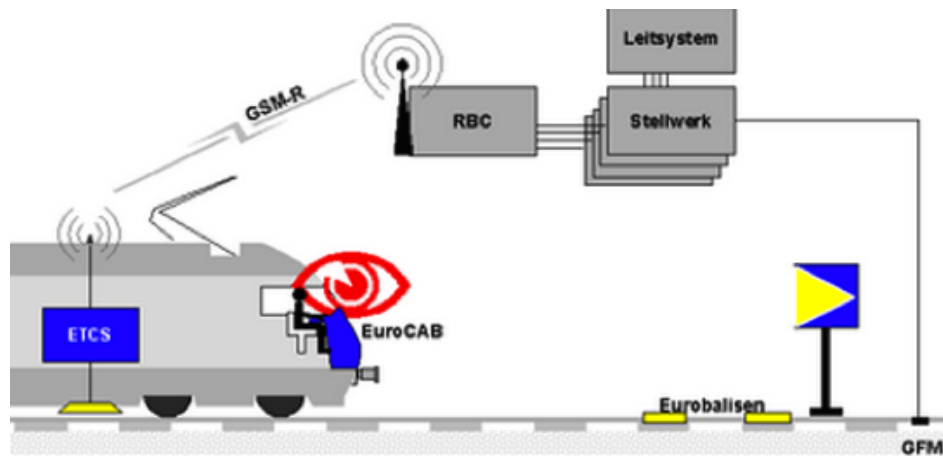


Figure 1.4: ERTMS/ETCS level 2

Level 3: train location and train integrity supervision no longer rely on track-side equipment such as track circuits or axle counters [1], the illustration is shown in Figure1.5. In this level, ETCS behaves beyond simplex train protection functionality, but as the full radio-based train control system. The controlling of the free state of railways is no longer required. Same as in level 2, train updates its position through beacons and sensors, such us axle transducers, accelerometer and radar. Besides, the train must have the capacity of notice the integrity with on board system on maximum of reliability. The movement authority for the following train will be considered up to the position of forward train. Hence, train needs to verify and report its position at certain time interval by communicating to the radio block center the location information. When the release of the traffic provided at real time, and the localization is performed frequently enough, the spacing between trains will approach to the absolute

safety distance (Moving Block). The release of the traffic lane is no longer decided by the fixed cantons. Level 3 is under development. Up to now, solutions for effective monitoring of the train integrity are very expensive and almost excluded from the old rolling stock of freight traffic[4].

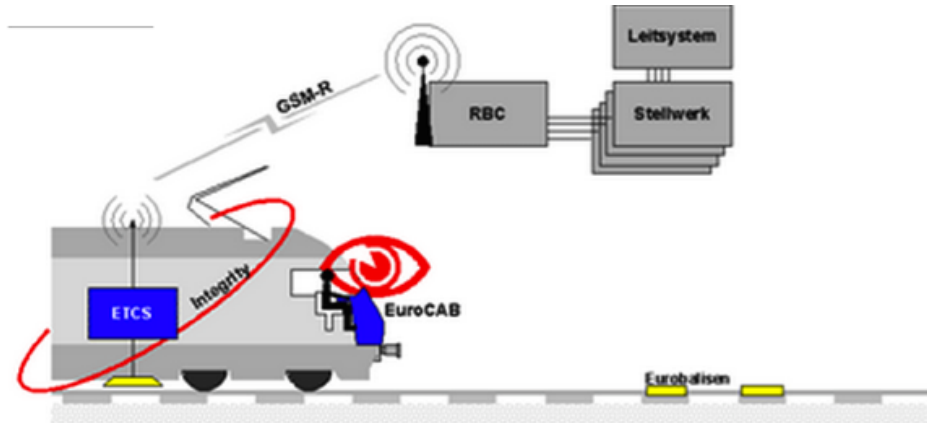


Figure 1.5: ERTMS/ETCS level 3

1.3 EuroRadio protocol for ERTMS/ETCS safety and security

The safety and security for ERTMS/ETCS system is mainly guaranteed by EuroRadio protocol [11]. EuroRadio protocol defines a safe communication system by adding a safety-related transmission layer between the applications and the open network. Open networks only offer data communication services for EuroRadio, but not focus on the safety and security issues. EuroRadio protocol has no limitation on the application functionality and application information flow, the type of networks used, and the physical architecture of the radio communication subsystem. Currently, railway uses GSM-R as the open communication network. Nevertheless, public networks (e.g. PLMN—public land mobile network) can be used as the radio communication part for the ETCS as well.

Safe data transmission is completed by an safety-related transmission system. The reference safety-related system architecture is shown in 1.6. It has three layers: application processing, safety-related transmission layer and the open network layer. The

data safety is mainly guaranteed by the safety-related transmission layer. EuroRadio protocol uses Data Encryption Standard(DES) as the information encryption method. It is a block cipher published in 1977 by the NBS as a US government norm. DES has been renamed Data Encryption Algorithm (DEA) during its adoption as an ANSI standard.

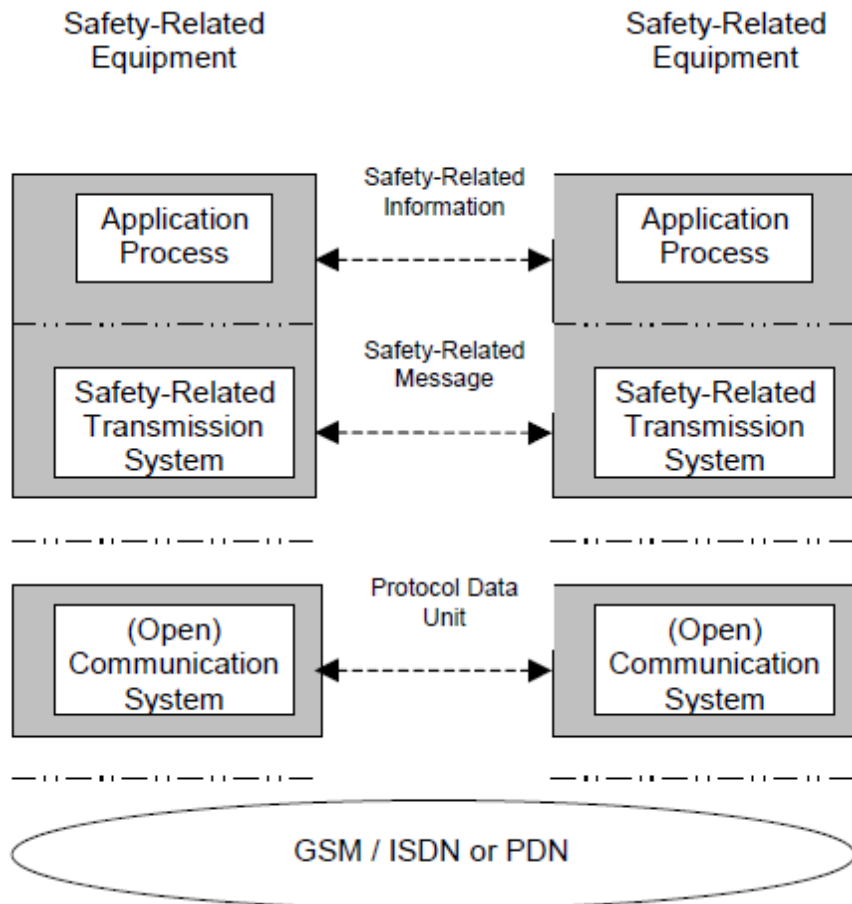


Figure 1.6: Structure of the radio communication system [11]

1.4 Disadvantages of the current GSM-R system and its defects

Global System for Mobile Communications-Railway is based on standard GSM system, which provides voice and data communication between the track and train. GSM-R uses specifically frequencies for railway application, and in different countries the

bands are different [12]. For example, the frequency bands in Europe are 876-880 MHz and 921-925 MHz, but in China and South Africa are 885-889 MHz and 930-934 MHz. In some other places, the 1800 MHz and 1900 MHz frequency bands are used. The GSM-R Mobile Unit offers data and voice communication service through a dedicated GSM-R network, which is distributed along the railway lines. On board equipment is supporting the voice radio service, which provides the communication between shunting or maintenance staff and the train driver. Meanwhile, the maintainers and track workers are using GSM-R hand held terminals for communication. In addition, GSM-R supports bi-directional data exchange service between the on board ERTMS/ETCS assembly and the RBC for the achievement of ATP. In most part of applications, for support both kind of services, the train need two separate on board mobile radio assemblies, shown in Figure1.7.

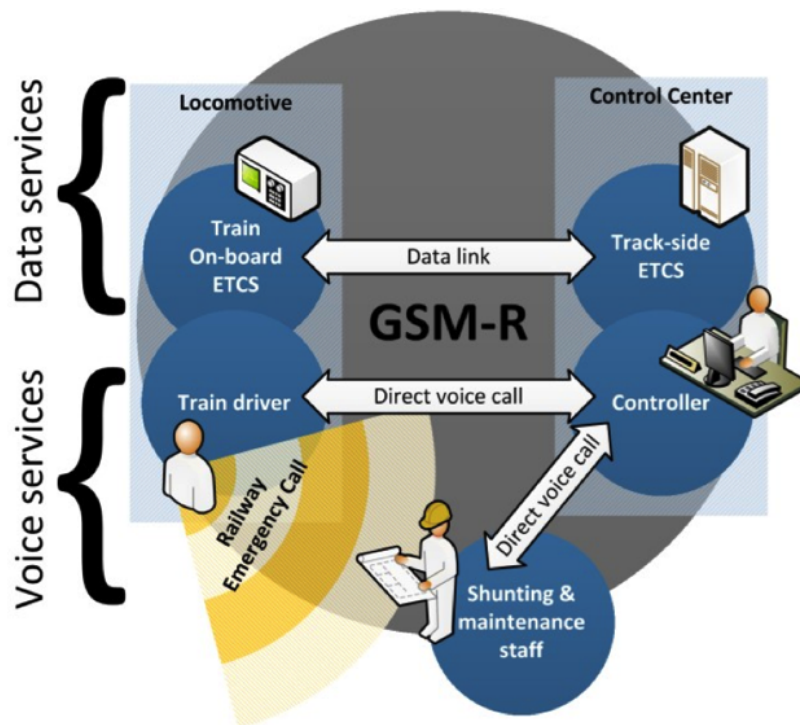


Figure 1.7: GSM-R system for railways [13]

GSM-R is a successful and widely used in railway control system, despite that, after few years of usage revealed some considerable shortcomings in terms of capacity and data transmission capabilities. Because of these shortcomings, GSM-R is becoming the element limiting the number of running trains in areas with high train concentration,

such as major train stations. Moreover, GSM-R cannot support advanced data services [14]. Some major defects are discussed in as follow.

Outdated technology: GSM is designed for commercial mobile network. However, railways used this technology as well. With the maturity of 3G technologies and the continuous development of Long Term Evolution (LTE), GSM is gradually becoming obsolete. Both equipment manufacturers and network operators are stop investing to old GSM technologies. GSM was constrained with hardware. The mobile terminals 20 years ago have much low computation capabilities and energy efficiency. GSM also has worse throughput capability than now. In GSM, signals are modulated with less flexible and efficient technology (Gaussian Minimum-Shift Keying) than nowadays, such as Orthogonal Frequency-Division Multiple Access (OFDMA) in LTE. In addition, in new technologies the modulation scheme can be chosen depending on signal quality. With increasing level of ETCS, the data communication becomes more and more important. GSM, designed for voice communication, is no more meeting the needs of modern railways. GSM-R, liking the initial GSM standard, offers only circuit-switched transmission mode that makes inefficient data communication and consumes much more resources than necessary[13].

Interference in frequency: The frequency interference is one of the most important issues that can not be ignored. As mentioned before, in Europe, GSM-R are operated in a dedicated frequency band 876-880 MHz and 921-925 MHz. Meanwhile, public operators use the neighboring band for commercial mobile network, which will cause interference to GSM-R system. In a real scenario, this problem is even deteriorated, since both railway and commercial operators try to achieve a maximum coverage along the tracks. The reduction of interference requests larger guard band between public and rail frequencies. It means that, commercial operators should abandon the usage of frequencies near the boundary of rail bands, which could interfere with GSM-R network. For example, from the end of 2009, China Mobile Communications Corporation (CMCC) promised to close all its services in GSM-R band [15].

Capacity issues: The limited radio capacity of GSM-R is a big problem for implementation on ETCS Level 2, where each train needs to transmit ETCS data continuously to RBC. On the 4 MHz band, the available GSM-R can place 19 channels of

200 kHz width and 8 time slots with fixed-throughput in each channel. GSM-R is also specified for speeds up to 500 km/h, hence mobile cells along the track should be very small with a maximum radius of 2, 3 km[16]. This reduces the radio capacity as well, because, the same channel cannot be reused by neighboring cells due to interference. Moreover, GSM-R offers only a circuit-switched mode, each of the train-RBC connection occupies one time slot continuously. But, due to the fact that ETCS messages are small and discrete, reserved time slot causes an enormous waste of radio resources. Although General Packet Radio Service (GPRS) can be considered as a solution for the insufficient number of radio channels, in fact, cannot meet the efficiency, speed and economic requirements for data transmission.

Other problems: GSM-R is a dedicated network used by a relatively small industry. This attribute not only builds barriers for sharing the benefit from the commercial mobile operators or some other public services, but also built the barriers for introduction of new services. GSM-R is no longer good enough to satisfy the growing demand in terms of throughput, delay and the new requirement (such as video surveillance or Internet access for passengers). Now that, GSM-R cannot support advanced data services. Hence, modern technologies should take place of GSM-R, and take over the railway communication, such as UMTS (Universal Mobile Telecommunications System), LTE, Wi-Fi, and Satellite communication.

1.5 Equipment manufacturer for ERTMS and ETCS systems

Many companies provide device for the ERTMS–train control and management system. Ansaldo STS, as the global leader of ERTMS, has the entire products chain for ERTMS system. Bombardier provides the wayside components, and Alstom is also a famous company for the ERTMS equipment making. All this companies have many contributions on the ERTMS system development and they have a lot of experiences on ERTMS system implementation in many countries.

There are also many big companies working for the ETCS communication subsys-

tem. Nokia competing with ZTE supplies Flexi Base Stations, fiber optic repeaters and Transcoder/ Sub-Multiplexer(TCSM) for GSM-R radio network. Huawei is well known for the trackside device. It offers some integration GSM-R solution for the railway control. For example, Huawei BTS3900 Base Transceiver Station can support GSM and LTE modes for GSM-R-to-LTE evolution that can well protect customer investment.

Siemens works for the on board equipment of ETCS. The on board equipment can flexibly adapt to the different vehicles. Siemens offers the pre-equipping solution for modern tractive units as well, such as on-board computer with a state-of-the-art platform for PZB 90 automatic train.

Funkwerk developed some handheld device for voice service, *e.g.* focX-Handhelds, which are designed for protection class IP65. For instance, DualfocX is able to listen to GSM-R and 450 MHz channel at the same time. This allows implementing intelligent call pre-emption scenarios covering both radio technologies.

1.6 Costs and reduce for railway operation

Reducing the cost of railway operation is always a big challenge for both the train operators and railway infrastructure operators. The typical major cost for train operation can be categorized into 6 sections: rolling stock CAPEX (CAPital EXpenditure), rolling stock OPEX (OPERation EXpenditure), train staff, operation and customer management, energy, and other overhead. In fact, the rolling stock CAPEX and OPEX take about 40% of the total cost, while the train staff account for about 23% of the total cost [17].

Part of the train CAPEX and OPEX is occupied by running ERTMS/ETCS protocol. ERTMS/ETCS protocol needs to be supported by both the roadside and the on board devices. In ETCS, GSM-R is a dedicate network, which occupies a big proportion in the rolling stock CAPEX and OPEX.

The ETCS is composed by two unions: trackside device and the OBU (On Board Union). OBU can calculate the maximum permitted speed at any given time with the collected information transmitted by the trackside union. Train will slow down automatically, when its reaches the limitation or exceeds the maximum speed. The

OBU has different costs for various models depending on the type of train. In general, the OBU cost around €100 000 for new equipment, €200 000 to €300 000 to adapted with existed on board equipment. On the trackside, the cost depends on the traffic density. But adapting with ETCS often requests complete renovation of the line. For this reason, according to [18], the ETCS with GSM-R costs 350 to 500 k€/km.

In addition, once ETCS has been mounted, both OBU and trackside device need to be checked frequently. If some instrument is broken, it needs to be maintained as well. Both human cost and infrastructure cost should be considered into the train staff and rolling stock CAPEX and OPEX, too.

In order to reduce railway operation cost, we need to focus on main aspects in the total cost, the rolling stock CAPEX and OPEX, the staff cost as well as the passenger organization and management cost.

To our interesting, from the ERTMS point of view, on one side, the advanced protocol can shrink the safety distance between two trains and increases the efficiency of rail usage. This can help the operators to reduce the OPEX. On the contrary, involved in GSM-R adds new CAPEX and OPEX for the operator. Thus, under the condition of guarantee the network safety and reliability, using public communication networks to replace the dedicate GSM-R could be a good solution to reducing the railway cost. This because public networks have already been installed along the track by public mobile network operators, and it also does not request the special train staff to check and maintain. Moreover, the public networks are always deployed with modern technologies. This allows ERTMS to have more services in the future.

Chapter 2

Train control signaling data transmission solution based on Multipath TCP links

2.1 Introduction

European Rail Traffic Management System/European Train Control System is the most advanced standard for managing and controlling railway traffic. The deployment of ERTMS-ETCS is contributing to a global standard in terms of both interoperability among different national systems, and highest safety level achieved. In ERTMS three functional levels of automated control are foreseen. In ERTMS/ETCS, the GSM-R communication subsystem is one of its main components.

Exploiting old GSM technology, GSM-R has now reached its technological limits (*i.e.*, technological obsolescence), with adjacent channels interference, and limitation of transmission capacity. On the other hand, extension to secondary lines and freight trains of safe and efficient traffic management systems, originally designed for high speed trains, requires cost effective solutions for both train localization and communications. As a matter of fact, in Italy and more in general in Europe only the fast train lines have been equipped with GSM-R. On the other side, for secondary railway lines that represent the 50% of the total European railway network [19, 20], the deployment of

GSM-R is considered economically not sustainable by the infrastructure manager *e.g.*, RFI (Rete Ferroviaria Italiana) in Italy. While ERTMS/ETCS requirements for high-speed railway have been clearly defined, delay requirements for secondary lines have not been specified yet. Nevertheless, railway infrastructure manager estimated that a delay less than 2 seconds is acceptable in such a scenario.

Leveraging on the above motivations, GSM-R represents a limited technology that is no longer able to effectively support current and further developments in ERTMS/ETCS. Thus, new communication systems for railway applications should be defined with the aim of both complementing GSM-R in the short/medium term, and then replacing it in the next future [13, 14].

In order to improve GSM-R KPIs (Key Performance Indicators), many researchers are working on the optimization of old GSM technology, such as building up new shadow fading models, and reorganizing the distribution of base stations by exploiting an extensive measurement campaign in HSR (High Speed Railway) cells [21]. Although those methods can improve GSM-R KPIs, there are a lot of limitations. First of all, in order to have an accurate shadow fading model, data acquisition should cover all base stations along a railway, with significant impact on both cost and time. Second, the shadow fading is strictly dependent on the local electromagnetic environment. Third, the redeployment of base stations may need massive cooperation among the rail infrastructure manager, local administrations, and residents, with an increase in cost and time.

Leveraging on previous aspects, many researchers tend to use new technologies to replace outdated GSM-R [22, 23]. The most popular solution is to introduce LTE technology into the train control communication system. Similar as GSM-R system, the alternative solution is also based on a dedicate network system *i.e.*, the so-called LTE-R (LTE for Railway). Experimental campaigns of LTE-R have been carried out in many countries, proving that, from the technical point of view, LTE-R can effectively compensate the shortage of GSM-R, and fulfill the demand of train control systems. In [24] Zayas *et al.* foresee that in 2025 LTE will become the new bearer network for railways in Europe, while Baek *et al.* [25] provide field-test results for LTE-R train control system in Korea. A new methodology has also been presented to account for

the predicted radio coverage, the handover issues, and the number of mobile network operators. In [22, 26, 27] Sniady and Soler use LTE-R as an alternative to GSM-R, and show that LTE-R can fulfill the requirement of seamless ETCS connectivity. Another system exploiting LTE has been proposed by Yan and Fang in [28]. The authors present an LTE based U-plane/C-plane decoupled network architecture for train control system. Lei *et al.* [29] also present an LTE-R communications architecture for train control service. Finally, in [30] a review on routing approaches in the context of vehicles has been presented.

Although LTE-R can fix most of the issues related to GSM-R, there are still some important unsolved problems. For instance, full coverage of both high speed and secondary lines with LTE-R not only requires a substantial investment, but also a long deployment period. In addition, in harsh environments served by rails, such as desert and gobi, deployment and maintenance of land mobile communication infrastructures is quite hard. Those scenarios represent the “totally-disconnected” case, where the only cost effective network coverage is provided by satellite networks ¹. Moreover, both GSM-R and LTE-R are private communication systems generally owned by railway infrastructure managers, that make use of a dedicated spectrum effectively employed only for a limited amount of time, with a waste of radio resources and high OPEX.

Some other researchers propose WiMax (Worldwide Interoperability for Microwave Access) as the alternative technology of GSM-R [31–33], although this technology is not widespread as LTE. In [33] a deep comparison between two candidate architectures respectively based on WiMax and LTE is carried out by Aguado *et al.* The reported results proved that, at least for low speed trains, both technologies can fulfill the ERTMS/ETCS requirements.

Jia-Yi Zhang *et al.* present a multi-system-based access architecture for high-speed railways in [34]. Three different networks are considered in the communication infrastructure *i.e.*, (i) the ground network, (ii) the ground-to-train network, and (iii) the

¹Notice that the totally-disconnected scenario is intended in terms of availability of connectivity links. The proposed technique is designed to provide connectivity only through PLMN or satellite links. Of course, track circuits and balises are still available, and the train collects its location information from the balises and sends to control center by the on-board communication system.

train network. Different radio access technologies (*e.g.*, WLAN, 2G and 3G) are used to support the transmission in accordance with different networks. This architecture can be exploited not only for on-board communications, but also as an auxiliary system for railway signaling to increase railway transportation safety in the context of Positive Train Control (PTC) [35]. Finally, in [36] Liu *et al.* present a topology discovery protocol in wireless linear networks in order to enhance the wireless train backbone communications.

As previously noted, solutions based on single technology and dedicate private network result in the waste of both spectrum resources and infrastructures, and then rise up both CAPEX and OPEX . Hence, in [37] Neri *et al.* initiate a multi-bearer communication architecture, which takes advantage of the integration on PLMNs and satellite networks.

The integration of PLMNs and satellite technology can represent a viable solution to replace existing GSM-R communication subsystem in ERTMS/ETCS. As an instance, in [38] the authors present the architecture of a novel railway safety monitoring system based on the Chinese Area Positioning System (CAPS). CAPS uses communication satellite transponder in-orbit as a relay to provide position, velocity and time information services. The proposed system can provide trains safety and efficient operations.

Regarding the PLMNs, the use of such networks for railway applications has been addressed recently in [37] and [39]. In [37] the smart routers on board of the train and in the train control center can “intelligently” select the available network time by time (PLMNs or the satellite network) to meet the predefined QoS (Quality of Service) level. The main disadvantage of this solution is the handover procedure occurring among PLMNs and/or from the satellite network to a PLMN, and vice versa, that may lead to high latencies and unnecessary data retransmissions. Indeed, when the latency is larger than the maximum value allowed by the ERTMS/ETCS, (typically 6 s), the train will be forced to move to a “safe mode”, or in the extreme case, to stop.

We point out that in this thesis our goal is to provide a feasible answer to the needs of secondary lines with trade-off between costs and performance (in terms of availability), while guaranteeing a safety level (*i.e.*, connectivity and reliability) through the combined use of both PLMNs and satellite links.

Following the concept of heterogeneous networks, in this thesis we extend a previous work [39], where we introduced a new architecture based on the integration of both best effort PLMNs and QoS-guaranteed satellite networks. We exploit Multi-Path Transmission Control Protocol, with the aim of (i) optimizing the joint use of the access networks –in terms of cost and usage of radio resources–, and (ii) full match the availability, reliability and temporal requirements of ERTMS/ETCS. Through the use of MPTCP adding and dropping approach, we can dynamically exploit different technologies and the physical characteristics of each link. As an instance, assuming the availability of GSM, 3G and satellite networks, different subflows can rely on such technologies. In order to enhance and make flexible the existing railway control communication subsystem, our approach can be a viable solution to replace current dedicate GSM-R.

Also, we highlight that our proposed communication subsystem is based on the MPTCP approach, where link reliability can be guaranteed. MPTCP can deliver data via multiple paths simultaneously without modify the application layer, allowing an easy merging among existing and future radio access technologies. In particular, we investigate a logic for dynamically adding and dropping multiple paths *i.e.*, the subflows in the MPTCP lexicon, and handling subflow priorities in order to meet the ERTMS/ETCS QoS requirements. This logic is comprised of the twofold concept of (i) outage map, built on historical network measurements, representing the system long term memory, and (ii) current network status provided by real-time measurements. Finally, a real scenario test demonstrates that the proposed architecture can offer a stable communication even during handover, and fulfill the demands for ERTMS/ETCS system. Experimental results show the performance on different scenarios based on the typical PLMN and satellite networks parameters.

To summarize, the main work of chapter 2 is proposing a novel network architecture of the integrated railway system, comprised of multiple PLMNs, and QoS-guaranteed satellite links, in order to reduce the OPEX and minimize the CAPEX for the communication subsystem. Indeed, actually the communication subsystem for ERTMS is based only on GSM-R, which is a dedicate communication system based on 2G technology, with higher CAPEX and OPEX. In the proposed framework, we present a novel strat-

egy for adding and dropping subflows in the context of MPTCP, aimed at minimizing the expenditure on the satellite traffic still fulfilling the safety requirement.

2.2 Overview of multipath TCP protocol

With the rapid development of wireless communication technologies, networks become heterogeneous with various access technologies, such as xDSL, Ethernet, Wi-Fi, Cellular/Cellular LTE, Satellite, and so on. Now days, most of the equipment have more than one interface (by different wired/wireless access technologies) to connect with the network. Comparing with single-path data transport, multipath transmission have multiple available links simultaneously, and have the ability to shift traffic from congested paths to not congested ones. Many solutions have been proposed for multipath transmission from a transport layer perspective. In order to maximize the usage of network resource and achieve reliable transmission of multi-path, IETF (The Internet Engineering Task Force) working group extended the traditional TCP functions, and the first idea of Multipath TCP (MPTCP) was proposed on one IETF draft in 1995, authored by Christian Huitema[40].

MPTCP is a set of extensions to regular TCP (Transmission Control Protocol) to provide a Multipath TCP service, which enables a transport connection to operate across multiple paths simultaneously [41]. It supports more than one pair of IP addresses on both Multihome hosts. MPTCP has a good ability to increase redundancy and end-to-end throughput.

Between the application layer and the transport layer, IETF group added a MPTCP support layer. The regular TCP flow only works as the subflows. Hence, from the application layer on both sides of the communication, the transport layer is still one-way connection, shown in Figure2.1.

Like the regular TCP, to start a connection of MPTCP also need experience of all three-way handshake. But the difference is that, during the MPTCP initial handshake process, SYN, SYN\ACK and ACK carry MultiPath CAPable (MP_CAP) information. For compatibility with regular TCP, this information is transferred via TCP options field, numbered from IANA (Internet Assigned Number Authority), to detect

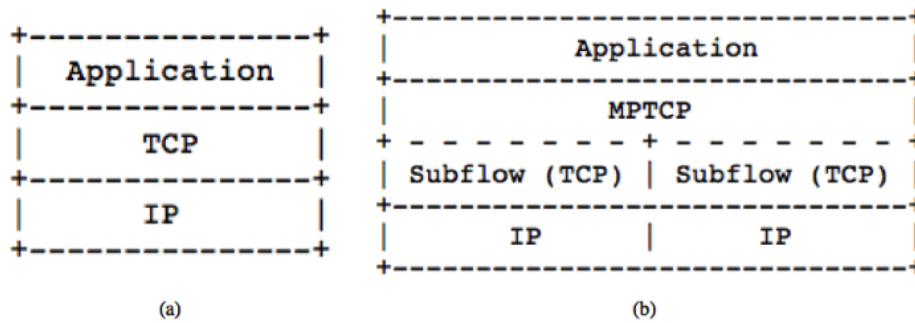


Figure 2.1: Comparison of Standard TCP and MPTCP Protocol Stacks: (a)Regular TCP stack, (b) Multi-Path TCP stack

whether both host and the middlebox are MPTCP-aware devices, shown in Figure 2.2. It also contains 64 bits key for each host, which will be used to authenticate the additional subflows during current connection.

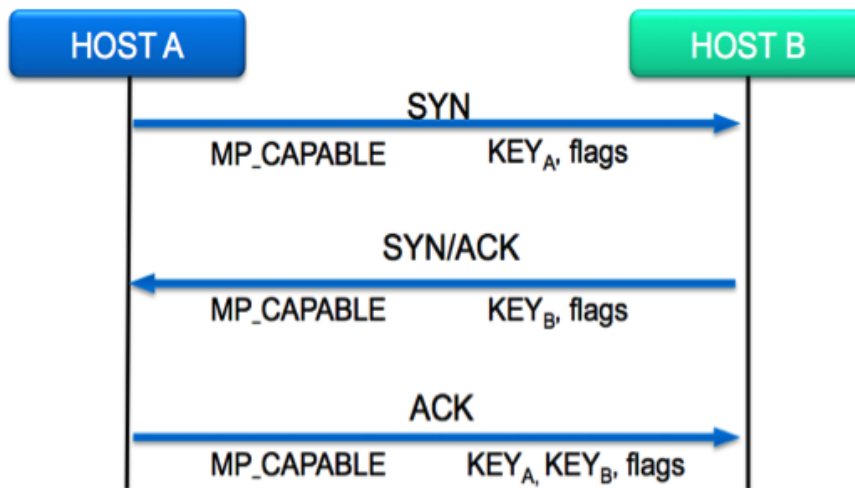


Figure 2.2: Three-way handshake with options and keys

Either host in a connection can begin a new subflow. The swap of token and HMAC (key-hash message authentication code) can prevent the injection attacks. The new subflow is established via SYN packets with MP_JOIN option. Hosts can also turn off the subflow at any time.

When there are multiple paths in one connection, the senders can decide which subflow being used to send data, and the priority of subflow can be changed as well. Every subflow has two kinds of sequence number, data sequence number and subflow sequence number. Data sequence Mapping is expressed in terms of starting sequence

number for the subflow and the data level[41].

Many researchers did their studies and experiments based on behaviors, performance, packet scheduling and testbeds of MPTCP. Kostopoulos *et al.* in [42] discuss the roles of adopters and stakeholders in the deployment and adoption of MPTCP. During the deployment process, the role of end users is not important, because they are not needed to make a conscious adoption decision. The deployment of MPTCP is mainly in the hands of software vendors, which need to make the deployment decision of enabling MPTCP by default.

In [43] and [44] Singh *et al.* compare the performance and fairness of different existing congestion control algorithms, and propose an algorithm called “delay-delay”. This technique is based on delay congestion, and it can accurately detect a bottleneck and improve throughput of MPTCP. A fluid-based MPTCP algorithm is proposed by Zhao *et al.* in [45]. The proposed approach also focuses on delay congestion event. The delay-based condition is used to identify packet loss due to the wireless link errors, and this model can improve TCP-friendliness and window fluctuation performance in different scenarios. Hwang *et al.* [46] present the fluid model for MPTCP as well, that covers a broad class of MPTCP algorithms, and identify the exact property. They illustrate the impacts of protocol parameters on TCP friendliness, responsiveness, and window oscillation, and demonstrate the inevitable tradeoff among these factors. Furthermore, they propose Balia (Balanced linked adaptation) MPTCP algorithm, which can strike a good balance among the protocol properties. Finally, Ni *et al.* [47] present a fine-grained forward prediction based dynamic packet scheduling mechanism for MPTCP by considering the congestion window, the Round Trip Time (RTT), and packet loss rate. The simulation results show that the mechanism has a well performance on both global throughput, and on the number of out-of-order packets in lossy networks.

Advantage: MPTCP increases redundancy and end-to-end throughput, and achieves a better usage of network resources. Each subflow maintains its own sequence number, thus the sub-data streams use multiple simultaneous transmissions, and the receiver can operate for different sub-stream from the received data sequence of the original assembly[48]. Moreover, each subflow has its own congested window, that allows the data movement from the congestion path to non-congested path.

MPTCP provides a higher reliability during the connection. The MPTCP information is carried by TCP option field, and fall back mechanism can support the regular connection. The exchanging of key and token ensures the safety of the additional subflow. Random initial of sequence number can successfully reduce hacker attacks as well.

MPTCP accomplishes a superior flexibility. This protocol can be applied on any multihome assembly. Furthermore, on the applications side, the connection can be considered as one way connection, since the MPTCP in the transmission layer seems a black box. IP addresses, subflow can be associated or removed at any time during the connection, and the priority for the subflow can be modified, too.

MPTCP is a set of extensions to regular TCP protocol, and it inherits all the advantages from regular TCP, such as the stable and exquisite structure, low data overhead and complete error processing mechanism. MPTCP is a platform-independent protocol, which can be used on different platforms, such as Windows, Mac OS and Linux, or even the combination of the above platforms. It can truly realize the heterogeneous network connection. MPTCP has an excellent recovery mechanism. When some part of the network is intruded, attacked or destroyed, the rest part can still fully available and operational. MPTCP allows to add new services to the network without disruption of the existing ones. MPTCP has the efficiency congestion control mechanism. Besides slow start, congestion avoidance, fast retransmit and fast recovery, MPTCP has different congestion windows on the different paths, which is management to send more packets on less congestion path.

Limitations: MPTCP link has more complex operation than standard TCP. MPTCP establishment requires the server and the client report all the available address information for the further usage. The other significant problem is packet Out-of-order, which result in a throughput degradation [49, 50]. On assembly side, MPTCP put forward higher requirements on the capability of the equipment, such as more powerful processor and larger buffer. Meanwhile, middlebox is also great challenge, such as firewall, NAT, box and router etc.

2.3 New architecture of train control communication system

Although the railway system is classified as safety-related service, not as safety-critical service, availability is still of primary concern. We remark that in case of an outage event lasting for the maximum acceptable latency, the train will automatically enter the safe mode, and will stop.

With respect to the communications, since best effort PLMNs alone cannot guarantee the acceptable availability, we complement them with satellite links providing the required QoS. Also, in order to diminish both CAPEX and OPEX, fulfill the requirements of current train control standards, meanwhile optimize the usage of radio resources to ensure the high availability of the communication subsystem, in our proposed architecture we adopt MPTCP in order to handle the parallel usage of both QoS-guaranteed and best effort paths.

As illustrated in Figure 2.3, the best effort links are supported by PLMNs of different telco operators, while the QoS-guaranteed link is provided by a satellite network. Thus, whenever the latency approaches critical values, the related traffic is routed through the satellite. We obtain this behavior by acting on the adding and dropping logic of MPTCP.

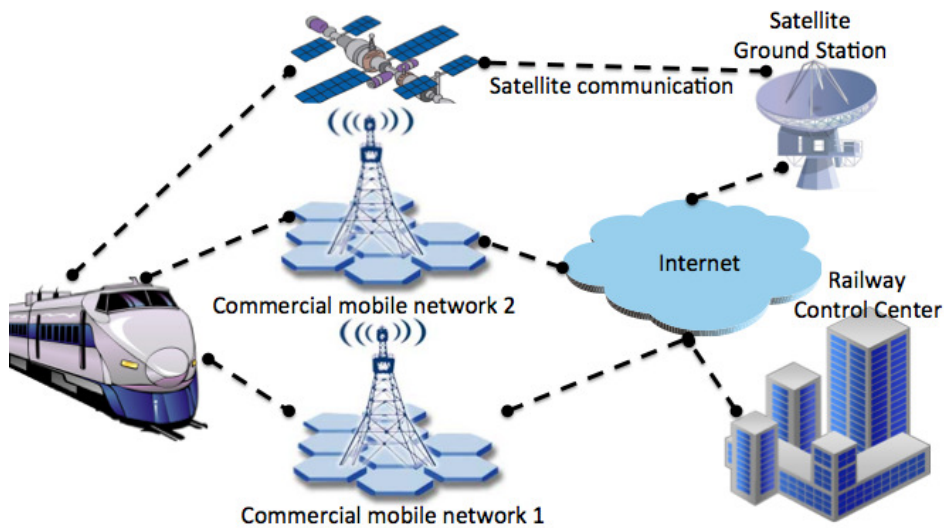


Figure 2.3: Network architecture of the integrated railway system, comprised of multiple PLMNs, and satellite network connectivity links (dotted lines).

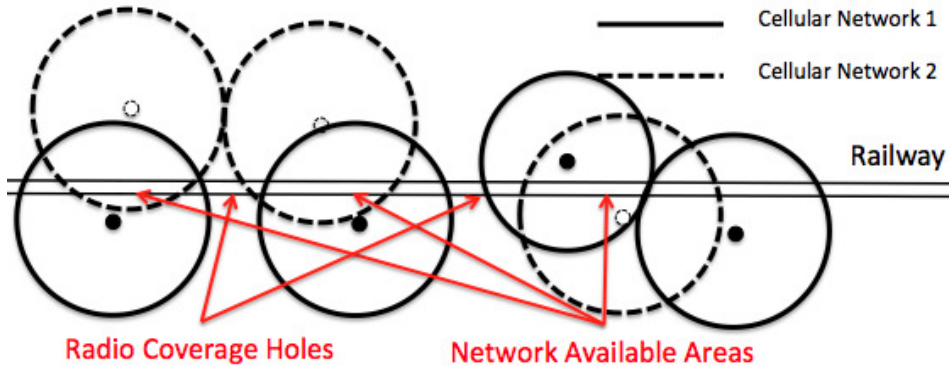


Figure 2.4: Public cellular network coverage along the railway track. Uncovered areas are called as radio holes.

In general, to minimize OPEX, PLMN links should have higher priority than the satellite link whose usage should be limited to those situations in which PLMNs cannot fulfill the QoS requirements. However, as enlightened by the experiments performed in the ESA ARTES 20 3InSat project, hard vertical handover may lead to latencies that far exceed the maximum tolerable values.

Resorting to MPTCP, hard vertical handover can be converted into smooth handover. According to our proposed architecture, the connectivity switching follows a “soft” handover paradigm, where train’s location is exploited to execute proactive handovers. Data link over satellite will be established before no active PLMNs. As soon as the satellite link becomes stable, a part of the throughput will move to the satellite link, while the other part will still be delivered over PLMN links. For this reason, no latency of handover between satellite and PLMN, and among PLMNs, occurs. The data pass from one link to another link smoothly, and seamless.

In order to assess the performance of PLMN service, we adopt the methodology presented in [37], where the overlap percentage is defined as the ratio between the sum of time intervals for crossing the overlapped areas, and the travel total time. From this computation, we will derive the usage percentage of satellite connectivity. For the analysis of PLMN outage performance. We are assuming that the *train-control center* communication interrupts when the train across the radio coverage holes. Let l_i be the

time interval for the train to cross the i -th area, served by two overlapping PLMNs, namely PLMN 1 and PLMN 2, and let T_{tot} [s] be the total time of the travel. Then, the overlap percentage L is defined as the ratio between the sum of time intervals for crossing the overlapped areas, and the travel total time *i.e.*,

$$L = \frac{\sum_{i=1}^{N_l} l_i}{T_{tot}}, \quad (2.1)$$

where N_l is the number of overlapped areas. Moreover, let Q_i be the time interval required for the train to cross the i -th radio hole, and T_{tot} be the overall time required for a ride, then the outage percentage P_{out} can be expressed as:

$$P_{out} = \frac{\sum_{i=1}^{N_Q} Q_i}{T_{tot}}, \quad (2.2)$$

where N_Q is the number of radio holes along the railway. Notice that the outage ratio represents the fraction of time in which traffic essentially flows through the satellite network. In practice, for a more accurate evaluation of the use of the satellite channel, we need to consider also data traffic generated when the subflow priority logic adds satellite flows in areas where service can be granted by some PLMNs.

For a more accurate evaluation of the use of the satellite network, we resorted to a Monte Carlo simulation. The results reported here refer to an Italian scenario. According to [51], in Italy the availability of each PLMN, with respect to land coverage, including 2G and 3G, is not worst than 87%, while the availability of at least one PLMN is 91%. In our simulations, the conservative hypothesis of a single PLMN availability equal to 85%, and an overall PLMN availability of 90% have been adopted. A set of synthetic scenarios compliant with these hypothesis is reported in Table 2.1.

