UiO : Department of Informatics
University of Oslo

Designing a prototype of Telemedicine system utilizing multiple Mobile BroadBand (MBB) networks

Ram Shrestha

Master's Thesis Spring 2014

Designing a prototype of Telemedicine system utilizing multiple Mobile BroadBand (MBB) networks

Ram Shrestha

20th May 2014

Abstract

Telemedicine system provide medical services to those who can not access it due to distance. In such systems, the patient's critical information, including medical data, images, audio and video, must be delivered to the hospital in time and in a robust fashion. Technologies such as Mobile Broadband Networks enables Telemedice, however in order to deliver these mostly delay-sensitive data, a robust communication system design is crucial. In this thesis, we design a communication system using multiple MBB links in order to minimize the delay of the different transmitted applications. We design the system in transport layer and specifically use multipath TCP (MPTCP) for data transmission. In order to reduce the application delay of different applications, we propose an algorithm called MPTCP Multi Stream (MPTCP-MS). MPTCP-MS reduces the delay of different applications by assigning these applications to specific paths. The proposed algorithm has been implemented in the linux kernel and tested in a real-world environment in Nornet Edge testbed. Our results show that the proposed algorithm outperforms the current MPTCP implementation in terms of delay and provides a viable solution to telemedicine system to transmit different data at the same time with minimum delay.

Acknowledgements

I would like to express my sincere gratitude to my supervisors Professor Paal Engelstad at Oslo and Akerhus College of Applied Science and Postdoctoral Researcher Ozgu Alay at Simula Research Laboratory for their support, guidance and motivation throughout my thesis work. Without their support and guidance, this thesis wouldn't have accomplished.

I am also thankful to Simone Ferlin-Oliveira at Simula Research Laboratory for providing the solutions for issues and problems related to the technical difficulties.

Finally, I want to thank my beloved family members and all of my friends and seniors including Yuba Raj Siwakoti, Kamal Khanal, Suraj Shrestha, Sijan Gurung, Santosh Joshi, Manoj Adhikari for providing continuous support and inspiration to accomplish my thesis work up to mark.

Ram Shrestha
20 May 2014

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Telemedicine, which actually means, “healing at a distance” [**1**] was evolved in 1970s. Telemedicine system utilizes modern Information and Communications Technologies (ICT) to take care of patient’s issues and problems by increasing access of medical professionals. Telemedicine system is a technology, which is responsible for transmitting a real time medical data from one site to another site through electronic communications technologies to get the medical care, assistance remotely. Telemedicine system is a rapidly growing application, which facilitates two-way telecommunication of important medical information of patients such as real time audio video, documents, still images, Magnetic Resonance Image (MRI), X-ray scans, live cardiograms etc. over an interactive audio-video application from anywhere and anytime using available networks (Mobile broadband networks)[**2**]. In rural areas, there might not be enough medical resources like modern equipments, medical professionals and so on. Furthermore, there is a need to take care of patients and to reduce the waiting time of treatment, as patients are located in far locations. From this point of view, Telemedicine service has to adapt into rural areas. Remote areas like the middle of huge jungle, isolated places, and disaster scenarios where medical professionals are not available to provide good medical treatment; Telemedicine will be a great solution in such situations. Telemedicine system provides a path of communication between specialized medical personnel in urban areas and patients in rural areas, which enables the patients to obtain specialized health care services without any need to travel to the care centers.

Telemedicine could be defined as:

*"The delivery of health care services, where distance is a critical factor, by all health care professionals using information and communication technologies for the exchange of valid information for diagnosis, treatment and prevention of disease and injuries, research and evaluation, and for the continuing education of health care providers, all in the interests of advancing the health of individuals and their communities" [**3**]*

Telemedicine has changed the way of traditional diagnosis and treatment of patients. Telemedicine system will be the best solution when health professionals are unreachable in the spot where emergency treatment is required. There are lot of diseases which needs emergency treatment like heart attack, acute stroke, traumas, chronic diseases etc.. These conditions need instant treatment at right time otherwise lives can be in danger. Mobile Telemedicine is used in conditions such as patients in ambulatory care or in other mobile situations need to communicate with medical personnel who are in different regions. Using these systems, experts (doctors) can consult or examine patients' report remotely, which is helpful to save patient's life.

Norway was one of the first countries, which took an interest in Telemedicine system [**4**]. Norway has about 4.5 million population living in total land area of 386958 km2 and a very low population density i.e. 13 per square kilometer. For the health and social care, Norway spends about 7-8 % of the gross national budget or 35 % of the Norwegian state budget annually [**5**]. As it has a very low density of population, the Norwegian people might have to travel long distances to get appropriate medical services. Furthermore, hospitals are distributed and some may not contain all medical resources such as the medical experts (doctors) in particular fields. To tackle these situations, Telemedicine system has emerged in Norway. Norwegian Center for Integrated Care and Telemedicine (NST) is the world's largest research and development center in Telemedicine and eHealth System and situated in Tromsø in northern Norway [**6**]. Since 1993, NST has committed to integration of health care between patients and medical professionals. The aim of center is to contribute society with Telemedicine and eHealth practice in the health related field and also research on the future solutions in it [**6**].

The history of Telemedicine system depends on the development in technology. In the mid to late 19th century [7] Telemedicine was introduced, where telephone wires are used to transmit patient's data like electrocardiograph [8]. In 1960's, Telemedicine started in the military field, space technology sector and some of them are introduced as commercial system and use them [7, 9]. In the beginning, Telemedicine used the electronic equipments like television to interact with specialist and medical professionals and provided the good medical advice for patients. Lots of researchs and systems are already introduced about Telemedicine and they are using various communication infrastructures. With the increasing performance of technologies, Telemedicine system needs to evolve to adopt those technologies. Existing Telemedicine system utilized Second-Generation (2G) Global System for Mobile Communications (GSM), Third-Generation (3G) Universal Mobile Telecommunications System (UMTS), Satellite connections and others means of communications to communicate with patients and medical professionals remotely.

1.1 Motivating Application

We consider a Telemedicine system where an ambulance tries to communicate to the hospital. We assume there is a patient that has to be taken care of and his/her condition is time sensitive. In the cases of emergency, a diagnosis of condition involves various examination procedures and need instant medical care. Before the patients arrive at the hospital, some basic treatment or therapy must start. Initially, we have to perform the diagnosis or start initial treatment in an ambulance on the way to hospital. For this we need a trained personal who have the knowledge of the system installed in ambulance i.e. Telemedicine system as well as medical equipment that collects the information of patients. The initial treatment is based on a communication between the medical experts or specialist available in hospital and the patients carrying ambulance. The real-time medical information of patients like instant physiological data streams (MRI, ECG, X-rays, blood pressure etc.), audio and video are needed to transfer from moving ambulance to the hospital using available public carrier networks. Based on a

information received in hospital, medical professional analyze and return the possible treatment solutions. Throughout this thesis, different types of medical information is assumed as different applications with different requirements.
In order to cure the patient on the ambulance, there has to be a good communication link between the ambulance and the hospital that is capable of transferring delay sensitive information to the hospital in time. In most of the cases, this communication takes place over MBB networks since MBB provides the wireless connectivity to access the Internet using various cellular technologies like GSM, GPRS, EDGE, CDMA2000, 802.11 etc. For high-speed data transmission rate Third generation mobile communication technology (3G) is preferred in this project. 3G can transmit voice call and data information such as email, fax, instant messaging etc. simultaneously. The main reason of choosing 3G cellular networks is its wide area coverage and the capability to serve with moving vehicles. 3G is a new generation of mobile communication technology that combines wireless and multimedia communication with the Internet including multimedia communications like music, images, video and other streaming media, e-commerce, conference call, web browsing and other multimedia services [18]. The rapid development of the wireless technologies and the features provided by these cellular providers have overlapping coverage areas which allows us to connect more than one provider at the same time. On the other hand, each network has its own infrastructure and configuration that makes them behave differently. In order to provide increased bandwidth as well as to guarantee QoS requirements of different applications, these different providers can be accessed simultaneously through different multipath transmission methods using protocols such as multipath TCP.

1.2 Problem Statement

The given problem statement in this project was as follows:

Designing a prototype of real time Telemedicine system utilizing multiple mobile broadband networks.

In this thesis, we focus on the communication system design of a Telemedicine system and design a communication system that is capable of transmitting different applications simultaneously over multiple paths to provide the required bandwidth as well as to guarantee QoS requirements of different applications. MBB networks are choosen as the transmission medium and each path is assigned to a different MBB operator. The designed Telemedicine system is capable of transmitting different applications using their respective protocols simultaneously over multiple paths efficiently. More specifically, we consider the design of the sender that schedules different applications simultaneously over multiple links and on the client side, these applications are received and saved as illustrated in Figure 1.1 below:

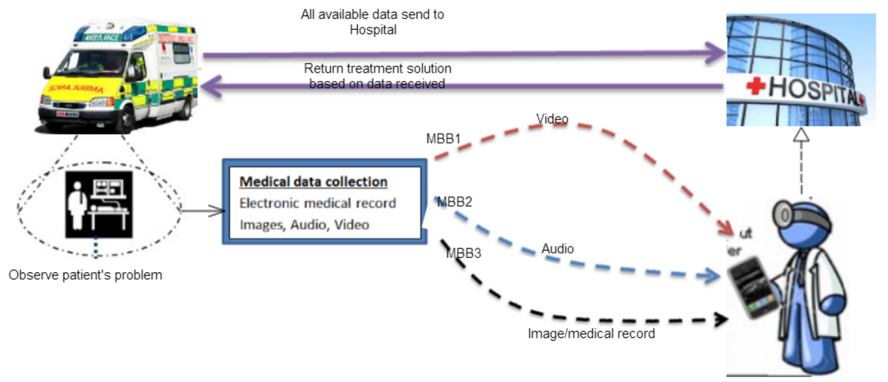


Figure 1.1: System architecture of Telemedicine system based on multiple MBB networks

Considering the delay sensitivity of Telemedicine data, we further propose a method to reduce the application delay by exploiting the multiple paths. The proposed algorithm modifies the current MPTCP implementation by specifying the path each application will be transmitting. We carried out experiments in a real testbed utilizing NorNet Edge [35] (see section 2.6 for more detail).Through extensive experiments, it has been shown that the proposed method outperforms the current MPTCP implementation.

Hardware and Software Required for Different Types of Telemedicine Systems

- **Central Processing Unit**
- **Media Acquisition**
  - **Medical Data (see section 2.2.1)**
  - **Image (see section 2.2.2)**
  - **Audio (see section 2.2.3)**
  - **Video (see section 2.2.4)**
- **Packet Scheduling Algorithms(see section 2.7)**
- **Video Streaming (see section 2.2.5)**
- **User Interface**
- **Communication Media/Interface**
- **Special Medical Equipment**

1.3 Related Work

*Performance of a 3G-Based Mobile Telemedicine System [***10** *]* A group of four students E. A. Viruete Navarro, J. Ruiz Mas, J. Fernández Navajas, C. Peña Alcega, University of Zaragoza had studied and designed a system to evaluate the performance of a mobile Telemedicine system based on third-generation (3G) mobile networks. This system is designed for communicating the personnel of an ambulance with medical experts in a remote hospital and offers real-time transmission of medical information and video-conference, other non real-time services as well. It used Universal Mobile Telecommunications System (UMTS) mobile access for the transmission media. The system architecture of designed system was based on advanced signaling protocols in IPv4/IPv6 3G that allow multimedia multi-collaborative conferences. The system was specially operated over 3G mobile networks to increase the quality of services offers using appropriate codecs.

To measure the performance of 3G mobile Telemedicine system has been tested to improve QoS by dimensioning dejitter buffers. Lots of tests have been done using over 64/128 Kbps (Uplink/Downlink) IPv4 UMTS accesses in urban scenarios (coverage level is high with low speed and the vehicle is in static position). In one end packets, are captured to collect the characteristics of the traffic injected in uplink i.e. jitter and IP level bandwidth and in the other end, it collects networks behavior i.e. packet loss rate and jitter.

Average bandwidth results: The research shows the total bandwidth used by all real-time medical user services fits in a specified channel i.e. 64 Kbps UMTS, even when the transmission with lowest efficiencies and the most bandwidth-consuming codec rates are used. The average IP-level bandwidths obtained in both endpoints seems very similar, and there is no packet losses observed. Therefore, from this experiment they summarize that the network does not change traffic characteristics regarding packet loss and bandwidth.

Jitter results: For jitter test, they prepare a test environment and tested two days with every hour of 48 experiments with operating the highest codec rate of all real-time medical services. Audio and medical data are normally generated every 0.06 and 1 second respectively. Hence the jitter effect of the audio services shows consistently distributed. The system causes 140 milliseconds and more jitter due to the time taken by UMTS uplink channel to transmit large size medical data packets of around 1300 bytes. For the video, the jitter is low because of its smaller packets size and not uniformly spaced. It says that huge dejitter buffer is sufficient to support all possible jitter effects and we have to make appropriate buffer dimensioning of dejiter.