2.4 Paths adding and dropping strategies based on new architecture

The policy about “add and drop” subflows in MPTCP has been investigated in several works, but still represents an issue since there are no efficient mechanisms to identify the available paths between source and destination hosts. Then, the optimal

Table 2.1: Synthetic Scenarios for PLMN Outage Performance Evaluation

Group	Availability of Cellular Network 1	Availability of Cellular Network 2	Overlap Perc. (%)	Average Outage Perc. (%)	Outage Variance (%)
1	85%	85%	80	26.00	3.81
			75	22.12	3.57
			70	18.25	3.33
			65	14.37	3.09
			60	10.50	2.86
2	85%	90%	80	23.99	2.65
			75	20.00	2.48
			70	15.99	2.32
			65	12.00	2.16
			60	8.00	1.99
3	90%	90%	80	24.00	2.66
			75	19.75	2.48
			70	15.50	2.32
			65	11.25	2.16
			60	7.00	1.99

path selection is still a challenge. Barzani *et al.* [52] investigate MPTCP performance w.r.t three aspects *i.e.*, (*i*) path characteristics, (*ii*) configuration parameters, and (*iii*) scheduling policies. Moreover, they study the impact of the choice of initial path in establishing a MPTCP connection [53].

Another packet scheduling has been introduced by Hwang in [54]. The authors demonstrate that for low rate data transmissions, delay is much more important than network bandwidth, and the existing default MPTCP packet scheduler may choose a slow path if the congestion window of the faster path is not available, thus leading to a longer flow completion time. Hence, in [54], the slow path is temporarily frozen, so that data can be delivered quickly over faster paths. Krupakaran *et al.* [55] determine the exact set of multiple paths existing between the source and destination hosts, and make decision based on the number of subflows. They also propose a modified version of Traceflow protocol that helps to determine the path taken by a flow through a network. Authors of [56, 57] present a cross-layer signaling technique to get the path diversity map offered by a Locator/Identifier Separation Protocol (LISP) network. This approach helps MPTCP to create an adequate number of subflows to improve the

transfer time. Also, Baidya *et al.* [58] use a slow path adaptation method to improve the performance of MPTCP. In their method, each subflow will be marked as “good” or “bad”, corresponding to its quality level. Once a subflow is marked “bad”, it will not be involved in the data transmission, until its quality will become good enough.

Finally, the priority of adding and dropping subflows has been also discussed in [41], where a subflow can be set as “backup subflow”, which will not take charge of the transmission, unless all other subflows are congested or unconnected. However, this strategy has some disadvantages, such as the participating nodes do not know in advance whether associated forwarding path can actually support MPTCP subflows. Thus, as a solution, the authors in [59] add a probe flag in MPTCP protocol to indicate to the remote peer that the probing is associated with this subflow.

Leveraging on main research works in the field of MPTCP scheduler, we notice that there are still some issues needed to be discussed, especially when adopting MPTCP in the context of railway applications. First of all, all the previous works consider MPTCP in data center scenarios, which are different from railway scenarios. With the term “data center scenarios” we refer to applications like e-mail, social networking, Internet browsing, which are not critical with respect to latency and delay. The main purpose for MPTCP in the data center scenarios is to improve the throughput. In railway scenario the main MPTCP scheduler goal is the integration of different communication networks in order to reach high level of availability, reliability, and low latency, to support train control and traffic management, while reducing the CAPEX and OPEX. Second, many works are based on the stable network environments, but in the real situations, the network environment changes at every moment (*e.g.*, due to dynamic network conditions).

To overcome the mentioned problems, we propose our new subflows adding and dropping strategies of MPTCP, specially based on railway using scenario. Our dynamical subflows adding and dropping strategies is aimed to guarantee the reliability and availability of the ETCS communication subsystem. It is on the basis of the current status of each PLMN and the predicted performance in the next ΔT seconds, in order to prevent outages of the end-to-end TCP connection. We present a simple approach exploiting MPTCP, in order to not affect the computational cost in the control system.

The proposed strategies are a trade-off between the efficiency and complexity of the system (hardware and software), with the focus on the cost reduction. A more complex strategy needs more parameters, which require more probes installed on the train with consequently higher cost.

The concept of building and updating an outage probability map has been already exploited by the authors in [60]. Now, in this section, we refer to this concept as evolved in the context of the rail scenario, based on the fact that trains usually travel on the same tracks at the same time of the day. At this aim, we observe that, given the train itinerary $\mathbf{X}[m]$, mapping the train mileage m into the track geographic coordinates \mathbf{X} , and the train mileage $m(t)$, velocity $v(t)$, and acceleration $a(t)$, at time t provided by the train Location Determination System, the predicted train location $\hat{\mathbf{X}}(t + \Delta t)$ at time $t + \Delta t$ can be computed as

$$\hat{\mathbf{X}}(t + \Delta t) = \mathbf{X}[\hat{m}(t + \Delta t)], \quad (2.3)$$

where $\hat{m}(t + \Delta t)$ is the time predicted train mileage at time $t + \Delta t$, based on the current train status *i.e.*,

$$\hat{m}(t + \Delta t) = m(t) + v(t)\Delta t + \frac{1}{2}a(t)(\Delta t)^2. \quad (2.4)$$

Then, the predicted outage probability $\hat{P}_{out}^{(n)}(t)$ for the n -th network at time t , on the time interval of ΔT seconds, can be computed as

$$\hat{P}_{out}^{(n)}(t) = \underset{\Delta t \in [0, \Delta T]}{Max} P_{outMap}^{(n)}[\hat{\mathbf{X}}(t + \Delta t); (t + \Delta t)_{\text{mod } 24h}] \quad (2.5)$$

where $P_{outMap}^{(n)}[\mathbf{X}, t_{day}]$ is the outage probability of the n -th network, at location \mathbf{X} , at the time of the day $t_{day} = t_{\text{mod } 24h}$ estimated on the basis of the recorded temporal time series of radio coverage, link status, and effective QoS provided by the monitoring system.

The estimation of the outage probability follows well-known approaches used in traditional cellular networks. Indeed, the issue of expressing the outage probability in cellular networks in terms of the features of the physical layer such as received power level, signal-to-noise and interference ratio, etc., has been extensively addressed in the literature.

Since a direct estimate of the outage probability associated to each network would require an extensive use of each link, in order to have the required accuracy, here we resort to the prediction of the outage probability on the basis of some parameters characterizing the link status measured either by means of traffic data or probe signals. Moreover, while the outage probability associated to the current location of the receiver is derived from these measures, the outage probability associated to the locations that the receiver will occupy in the short future are predicted on the basis of the outage probability map, compiled with the aid of historical data. Obviously, the outage probability map accounts for seasonal trends derived from the observed time series. In practice a cyclostationary model of the $P_{out}(t)$ could be adopted.

As illustrated in Figure 2.5 where the block diagram of the “adding subflow” logic is reported, a MPTCP connection is setup after the completion of the MPTCP handshake and authentication over the subflow with QoS guaranteed bearer. Then data start to be routed over it.

At the same time an empty list of active PLMNs is created, while the physical layer starts to continuously scan the frequency bands associated to known PLMNs, in order to determine if any PLMN is in the receiver range. When the n -th PLMN is detected, the related predicted outage probability $\hat{P}_{out}^{(n)}$, and the current Quality of Service $QoS^{(n)}$ provided by the real time network status monitoring, are tested. If $\hat{P}_{out}^{(n)}$ is lower than a predefined threshold T_{add} and $QoS^{(n)}$ is greater than the minimum acceptable Quality of Service QoS_{add} , the system will add the n -th network to the active PLMN set list, will enable the corresponding logical interface, and add a new subflow over the new PLMN. The distribution of packets among subflows is based on the congestion status of each subflow and the network scheduling.

In addition to the discovery of new PLMNs, the system continuously monitors the existing subflows in order to determine if any of them has to be dropped. As illustrated in Figure 2.6, for each PLMN in the active set the related predicted outage probability $\hat{P}_{out}^{(n)}$ and the current Quality of Service $QoS^{(n)}$ are tested.

If either $\hat{P}_{out}^{(n)} > T_{drop}$ or $QoS^{(n)} < QoS_{drop}$, where T_{drop} is the outage probability drop threshold, and QoS_{drop} is the minimum acceptable drop QoS level, and other active PLMNs exist, the system will directly close the PLMN interface, drop the subflow, and

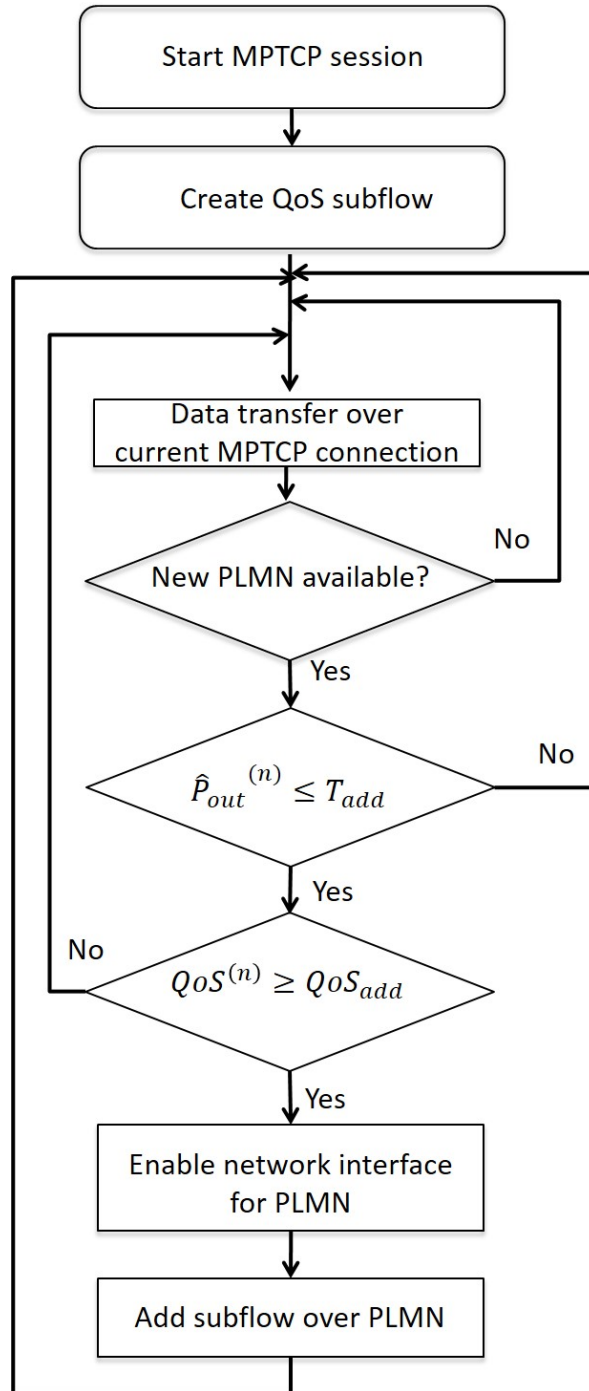


Figure 2.5: Adding subflow strategy for the proposed MPTCP-based network architecture.

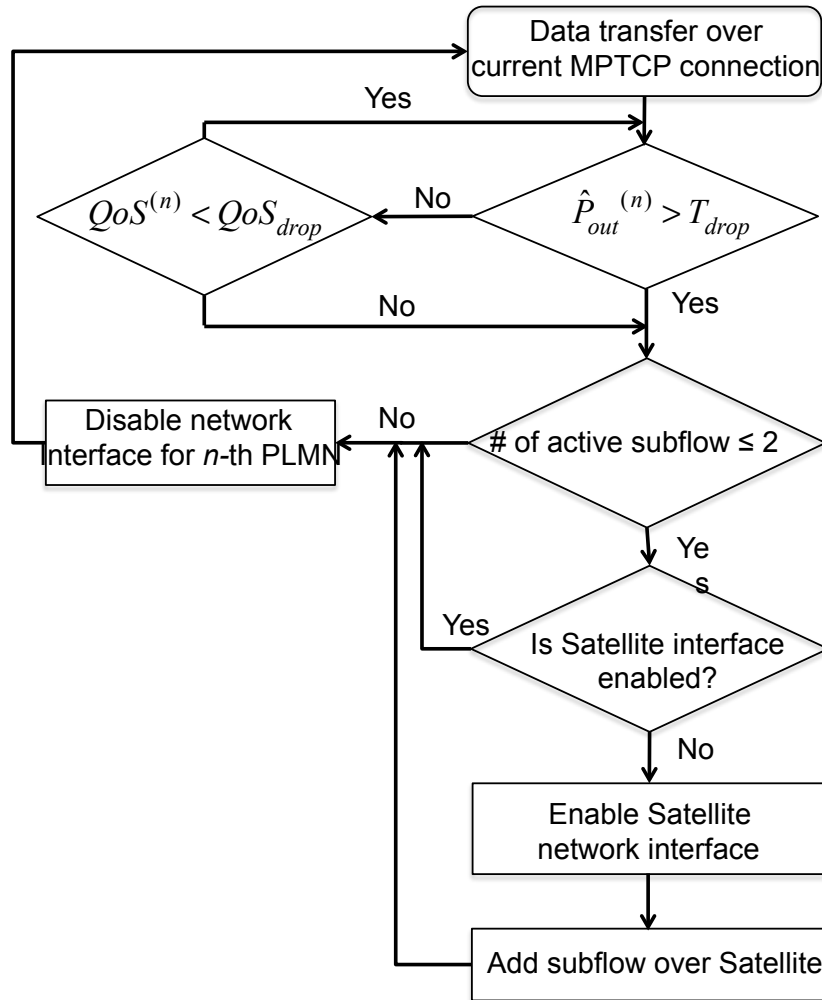


Figure 2.6: Dropping subflow strategy for the proposed MPTCP-based network architecture.

the data transmission will automatically move to other active paths. The subflow selection is taken on the basis of both outage and cost metrics, and so the use of –less expensive– PLMN subflows is privileged with respect to the use of –more expensive– satellite subflows.

Notice that due to its global coverage, subflows with satellite networks are mainly used in harsh environments. Moreover, in such scenario, there are areas along the railway tracks that are not served by PLMNs or where the QoS level for successful train control is not reached for the majority of the time. These areas are called as “PLMN radio holes”, as depicted in Figure 2.4, meaning that the communication links are provided only by satellite networks [37].

In order to illustrate the logic behind subflows management in train control applications, let us consider a typical train journey. Typically, the train control communication begins at the railway station, and the MPTCP session should start with the satellite connection or, if available in station, a dedicated WiFi access link. After the completion of handshake and authentication, data start to deliver over the QoS subflow. Meanwhile, the empty list of active PLMNs (such as Telecom Italia, Wind, Vodafone, H3G, etc. in Italy) has been created. Then, the system begins to add new subflows according to the “adding subflow” logic, and data transfer over all active subflows.

When there are more than two active PLMNs links in current MPTCP session, the QoS link will be switched to a backup status. It means that the QoS link does not transfer data but holds on as a backup line. Notice that PLMNs are best-effort networks, not ensuring connectivity links anytime and anywhere. Thus, in order to reach higher connectivity level, it is expected to have several available PLMNs (*i.e.*, > 2). In general, the number of active PLMN subflows depends on the required throughput and link capacities. Nevertheless, for train control, the throughput is a few kByte/s. Therefore, the chosen threshold on the number of active PLMN subflows can be considered enough in main applications.

With the movement of the train, if QoS level or predicted availability of n -th PLMN changes, the corresponding subflow can be dropped according to the “dropping subflow” logic. When active PLMNs links are less than or equal to two, the QoS (satellite) link will wake up to normal status. The management logic of subflows is described in Figure 2.7.

In railway scenario, the transmission of train controlling signal cannot be isolated from network changes. Thus, real environment experiments should be executed. To fulfill the above-mentioned issues, we carry out the experiments according to the following two aims:

1. To assess the stability, continuity and reliability of the proposed network architecture, a test scenario has been implemented based on both city and harsh environment characteristics;
2. To analyze the performance on typical railway scenarios, the experiments on

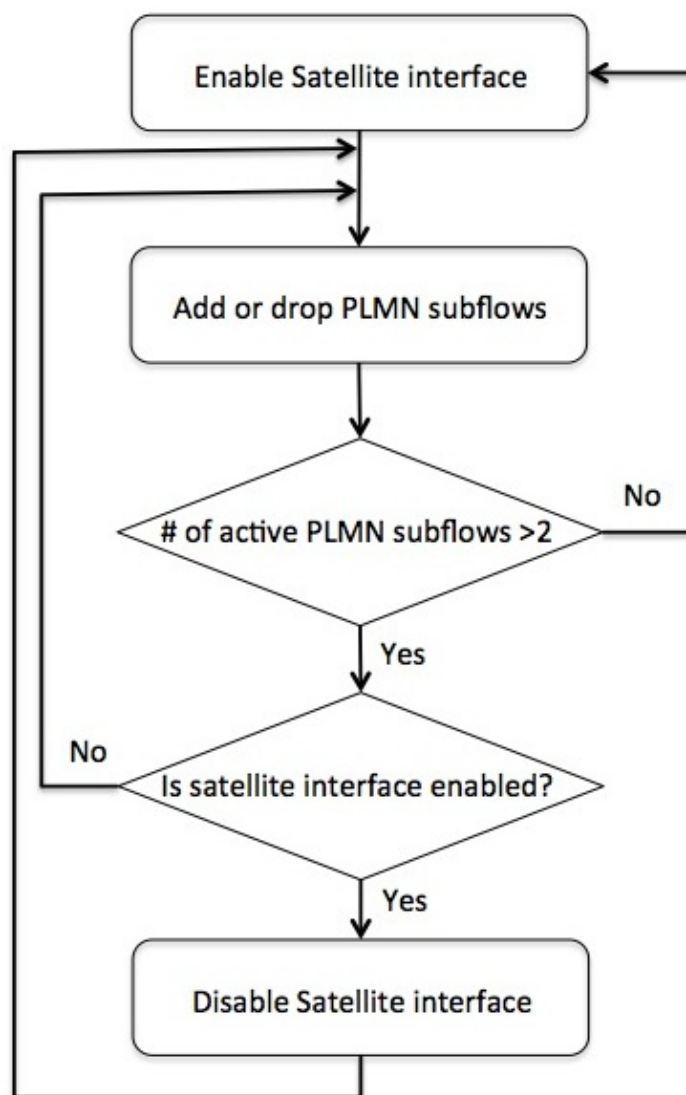


Figure 2.7: The management of QoS subflow status.

“static network”² and a “dynamic network”³ setting have been carried out.

²With the term “static network” we mean a network scenario where the network characteristics are assumed that do not change.

³With the term “dynamic network” we mean a network scenario where the network characteristics are assumed that change with time.

2.5 Paths adding & dropping strategies evaluation on proposed architecture

We describe the experimental setup for the assessment of the proposed network architecture. A brief description of the scenarios considered in our testbed –namely, city and harsh scenarios– is reported hereafter.

To better understand how the adding and dropping strategies work and to evaluate whether the architecture can well support the communication between train and control center, we give some examples for adding and dropping strategies based on the city and harsh environment scenario, and execute an experiment for the assessment of proposed network architecture. A brief description of the test design in our testbed for evaluating the link adding and dropping strategy is reported hereafter.

We need to remark that a comparison with related architectures is a difficult task. For the best of our knowledge, till now there are no related architectures for ERTMS/ETCS exploiting the public communication systems.

2.5.1 Evaluation scenarios design

In the evaluation procedure, we assumed that the train departs from the city and then it moves to the harsh environment, as depicted in Figure 2.8. City area is usually well covered by overlapping PLMNs, thus in this area, PLMNs can achieve a high QoS level. For this reason, the train control communication is expected to be mainly provided by PLMN links.

In low density areas, (*e.g.* desert, wilderness, and primary forest) cellular network is not always available and radio holes also exist at those places. In addition, trains travel at higher speeds and this increases network disconnections. On the contrary, in a typical harsh environment, where the number of satellite masking artifacts (*i.e.*, buildings, bridges, tunnels, etc.) is quite reduced, the availability of satellite connectivity is guaranteed due to the global coverage. Thus, the communications in harsh environment are generally guaranteed by satellite links.

In the case of mountains, satellite connectivity is not always available due to mul-

tipath effect, and the links cannot work very well (*i.e.*, the connectivity is not stable). In case of tunnels, it is even harder and impossible to connect via satellite networks. However, in case of no connectivity, we can exploit dedicated (5G) access networks possibly with satellite backhauling.

2.5.2 Adding and dropping strategies description

Figure 2.8 depicts the test scenario including city and harsh environments. In city scenario, communication is provided by both the satellite network and three cellular networks (*i.e.*, cellular network 1, 2, and 3). At startup, the train and control center communication is established with the QoS guaranteed network (satellite link); if the outage probability and the QoS level for cellular networks overcome the corresponding thresholds, the train will start to execute the communication link adding strategy. In this case, while train moving forward, the system checks for other available PLMNs and adds them to the candidates network list. In the considered scenario, the system adds another subflow supporting by cellular network 1 to the active session, other than the satellite one, and data packets start to be delivered over multiple subflows. Later on, the third subflow is added over cellular network 2 with the same adding strategy as before. Then the fourth subflow is added via cellular network 3.

According to the paths management policy, once there are more than two active PLMNs in a session, the satellite interface will shut down, and all data pass through the best effort links. At this time, if the transmission capacity is good enough, the train can provide Internet access through the on-board WiFi service to the passengers with the exceeding bandwidth.

The train keeps on going forward and then comes into the harsh environment, where the cellular network 3 falls below a corresponding threshold. At this moment, the system initiates its link dropping strategy. Since the number of best effort links is greater than two, the fading subflow is closed directly. Then, there are two PLMNs left in this session, thus the satellite interface is turned on ⁴. Now, the data packets are transported over both best effort and QoS network.

⁴Satellite interface should be switched on, if there are less than or equal to two best effort links

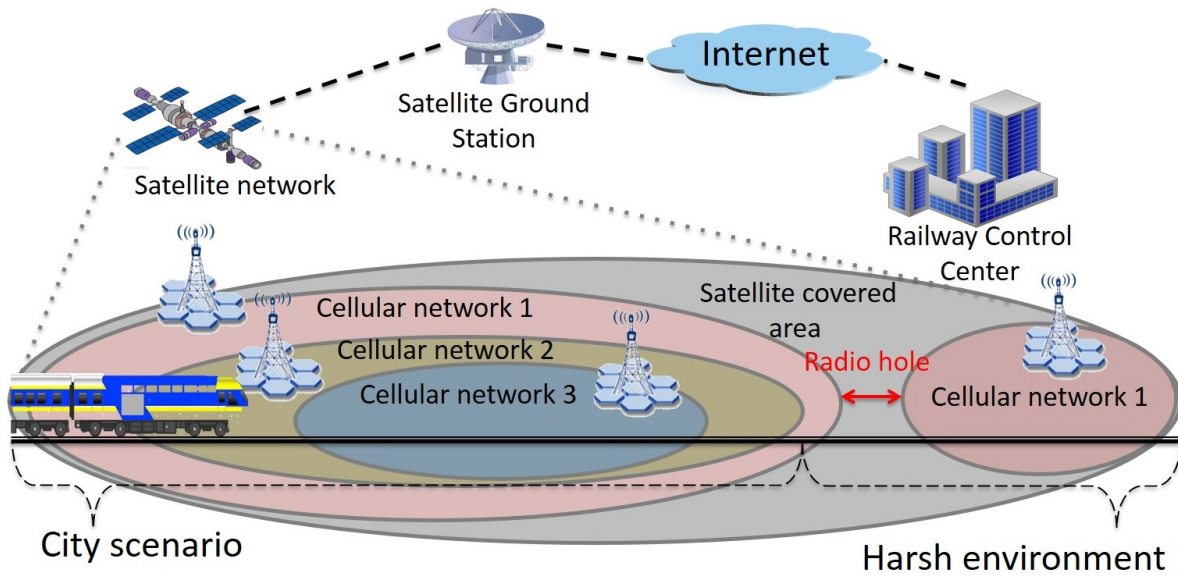


Figure 2.8: Test scenario with overlapping QoS link and best effort links (cellular network 1, 2 and 3).

Gradually, the cellular network 2 breaks down as well and train drives into harsh environment. When the connection over cellular network 1 also drops, the train falls into the radio hole, where only satellite connectivity is guaranteed and all data transmission will be handled exclusively by satellite link. Finally, as soon as a new available PLMN (cellular network 1) comes out, the adding strategy will be executed again.

2.5.3 Testbed software and hardware setting

The experiments have been executed at system level. The testbed has been built on a Dell Inspiron 13 computer; with 2.5 GHz Intel four cores i7 processor, 8 GB RAM and Windows 10 operating system. VirtualBox 4.3.16 [61] has been installed over the host system. In the virtualized environment, there were set four Virtual Networks named (i) vboxnet0, (ii) vboxnet1, (iii) vboxnet2, and (iv) vboxnet3.

Two virtual machines with Ubuntu 14.04 32 bit operation systems have been installed in VirtualBox. Each virtual machine has been configured with 2048MB RAM, 50 GB dynamic storage and four network adapters matching with four virtual networks, respectively. In each virtual machine, a MultipathTCP enabled Linux Kernel (version 3.16) has been built [62]. The traffic was generated by D-ITG (Distributed Internet Traffic Generator) [63]. With this software, we can also acquire the network perfor-

mance features, such as average bit rate, delay, jitter, and delay standard deviation. All the above features can be acquired on both two transceiver sides. An emulator, written in Python, is used to create the dynamical network profile for the selected scenario and the two virtual machines are time synchronized by Network Time Protocol (NTP) [64]. Both inbound and outbound traffic was monitored and recorded through Wireshark [65]. The testbed illustration is shown in Figure 2.9.

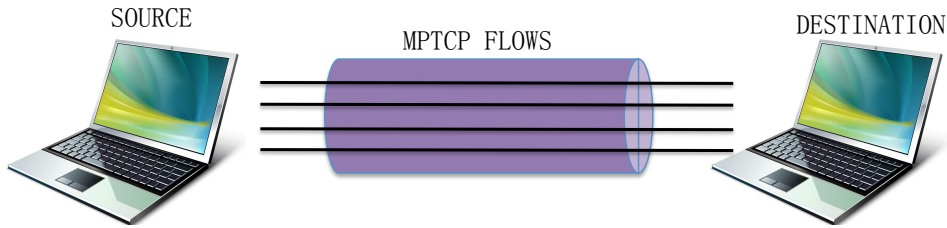


Figure 2.9: Illustration for testbed.

2.5.4 Obtained Results

A detailed picture for tested scenario is shown in Figure 2.10. For our test scenario, the inner ellipses areas represent the regions covered by PLMNs. Values of delay and bandwidth employed in the performance evaluation have been set in accordance to the results of the trial campaign performed in ESA ARTES 20 3InSat Project [66]. The data transmission rate has been set to 400 kbit/s.

The simulation result is shown in Figure 2.11. The communication begins with satellite network (black line) at 0s. After 15s, cellular network 1 joins into the active session and data pass through both two networks (black and red lines). Then, at 50s, cellular network 2 joins in and at 150s cellular network 3 is added as well (green and blue lines, respectively). Then according to the path management policy, the satellite interface shuts down around 180s (black line goes down). Thus, from 180s to 280s the data transmission is supported only by cellular networks. With the fading of cellular network 3 at 280s (blue line goes down), satellite interface switches on (black line appears again at 290s), and then cellular network 2 and 1 break down at around 370s and 450s, respectively (green line and red line disappear one by one). The radio hole lasts about 80s, from 450s to 530s, where only satellite network is active (all throughput

is over black line). After that the cellular network 1 comes out again at 530s, and data transport from both satellite network and cellular network 1 till the end of transmission (black line and red line working together).

We evince that the proposed architecture can well support railway communications. During the entire transmission interval, both in city and harsh scenarios, the total throughput does not change so much. The adding and dropping procedures among networks are taken smoothly. This can provide the required stability and reliability of the data transmission for train and control center communication.

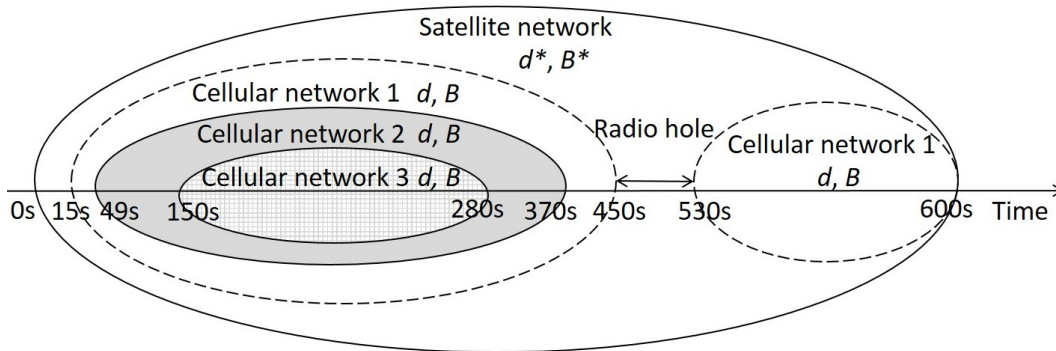


Figure 2.10: Test scenario for city and harsh environments. In the city environment, three cellular networks exist with delay $d = 150$ ms, and bandwidth $B = 2048$ kb/s. They are illustrated by dashed line, gray and grid ellipses. The city environment is also covered by satellite network with higher delay and lower bandwidth (*i.e.*, $d^* = 500$ ms, $B^* = 512$ kb/s). In the harsh environment, there is a radio hole, and it is only covered by one cellular network (dashed circle) and satellite network (big solid line circle). The cellular network has delay $d = 150$ ms, and bandwidth $B = 2048$ kb/s; while the satellite network has higher delay and lower bandwidth, $d^* = 500$ ms, $B^* = 512$ kb/s. For easy of reference, the train moves from left to right along the railway track, and the time corresponding to the transition between two adjacent areas is also reported.

2.6 Further studies on MPTCP performance

Now, in order to well evaluate the performance –expressed in terms of stability, continuity and reliability– of the proposed architecture, we present the analysis of MPTCP performance, considering both *static* and *dynamic* experiments. With the term static, we mean a scenario where channel delay does not change along time. On the other hand, a dynamic scenario occurs where the channel delay changes in different channels,

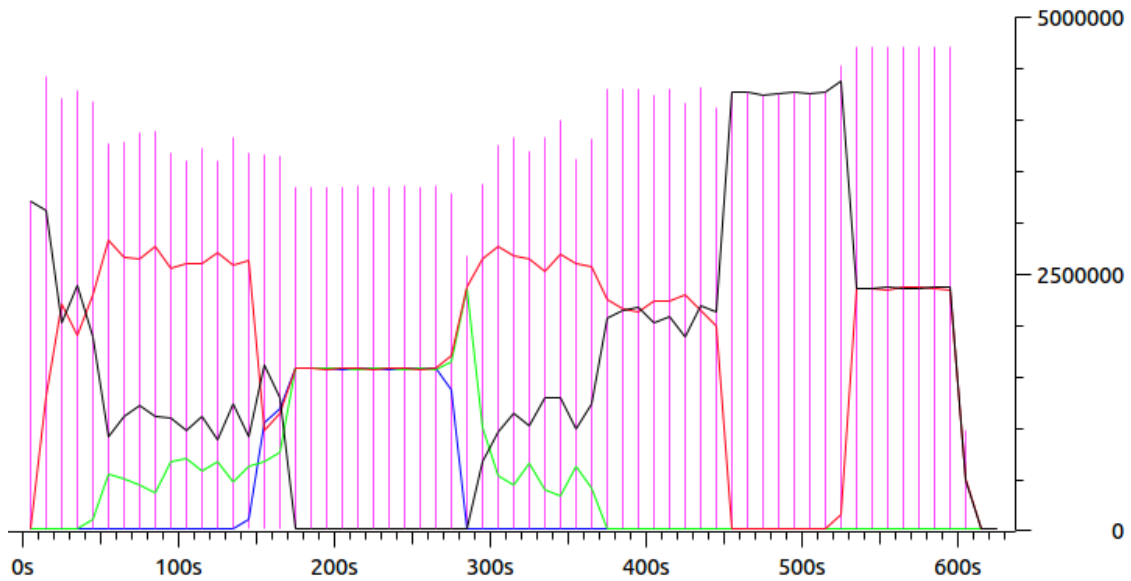


Figure 2.11: Throughput [bit/s] experienced for different connection links *i.e.*, satellite network (*black line*), cellular network 1, 2 and 3, corresponding to *red*, *green*, and *blue lines*, respectively. Light purple bars show the total throughput during the entire connection period.

Table 2.2: Parameters used in the simulations.

Network	Delay [ms]	Jitter [ms]	Loss Rate
Cellular	150	15	0.5%
Satellite	500	50	2%

specifically one channel is meant having a fixed delay, and the other channel with a dynamic delay.

2.6.1 Static Experiments

In static tests, all the scenarios have two subnets with round robin scheduler. Cellular and satellite links are imitated as the network environment. The two links are characterized by typical delay, jitter and packet loss values [66], as illustrated in Table 2.2. We assume two values for channel bit rate and buffer size *i.e.*, (i) 512 kbit/s with 512 kbit buffer size, and (ii) 2048 kbit/s with 2048 kbit buffer size. We assess six test scenarios as shown in Table 2.3. For each scenario we consider two network interfaces (*i.e.*, eth0, and eth1), each characterized by specific values for network parameters. As an instance, for scenario S1, we consider eth0 interface with 500 ms delay,

Table 2.3: Static Experiments Setting

Scenario	Delay [ms]	Jitter [ms]	Pkt Loss	Capacity [kbit/s]	Buffer [kbit]
S1 (<i>asym</i>)	500	50	2	512	512
	150	15	0.5	2048	2048
S2 (<i>sym</i>)	150	15	0.5	2048	2048
	150	15	0.5	2048	2048
S3 (<i>sym</i>)	500	50	2	2048	2048
	500	50	2	2048	2048
S4 (<i>sym</i>)	500	50	2	512	512
	500	50	2	512	512
S5 (<i>asym</i> + <i>sym</i>)	500	50	2	2048	2048
	150	15	0.5	2048	2048
S6 (<i>sym</i> + <i>asym</i>)	150	15	0.5	512	512
	150	15	0.5	2048	2048

50 ms jitter, and 2% of packet loss.

The terms “*asym*” and “*sym*” mean the scenarios have a (a)symmetric behavior for the two interfaces *i.e.*, eth0, and eth1, and for different features, respectively. The terms (i) “(*asym* + *sym*)” and (ii) “(*sym* + *asym*)” mean that the scenario has a hybrid behavior, with (i) asymmetric delay, jitter, and packet loss, with a symmetric capacity and buffer, and (ii) symmetric delay, jitter, and packet loss, with an asymmetric capacity and buffer, respectively. Each scenario lasts 60 seconds, and is comprised of five experiments with transmission rates [24, 40, 120, 240, 400] kbit/s. On both sender and receiver sides, packet information and network performance features are recorded. Notice that the values of transmission rate are lower than the channel capacity, because we consider railway control data transmissions, and typically train-control center communication bit rate is lower than 12.2 kbit/s (voice service) [37].

For the experimental results, we obtain (i) the average bitrate, (ii) the average delay, (iii) the delay STD (standard deviation), and (iv) the average jitter, as shown from Figure 2.12 to 2.15. Each result has been assessed for different scenarios (*i.e.*, from S1 to S6), versus different transmission rates *i.e.*, [24, 40, 120, 240, 400] (kbit/s).

In Figure 2.12, we find that different network environments do not strongly influence the receiving bitrate, and we experience very similar values independently from the particular scenario.

Figure 2.13 shows the average delay for different scenarios versus the transmission rate. The behavior of each "cluster" (*i.e.*, a group of experimental results obtained for different scenarios and for a given transmission rate) looks similar to the others. As an instance, for different bitrates, we notice that the highest value of delay is experienced in S5, and the lowest value is for S6. S5 has the asymmetric setting on channel delay and jitter, and the symmetric setting on channel capacity and buffer size, as shown in Table 2.3.

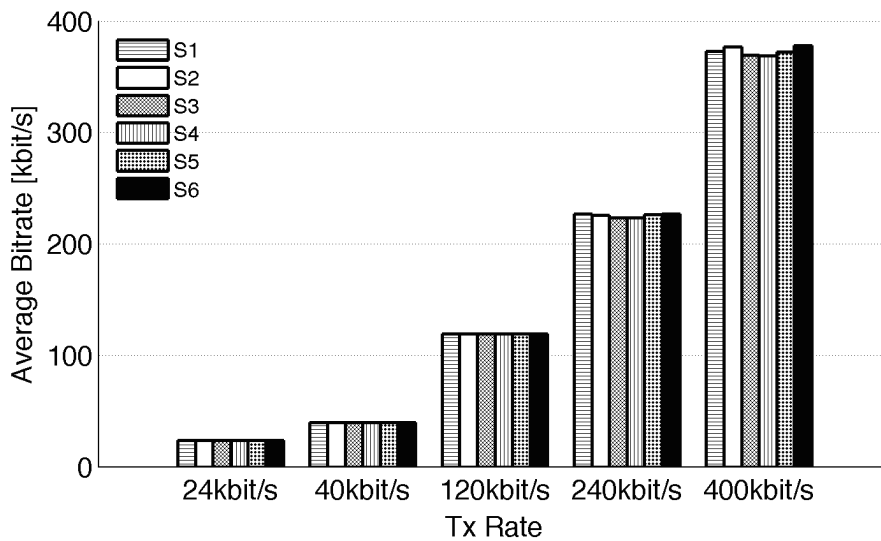


Figure 2.12: Static experiments. Average receiving bitrate vs. transmission rate for different scenarios. The behavior is independent from the particular scenario.

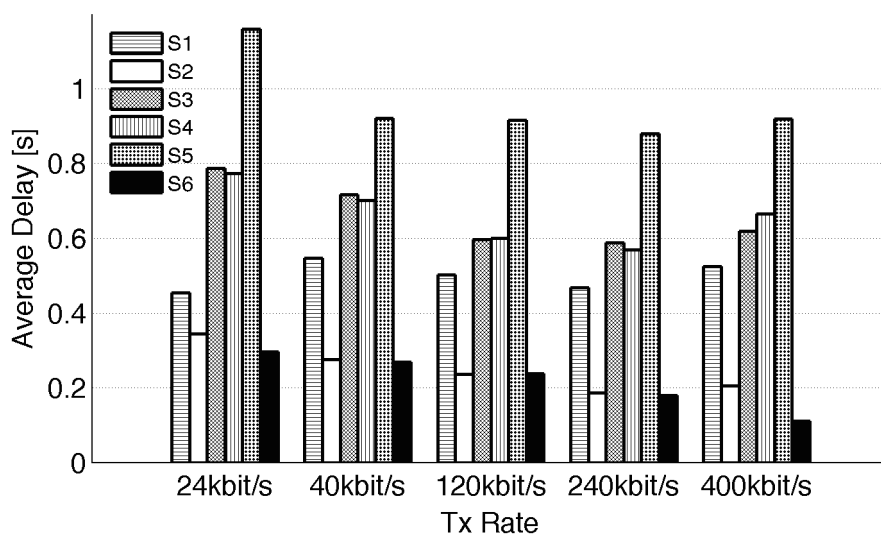


Figure 2.13: Static experiments. Average delay vs. transmission rate for different scenarios.

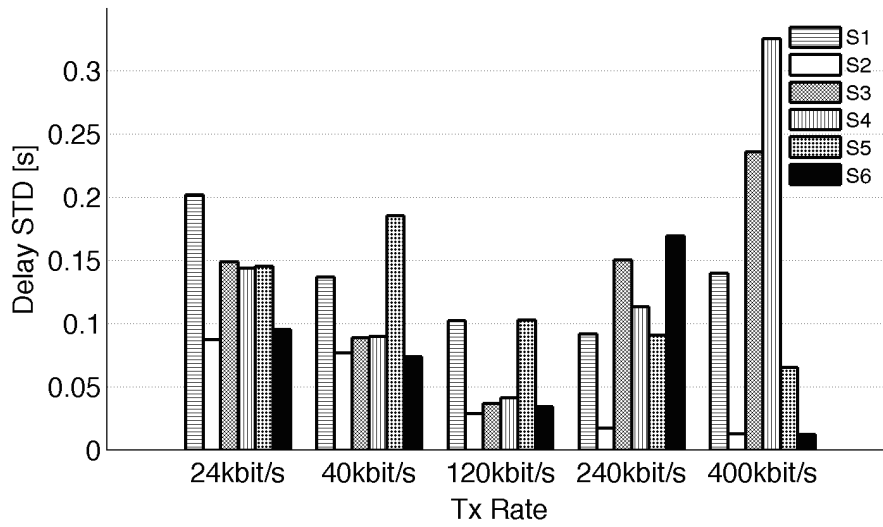


Figure 2.14: Static experiments. Average delay STD vs. transmission rate for different scenarios.

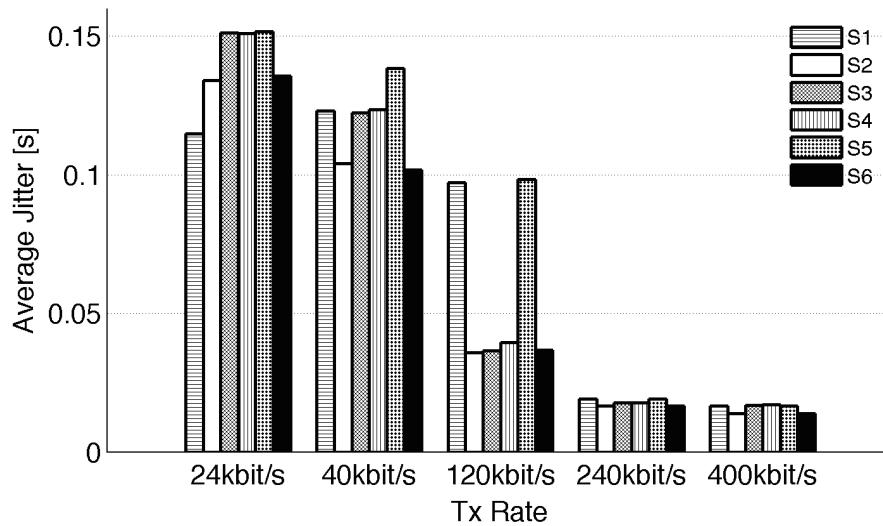


Figure 2.15: Static experiments. Average jitter vs. transmission rate for different scenarios.

This means that there are no big disparities for different transmission bitrates, but the divergence appears among different scenarios.

From Table 2.3, S1 (*asy*) has the same channel delay and jitter as S5 (*asy + sym*), but asymmetric channel capacity and buffer size. Hence, the average delay is evidently lower than S5. This means that the best channel (*i.e.*, smaller delay and jitter) should have the biggest buffer size. This can force the large part of data transfer over the best channel. Moreover, S5 has smaller channel delay than S3 and S4, same buffer size as S3 and larger buffer than S4. However, S5 has the worse performance on average delay

than the other two. It shows that with round robin scheduler the symmetric channel delay and jitter scenario has better performance than the asymmetric scenarios.

S3 and S4 have the same symmetric behavior, but channel capacity and buffer size in S3 are higher than those in S4. However, the average delay for those two scenarios are almost the same (*e.g.*, for 24 kbit/s we obtain an average delay of 0.787 s, and 0.773 s for S3 and S4, respectively). This means that, even if the channel capacity and buffer size increase, the delay does not change because of the transmission rate is always lower than the channel capacity.

Finally, we compare S2 and S5, S6 and S1. It is easy to know that under the same channel capacity and buffer size setting, the symmetric channel delay and jitter scenarios (*i.e.*, S2 and S6) have the smaller average delay (*i.e.*, 0.344 s in S2 is lower than 1.159 s in S5). This means that if we use two PLMNs to support the transmission simultaneously, it is better to select PLMNs with similar conditions (*i.e.*, delay and jitter settings). So, we can conclude that, with round robin scheduler when the transmission rate is lower than the channel capacity and buffer size, the symmetric delay and jitter scenarios should have the minimum average delay (*e.g.*, S2 and S6 have very low average delays).

In Figure 2.14 the delay standard deviation changes fast among different tests. But we can find that for different transmission rates, S2 (minimal delay and jitter setting and maximal channel rate and buffer size) always has the lowest STD. That means the better channel condition can have more stable transmission status.

Finally, average jitters are quite similar for the same transmission rate, as depicted in Figure 2.15. Unlike the average delay and the delay standard deviation, with the same transmission rate, jitter does not change a lot among different scenarios. When the transmission rate is much lower than the total buffer size, with the increase of transmission rate, the average jitter decreases fast. Transmission rate at 400 kbit/s gets the smallest average jitter in our experiments (*i.e.*, 0.016 ms).

To summarize, the static experiments have shown that based on the typical parameters of cellular and satellite networks, when the transmission rate is lower than the channel capacity and buffer size, the scenarios with symmetric delay and jitter have the minimum average delay. Also, the scenarios with symmetric channel delay and jitter

outperform the asymmetric one. Average jitter is mainly depended on the transmission rate, when the transmission rate is much lower than the total buffer size, and it decreases with the increasing of transmission rate.

2.6.2 Dynamic Experiments

In the dynamic experiments, all the scenarios are tested with two network interfaces (*i.e.*, eth0 and eth1). The data transmission always starts through eth0. Each test lasts 1 minute, and the channel delay for eth1 is assumed dynamic, changing after 30 seconds. In particular, we test two values of channel delay (*i.e.*, 100 ms, and 1000 ms), and two values of bit rates (*i.e.*, 512 kbit/s, and 2048 kbit/s). We obtain 16 tests, each one assessed for three values of transmission rates *i.e.*, [12, 40, 400] kbit/s. Scenarios from T1 (test 1) to T8 (test 8) have a 100 ms delay for eth0, while scenarios from T9 (test 9) to T16 (test 16) have a 1000 ms delay. The setting of the dynamic experiments is illustrated in Table 2.4.

The analysis of the experimental results concentrates on the same factors for the static experiments on the receiver side, such as (*i*) the average bitrate, (*ii*) the average delay, (*iii*) the delay STD, and (*iv*) the average jitter.

The average receiving bitrate is shown in Figure 2.16. The behavior is very similar among all the tests. This means that as far as the transmission rate is lower than the channel capacity, different setting of delay and channel rate does not influence the receiving bitrate. The same consideration has been taken for the static experiments as well. On the other side, the average delay highly depends on the setting of different tests (specifically, the delay of eth0 interface). As shown in Figure 2.17, scenarios from T1 to T8 have less average delay than results in T9-T16. Notice that scenarios from T1 to T8 with a sending rate of 40 kbit/s (gray columns) have the lowest average delay. For all three kinds of sending rate, from T5 to T8 the average delay is lower than the other scenarios. Finally, from T9 to T16 we can observe that the highest delays occur for lowest transmission rate (*i.e.*, 12 kbit/s).

The delay standard deviation shows the distribution of packet delay for each test. As expected, the lower standard deviation means a more stable transmission. The delay

Table 2.4: Dynamic Experiments Setting

Scenario	Delay [ms]			Bitrate [kbit/s]	
	Eth0	Eth1	Eth1	Eth0	Eth1
	0s-60s	0s-30s	30s-60s	0s-60s	0s-60s
T1	100	100	1000	512	512
T2	100	100	1000	512	2048
T3	100	100	1000	2048	512
T4	100	100	1000	2048	2048
T5	100	1000	100	512	512
T6	100	1000	100	512	2048
T7	100	1000	100	2048	512
T8	100	1000	100	2048	2048
T9	1000	100	1000	512	512
T10	1000	100	1000	512	2048
T11	1000	100	1000	2048	512
T12	1000	100	1000	2048	2048
T13	1000	1000	100	512	512
T14	1000	1000	100	512	2048
T15	1000	1000	100	2048	512
T16	1000	1000	100	2048	2048

STD does not only depend on the fix delay level, but also on the transmission rate, as shown in Figure 2.18. In the first eight tests, the lowest delay STD occurs for a sending rate of 40 kbit/s (gray columns). Also, scenarios T1-T8 have much higher STD than T9-T16, but only in the case of 400 kbit/s.

Finally, the average jitter is depicted in Figure 2.19. Tests with 400 kbit/s (white columns) have a stable behavior with respect to other transmission rates. Tests from T1 to T8, for transmission rate 12 kbit/s (black columns) and 40 kbit/s (gray columns), have less jitter than tests from T9 to T16.

To summarize, in the dynamic experiments we observe the following behavior:

- the largest average delay (*i.e.*, 1.533 s) is experienced in T11 for 12 kbit/s, and has the biggest average jitter (*i.e.*, 0.172 s);
- tests with eth0 1000 ms delay (*i.e.*, from T9 to T16) have larger average delay and jitter w.r.t. other scenarios.
- the largest average delay has the smallest delay STD (*i.e.*, 0.673 s);
- the largest sending rate has the smallest average jitter.

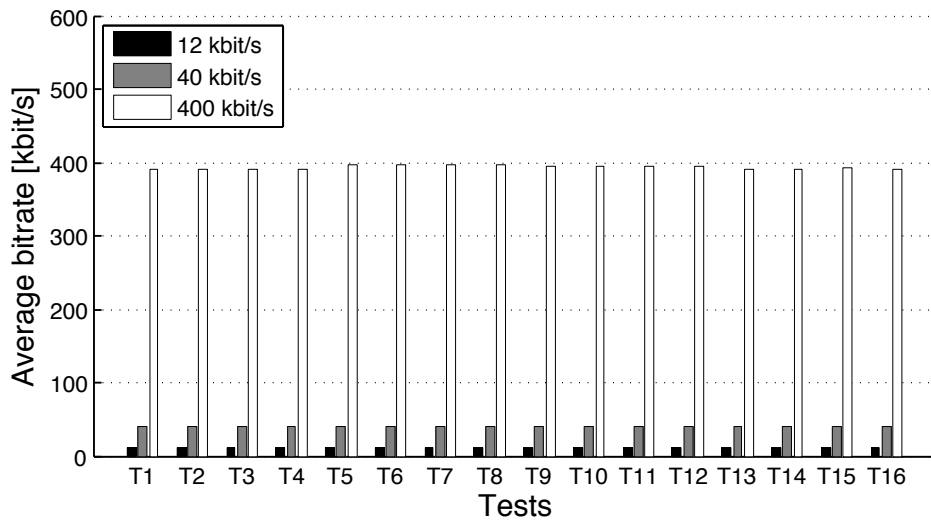


Figure 2.16: Dynamic experiments. Average receiving bitrate vs. transmission rate for different tests.

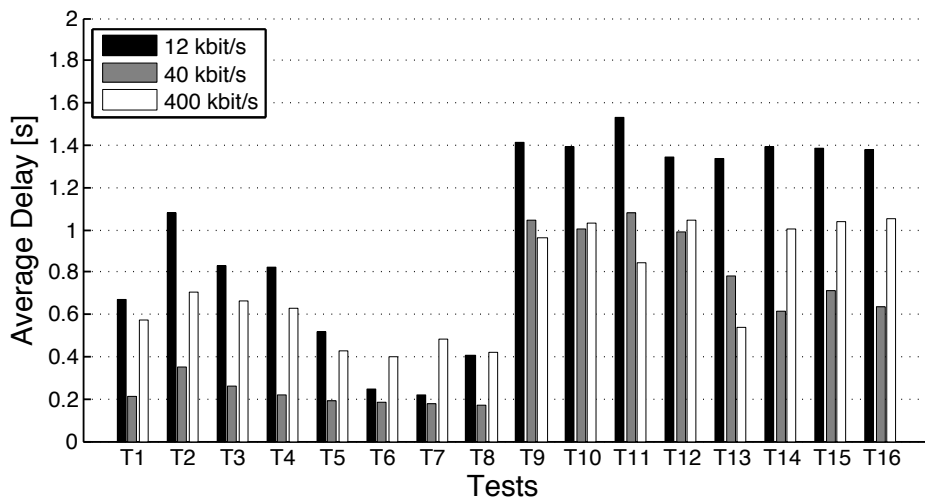


Figure 2.17: Dynamic experiments. Average delay vs. transmission rate for different tests.

2.7 Conclusions

In this chapter, a new communication system architecture is presented, as viable alternative solution to GSM-R for train control/management system. We consider the coexistence of multiple wireless networks (*i.e.*, PLMNs) with satellite networks, so that the communication can be executed simultaneously over multiple subflows via MPTCP. A dedicated MPTCP “add and drop” subflow logic has been introduced, with the aim of providing handover procedures for seamless connectivity.

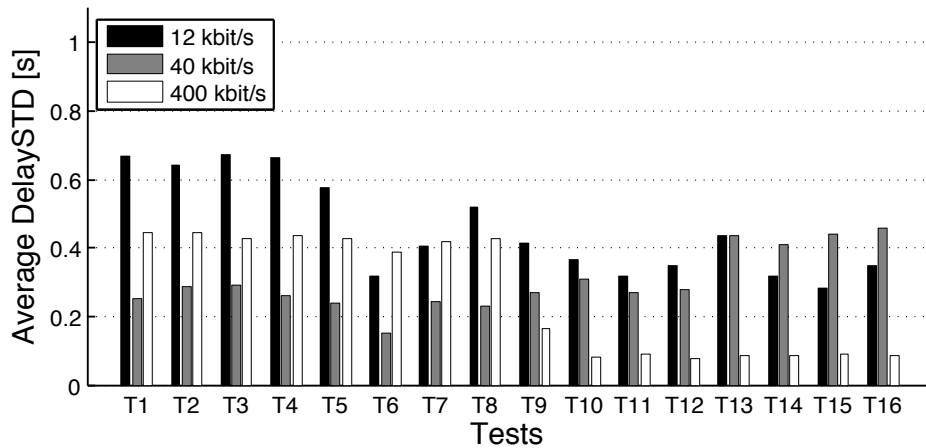


Figure 2.18: Dynamic experiments. Average delay STD vs. transmission rate for different tests.

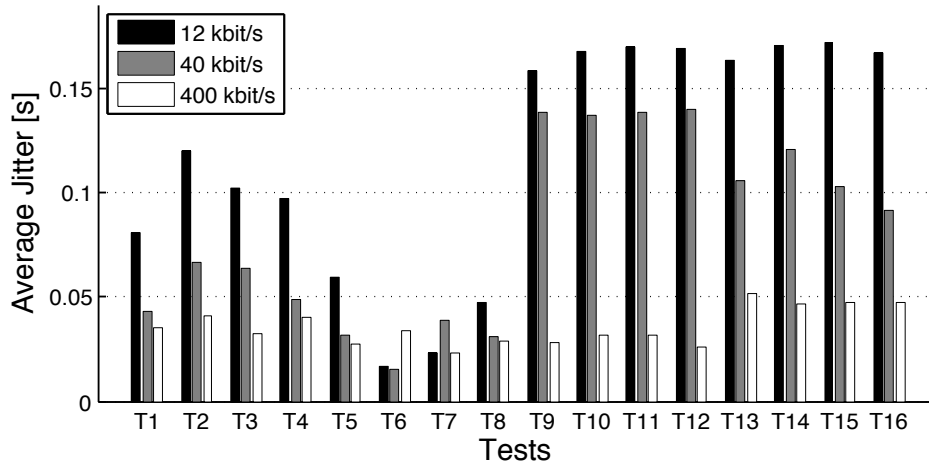


Figure 2.19: Dynamic experiments. Average jitter vs. transmission rate for different tests.

The static experiments have shown that based on the typical parameters of cellular and satellite networks, when the transmission rate is lower than the channel capacity and buffer size, the scenarios with symmetric delay and jitter have the minimum average delay. Also, the scenarios with symmetric channel delay and jitter outperform the asymmetric one. Average jitter is mainly depended on the transmission rate, when the transmission rate is much lower than the total buffer size, and it decreases with the increase of transmission rate.

Finally, the proposed architecture has been assessed also in dynamic scenarios. As expected, best performance are for scenarios with lowest fixed channel delays. The dynamic experiments have shown that larger average delay gets smaller delay deviation;

while jitter depends on the sending rate, the larger sending rate the smaller average jitter.

As future work, we aim to assess the proposed approach via an experimental testbed, and compare the simulation results with those coming from the real testbed.

Chapter 3

Train control voice service supporting based on Multipath UDP links

3.1 Introduction

ERTMS has two main components, ETCS and GSM-R. There are four main services carried by GSM-R: control information and signaling exchanging between train and control center, voice communication between train driver and control center staffs, regular voice communication between control center staffs and railway maintenance workers, and emergency call between train drivers and railway maintenance worker.

In chapter 2, we have proposed the signaling data transmission solution by introducing MPTCP protocol in a new communication architecture with public networks. In this chapter we will give out our MPUDP solution for the voice service which used to be supported by GSM-R. In particular, we use IP based voice service for the train control system instead of the traditional voice call. The RaptorQ encoded voice packets will be dispatched over multiple UDP links.

In order to better understand our new MPUDP solution, first we need to answer some easy questions. What is VoIP? How the VoIP works? Is there any advantage by using VoIP? What is the business model for it? What is the different between our new MPUDP solution and the traditional VoIP solution?

1. **What is VoIP?**

VoIP is an acronym for Voice over Internet Protocol, which, in more common, uses for phone service over Internet. It sends voice information in digital form in discrete packets over Internet, rather than by using public switched telephone network (PSTN). VoIP has a set of facilities to manage the delivery of voice communications and multimedia sessions, which is supported by a methodology and group of technologies over Internet Protocol (IP) networks. Some services such as IP telephony, Internet telephony, broadband telephony, and broadband phone service are always associated with VoIP.

2. How the VoIP works?

VoIP uses the real-time protocol (RTP) to ensure that packets are delivered in a timely way, however, it is currently just offer the best effort service over public networks. A better service is possible with private networks managed by an enterprise or by an Internet Telephony Service Provider (ITSP).

The procedure to initiate a VoIP telephone calls are similar to the traditional digital telephony, but also involves signalling, channel setup, digitization of the analog voice signals, and encoding. Then the digital information is packetized, and transmission occurs as IP packets over a packet switched network, where the traditional call always delivers over PSTN.

Currently, there are many instruments, platforms and software, which can support the VoIP services, such as WeChat, WhatsApp, and Skype. A standard implementation is shown in Figure3.1

3. Is there any advantage by using VoIP?

There are two major reasons to use VoIP: lower cost and increased functionality.

a) Lower Cost

In general, phone service via VoIP costs less than equivalent service from traditional sources. This is alternative function of traditional phone services. There are some cost savings due to using a single network to carry voice and data. This is especially true when users have existing under-utilized network

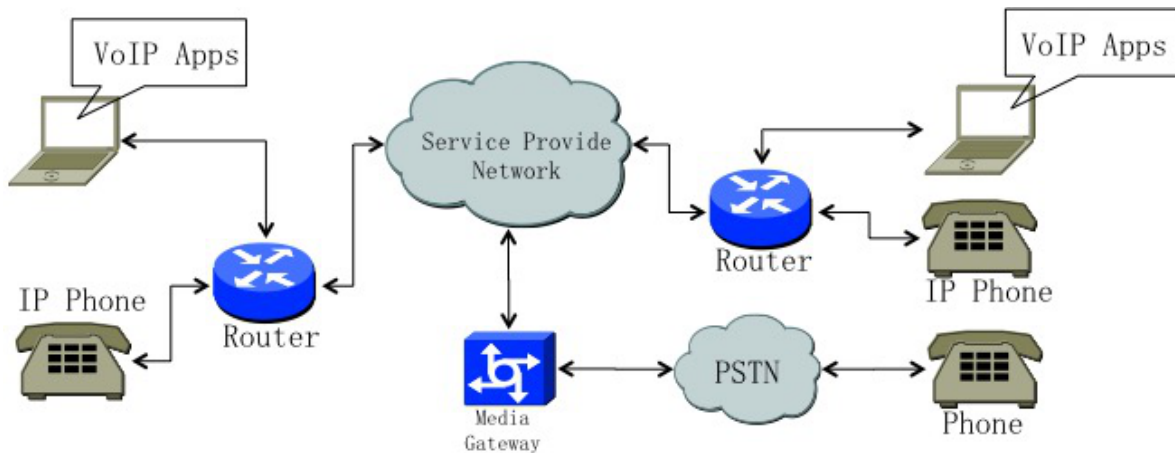


Figure 3.1: VoIP Standard implementation

capacity that they can use for VoIP without any additional costs. In the most extreme case, users see VoIP phone calls (even international) as free. While there is a cost for their Internet service, using VoIP over this service may not involve any extra charges, so the users view the calls as free. There are a number of services that have sprung up to facilitate this type of "free" VoIP call. Examples are: Free World Dialup and Skype.

b) Increased Functionality

VoIP makes easy some things that are difficult or impossible with traditional phone networks. For example:

- i. Incoming phone calls are automatically routed to your VoIP phone wherever you plug it into the network.
- ii. Take your VoIP phone with you on a trip. Anywhere you connect it to the Internet, you can receive your incoming calls.
- iii. Call center agents using VoIP phones can easily work from anywhere with a good Internet connection.

4. What is the business model for it?

The continuous development of network technology has influence in many industrial sectors. In particular, VoIP is one of the representative changing in telecommunication industry.

The VoIP providers can be divided into three main groups. The first group of VoIP providers offer the same business models, technical solutions and architectures as the legacy telephone network. The second group of providers, such as Skype, has created closed networks for users, who can get the benefit of free calls and convenience while potentially charging for access to other communication networks, such as the PSTN. However, this has limited the freedom of users to mix-and-match third-party hardware and software. Third generation providers, such as Google talk, use a Federated VoIP model to support the services. Federated VoIP uses voice over IP between autonomous domains in the public Internet without the deployment of central virtual exchange points or switching centers for traffic routing. These solutions typically allow dynamic interconnection between users in two different domains on the Internet.

According to the payment model, the business model for VoIP can be classified in three categories, free Internet telephony, advertising-based Internet telephony and integration free Internet telephony.

- a) Free Internet telephony: VoIP providers offer subscribers with full VoIP functionality services. In order to increase the facilities, some of the providers furnish hardware for VoIP services as well.
- b) Advertising-based Internet telephony: a number of providers use advertising as a business model. Most advertising-based products do not provide the full functionality, in other words, they have limitations, such as limited in specific locations or restricted by the time for each call.
- c) Integration free Internet telephony: some platform offer this category services for users such as Voxalot and Mysipswitch. Those platforms allow you to merge several free Internet phone (VoIP) providers and effective use of them.

5. What are the different between our new MPUDP solution and the traditional VoIP solution?

In the traditional VoIP solution, the voice data is delivered always over only one link and it has no recovery strategy for lost packets. The service quality is highly depending on the link delay and packet loss rate.

In our MPUDP solution, we introduce multiple UDP links in the communication architecture to support the voice packets delivery. Meanwhile, a FEC(forward error correction) coding protocol-RaptorQ is used in MPUDP solution, too. Before the voice packets sent to the UDP link, we encode them with RaptorQ protocol. RaptorQ can bring in the redundancy during encoding procedure, and decoding can be executed regardless with packets order. By the help of RaptorQ, MPUDP solution well solves the packets arriving out of order problem, and rises up the packet loss tolerance, as well.

3.2 Studies on existed VoIP multipath transmission solutions

In the railway using scenario, we will implement the VoIP service over multiple paths. That means during one communication session, the voice packets will be delivered over more than one flow, and the flows will not influence each other. Multipath VoIP service should have more reliability and capacity than the traditional VoIP service (Figure3.2). There are three solutions that we will discuss for the multipath voice service.

3.2.1 Multipath RTP (MPRTP) protocol

Nowadays, the Internet devices always can support more than one network path between two end users. Thus, some transport protocols, such as MPRTP and Multipath TCP (MPTCP), can support the multiple transmissions.

Some researches have been done for the multiple transmissions based on RTP protocol. In [67], authors proposed multi-stream transmission of real-time voice over best-effort packet networks. The results show that transport over multipath can significant reduce end-to-end latency and loss rates compared to FEC protected single-path trans-

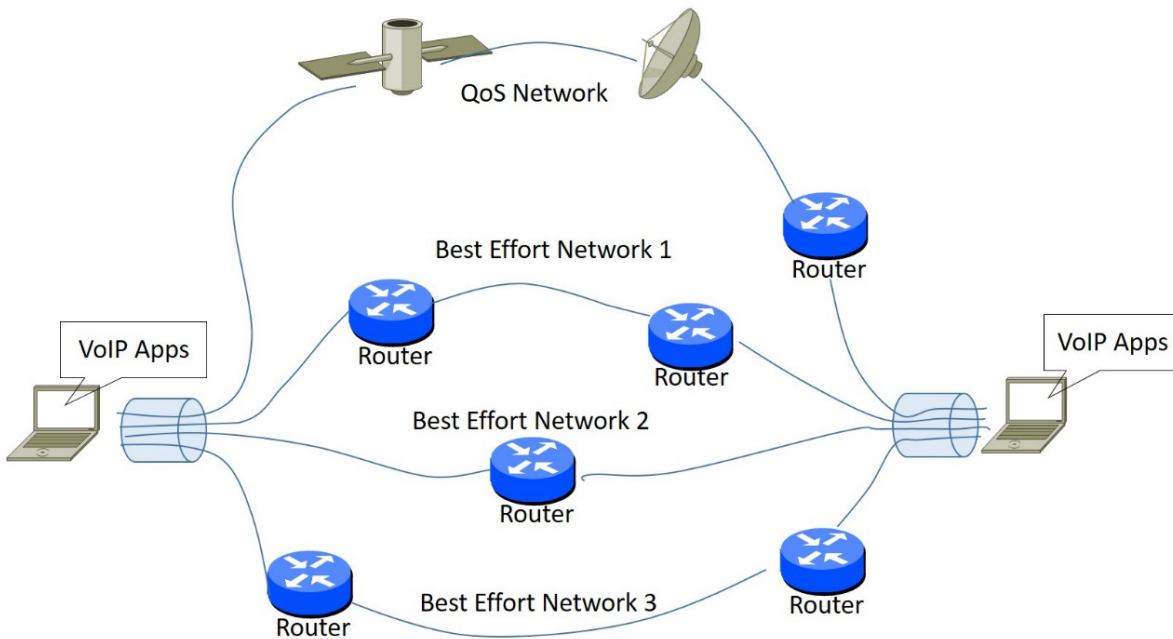


Figure 3.2: VoIP normal implementation

mission at the same data rate. John G. Apostolopoulos in [68] introduced two architectures for achieving path diversity, and examine the effectiveness of path diversity in communicating video over a lossy packet network.

3.2.1.1 Architecture and stack

In a typical scenario, an RTP session uses a single path. In an MPRTP scenario, the control signal is over a TCP path; while an RTP session uses multiple subflows (each subflow uses a different path), as described in Figure3.3. Figure3.4 illustrates the stack for the MPRTP solution.

Application layer sends single flow to MPRTP manager and MPRTP manager splits it into multiple RTP subflows. Each subflow is sent along a different path to the receiver. To the network, each subflow appears as an independent, well-formed RTP flow. At the receiver, the subflows are combined to recreate the original RTP session.

3.2.1.2 Advantages and disadvantages

This solution mainly bases on MPRTP protocol. MPRTP is independent protocol, and it extends the RTP/RTCP protocol. It has several control mechanisms itself, such

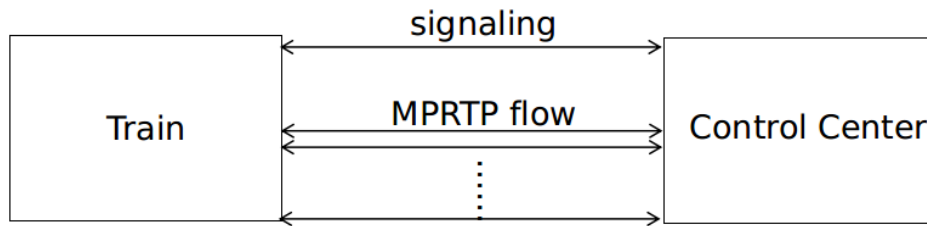


Figure 3.3: Architecture for MPRTTP

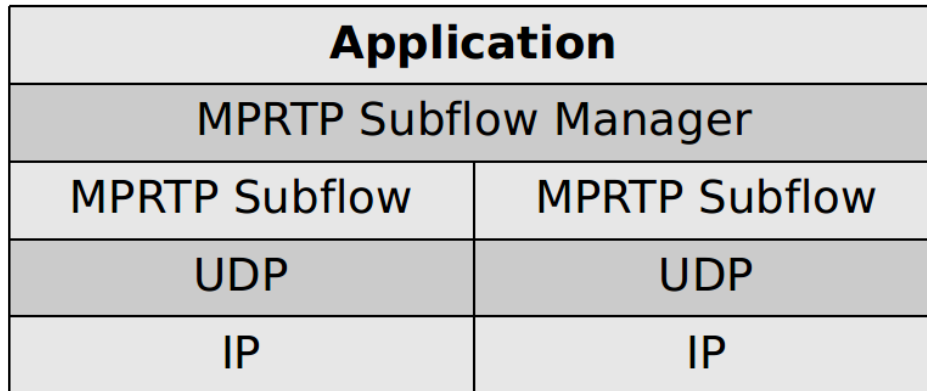


Figure 3.4: Protocol stack for MPRTTP solution

as connectivity check, adding new or updating interface and congestion control. Those mechanisms can support the application layer to well manage the session connection. Moreover, MPRTTP protocol should be backward compatibility, all applications which base on RTP protocol, can move to MPRTTP protocol without modification. Meanwhile, MTRTP usually bases on UDP protocol, and this should provide better performance on delay and jitter compare with delivery over TCP protocol.

But there are also some disadvantages. The MPRTTP protocol is still published as draft version, thus it may change in the future. All RTP paths are based on UDP protocol, thus the connections are best effort connections, which cannot guarantee the QoS. That means some packet may be lost during the delivery. In addition, there is only GStreamer implementation but not a system level implementation for this moment, which brings in the difficulties for system level realization.

3.2.2 Multipath TCP (MPTCP) protocol

MPTCP protocol is the expending for TCP protocol, and it has been defined by IETF with RFC6824 [41]. MPTCP can well use the network resource redundancy

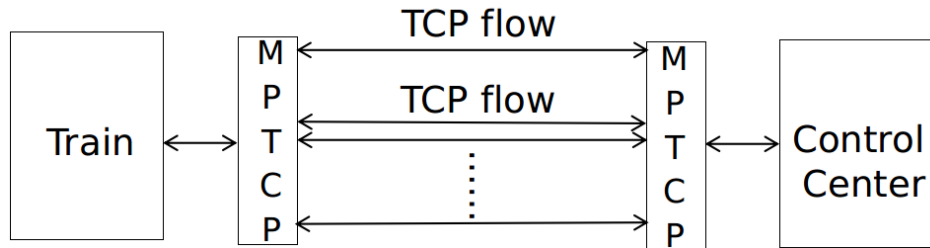


Figure 3.5: Architecture for MPRTP solution

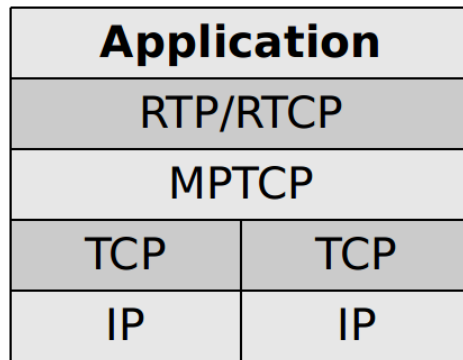


Figure 3.6: Protocol stack for MPTCP solution

by delivery the packets through several path . All paths are independent with each other, thus breaking or congestion of one connection path will not influence the other connections.

3.2.2.1 Architecture and stack

The MPTCP based connection has at least one TCP connection path. The packet will be delivered over several different TCP link. The architecture is shown in Figure3.5. The protocol stack is shown in Figure3.6. Between application layer and TCP layer, the protocol adds a MPTCP layer, which can distribute the packets in to different TCP paths based on the scheduling strategy and the connection capacity and congestion status. On the contrary side, before pass the receiving packets to the receiver, MPTCP layer reorganize the packets with a correct order as it in the sender side.

3.2.2.2 Advantages and disadvantages

The MPTCP protocol has many advantages as a candidate solution. First, MPTCP has been defined by IETF as RFC6824. It has been implement on the Linux system

and many researches have tested the performance for this protocol. Second, we already have working experiences on MPTCP protocol. In last chapter, we proposed a new architecture for the railway communication by means of MPTCP, and we have successfully implemented this architecture on system level. Third, we have learnt the MPTCP protocol characteristics on dynamic railway using scenarios. Hundreds tests have been done for performance evaluation of MPTCP based dynamic symmetry or asymmetry networks. Fourth, when the voice packets are delivered over TCP links, the lost packets will be retransmitted by the retransmission mechanism. For this reason, MPTCP solution can be considered as a QoS guaranteed solution for multipath VoIP packets delivery.

However, this solution also has some disadvantages. First, the VoIP applications are designed for UDP transmission, and they may need some modification to fit the TCP based transmission. Second, the packets delivered over TCP will lead to the low performance on delay. Third, since the packets are transported over different paths, the out of order problem is also need to be taken into consider.

3.2.3 MPT-GRE in UDP solution

The single-path transmission cannot take the advantages from the multiple available interfaces. RFC 6824 "TCP Extensions for Multipath Operation with Multiple Addresses" introduced extensions for the TCP protocol to extend the current TCP implementations for supporting multiple paths. However, the MPTCP works in the transport layer and is restricted to the TCP protocol. Multipath UDP has a significant defect, since the UDP protocol is designed for single packet and with no states. The packets arrived in different paths has no relevant with each other will cause the out of order problem and this problem is really harm for the application. So we cannot direct use multipath UDP as the solution. In order to solve this problem, MPT-GRE in UDP (multipath-Generic Routing Encapsulation in User Datagram Protocol) can be considered as a good solution based on UDP transmission. In this solution bases on the IETF draft GRE-in-UDP Encapsulation [69], GRE protocol can generate a tunnel for UDP encapsulation, and the multipath layer is under the GRE layer, then the packets

are delivered over multiple UDP paths. MPT-GRE in UDP can fix the gap for multiple UDP transmission [70]. Authors in [71] have successfully implemented the MPT-GRE in UDP transmission for HD video streams.

3.2.3.1 Architecture and stack

The architecture for MPT-GRE in UDP solution is slightly different from the previous solutions. Because the multipath layer is between the GRE layer and UDP layer. The architecture is shown as follow in Figure3.7. The application sends out packets and those packets are delivered over UDP protocol. After that, packets are encapsulated with GRE head and distribute by the MPT environment to different UDP flows. On the receiver side, the MPT environment recollects the packets and saves in the buffer to reorder the arrived packets. Then packets are passed to the receiver application by UDP transmission. For applications on both sides, the transportations are executed as in a normal UDP way. The protocol stack is picked in Figure3.8

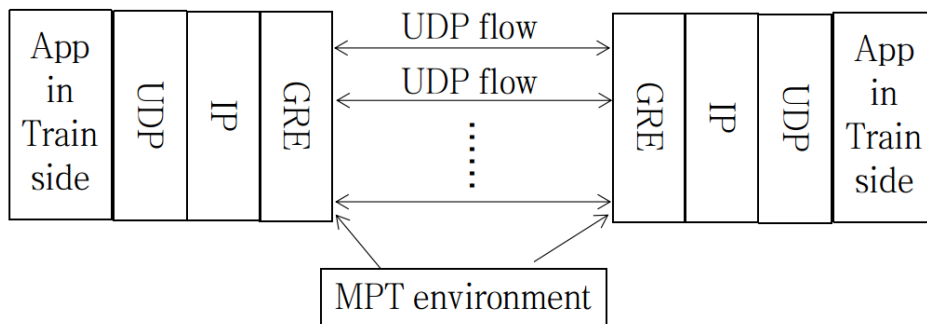


Figure 3.7: Architecture for MPT-GRE in UDP solution

3.2.3.2 Advantages and disadvantages

The purpose of MPT is to support the multiple transmissions over UDP protocol. The losing packets will not be retransmitted in this solution. This strategy can improve the quality of using experience. That means the lost packet may leads to a small silence slot instead of the long delay. MPT offers a map of packets to multiple interfaces dynamically, and this can solve the out of order problem. In addition, MPT does not require the modification of application software. Because from the application's point

Application	
RTP/RTCP	
UDP	
IP	
GRE	
MPT ENVIRONMENT	
UDP	UDP
IP	IP

Figure 3.8: Protocol stack for MPT-GRE in UDP solution

of view, it uses only single logical interface for the communication (doesn't see the multipath from application layer).

There are also disadvantages for this solution. For instance, MPT-GRE in UDP is still in library state, in the other words, there is no reliable implement for this solution now. What's more, even the GRE in UDP is still in the draft status and it is not a protocol yet. Beside, since the packets are delivered over UDP paths, there is no significant relevant between different paths. For the reason that the different network has different delay (e.g. WI-FI or 3G), the packets scheduling will be broken down and that caused packets out of order effect. However, the MPT layer should be able to reorder the packets by adding map on each packet. To reorder the arrived packets may need a big buffer and that also leads to a large delay on the receiver side. MPT environment can be considered as a good solution for multipath VoIP, but there are still some problems for the multipath UDP transmission. For instance, when one of the UDP paths gets congestion, MPT environment cannot automatically distribute the packets to other paths, but continuously distributes packets to all paths as the out-weight setting. What's more, for the MPT environment, when one physical interface shuts down and restarts, the packets will not automatically deliver through this interface again. The interface reactivation need to call the `"/mpt int XXX up"` command manually. For this reason, we need another solution to deal with the path congestion and interface reactivation problems.

3.3 Proposed new solution for train control voice service based on Multipath UDP links

All existed solutions have some drawback on the railway using scenario. Simple implementation of multipath UDP transmission has significant defects. There isn't a literal standard for multipath UDP and there isn't implementation for simple multipath UDP as well. UDP is the protocol for packets transportation without state. Thus, there is no relevance between two UDP packets. If we only focus on the transmission layer, it doesn't matter whether there is relationship between packets. But at the application layer, two UDP packets delivered through different paths must be received in same order as they were sent. Otherwise, the disordered packets will strongly influence the application on receiver side. This may lead to a heavy deterioration on the QoS.

Leveraging on the above mentioned motivations, we need some better solution, specially for railway scenario. In our work, we focus our attention on the railway voice communication, based on both QoS and best effort networks. We coalesce the RaptorQ protocol together with multipath UDP transmission to support the voice communication. The RaptorQ can well solve the packet loss and out of order problem, specially for big delay difference and large packet loss rate situations.

3.3.1 Architecture for MPUDP solution

The proposed communication system architecture is shown in Figure3.9. The multiple UDP links are generated between two gateways. The data will be sent and received by multiple UDP links simultaneously. The MP-UDP software should be installed on both sides of the gateways. Media softwares connect to the gateway through single link. It is worth to mention that the media software does not need to modify their I/O interfaces. In the other words, for the media software, it seems to communicate by the regular RTP protocol over single UDP link. In fact, when there is only one path between two gateways, the connection falls back to the regular UDP connection.

The MP-UDP software is designed with three kinds of schedulers: basic, balance and RaptorQ.

Basic simply duplicates the packets and sends the same packets over each possible UDP link.

Balance periodically sends the prob packets over each link, and the next few packets will continually send over the lest delay UDP link till the next period, when the prob packets reporting with the new delay condition of each link.

RaptorQ has the most complicated strategy. Thus, RaptorQ scheduler needs more data processing time than other two schedulers. In fact, it has a limitation on throughput for the real time processing. To maximum reduce the latency and improve the throughput, RaptorQ binds a group of voice packets together and encodes/decodes them at one time. In details, according to the length of waiting queue, RaptorQ encoding module first gathers a dynamical number of packets in its encoding buffer, then encodes them together. For example, after the RaptorQ encoding module fetching a packet, if the media software waiting queue is empty, RaptorQ will encodes the packet at once. But, when the waiting queue still contains 1 to 5 packets, RaptorQ will only put the fetched packet in its encoding buffer and goes to fetch another packet at once. The encoding happens only when the encoding buffer is equal to or more than 5 packets. All packets in encoding buffer will be bounded together and encoded at one time. If the waiting queue has more than 5 packets, the encoding procedure occurs when there are no less than 10 packets inside encoding buffer. In addition, the encoding symbol size automatically changes according to the packets size as well. Symbol size can be set as 32bytes, 64bytes, 128bytes, 256bytes and 512bytes. RaptorQ encoding module should always choose the less padding symbol size to encode packet. This two strategies can help RaptorQ scheduler to deal with the throughput limitation problem.

For better understanding how the MP-UDP software to work, here (Figure3.10) comes an example on how the media softwares to send and receive the packets with RaptorQ scheduler.