*An overview of Recent End-to-End Wireless Medical Video Telemedicine Systems using 3G [***11** *]* A group of researchers (A. Panayides , M.S. Pattichis , C. S. Pattichis , C. N. Schizas , A. Spanias , and E. Kyriacou) gives the overview of recently available End-to-end wireless medical video Telemedicine systems utilizing 3G mobile telecommunication technology. The study is focused in explaining the basic components of system along with a overview of the recent advances in the field. The main objective of their study was to summarize and highlight the challenges and trends associated with implementation of diagnostically-based systems. The system architecture they proposed involved collecting a medical video that was done by high quality camera and/or a portable ultra sound device, obtained raw video was pre-processed for encoding and it used FFmpeg tool for real-time encoding, then encoded video is compressed by the video coding layer (VCL) of H.264/AVC and that was transmitted through transmission medium (H.264 to RTP/UDP/IP), which was handled by the network abstraction layer (NAL) of H.264/AVC. They designed a system for efficient encoding and transmission of medical video over the 3G networks which was further categorized as diagnostic region-of-interest (ROI) and non-diagnostic ROI based systems. Various case studies of different authors are discussed in this paper and key factors like efficient encoding, flexibility to data losses through transmission channels and transformation to varying network conditions are mentioned for successful setup a system.

Several other researches found on Telemedicine system using various wireless technologies. An article on Mobile Telemedicine Systems Using 3G Wireless Networks [**12**] by Yuechun Chu and Aura Gan (University of Massachusetts), presented a teletrauma system that can provide real-time audio, video and various medical information input between level I trauma center and ambulance. Presented system is based on a 3G wireless network that transmits the information collected from some sensors to the medical professional for health-care through portable devices like smart mobile phones or personal digital assistant (PDA). The research mentioned about the various challenges that may face by the systems such as Limited and Fluctuant 3G Links, Transmission of Bandwidth-hungry Medical Information, Transfer of multiple media streams simultaneously. To overcome these challenges, each system a software architecture that prioritizes, differentiates and transforms the medical information was implemented so that critical data is transmitted efficiently, reliably and with high quality.

1.4 Thesis Structure

The structure of this thesis is as follows: The first chapter, Introduction, introduces the motivation for doing this research. Several past related works are discussed and problem statement about the project is mentioned in this section. The second chapter, Background, provides general overview of Telemedicine system and its components required to develop new application for transmission system. In the third chapter, the methodology to develop the whole system and data of interest are discussed and some important scripts are mentioned. The fourth chapter, result and analysis, discusses on the implementation of the experiments and tests, displays results and analysis of results. In the fifth chapter, discussion and future work, provides a general summary of this thesis. It also disscusses about the practical implementation of the project, problem faced during project work and possible future works. Finally, last chapter, Conclusion, summarized the final study of this project work.

Chapter 2

Background

2.1 Telemedicine

Telemedicine is not a kind of special medical treatment; it is somehow similar as traditional kind of treatments only changes with some latest technologies. It is a medical service, which is the combination of traditional medicine and modern communication technologies system [**13**]. In Traditional treatment, the treatment of patient has to be done in the physical presence of Doctor but Telemedicine system works without physical presence of medical professionals. At the spot only they can communicate and give instructions about cure by sitting in hospital. Telemedicine refers to reducing health care costs, improving standards of medical treatment and the means of communication between medical experts and patients who needs health problems [**14**].
Here is the list of Services provided by Telemedicine system [**2**]:

- Primary care and specialist referral services
  This service may be involved in initial care of the patients. The specialist personnel provides pre-hospital care and refers for the appropriate solutions. Consultation with Medical professionals can be done remotely. This can be done with the use of two-way interactive video/audio communication or sending patient's vital signs, pictures of diagnostic or recorded video clips for the further review or diagnosis.

- Remote patient monitoring and treatment
  In remote patient monitoring, the health situation of patient is

monitored by health professionals remotely. The patient has a central system, which collects information and condition of patient from monitoring equipment or sensors e.g. heart ECG, blood glucose, blood pressure etc. and monitored through the central monitoring system, which is controlled and supervised by health professionals or medicine specialists. According to the sign and symptoms, medical experts give the appropriate suggestions and give treatment.

- Consumer medical and health information
  This section includes the communication and getting health information from medical specialist to the consumers through the use of wireless Internet and multimedia devices. To provide support for health care they can make a group discussion on-line.

- Medical education
  It provides special medical education for the interested health professional in remote location. Organizing various education seminars about health-care in remote locations will help to educate multiple health professionals.

With the rapid development and increasing availability of ICTs in low cost and the mechanism to replacement analog forms of communications with digital methods, gives wide interest and more effective ways for health-care provider in the medical field of Telemedicine system [**8**, **9**]. The development and popularization of the Internet has increased day by day and also has improved the ICT infrastructure, which is positive sign for developing the Telemedicine system with web-based applications like e-mail, audio/video conferencing, consultations etc.
Advantages of Telemedicine system:[**2**, **12**]

- Improved access to health care services
  From more than 40 years, Telemedicine provides better access to health care services to patient at a remote location. Patients can interact with medical professionals and exchange their problems or information in remote sites and get better care. Telemedicine technologies enable early diagnosis, consultations, and treatment in emergency cases.

- Cost-efficiencies of health care services
  The reason for developing and implementing Telemedicine system is to reduce cost in health care services. Telemedicine has reduced cost of health care by avoiding unnecessary transportation to primary care of patient, reducing travelling time, decrease in hospital stay, reducing waiting time for consultation with professionals. And the efficiency of Telemedicine is increased by better management of chronic diseases and shared medical professionals staffing. It also reduces cost by utilization of limited healthcare resources like physical specialist, expensive medical resources remotely.

- Improved quality of care
  The study shows that the quality of healthcare services provided by Telemedicine is improving than traditional treatment method. As per various specialists, Telemedicine provides a superior solutions with great outcomes and satisfaction of patient.

- Patient demand
  The demand of Telemedicine system is in increasing order because of its benefits over medical field. The main impact of Telemedicine is on the patients, their family and their community. Telemedicine technologies help to make a patient life as a normal lifestyle; they don't need to get stress to meet the medical professionals.

2.1.1 The Personnel

For the perfect efficiency of the Telemedicine system in a real time system, it must require a suitable trained, skilled, or committed personnel. The person with good knowledge about clinical materials is required in both ends of Telemedicine system. For example in the case of emergency in ambulance, there must have one medical professional or trained employee who can handle the emergency patient contact requirements and on the other end i.e. in hospital there is obvious clinical experts or doctors must be available. Trained staff must be comfortable and can handle the emergency situation and can care the patient. For this purpose, they may need some prior training and knowledge about the technology used in Telemedicine system and equipment installed on ambulance and working mechanism of Telemedicine system.

At the other end i.e. hospital, personnel like medical expertise or specialist must be available when such an emergency cases needed. The Telemedicine system only works properly if we can provide these two factors appropriately i.e. the availability of an appropriate medical professionals and the reliability of the needed equipments. The performance and usage of Telemedicine will decreases with following reasons: technical reason or availability of appropriate staff. So, for the proper function of Telemedicine system, we must ensure that we should provide sufficient well trained staff and the links used to communicate between two ends must be carefully planned with minimum delays and high throughputs.

2.1.2 The Technology

The main part of the Telemedicine system is the technology system used in it. It is very essential to make a successful link between two end points and this is done through technology. There are lots of developments and improvements in communication technology system, which must be used in our Telemedicine system so that it works properly. The technologies like video conferencing, transmitting the patient's information via some means of communication have helped to minimize the distance between healthcare professionals i.e. doctor's office and medical equipment. All the equipments of Telemedicine system must need to work properly, since any failure in these equipments may cause dis-connectivity of links. If the link between these two points disconnects, the performance of Telemedicine system is degraded and even patient cannot get proper care and treatment. There are lots of modern computers and latest technologies which are available while integrating these components with our system. We have to be very careful and should have close attention to ensure reliability of system and ease of use.

2.1.3 The Requirements of Telemedicine Information System [15]

For the development of the Telemedicine system various kinds of technology and information are required. The different function requirement of Telemedicine system is described as follows in figure 2.1:

1. The signal collection system

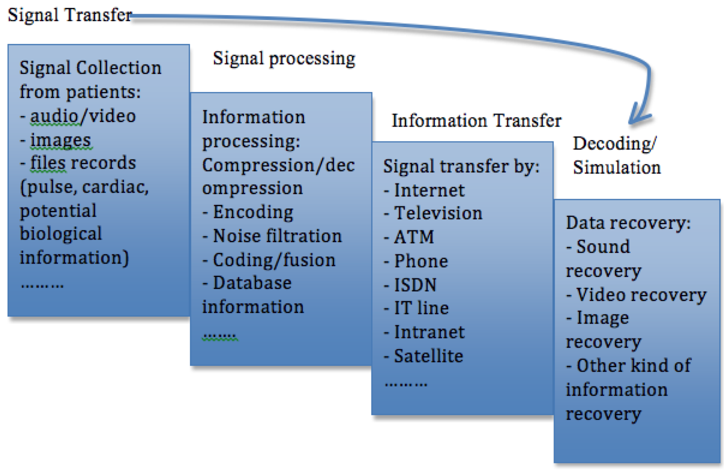


Figure 2.1: Telemedicine Information Systems [**15**]

The information or data of the patient can be collected by special medical equipment or sensors. The collected information could be images, pulse rate, voice, cardiac potential and history of the patient case. The Telemedicine system should have a user-friendly interface and easy to access. It is the preliminary phase of communicating the medical professional and patient.

2. Signal processing system
   When informations or signals are collected properly, they have to be transferred through one site to other site of the Telemedicine system. To do that, various signal processing techniques are required. It may include encoding, compressing/decompressing, noise filtration, coding and fusion technique to proceed signal without any loss.

3. Signal transmission system
   As the rapid development and improvement of information technologies, the signal or information transmission from one site to another becomes easier and faster. This section required standard communication infrastructure and technology such as ISDN, ATM, Internet, mobile broadband networks, satellite etc. Telemedicine is a critical system that requires more flexible communication between two sites without any delay as possible and with high performance. Multiple mobile broadband networks provide such facilities so that we can use these technologies to develop a reliable Telemedicine system.

4. Information decoding and simulation
   The received data or information must be converted or decoded into the proper format that is necessary for medical specialist. We have to be very careful about the recovered information, it has to be good enough to produce the original data signals as taken from patient's body. At this end, nice interactive interface can be developed that can interact and present information with medical professionals.

2.2 Different Data Types and Their Requirements

Information source	**Type**	**Typical file size**
Electronic stethoscope	Audio	100KByte
ECG recording	Data	100KB
Chest X-ray	Still image	1MB
Fetal ultrasound recording (30s)	Videos	10MB

Table 2.1: Examples of clinical information and their size [**16**]

There are different types of the medical information of patient which can be found according to the clinical situations as shown in table 2.1. The different types of medical data collected can be documents, audio, video, still images, electronics medical records etc. and they have their own size according to the data available on it. Hence, Telemedicine system can use many sources of data for evaluation and good treatment. We have to be careful before making decisions about choosing the medical equipment, which is required in Telemedicine system. In the past projects, it has been shown that transmitting all the possible type of information can lead to set up high functionality equipment and that increase the maintenance costs. To minimize the costs, we can only choose and limit the initial system for clinical goal [**16**]. Different type of data types are described below [**16**]:

2.2.1 Medical Data

The aim of the Telemedicine system is to transmit the available patient's medical information through remote to medical specialist. Patient's medical documents such as reports, letters or static medical reports can be transmitted in digital form only if the data exists in the digital form that can be read by computer. Otherwise, we have to make digital form of paper documents and transmit as images. Various digital equipments are available for this purpose like scanner, taking picture etc. TCP is considered as a very suitable transmission protocol to guarantee the transmission of records effectively and without any loss [**12**].

2.2.2 Image

In Telemedicine, two types of images are used – those of undefined image quality and those of with particular image quality. These depend upon the condition of patients. The photocopies of images are with the undefined image quality and the originals are with good image quality. In many cases and purposes, photocopies may be perfectly acceptable and legible but not good as an original one. Selecting the image matters in size as well, if the diagnosis accepts with photocopies then transmitting such a file may reduce the traffic in system. But for some type of crucial cases like studies for radiographs, it requires best quality images as possible as original.
A normal photographic image may be sufficient for many Telemedicine purposes. For instance, there are various inexpensive devices available such as low-cost digital cameras and flatbed scanner to provide and capture good images and digitize charts or paper based results like electrogram (ECG) traces, X-rays etc. Modern and expensive equipments are involved for generating high-quality diagnostic images like high-resolution X-ray film digitizers. To guarantee the transmission of data (images), reliable transmission control protocol (TCP) section 2.3.1 can be used [**12**].

2.2.3 Audio

In traditional analogue transmission technique, over long distance transmission, effects in noise and quality loss. These effects can be solved by digital signal transmission. It offers various advantages, without any degradation of signal quality digital signals can be transmitted over long distance within network. Compression is one way to transmit a live or recorded voice in a less data format than the original signal and transmit it over 3G wireless networks. To capture the audio, we can use our personnel Computer that are well equipped with a sound card. Suitable microphone is enough for capturing audio for Telemedicine purposes. If we need direct audio output from medical equipment like ultrasound scanner, we can simply connect our PC's card directly to the equipment. Collected audio signals are also in small size compared to video. So to guarantee the transmission of signals TCP can be used [**12**].