We assume that voice packets are first sent from software 1 (SW1) to software 2 (SW2), then SW2 replies SW1 and sends its own voice packets back to SW1. SW1 sends original voice packets to gateway 1 (GW1) through the physical interface eth0. Once GW1 receives a packet, it passes this packet to a dynamic encode buffer, and encode procedure will begin only if the buffer is full. The size of the encode buffer

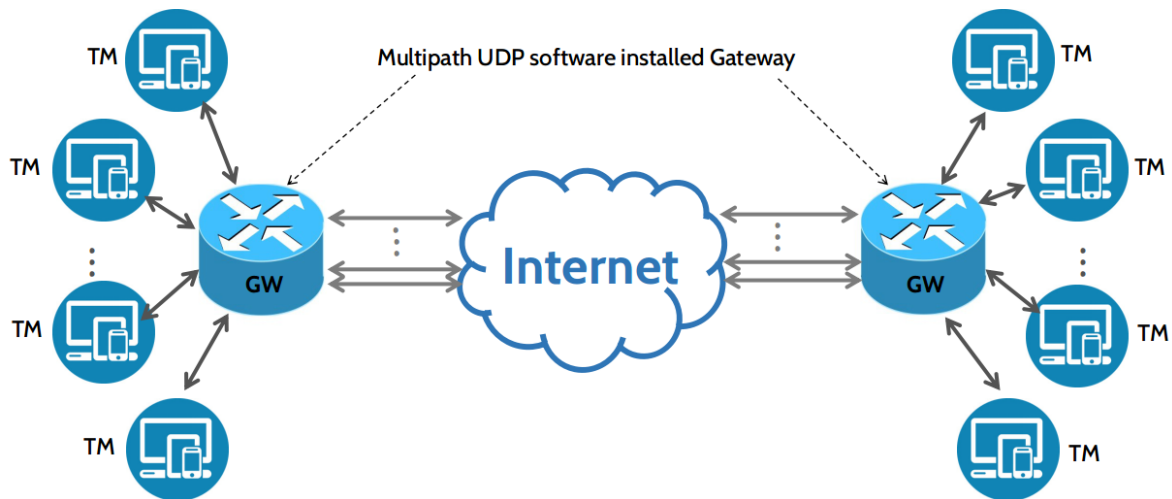


Figure 3.9: The architecture of proposed solution.

depends on how many packets waiting to be encoded. Packets in the encode buffer will be bound together as a single big packet, and this big packet will be encoded into at least two symbols according to the overhead rate by RaptorQ protocol. For example, with 0% of overhead rate, each big packet will be encoded into two basic symbols, with 50% overhead, each big packet will be encoded into two basic symbols and one overhead symbol. All symbols will be evenly distributed over all possible UDP links, that means three symbols will be delivered over three different UDP links. On GW2 side, the symbols are received by multiple physical interfaces (eth0 to ethn), and passed to the dynamical decode buffer. Decode procedure can be executed when the receiver has the same quantity of symbols as the basic symbols. For instance, with overhead 50%, three symbols will be sent out, and the big packet can be decoded when any two of this three symbols are received. Then the big packet will be converted to original voice packets. MP-UDP software passes this packets to media software SW2 over only one interface which connects with SW2. For replying to SW1, SW2 sends out its packets to SW1 following the same steps.

3.3.2 Stack of MPUUDP solution

The architecture for RaptorQ codes solution is on the application level, and it is different from all previous solutions. RaptorQ code environment receives a normal RTP session from the application, and then it encodes and splits the packets into multiple

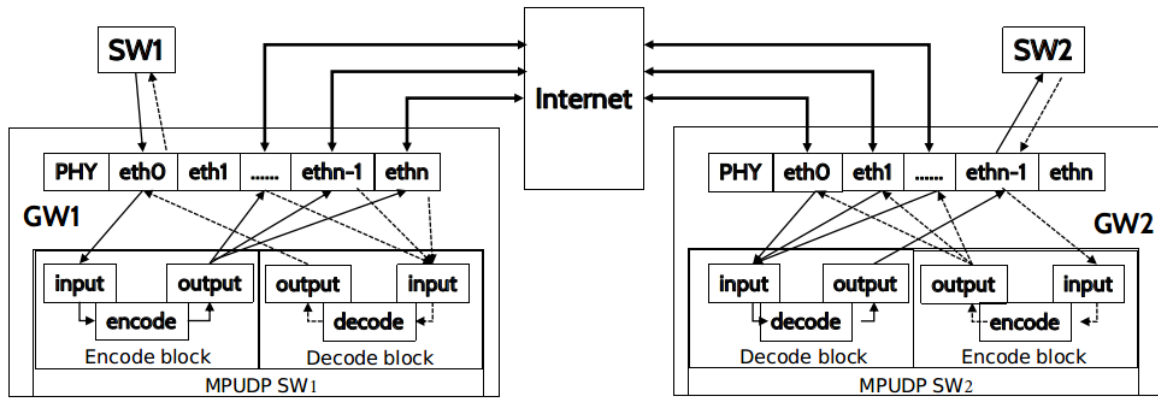


Figure 3.10: Data sending and receiving example for MPUDP software.

UDP subflows. Each subflow is built on a different path. On the network point of view, each subflow appears as an independent, well-formed UDP flow. At the receiver side, the packets from different subflows are combined and decoded at the RaptorQ code environment, and then RaptorQ code environment recreates the RTP session. That means, the VoIP application receives an original RTP session from the RaptorQ code environment. The architecture is shown as follow in Figure3.11, and the protocol stack is picked in Figure3.12.

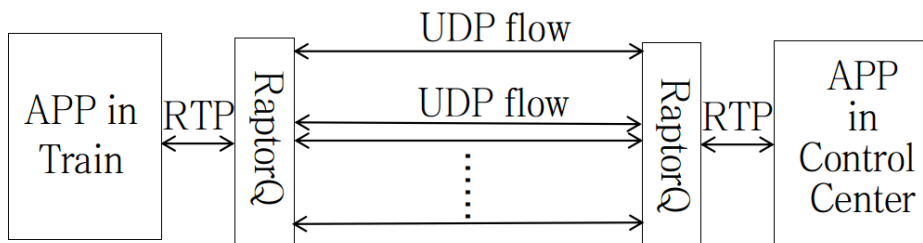


Figure 3.11: Architecture for MPUDP solution with of RaptorQ scheduler

Application	
RTP/RTCP	
MP-UDP software	
UDP	UDP
IP	IP

Figure 3.12: Protocol stack for MPUDP solution with of RaptorQ scheduler

3.3.3 Advantages and disadvantages for MPUDP solution

The aim for MPUDP solution is to support the real time RTP session over multiple UDP paths and this solution can automatically move the traffic from congestion/broken paths to the active paths and improve the total throughput. The decoder is able to rebuild the source block from any set of encoding symbols equal to or slightly more than the number of basic symbols. For this reason, RaptorQ protocol can well solve the out of order problem for the multipath UDP solution. The MPUDP solution add an extra environment between RTP/RTCP and UDP protocols, and from the VoIP application point of view, the session stream is delivered over a single RTP flow. That means, using MPUDP environment does not request the modification of VoIP applications. During a communication session, all UDP subflows work as independent normal UDP flows. In other words, the breaking of current path will not influence the other paths and the switching on/off the current interface will not influence the other interfaces, too. Moreover, since the MP-UDP software is on the application level, it can be used in different operating systems with minor modifications.

There are also some significant defects. The most difficult problem is the implementation. Since we are the first one who propose this solution, there is no implementation on this environment at all. Thus, we need to develop whole software by ourselves. Second, in order to enlarge the tolerance of packet loss, RaptorQ encoder need to add the redundancy overhead symbols, this will increase the network throughput overhead. Third, the decoder can be execute only when there are enough encoding symbols received, and this may need an addition requirement on buffer size and need to face a delay problem as well.

3.3.4 Overview and related works of RaptorQ protocol

Raptor code is a kind of fountain codes, and it is based on the forward error correction (FEC) technology at the application layer. RaptorQ, which comes from Raptor coder, is a new generation fountain code. It aims to minimum the redundant FEC information during the en/decode procedure, and achieve a better performance on the increased encoding and decoding complexity situation [72].

In [73], the authors introduced the differences between RaptorQ code and Raptor code, and they portrayed the advantages for RaptorQ code as well. Five important differences are highlighted in their paper. First, the algebra in RaptorQ is over GF(256) (Galois Field) as compared to GF(2) for Raptor codes. Second, the first encoding step in RaptorQ is padding source block with zeros. Third, RaptorQ use enhanced two-step pre-coder to encode. Fourth, a superior algorithm for LT encoding is used in the final step of RaptorQ. Finally, RaptorQ supports a wider range of source symbols and encoding symbols. Based on all these differences, RaptorQ has more coding capacity, efficiency and flexibility than Raptor codes.

Some other researchers have evaluated the performances of the RaptorQ code, and a part of them have compared RaptorQ with Raptor code as well. C. Bouras et al.[74] assessed the RaptorQ code over 3GPP multicast services. According to their experiments results, in view of achieving a completed MBMS (Multimedia Broadcast/Multicast Services) coverage, RaptorQ needs an overhead about 12% for 5MHz bandwidth system and half for 10MHz bandwidth system. RaptorQ codes performs better under the shorter source symbol and wider system bandwidth situation. T. Mladenov et al.[75] involved RaptorQ on NIOS II embedded system co-working with inactivation decoding Gaussian elimination (IDGE) algorithm. Their tests serviced for multimedia broadcast and content delivery based on MBMS and Digital Video Broadcasting (DVB-H) standards. A plenty of simulation results showed that comparison between Raptor code and RaptorQ code, RaptorQ has more reliability than Raptor code. For probability of successful decoding of 99.9%, RaptorQ requires one extra symbol, while Raptor code needs 16 extra symbols. But the decoding/encoding time for Raptor code is significantly shorter than RaptorQ code. Here we defined that K as block size and T as symbol size. When $K = 32, 256$ and 1024 , Raptor code with $T=128$ is 10.74, 13.97 and 14.02 times (respectively) faster than RaptorQ code in inverting the pre-code matrix and decoding the encoded symbol vector. Comparing the RaptorQ code with itself, the smaller symbol size can en/decode much more faster than the bigger symbol size. For example, with $K=32, 256$ and 1024 , the RaptorQ IDGE algorithm with $T=128$ is 2.92, 3.32 and 3.29 times faster than $T=512$, respectively. Always with $K=32, 256$ and 1024 , the similar results were obtained on the case of comparison between $T=128$ and

$T=1024$ are 6.03, 6.66 and 6.36. What's more, RaptorQ consumes more energy than Raptor code, especially in the large block and symbol size occasion. For this reason, in case of the real-time application scenarios, it is better to have some strong processors to support RaptorQ codes.

RaptorQ has been tested on many other different applications. Linjia Hu et al. [76] used RaptorQ code on both graphics processing units (GPUs) and CPU. They got the similar conclusion as we mentioned before. The en/decode time highly relates on the block size and symbol size. In their test bed, the GPU (GPU-PACKED-GE) works better than the CPU (CPU-SPARSE-IDGE). Authors in [77] introduced an efficient application of AL-FEC scheme, and they deployed this policy over mobile multicast standards based on RaptorQ code. Then, in [78–80], they further exploited their AL-FEC scheme to online algorithm with the help of RaptorQ code. A. Andó and the other researchers, who are from University of Palermo, have worked on the RaptorQ code since 2012. They have involved RaptorQ code in the Free Space Optics (FSO) link quality enhancement strategies [81–86]. Their experiments demonstrated that, all rateless codes (e.g. Luby Transform codes, Raptor and RaptorQ codes) can improve the link quality for FSO, especially the RaptorQ code. The employment of RaptorQ code can strongly decrease the packet error rate, and even achieve the error-less status for receiving data with a properly setting of encoding/decoding parameters. The authors in [87] and [88] used RaptorQ code for the real-time transmission scenarios. In [87] RaptorQ code was used for supporting a channel-adaptive transmission strategy. They proved that RaptorQ code can enhance the system transmission efficiency and reliability. Authors of [88] used RaptorQ code in real-time MPEG Media Transport (MMT) transmission. And RaptorQ code can recover the packet loss for unreliable UDP link. Authors in [89] and [90] embedded RaptorQ together with 802.11 WLANs protocol, and they demonstrated that RaptorQ enabled carousels (compared to standard carousels) can significantly reduce the average response time and increase the percentage of users' satisfactions in the multicast network.

From all above, we can conclude that, RaptorQ code is an FEC code, and it can be considered as the next generation of Raptor code. Comparing with Raptor code, RaptorQ code has higher capacity, efficiency, flexibility and reliability on en/decode.

But the only defect is that with the same block size and symbol size, RaptorQ code needs more en/decoding time and stronger processor than Raptor code, in particular in the big block and symbol size occasion. Comparing with RaptorQ code itself, the en/decoding for smaller symbol size (such as $T=128$) is much faster than the bigger symbol size (such as $T = 1024$). The RaptorQ code has been used in many different scenarios, and on all those scenarios RaptorQ code can improve the reliability and capacity for the link or transmission qualities.

3.4 MP-UDP software architecture design

This MP-UDP software is designed as process-oriented software. The RaptorQ en/decode parts are written in C language, and the other parts are written in Python. It should be run based on Linux system. MP-UDP software intends to create a multipath UDP environment for media software (eg.VoIP).

The transmission is over UDP network protocol, which can well meet the strict delay requirement from the real-time media software. The interfaces configurations for this MP-UDP software can be divided into two different categories: sending interface and receiving interface. Sending interface refers to the interface which sends data to the network. Sending interface sends out those UDP data packets which come from encode block. Receiving interface refers to the interface which receives UDP data from the network. Those received UDP packets will be passed to the decode block.

MP-UDP software has transmission paths management methods and congestion control mechanism, which can increase the capacity, flexibility, stability and reliability for the data transmission.

MP-UDP software includes 8 parts: flow acquisition, flow delivery, encoding, decoding, congestion control, path scheduling, MP data splitter, and logging. The components structure is shown as follow in Figure3.13.

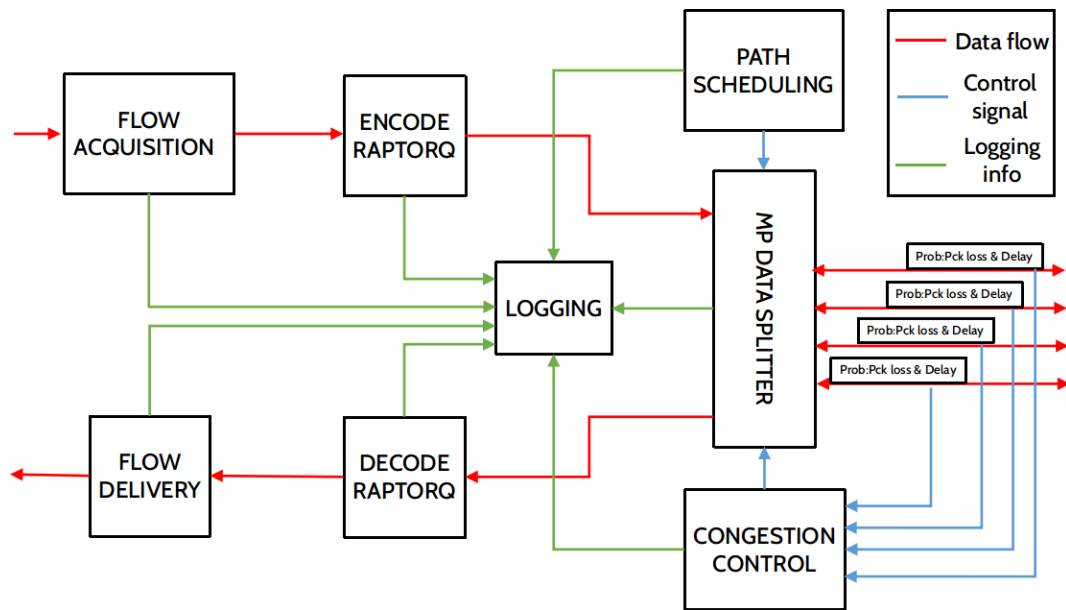


Figure 3.13: MPUDP system components

3.4.1 Flow Acquisition

Component Flow acquisition is the outside data accessing interface for MP-UDP software. In the other words, it can be considered as the media stream data input interface between the user software (media software) and MP-UDP software. This part should continuously receive data from the connected user software and it should pass the data streaming to the encode module with the predefined data format, block size and symbol size.

For RaptorQ scheduler, the Flow acquisition module should be able to get or receive the useful UDP packets form the income flow. It should be able to set/change the buffer size, and of course pass those received data to Encode RaptorQ module. The coming data with dynamic data rate will be put into a buffer. When buffer is full or the received data reach a certain block size, Flow acquisition module should pass this data to the Encode RaptorQ module. The size of the Flow acquisition buffer is decided by the waiting queue of income voice data and the output data rate should be more stable than the input data rate.

For other schedulers, flow acquisition packages the packets with additional information, such as packet IP, time stamps.

3.4.2 Encode RaptorQ

Encode RaptorQ is only used for RaptorQ scheduler. This module is the data processing part for the input data from media software.

The Encode RaptorQ module is one of the most important part for MP-UDP software. The RaptorQ encoding method helps the packets to solve the packet loss and out of order problems in the receiver side. Because, in the multiple link case, the UDP packets will be delivered over different UDP links and each of the UDP link may have different delay and the packets may disorderly arrive at the receive side, but RaptorQ decoding method does not crucial rely on the packets order, it can decode the block of data once it gets a certain subset of the encode packets.

The encode RaptorQ module should be able to encode the input block and encoding velocity should fast enough to meet the requirement for real-time communication. The overhead rate of RaptorQ encoding can be changed according to the channels capacities.

3.4.3 MP Data Splitter

MP data splitter is the inside and outside interface for the encoded UDP packets between two gateways.

The purpose for MP data splitter is to distribute the packets in the proper subflow according to the scheduling strategy and it collects the data from the contrary gateway as well.

For sending mode, data from encode module or flow acquisition module should be able to deliver over multiple paths. The MP data splitter takes the role to distribute data packets in different links. MP data splitter will choose a link path for each packet according to the path scheduling.

For receiving mode, with RaptorQ scheduler, MP data splitter collects the RaptorQ encode UDP packets form the counterpart gateway, and pass those data to the decode RaptorQ module. With other schedulers, MP date slitter should pass the packaged packets to flow delivery module directly.

3.4.4 Decode RaptorQ

Decode RaptorQ is only involved in RaptorQ scheduler. This module is the data processing part for the data which will be pass to media software.

The Decode RaptorQ module takes charge of the decoding work for the received packets from the contrary gateway. Once the received packets reach a certain subset of the original encode set for each block, the RaptorQ decode processing will begin and Decode RaptorQ part also connects with the Flow delivery, in the other word, decode module prepares the UDP stream data for Flow delivery module.

The decode RaptorQ component should be able to decode the data block from a certain number of the receiving UDP packets from the remote gateway. The decoding should not depend on the order of the received UDP packets. The decoding processing velocity should meet the requirement from real-time communication. In order to continuously receive packet from MP data splitter, once the decoding finishes for one block, the data, which stay in the decoding buffer, should be clear out. The decoded data should be send to Flow delivery and wait for the delivery to media software.

3.4.5 Flow delivery

Component Flow delivery is the MP-UDP software data export interface to media software.

The Flow delivery module is the data export interface between user software and MP-UDP software. This module should continuously deliver the date to the terminal software with correct packets order and stable data rate.

For RaptorQ scheduler, the Flow delivery module should be able to receive the decoded data from Decode RaptorQ module and save in its buffer temporarily in order to make sure the correct order for the block packets. This module should also be able to send out the data streaming to media software with a stable data rate.

For other scheduler, the flow delivery module should be able to unpackage the receiving data from MP data splitter and pass the pure and reordered stream date to media software.

3.4.6 Congestion Control

This module servers only for the balance scheduler.

Congestion control is designed to make the best decision for distributing the received packets which come from the streaming. The decision should base on the link delay status of each UDP link.

The congestion control part contains the congestion control algorithm and carries out the distribution plan based on the link status and congestion algorithm. Congestion control module can help MP-UDP software to carry out an optimization solution to distribute the packets on multiple UDP links. In order to get the link capacity for each possible link, we send probes over every possible path in a given time interval. The time interval can be set in the configuration file. All these link condition information will be passed to the congestion control module as the features, which will be used in the congestion algorithm.

3.4.7 Path Scheduling

Users can set/change the path scheduling with this module. This module aims to give the scheduling to all possible UDP links for the encoded packets distribution. Several path-scheduling functions are predefined in this module. The user can set/change the scheduling strategies through this module.

3.4.8 Logging

Logging module is the recording component for the MP-UDP software. This part aims to record all activities occurring during the usage of MP-UDP software. Logging module takes the role of recording all activities form every module of the software. Every module output their real-time activity information to the logging module. And the logging module records all activities information in the logging file and generates a logging document if need.

3.5 MP-UDP software behaviors evaluation with RaptorQ scheduler

In order to know if our MP-UDP software can work well or not, first we should have a basic knowledge about the requirements of VoIP service. For this reason, we did some researches on the works relating to the quality assessment of VoIP.

3.5.1 Related works of VoIP quality assessment

The voice can be IP based and seamlessly delivered over Internet with an integrating of different kinds of networks. Voice quality is ultimately adjudged by the listener and thus, speech quality is inherently perceptual or subjective in nature [91]. The Mean Opinion Score (MOS) test provides a widely accepted measure for subjective speech quality [92]. A numeric scale ranging from 1 (unacceptable) to 5 (excellent) has been set in the MOS test and the final result is considered as the average of accepted subjective score for the testing audio piece.

According to [93], the MOS score can be convert to the R value and there are a big amount of factors which can lead to an impairment in the voice quality such as packet loss, delay and jitter. Recommendation [94] depicts the different delays for VoIP based on diverse codec types. It also draws up the relation between the user satisfied level and the mouth-to-ear-delay time. The curve is shown in Figure3.14.

From Figure3.14 we can see that when the mouth to ear delay is less than 200ms, most of the users are very satisfied. Delay from 200ms to 280ms, users are still satisfied. In the interval of 280ms to 380ms, a part of the users are not satisfied, and from 380ms to 550ms, many users are dissatisfied. When the delay is larger than 550ms. Nearly all users are dissatisfied. Based on this result, we can summarize that for our test, the maximum total delay should be less than 550ms.

In fact, many other researchers have done a great number of works on the VoIP quality assessment. Amit Chhabra, et al [95] gave the assessment of VoIP E-model over 802.11 wireless mesh network. They mentioned the different attributes of corresponding codec, and highlight that the R value can be diminished by many artifacts, which mainly

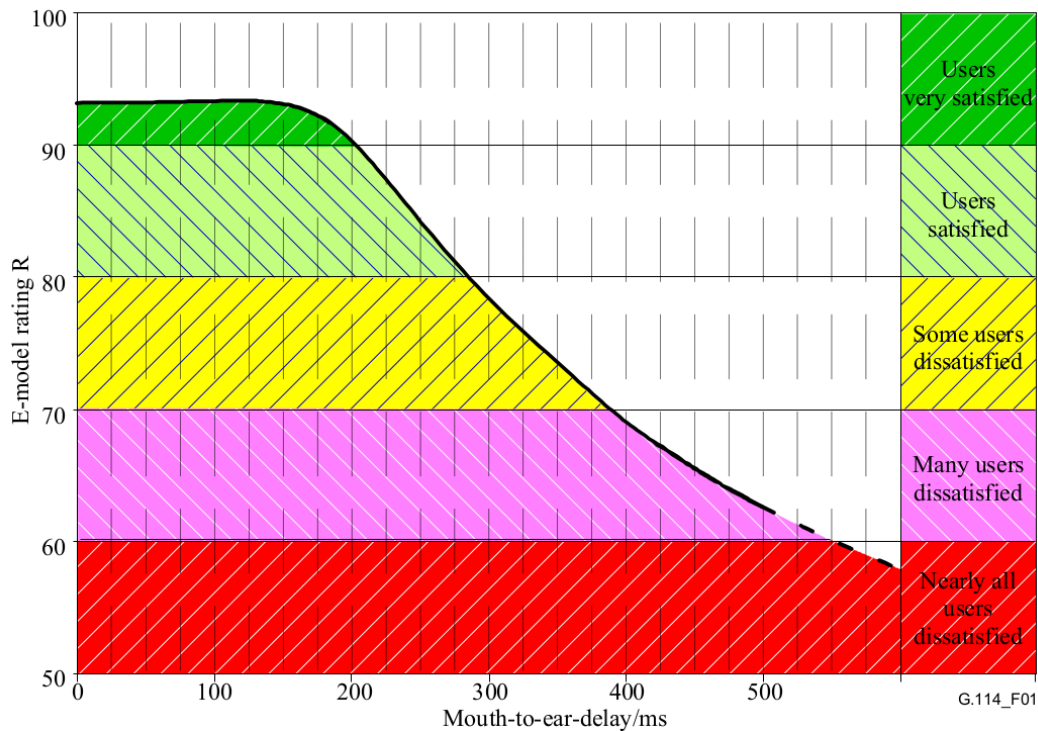


Figure 3.14: R value versus Mouth to ear delay

include the codec type, packet loss, jitter and delay. But some times the R value can be compensated with an Ad factor, that is advantage parameter that a person in an environment can tolerate. The value of Ad is more in satellite networks than in others. In addition, the Ad factor usually lies in the rang of 0 to 20. Y. Han, et al, in [96] have done some researches on the accuracy analysis of PESQ [97] (Perceptual Evaluation of Speech Quality) and E-Model [98], which are two main objective quality assessment functions. They tested the different impairments for the same packet loss rate based on different languages. In addition, they did quality evaluation on different languages and figured out that this two methods can well indication the quality in a certain period of the time during the call. In [99] H. Assem, et al, proposed an APU (Acceptable, Poor, Unacceptable) approach. They separate the MOS value into three categories(i.e. Acceptable, Poor, Unacceptable groups) according to different codec types, shown in Table3.1.

And they also defined delay into three intervals (i.e. 0-150ms, 150ms-400ms, and 400ms-infinity) respecting to the same APU categories, respectively. Meanwhile, they illustrated the MOS trends with different packet loss rate, codec type and burst ratio.

Codec	Acceptable	Poor	Unacceptable
G711	3.28-4.41	2.14-3.28	1-2.14
G723 5.3k	2.86-3.79	1.93-2.86	1-1.93
G723 6.3k	2.96-3.95	1.98-2.96	1-1.98
G726 16k	2.16-2.74	1.58-2.16	1-1.58
G726 32k	3.04-4.07	2.02-3.04	1-2.02
G726 40k	3.16-4.24	2.08-3.16	1-2.08
G729	3.26-4.13	2.13-3.26	1-2.13
GSM FR	2.64-3.46	1.82-2.64	1-1.82
G729 A	3.06-4.10	2.03-3.06	1-2.03

Table 3.1: APU model for the MOS of some codecs [99]

In [100], N. Khitmoh, et al, did a subjective test for VoIP quality, and the test based on Thai language. They set three kinds of delays (0ms, 400ms, and 800ms) and different packet loss rate (from 0% to 10%). And from their results we can find even with 10% of packet loss rate and 0.8 second delay, the MOS value can arrive at 3.29 (with standard deviation 0.53). If we compare this MOS to APU method, we can conclude that end users can still accept the voice quality with 10% of packet loss rate and 0.8 second delay. T. Daengsi, et al, in [101] introduced an enhanced objective method for VoIP quality evaluation based on E-model for G.729. And they verified their method with MOS values on Thai language as well. The experiments were taken under the delay rang 0-400ms and the packet loss rang 0-6%. L. Angrisani, et al, in [102] depicted the influence of IPDV (Internet Protocol Delay Variation) to the MOS. IPDV can be simply considered as jitter. And from their results we can conclude that with enlarging the jitter buffer, the MOS can be improved somehow. And the IPDV method can well reflect the quality level. The authors in [103] they appraised the impairment of quality with three factors Packet loss(0-30%, step 3%), Delay (0ms-300ms, step 25ms) and jitter(0ms-50ms, step 5ms). And from their result we find that the jitter plot has the fastest slop downgrade, while the delay has the slowest one and the MOS decline of packet loss stays in the middle. Thus, we can conclude that with the amount of time, the influence of jitter is stronger than delay. Besides, we can find from their result, for AMR [104] (Adaptive Multi-Rate codec) codec, the higher bit rate gets the better quality.

3.5.2 Testbed hardware and software requirements

The RaptorQ library performances evaluation experiments are implemented at software level. The test bed is built in a DELL inspiron 13 7000 series computer, with Intel Core i7-6500U CPU 2.50GHz \times 4; 7.5GB memory and Linux Ubuntu 16.04 operation system.

The Raptorq test is written in C language. In order to run the C codes, the computer must be installed with GCC compiler. In addition, to run the test, the RaptorQ library [105], lz4 library, and pthread library are requested, too. In our case the codes are edited, compiled and run under the integrated development environment (IDE) Eclipse.

The MP-UDP software is written in Python2.7. To run this code, you need make sure the computer has been installed the python2.7 and all the libraries requested in the code. Especially, the libraptorq library in python2.7.

To test MP-UDP software, we are emulate a real end to end communication environment, for this reason, the communication will be taken place in two virtual machines with three network interfaces on each, one for time synchronize, two for deliver traffic.

To make the time synchronize we need NTP to set one computer as time server and the other as client. And always synchronize this two end terminals during all tests.

To generate the packet, we need two kinds of packet generator, one is iperf[106] the other is D-ITG[63].

3.5.3 Path adding and dropping test for MP-UDP software

In the path adding and dropping test scenario, we assume that the train travels along the track, which is represents with arrow line in Figure3.15 .

The entire trip lasts 100s, the train continuously sends out the packets to control center, with a constant bitrate 100kbit/s, while the RaptorQ overhead rate is 150%. The satellite network covers all travel distance, whilst PLMNs can be accessed from 0s to 40s and from 80s to 100s. From 0s to 40s, the communication is support by two kinds of networks. Then, from 40s to 80s, all packets sent over PLMN are lost. Later, from 80s to 100s, PLMN comes up again and is able to deliver packets. This means that there is a radio hole in the middle, where is not covered by any PLMN network.

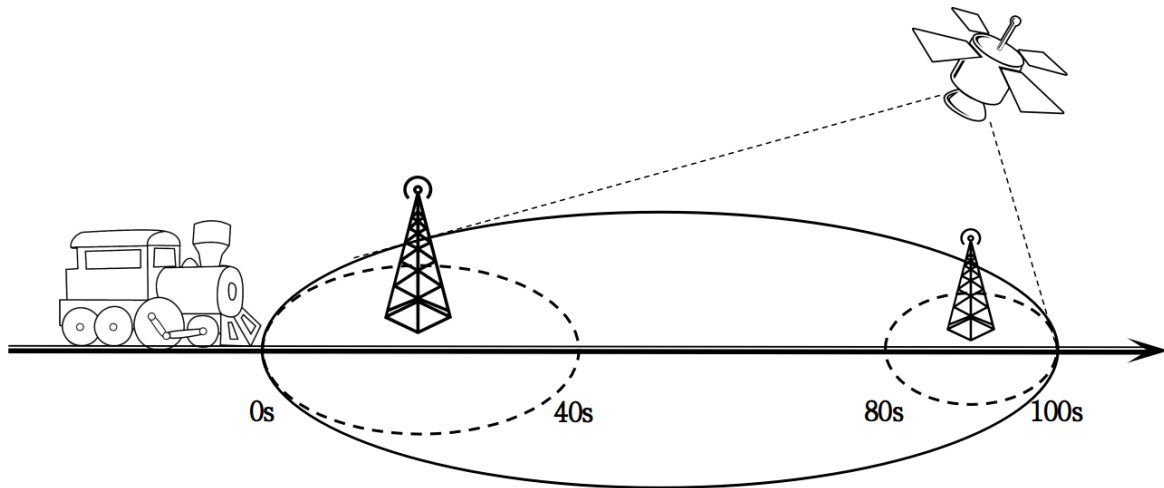


Figure 3.15: Test scenario

Parameters	Satellite	PLMN
Delay	500ms	150ms
Jitter	50ms	15ms
Loss	15%	15%
Rate	512kbit	2048kbit

Table 3.2: Network parameters

Although, during the radio hole, only satellite takes charge of the data delivery, from the media software point of view, there is no influence on sending and receiving data.

The satellite and PLMN network parameters are acquired from the real environment. In our test, the PLMN is set with 150ms delay, 15ms jitter, and bitrate 2048 kbit/s. The satellite network has a higher delay and jitter, that are 500ms delay and 50ms jitter. Satellite has bitrate 512kbit/s, which is less than PLMN. Both networks have a same packet loss rate 15%. All parameters are depicted in Table3.2.

3.5.3.1 obtained results

In this test, the total sending time is 100s while the total receiving time is 100.2s. The sending bitrate on iperf client side is 100kbit/s, meanwhile, on iperf server side, the receiving bitrate stays the same. Although, for both satellite and PLMN, the packet loss rate are 15%, thanks for RaptorQ overhead algorithm, there is neither packet loss nor out of order packet on iperf server side.

The traffic over satellite and PLMN depicted in Figure3.16 are monitored by Wire-

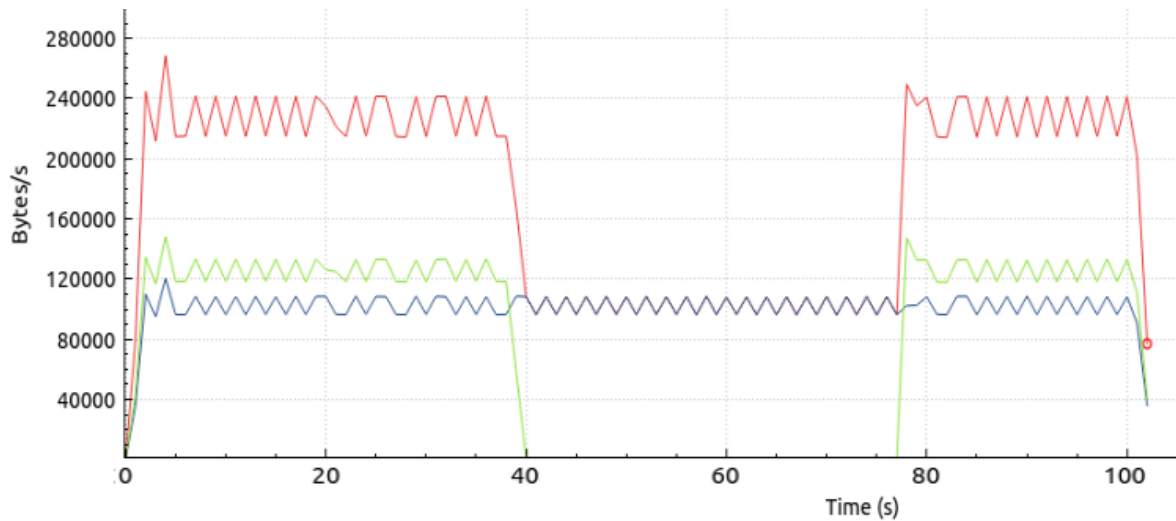


Figure 3.16: Test results.

shark installed in router virtual machine. The red line represents the total throughput, the green line describes the throughput over PLMN network, and the blue line is the traffic over satellite. At beginning, packets delivered over both networks can be received, then, the PLMN link is broken. It is easy to know that during the radio hole, the traffic can pass only over satellite network. Finally, when the PLMN comes up, the link can recover automatically.

The result shows that, the MP-UDP software can use more than one UDP links to transmit the packets, and the paths can be added and dropped at any time during the transmission time. If there is enough bandwidth to sustain the transmission, broken of any UDP link will not give big influence on the overall communication.

3.5.4 Performance assessment for RaptorQ en/decode function

We propose a solution for VoIP service based on multiple UDP transmission links. From the others previous works we can know that VoIP service is sensitive with the time delay, since it is a kind of real time voice service. En/decode time may take a big portion in the delay time budget. For this reason, we need some further researches on the en/decode time with diverse relevance of symbol/subsymbol size and file dimension on account of our particular using scenario.

Some of the researchers have done a few tests based on the RaptorQ code protocol,

and from their results we can summarize that, RaptorQ code has higher capacity, efficiency, flexibility and reliability on en/decode than the old Raptor code. But with the same block size and symbol size, RaptorQ code needs more decoding time and stronger processor than Raptor code, in particular in the big block and symbol size occasion, and the symbol size influence the en/decode speed.

Besides symbol/subsymbol sizes, the overhead also has a big influence on the channel transmission capacity, the success decode rate, and the en/decode time. It is one of the most important features in the RaptorQ en/decode Receiving as well. In this chapter, we will do some series of tests, and all of them will focus on the total encode and decode time.

3.5.4.1 En/decode time consume with different symbol and subsymbol sizes for various file dimensions with 100% overhead

From [107] we know that without packet loss, to have 99.9% of successful decoded RaptorQ need one extra symbol. However, in our case, one overhead is not enough. For this reason, in this type of test, we use 100% overhead to guarantee the successful en/decode. Here, we want to find out some correlations among different symbol sizes, different subsymbol sizes, and different file dimensions on en/decode time.

1. Test parameters setting and test design

ITU-T-G.114 has displayed the relationship between the mouth-to-ear delay and the R value, which has been shown in Figure 3.14. As we mentioned before, after delay of 550 ms, nearly every user is not satisfied with the quality. In other words, we can say, if we follow a very strict principle, our delay budget will be total 550ms. But from [100] we learned that even with a 800ms delay and 10% of packet loss, the MOS is 3.29 (with standard deviation 0.53), and converting to R value is around 60. Based on this, we can consider that under the more loose condition, our delay budget can go up to 800ms. In fact, in the reality we can get an even larger acceptable delay such as 1000ms, however, here we use 800ms [94] as our delay budgets.

The delay budget should include the VoIP codec delay (encoding and decoding), the RaptorQ en/decoding delay, the burst delay, and transmission delay, shown in Figure 3.17. So, the delay budget for RaptorQ en/decode can be presented as equation 3.1:

$$D_{RaptorQ} = 400ms - D_{VoIPEncode} - D_{Burst} - D_{TransmissionTime} - D_{VoIPDecode} \quad (3.1)$$

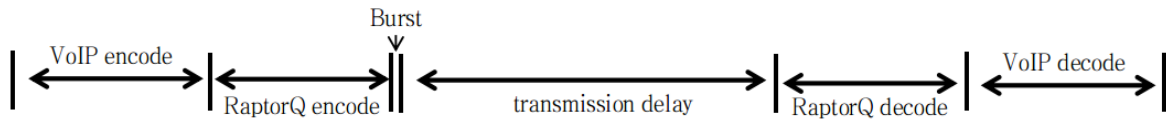


Figure 3.17: Delay budget

In particular, on most of our occasions, we use PLMNs (Public Land Mobile Networks) as our transmission systems, and the typical delay for this system is from 80ms-110ms. Then, the variance of IP based VoIP codec delay is from 0.25ms to 97.5ms (one frame for per packet). We pick up the maximum values for both of these two issues (for easy calculation, we consider the codec delay as 100ms). Since comparing with the packet size, the burst speed is fast enough (1kbyte packet need about 2ms), we ignore the burst time consuming here. In sum, our RaptorQ en/decode delay budget equals to 490ms (see equation 3.2).

$$D_{RaptorQ} = 800ms - 100ms - 110ms - 100ms = 490ms \quad (3.2)$$

In this test, we would like to test en/decode time for fixed sized files with symbol size 512 byte and 256 byte. We use the fixed overhead ratio, i.e. 100%. As an example, when the file is 1024 byte, the overhead is 1024 byte as well. Later, we also have some tests to carry out the relationship between overhead and toleration of the packet loss rate (we consider only on the packet loss rate less than 20%).

The tests will be executed on the software level. The software includes the configuration file analysis, read in component, encode part, transmission simulation,

decode part and the write out component. The procedure time is considered from the beginning of encode part to the ending of decode part. The structure has been shown in Figure3.18. The file sizes are selected as 1kbyte, 2kbyte and 5kbyte, while the symbol size are 512byte and 256 byte. Subsymbol sizes are select as 64byte, 128byte, 256byte for symbol size 512 byte, while 32byte, 64byte and 128byte for symbol size 256byte.

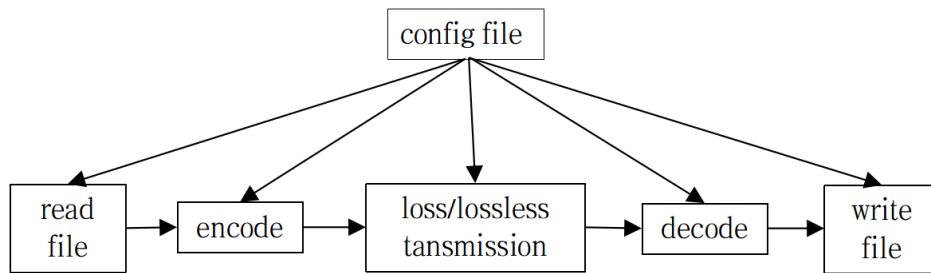


Figure 3.18: Set up for test 1

2. Results and analysis for various file dimensions with 100% overhead

The test results with symbol size 512 byte are shown in Table3.3 and with symbol size 256 byte are shown in Table3.4.

symbol size	subsymbol size	overhead	file_size(kByte)	runing time1 (second)	runing time2 (second)	runing time3 (second)	average (ms)
512	64	2	1	0.02402	0.02361	0.02361	23.75
	128			0.02361	0.02686	0.02918	26.55
	256			0.02918	0.02876	0.02388	27.28
512	64	4	2	0.04360	0.04288	0.04330	43.26
	128			0.04280	0.04252	0.04473	43.35
	256			0.04394	0.04381	0.04142	43.05
512	64	10	5	0.10078	0.10487	0.09500	100.22
	128			0.10165	0.09474	0.10078	99.06
	256			0.09890	0.10048	0.10107	100.15

Table 3.3: Results of test1, with symbol size 512 byte

From Table3.3 and Table3.4 we can find the follow points:

- a) The en/decode time ascends with the increasing file size.
- b) For file size 1kbyte, there is no significant different between symbol size 512byte and 256byte, and the en/decode time is around 26ms.

symbol size	subsymbol size	overhead	file_size(kByte)	runing time1 (second)	runing time2 (second)	runing time3 (second)	average (ms)
256	32	4	1	0.02539	0.02543	0.02629	25.70
	64			0.02692	0.02631	0.02663	26.62
	128			0.02621	0.02662	0.02543	26.09
256	32	8	2	0.04940	0.04829	0.04251	46.73
	64			0.04878	0.04947	0.04277	47.01
	128			0.04780	0.04343	0.04355	44.93
256	32	20	5	0.19083	0.19171	0.19201	191.51
	64			0.19091	0.19016	0.19078	190.62
	128			0.12450	0.11539	0.12483	121.57

Table 3.4: Results of test1, with symbol size 256 byte

- c) For file size 2kbyte, symbol size 256byte is slower than 512byte, but still comparable with 512byte. The en/decode time for 512byte is around 43ms and 46ms for 256byte.
- d) For file size 5kbyte, symbol size 256byte is much slower than 512byte, but both of them are too slow to meet our delay budget. The delay for 512byte is about 100ms and for 256byte, delay time is strongly depending on the subsymbol size.
- e) When the file size is less than 2kbyte, both the 512byte and 256byte symbol size can fulfill with our delay request. But 5kbyte file is too slow to use in our case.
- f) At most cases, for same file size and sybmol size, the execution time for different subsymbol size is similar with each other, exception the 5kbyte file for 256byte symbol size.

3.5.4.2 Overhead requirement for different Packet Loss Rate (PLR<20%) with 100% successful en/decode

RaptorQ decoding is regardless the symbol's order, and on the other side, adding more overhead can increase the toleration of packet loss rate. At previous, we did not consider the packet loss and here we will focus on the indispensable overhead number, which can help RaptorQ to successfully en/decode against a certain packet lose rate.

1. Test parameters setting and test design

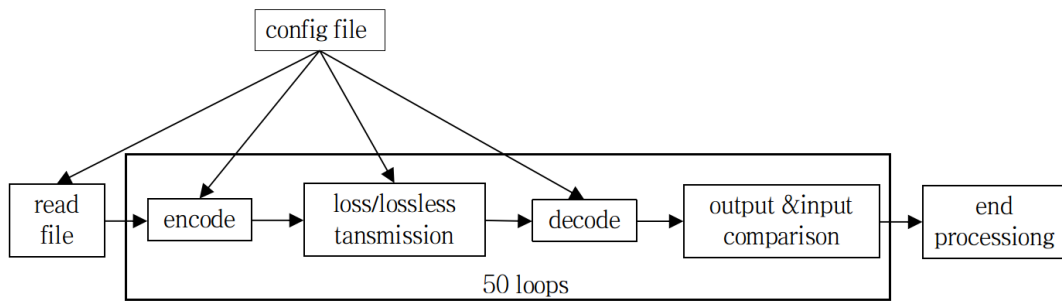


Figure 3.19: Set up for test 2

We test the same file sizes, symbol and subsymbol sizes as last section, except 5kbyte file with 256byte symbol size, because under this condition, the en/decode delay is too large to achieve our request. However, for this group of tests, each en/decode execution will be circulated for 50 times. And the indispensable overhead number means it should ensure 100% of successful en/decode for all 50 times of operations. The en/decode time is considered as the average receiving time of 50 operations. The packet loss rate are selected as 1%, 3%, 5%, 10%, 15% and 20%.

The tests will be executed on the software level as well. And test program include the configuration file analysis, read in component, loop start, encode part, transmission simulation, decode part, the input & output comparison part and loop finish. The procedure time is considered as the average of each running, and the en/decode running time is from the beginning of encode part to the ending of decode part. The structure has been shown in Figure3.19.

2. Results and analysis for different Packet Loss Rate (PLR<20%) with 100% successful en/decode

The test results for overhead versus packet loss rate are shown in Table3.5 to Table3.12. From Table3.5 to Table3.12 and Figure3.20 and Figure3.21 we can find the follow points:

- a) From Table3.5 to Table3.10, we know that with same symbol size, file size and packet loss rate, different subsymbol sizes have the same indispensable overhead number and similar receiving time. So we can say that we can use

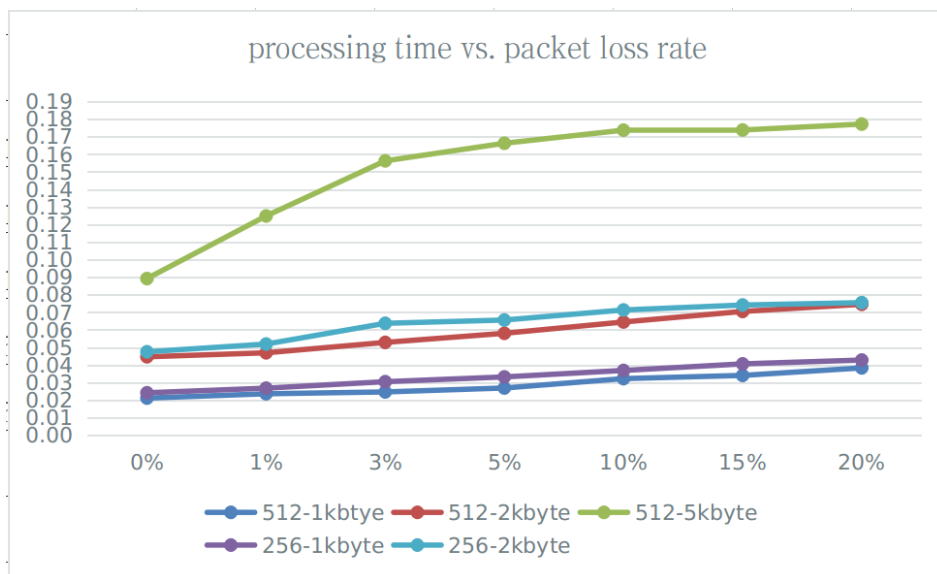


Figure 3.20: Receiving time vs. packet loss rate

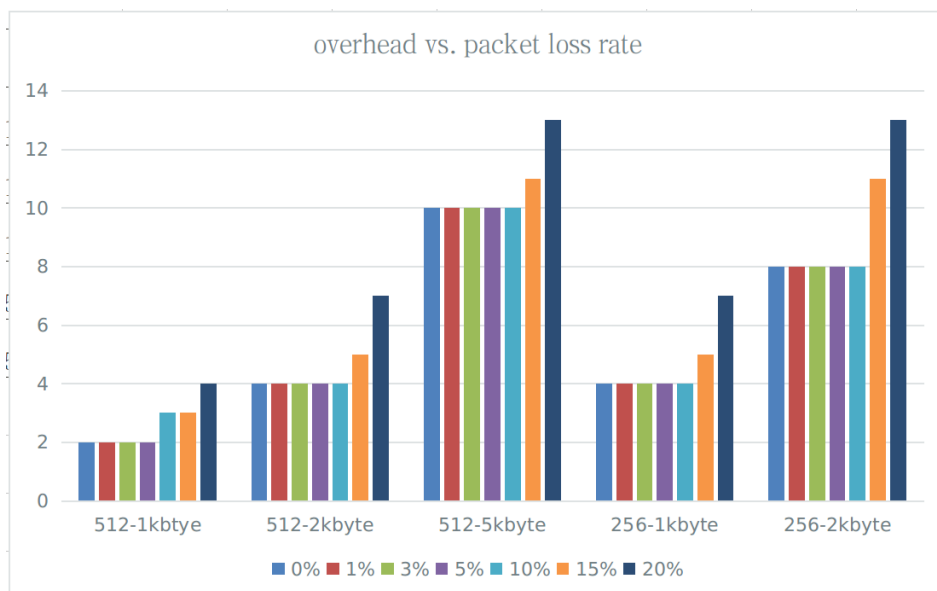


Figure 3.21: overhead vs. Packet loss rate

Attributes			1%pkloss50runs		
symbol size	subsymbol size	file_size(kByte)	drop_runtimes	overhead	time(s)
512	64	1	6	2	0.024251
	128		6	2	0.023471
	256		6	2	0.023081
512	64	2	13	4	0.047017
	128		11	4	0.046923
	256		11	4	0.046842
512	64	5	22	10	0.124848
	128		22	10	0.124676
	256		22	10	0.12498
256	32	1	11	4	0.027043
	64		11	4	0.026663
	128		11	4	0.026668
256	32	2	20	8	0.052069
	64		20	8	0.051771
	128		20	8	0.051859

Table 3.5: Packet loss rate 1%

Attributes			3%pkloss50runs		
symbol size	subsymbol size	file_size(kByte)	drop_runtimes	overhead	time(s)
512	64	1	9	2	0.025026
	128		9	2	0.024371
	256		8	2	0.024521
512	64	2	19	4	0.052351
	128		20	4	0.053256
	256		20	4	0.052932
512	64	5	41	10	0.155938
	128		42	10	0.156447
	256		42	10	0.156278
256	32	1	20	4	0.030322
	64		20	4	0.030915
	128		20	4	0.030195
256	32	2	38	8	0.064748
	64		38	8	0.063386
	128		38	8	0.063019

Table 3.6: Packet loss rate 3%

any subsymbol size as we want, and this will not have big influence on the en/decode results.

- b) Since subsymbol size does not influence a lot the receiving time, we use the average value for all three kinds of receiving times as the en/decode time of a certain pair of symbol size and file size. Then we find that, for 512-1kbyte (i.e. 512byte symbol size and 1kbyte file size) and 256-1kbyte, 512-2kbyte and 256-2kbyte the receiving time are very similar for each pair. But generally, 512 symbol size is a bit faster than 256 symbol size. See

Attributes			5%pkloss50runs		
symbol size	subsymbol size	file_size(kByte)	drop_runtimes	overhead	time(s)
512	64	1	15	2	0.027469
	128		15	2	0.026645
	256		15	2	0.026542
512	64	2	27	4	0.058025
	128		23	4	0.058027
	256		27	4	0.05803
512	64	5	48	10	0.166372
	128		48	10	0.165933
	256		48	10	0.166396
256	32	1	27	4	0.033207
	64		27	4	0.033132
	128		27	4	0.033229
256	32	2	43	8	0.065798
	64		43	8	0.065553
	128		43	8	0.065423

Table 3.7: Packet loss rate 5%

Attributes			10%pkloss50uns		
symbol size	subsymbol size	file_size(kByte)	drop_runtimes	overhead	time(s)
512	64	1	30	3	0.032985
	128		30	3	0.031841
	256		31	3	0.031839
512	64	2	27	4	0.064224
	128		27	4	0.06444
	256		28	4	0.06474
512	64	5	50	10	0.173978
	128		50	10	0.173301
	256		50	10	0.173785
256	32	1	38	4	0.036887
	64		38	4	0.036755
	128		38	4	0.037009
256	32	2	47	8	0.071516
	64		47	8	0.07142
	128		47	8	0.071003

Table 3.8: Packet loss rate 10%

Attributes			15%pkloss50uns		
symbol size	subsymbol size	file_size(kByte)	drop_runtimes	overhead	time(s)
512	64	1	35	3	0.034606
	128		35	3	0.033925
	256		35	3	0.03362
512	64	2	47	5	0.070164
	128		47	5	0.070274
	256		47	5	0.071017
512	64	5	50	11	0.173825
	128		50	11	0.173489
	256		50	11	0.174055
256	32	1	47	5	0.040636
	64		47	5	0.040612
	128		47	5	0.040597
256	32	2	50	11	0.074895
	64		50	11	0.073963
	128		50	11	0.073529

Table 3.9: Packet loss rate 15%

Attributes			20%pkloss		
symbol size	subsymbol size	file_size(kByte)	drop_runtimes	overhead	time(s)
512	64	1	45	4	0.038383
	128		45	4	0.038369
	256		45	4	0.038249
512	64	2	50	7	0.074525
	128		50	7	0.074359
	256		50	7	0.074756
512	64	5	50	13	0.177096
	128		50	13	0.177086
	256		50	13	0.177258
256	32	1	50	7	0.042811
	64		50	7	0.04299
	128		50	7	0.042597
256	32	2	50	13	0.075212
	64		50	13	0.075629
	128		50	13	0.075596

Table 3.10: Packet loss rate 20%

PLR	0%	1%	3%	5%	10%	15%	20%
512-1kbyte	0.02113	0.02360	0.02464	0.02689	0.03222	0.03405	0.03833
512-2kbyte	0.04465	0.04693	0.05285	0.05803	0.06447	0.07049	0.07455
512-5kbyte	0.08920	0.12483	0.15622	0.16623	0.17369	0.17379	0.17715
256-1kbyte	0.02417	0.02679	0.03048	0.03319	0.03688	0.04062	0.04280
256-2kbyte	0.04751	0.05190	0.06372	0.06559	0.07131	0.07413	0.07548

Table 3.11: Receiving time vs. packet loss rate

PLR	0%	1%	3%	5%	10%	15%	20%
512-1kbyte	2	2	2	2	3	3	4
512-2kbyte	4	4	4	4	4	5	7
512-5kbyte	10	10	10	10	10	11	13
256-1kbyte	4	4	4	4	4	5	7
256-2kbyte	8	8	8	8	8	11	13

Table 3.12: overhead vs. Packet loss rate

Figure3.20.

- c) With the increasing of packet loss rate, the en/decode times are increasing. This because the lost packet adds the waiting time and difficulty of decode receiving.
- d) From Figure3.21 we find that 100% overhead can stand the packet loss rate till 5%. Then some of the test occasions need more overhead to ensure the successful decode.
- e) In our test, for same packet loss rate, we need more overhead for 512 byte symbol size than 256 byte symbol size for a same file size(here the file size is always more than or at least equal to 1kbyte). But we should mention that, in the test, the en/decode time for 5kbyte file with 256 byte symbol size is too slow to use.

In brief, we can say that in our case, when the file size is more than 1kbyte, the subsymbol size is not a very important variable. Besides, 512 symbol size performance a litter better on the en/decode receiving speed. 100% of overhead can talent 5% of packet loss rate, and under 5% packet loss condition, the 256 byte symbol size performances a few better than 512 byte, but not a big difference. However, for RaptorQ, every symbol should be sent separated, for this reason, we should also take the IP packet head consuming into condition, and the larger symbol size has the less IP packet head proportion.

Network	Delay (ms)	Jitter (ms)
Cellular(C)	150	15
Satellite(S)	500	50

Table 3.13: The networks delay and jitter parameters, we use C as Cellular network and S as satellite network from now on.

3.5.5 MP-UDP software performance evaluation with RaptorQ scheduler

We will test the all these three kinds of schedulers for our MP-UDP software, they are Basic, Balance and RaptorQ. From [108] we know that, the average call duration is 107.1s. Thus, in our tests, to simulate the voice communication, we set 110s as the our transmission duration. According to [95] and [97], E-model and PESQ are used as two standard VoIP MOS value objective assessment tools. In those tools, the MOS is calculated based on the noises, the delay, the packet loss and the other factors. In our case, we can not change the environment noises and the factors, so we focus our test on the channel delay and the packet loss. Especially, in the RaptorQ scheduler, the packet size will influence the delay as well, so we well test the RaptorQ with different packet size. To do the following tests, we use the same computer, which we have used in 3.5.4. The hardware and software requirements have already been depicted in section 3.5.2.

3.5.5.1 Test design

We will test the RaptorQ scheduler with 2 subflows. These tests will help us to study the performance of RaptorQ with different delay, packet loss rate and overhead rate. RaptorQ scheduler is regardless the packet order, we would like to qualify if it can well against the out-of-order problem.

The tests will over 2 separate networks (2 subflows), which cannot talk to each other. The test will consider two main scenarios, one is the communication over two PLMNs, the other is the communication over one PLMN and one satellite link. Respect to [109], we set the cellular network with delay 150ms, Jitter 15ms, whilst the satellite network with delay 500ms and Jitter 50ms. Parameters are shown in Table3.13. The two main scenarios will be test under multiple packet loss rate, 0%, 5%, and 15%. As we

Test number	Network	Generator	Duration (s)	overhead	codec	Packet loss (%)
1	C / C	D-ITG	110	0	G.711.1	0 / 0
2	C / C	D-ITG	110	1	G.711.1	0 / 0
3	C / C	D-ITG	110	1.5	G.711.1	0 / 0
4	C / C	D-ITG	110	0	G.711.2	0 / 0
5	C / C	D-ITG	110	1	G.711.2	0 / 0
6	C / C	D-ITG	110	1.5	G.711.2	0 / 0
7	C / C	D-ITG	110	0	G.729.2	0 / 0
8	C / C	D-ITG	110	1	G.729.2	0 / 0
9	C / C	D-ITG	110	1.5	G.729.2	0 / 0
10	C / C	D-ITG	110	0	G.729.3	0 / 0
11	C / C	D-ITG	110	1	G.729.3	0 / 0
12	C / C	D-ITG	110	1.5	G.729.3	0 / 0
13	C / C	D-ITG	110	0	G.723.1	0 / 0
14	C / C	D-ITG	110	1	G.723.1	0 / 0
15	C / C	D-ITG	110	1.5	G.723.1	0 / 0
16	C / C	D-ITG	110	0	G.711.1	5 / 5
17	C / C	D-ITG	110	1	G.711.1	5 / 5
18	C / C	D-ITG	110	1.5	G.711.1	5 / 5
19	C / C	D-ITG	110	0	G.711.2	5 / 5
20	C / C	D-ITG	110	1	G.711.2	5 / 5
21	C / C	D-ITG	110	1.5	G.711.2	5 / 5
22	C / C	D-ITG	110	0	G.729.2	5 / 5
23	C / C	D-ITG	110	1	G.729.2	5 / 5
24	C / C	D-ITG	110	1.5	G.729.2	5 / 5
25	C / C	D-ITG	110	0	G.729.3	5 / 5
26	C / C	D-ITG	110	1	G.729.3	5 / 5
27	C / C	D-ITG	110	1.5	G.729.3	5 / 5
28	C / C	D-ITG	110	0	G.723.1	5 / 5
29	C / C	D-ITG	110	1	G.723.1	5 / 5
30	C / C	D-ITG	110	1.5	G.723.1	5 / 5

Table 3.14: Test design (part 1)

Test number	Network	Generator	Duration (s)	overhead	codec	Packet loss (%)
31	C / C	D-ITG	110	0	G.711.1	5 / 15
32	C / C	D-ITG	110	1	G.711.1	5 / 15
33	C / C	D-ITG	110	1.5	G.711.1	5 / 15
34	C / C	D-ITG	110	0	G.711.2	5 / 15
35	C / C	D-ITG	110	1	G.711.2	5 / 15
36	C / C	D-ITG	110	1.5	G.711.2	5 / 15
37	C / C	D-ITG	110	0	G.729.2	5 / 15
38	C / C	D-ITG	110	1	G.729.2	5 / 15
39	C / C	D-ITG	110	1.5	G.729.2	5 / 15
40	C / C	D-ITG	110	0	G.729.3	5 / 15
41	C / C	D-ITG	110	1	G.729.3	5 / 15
42	C / C	D-ITG	110	1.5	G.729.3	5 / 15
43	C / C	D-ITG	110	0	G.723.1	5 / 15
44	C / C	D-ITG	110	1	G.723.1	5 / 15
45	C / C	D-ITG	110	1.5	G.723.1	5 / 15
46	C / C	D-ITG	110	0	G.711.1	15 / 15
47	C / C	D-ITG	110	1	G.711.1	15 / 15
48	C / C	D-ITG	110	1.5	G.711.1	15 / 15
49	C / C	D-ITG	110	0	G.711.2	15 / 15
50	C / C	D-ITG	110	1	G.711.2	15 / 15
51	C / C	D-ITG	110	1.5	G.711.2	15 / 15
52	C / C	D-ITG	110	0	G.729.2	15 / 15
53	C / C	D-ITG	110	1	G.729.2	15 / 15
54	C / C	D-ITG	110	1.5	G.729.2	15 / 15
55	C / C	D-ITG	110	0	G.729.3	15 / 15
56	C / C	D-ITG	110	1	G.729.3	15 / 15
57	C / C	D-ITG	110	1.5	G.729.3	15 / 15
58	C / C	D-ITG	110	0	G.723.1	15 / 15
59	C / C	D-ITG	110	1	G.723.1	15 / 15
60	C / C	D-ITG	110	1.5	G.723.1	15 / 15

Table 3.15: Test design (part 2)

Test number	Network	Generator	Duration (s)	overhead	codec	Packet loss (%)
61	C / S	D-ITG	110	0	G.711.1	0 / 0
62	C / S	D-ITG	110	1	G.711.1	0 / 0
63	C / S	D-ITG	110	1.5	G.711.1	0 / 0
64	C / S	D-ITG	110	0	G.711.2	0 / 0
65	C / S	D-ITG	110	1	G.711.2	0 / 0
66	C / S	D-ITG	110	1.5	G.711.2	0 / 0
67	C / S	D-ITG	110	0	G.729.2	0 / 0
68	C / S	D-ITG	110	1	G.729.2	0 / 0
69	C / S	D-ITG	110	1.5	G.729.2	0 / 0
70	C / S	D-ITG	110	0	G.729.3	0 / 0
71	C / S	D-ITG	110	1	G.729.3	0 / 0
72	C / S	D-ITG	110	1.5	G.729.3	0 / 0
73	C / S	D-ITG	110	0	G.723.1	0 / 0
74	C / S	D-ITG	110	1	G.723.1	0 / 0
75	C / S	D-ITG	110	1.5	G.723.1	0 / 0
76	C / S	D-ITG	110	0	G.711.1	5 / 5
77	C / S	D-ITG	110	1	G.711.1	5 / 5
78	C / S	D-ITG	110	1.5	G.711.1	5 / 5
79	C / S	D-ITG	110	0	G.711.2	5 / 5
80	C / S	D-ITG	110	1	G.711.2	5 / 5
81	C / S	D-ITG	110	1.5	G.711.2	5 / 5
82	C / S	D-ITG	110	0	G.729.2	5 / 5
83	C / S	D-ITG	110	1	G.729.2	5 / 5
84	C / S	D-ITG	110	1.5	G.729.2	5 / 5
85	C / S	D-ITG	110	0	G.729.3	5 / 5
86	C / S	D-ITG	110	1	G.729.3	5 / 5
87	C / S	D-ITG	110	1.5	G.729.3	5 / 5
88	C / S	D-ITG	110	0	G.723.1	5 / 5
89	C / S	D-ITG	110	1	G.723.1	5 / 5
90	C / S	D-ITG	110	1.5	G.723.1	5 / 5

Table 3.16: Test design (part 3)

Test number	Network	Generator	Duration (s)	overhead	codec	Packet loss (%)
91	C / S	D-ITG	110	0	G.711.1	5 / 15
92	C / S	D-ITG	110	1	G.711.1	5 / 15
93	C / S	D-ITG	110	1.5	G.711.1	5 / 15
94	C / S	D-ITG	110	0	G.711.2	5 / 15
95	C / S	D-ITG	110	1	G.711.2	5 / 15
96	C / S	D-ITG	110	1.5	G.711.2	5 / 15
97	C / S	D-ITG	110	0	G.729.2	5 / 15
98	C / S	D-ITG	110	1	G.729.2	5 / 15
99	C / S	D-ITG	110	1.5	G.729.2	5 / 15
100	C / S	D-ITG	110	0	G.729.3	5 / 15
101	C / S	D-ITG	110	1	G.729.3	5 / 15
102	C / S	D-ITG	110	1.5	G.729.3	5 / 15
103	C / S	D-ITG	110	0	G.723.1	5 / 15
104	C / S	D-ITG	110	1	G.723.1	5 / 15
105	C / S	D-ITG	110	1.5	G.723.1	5 / 15
106	C / S	D-ITG	110	0	G.711.1	15 / 5
107	C / S	D-ITG	110	1	G.711.1	15 / 5
108	C / S	D-ITG	110	1.5	G.711.1	15 / 5
109	C / S	D-ITG	110	0	G.711.2	15 / 5
110	C / S	D-ITG	110	1	G.711.2	15 / 5
111	C / S	D-ITG	110	1.5	G.711.2	15 / 5
112	C / S	D-ITG	110	0	G.729.2	15 / 5
113	C / S	D-ITG	110	1	G.729.2	15 / 5
114	C / S	D-ITG	110	1.5	G.729.2	15 / 5
115	C / S	D-ITG	110	0	G.729.3	15 / 5
116	C / S	D-ITG	110	1	G.729.3	15 / 5
117	C / S	D-ITG	110	1.5	G.729.3	15 / 5
118	C / S	D-ITG	110	0	G.723.1	15 / 5
119	C / S	D-ITG	110	1	G.723.1	15 / 5
120	C / S	D-ITG	110	1.5	G.723.1	15 / 5

Table 3.17: Test design (part 4)

Test number	Network	Generator	Duration (s)	overhead	codec	Packet loss (%)
121	C / S	D-ITG	110	0	G.711.1	15 / 15
122	C / S	D-ITG	110	1	G.711.1	15 / 15
123	C / S	D-ITG	110	1.5	G.711.1	15 / 15
124	C / S	D-ITG	110	0	G.711.2	15 / 15
125	C / S	D-ITG	110	1	G.711.2	15 / 15
126	C / S	D-ITG	110	1.5	G.711.2	15 / 15
127	C / S	D-ITG	110	0	G.729.2	15 / 15
128	C / S	D-ITG	110	1	G.729.2	15 / 15
129	C / S	D-ITG	110	1.5	G.729.2	15 / 15
130	C / S	D-ITG	110	0	G.729.3	15 / 15
131	C / S	D-ITG	110	1	G.729.3	15 / 15
132	C / S	D-ITG	110	1.5	G.729.3	15 / 15
133	C / S	D-ITG	110	0	G.723.1	15 / 15
134	C / S	D-ITG	110	1	G.723.1	15 / 15
135	C / S	D-ITG	110	1.5	G.723.1	15 / 15

Table 3.18: Test design (part 5): From test 1 to test 60 with two cellular networks, from test 60 to test 135 with one cellular network and one satellite network. These tests are focus on the different packet loss rate and codec types.

mentioned before, each test will last 110s. Each test will be done 3 times with different overhead 0%, 100% and 150%. The test list is shown in Table3.14 to Table3.18. The traffic is generated by D-ITG[63], and the codec group are G.711, G.729, G.723. In particular, for G.711, there are two different functions to send out packets, G.711.1 sent out one voice slice at one time and G.711.2 send out two slices a time. The same for G.729, G.729.2 send out two slices a time while G.729.3 send out three of them. Since different slice number will lead to a different codec delay, here after, we consider G.711.1 and G.711.2, G.729.2 and G.729.3 as different codec types.

The test is executed on the software level. The communication will be setup between three virtual machines. One is used as a client, another is server, and the third as a router in the middle. The router is used as a time server as well. In all tests, the three virtual machines are time synchronized with each other. The traffic will be generated on the client side by two kinds of traffic generators D-ITG and iperf [106]. We use Netem [110] to simulate the channels delay and packets loss. In order to avoid the influence from MP-UDP software, the Netem is executed on the router virtual machine. In this

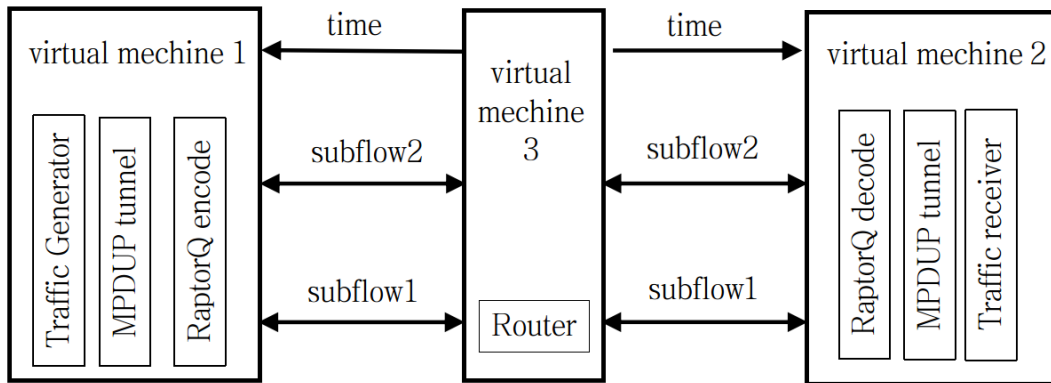


Figure 3.22: Test setup for scheduler RaptorQ

way, the tunnel built up by MP-UDP has no effect on the delay and packet loss setting. Further more, before each test, we use iperf to generate 5 seconds payload over each possible link to confirm if the channel has been set with the correct packet loss rate, and if the Netem is working properly.

Once we make sure that the test bed is working as our expecting, we will begin our tests. The MP-UDP software will generate a tunnel containing two subflows. On the client side, traffic is generated and sent to the server side though the tunnel. On the server side, traffic will be received by the corresponding software and as well as recorded with the performance parameters. From the media software's point of view, the communication is over single link. Two virtual machines are synchronized with time server (router) virtual machine by one of the subflow, which will involve in the data communication as well. The time synchronization is done by Network Time Protocol (NTP) software. The time synchronization needs very few packets, thus it will not influence the channel capacity. The structure has been shown in Figure3.22.

3.5.5.2 Results and analysis based on two cellular networks

We will separate the analysis in two main parts based on the simulated transmission environment (the transmission over two cellular channels or one cellular channel and one satellite channel). In both two parts, we will focus on the total receiving time, the average delay and the lost packet number. In this section, we depict the results and conclusions for two cellular networks.

1. Receiving time

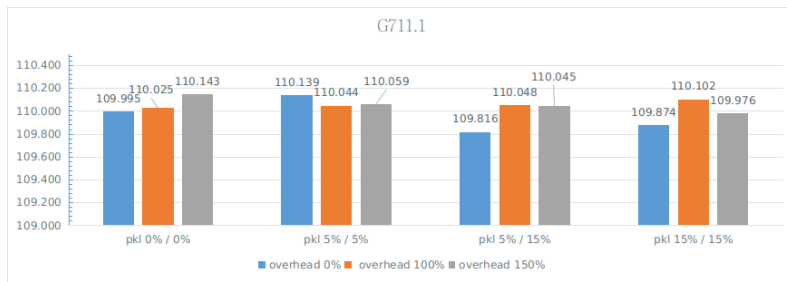
We plot the receiving time in Figure3.23 and Figure3.24. In Figure3.23, each figure stands for a different kind of codec type. In every figure, according to the different packet loss setting of these simulated channels, we group different overhead rate in the same cluster. In Figure3.24, we separate the receiving time based on the overhead rate. At this time, the clusters depend on the codec types, the diverse packet loss settings are represent by mutative color columns. As we can see in these following figures, they show that, there are not a unified rule to summarize the receiving time changing. Generally speaking, the receiving time always stays in the interval from 109 seconds to 111 seconds. The diversity of codec types, packet loss rate and overhead rate, have no significant, or in other word, have no visible influence on the receiving time. We can use this characteristic to against the packet loss by adding more overhead.

2. Average delay

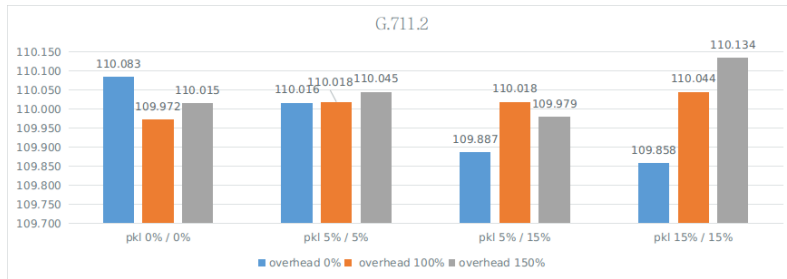
After we have analyzed the total receiving time, it comes to the average delay. The average delay is measured by seconds, and the results have been plot as the column chart in Figure3.25. According to Figure3.25, we conclude that, except codec G.723.1, for all the other 4 codec with overhead 0%, channel packet loss rate 5% / 15% has always the biggest average delay. In fact, the average delay for same codec but different overhead rate does not change a lot. From Figure3.26, we can find that the G.711.1 has the largest average delay, whilst G.723.1 has the smallest average delay.

Figure3.27a respects to overhead 0%. The average delay of all kinds of codecs and all kinds loss rates is 198.54ms. Loss rate 5% / 15% has the largest different among different codecs. Loss rate 0% / 0% has the most stable performance on average delay.

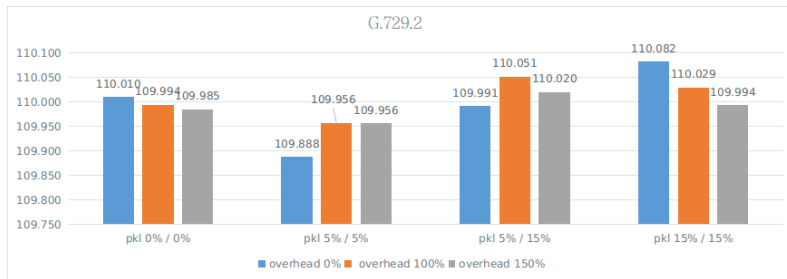
Figure3.27b is based on overhead 100%. The average delay of all kinds of codecs and all kinds loss rates is 242.126ms. The maximum average delay, which is 315ms, appears at the case of loss rate 15% / 15% with codec type G.711.1, meanwhile,



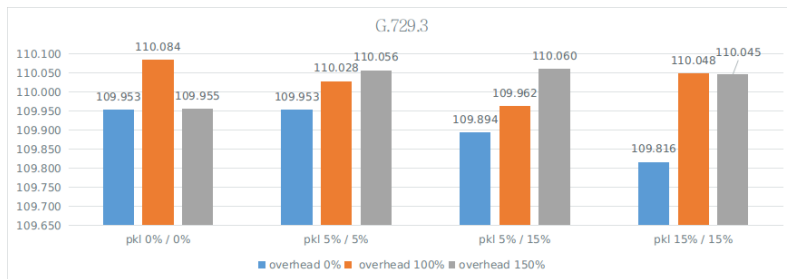
(a) Receiving time for G.711.1



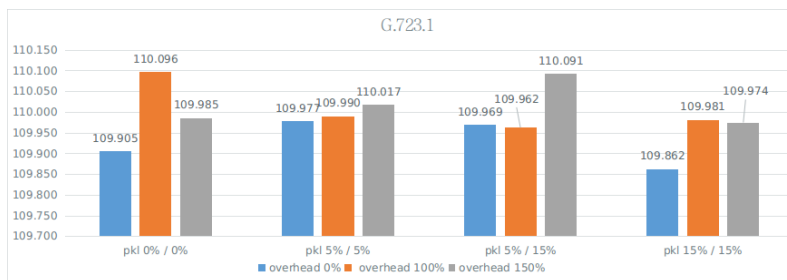
(b) Receiving time for G.711.2



(c) Receiving time for G.729.2

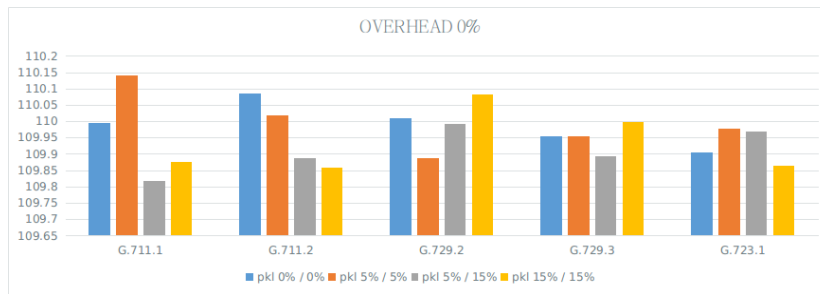


(d) Receiving time for G.729.3

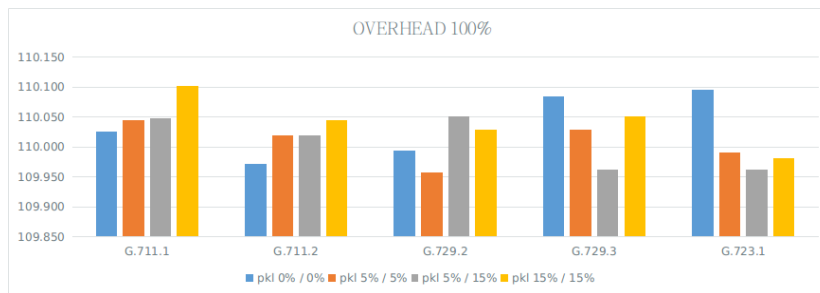


(e) Receiving time for G.723.1

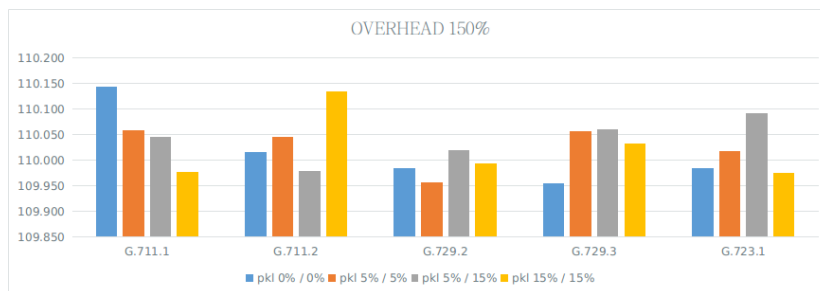
Figure 3.23: Receiving time for different codecs



(a) Receiving time for overhead 0%



(b) Receiving time for overhead 100%



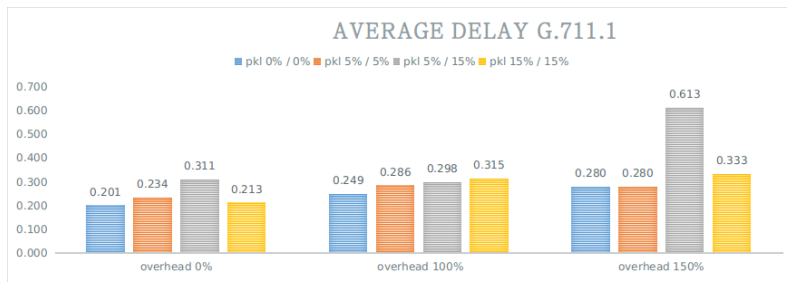
(c) Receiving time for overhead 150%

Figure 3.24: Receiving time for different overheads

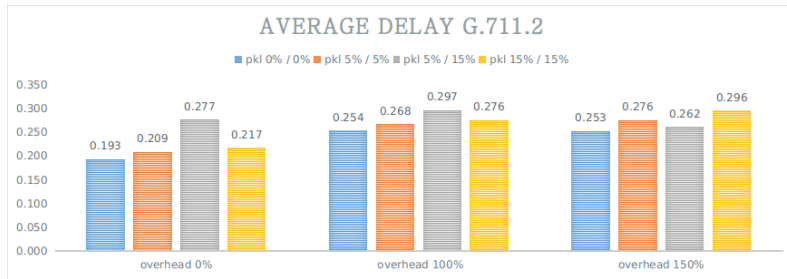
the minimum average delay is 168ms on the case of loss rate 0% / 0% with G.723.1 codec.

Figure 3.27c presents those behaviors with overhead 150%. The average delay of all kinds of codecs and all kinds loss rates is 266ms. There is a strange average delay behavior on the case of loss rate 5% / 15% with codec type G.771.1. This unexpected behavior may cause by the error within the virtual machine environment, but not the problem of the MP-UDP software.