2.2.4 Video

Telemedicine system involves real-time video transmission between remote sites and hospital for the consultation and diagnosis purposes between a medical specialist and a patient. In case of video transmission, the issue on quality of video may arise, since unsurprisingly the higher the video quality, the higher the cost of the equipments and the transmission. In the existing Telemedicine applications, they use commercial videoconferencing applications to transmit videos. In our case, we are interested in the streaming video over various 3G links. Video can be captured by the various videoconferencing equipments, they are based on CODEC (coder/-decoder), which handles video pictures before transmission with compression at one endpoint and before display it decompress the received video pictures in other end. In current scenario, various multimedia devices are equipped with inbuilt camera like smart phone, PC's web-cam, tablets etc. which can be used to capture video picture and maintain videoconferencing functions. The requirement of Telemedicine system is feasible and realtime video transmission. For this requirement as well as tolerance to frame loss of real-time video, user diagram protocol (UDP) is considered as a very suitable transmission protocol [**12**] with its feature that it uses simple datagram with no congestion control.

2.2.5 Video Streaming

In this project we are interested to develop an audio/video streaming application. We need to encode and decode a multimedia file before and after transmission between two sites in network system. Streaming a data in network means a data stream from network interface not from our local drive. The process and protocols used for streaming multimedia files has been described below [**19**].

Protocol Used for Multimedia File Streaming

Streaming uses various protocols for stream multimedia files. There are a number of protocols developed which supports multimedia streaming such as multicast, RTP, RTCP, RTSP and UDP. Multicast is a process of sending information to multiple receivers at same time. To support multicast purpose it should reserve some IP address space that is known as class D IP addresses. Among these multicast [**17**] is more complex to deploy and difficult to manage in network layer so we can use combinations of RTP, RTCP and RTSP. The real time transport protocol (RTP) is a format of packets that can deliver audio video over networks. It is the bearer channel and mostly used in communication system, which involve in streaming media file. RTP control protocol (RTCP) is a separate signaling channel that is partner of RTP but doesn't transmit media stream itself. Real time streaming protocol (RTSP) is designed to select and control the streaming media file in communication system. For media streaming delivery and control, RTSP uses the RTP protocol. RTSP is some how similar to HTTP feature that uses TCP to maintain end-to-end network connections. All of the above mentioned protocols are used together for streaming media file.

Streaming Requirements

Streaming is a technique of transmission of multimedia data with steady and continuous flow from sender to receiver end such as in video conferencing audio/video streaming. Streaming has to maintain its minimum quality when data is in process to stream. To do this, streaming requires multiple methodology or techniques. The streaming of a multimedia file requires high bandwidth and it depends on time constraints. Table 2.2 represents the requirement of data rates according to the different multimedia file formats: The main advantage of streaming is the client or receiver starts to receive a data before the sender transmit entire data. To get the higher quality output data stream, the high bandwidth is required. The requirement of steaming is divided into two categories: application related and network related. Requirements like interactivity and start-up delay are application related requirements. The interactivity requirement is like a normal human nature needs. All the users await the same interactivity as they used to watch the movie in high quality format or they want to listen audio

Types	**Data rate approx.**	**Sample or frame size**	**Frequency**
Telephone speech	64Kbps	8 bits	8000/sec
CD quality sound	1.4Mbps	16 bits	44000/sec
Standard TV video (uncompressed)	120 Mbps	up to 640x480 pixels, 16 bits	24/sec
Standard TV video (MPEG-1 compressed)	1.5Mbps	variable	24/sec
HDTV video (uncompressed)	1000-3000Mpbs	up to 1920x1080 pixels, 24 bits	24-60/sec
HDTV video (MPEG-2 compressed)	10-30Mbps	variable	24-60/sec

Table 2.2: Characteristics of typical multimedia streams [**18**]

without disturbed. The delay is a kind of variable that can be caused by the transmission rate and data rate between sender and receiver end points and it is also known as jitter. With the improvement in bandwidth, start-up delay is getting minimized and we are getting new techniques for fast deliverance of quality multimedia files. The most prevalent case for getting delayed is related to waiting for multimedia file with high quality. The streaming and receiving application must support these features and using above defined protocols, we can achieve this functionality. There are various factors like throughput, jitter, transit delay, error rate, which influence the received signal.
To provide the proper streaming a multimedia file to the receiver end, the network should provide bandwidth that is required. If the throughput is not enough, the end users may experience jitter and transit delay where jitter is an undesired factor, which happens in all communication technologies links. It is the fluctuation of delay from end-to-end users. When packets are sending in a stream, the delay between two receiving packets is jitter. In network system, data have to transmit one after another packet, but due to some factors, the sequence of transmission results the transit delay. The transit delay depends on the medium of transmission of data. Error rate in transmission of multimedia files is dependent in time. If the receiver receives the correct sequence of data packets then the error rate is minimal otherwise the receiver doesn't get the correct packet sequence to

give a meaningful information about received data. To make flexibility on error rate, buffer mechanism can be used and before data have to be stream the buffer have to filled with blocks of packets.

2.3 Mobile Transport Layer

In the case of the Internet, to provide mobility support for most of the applications rely on a transport layer like Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). The main functions of the transport layer are multiplexing/de-multiplexing of data packages from/to applications based on Internet. UDP is a connectionless protocol that is used in multimedia message transfer and the service provided by UDP doesn't give guarantees about reliable data transfer. UDP has fast transfer mechanism because it doesn't have recovery option when error checking. While TCP is connection-oriented protocol which reliability of data transfer is very high. It is much more complex than UDP and to use in mobile environments, it needs some special mechanisms. However, TCP is slower than UDP, it gives guarantee that the transferred packet remains intact and arrives as same as it sent. The main difference between TCP and UDP is TCP based on connection oriented and UDP is connectionless between two applications. TCP contain built-in mechanisms to behave like a network friendly in a network connection. For example, if connection between TCP encounters some problem i.e. packet loss, then it assumes that due to network internal congestion to overcome this situation TCP slowdowns the transmission rate. TCP have error checking and recovery mechanisms. These are the reasons to choose or stay with network friendly TCP protocol [**20**].

2.3.1 Traditional TCP

- **Congestion Control**
  In the development phase of TCP, it has been designed for established networks with fixed end-systems. Data are transmitted through fiber optics, copper wires, network adapters or routers with some special hardware etc. and works without data transmission errors. Typically, there is no packet loss in fixed network but again it may have some

packet loss and it is due to a temporary overload in the path of transmission and it is known as a state of congestion at a node [**20**]. Congestion condition may appear any time even the network is designed properly. When the input rates of packet is higher than the output link, the routers have no choice to drop the packets. This condition is congestion. The sender receives the acknowledgement for the missing packet and assumes it due to congestion. The retransmission of such lost packet is not feasible this time because it may cause more congestion although it is not sure that the lost packet will transmit properly. To control congestion, TCP dramatically slows down the transmission rate. TCP guarantees to sharing the bandwidth even there is heavy traffic load in network.

- **Slow start**

  When the congestion is detected in network system, the behavior it shows by TCP is called slow start [**20**]. In a network system, there is idea of congestion window, which is always calculated by sender for a receiver. The size of the congestion window is assumes to one TCP packet size. The transmission of data in network is that sender sends a packet and waits for acknowledgment for that packet from receiver. If the sender gets acknowledgement then it sends two packets by increasing the congestion window by one i.e. congestion window = 2. Again if sender receives two corresponding acknowledgement it adds 2 to the previous congestion window and becomes 4 congestion window, one for each acknowledgement. Similarly, congestion window doubles in every acknowledgement receives by sender and to finish this one process it takes one round trip time (RTT). Such an exponential growth of congestion window is a part of the slow start mechanism.

  The growth of congestion window doubles every round until it reaches at the congestion threshold otherwise it may become too large. Then each time when sender receives acknowledgement the transmission rate is increased by linear or adding 1 to the congestion window. The increase in linear transmission rate by one congestion window continues until the time-out at the sender that detects a gap in transmitted data or due to missing acknowledgement because of same packet is acknowledgement continuously. In both cases sender sets the congestion threshold as a half of the current size of congestion window [**20**].

- **Fast retransmit/fast recovery**
  In TCP, if a receiver receives a packet from sender then only it sends acknowledgement about that packet to sender. There are two things which can reduce the congestion threshold, one is if the acknowledgement for the same packet is received by the sender and other one is that receiver receives all packets from sender up to acknowledged packet in sequence. Not only from severe congestion packet loss may occur due to simple transmission error. Such loss packets can retransmit by sender before the sending timer expires this mechanism is known as fast retransmit [**20**]. The sender can perform a fast recovery from the packet loss by continuing the transmitting data with the current used congestion window. Fast recovery mechanism of packet loss can improve the TCP efficiency. Using TCP fast retransmit/fast recovery define congestion and activates slow start mechanism.

Implication on Mobility

In fixed network, the slow start mechanism is most useful, but if we use mobile sender and receiver it decreases the efficiency of TCP because of slow start uses the wrong assumptions. In slow start congestion situation is occurs only in missing acknowledgement, but in wireless and mobile system it may not be the reason for packet loss. It is common that, Wireless links have more error rates than fixed network copper or fiber cable links. Mobility in a network system itself may cause a packet loss.
For a mobile end-system it is not possible to easily handover from one type of access point to another. For example, in mobility it provides mobile IP for an individual sender and receiver it may change during transmission data. When using this IP, while the mobile end-system moves to the new foreign agent there may be some packets remaining to transmit from the old foreign agent. In such a condition a packet loss can be occurred because the old foreign agent may not be able to forward those loss packets to the new foreign agent and it is caused by the rerouting traffic problem [**20**].

The TCP mechanism cannot distinguish the making decision of packet loss due to congestion and missing acknowledgements via time-outs between the different causes. TCP may have a problem in error control mechanism or misused of congestion control for packet loss. Both reason are different but TCP cannot distinguish which one reason causing packet loss. The slow start mechanism doesn't help in case of wireless links transmission error or during handover of one access type link to another [**20**]. This behavior results the degradation of TCP performance if used with wireless link or mobile nodes.

Classical TCP Improvements

The overview of the classical TCP improvements for mobility is given in the table 2.3:

TCP Over 2.5G/3G Wireless Networks

Based on a various characteristics of a network systems such as data rates, latency, jitter, packet loss etc. the following configuration parameters can be described to adapt TCP over wireless network environments [**21**]:
Large windows: Congestion window mechanism is used by TCP protocol to control transmission rate in a network systems. TCP over 2.5/3G should support huge enough window sizes based on the bandwidth delay product that was found in the wireless communication techniques. To overcome this limitation, we can make larger buffer size and with the help of the window scale option [**22**]. To increase the performance specially for a short transmissions, we have to increase the window size i.e. 2 to 4 segments.
Limited transmit: Limited transmit [**23**] is useful when we need to transmit small amounts of data. It is an extension of fast retransmit/Fast recovery with small congestion windows for all TCP connections in system. This technique is effective if a huge number of segments are lost in a window or when the size of congestion window is small that may avoid some retransmissions because of TCP timeouts.
Large IP Maximum Transfer Unit (MTU): To increases the congestion window of TCP faster, we need to make larger MTU. MTU is the maximum size of an IP datagram that supported by a link layer and it fragments IP datagrams into PDUs. Larger MTUs probably increase the performance by

increasing congestion window of TCP connection systems. In TCP over 2.5/3G, designers are free to choose small MTU values (like 576 bytes) to large values (like 1500 bytes for IP pakets on ethernet) that supports by the type of they used [**21**].
Selective acknowledgement (SACK): SACK [**24**] allows retransmission of the selected packets. It is more beneficial than other standard cumulative scheme.
Explicit Congestion Notification (ECN): ECN [**28**] allows a receiver to warn a sender of congestion occurred in the network system. It was done by setting the flag ECN-Echo upon receiving an IP packet that has experienced the congestion previously. This technique help us to determine packet losses due to congestion or transmission error. We can achieve this only if the ECN routers are deployed in network system.
Timestamp [**21**]: TCP connection in a network with massive congestion windows might give more benefits from lot of frequent RTT samples given timestamps by adapting faster to ever-changing network conditions. Timestamps helps to reduce the effect of bandwidth oscillation and it can spikes higher delay that can be tolerated by TCP without facing falsr timeout.

2.3.2 MPTCP (MultiPath TCP)

TCP is the most important and widely used network protocol in the communication system. Most of the Internet applications are using this protocol such as peer-to-peer file sharing, world wide web (WWW), File Transfer Protocol (FTP), e-mail, some small streaming applications and many more. Traditional TCP protocol only supports the single path communication for transmitting data. However, the improvement and development of new technologies gives much and more electronics devices like smart phones, tablets etc. equipped with multiple interfaces, which can access the Internet through multiple heterogeneous access technologies. Current TCP protocol doesn't support the functionality of using multiple interfaces simultaneously. Hence the MultiPath TCP (MPTCP) is introduced which enables the simultaneously use of multiple interfaces/IP-addresses. MPTCP is an improvements and modification of traditional TCP with adding features of enabling multiple interfaces and spreading data across multiple subfolws.

In this research, we are targeting the transmission system that enables different applications with different delay requirements. The critical and important data sets or multiple streams in Telemedicine system must have to transmit at the same and right time. To enable this functionality we can use all available network capacity as possible as we can. To do so, we have investigate the best approach i.e. utilizing MPTCP. This functionality can be done by using available paths in network system. Each end host utilizes MultiPath TCP to spread the traffic across multiple paths. Benefits of using MPTCP are better throughput, better resource utilization and smoother reaction to failures [**40**]. MPTCP is a new technique for efficient load balancing, rather than using hashes and ECMP in the network, MPTCP does load balancing in each end nodes. Detailed of MPTCP [**40**] is given in RFC 6182 [**41**] and 'TCP Extensions for Multipath Operation with Multiple Addresses' Internet draft [**42**]. MPTCP is an expansion of the TCP/IP stack. The byte stream of the application in network system is split into multiple subflows. For each interfaces MPTCP creates multiple subfolws. Each subflows consists of congestion control mechanism so that MPTCP can manage paths of various bandwidth. Such congestion control mechanism takes care of the network traffic; it helps to move the congested path to a link with less congestion. This feature leads the MPTCP to the load balancing, it manages the paths according to the load of the traffic accordingly.

How does MPTCP Works?

Figure 2.2 describes the working mechanism of MPTCP. In the MPTCP enabled kernel system with multiple interfaces, TCP component is divided into a MPTCP component and for each interface it split into TCP subflow components. The application sends the byte stream to the MPTCP components and it splits the received byte stream into multiple segments. The divided byte stream then handed to the TCP subflow components and transmitted through the multiple interfaces. Each TCP subflow performs as a traditional TCP flow to the network system.
The components of MPTCP include various functions that manages the path management by finding and implementing available multiple paths to destination. The received stream is divided into multiple segments and numbered using packet scheduling and divided segments are transmitted

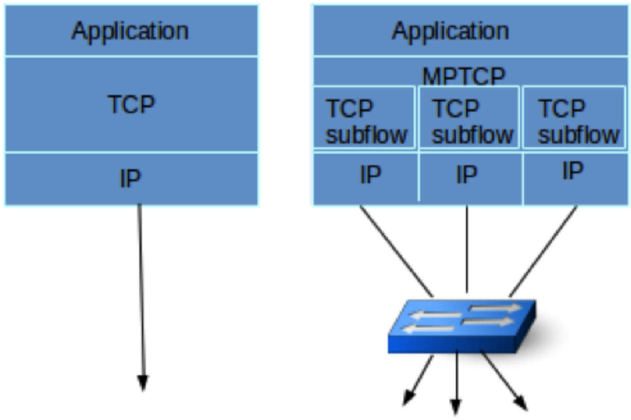


Figure 2.2: Functionality of Traditional TCP and MultiPath TCP

via one of the available subflow. By using numbers marked by sender the receiver can put the receiving segments in the correct order and as in original form of byte stream. All the subflows consist of congestion control mechanism that helps to load balancing of the transmission. If any of the subflow gets congested, the traffic of that subflow is changed to the less congested subflow. When one of the subflow fails or disconnected, the retransmissions of the flow is maintained on another subflow and the transmission continues. Figure 2.3 illustrates the initial connection setup and additional setup for subflow connection of MPTCP for server X and server Y, here server X initiated both initial and additional subflow setup. The initial connection setup of MPTCP is similar to the traditional TCP setup using SYN, SYN/ACK and ACK flags sequence with some difference. The only changes are that the MPTCP enabled server includes the MP_CAPABLE option in each sequence. Server X sends a SYN with the MP_CAPABLE option including its authentication key and other flags for checksums and for cryptographic algorithm. For this MPTCP connection, 64-bit authentication key is used to authenticate the future subflows [**43**]. If the server Y is also MPTCP enabled then it also returns SYN/ACK with the MP_CAPACABLE option and its authentication key and other flags. If One of the servers i.e. server Y is not MPTCP capable then it doesn't recognize the MP_CAPABLE option and it only replies SYN/ACK without MP_CAPABLE option. In this case, server X and server Y communicate with only normal TCP connection mode. However,

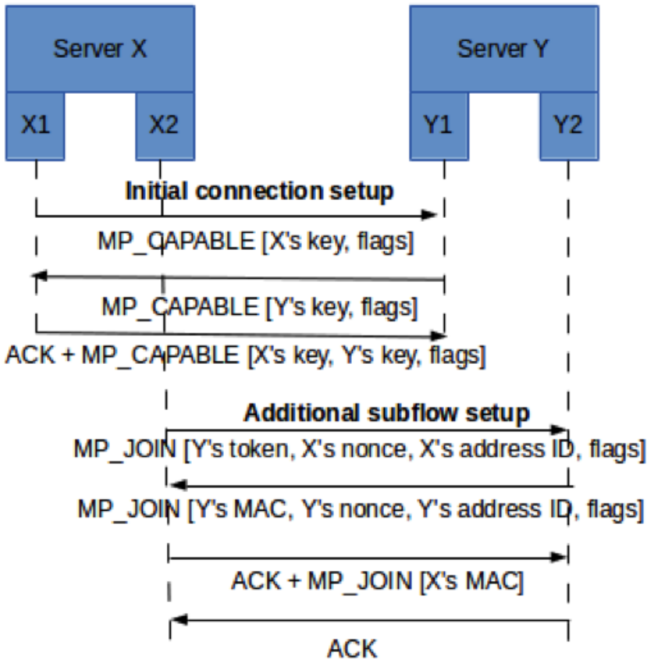


Figure 2.3: MPTCP connection setup

if server Y is MPTCP capable then it also returns SYN/ACK with the MP_CAPACABLE option and server X replies an ACK with MP_CAPABLE option and authentication key of both server and respective flags that completes the initial MPTCP connection setup.

The additional subflows setup is also similar to normal TCP connection setup, which additionally includes MP_JOIN option. Server X starts communication sending SYN packet containing MP_JOIN option with token, random number, own address ID and flags. The token is SHA-1 hash of the receiver's key [**43**] and used by receiver (server Y) to identify the connection between them. To prevent replay attacks on authentication technique the random number (nonce) is used. Where address ID represents the source address (X2) of sending packets. The flags contain the information used by the sender to inform to other server to use this subflow as backup in case of other path have failed and can be used this subflow immediately. When receiver receives a valid SYN packet with MP_JOIN option and token, the receiver reply with a SYN/ACK with MP_JOIN option and including a Message Authentication Code (MAC), random number (nonce) and own address ID(Y2) [**43**]. Now server X replies ACK with the MP_JOIN option with own MAC. After this, there is a four-way handshake by server Y is done with sending ACK to server X. Finally when the server X receives final ACK then it sets the connection to the ESTABLISHED state and communication continues.

2.3.3 User Datagram Protocol (UDP)

UDP is another type of transport layer protocol, which is defined for use with IP in network layer protocol. It is alternative to the TCP, which offers a limited amount of services in network where data are exchanged between computers that uses IP. Like TCP, it also divides data into packets called datagram and transmits from one system to another. Unlike TCP, UDP is based on connectionless protocol that means one end can send load of data in packet forms to other end and that may be the end of the connection between them. Its stateless feature helps to save processing time for servers that needs to answer small queries to the large number of clients. It is very convenient for applications that requires efficient, fast transmission of data such as game [**27**]. Using UDP, we can send multiple messages as chunks of packets. It is faster than TCP because there is no recovery option even it does error checking on packets. UDP provides a port number that helps to distinguish various user requests and also it has checksum facility, which verifies that, the information arrived to the receiver intact. UDP sends packets individually and they are checked for integrity only if data arrived in receiver. The UDP header size is 8 bytes and consists of four fields length of 2 byes each: source port, destination port, UDP length and UDP checksum.

2.4 Tools Used for Multimedia Streaming

2.4.1 Multimedia Framework for Streaming

FFmpeg [25]

FFmpeg is one of the popular open source multimedia framework that is able to process all kind of multimedia things such as encode, decode, transcode, stream, multiplexing, demultiplexing, filter and play any kinds of files with any formats. It supports any kinds of format with designed by some specific standards committee, a cooperation or the community, either it is very old or latest formats. It contains various types of developers'

libraries that can be used by application such as libavformat, libavcodec, libavfilter, libavutil, libswscale, libavdevice and libswresample. As well as FFmpeg provides four kinds of tools, i.e. ffmpeg, ffserver, ffplay and ffprobe that can be used by end users for transcoding, streaming, playing and stream analyzing respectively. The FFmpeg project gives a best technically possible solution for both application developers and end users. To solve security issues in open source software it reviews the code and provides possible updates through releasing new version frequently.
FFmpeg is a very useful command line tool that can convert any multimedia files (audio/video) between any formats. ffmpeg can convert between random sample rates and resize any video with a high quality polyphase filter on the fly. It reads from random input files that may be regular files, network streams, pipes, grabbing devices etc. and it produce output file that we want. The basic syntax of ffmpeg is written as below:
ffmpeg -i in.avi -b:v 64k -bufsize 64k out.avi
Here –i option represents the input of multimedia file which have to convert, and other options are to set bitrate of the output file, and without any option is considered to be the output file name. Here we can provide any kind of formats in input section.
ffserver is a multimedia-streaming server tool that is useful for live broadcasts. It supports various live feeds, time shifting on live feeds and streaming from various file types. ffserver has a facility to seek the past of each live feed. It works by forwarding pre-recorded stream that is read from file or streams encoded by ffmpeg. We have to configure ffserver such that the kind of file is being streamed or the properties of videos through configuration file which can be specified through the option -f. If we haven't explicitly specified, it will read from /etc/ffserver.conf file. FFserver receives stream files as prerecorded files or FFM streams from instance of ffmpeg as an input, then streams them over HTTP/RTP/RTSP/UDP.
The ffserver instance will listen on port that is assigned in the configuration file. We can send one or more FFM stream and launch one or more instances of ffmpeg to the defined port where ffserver is supposed to receive them. The input stream is known as feeds and defined in a <Feed> section of the configuration file and that must be identified by unique name. For every feed we can have various output streams in different formats that can be specified in <Stream> section of configuration file. The feed URL looks like:

http://ffserver_ip_add:http_port/feed_name
where *ffserver_ip_add* will be the IP address of ffserver installed machine and *http_port* is the port number that is listening and *fee_name* is a name of feed that is defined in configuration file.
Every feed is linked to a stored file on disk, which is used to transfer pre-recorded data to a respective player as soon as possible to the stream when new content is added. The stream access for HTTP URL looks like:
http://ffserver_ip_add:http_port/stream_name[options]
where *ffserver_ip_add* will be the IP address of ffserver installed machine and *http_port* is the port number that is listening and *stream_name* is a name of stream defined in the configuration file. *options* will be the list of options which defines how the stream is served by the ffserver.
ffplay is a simple and portable type of media player that is based on ffmpeg libraries and on Simple DirectMedia Layer (SDL) library [**25**]. It is frequently used for the different FFmpeg APIs as a testbed.
ffprobe is a tool for analyzer of multimedia stream file. It collects data from various multimedia streams and prints in a machine- and human- readable pattern [**25**].

Encoding

FFmpeg encodes a video and audio through various encoding techniques. For video, it uses following encoding mechanisms: x264 encoding, Xvid/DivX/MPEG-4 encoding, VFX encoding, vpx (WebM) encoding and for audio it uses MP3 encoding, AAC encoding [**26**]. Each encoding technique has their own goal.
X264 encoding; is an H.264/MPEG-4 AVC type of encoder and it achieves the best quality of video even at low bit-rates. For general use, two types of rate control modes are suggested: Constant Rate Factor (CRF) or Two-Pass ABR (Average Bit Rate).
Xvid/DivX/MPEG-4 encoding; both DivX and Xvid are implementations of the MPEG-4 standard and need to use –c:v mpeg4 option to encode in theses formats.
VFX encoding; each VFX pipeline requires a method of converting still frames into sequence of motion. The produced motion frames can be played back on a large projector or screen for the purpose of doing a review.

There is a possibility for play back the resources in high-resolution frames but it requires huge amount of bandwidth.

2.5 Technology That Enables Telemedicine Communication

2.5.1 Mobile Broadband (MBB)

The improvement of the mobile broadband technology and Internet has changed the way to access and use of Internet. It is becoming the most substantial part of the communication technologies infrastructure in the world. Meanwhile, portable devices like smart-phones, tablets, notebook, laptops etc. are becoming more popular in many fields of users. To provide high-capacity 3G and 4G mobile networks for these digital devices is a great challenge for the telecoms operators and Internet service providers (ISP). With the development and increased use of such a latest technologies the use of MBB networks has exploded over the last few years and the study shows that the mobile traffic in 2012 was nearly 12 times greater than the total traffic in 2000 worldwide and The annual estimated MBB traffic at the growth rate of 66% towards 2017 [**29**].
The MMB networks should facilitate continuous services of the different Internet applications and protocols, so that a consumer feels it to be equivalent quality to a wired network services. Following are the key requirements to increase the MBB networks services [**30**]:

- **High bandwidth**: MBB networks should provide a reliable link with high bandwidth that can be shared by multiple consumers. A system must have a control policy that allows the resources to be efficiently switched between digital devices on a packet-by-packet basis.