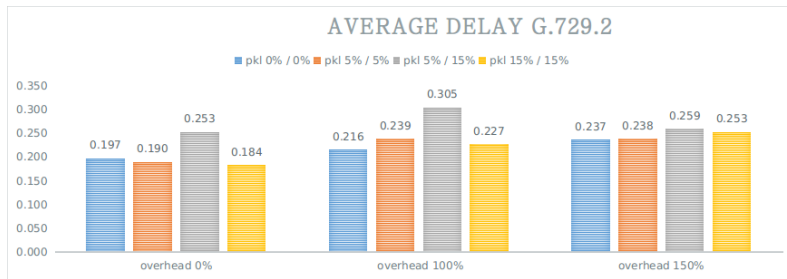
The last part of the average delay analysis is the comparison among overhead rates. From Figure 3.28, we easily notice that, the overhead with 150% has the the biggest average delay. This because with larger overhead rate, we should



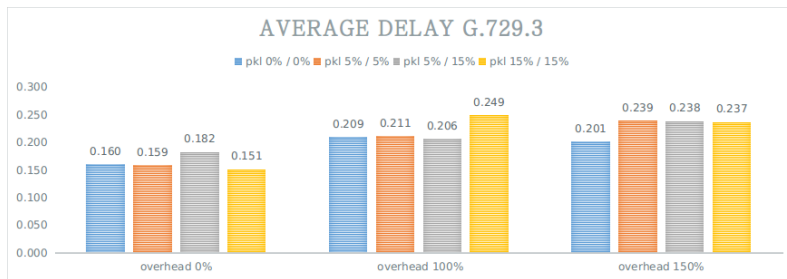
(a) Average delay for G.711.1



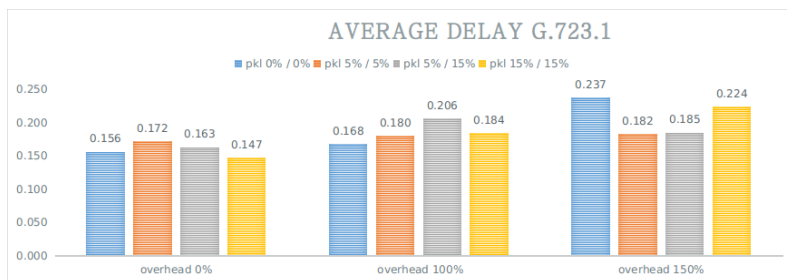
(b) Average delay for G.711.2



(c) Average delay for G.729.2



(d) Average delay for G.729.3



(e) Average delay for G.723.1

Figure 3.25: Average delay for different codecs

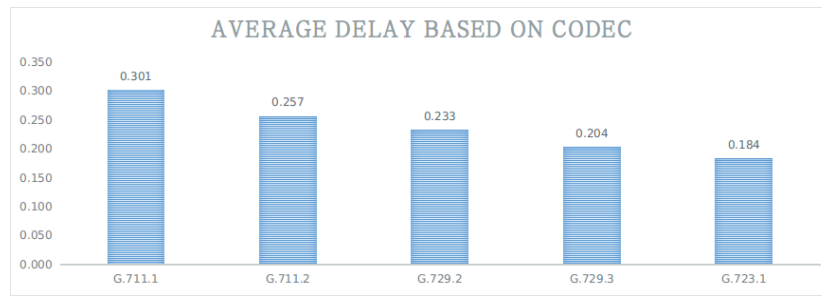


Figure 3.26: Average delay according to different codec types

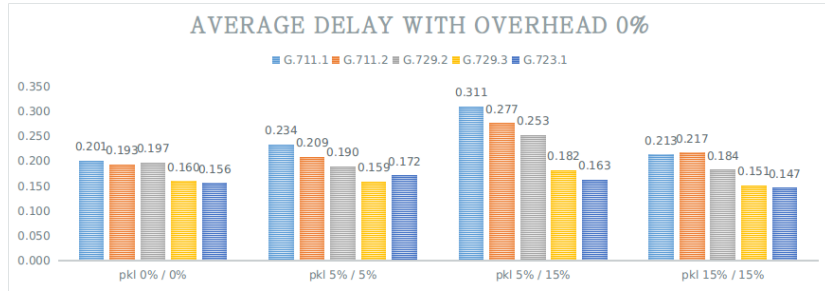
spend more time to drop the useless overheads. When the channel loss rate is 0%, it is useless to use a high overhead rate, but with the loss rate increasing, the higher overhead rate we have, the higher capacity to tolerant the packet loss we get. Moreover, from left to right in Figure3.28, the average delay is decreasing with different codec types. G.711.1 has the worst performance on average delay, while the G.723.1 has the best behavior on this issue.

3. Jitter

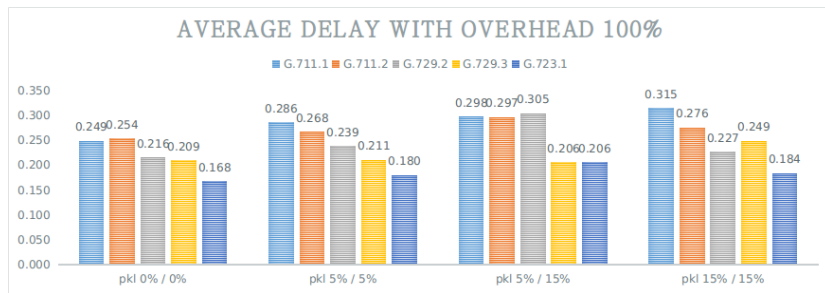
Figure3.29 shows the average jitter for diverse codec types with all kinds of overhead rates and packet loss rates. In Figure3.29, we know that, the G.711.2 stays the worst place because of the largest jitter value. G.729.3 has the smallest jitter value, this because the packets size for G.729.3 is 30 bytes and it is very close to the RaptorQ symbol size 32bytes. In other words, G.729.3 is encoded with less padding bytes than other codecs, so it has a perfect size for RaptorQ encode and decode. This instinct has advantage on saving encode and decode time. G.711.1 and G.723.1 have a similar jitter grade. In last section, we have known that the G.729.3 has the second last average delay. With both good performances on average delay and average jitter, G.729.3 can be consider as a good choice of the VoIP codec type.

From Figure3.30a to Figure3.30c we can find that G.729.3 not only has the smallest average jitter value, but also has the smallest jitter values in every kinds of packet loss rates and with different overhead rates. We should say that the G.729.3 has the best performance on the stability of the transmission.

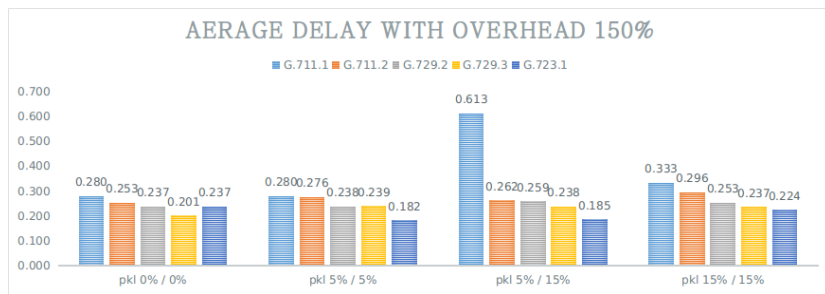
Moreover, we can find from Figure3.26 and Figure3.30, that G.711.2 has not the



(a) Average delay for Overhead 0%



(b) Average delay for Overhead 100%



(c) Average delay for Overhead 150%

Figure 3.27: Average delay for different overheads

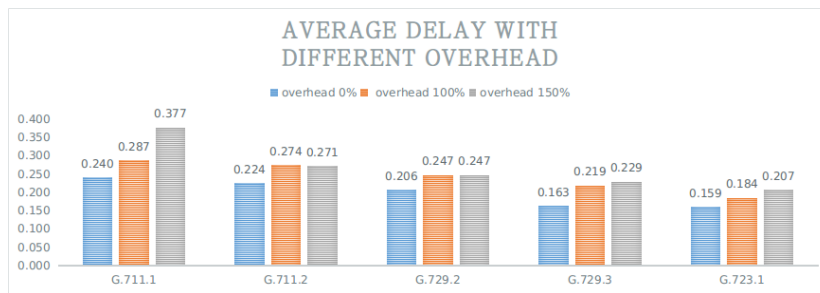


Figure 3.28: Average delay according for different overhead rates

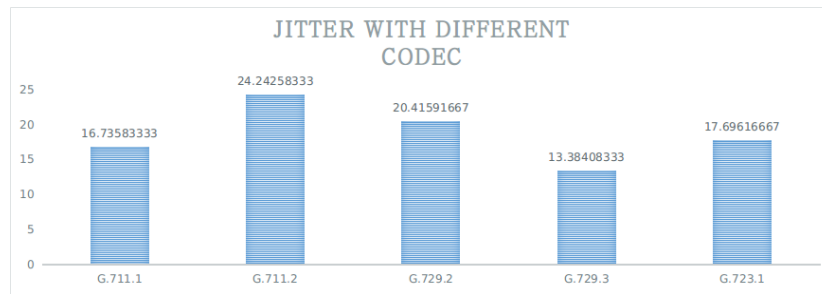
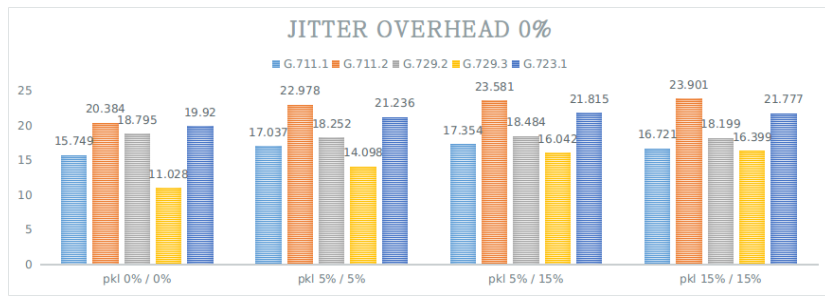


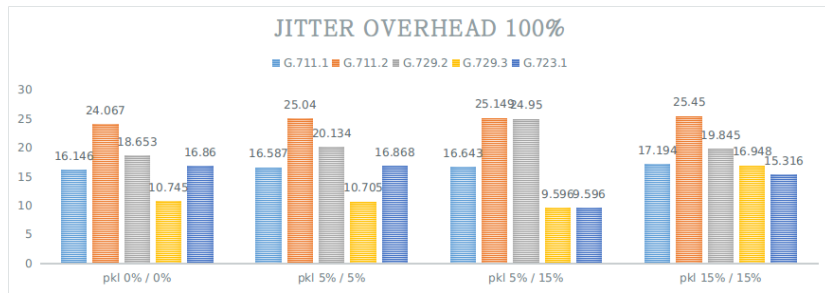
Figure 3.29: Average jitter according to different codec types

largest average delay, but the biggest jitter. In MP-UDP software, the decode capacity depends on both the packet rate and the packet size. According to our old experiences, the packet size close to 1k byte has the best decode rate and the lower packet rate has the better en/decode capacity. Back to this case, G.711.2 has the less bit rate than G.711.1, and half the packet rate. In details, G.711.1 has the packet size of 120 byte whilst the G.711.2 has the packet size 200 byte. That means, comparing with G.711.1, G.711.2 has the packet size closer to 1k, and has less packets to send out. G.711.2 needs less en/decode time than G.711.1, this is the reason why G.711.2 has less delay than G.711.1. However, when it comes to jitter, obviously, since G.711.2 has less packet rate, it is easy to have more jitter than the larger packet rate.

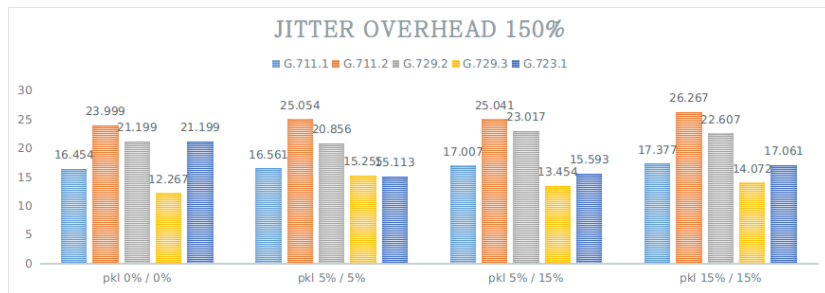
It is worth to notice that with packet loss rate equals to 0%, three kinds of overhead rates have quite similar behaviors on jitter. This means, if the transmission is over a perfect channel condition, add a certain rate of overhead will not influence the entire jitter performance. Or in other way to say, when the channel capacity is good enough, overhead isn't harmful for the jitter performance. Further more, when the transmission channels are not with the perfect conditions, the jitter performance isn't disturb by the proper setting of overhead rate, too. In fact, if we look twice at Figure3.31, it is easy to find out that not only G.729.3, but also all this five types of codec, 100% of overhead (not the 0% overhead) has the best performance on jitter. This may relate to the fact that, 100% overhead has a good capacity to deal with most of the tested packet loss case. For 150% overhead, the additional overhead adds the processing time consuming on



(a) Average jitter for Overhead 0%



(b) Average jitter for Overhead 100%



(c) Average jitter for Overhead 150%

Figure 3.30: Average jitter for different overheads

dropping the useless overhead symbols, which leads to an extra jitter.

The G.711.1 is the only codec has a very similar jitter value of all kinds of overhead rate. This may due to the highest packet rate in this five codec types.

4. Packet loss

From the test results we figure out that, with no packet loss channels, the packet loss rate for transmission generated by D-ITG is 0% as well. Thus, in Figure3.32 we have only three kinds of packet loss rate. Columns in the following figure show that, the receiving packet loss rate increases with the increasing of channel packet loss rate. For the reason that the MP-UDP software patches several packets together at one time of encoding procedure, the 5% of the channel loss may lead

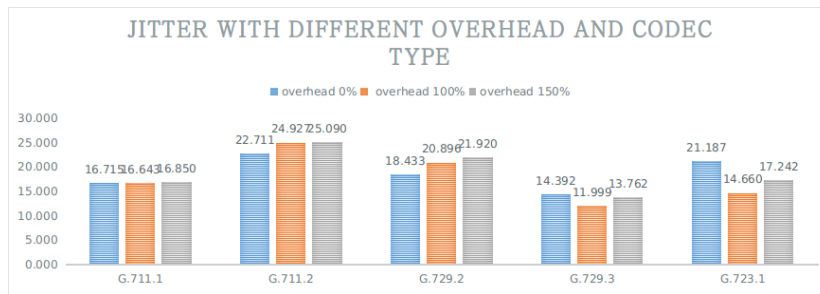


Figure 3.31: Average jitter for different codec types with diverse overhead rates

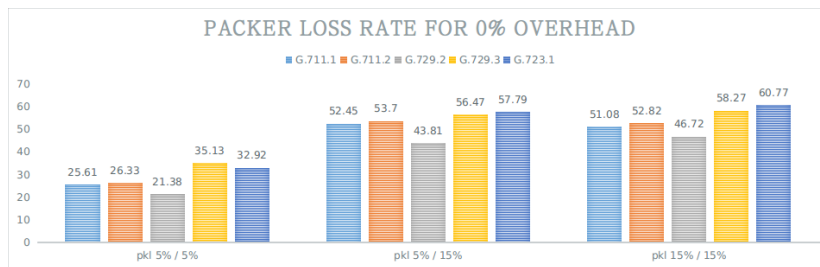


Figure 3.32: Packet loss rate for five types of codec transmissions with 0% overhead for three different kinds of channel packet loss rate setting

to more than 20% of the transmission packets loss. If we see carefully about the last two clusters, which are the groups with channel loss rate 5% / 15% and 15% / 15%, there are no significant different between this two groups, all of them have the values around 55%. For all three clusters, the codec G.729.2 performance the best.

With the overhead 100% and 150%, many of the tests have no transmission packet loss, and we use a table (Table3.19) to show the results. From Table3.19, we learned that with overhead more than or equal to 100%, the packet loss rate for all kinds of channel capacities is very close to 0% or even equal to 0%. This means, overhead gives a huge contribution on dealing with the channel packet loss problem. Here, we conclude that adding overhead will not have harmful influence on the total transmission time, the delay and jitter behaviors, but strongly improve the toleration of packet loss.

5. Summary on two cellular networks scenario

We analyzed the total transmission time, the average delay, the average jitter and the packet loss rate based on two cellular channel circumstances. Then we get

channel loss rate	Overhead	G.711.1 (%)	G.711.2 (%)	G.729.2 (%)	G.729.3 (%)	G.723.1 (%)
5% / 5%	0%	25.61	26.33	21.38	35.13	32.92
	100%	0	0	0.04	0	0
	150%	0	0	0.04	0	0
5% / 15%	0%	52.45	53.7	43.81	56.47	57.79
	100%	0	0	0.15	0	0
	150%	0	0	0	0	0
15% / 15%	0%	51.08	52.82	46.72	58.27	60.77
	100%	1.06	1.55	2.51	1.56	0.9
	150%	0.1	0	0.25	0.08	0.18

Table 3.19: The packet loss rate for five codec types with different channel loss rates and the overhead rates

following conclusions.

With different kinds of codec types, packet loss rate and overhead rate, there is no significant difference on the total transmission time. In other words, codec types, packet loss rate and overhead rate have no visible influence on the receiving time. For the average delay, except codec G.723.1, channel packet loss rate 5% / 15% has always the biggest average delay. When we focus on the codec types, the G.711.1 has the largest average delay, whilst G.723.1 has the smallest average delay, and the average delay is changing with different codec types. From the angle of loss rate, Loss rate 0% / 0% has the most stable performance on average delay. Finally, from the overhead's direct, the overhead with 150% has the the biggest average delay.

Now we move to the average jitter. The G.711.2 has the worst performance on jitter and G.729.3 has the best. G.711.1 and G.723.1 have a similar jitter grade. G.729.3 not only has the smallest overall average jitter value, but also has the smallest value in every kind of packet loss rate and every different overhead rate. In a word, the G.729.3 has the best performance on the stability of the transmission. 100% of overhead has the best performance on jitter. The G.711.1 is the only codec has a very similar jitter value of all kinds of overhead rate.

The last parameter is packet loss rate. With no packet loss channels, the packet loss rate for transmission is 0% as well. With 0% overhead, the packet loss rate

increases with the increasing of channel packet loss rate. Due to the RaptorQ encode strategy, without overhead, 5% channel loss may lead to more than 20% of the transmission packets loss. For channel loss rate 5% / 15% and 15% / 15%, there are no remarkable different between this two groups, all of them have the loss rate around 55%. With overhead more than or equal to 100%, the packet loss rate for all kinds of channel capacities is very close to 0%. Overhead gives a huge contribution on dealing with the channel loss problem, at the meantime, it will not impact the total transmission time, the delay and jitter behaviors.

3.5.5.3 Results and analysis based on one cellular and one satellite network

We have analyzed four features based on the scenario one (two cellular networks), now it is time to analyze those same four features under the second scenario: one cellular and one satellite network.

1. Receiving time

From now on, we start the studies about the behaviors over one cellular link and one satellite link. As previous, we begin with the total transmission time. The receiving time should almost equal to the sending time.

From Figure3.33a we learned that similar as last scenario, with different codec and different channel packet loss, the receiving time is always fall in the interval 109s to 111s. Thus, we can say that, for our MP-UDP software, the receiving time is always acceptable for this 5 different type of codec.

From Figure3.33b we find that, the shortest receiving time is not with 0% of packet loss, but on the high packet loss side (on the right side in this figure). 0% of overhead has the shortest receiving time (except for the 0% of channel loss rate). The reason for this phenomenon is that overhead adds the probability of successfully decode, however, it also leads the packet staying a longer time in the buffer to be decoded. Meanwhile, the new arriving symbols, which belongs to those decoded packets, need to be dropped, too.

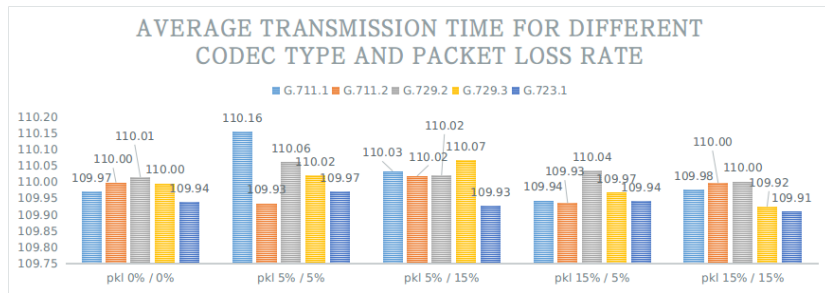
Figure3.33c and Figure3.33d show the receiving time respects to the codec types. The overhead gives different influence on the receiving time for different codec types. But showing as an average receiving time in Figure3.33d, the G.723.1 has the shortest receiving time whilst the G.729.3 has the longest one. In fact, since the receiving times for different codec types and overhead values are all acceptable, we don't have to pay much attention on the codec. As a good result, if we consider only with receiving time, we can freely choose any of codec types.

2. Average delay

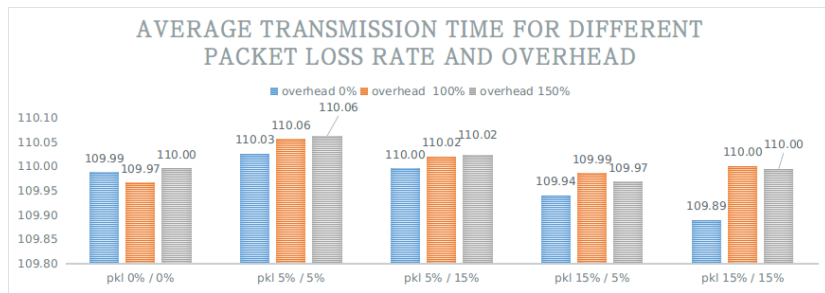
Delay is one of the key features for the mean opinion score. In practice, we want the delay as small as possible. Figure3.34a indicates average delay behaviors with 0% of the overhead rate, the largest average delay appears at the packet loss rate 0% for all kinds of codec types. The reason why is that the two channel with a great gap of transmission delay, and the packet arriving through the cellular channel should wait for the other packets arriving via the satellite link to decode. This may enlarge the average delay. But when we add packet losing in connecting channels, some of the packet can not be decode anymore. Though channel packet loss impacts the receiving packet loss rate, it do decrease the average delay.

Figure3.34b demonstrates the average delay with overhead 100%. From this figure we notice that the average delay with 5%/5% packet loss have a higher average delay level comparing with 0% overhead situation. This because thanks for the overhead symbols transmission, some symbols that can not be decoded with 0% of overhead can be decoded at this time. The waiting time for symbols in the buffer affects the average delay. In fact, if we see clear, with overhead 100%, all this five kind of channel loss rate increase their average delay somehow. The reason is the same as we have explain as 5%/5% packet loss rate.

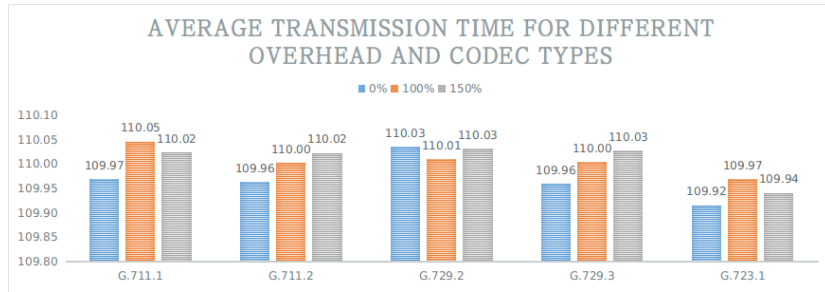
Figure3.34c reflects the behaviors of 150% overhead. From the results we know that the average delay increases a bit than the 100% overhead cases. This is not only because the waiting symbols in the buffer, but also because some overhead symbols need to be dropped after the packet has been decoded.



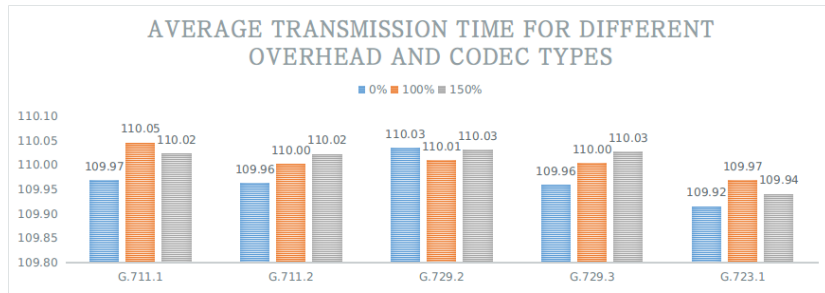
(a) Average transmission time with different packet loss rates and different codec types



(b) Average transmission time for different channel packet loss rates and overhead rates regardless the codec type

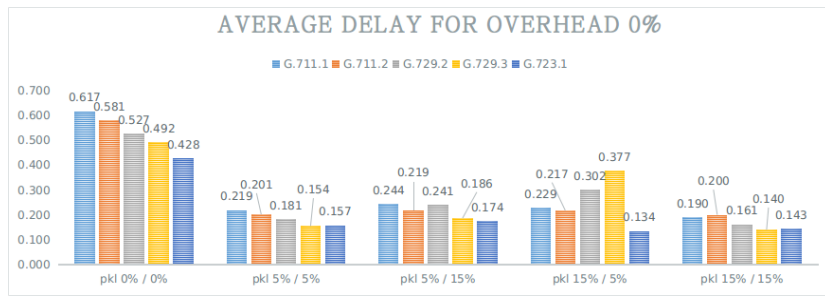


(c) Average transmission time for different overhead rates regard different the codec types

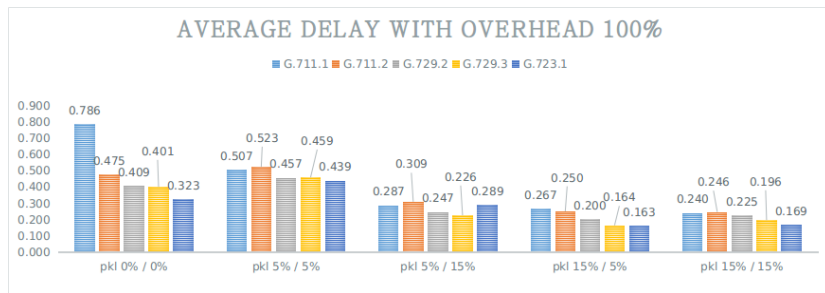


(d) Average transmission time for different codec types

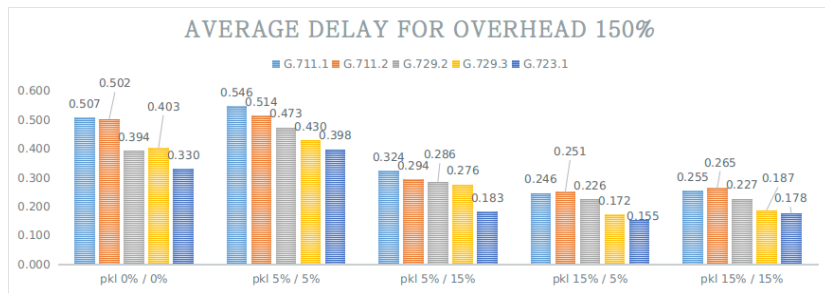
Figure 3.33: Average transmission time for different overhead rates and codec types



(a) Average delay time with different packet loss rate and different codec types for overhead 0%



(b) Average delay time with different packet loss rates and different codec types for overhead 100%



(c) Average delay time with different packet loss rates and different codec types for overhead 100%

Figure 3.34: Average delay for different overhead rates

Figure 3.35 is the average delay for different overhead. For example, the blue column in the first cluster is the average for overhead rate 0%, 100% and 150% of G.711.1 codic without packet loss. Generally speaking, the average delay goes down with the rising of the packet loss rate. The reason for this phenomena can be described as that the average delay in Figure 3.34 is impacted by three kinds of overhead rate. The dropping of useless overhead consumes processing time, and this will increase a little to the average delay. With a higher channel loss rate, some of the redundancy overhead symbols have been dropped naturally, and this helps the decoding procedure to save time.

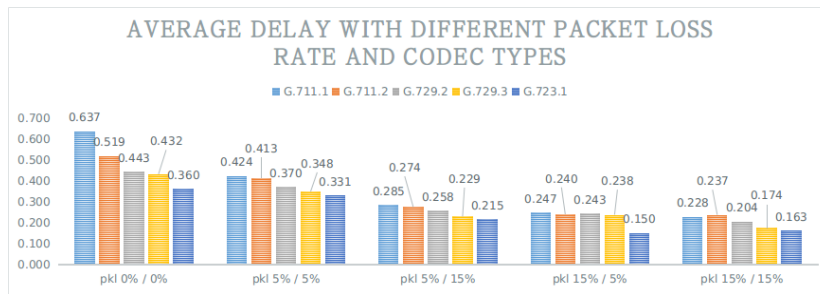


Figure 3.35: Average delay all kinds of overhead rate.

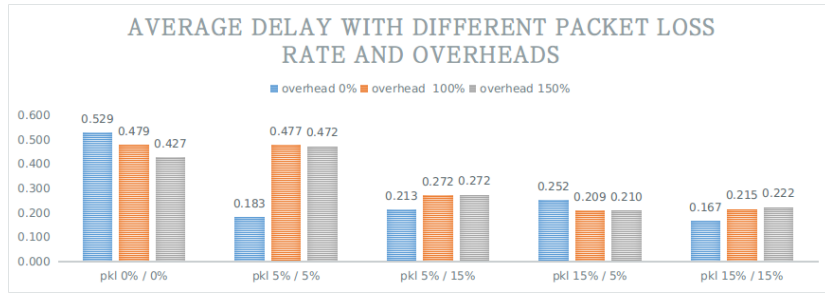
Figure 3.36a shows the average value of different codec types. For instance, the first blue column is the average value for five kinds of codecs with 0% overhead and 0% / 0% channel loss rate. From this figure, there is an interesting phenomenon, which is the performances for overhead 100% and 150% are always similar. That because in our MP-UDP software, we always separate a group of packet into two symbols. So the overhead 100% will add 2 additional symbols whilst 150% overhead adds 3 additional symbols. In account of the processors capacities, one additional symbol for each encode and decode group will not affect the delay a lot.

Figure 3.36b talks about the reaction of each codec. The difference between the different codec groups is larger than in side the same group. In details, the first two clusters are the same codec group, the third and fourth clusters are the second group, while the last cluster is the third group. In fact, from only the point of average delay, the G.723.1 performs the best.

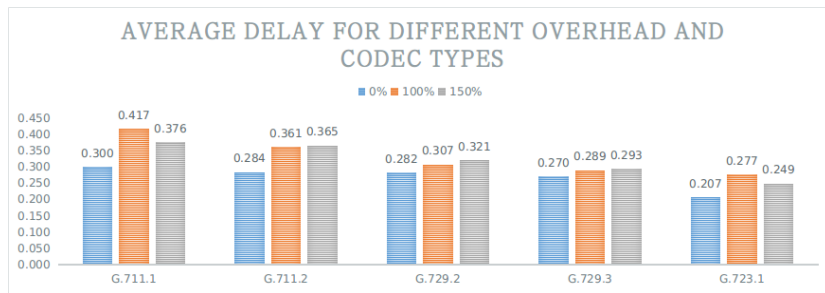
Those data shown in Figure 3.37 are only separated by the codec. We got the similar results as the cellular transmission environment, the overall average delay diminishes from left to right.

3. Jitter

From Figure 3.38 we got the data about the jitter for overhead 0%, 100% and 150%. If we compare each two of those three figures, we will see that the shape of overhead 0% is different from the other two. On the contrary, overhead 100% and 150% have very similar shapes. According to this, we can infer jitter is one parameter may have some relation with the packet loss rate.



(a) Average delay for different overhead rates regards the channel loss rates



(b) Average delay for different overhead rate regards the codec type

Figure 3.36: Average delay for different overhead rates and packet loss rate

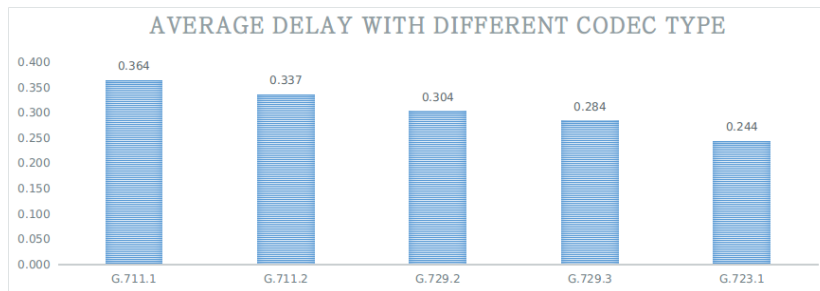
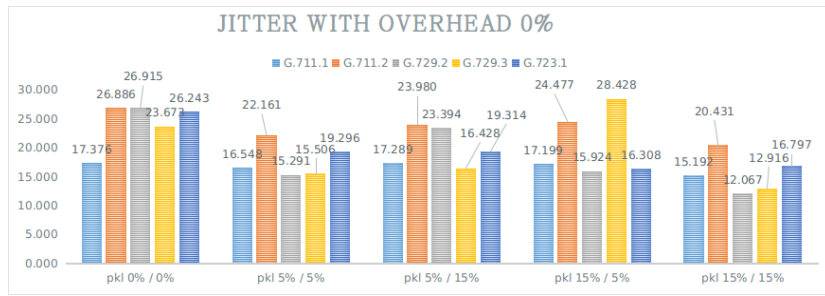


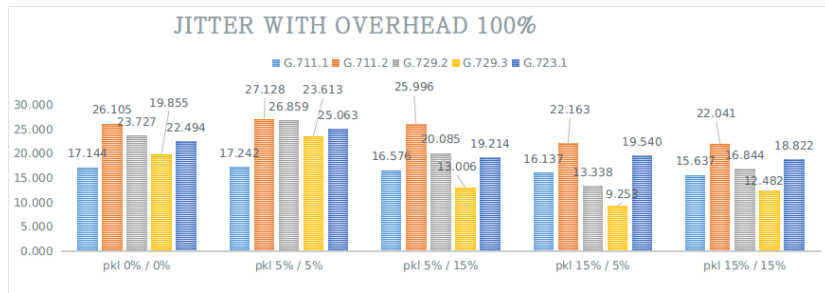
Figure 3.37: Average delay time with different different codec types

Figure 3.39 gives us a view about the jitter of different codec types. The way to calculate the data in Figure 3.39 is similar to Figure 3.35, which is the average with all three kinds of overhead. G.711.2 has the largest jitter value. Except 0% of the packet loss rate, the other channel packet loss rates have almost the same level of jitter.

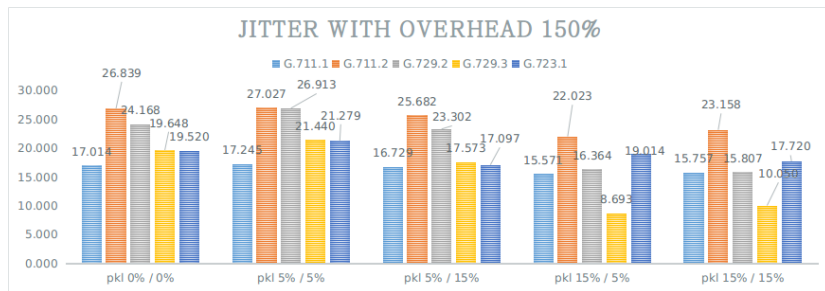
Figure 3.40 is the overall jitter for each codec type. Comparing Figure 3.39 with Figure 3.29, which is the same figure for two cellular links, it is easy to find that two figures have similar trend. For RaptorQ scheduler with a suitable setting of overhead, the jitter is affected more by the codec types rather than the link delay.



(a) AJitter with different packet loss rate and different codec type for overhead 0%



(b) Jitter with different packet loss rate and different codec type for overhead 100%



(c) Jitter with different packet loss rate and different codec type for overhead 150%

Figure 3.38: Jitter with different overhead rates

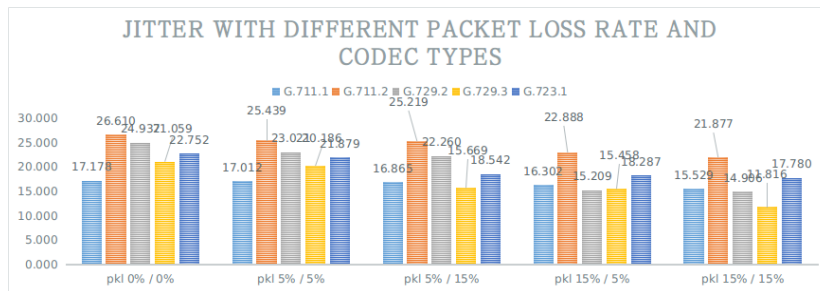


Figure 3.39: Jitter with different packet loss rate and codec types

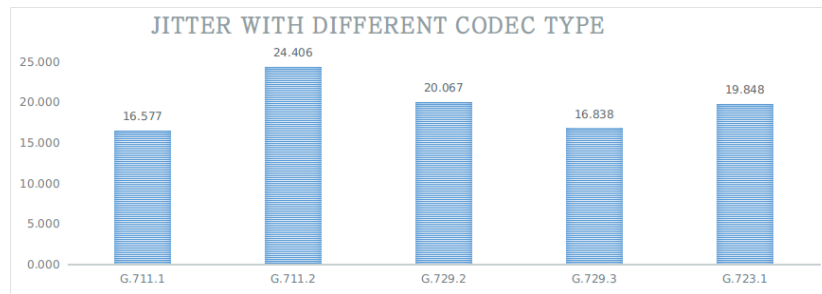
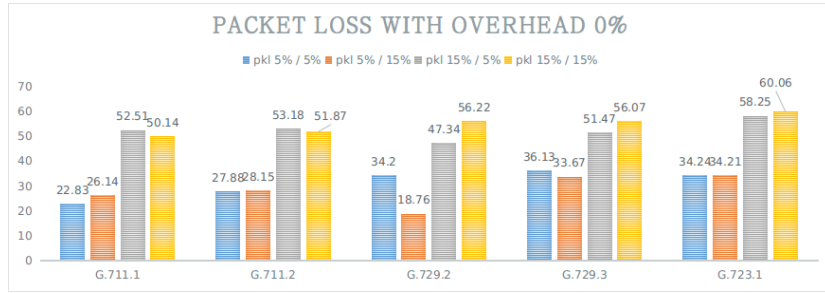


Figure 3.40: Jitter with different codec types

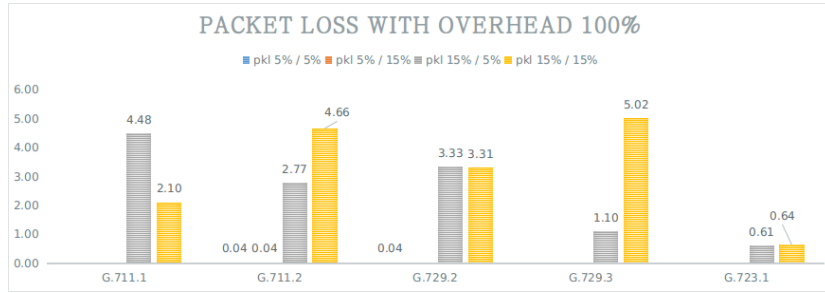
4. Packet loss

Here we begin the analysis for packet loss rate of different overheads and different codec types. Figure3.41a is the packet loss rate with overhead 0%. In Figure3.41a, we can see the pure reacts only depending on the codec types. Here we haven't shown the results of 0% channel loss rate, because they are always 0. From Figure3.41a we see that on one side, 5% / 5% and 5% / 15% have similar results, here 5% / 15% represents the cellular link with 5% of the loss rate while satellite link with 15% of loss rate. One other side 15% / 5% has similar value as 15% / 15%, here the cellular link has 15% loss rate and satellite link has 5% loss rate. According to this results, we can infer that, the transmission packet loss mainly depends on the less delay channel. We say the RaptorQ shows the similar feature as Balance. In another way to say, if there are two links to support the transmission, one is the cellular link with less delay and the other is the satellite link with more delay, the receiving packet loss rate impair more by the cellular link.

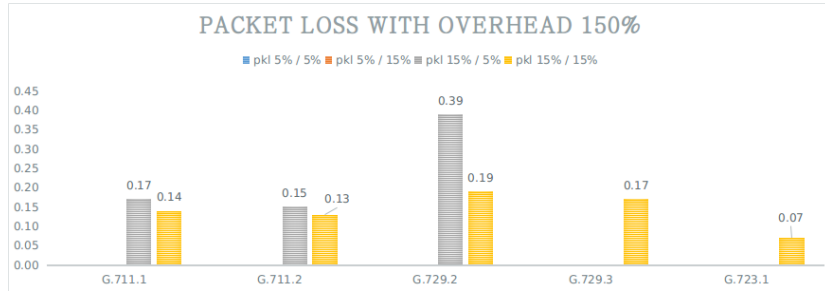
Figure3.41b and Figure3.41c show the packet loss rate of overhead 100% and 150%. Obviously, in these two situations, for the channel loss 5% / 5% and 5% / 15%, transmission loss rates are almost 0%. Moreover, for the channel loss rate 15% / 5% and 15% / 15%, transmission loss rate has the maximum value 5.02% in the case of 100% overhead, whilst 0.39% under the condition of overhead 150%. That means adding overhead can significant improve the transmission quality over lossy channels. Table3.20 shows some details about the packet loss rate we got from these tests.



(a) Packet loss rate with overhead 0%



(b) Packet loss rate with overhead 100%



(c) Packet loss rate with overhead 150%

Figure 3.41: Packet loss rate with different overhead rates

channel loss rate	overhead rate	G.711.1 (%)	G.711.2 (%)	G.729.2 (%)	G.729.3 (%)	G.723.1 (%)
5% / 5%	0%	22.83	27.88	34.2	36.13	34.24
	100%	0	0.04	0.04	0	0
	150%	0	0	0	0	0
5% / 15%	0%	26.14	28.15	18.76	33.67	34.21
	100%	0	0.04	0	0	0
	150%	0	0	0	0	0
15% / 5%	0%	52.51	53.18	47.34	51.47	58.25
	100%	4.48	2.77	3.33	1.1	0.61
	150%	0.17	0.15	0.39	0	0
15% / 15%	0%	50.14	51.87	56.22	56.07	60.06
	100%	2.1	4.66	3.31	5.02	0.64
	150%	0.14	0.13	0.19	0.17	0.07

Table 3.20: The packet loss rates for one cellular link and one satellite link

5. Summary for scenario two: one cellular and one satellite networks

Receiving time is always fall in the interval 109s to 111s. The overhead gives different influences on the receiving time for different codec types, and it adds the probability of successfully decode, however, it adds the receiving time as well. Delay is one of the key feature for the mean opinion score, channel loss can increases the transmission packet loss rate, but somehow decreases the average delay. Overhead will recover the packet loss, but increases a bit the average delay, too. Delay depends on the codec type, and the behaviors for two kinds of link environment (two cellular links and one cellular link one satellite link) got similar delay performance trend. Jitter is one parameter may also relate to the packet loss rate, and it strongly depends on the codec type rather than the link delay condition. The transmission packet loss mainly depends on the less delay channel, and overhead can significant improve the transmission quality over a lossy channel.