- **Low latency**: To get the high bandwidth of the link the latency should be less. When consumers get connected to the requested resources link, they are supposed to get high burst rate and transmission of packets with minimal delay. Delay can be minimized for achieving link reliability by using mechanisms like Automatic Request requests (ARQs)

- **Quality of Services (QoS)**: Typically communication between wirelesses is resource constrained. QoS provides of means for effectively dividing the available resources. Network QoS can be affected by the different layout of a service provider's wireless access-points. Different service providers can provide different level of services for users for their benefits and can derive revenues.

Evolution of MBB

This section describes the evolution and migration of mobile broadband technologies from 1G to 4G.The Third Generation Partnership Project (3GPP) was formed by standard-developing organizations from all around the world. 3GPP solved and maintain the problem of parallel development of standards in various region of world. Currently 3GPP consists following organizational partners i.e. In Japan - Association of Radio Industries and Businesses (ARIB) and Telecommunications Technology Committee (TTC), China – China Communications Standards Association (CCSA), Europe – European Telecommunications Standard Institute (ETSI), USA – Alliance for Telecommunications Industry Solutions (ATIS), Korea – Telecommunications Technology Association (TTA) [**31**]. 3GPP has multiple phases in progress, it started with EDGE and then UMTS that is continued as today's 3G technologies like HSPA, HSPA+. Further development and research is evolving with LTE and more further it evolving to LTE advanced which is considered as 4G [**32**]. The generation transition from 1G to 4G has been summarized in table 2.4. In 1980s, 1G is introduced that works for analog cellular technologies. For digital communication 2G is deployed in 1990s with GSM and CDMA IS-95 technologies. It provides digital services like short messaging and lower speed data. ITU's IMT-2000 specified various requirements for 3G networks such as 144 kbps of throughput at mobile speed, 384 kbps at pedestrian speed and 2 Mbps in indoor environments. After following these standards HSPA and CDMA2000 EV-DO are introduced as primary 3G technologies but WiMax was designated as official. Evolution of MBB can be summerized in figure 2.4.

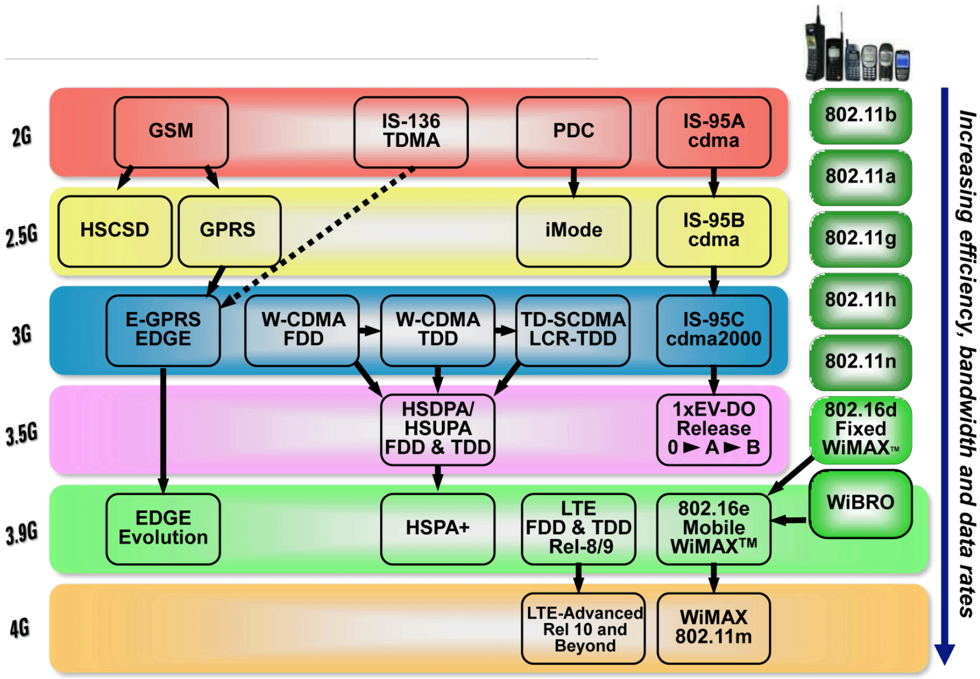


Figure 2.4: Evolution of MBB (courtesy of Agilent Technologies)

2.5.2 The Third Generation Mobile Communication Technology (3G)

Background of 3G

The first generation (1G) of mobile telecommunications technology only belongs to the analog communication system, which is used to voice transmission services with low quality. The second generation (2G) mobile telecommunication technology is the improvements of 1G, which uses digital modulation techniques for the transmission and to support low-speed data service of 1G. 2G digital mobile communications introduce the data services over the mobile-communications networks; an initial data service provides by 2G was text messaging (SMS), fax and enables circuit-switched data services for email and other data applications. The initial data rates of 2G were up to 9.6 kbps [**31**]. Packet data over mobile communication systems became a reality at middle of 1990s with the development of General Packet Radio Services (GPRS) in Global System for Mobile communications (GSM) system and it was referred to as 2.5G. In the early 1990s, mobile technology providers initial steps towards 3G with more feature and high bandwidth strength than 2G and 2.5G. 3G telecommunication networks provide mobile broadband access with high band such as several Mbps to the di-

gital devices like smartphones, tablets, notebook, and mobile modems for laptops.
The research activity on a 3G mobile communication system was done side by side during the evolution and deployment of 2G. The first phase of initial 3G researches was carried out in EU-funded project Research into Advanced Communications in Europe (RACE) in Europe, which was named as Universal Mobile Telecommunications Services (UMTS). In the second phase of RACE, the project Advanced Time Division Multiple Access (ATDMA) and the project Code Division Test Bed (CODIT) developed more 3G concepts based on Wideband Timed Division Multiple Access (WTDMA) Wideband Code Division Multiple Access (WCDMA) technologies. The next phase of European research was Advanced Communication Technologies and Services (ACTS), which include UMTS-related project called Future Radio Wideband Multiple Access System (FRAMES) and this project uses multiple access concept that included both Wideband TDMA and Wideband CDMA components [**31**]. Not only in Europe, development and research of 3G activities was going on in all around the world. In Japan, the WCDMA based 3G wireless communications was defined by the Association of Radio Industries and Businesses (ARIB) and also Korea started to work on it. In United States, it also developed WCDMA technology based WIMS within T1.P1 committee [**31**].

Definition of 3G

3G is a short form of the third generation mobile communication technology. It is a mobile broadband wireless network, which supports high-speed data transmission rate and can transmit voice call and data information such as email, fax, instant messaging etc. simultaneously. The research and implementation on a 3G mobile communication was started in International Telecommunication Union (ITU) in the mid 80s. 3G is a new generation of mobile communication technology that combines wireless and multimedia communication with the Internet including multimedia communications like music, images, video and other streaming media, conference call, e-commerce, web browsing and other multimedia services [**31**]. 3G packet switched and packet-switched data services has set the evaluation criteria according to data rates as [**31**]:

- Up to 2 Mbps in an outdoor environment
- Up to 144 kbps in a pedestrian environment
- Up to 64 kbps in a motion vehicular environment

All 3G network technologies were compared with these benchmark numbers. However, nowadays all deployed 3G system's services data rate is beyond 2 Mbps.

Basic Feature of 3G

In 3G technologies, the information or data is split into packets before they are transmitted and using some technique it reassembled the information at the receiving end. The main network feature of 3G systems is wireless interface technology in cellular mobile communication system. It includes multiple access/ duplex mode, cell multiplexing, modulation, radio channel parameters, the application frequency, channel coding and error correction, multiplexing mode, frame structure, the physical channel structure and many other aspects [**31**]. The innovations of 3G wireless mobile communication technologies are in the following field [**18**]:

- Use of high frequency spectrum
- Achieving multi-service and multi-rate transmission
- Use of broadband radio frequency channel to support high-speed services
- Use of adaptive antennas and software radio technology
- Fast power control

The 3G Standards

The first generation mobile-communications system is all about the analog modulation that uses the frequency division multiple access (FDMA). The main demerit of this technology is that its spectrum utilization is very low and signaling interface with voice service. Second generation mobile-communications system introduced the digital modulation that uses time division multiple access (TDMA), which increase the performance and capacity of the system by making independent channel for signal transmission. Still this system has some limitation, it does not improve the hand-off performance [**39**]. To improve this code division multiple access (CDMA) was introduced with various added features such as simple frequency planning, high factors of frequency reuse, large system capacity, good anti-multipath capacity, soft capacity, high communication quality, soft switching capacity for great potential for development. CDMA is the third generation mobile communication technology based-system. Here are brief introduction of some 3G standards [**33**]:

- **WCDMA**: WCDMA, means broad band code-division multiple access, is proposed by the European broadband CDMA technology that is based on GSM system developed 3G technology standard. Japanese and American companies are also supporters of this technology. United States of Ericsson, Alkatel, Lucent, Nokia, Nortel, NTT of Japan, Sharp, Fujitsu and other more manufacturers are involved. Existing GSM network technology is used to setup this system, which is easier for system providers to transit. It proposed GSM (2G)-GPRS-EDGE-WCDMA (3G) standard evolution strategy. The GSM system is more popular in Europe and Asia that makes this standard more useful to accept. Hence, WDCMA has inherent advantages from the view of market.

- **CDMA2000**: CDMA2000 is an expansion of 2G's CDMA that is also known as CDMA multi-carrier. It is managed by the north American Qualcomm., Lucent, Motorola and Samsung companies. The system is derived from original structure of CDMAOne and directly upgraded to 3G with low construction cost. However, the coverage of this system is less compared to W-CDMA but research and development of CDMA2000 is going in the fastest progress. This system standard raised the evolution strategy as CDMA-IS95(2G)-CDMA2001x(2.5G)-CDMA2003x(3G). The main difference between CDMA2003x and CDMA2001x is on the application of multi-carrier technology.

- **TD-SCDMA**: Time division-synchronous code division multiple access (TD-SCDMA) is one of the standard of wireless communication technology. It was first introduced by China and based on Radio Transmission Technology (RTT). China has completed the TD-SCDMA standard with international cooperation and becomes a member of the CDMA TDD standard [**34**]. TD-CDMA has the characteristics of low radiation so it well known as "green 3G" [**34**].this standard will be integrated with various leading technologies such as intelligent wireless, software radio technology, synchronous CDMA etc. with multiple advantages of flexibility of business support, spectrum efficiency, cost and other aspects. Additionally, this standard gets the huge attention from major telecom equipments vendors of the large domestic market of China, all the equipment manufacturers implement this standard. This standard directly upgraded to the 3G from 2G, there is no intermediate links of 2.5G.

- **WiMax**: Worldwide Interoperability for Microwave Access (WiMax), also known as IEEE802.16 wireless metropolitan area network. WiMax technology must need to avoid authority or authorized by the combination of microwave equipment. Due to lower cost that will increase the market of wireless broadband technology and improves awareness of service providers and enterprises. The plenary meeting of wireless communications technology in the International Telecommunication Union (ITU), Geneva at October 19, 2007, after most countries voted WiMax was approved to become official 3G standard followed by WCDMA, CDMA2000 and TD-SCDMA [**31**].

2.6 Nornet Edge (NNE) [35]

NNE is a dedicated framework for experimentation and measurements in multiple mobile broadband networks to determine and compare performance and quality of multiple network operators. Each measurement node is a Linux-based embedded computer that runs a standard Linux distribution (Debian Wheezy with 3.08 kernel) and is connected to various mobile broadband providers. It includes an extensive backend system that consists of large number of servers for controlling and monitoring the measurement nodes, deploying like SSH proxy server, which enables to login remotely to node. The system can process and manage experiments to run long-term and parallel on various network connections simultaneously and the collected measurement data can be stored and visualized later on. The overview of NNE is given in figure 2.5 below: NNE supports multi-homing

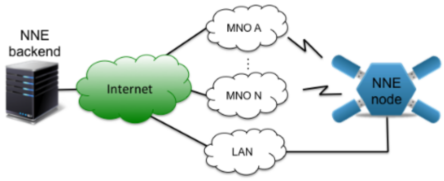


Figure 2.5: Overview of NNE [**35**]

concept, which enables multiple connections and transmission with various network providers simultaneously and that can be provided by using SCTP and MPTCP protocols. Currently, NNE node is connected with 4 MBB providers Telenor, Netcom, Tele2, and Ice. Netcom and Telenor is providing the 2G/3G/4G GSM networks nationwide, while Tele2 is building 2G/3G GSM network in specific reasons and Ice is providing a data-only CDMA network with lower frequency band with 450 MHz. There are various features of NNE including:

- **Unprecedented scale and nationwide geographic coverage**: currently NNE consists of more than 400 measurement nodes and they are distributed all over the Norway. Huge number of measurement nodes and its area of coverage help to give a view of the characteristics of whole network system from main cities to remote areas.

- **Fully programmable measurement nodes**: The measurement nodes are Linux-based embedded computer that run a standard Linux distribution (Debian Wheezy with 3.08 kernel). They give great flexibility and powerful performance to run the various types of measurements that depends on size and bandwidth experiments.

- **Multi-homed measurement nodes**: To provide multiple communication links for experiments all NNE nodes are connected to more than 2 MBB providers and fixed or wireless networks.

- **Rich context information**: NNE nodes include the built-in support system to collect the detailed information of whole network system like cell ID, mode of connection, signal strength etc., location and time of the measurement experiment.

- **Advanced system for experiment management**: NNE consists of a central server for storing and transferring the measurement's result from the developer who involved in taking experiments.

2.6.1 Multi-Homing [37, 38]

Multi-homing is a technique to provide reliable and enhanced Internet connectivity in a network system without any performance degrading. It is configured in a system by defining multiple IP addresses and assigning more than one network interface in one computer. Multi-homing reduces the probability to disconnect the entire network system if one of the connections goes down. In order to increase the robustness of Internet connectivity of a network system, endpoints have to connect with multiple Internet service providers (ISPs) together so does multi-homing. Furthermore multi-homing allows performing load-balancing by decreasing the number of systems connecting to the Internet via any single connection. Distributing the load over various network connections increase the performance and decrease the wait times.

2.7 Packet Scheduling Algorithms

Packet scheduling is the process of assigned the shared transmission resources to the appropriate users at a given time to achieve performance guarantee. It is expected that, using proper packet scheduling algorithms the transmission of a resources in packet form over links will possibly make greater resource utilization than traditional circuit-based communications. The packet-switched environment is therefore accepted in most efficient system nowadays. Still, it may be lead to critical problems when different packets combine with Quality of Service (QoS) such as jitter, throughput required, tolerated delay etc. or size of packets competing for the finite resource transmission. When the traffic load in network is comparatively heavy, to fulfill the QoS requirements of individual users, the conventional first-come-first-serve process may no longer be an efficient way to use the feasible transmission of resources. In such case, the applicable packet scheduling algorithms are designed to manage the order of packet

transmission, which will help to obtain the required different QoS of users or adjust fairness of a system for increase services performance and the system capacity. From above advantages and benefits Packet Scheduling Algorithm is becoming the most essential function in communication network technology both wired and wireless.
The list of properties of packet scheduling algorithms is described below [**36**]:

- **Efficiency**: Packet scheduling algorithm's main function is to manage the order of packet transmission in the network system in such a way that it achieves each user QoS requirements. Packet scheduling algorithm is more efficient than other mechanisms if it can support larger region of capacity and also it can manage the same QoS guarantee for individual user even the amount of traffic is huge or there is more served users.

- **Protection**: After efficiency to get QoS guarantee, another important property of packet scheduling algorithm is protection. The flow of the network has to provide an individual virtual links so that the traffic characteristics of a single flow link will have less effect to the quality of service of other flows as achievable. In many packets scheduling context, the property Protection is sometimes indicated as flow isolation. Flow isolation can help the network system to provide QoS guarantee by flow-by-flow technique, which are not dependent of the other flows with traffic demand. Flow is a logical unit in a network system, which means sequence of input packets and is associated with input packets. When users send packets with higher rate rather than they mentioned the per-flow QoS guarantee could prevent to be degraded. Services with performance guarantee can also be allowed by dividing the users logically, which are responsible to get wide range of traffic characteristics and QoS requirements while supplying protection from each other.

- **Flexibility**: Packet scheduling algorithm is cable to support various users with different network traffic characteristics and QoS requirements. Nowadays, the most practical network integrated system has to provide applications with broad difference of performance requirements and traffic characteristics, which are, provided by packet scheduling algorithm.

- **Low Complexity**: Due to the improvement of transmission rate and bandwidth of communication system in recent scenario, processing speed of packets must match the requirement that's why it is becoming more critical. Packets scheduling algorithms is responsible for processing packets through a network so it is becoming important concern and have acceptable computational complexity to be implemented.

2.7.1 Packet Scheduling Algorithm in Wired Systems [36]

This system is based on the physical medium communicate between packets which gives a stable and robustness in system. Here systems are directly connected to the medium so that packet error rate (PER) should be ignored and link capacity with unit bits/sec will be constant. In literature, this model is mentioned as error-free channel. There are several packet scheduling algorithms in wired systems are introduced such as First Come First Serve (FCFS), Round Robin (RR), Strict Priority, Earliest Deadline First (EDF), Generalized Processor Sharing (GPS), Packet-by-packet Generalized Processor Sharing (PGPS) etc.

2.7.2 Packet Scheduling Algorithm in Wireless Systems [36]

Wireless network is divided with their range of transmission. Short transmission range that will be some meters (ten) range like wireless local area network (WLAN) and macrocell. Long transmission range with wide range of about thousands of meters or even several kilometers like environments of femtocell based on wideband code division multiple access (WCDMA), worldwide interoperability for microwave access (WiMAX) and long term evolution (LTE). In wireless medium, packet transmission will be affected by the location that may occur packet loss, fading and shadowing which makes no ignorable of PER and also link capacity may vary. In literature, this model is mentioned as error-prone channel. There are several packet-scheduling algorithms in which wireless systems are introduced such as Idealized Wireless Fair Queuing (IWFQ) algorithm, Improved Channel State Dependent Packet Scheduling (I-CSDPS) algorithm, Channel-condition Independent packet Fair Queuing (CIF-Q) algorithm, Proportional Fair (PF) algorithm etc.

Approaches	Mechanism	Advantages	Disadvantages
Indirect TCP	splits TCP connection into two connections	simple, isolation of wireless link	higher latency at handover, loss of TCP semantics
Snooping TCP	"snoops" data and acknowledgements, local retransmission	transparent for end-to-end connection, MAC integration possible	bad isolation of wireless link, problematic with encryption
M-TCP	splits TCP connection, chokes sender via window size	handles long term and frequent disconnections, maintains end-toend semantics	processing overhead due to bandwidth management, bad isolation of wireless link
Fast retransmit/fast recovery	avoids slow-start after roaming	simple and efficient	mixed layers, not transparent
Transmission /time-out freezing	freezes TPC state at disconnect, resumes after reconnection	independent of content or encription, works for longer interrupts	changes in TCP required, MAC dependent
Selective retransmission	retransmit only lost data	very efficient	slightly more complex receiver software, more buffer needed
Transaction oriented TCP	combine connection setup/release and data transmission	efficient for certain applications	changes in TCP required, not transparant

Table 2.3: Overview of classical TCP enhancements for mobility

Generation	Requirements	Comments
1G	No official requirements. Analog technology	Deployed in 1980s
2G	No official requirements. Digital technology	First digital systems. Diployed in the 1990s. New services such as SMS and low-rate data. Primary technologies includes IS-95 CDMA and GSM
3G	ITU's IMT-2000 required 144 kbps mobile, 384 kbps pedestrain, 2mbps indoor	Primary technologies. Include CDMA2000 1X/EVDO and UMTS-HSPA. WIMAX now an official 3G technology.
4G(Initial technical designation)	ITU's IMT-Advanced requirements include ability to operate in upto 40 Mhz radio channels and with very high spectral efficiency.	No commercially deployed technology meets requirements today. IEEE 802.16m and LTE-Advanced being designed to meet requirements.
4G(Current marketing designation)	Systems that significantly exceed the performance of initial 3G networks. No quantitive requirements	Today's HSPA+, LTE, and WIMAX networks meet this requirements.

Table 2.4: 1G to 4G [**29**]

Chapter 3

Approach

3.1 System Setup

As mentioned in the problem statement, this research project aims to consider the design of the communication system that schedules different applications to different links. Note that different applications require different transmission protocols. The main focus of this thesis is, however, TCP based applications. In this content, we further study the multi-path extension of TCP, namely MPTCP 2.3.2. On the other hand, UDP 2.3.3 is widely used for video streaming type of applications, therefore, we also consider UDP and provide support with video streaming through FFMPEG player. However, the majority of the work carried out in this thesis has been done on TCP, therefore, the remainder of this thesis, the main focus (including the results section) will be on TCP and MPTCP.

3.1.1 TCP based Applications

For TCP based applications, we consider that we have two different applications that has to be transmitted using TCP over two different paths. First, the applications has been generated at the client (to be placed in the ambulance). This is achieved through the scripts that is running on the client. This script generates TCP packets with a packetID to differentiate different applications and sending timestamp for further use and analysis. Two applications in the client are generated with varying data transmission rates mimicking applications with different bandwidth requirements. A laptop is used as the client. All these packets are then transmitted to the Nornet edge node containing multiple interfaces that are connected to different operators (Telenor and Netcom). Client computer and Node are connected with crossover cable and assumes to be placed in the ambulance in far distance, whose main aim is to transmit received packets from laptop to the third section server with multiple options. The node transmits all the received packets to the server (to be situated in the Hospital) using multiple paths. In this thesis, we consider three different transmission protocols to transmit received packets to server: normal TCP, default MPTCP, and proposed MPTCP with selected interface (MPTCP-MS). Normal TCP selects the interface with miminum metric and transmits over that path. The default MPTCP multiplexes the received packets of different applications and schedules these packets randomly to different paths while the proposed method selects a specific interface for each application. As we discussed in the MPTCP section, to communicate with MPTCP option both end point must have enabled MPTCP functionality otherwise the communication works as in normal TCP. Finally, the server receives and saves all the packets that are sent from client via node. These packets are then used for further analysis.

We assume that the client and server are in far distance location and the communication is made via different MBB providers, the 3G wireless communication provided by Telenor and Netcom service providers. The packets coming from two different applications with different links are randomly received in server. It separates and manages the received packets according to the application by specific packetID and also adds the received time stamp. The main aim of this research is to find the effects and characteristics of network delay performance in transmission with different applications with different packets rate through different links. Specifically, we consider one-way delay 3.4.1 that is the difference between the sending time and received time.

3.1.2 UDP based Applications

In Telemedicine system, it is very critical to send patient's information to the medical professionals at correct time or without any delay. There may be tolerable in loss but delay is very sensitive in patients information otherwise treatment will be affected. For the transmission of delay sensitive and loss tolerance data like video, audio, multimedia files etc. we use UDP protocol. The UDP is faster than TCP because UDP doesn't contain flow control, congestion control and handshaking mechanisms. For video and audio streaming, we have explored one of the popular open-source tool i.e. ffmpeg 2.4.1. It enables to convert, record and stream the audio and video files, almost all of the formats are supported by this tool. The general guide for Compilation and Installation ffmpeg in Linux system (Ubuntu) is given in the link [**49**]. It also installs required components and libraries including ffserver 2.4.1 and ffplay 2.4.1.

We can use ffplay for stream a video that was given by ffmpeg as input using UDP protocol. The syntax for ffmpeg is given by:

ffmpeg -i filename.avi -f mpegts udp://ip_of_machine:port

Where -i option gets the input file as filename.avi and it streams to the assign port and ip_of_machine. To play video in ffplay the syntax is given below:

ffplay udp://ip_of_machine:port

To Stream a video using ffserver we have to edit configuration file i.e. ffserver.conf. ffserver only works by forwarding any streams that was encoded by ffmpeg. First of all, ffserver need to debug the configuration file using following command:

ffserver -f ../ffserver.conf

If it is error free then, we are ready to stream file. The sample stream written in configuration file for mp3 format is given below:

```
<Stream test_stream.mp3>
Feed test_feed.ffm
Format mp2
AudioCodec mp3
AudioBitRate 64
AudioChannels 1
AudioSampleRate 44100
NoVideo
</Stream>
```

To stream this feed we have to encode the input file with ffmpeg and can be done as:

ffmpeg -i filename.mp3 http://ffserver_ip:port/test_feed.ffm

Afterwards, we can stream that feed from media player by giving a url like:

http://ffserver_ip:port/test_stream.mp3

3.2 Tools and Scripts

3.2.1 Software

Python programming language is chosen for writing programs. Python is chosen because of its characteristics; easy to learn and use and also it is platform independent. Python is a high level object-oriented programming language and contains huge and comprehensive standard libraries. For statistics study and data analysis purpose R programming language is chosen. R is a powerful free and open-source programming language for statistics computations and designing a graphics or plot the data for further analysis. In this thesis, using R programming language we generate various graphs like histogram, box-plot, CDF graph for plotting the data and analysis.

Here are the lists of general python scripts used in this thesis with its functionality:

- pathinfo.py[1]
  This script provides the information about the network path and RRC by the carrier's in-terms of MODE, SUBMODE, RRC. MODE will be No service, GSM, WCDMA, and LTE. We can change the MODE 3G to 2G and vice-versa using this script. Another function of this script is to map the MCC (Mobile Country Code) to MNC (Mobile Network Code) to uniquely identify mobile broadband providers. Here it is done as: 'ppp0': '24201', 'ppp1': '24202', 'ppp2': '24007', 'ppp3': '24205' . So, we can say that ppp0 is Telenor with MCC 24201, ppp1 is Netcom with MCC 24202.

- interfaceinfo.py[2]
  This script is used to get the information of the interfaces available in the system. It retrieves name of the interface, gets the information of current state of the interface.

- mptcp_sockopts_client.py[3]

[1] https://github.com/grungerambow/test/blob/master/pathinfo.py
[2] https://github.com/grungerambow/test/blob/master/interfaceinfo.py
[3] https://github.com/grungerambow/test/blob/master/mptcp-ms-client.py

This is the main script that gives various MPTCP options for specific TCP sockets. It gives the detail information of TCP subflows created by socket. It uses various library files to get the information. We can set the rating of interfaces as we want, rating 255 is disabled and 0 means interface enabled so that we can choose the specific interface for data transmission. Other various functions are available to set the MPTCP options in sockets. This script is placed in the client side and used by the measurement scripts.