3.5.6 Performance comparison among basic, balance and RaptorQ schedulers

In this section, we will compare three kinds of schedulers, which we have involved in MP-UDP software, those are Basic, Balance and RaptorQ (overhead 150%). We will test the transmission always over two main scenarios, which are two cellular networks and one cellular one satellite network. The comparison will focus on the bandwidth on receiving side, the jitter, the packet loss rate and the out-of-order rate. By comparing those behaviors features, we can get some useful conclusion on the scheduler selection.

3.5.6.1 Test design

This time, we need the parameter of out of order packet rate. For this reason, we change our traffic generator from D-ITG to iperf. Each test lasts 110 seconds as before. In order to simulate the same codec packet size, and packet rate as previous, we set the packet length and bandwidth for five codec types (G.711.1, G.711.2, G.729.2, G.729.3 and G.723.1) as Table3.21.

codec type	IP head (byte)	voice payload (byte)	total payload	packet rate (pps)	bandwidth real(kbps)	bandwidth test(kbps)
G.711.1	40	80	120	100	94	100
G.711.2	40	160	200	50	78	80
G.729.2	40	20	60	50	24	25
G.729.3	40	30	70	33	18	20
G.723.1	40	20	60	33	16	20

Table 3.21: The transmission packets length, packet rate and bandwidth for each codec

In this test section, we use the same channel loss rate as last section: for two cellular networks, we test with 5% / 5%, 5% / 15% and 15% / 15%; for one cellular and one satellite network, we test with 5% / 5%, 5% / 15%, 15% / 5% and 15% / 15%. The tests list is shown in Table3.22 to Table3.24.

The test set up is same as before, the simulation involve in two subflows. Two virtual machines work as the client and the server. Another virtual machine works as the router. Three computers are time synchronized. The transmission is over UDP protocol. Only difference is that the traffic generator is iperf but not D-ITG.

3.5.6.2 Results analysis

1. Receiving bandwidth

The analysis will begin with the bandwidth on the receiving side. The sending bandwidths for different codec type are shown in Table3.21. The receiving bandwidth is influenced by the receiving time, receiving packets number and packet size. Since the packet size is fixed as it sent out, we say that the receiving bandwidth is decided by the receiving time and packets number. During our tests, all these tests have receiving time from 109.9 to 110.2 seconds. Most of these tests have 110 seconds the same as sending time. For this reason, we can consider that comparing between the sending bandwidth and receiving bandwidth depends on the packet number sent and received. Figure3.42 shows the bandwidth for two cellular networks with different schedulers. Basic has the receiving bandwidth as twice of the sending bandwidth, while Balance and RaptorQ have the similar receiving bandwidth as the sending one.

Test list	Network	Generator	Duration (s)	Scheduler	codec	Packet loss (%)
136	C / C	iperf	110	basic	G.711.1	0 / 0
137	C / C	iperf	110	balance	G.711.1	0 / 0
138	C / S	iperf	110	basic	G.711.1	0 / 0
139	C / S	iperf	110	balance	G.711.1	0 / 0
140	C / C	iperf	110	basic	G.711.1	5 / 5
141	C / C	iperf	110	balance	G.711.1	5 / 5
142	C / C	iperf	110	RaptorQ(OH1.5)	G.711.1	5 / 5
143	C / C	iperf	110	basic	G.711.2	5 / 5
144	C / C	iperf	110	balance	G.711.2	5 / 5
145	C / C	iperf	110	RaptorQ(OH1.5)	G.711.2	5 / 5
146	C / C	iperf	110	basic	G.729.2	5 / 5
147	C / C	iperf	110	balance	G.729.2	5 / 5
148	C / C	iperf	110	RaptorQ(OH1.5)	G.729.2	5 / 5
149	C / C	iperf	110	basic	G.729.3	5 / 5
150	C / C	iperf	110	balance	G.729.3	5 / 5
151	C / C	iperf	110	RaptorQ(OH1.5)	G.729.3	5 / 5
152	C / C	iperf	110	basic	G.723.1	5 / 5
153	C / C	iperf	110	balance	G.723.1	5 / 5
154	C / C	iperf	110	RaptorQ(OH1.5)	G.723.1	5 / 5
155	C / S	iperf	110	basic	G.711.1	5 / 5
156	C / S	iperf	110	balance	G.711.1	5 / 5
157	C / S	iperf	110	RaptorQ(OH1.5)	G.711.1	5 / 5
158	C / S	iperf	110	basic	G.711.2	5 / 5
159	C / S	iperf	110	balance	G.711.2	5 / 5
160	C / S	iperf	110	RaptorQ(OH1.5)	G.711.2	5 / 5
161	C / S	iperf	110	basic	G.729.2	5 / 5
162	C / S	iperf	110	balance	G.729.2	5 / 5
163	C / S	iperf	110	RaptorQ(OH1.5)	G.729.2	5 / 5
164	C / S	iperf	110	basic	G.729.3	5 / 5
165	C / S	iperf	110	balance	G.729.3	5 / 5
166	C / S	iperf	110	RaptorQ(OH1.5)	G.729.3	5 / 5
167	C / S	iperf	110	basic	G.723.1	5 / 5
168	C / S	iperf	110	balance	G.723.1	5 / 5
169	C / S	iperf	110	RaptorQ(OH1.5)	G.723.1	5 / 5

Table 3.22: transmission test list (part1).

Test list	Network	Generator	Duration (s)	Scheduler	codec	Packet loss (%)
170	C / C	iperf	110	basic	G.711.1	5 / 15
171	C / C	iperf	110	balance	G.711.1	5 / 15
172	C / C	iperf	110	RaptorQ(OH1.5)	G.711.1	5 / 15
173	C / C	iperf	110	basic	G.711.2	5 / 15
174	C / C	iperf	110	balance	G.711.2	5 / 15
175	C / C	iperf	110	RaptorQ(OH1.5)	G.711.2	5 / 15
176	C / C	iperf	110	basic	G.729.2	5 / 15
177	C / C	iperf	110	balance	G.729.2	5 / 15
178	C / C	iperf	110	RaptorQ(OH1.5)	G.729.2	5 / 15
179	C / C	iperf	110	basic	G.729.3	5 / 15
180	C / C	iperf	110	balance	G.729.3	5 / 15
181	C / C	iperf	110	RaptorQ(OH1.5)	G.729.3	5 / 15
182	C / C	iperf	110	basic	G.723.1	5 / 15
183	C / C	iperf	110	balance	G.723.1	5 / 15
184	C / C	iperf	110	RaptorQ(OH1.5)	G.723.1	5 / 15
185	C / S	iperf	110	basic	G.711.1	5 / 15
186	C / S	iperf	110	balance	G.711.1	5 / 15
187	C / S	iperf	110	RaptorQ(OH1.5)	G.711.1	5 / 15
188	C / S	iperf	110	basic	G.711.2	5 / 15
189	C / S	iperf	110	balance	G.711.2	5 / 15
190	C / S	iperf	110	RaptorQ(OH1.5)	G.711.2	5 / 15
191	C / S	iperf	110	basic	G.729.2	5 / 15
192	C / S	iperf	110	balance	G.729.2	5 / 15
193	C / S	iperf	110	RaptorQ(OH1.5)	G.729.2	5 / 15
194	C / S	iperf	110	basic	G.729.3	5 / 15
195	C / S	iperf	110	balance	G.729.3	5 / 15
196	C / S	iperf	110	RaptorQ(OH1.5)	G.729.3	5 / 15
197	C / S	iperf	110	basic	G.723.1	5 / 15
198	C / S	iperf	110	balance	G.723.1	5 / 15
199	C / S	iperf	110	RaptorQ(OH1.5)	G.723.1	5 / 15
200	C / S	iperf	110	basic	G.711.1	15 / 5
201	C / S	iperf	110	balance	G.711.1	15 / 5
202	C / S	iperf	110	RaptorQ(OH1.5)	G.711.1	15 / 5
203	C / S	iperf	110	basic	G.711.2	15 / 5
204	C / S	iperf	110	balance	G.711.2	15 / 5
205	C / S	iperf	110	RaptorQ(OH1.5)	G.711.2	15 / 5
206	C / S	iperf	110	basic	G.729.2	15 / 5
207	C / S	iperf	110	balance	G.729.2	15 / 5
208	C / S	iperf	110	RaptorQ(OH1.5)	G.729.2	15 / 5
209	C / S	iperf	110	basic	G.729.3	15 / 5
210	C / S	iperf	110	balance	G.729.3	15 / 5
211	C / S	iperf	110	RaptorQ(OH1.5)	G.729.3	15 / 5

Table 3.23: transmission test list (part2).

Test list	Network	Generator	Duration (s)	Scheduler	codec	Packet loss (%)
212	C / S	iperf	110	basic	G.723.1	15 / 5
213	C / S	iperf	110	balance	G.723.1	15 / 5
214	C / S	iperf	110	RaptorQ(OH1.5)	G.723.1	15 / 5
215	C / C	iperf	110	basic	G.711.1	15 / 15
216	C / C	iperf	110	balance	G.711.1	15 / 15
217	C / C	iperf	110	RaptorQ(OH1.5)	G.711.1	15 / 15
218	C / C	iperf	110	basic	G.711.2	15 / 15
219	C / C	iperf	110	balance	G.711.2	15 / 15
220	C / C	iperf	110	RaptorQ(OH1.5)	G.711.2	15 / 15
221	C / C	iperf	110	basic	G.729.2	15 / 15
222	C / C	iperf	110	balance	G.729.2	15 / 15
223	C / C	iperf	110	RaptorQ(OH1.5)	G.729.2	15 / 15
224	C / C	iperf	110	basic	G.729.3	15 / 15
225	C / C	iperf	110	balance	G.729.3	15 / 15
226	C / C	iperf	110	RaptorQ(OH1.5)	G.729.3	15 / 15
227	C / C	iperf	110	basic	G.723.1	15 / 15
228	C / C	iperf	110	balance	G.723.1	15 / 15
229	C / C	iperf	110	RaptorQ(OH1.5)	G.723.1	15 / 15
230	C / S	iperf	110	basic	G.711.1	15 / 15
231	C / S	iperf	110	balance	G.711.1	15 / 15
232	C / S	iperf	110	RaptorQ(OH1.5)	G.711.1	15 / 15
233	C / S	iperf	110	basic	G.711.2	15 / 15
234	C / S	iperf	110	balance	G.711.2	15 / 15
235	C / S	iperf	110	RaptorQ(OH1.5)	G.711.2	15 / 15
236	C / S	iperf	110	basic	G.729.2	15 / 15
237	C / S	iperf	110	balance	G.729.2	15 / 15
238	C / S	iperf	110	RaptorQ(OH1.5)	G.729.2	15 / 15
239	C / S	iperf	110	basic	G.729.3	15 / 15
240	C / S	iperf	110	balance	G.729.3	15 / 15
241	C / S	iperf	110	RaptorQ(OH1.5)	G.729.3	15 / 15
242	C / S	iperf	110	basic	G.723.1	15 / 15
243	C / S	iperf	110	balance	G.723.1	15 / 15
244	C / S	iperf	110	RaptorQ(OH1.5)	G.723.1	15 / 15

Table 3.24: transmission test list for two main kinds of scenarios with different codec types, schedulers and channel loss rates (part3).

For Basic scheduler, Figure3.42a, with the increasing of channel packets loss rate, most of codecs have a decreasing receiving bandwidth (except G.729.3 for 5%/15%, this maybe the problem happened just once). This because, Basic scheduler simply duplicates the same packet in two subflows, and the more packets lossing, the less bandwidth getting on receiving side.

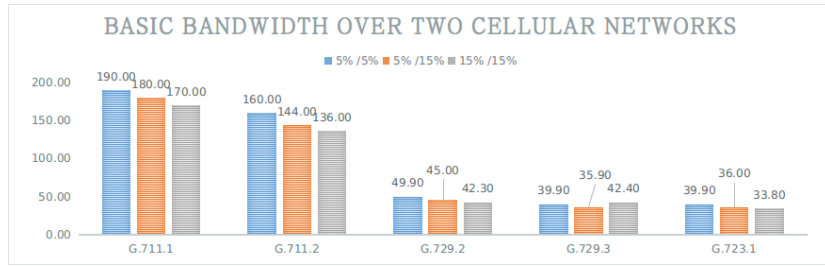
Figure3.42b show the bandwidth of Balance scheduler. It gets the similar performance as the Basic, but with less amount of decreased bandwidth. The reason is that Balance send every packet just once, and it will select the less delay channel to send the packet. We get less receiving bandwidth with 5%/15% than 5%/5%. The decision of choosing subflow will not relate to the channel packet loss rate and two channel with same delay setting, so the packet will be send to any of this two subflow randomly.

Figure3.42c is the bandwidth of RaptorQ scheduler. We can find that with different packet loss rate, the bandwidth does not change too much, and just few degeneration appears. As we have discussed before, with 150% overhead, the RaptorQ can well cover the channel packet loss. That is why RaptorQ has few degradation on the bandwidth on receiving side.

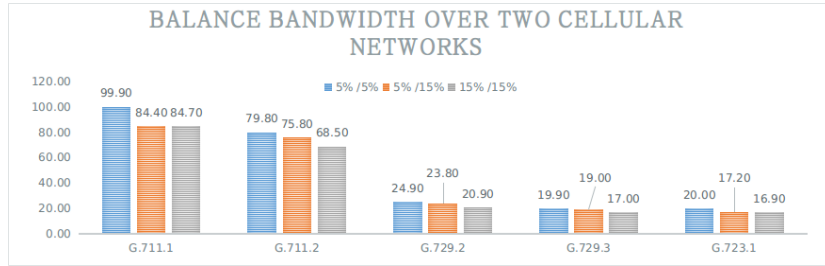
Figure3.43a shows the bandwidth for one cellular and one satellite networks of Basic scheduler. Comparing with last scenario, we find that because the Basic repeat the packets twice during the transmission, the 5%/15% and 15%/5% have the similar performances. The total amount of packets received are more or less the same.

Figure3.43b is the receiving bandwidth of Balance, most of the codecs have more bandwidth on 5% / 15% than 15% / 5% because the packets are sent over the faster link but may not be the less loss link.

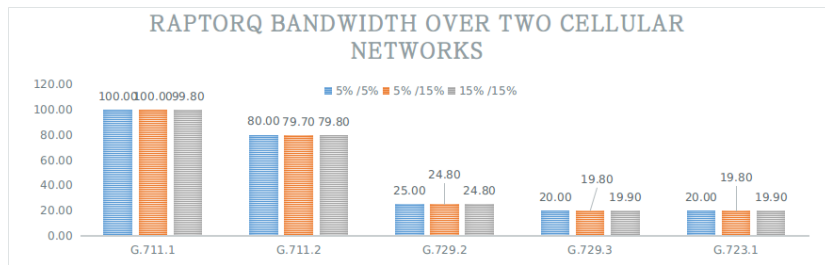
RaptorQ (Figure3.43c), as we have analyzed, can recover the lost packets by receiving those overhead symbols. We have got to know that the RaptorQ depends more on the faster subflow as well. For this reason, the RaptorQ gets lest bandwidth on the channel loss rate 15%/5% than 5% /15%.



(a) Receiving bandwidth with two cellular networks by Basic scheduler



(b) Receiving bandwidth with two cellular networks by Balance scheduler

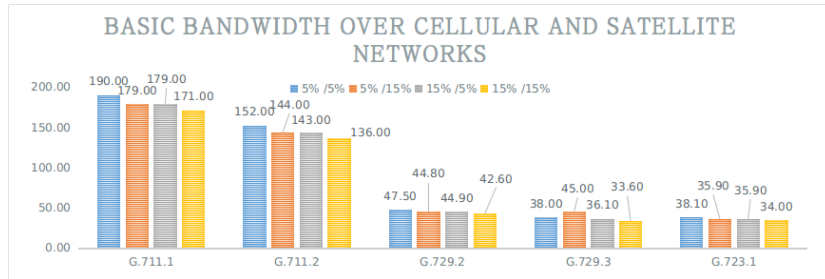


(c) Receiving bandwidth with two cellular networks by RaptorQ scheduler

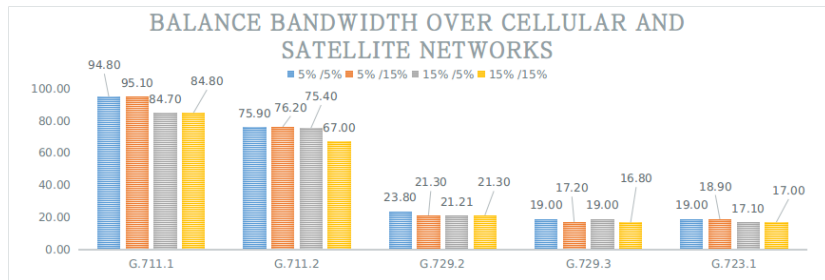
Figure 3.42: Receiving bandwidth with two cellular networks

2. Analysis of jitter

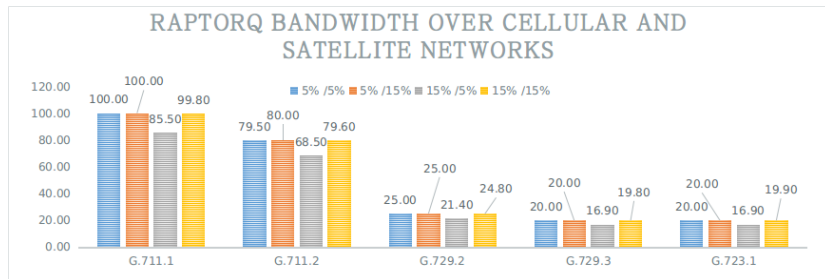
The average jitter for two cellular networks is shown in Figure3.44. Figure3.44a counts the average of different packet loss rate for each codec type. So we call it "the average jitter with different channel loss rate". Meanwhile Figure3.44b is the average for all kinds of codecs for diverse channel loss rate, for this reason, we call Figure3.44b the average jitter with different codecs. In Figure3.44 RaptorQ always has the largest jitter (except G.711.1, in Figure3.44a). Because RaptorQ always gathers more packets together to encode and decode at one time. This instinct will introduce some more jitter on the receiving side. Basic has the most stable jitter performance, since the packet is sent repeat on two subflows and the subflows have the same delay and jitter performances. Balance makes the



(a) Receiving bandwidth with one cellular and one satellite networks by Basic scheduler



(b) Receiving bandwidth with one cellular and one satellite networks by Balance scheduler

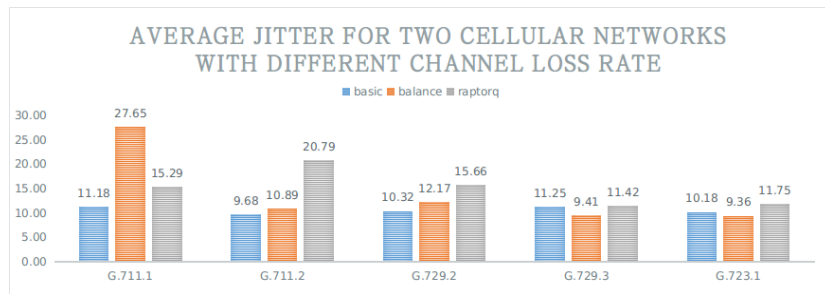


(c) Receiving bandwidth with one cellular and one satellite networks by RaptorQ scheduler

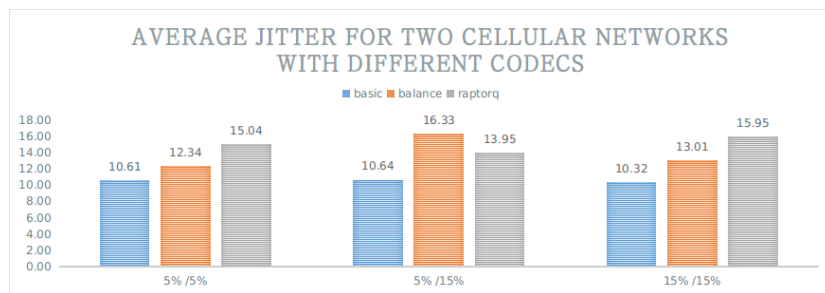
Figure 3.43: Receiving bandwidth with one cellular and one satellite networks

selection of subflow according to the prior delay testing, and the current subflow used may not be the faster one. Thus, comparing with other two scheduler it changes more its jitter with different codecs.

Figure3.45 depicts the average jitter for one cellular and one satellite network, the name principle we have explained before with Figure3.44. As we explained before, the Basic fairly sends on both subflows, so it has the average jitter around 200ms. Balance decides the subflow by prior delay knowledge, and under this scenario, the faster one is clear to be the cellular network, so that Balance sends it packets continuously on the cellular link. However, Balance sometimes needs



(a) Average jitter for two cellular networks with different channel loss rate



(b) Average jitter for two cellular networks with different codes

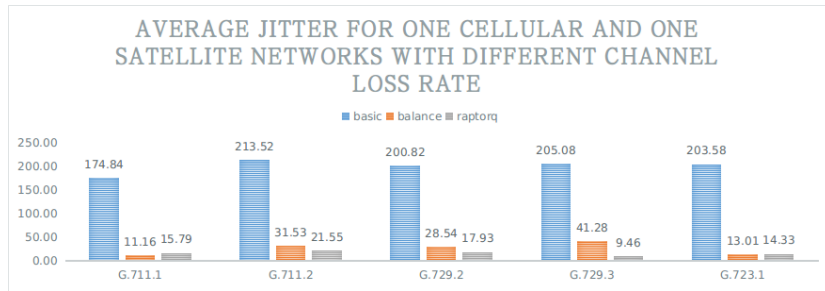
Figure 3.44: Average jitter for two cellular networks

to send prob packets over satellite link to keep this link alive. RaptorQ mainly depends on the faster link, and its jitter is more stable than other two.

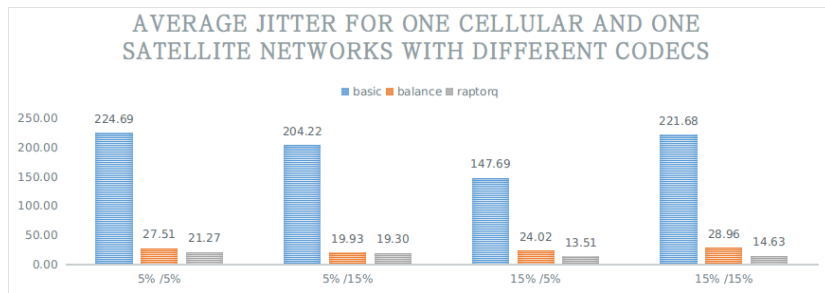
3. Packets loss rate

The packet loss rates for two cellular networks are shown in Figure3.46. Generally speaking, RaptorQ has the best performance on the packet loss rate, while Balance has the worst performance on this issue. This results match with what we expect. Because Balance has no recover strategy for the lost packet, but RaptorQ can decode with only a certain subset of the symbols. Figure3.46b says that for Basic and Balance, with the increase rate of channel loss, the packet loss rate increases at the same time.

Figure3.47 show the packet loss rate of the cellular and satellite networks. We got the same conclusion as before. But the other thing we should mention is that for Basic and Balance schedulers, the increasing of channel delay increases the packet loss rate, too. However, RaptorQ gets some similar results for two scenarios. We can suggest that when the channel condition is not good enough, RaptorQ can

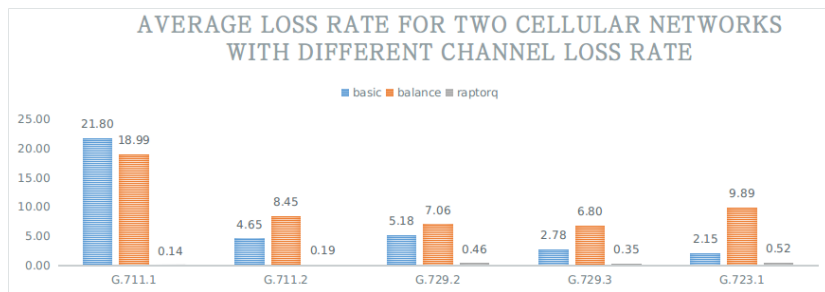


(a) Average jitter for one cellular and one satellite networks with different channel loss rate

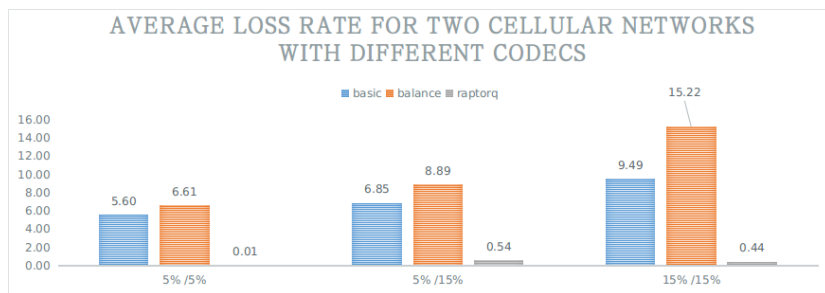


(b) Average jitter for one cellular and one satellite networks with different codes

Figure 3.45: Average jitter for one cellular and one satellite networks

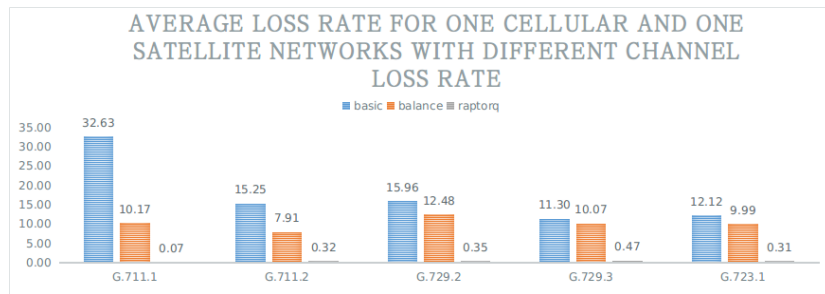


(a) Average loss rate for two cellular networks with different channel loss rates

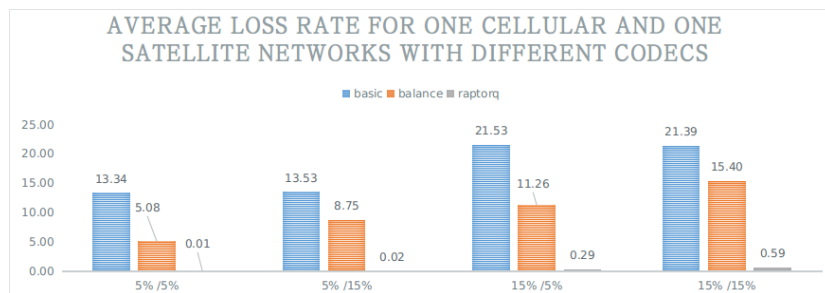


(b) Average loss rates for two cellular networks with different codecs

Figure 3.46: Average loss rate for two cellular networks



(a) Average loss rates for one cellular and one satellite networks with different channel loss rates



(b) Average loss rates for one cellular and one satellite networks with different codecs

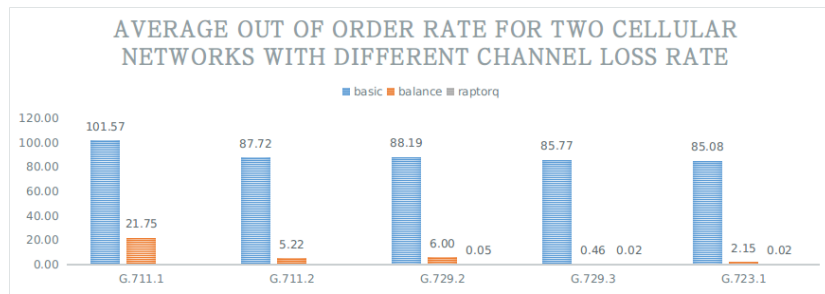
Figure 3.47: Average loss rate for one cellular and one satellite networks

be consider as a good solution to get the higher reliability.

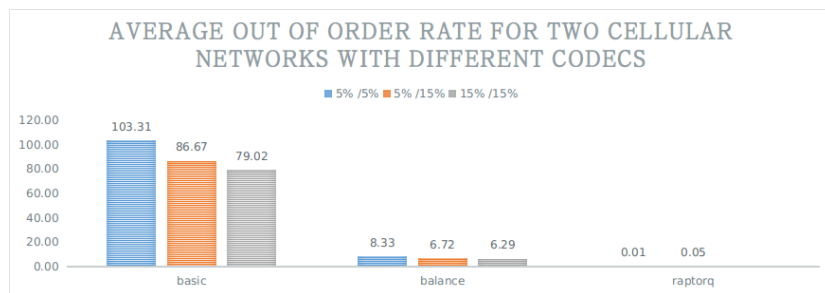
4. Out of order rate

Figure3.48 is the out of order rates for the two cellular networks scenario. The differences between three schedulers are very big. The Basic has a serious problem on the out of order rates. In Figure3.48a G.711.1 has more than 100% of the out of order rates, and other codec has around 85% of the out of order rates. This because Basic duplicates all packets and the software considers all duplicated packets as the out of order packets. Balance has less problem than Basic. But sometimes the out of order happens as well. For Balance, the subflow selection is based on the prior delay feedback, the channel condition may have changed for the current packet. RaptorQ does not have this problem. Since RaptorQ gathers several packets together to encode at one time, and the decode procedure is regardless to the symbols order. Thus, the decoded packets are always in the correct order.

Figure3.48b talks about the disorder rate of different channel loss rate. It is



(a) Average out of order rates for two cellular networks with different channel loss rates



(b) Average out of order rate for two cellular networks with different codecs

Figure 3.48: Average out of order rate for two cellular networks

clear that for Basic and Balance, the out of order rates are decreasing with the increasing of packet loss rates, and the reason we have discussed before.

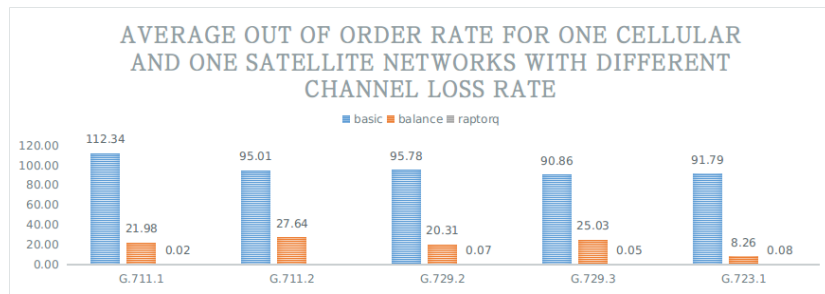
Figure 3.49 illustrates the behaviors on the second scenario. The out of order rates are increased by the increasing channel delay gap. Basic increases its rates from 101.52% to 112.34% for G.711.1 codec and other codecs from around 85% to 93%. RaptorQ still has a very good capacity on recover the out of order problem, and the out of order rates are always less than 0.1%.

3.5.6.3 Advantage and disadvantage summary for different schedulers

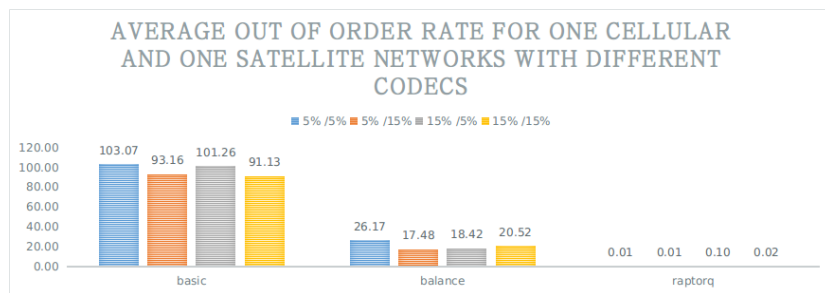
We have done all the tests for better understand the behaviors of RaptorQ and for comparing these three kinds of schedulers. According to the tests results, we conclude the advantages and disadvantages for each scheduler as follow.

1. Advantage

Basic scheduler has the simplest packets sending strategy. It is the easiest one to implement, and has the least processing time among three schedulers. It just



(a) Average out of order rate for one cellular and one satellite networks with different channel loss rate



(b) Average out of order rate for one cellular and one satellite networks with different codecs

Figure 3.49: Average out of order rate for one cellular and one satellite networks

duplicates each packet and sends the same packet on each subflow. This instinct helps Basic to recover some channel packet loss on the receiving side when the two channels have similar channel delay and jitter. Moreover, in two cellular networks scenario, Basic has a smaller jitter than the other two schedulers.

Balance scheduler sends the current packet by picking up the less delay link that is tested in every certain interval by the prob packet. It can achieve a high throughput about 7 mega bit per second. Comparing with Basic, Balance requests less bandwidth to send the packets, and when the channels have big gap on channel delay and jitter, Balance can have a stable performance on average jitter. Besides, since Balance always tries to send packets over faster link, there is less out of order problem than Basic.

RaptorQ scheduler collects more packets together and encode at one time. RaptorQ can decode the packet with a subset of sending symbols, thus with a proper setting of overhead, RaptorQ has no problem to recover the lost packets in any tested packet loss rates. Further more, the decoding procedure is regardless the

symbols order, this make RaptorQ tolerance more delay difference between two channels. As a result, in one cellular and one satellite networks scenario, RaptorQ has the outstanding performances on all the jitter, packet loss rate and out of order rate.

2. Disadvantage

Basic scheduler needs two times of the sending bandwidth, and it has heavy problems on the out of order rate. Not only for this, in the cellular and satellite networks scenario, the big delay difference also introduces a large jitter value and more packet loss rate for Basic scheduler.

Balance scheduler has no packet loss recover strategy, thus in two cellular network scenario, it has the largest packet loss rate. In fact, for one cellular and one satellite network, it has a bad performance on packet loss rate, too. Balance has around 7% and 20% of out of order rate for two scenarios, so we can say it has some problems on the out of order rate due to the different channels delay.

RaptorQ scheduler is very difficult to implement. The encode and decode procedure time is around 150ms. This consumes a lot of mouth to ear delay budget. Moreover the RaptorQ has a limitation on the transmission bandwidth. For the voice service there is no significant drawback by the limitation bandwidth, because the maximum bandwidth for VoIP is less than 100k/s. But if we shift his technology to other service, such as the real time video stream transmission. The bandwidth limitation could be one of the critical problems.

Chapter 4

On Board media quality assessment method

Nowadays, video camera starts to be used in the train positioning service in order to achieve a higher level of positioning accuracy. Meanwhile, with the increasing train speed, real time video monitor for supervising driving conditions, and video call service between driver and control center are also required in the train control system.

Up to now, our architectures only service for the data and voice transmission of railway control system, but in the future, they should bear the railway driving monitor or position reference video transmission and on board public services as well.

In addition, from the passenger's side, nearly every people has the experience of travel with train. Sometimes, it takes us several hours to arrive at our destination. What can we do to kill the boring on board time? Most of the passengers prefer to watch a video/film or play video games during their journey. However, this two entertainments always need the real time communication support over wireless network. In fact, the video quality has a big influence on the on board quality of experience.

In order to guarantee the transmitted video quality in control system as well as to achieve a high quality of experience for train passengers, train operators need some methods to monitor the real time transmitting video quality. Video quality assessment method is one of the most important monitor manners. The video quality assessment can be mainly separated in subjective and objective ways. For each single piece of

video, the subjective one needs involve in many observations. Thus, it is very difficult to implement. On the counterpart, the objective evaluation can be classified to full reference, half reference and no-reference assessment according to the amount of source video information presenting in the measurement.

In the railway using scenario, the video quality assessment method should be able to evaluate the real time video quality with as less original video information as possible. Because, on one side, during the video quality evaluation, it is difficult to have the original source video as reference; on the other side, attaching additional reference information to the transmitted video requests more bandwidth and increase the Internet traffic usage to the passenger.

To solve this problem, in this chapter, we will introduce one no-reference video quality assessment method by means of spatial and temporal degradations from the received video.

4.1 State of art studies of video quality assessment methods

No-Reference video quality assessment is a challenging task since the quality evaluation is not based on a priori knowledge of the original signal. we propose a No-Reference quality metric for IP-based video transmission. It is based on the evaluation of the artifacts introduced by H.264/AVC coding and by the impairments caused by the specific transmission system. In particular, estimated temporal and spatial impairments are pooled by means of an Additive Log-logistic Model. Performed experiments show the effectiveness of the proposed system in matching the subjective judgment. Furthermore, the collected scores have been compared with the state of art video quality metrics. The goal of objective image and video quality assessment research is to design quality metrics that can automatically predict perceived image and video quality.

The design of No-Reference (NR) metrics, is a challenging task and is broadly investigated in the literature. A neural network based Video Quality Metric (VQM), to evaluate the impact of temporal jerkiness due to frame freezing resulting from the

packet loss or late arrival has been presented in [111]. A NR-VQM dedicated to MPEG coded video has been proposed in [112]. In [113] a decision tree based impairments detection technique for high definition H.264/AVC encoded video is presented. Where, the impact of Packet Loss Rate (PLR) artifact is largely been considered. In [114], VQM based on genetic algorithm is presented by considering losses on the video content and other several parameters (i.e. number of consecutive slice drops, the percentage of picture's slices, etc.) which have been extracted from the received video stream. Authors in [115] proposes a analytical model for no-reference video quality assessment model called pulse intensity metric. The metric is designed by considering both video play-out rate and network throughput. A video quality metric based on the spatio-temporal natural scene statistics and motion coherency in the video scenes has been proposed in [116]. A VQM based on the statistical estimation of the Peak Signal-to-Noise Ratio (PSNR) of the coded transform coefficients is presented in [117] and the approach has been exploited for H.264/AVC encoded sequences in in [118]. The metric with the consideration of the characteristics of the human visual system estimating the coding error and the perceptual weighting of this error is presented in [117].

Moreover, in [119] the video quality is assessed by considering specific features of the H.264/AVC encoding such as blocking, blurring, and spatial activity. Those features have different impact on the perceived quality depending on the characteristics of the video itself such as low bit rate, the presence of blocking and/or blurring. According to these parameters the authors define four classes that are characterized by different weights for the considered features. As mentioned in [120], the categories of artifacts mainly affecting the perceived quality of a compressed video are block distortion (blocking artifacts introduced by block-based motion compensation and transform coding), jitter, and jerky/unnatural motion. In [121], the perceived annoyance of artifact combinations and the perceived strength of the respective coding artifacts are analyzed. In [122], the modeling of the distortion due to the lack of motion vectors, prediction residuals and temporal propagation in H.264/AVC encoded videos is performed.

Here our aim is to propose a VQM which could work in more generic scenario by considering the overall impact of different channel impairments and encoding artifacts. For the metric design correlation values between the successive frames, repeat line and

number of the broken blocks of the received video has been considered as a possible features which are resulting due to the transmission impairments and encoding artifact. Moreover, to estimate the quality score from the features a Additive Log-logistic Model (ALM) [123] has been used. To validate the results estimated from the proposed metric, a subjective study has been conducted. The performance of the proposed metric is also compared with other widely discussed metrics. The results of the analysis shows that the proposed metrics outperform than the considered no reference metrics.

4.2 Proposed Approach

The proposed metric is aimed at evaluating the impact on the perceived quality of the distortions caused by packet losses as well as by the presence of packets affected by errors. Since the overall effects depend on the lost information (e.g, motion vectors, Discrete Cosine Transform (DCT) coefficients, etc.) as well as on the concealment algorithms adopted, we first employ a set of detectors, each tuned to a specific type of impairment, in order to evaluate the strength and the length of each type of distortion. Then the partial distortions are combined into an overall distortion. Based on [123], in presence of multiple impairments, the **frame** quality score can be obtained by combining the partial distortions d_i , associated to each type of impairment, by exploiting an ALM model, defined as follows:

$$f_{frame}(d) = \frac{1}{1 + [\sum d_i^p]^{\frac{1}{p}}} \quad (4.1)$$

where $d = (d_1 \ d_2 \ \dots \ d_N)^T$, p is constant, which is obtained by linear regression during training section. The video quality is considered as the average of each frame, show in equation (4.2):

$$Score_{video} = \frac{\sum_{i=1}^{i=N} f_{frame}}{N} \quad (4.2)$$

where N is total number of frames.

Impairment detection and visibility are evaluated on the basis of indicators extracted from some features of the rendered video, mapping of those indicators to partial distortions has been performed by matching the related ALM to experimental observations providing subjective quality assessment as in [123]. Here, we focus on two main impairments: broken blocks and repeated lines, shown in Fig 4.1. It worth to mentioned that we consider three color components Y, C_b, C_r for each frame, and we calculate partial distortions d_i for three components, respectively. In the following, we take out one component as the example. First, we describe how to obtain the correlation coefficient between consecutive frames. Then, we present the information about temporal and spatial degradation, which contribute to the different types of partial distortions. Finally, we discuss the generation of partial distortions and the combination of overall distortion vectors.

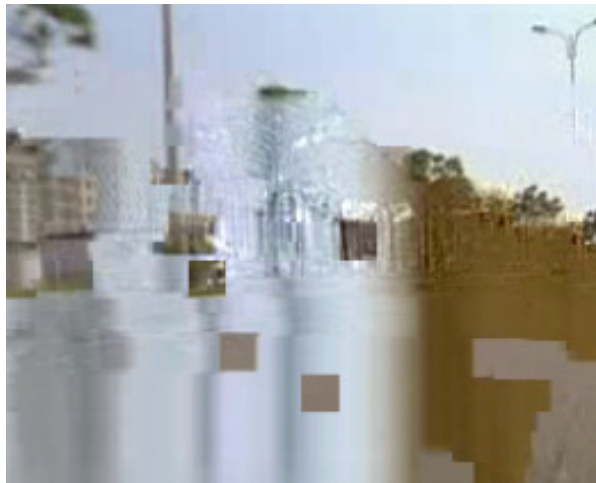


Figure 4.1: Two main artifacts appear in the frame, they are broken blocks and repeat lines.

In order to detect and classify impairments at the output of the video player, the received video sequence is first partitioned into static and dynamical shots, according to the amount of changes between consecutive frames. Next, the static shots are evaluated to detect if the small amount of changes corresponds to a real static scene or to the *freeze* of entire frames or part of them. The freezing effect is due to the decoder repetition of

portions or whole of the last correctly decoded frame until the next correct information is available. In this case, the perceived quality of the played video will depend on the dynamics of the scene. In fact, although only error free frames are played, the motion of objects composing the scene may appear unnatural, due to its stepwise behavior (jerkiness effect). The dropping mechanism can also be caused by a playback system that is not fast enough to decode and display each video frame at full nominal speed. It is worth noticing that the same experience is perceived in presence of *frame freezing* artifacts or by repetition of the same frame. At the same time, the dynamical shots are tested to verify the presence of isolated and clustered corrupted blocks.

4.2.1 Global temporal analysis

Let F_k be the k^{th} generic frame of the video sequence. It can be partitioned in $N_r \times N_c$ blocks $B_k^{(i,j)}$ of $r \times c$ pixels with top-left corner located in (i, j) . Let $\Delta F_k = F_k - \bar{F}_k$ and $\Delta B_k^{(i,j)} = B_k^{(i,j)} - \bar{B}_k^{(i,j)}$ denote the deviation of the luminance of the k^{th} frame and of the block $B_k^{(i,j)}$ from the corresponding mean values.

The normalized inter-frame correlation coefficient ρ_k between the k^{th} and the $(k-1)^{th}$ frames is defined as:

$$\rho_k = \frac{\langle \Delta \mathbf{F}_k, \Delta \mathbf{F}_{k-1} \rangle}{\|\Delta \mathbf{F}_k\|_{L_2} \|\Delta \mathbf{F}_{k-1}\|_{L_2}}. \quad (4.3)$$

The inter-block correlation $\rho_k^{B(i,j)}$ can be computed as:

$$\rho_k^{B(i,j)} = \frac{\langle \Delta \mathbf{B}_k^{(i,j)}, \Delta \mathbf{B}_{k-1}^{(i,j)} \rangle}{\|\Delta \mathbf{B}_k^{(i,j)}\|_{L_2} \|\Delta \mathbf{B}_{k-1}^{(i,j)}\|_{L_2}}. \quad (4.4)$$

where $\langle \bullet, \bullet \rangle$ denotes the inner product and $\|\bullet\|_{L_2}$ the L_2 -norm.

The inter-frame correlation presents a spiky behavior with values close to one in correspondence of *frozen* frames. However, such a behavior is not sufficient to identify a partial or total frame loss, since in the case of static scenes, consecutive frames present a high inter-frame correlation.

4.2.2 Local temporal analysis

The local temporal analysis is performed in two stages. The aim of the first one is to identify and to extract from each frame the blocks that are potentially affected by artifacts. This analysis is performed by classifying the blocks as: with medium content variations, affected by large temporal variations, with small content variations, depending on their temporal correlation $\rho_k^{B(i,j)}$.

The corresponding temporal variability map $\Gamma_k^V = \{\Gamma_k^{VB(i,j)}\}$ is computed by comparing the inter-frame correlation of each block with two thresholds θ_l and θ_h :

$$\Gamma_k^{VB(i,j)} = \begin{cases} 1, & \text{if } \rho_k^{B(i,j)} < \theta_l \\ 0, & \text{if } \theta_l \leq \rho_k^{B(i,j)} \leq \theta_h \\ 2, & \text{if } \rho_k^{B(i,j)} > \theta_h \end{cases} \quad (4.5)$$

where the two thresholds, θ_l and θ_h have been selected in order to grant $|P_{fa} - P_{md}| < \varepsilon_1$, where P_{fa} is the Probability of False Alarm, P_{md} is the Probability of Missed Detection, and ε_1 has been experimentally determined, during the training phase, by comparing the performances achieved by the temporal analysis algorithm with the scores provided by a group of video quality experts in an informal subjective test.

4.2.3 Spatial analysis

The blocks classified as potentially affected by packet loss undergo a spatial analysis that is performed by analyzing the presence of static regions, the presence of repeated lines and a check on the consistency of edges. In this work we focus on the detection of static regions detection for verifying whether a high correlation between the current block $\mathbf{B}_k^{(i,j)}$ and the previous one $\mathbf{B}_{k-1}^{(i,j)}$ is due to the loss of a single or multiple blocks or to the presence of a static region. To this aim, for each block with $\Gamma_k^{VB(i,j)} = 2$, it is checked if at least v among the surrounding blocks present a strong temporal correlation. In case of positive result, the block is classified as belonging to a static region and its potential distortion index $\Gamma_k^{CB(i,j)}$ is set to zero. The parameter v has been identified by experimental test. That is:

$$\Gamma_k^{CB(i,j)} = \begin{cases} 0 & \text{if } |\varsigma| > \nu \\ \Gamma_k^{VB(i,j)} & \text{otherwise} \end{cases} \quad (4.6)$$

where ς equals to how many surrounding blocks of $B_k^{(i,j)}$ with $\Gamma_k^{VB^{(p,q)}} = 2$.

Let E_l and E_r be the L_1 norms of the vertical edges respectively on the left and on the right boundary of the block, and with A_c , A_l and A_r the average values of the L_1 norms of the vertical edges inside the current block and of the left and right adjacent blocks, shown in Figure 4.2. A block with $\Gamma_k^{CB^{(i,j)}} \neq 0$ is classified as affected by visible distortion if:

$$\left| E_l - \frac{(A_c + A_l)}{2} \right| > \theta \quad \text{or} \quad \left| E_r - \frac{(A_c + A_r)}{2} \right| > \theta \quad (4.7)$$

where the threshold θ has been defined on the basis of experimental trials. In particular it corresponds to JND (Just Noticeable Difference) collected evaluated for the 90% of subjects. The same procedure is then applied to the horizontal direction. If the block edges are consistent (i.e. no visible distortion has been detected along horizontal and vertical directions) $\Gamma_k^{CB^{(i,j)}}$ is reset to 0.

Let TOT_k be the total broken block numbers in the k^{th} frame. TOT_k is presented by the total amount of blocks where $\Gamma_k^{CB^{(i,j)}} \neq 0$.

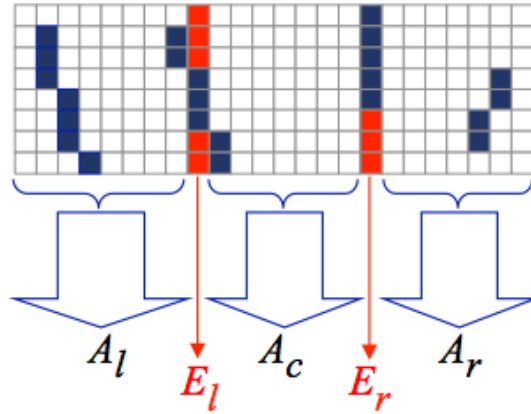


Figure 4.2: E_l and E_r are the vertical edges on the left and on the right boundary of the block, and with A_c , A_l and A_r are the average values inside the current block and of the left and right adjacent blocks.

Let $f_k[i]$ be the i^{th} row of the k^{th} frame. Starting from the m^{th} line of the frame, the L_1 -norm of the horizontal gradient component is computed and compared to a threshold λ_H . If $\|\nabla f_k[i]\|_{L_1} > \lambda_H$, the procedure is repeated on the previous line ($i - 1$) to check if consecutive lines are identical by comparing the L_1 -norm of their difference

with a threshold λ_V

$$\|f_k[i] - f_k[i-1]\|_{L_1} < \lambda_V. \quad (4.8)$$

This procedure is iterated until the test fails, thus meaning that there is a different information carried out by consecutive lines. After the repeated lines test has been performed, a binary spatial degradation map, $\Gamma_k^{RLB^{(i,j)}}$ of $[0,1]$ entries, is created where 1 corresponds to a block belonging to a vertical stripes region and 0 otherwise. The two thresholds, λ_V and λ_H have been set after a training process with a pool of experts trying to match the subjective impression of repeated lines.

Let RP_k be the total block numbers in the vertical stripes region of k^{th} frame. RP_k is counted by the total amount of blocks where $\Gamma_k^{RLB^{(i,j)}} \neq 0$.

The evaluation of the video quality metric NR_{ALM} is based on pooling together the total broken block numbers TOT_k , the total repeated line block numbers RP_k and the inter frame correlation ρ_k . Since the impairments influence not only the current frame, but also the consequent decoding frames, we joint the correlation coefficient ρ_k to both two types of artifacts. In order to balance the affect between ρ_k and two artifacts, we use logarithmic values to all the three metrics and considered only the positive value. In particular, d_1 presents the distortions caused by broken blocks, as:

$$d_1 = a_1 \log(\rho_k + 1)^{m_1} \log(TOT_k + 1)^{n_1} \quad (4.9)$$

d_2 presents the distortions caused by repeat lines, as:

$$d_2 = a_2 \log(\rho_k + 1)^{m_2} \log(RP_k + 1)^{n_2} \quad (4.10)$$

where $a_1, a_2, m_1, m_2, n_1, n_2$ are constant, which are obtained by linear regression during the training process. Respect to equation (4.1), $d = (d_{1Y} \ d_{2Y} \ d_{1C_b} \ d_{2C_b} \ d_{1C_r} \ d_{2C_r})^T$

4.3 Results and Discussion

To evaluate the effectiveness of the proposed method a widely used and publicly available video quality database [124] has been considered. Where 12 heterogeneous source sequences with different frame size (CIF and 4CIF) and six Packet Loss Rate (PLR) values (0.1%, 0.4%, 1%, 3%, 5%, and 10%) have been simulated and two channel

realizations have been selected for each PLR value. Accordingly, a total number of 156 test sequences and their corresponding MOS scores have been considered.

For each test video sequence the Y , C_b and C_r components have been considered. Impairment detection techniques have been applied for each component and their corresponding features (correlation between the successive frames, repeat line and broken block block) have been computed frame by frame basis and finally the average values have been considered for the video. Finally, to get the final quality score the Additive Log-logistic Model (ALM) [123] has been used and the coefficient of the ALM has been computed by using linear regression with the help of the training videos with their corresponding subjective scores. Out of 12 source video sequences which has been considered during the subjective study, 6 (3CIF and 34CIF) are used for training and 6 (3CIF and 34CIF) are used for testing. Performance of the proposed NR_{ALM} metric has been compared with four no-reference metrics (Naturalness Image Quality Evaluator (NIQE)[125], Blind Image Quality Index (BIQI)[126], Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE)[127] and BLind Image Integrity Notator using DCT Statistics - II (BLIINDS-II) Index [128]) and three full reference metrics (Peak Signal to Noise Ratio (PSNR), Structural SIMilarity (SSIM) [129], and Pixel domain version of Visual Information Fidelity (PVIF) [130]) by considering the video with two different frame sizes: CIF and 4CIF. The quality score is computed for every frame and finally the average score is used to represent the video quality.

Table 4.1: Performance comparison for CIF and 4CIF video sequences.

Metrics	CIF		4CIF	
	PLCC	SROCC	PLCC	SROCC
BIQI	0.3418	0.3109	0.0718	0.0747
NIQE	-0.7638	-0.7326	-0.0538	-0.3628
BRISQUE	-0.0954	-0.1012	0.5006	-0.3901
BLIINDS-II	0.4851	0.4610	0.6656	0.7050
NR_{ALM}	0.8464	0.9040	0.8779	0.9103
SSIM	0.8925	0.9725	0.9238	0.9674
PSNR	0.9178	0.9399	0.6286	0.9397
PVIF	0.9017	0.9723	0.8266	0.9510

To compare the performance of the metrics the Spearman Rank Order Correlation Coefficient (SROCCs) and Pearson's Linear Correlation Coefficient (PLCCs) between subjective and predicted scores have been computed. From Table 4.1, we can clearly notice that among all the considered no reference metrics: BIQI, NIQE, BRISQUE, BLIINDS-II and NR_{ALM} , the proposed NR_{ALM} has the highest value of PLCCs and SROCCs for both of the considered CIF and 4CIF video sequences. This is due to the fact that the proposed model not only consider the impact of the impairments as a broken block and repeat line, but also the correlation between the successive frames at the receiver sides. Furthermore, it is also notice that the performance of the NR_{ALM} metric measured in terms of PLCCs and SROCCs has very close score with widely used full reference metricises SSIM, PSNR and PVIF. From these results we can conclude that the proposed NR_{ALM} metric perform better than other considered no reference matrices and also the predicted score is also very close to full reference matrices. This means that the proposed no-reference NR_{ALM} performs as close as widely used full reference metrics.

4.4 Conclusion

Our No Reference video quality metric NR_{ALM} for IP based video transmission has been introduced in this chapter. It is based on the estimation of spatial and temporal degradations from the received video. Prediction video quality score is computed via pooling all features by means of additive log-logistic model. The performance of the metric has been compared with other widely discussed no-reference and full reference matrices with the help of CIF and 4CIF video sequences with their corresponding MOS score. The result shows that the proposed no-reference metric NR_{ALM} perform better than other considered no-reference metrics and also perform very close to widely used and discussed full-reference metric.

Conclusion and Future Development

In this thesis, we proposed two communication solutions for railway using scenario and a no reference video quality assessment method. These two communication solutions are *MPTCP* solution and *MPUDP* solution, which service for the data and voice transmission, respectively. Both these two architectures are based on the best effort networks and satellite network.

The MPTCP (multipath TCP) protocol has been adopted into the *MPTCP* solution. In this solution, we give the follow three contributions: the new communication architecture for the train signaling links, the subflows adding and dropping strategies, and the MPTCP protocol performances study based on the railway scenarios. On the target of supporting the VoIP service on railway, our *MPUDP* solution have three main contributions: the implementation of the MP-UDP software merging with RaptorQ protocol, the realization of three kinds of schedulers: Basic, Balance and RaptorQ, and the performance studies on RaptorQ with UDP protocol and the comparison of three kinds of schedulers.

Till now, we have carried out the solution for both the data and real time voice services for the ERTMS/ETCS system. However, along with the increasing of safety level in ERTMS system, the railway control communication system will not satisfy with supporting only real time voice service, also requires the real time video service, such as the real time driving monitor. Hence, our future work will focus on improving of MPUDP throughput, and try to realize the real time video service supporting, too.

No reference video quality assessment method helps train control system to guarantee the video call and real time monitor video quality. It can help the train operator to know if their service can meet the passengers requirement or not as well. It is based

on the estimation of spatial and temporal degradations from the received video. The performance of the metric has been compared with other widely discussed no-reference and full reference matrices with the help of CIF and 4CIF video sequences with their corresponding MOS score. The result shows that the proposed no-reference metric NR_{ALM} perform better than other considered no-reference metrics.

Bibliography

- [1] *European train control system*. 2016. URL: <http://en.wikipedia.org/wiki/ETCS>.
- [2] H. Hofestadt. “GSM-R: global system for mobile radio communications for railways”. In: *International Conference on Electric Railways in a United Europe, Amsterdam* (1995), pp. 111–115.
- [3] *European rail traffic management system*. 2016. URL: <http://en.wikipedia.org/wiki/ERTMS>.
- [4] S. Dhahbi et al. “Study of the high speed trains positioning system: European signaling system ERTMS/ETCS”. In: *Logistics, 2011 4th International Conference* (May 2011), pp. 468–473.
- [5] S. Hayat. “Modeling of new driver assistance system for dysfunction of the signaling system ERTMS/ETCS”. In: *Industrial Engineering and Systems Management (IESM), Proceedings of 2013 International Conference* (Oct. 2013).
- [6] International Union of Railways. *Overview of the implementation*. 2013. URL: <http://www.uic.org>.
- [7] P. Winter et al. “Compendium on ERTMS”. In: *Hamburg, Germany: DW Media Group GmbH-Eurailpress* (2009).
- [8] S. Wei et al. “Research and analysis of ETCS controlling curves model”. In: *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference 2* (Aug. 2010), pp. 178–181.

-
- [9] R. Sharma and R. Lourde. “Crosstalk reduction in balise and infill loops in automatic train control”. In: *Advances in Sensors and Interface IWASI 2nd International Workshop* (June 2007), pp. 1–6.
- [10] Y. Liu et al. “Formal modeling and verification of RBC handover of ETCS using differential dynamic logic”. In: *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium* (Mar. 2011), pp. 67–72.
- [11] UNISIG. “EuroRadio FIS:SUBSET-037”. In: *European Union Agency for Railways Document Register* (Dec. 2015).
- [12] S. Abed. “European rail traffic management system - an overview”. In: *Energy, Power and Control (EPC-IQ), 2010 1st International Conference* (Dec. 2010), pp. 173–180.
- [13] A. Sniady and J. Soler. “An overview of GSM-R technology and its shortcomings”. In: *ITS Telecommunications (ITST), 2012 12th International Conference* (Nov. 2012), pp. 626–629.
- [14] A. Sniady and J. Soler. “LTE for railways: Impact on performance of ETCS railway signaling”. In: *Vehicular Technology Magazine, IEEE (Issue: 2) 9* (June 2014), pp. 69–77.
- [15] L. Zhao, X. Chen, and J. Ding. “Interference clearance process of gsm-r network in china”. In: *Mechanical and Electronics Engineering (ICMEE), 2010 2nd International Conference 1* (Aug. 2010), pp. 424–428.
- [16] S. Ruesche, J. Steuer, and K. Jobmann. “The european switch”. In: *Vehicular Technology Magazine, IEEE 3* (2008).
- [17] Civity Management Consultants. “European Benchmarking of the costs, performance and revenues of GB TOCs Final Report”. In: *Prepared for the Office of Rail Regulation, Hamburg* (Nov. 2012).
- [18] Teodor Gradinariu, Burkhard Stadlmann, and Ioan Nodea. “Satellite navigation traffic control system for low traffic lines Actual status and future deployment in Romania”. In: *12th UIC ERTMS World Conference* (Mar. 2016).

-
- [19] A. Basili et al. “A roadmap for the adoption of space assets for train control systems: The Test Site in Sardinia”. In: *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL), Rome (2012)*, pp. 1–6.
- [20] F. Rispoli et al. “Recent progress in application of GNSS and advanced communications for railway signaling”. In: *Radioelektronika (RADIOELEKTRONIKA), 2013 23rd International Conference, Pardubice (2013)*, pp. 13–22.
- [21] R. He et al. “A Standardized Path Loss Model for the GSM-Railway based High-Speed Railway Communication Systems”. In: *Proc. of Vehicular Technology Conference (VTC Spring) (May 2014)*, pp. 1–5.
- [22] A.Sniady and J.Soler. “Impact of the traffic load on performance of an alternative LTE railway communication network”. In: *Proc. of 13th International Conference on ITS Telecommunications (ITST) (Nov. 2013)*, pp. 396–401.
- [23] J. Moreno et al. “A survey on future railway radio communications services: challenges and opportunities”. In: *Communications Magazine, IEEE* 53.10 (Oct. 2015), pp. 62–68.
- [24] A. Diaz Zayas, C.A. Garcia Perez, and P. Merino Gomez. “Third-Generation Partnership Project Standards: For Delivery of Critical Communications for Railways”. In: *Proc. of IEEE Vehicular Technology Magazine* 9.2 (June 2014), pp. 58–68.
- [25] J. H. Baek et al. “A LTE wireless communication interface test for on-board oriented train control system field test”. In: *Proc. of International Conference on Information and Communication Technology Convergence ICTC (Oct. 2014)*, pp. 22–24.
- [26] A. Sniady and J. Soler. “LTE for Railways: Impact on Performance of ETCS Railway Signaling”. In: *IEEE Vehicular Technology Magazine* 9.2 (June 2014), pp. 69–77.
- [27] A. Sniady and J. Soler. “Capacity gain with an alternative LTE railway communication network”. In: *Proc. of 7th International Workshop on Communication Technologies for Vehicles (Oct. 2014)*, pp. 54–58.

- [28] L. Yan and X. Fang. “Decoupled wireless network architecture for highspeed railway”. In: *Proc. of International Workshop on High Mobility Wireless Communications (HMWC)* (Nov. 2013), pp. 96–100.
- [29] et al. Lei Lei. “Stochastic Delay Analysis for Train Control Services in Next-Generation High-Speed Railway Communications System”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.1 (2016), pp. 48–64.
- [30] et al. J. J. Cheng. “Routing in Internet of Vehicles: A Review”. In: *IEEE Trans. on Intelligent Transportation Systems* 16.5 (Oct. 2015), pp. 2339–2352.
- [31] M. Aguado et al. “WiMax on Rails”. In: *IEEE Vehicular Technology Magazine* 3.3 (Nov. 2008), pp. 47–56.
- [32] M. Aguado et al. “Simulation framework for performance evaluation of broadband communication architectures for next generation railway communication services”. In: *Proc. of 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST)* (Oct. 2009), pp. 453–457.
- [33] M. Aguado et al. “4G Communication Technologies for Train to Ground Communication Services: LTE versus WiMAX, a simulation study”. In: *Proc. of WCRR 2011* (2011).
- [34] J. Y. Zhang et al. “A Multi-Mode Multi-Band and Multi-System-Based Access Architecture for High-Speed Railways”. In: *Proc. of IEEE 72nd Vehicular Technology Conference* (Sept. 2010), pp. 6–9.
- [35] Yanbo Zhao and Petros Ioannou. “Positive Train Control With Dynamic Headway Based on an Active Communication System”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.6 (2015), pp. 3095–3103.
- [36] et al. Y. Liu. “Topology Discovery for Linear Wireless Networks with Application to Train Backbone Inauguration”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.8 (Aug. 2016), pp. 2159–2170.
- [37] E. Del Signore et al. “On the Suitability of Public Mobile Networks for Supporting Train Control/Management Systems”. In: *IEEE Wireless Communications and Networking Conference (WCNC)* (Apr. 2014), pp. 3302–3307.

- [38] H. Zhengqun et al. “Railway Safety Monitoring System Based on CAPS”. In: *2013 Fifth International Conference on Measuring Technology and Mechatronics Automation, Hong Kong* (2013), pp. 838–840.
- [39] Y. Liu, A. Neri, and A. Ruggeri. “Integration of PLMN and Satellite Networks for Train Control and Traffic management via MPTCP”. In: *14th International Conference on ITS Telecommunications* (Dec. 2015), pp. 2–4.
- [40] C. Huitema. “Multi-homed TCP draft-huitema-multi-homed-0”. In: *IETF* (May 1995).
- [41] A. Ford et al. “TCP extensions for multipath operation with multiple addresses RFC 6824”. In: *IETF* (Jan. 2013).
- [42] A. Kostopoulos et al. “Towards Multipath TCP Adoption: Challenges and Opportunities”. In: *6th EURO-NF Next Generation Internet (NGI)* (June 2010), pp. 1–8.
- [43] A. Singh et al. “Enhancing fairness and congestion control in multipath TCP”. In: *Wireless and Mobile Networking Conference (WMNC)* (Apr. 2013), pp. 1–8.
- [44] A. Singh, A. Kongseng Mei Xiang, and C. Goerg. “Performance and Fairness Comparison of Extensions to Dynamic Window Coupling for Multipath TCP”. In: *Wireless and Mobile Networking Conference (WMNC)* (July 2013), pp. 947–952.
- [45] J. Zhao et al. “A fluid model of multipath TCP algorithm: Fairness design with congestion balancing”. In: *IEEE International Conference on Communications (ICC)* (June 2015), pp. 6965–6970.
- [46] Q. Peng et al. “Multipath TCP: Analysis, Design, and Implementation”. In: *Networking, IEEE/ACM Transactions on* PP.99 (Dec. 2014), p. 1.
- [47] D. Ni et al. “Fine-grained Forward Prediction based Dynamic Packet Scheduling Mechanism for multipath TCP in lossy networks”. In: *Computer Communication and Networks (ICCCN)* (Aug. 2014), pp. 1–7.
- [48] C. Raiciu, M. Handley, and D. Wischike. “Coupled congestion control for multipath transport protocols RFC 6356”. In: *IETF* (Mar. 2013).

- [49] H.A.Kim, B.hwanOh, and J.Lee. “Improvement of MPTCP performance in heterogeneous network using packet scheduling mechanism”. In: *Communications (APCC), 2012 18th Asia-Pacific Conference* (2012).
- [50] A. Alheid, D. Kaleshi, and A. Doufexi. “An analysis of the impact of out-of-order recovery algorithms on MPTCP throughput”. In: *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference* (2014).
- [51] *Opensignal website*. URL: <http://opensignal.com/coverage-maps/Italy/>.
- [52] B. Arzani et al. “Impact of Path Characteristics and Scheduling Policies on MPTCP Performance”. In: *28th International Conference on Advanced Information Networking and Applications Workshops WAINA* (May 2014), pp. 743–748.
- [53] B. Arzani et al. “Concise Paper: Deconstructing MPTCP Performance”. In: *IEEE 22nd International Conference on Network Protocols (ICNP)* (Oct. 2014), pp. 269–274.
- [54] J. Hwang and J. Yoo. “Packet Scheduling for Multipath TCP”. In: *Seventh International Conference on Ubiquitous and Future Networks (ICUFN)* (July 2015), pp. 177–179.
- [55] K. Krupakaran, A.P. Sridharan, and S.M. Venkatesan. “aravind-mptcp-optimized-subflows-00”. In: *MPTCP Working Group Internet Draft* (Apr. 2015).
- [56] M. Coudron et al. “Cross-layer Cooperation to Boost Multipath TCP Performance in Cloud Networks”. In: *IEEE 2nd International Conference on Cloud Networking (CloudNet)* (Nov. 2013).
- [57] M. Coudron, S. Secci, and G. Pujolle. “Augmented Multipath TCP Communications”. In: *21st IEEE International Conference on Network Protocols (ICNP)* (Oct. 2013), pp. 1–2.
- [58] S. Hassan Baidya and R. Prakash. “Improving the performance of Multipath TCP over Heterogeneous Paths using Slow Path Adaptation”. In: *IEEE International Conference on Communications (ICC)* (June 2014), pp. 3222–3227.

-
- [59] M. Boucadair and C. Jacquenet. “draft-boucadair-mptcp-probe-subflow-00 - Probing MPTCP Subflows”. In: *MPTCP Working Group Internet Draft* (July 2015).
- [60] T. Inzerilli et al. “A Cross-Layer Location-Based Approach for Mobile-Controlled Connectivity”. In: *Hindawi Intl. Journal of Digital Multimedia Broadcasting* 2010 (2010), p. 13.
- [61] *VirtualBox website*: 2017. URL: <https://www.virtualbox.org/>.
- [62] *Icteam website*: 2017. URL: <http://multipath-tcp.org/pmwiki.php/Users/DoItYourself>.
- [63] *D-ITG website*: 2017. URL: <http://traffic.comics.unina.it/software/ITG/>.
- [64] *Ntpq website*: 2017. URL: <http://doc.ntp.org/4.1.0/ntpq.htm>.
- [65] *Wireshark website*: 2017. URL: <https://www.wireshark.org/>.
- [66] *3InSat Project*: 2017. URL: <https://artes-apps.esa.int/projects/3insat>.
- [67] Yi J. Liang, Eckehard G. Steinbach, and Bernd Girod. “Multi-stream voice over IP using packet path diversity”. In: *Multimedia Signal Processing, 2001 IEEE Fourth Workshop* (Oct. 2001).
- [68] J. G. Apostolopoulos. “Reliable video communication over lossy packet networks using multiple state encoding and path diversity”. In: *in Proceedings Visual Communication and Image Processing* (Jan. 2001).
- [69] Lucy Yong et al. “GRE-in-UDP Encapsulation draft-ietf-tsvwg-gre-in-udp-encap-11”. In: *Network Working Group Standard Track* (Mar. 2016).
- [70] Bela Almasi et al. “MPT-GRE in UDP based Multipath Communication Library, User Guide”. In: *Network Working Group Standard Track* (Sept. 2015).
- [71] Bela Almasi et al. “MPT: a Solution for Eliminating the Effect of Network Breakdowns in case of HD Video Stream Transmission”. In: *6th IEEE international conference on cognitive infocommunications* (Oct. 2015).

- [72] C. Bouras et al. “Enhancing reliable mobile multicasting with RaptorQ FEC”. In: *Computers and Communications (ISCC), 2012 IEEE Symposium* (2012), pp. 82–87.
- [73] U. Kumar and O. Oyman. “QoE evaluation for video streaming over eMBMS”. In: *Computing, Networking and Communications (ICNC), 2013 International Conference* (2013), pp. 555–559.
- [74] C. Bouras et al. “Evaluating RaptorQ FEC over 3GPP multicast services”. In: *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC), Limassol* (2012), pp. 257–262.
- [75] T. Mladenov, K. Kim, and S. Nooshabadi. “Forward error correction with RaptorQ Code on embedded systems”. In: *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), Seoul* (2011), pp. 1–4.
- [76] L. Hu, S. Nooshabadi, and T. Mladenov. “Forward error correction with RaptorQ code on GPU”. In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013), Beijing* (2013), pp. 281–284.
- [77] C. Bouras and N. Kanakis. “A competitive AL-FEC framework over mobile multicast delivery”. In: *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia* (2013), pp. 305–310.
- [78] C. Bouras et al. “Deploying AL-FEC with Online Algorithms”. In: *2013 Seventh International Conference on Next Generation Mobile Apps, Services and Technologies, Prague* (2013), pp. 175–180.
- [79] C. Bouras and N. Kanakis. “An adaptive weighted online AL-FEC algorithm over mobile multicast networks”. In: *2014 IEEE Wireless Communications and Networking Conference (WCNC), Istanbul* (2014), pp. 1579–1584.
- [80] C. Bouras and N. Kanakis. “Online AL-FEC protection over mobile unicast services”. In: *Networks and Communications (EuCNC), 2015 European Conference on, Paris* (2015), pp. 229–233.

-
- [81] A. Ando et al. “Rateless codes performance tests on terrestrial FSO time-correlated channel model”. In: *Optical Wireless Communications (IWOW), 2012 International Workshop on, Pisa* (2012), pp. 1–3.
- [82] R. Pernice et al. “Rateless codes mitigation technique in a turbulent indoor Free Space Optics link”. In: *Photonics Conference, 2014 Third Mediterranean, Trani* (2014), pp. 1–3.
- [83] R. Pernice et al. “Packet loss recovery in an indoor Free Space Optics link using rateless codes”. In: *2014 16th International Conference on Transparent Optical Networks (ICTON)* (2014), pp. 1–4.
- [84] A. Ando et al. “Fading mitigation coding techniques for space to ground free space optical communications”. In: *Euro Med Telco Conference (EMTC), 2014, Naples* (2014), pp. 1–5.
- [85] Riccardo Pernice et al. “Error mitigation using RaptorQ codes in an experimental indoor free space optical link under the influence of turbulence”. In: *IET Communications* 9.14 (2015), pp. 1800–1806.
- [86] R. Pernice et al. “Moderate-to-strong turbulence generation in a laboratory indoor Free Space Optics link and error mitigation via RaptorQ codes”. In: *2016 18th International Conference on Transparent Optical Networks (ICTON), Trento* (2016), pp. 1–4.
- [87] Y. Zhang et al. “RaptorQ code based adaptive real-time transmission scheme for M/LWD system”. In: *Wireless, Mobile and Multimedia Networks (ICWMMN 2013), 5th IET International Conference on, Beijing* (2013), pp. 141–144.
- [88] Yoonseok Heo et al. “True realtime multimedia streaming system based on MMT”. In: *Multimedia and Expo Workshops (ICMEW), 2015 IEEE International Conference on, Turin* (2015), pp. 1–3.
- [89] B. Bulut et al. “A raptor enabled data carousel for enhanced file delivery and QoS in 802.11 multicast networks”. In: *2015 IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, LA* (2015), pp. 2032–2037.

-
- [90] B. Bulut, A. Doufexi, and A. Nix. “An Adaptive Raptor Enabled Data Carousel for File Delivery in 802.11 Multicast Networks”. In: *2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA* (2015), pp. 1–7.
- [91] M. A. Mehmood, T. M. Jadoon, and N. M. Sheikh. “Assessment of VoIP quality over access networks”. In: *2005 1st IEEE and IFIP International Conference in Central Asia on Internet, Bishkek* (2005).
- [92] “Methods for subjective determination of transmission quality”. In: *ITU-T Rec. P. 800* (Aug. 1996).
- [93] “The E-model: a computational model for use in transmission planning”. In: *ITU-T Rec. G.107* (June 2015).
- [94] “One-way transmission time”. In: *ITU-T Rec. G.114* (May 2003).
- [95] Amit Chhabra and Dheerendra Singh. “Assessment of VoIP E-model over 802.11 wireless mesh network”. In: *2015 International Conference on Advances in Computer Engineering and Applications* (2015), pp. 856–860.
- [96] Y. Han et al. “Accuracy analysis on call quality assessments in voice over IP”. In: *6th Joint IFIP Wireless and Mobile Networking Conference (WMNC), Dubai* (2013), pp. 1–7.
- [97] “Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs”. In: *ITU-T Recommendation P.862* (2001).
- [98] “Subjective Test Methodology for Evaluating Speech Communication Systems that Include Noise Suppression Algorithm”. In: *ITU-T Recommendation P.835* (2003).
- [99] H. Assem et al. “A new adaptive redundancy control algorithm for VoIP applications”. In: *2013 IEEE Global Communications Conference (GLOBECOM), Atlanta, GA* (2013), pp. 1323–1328.

- [100] N. Khitmoh, P. Wuttidittachotti, and T. Daengsi. “A subjective VoIP quality estimation model for G.729 based on native Thai users”. In: *16th International Conference on Advanced Communication Technology, Pyeongchang* (2014), pp. 48–53.
- [101] T. Daengsi and P. Wuttidittachotti. “QoE modeling: A simplified e-model enhancement using subjective MOS estimation model”. In: *2015 Seventh International Conference on Ubiquitous and Future Networks, Sapporo* (2015), pp. 386–390.
- [102] L. Angrisani et al. “Measurement of the IP Packet Delay Variation for a reliable estimation of the mean opinion score in VoIP services”. In: *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings, Taipei* (2016), pp. 1–6.
- [103] Weiwei Zhang et al. “Perceived QoS assessment for VoIP networks”. In: *2013 15th IEEE International Conference on Communication Technology, Guilin* (2013), pp. 707–711.
- [104] “3GPP. Mandatory Speech Codec speech Receiving functions; Adaptive Multi-Rate (AMR) speech codec; Transcoding functions.” In: *3GPP, Tech. Rep. 26.090* (Mar. 2011).
- [105] Luca Fulchir:libRaptorQ. Mar. 2017. URL: <https://github.com/LucaFulchir/libRaptorQ>.
- [106] Packet generator:iperf. 2017. URL: <https://iperf.fr/>.
- [107] T. Mladenov, K. Kim, and S. Nooshabadi. “Forward error correction with RaptorQ Code on embedded systems”. In: *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), Seoul* (2011), pp. 1–4.
- [108] J. Holub, J. G. Beerends, and R. Smid. “A dependence between average call duration and voice transmission quality: measurement and applications”. In: *2004 Symposium on Wireless Telecommunications* (2004), pp. 75–81.

-
- [109] Y. Liu et al. “A MPTCP-Based Network Architecture for Intelligent Train Control and Traffic Management Operations”. In: *IEEE Transactions on Intelligent Transportation Systems* PP.99 (2016), pp. 1–13.
- [110] *Netem*. 2017. URL: <https://wiki.linuxfoundation.org/networking/netem>.
- [111] Yuanyi Xue, Beril Erkin, and Yao Wang. “A Novel No-Reference Video Quality Metric for Evaluating Temporal Jerkiness due to Frame Freezing”. In: *IEEE Trans. on Multimedia* 17.1 (2015), pp. 134–139.
- [112] A.Neri et al. “No Reference Quality Assessment of Internet Multimedia Services”. In: *Procs. 14th European Signal Processing Conference (EUSIPCO)* (2006).
- [113] N. Staelens et al. “No Reference Bitstream-Based Visual Quality Impairment Detection for High Definition H.264/AVC Encoded Video Sequences”. In: *IEEE Trans. on Broadcasting* 58.2 (June 2012), pp. 187–199.
- [114] N. Staelens et al. “Constructing a No-Reference H.264/AVC Bitstream-Based Video Quality Metric Using Genetic Programming-Based Symbolic Regression”. In: *IEEE Trans. on Circuits and Systems for Video Technology* 23 (Aug. 2013), pp. 1322–1333.
- [115] M. Seyedehbrahimi, C. Bailey, and X. Hong Peng. “Model and Performance of a No-Reference Quality Assessment Metric for Video Streaming”. In: *IEEE Trans. on Circuits and Systems for Video Technology* 23.12 (Dec. 2013), pp. 2034–2043.
- [116] M.A. Saad, A.C. Bovik, and C. Charrier. “Blind Prediction of Natural Video Quality”. In: *IEEE Trans. on Image Processing* 23.3 (Mar. 2014), pp. 1352–1365.
- [117] A. Eden. “No-Reference Estimation of the Coding PSNR for H.264-Coded Sequence”. In: *IEEE Trans. on Consumer Electronics* 53 (2007), pp. 667–674.
- [118] T. Brandao and M.P. Queluz. “No-Reference Quality Assessment of H.264/AVC Encoded Video”. In: *IEEE Trans. on Circuits and Systems for Video Technology* 20 (2010), pp. 1437–1447.

- [119] T. Oelbaum, C. Keimel, and K. Diepold. “Rule-Based No-Reference Video Quality Evaluation Using Additionally Coded Videos”. In: *IEEE Journal of Selected Topics in Signal Processing* 3 (2009), pp. 294–303.
- [120] Y. Liu et al. “Error detection For H.264/AVC coded video based on artifact characteristics”. In: *IEEE Int. Conf. on Intelligent Computing and Intelligent Systems (ICIS)*. Vol. 3. 2010, pp. 293–298.
- [121] T. Wolff et al. “Modeling subjectively perceived annoyance of H.264/AVC video as a function of perceived artifact strength”. In: *Signal Processing* 90 (2010), pp. 80–92.
- [122] M. Naccari, M. Tagliasacchi, and S. Tubaro. “No-Reference Video Quality Monitoring for H.264/AVC Coded Video”. In: *IEEE Trans. on Multimedia* 11 (2009), pp. 932–946.
- [123] F. Zhang et al. “Additive Log-Logistic Model for Networked Video Quality Assessment”. In: *IEEE Trans. on Image Processing* 22 (2013), pp. 1536–1547.
- [124] Video Database. “<http://vqa.como.polimi.it/sequences.htm>”. In: (Apr. 2017).
- [125] A. Mittal, R. Soundararajan, and A. C. Bovik. “Making a Completely Blind Image Quality Analyzer”. In: *IEEE Signal Processing Letters* 22.3 (Mar. 2013), pp. 209–212.
- [126] A. K. Moorthy and A. C. Bovik. “A Two-Step Framework for Constructing Blind Image Quality Indices”. In: *IEEE Signal Processing Letters* 17.5 (2010), pp. 513–516.
- [127] A. Mittal, A. K. Moorthy, and A. C. Bovik. “No-Reference Image Quality Assessment in the Spatial Domain”. In: *IEEE Trans. on Image Processing* 21.12 (Dec. 2012), pp. 4695–4708.
- [128] M. A. Saad and A. C. Bovik. “Blind Image Quality Assessment: A Natural Scene Statistics Approach in the DCT Domain”. In: *IEEE Trans. on Image Processing* 21.8 (Aug. 2012), pp. 3339–3352.
- [129] Z. Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Trans. on Image Processing* 13.4 (2004), pp. 600–612.

-
- [130] H.R. Sheikh and A.C. Bovik. “Image information and visual quality”. In: *IEEE Trans. on Image Processing* 15.2 (2006), pp. 430–444.