- mptcp_sockopts_server.py[4]
  This is similar script as mptcp_sockopts_client.py, all the functionality is also same but this script was placed in the server side. Some of the selected functions are used in this section.

3.2.2 Hardware

As this thesis work is based on communication system, it uses several hardware components i.e. Laptop, crossover cable, modem with different carriers, Nornet edge node etc. The specification of hardware are described in table 3.1

3.3 Proposed Method

In this section, we will discuss the details of the proposed algorithm that enables selection of different interfaces for different applications in order to reduce the delay. We will first describe the algorithm and then the implementation in the MPTCP linux kernel.

3.3.1 Algorithm

The working logic of the proposed method is shown in flowchart 3.1 and figure 4.6. We are targeting to transmit different applications data through different communication links based on different delay requirements. Different applications in Telemedicine system may require sending multiple streams at the same time and it is critical that important files have to be sent at time without any delay. This system is proposed

[4]https://github.com/grungerambow/test/blob/master/mptcp-ms-server.py

Hardware	Specification
Laptop	Processor: Intel® Core™ i5-4200U CPU @ 1.60GHz × 4 Memory: 6GB OS: Linux, 3.11.0-19-generic-64bit
NorNet Edge node	Processor: Samsung S5PV210 Cortex A8 1GHZ processor Memory: 512MB RAM, 512MB NAND flash memory,16GB SD card OS: Linux,3.11.7norarm-gf87a1cd-dirty Other: 1 - 4 3G (UMTS) modems, 1 CDMA (1x EV-DO) modem, TP-LINK TL-WN822N USB Wi-Fi module
Server (VM)	OS: Linux, 3.11.10+
Modems	Huawei mobile broadband E353 HSPA+ USB stick Carriers: Telenor and Netcom

Table 3.1: List of hardware components used

to get the functionality to transmit multiple streams at the same time with minimum delay. We can send the most critical file through the path as we want. The connection path with high bandwidth and less delay can be identified and we can send the delay sensitive files through that link.

Two applications at client generate the packets with packet ID (PID1 and PID2) and transmit to the node with sending timestamps. The script in the Nornet edge node then checks the packet ID and sends the packets to the server on the selected link. Here packets with packetID 2 are transmitted via ppp1 (Netcom) and the packets with packetID 1 go through ppp0 (Telenor) to the server. The server (receiver) receives the packets and notes the receiving timestamps. Then OWD is calculated according to the sending and received timestamps for specific packets.

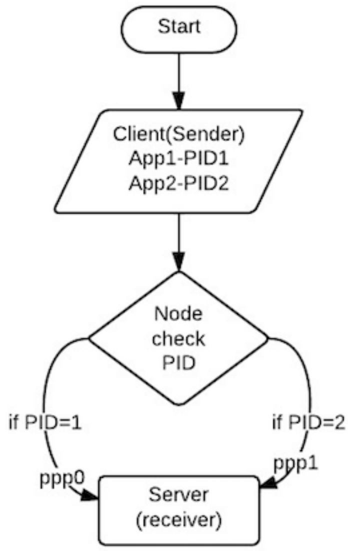


Figure 3.1: Flowchart of proposed method

3.3.2 Implementation on MPTCP Kernel

MPTCP Installation and Configuration

We can install MPTCP in several ways all processes and steps are mentioned in this link [**44**]. We can manually compile and install by retrieving source and also can install automatically with the apt-repository provided by the developers. After successfully installed, we need to configure the routing table [**45**] for defining and using several interfaces which one we want to use. To achieve the control in use of the specific interface and gateway rather than default one, we need to configure a numbered routing table per outgoing interface. The selection of the route is done in two phases, first the kernel of the system does a lookup in policy table, which we need to configure with ip rules and secondly the gateway is selected based on the destination address for the corresponding routing table. Now we can configure the MPTCP, for this we need to set a several sysctl variable and the syntax is given as:

sysctl -w net.mptcp.[name of the variable]=[value]

- net.mptcp.mptcp_enabled : To Enable/Disable MPTCP, we can set net.mptcp.mptcp_enabled with 1/0.

- net.mptcp.mptcp_checksum: To Enable/Disable the MPTCP checksum, we can set net.mptcp.mptcp_checksum with 1/0.

To set the path-managers at the run-time, we need to set the sysctl net.mptcp.mptcp_path_manager with different variables:

- 'default': this option for path-manager doesn't initialize different IP-addresses nor create new subflows.
- 'fullmesh': this option creates the full-mesh of subflows between all possible subflows.
- 'ndiffports': this option creates number of subflows (X) across the same pair of IP addresses and this was done by modifying the source-port. We can set this option greater than 1 if we want to control the subflows (X) number.

After successfully and correctly installing the MPTCP-kernel and configuring routing table as our requirements, we can run the MPTCP enabled system. To check whether MPTCP is installed and to list the established connections, we can simply run following command netstat -m, which gives list of active MPTCP connections.

3.3.3 Modification on MPTCP Kernel

The main goal of our project is to control over sub-flows, which we want to send packets based on packet ID in the header. We are not making multiplex packets according to the scheduler as default MPTCP. To get this functionality we have modify the MPTCP kernel as our requirements. We have added #define TCP_ALLOWED_SUBFLOW 35 to file /include/uapi/linux/tcp.h. 35 is chosen because it was the next free value in our kernel. In our main script, send_on_interface function iterates through the subflow 'path index' this is a number that MPTCP gives to each of the subflows. Whenever it finds, for example, ppp0 it sends packets with packetID 1 and when it finds ppp1 it sends packets with packetID 2, and it has been shown as below:

```
setsockopt(socket, TCP_ALLOWED_SUBFLOW, 1)
send(socket, "ABC") # packets with packetID 1
setsockopt(socket, TCP_ALLOWED_SUBFLOW, 2)
send(socket, "DEF") # packets with packetID 2
```

```
send(socket, "XYZ") # default MPTCP transmission
```

In pro setsockopt we trigger a 'state', over which subflow, from now on, packets will be send. This is valid until the next change that means, when a packet from other ID comes in and this is saved in the kernel variable 'allowed_subflow' (this variable exists in pro socket). When data goes down to the kernel, the data is delivered to the kernel in a 'socket buffer' (an interface between user and kernel space) and this data is marked with the current 'allowed_subflow' value (we 'glue' this value to the data). Then, at some point, when the MPTCP scheduler is 'asked' on which subflow the data in the buffer has to be sent, the scheduler respects the 'marking' on the data. In our main code, if 'allowed_subflow' = 0, the scheduler works normally, as MPTCP is originally designed. The term 'marking' the packets means, we 'mark' the data saying that it should go out on subflow X, in this way, the socket buffer is already sent on all other subflows, but NOT on subflow X. This is because MPTCP scheduler sends the data ONLY once on a given subflow. And if we say it was already sent on the others, only the subflow we want to send data is actually left. We also switch off the retransmission because, otherwise, MPTCP could send data doubled on the subflow.

3.4 Measurement Methodology

3.4.1 Data of Interest

We are mainly interested in the delay performance of the system, therefore we focus on one-way delay throughout this thesis.

One-Way Delay (OWD)

In the field of computer networks, it is critical to evaluate the network performance. The most important network performance parameters according to the quality of service of the network communication is the one-way delay (OWD) which gives more information than round-trip time (RTT). It is quite difficult to measure the OWD, various factors can be affected such as synchronization uncertainties, the operating system, the threads that are synchronous to the application measurement. OWD gives a better characterization of the network parameters. The measurement of OWD in the network communication is important to improve and guarantee the performance of real-time applications like video conferencing and voice over Internet protocol. This thesis is also based on the real-time application with critical medical file transmission so we have to minimize the OWD as possible as we can for reliable and guarantee file transmission.

Network delay is the main factor to evaluate the performance of networks. Network delay includes three components equipment delay, transmission delay and propagation delay [**46**]. Equipment delay is the delay that is introduced by the equipment before it transmits through the equipment used in transmission system. It includes packet switching, processing time, and queuing delays and also depends on congestion and the total load in network. Transmission delay is the time that takes to transmit whole packets that depends on data rate, distance and media used in a network system for packet transmission. Propagation delay is the time between the time taken to send the first bit of packet by sender equipment and receive that bit in receiver equipment. These three delays gives the time taken to travel sending packets from source to destination and this time is known as OWD [**47**]. The OWD is the time between the emission of the first bit of a sending packet by the sender and the last bit of that packet received by the receiver. In other word, sum of the transmission and propagation delay is OWD [**48**] and defined in the figure 3.2.

It is difficult to get the actual OWD because the synchronization between the clocks of source and destination cannot be guaranteed. To achieve this, different solutions are proposed and the search for effective way is still in progress. There are other several factors such as packetization, operating system, interruptions, packet compression etc. that can affect the sender timestamps and receiver timestamps in the packet even if the clocks are correctly synchronized so the OWD may varies. In this project for time synchronization one python script timesyn.py is created and start the script before running the experiments and tests for measurements in both client laptop and server. timeesyn.py include following commands:

```
#!/usr/bin/python
import subprocess
subprocess.call("/etc/init.d/ntp stop", shell=True)
```

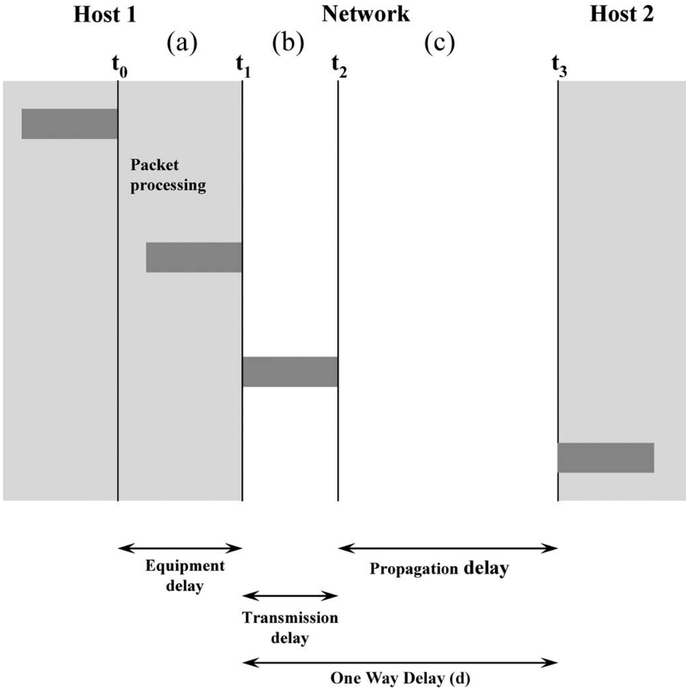


Figure 3.2: a) Equipment delay, b) Transmission delay, c) Propagation delay, and d) One-way delay

subprocess.call("pkill ntpd", shell=True)
subprocess.call("ntpdate 129.240.2.6", shell=True) #ntp1.uio.no
subprocess.call("/etc/init.d/ntp start", shell=True)

This script synchronized both machines (client and server) to the same NTP server (in this case ntp1.uio.no). This process is not eliminating the whole time synchronization problem but it will minimize it. Even we use NTP time synchronization there is more issue and delay may be difference than actual. To avoid this, we use OWD normalization concept in this thesis. OWD normalize was computed by subtracting minimum delay with each calculated OWD and divide obtained value by the minimum delay.

3.4.2 Methodology

To measure the one-way delay, different applications with different packets rate are transmitted through different connection mechanism and links (MBB providers) and OWD is calculated as described above. The setup for this will be: Client computer and Node with MPTCP installed are connected with crossover cable and assumes to be placed in the ambulance. Server is a machine with MPTCP installed which receives packets and assumes to be in the Hospital. DHCP server is installed in client machine and it gives same range private IP for node so that it can communicate each other. Client machine doesn't connect with external Internet providers. Two types of application are generated by client and transmitted to the node. Two different TCP sockets with different port are maintained. Initially, node acts as a server, which receives the packets coming from client. The rate and size of sending packets varies with type of applications. While sending packets client adds the packet ID and sending timestamps in the header of each packet. For e.g. Application 1 gets the packet ID as 1 and application 2 get packet ID as 2. Node consists of two modems with Telenor (ppp0) and Netcom (ppp1) carriers that provide Internet connection. We can change the carriers from 3G to 2G and vice-versa for experiments. Now node works as a client and received packets then transfers to the server simultaneously.

The experiments and measurements are carried out in 3 scenario: scenario 1: Using normal TCP connection, scenario 2: Enabling MPTCP and scenario 3: Selected interface with enabling MPTCP, and our proposed method is scenario 3 figure 4.6. The server received the packets and adds the received timestamps and separates the packets according to the packet ID and saves in different files. This file now used for measuring the OWD of the whole system.

3.5 Methods Used to Analyze Data

For visually representing the data for statistics and analysis with respect to the OWD, following techniques are used.
The histogram is the graphical representation of collected data sets and plots them, that help to analyze and get the variation of data. Boxplot is a quick way of graphically representation of data sets through a quartile. Boxplot displays the full range of variation from minimum to maximum value and the median value. The rectangle in the boxplot is the first to third quartile known as Interquartile range (IQR), inside this it contains median and maximum and minimum as "whiskers" above and below the box. The value more above the third quartile and more below the first quartile is known as Outliers.

3.6 Alternative Approach

The alternative possible approach for cover multiple addresses to the transport layer will be using SCTP (Stream Control Transmission Protocol). SCTP also supports multiple IP addresses per connection and include multiple paths functionality. SCTP is mainly developed for signaling in telephone networks. SCTP is capable of multi-homing as well as multi-streaming. SCTP is message-oriented protocols, however, MPTCP is byte-oriented protocol. The transmission mechanism of SCTP is, it bundles different small messages into a single packet and transmits and it also avoids overhead problem in network system. To increase the payload data throughput by utilizing the multiple network paths the extension of SCTP i.e. CMT-SCTP (Concurrent Multiple Transfer SCTP) is used. The CMT-SCTP allows a load sharing functionality by using all available network paths as MPTCP. The reason for not choosing SCTP in our research is from its drawbacks i.e. NAT (Network Address Translation) boxes and firewalls are unable to process SCTP packets and they'll discard them [**50**] and SCTP exposes different API from the socket API to the applications in network system.

Chapter 4

Result and Analysis

This chapter discusses in detail about the procedure and methodology used for the tests and the results obtained from the experiments followed by the analysis of obtained result.

4.1 Experiment Setup

The experiment setup and results are carried under 3 different scenarios that are described below:

4.1.1 Scenario 1

In this scenario the communication between server and node will be in normal TCP connection. The structure is given in figure 4.1. Two

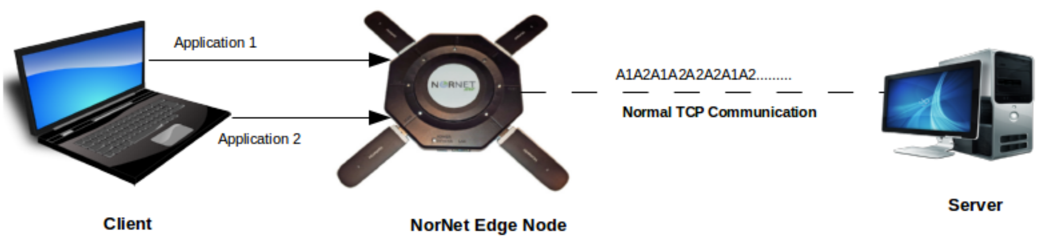


Figure 4.1: Experiment setup with normal TCP connection

different applications generate different TCP packets with different size for transmission. Each application assigns with a packet ID and adds the timestamps in the packet header before transmitting. client.py[5]

[5] https://github.com/grungerambow/test/blob/master/client.py

and client1.py[6] are two scripts that generate the TCP packets with the specified packets rates and packet ID. In this experiment, we use different packet rates for different application i.e. 2 and 10 packets per second for application 1; 5 and 25 packets per second for application 2. Both application need to run in same time, so we have manage one python script main_client.py that executes both application at the same time and code is looks like:

```
#! /usr/bin/python

import os
import time
os.system("./timesyn.py")
os.system("./client.py 192.168.2.53 9090 1&")
os.system("./client1.py 192.168.2.53 9091 1&")
```

Before starting the experiment, we must run the script timesyn.py for the time synchronization purpose.
In the node script receive_send.py[7] was executed. It was used as a mediator or gateway for transmission TCP packets coming from client to server. In this scenario, the server and node are connected through normal TCP connection. The TCP packet transmission is normally done in the way of traditional TCP connection. TCP communication starts with interface with lowest metric. In this case it uses ppp0 interface to get connection and communicate. The node and server's kernel are MPTCP installed and configured so we need to disable MPTCP in this scenario and that can be done by executing following command:

sysctl -w net.mptcp.mptcp_enabled=0

In server side we execute the server.py[8] script, which is listing in specific port number. When the TCP connection is established with node it starts to receive packets. It adds received timestamps and separates the coming packets with respective packet ID. The experiments are carried out for 1 minute and collect 10 samples in the interval of 10 minutes. Finally, we have two files with different packet ID and packet information i.e. packet size, sending timestamps, packet ID and received timestamps. These

[6] https://github.com/grungerambow/test/blob/master/client1.py
[7] https://github.com/grungerambow/test/blob/master/receive_send.py
[8] https://github.com/grungerambow/test/blob/master/server.py

attributes are used for further analysis based on the OWD. Histograms, box-plot are used to plot the records and analyze OWD.

4.1.2 Scenario 2

In this scenario, the communication between server and node will be an MPTCP enabled connection. The structure is given in figure 4.2. The

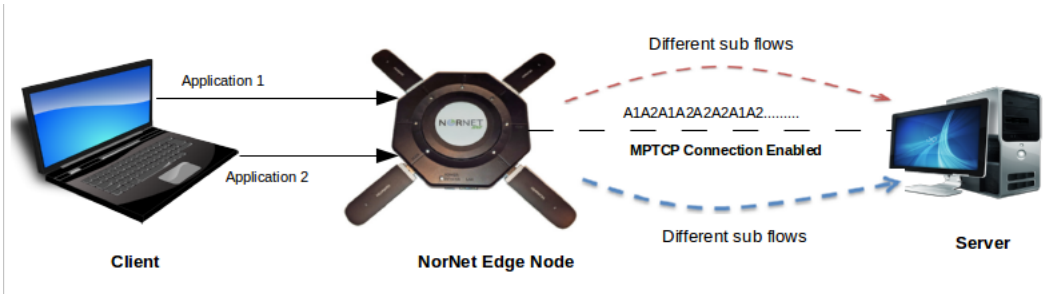
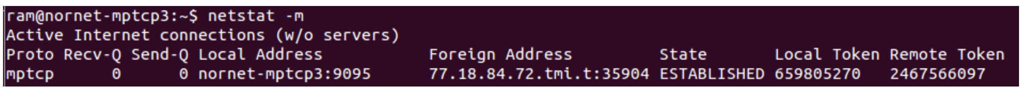


Figure 4.2: Experiment setup with MPTCP connection enabled

procedure of this test experiment is similar to above scenario 1; only difference is both client and server are enabled MPTCP connection for transmission of TCP packets and it can be done by executing following command:

sysctl -w net.mptcp.mptcp_enabled=1

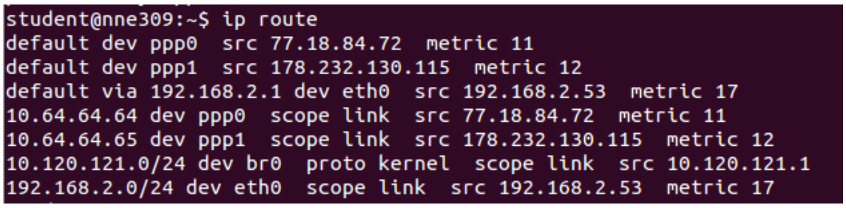
we can check the list of active MPTCP connections by executing command netstat –m figure 4.3. MPTCP always starts with the interface with the



```
ram@nornet-mptcp3:~$ netstat -m
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       Local Token Remote Token
mptcp      0      0 nornet-mptcp3:9095      77.18.84.72.tmi.t:35904 ESTABLISHED 659805270   2467566097
```

Figure 4.3: List of active MPTCP connections using netstat –m command

lowest metric, then it will open a second connection on another available interface. We can check the lowest metric with ip route command fig 4.4. In our case ppp0 contain lowest metric i.e. 11. Afterwards, both server and client have MPTCP enabled, MPTCP will open an extra TCP connection on each extra interface of the client (Node). Enabling MPTCP, it creates multiple virtual sub-flows, so that packets can transmit via these sub-flows. The transmission of packets utilizing MPTCP is illustrate in figure 4.5: Figure 4.5 depicts a packet transmission with two clients one with 10pkts/s and another with 25pkts/s through the node with two active interfaces



```
student@nne309:~$ ip route
default dev ppp0  src 77.18.84.72  metric 11
default dev ppp1  src 178.232.130.115  metric 12
default via 192.168.2.1 dev eth0  src 192.168.2.53  metric 17
10.64.64.64 dev ppp0  scope link  src 77.18.84.72  metric 11
10.64.64.65 dev ppp1  scope link  src 178.232.130.115  metric 12
10.120.121.0/24 dev br0  proto kernel  scope link  src 10.120.121.1
192.168.2.0/24 dev eth0  scope link  src 192.168.2.53  metric 17
```

Figure 4.4: List of interfaces in node using ip route command

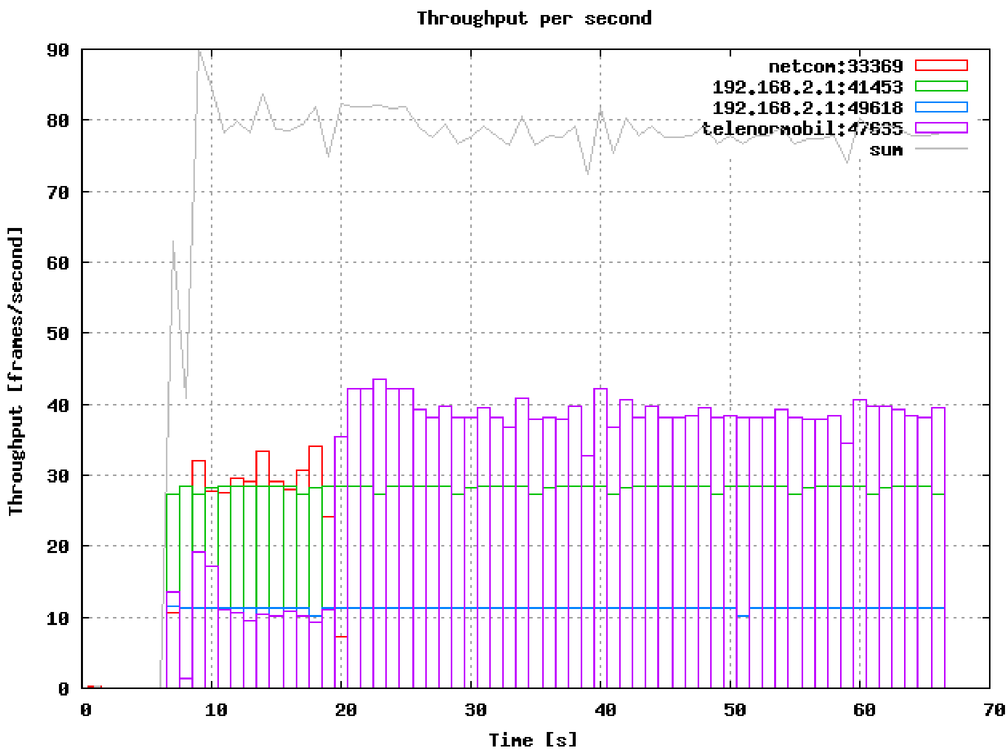


Figure 4.5: Throughput graph for packet transmission utilizing MPTCP enabled

(Netcom, Telenor) to the server. It shows that initially the incoming packets are trying to send on Telenor (10) and Netcom (25) packets but after 20 seconds it switches to only Telenor (about 35-40 pkts/s). This is the functionality of the default MPTCP enable communication system. Initially MPTCP creates connection within available interfaces, and after it selects the path with minimum RTTs.

4.1.3 Scenario 3

In this scenario, the communication between server and node will be in MPTCP connection. Additionally, the transmission of TCP packets are through selected interface. The structure of the proposed solution is given in Figure 5 and identified as MPTCP-MS in all graphs. The procedure

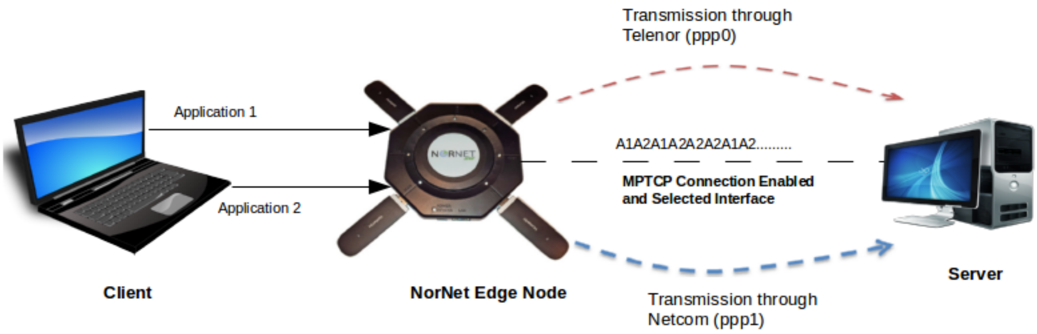


Figure 4.6: Experiment setup with MPTCP connection enabled and transmission through selected interfaces (MPTCP-MS)

for generating and sending TCP packets with different application from client to node is similar as above scenarios. The addition functionality is added in the node, so that the packets coming from client with different packet ID is distributed through the different interface. Here also both node and server's kernels are MPTCP enabled and can be done by executing following command:

sysctl -w net.mptcp.mptcp_enabled=1

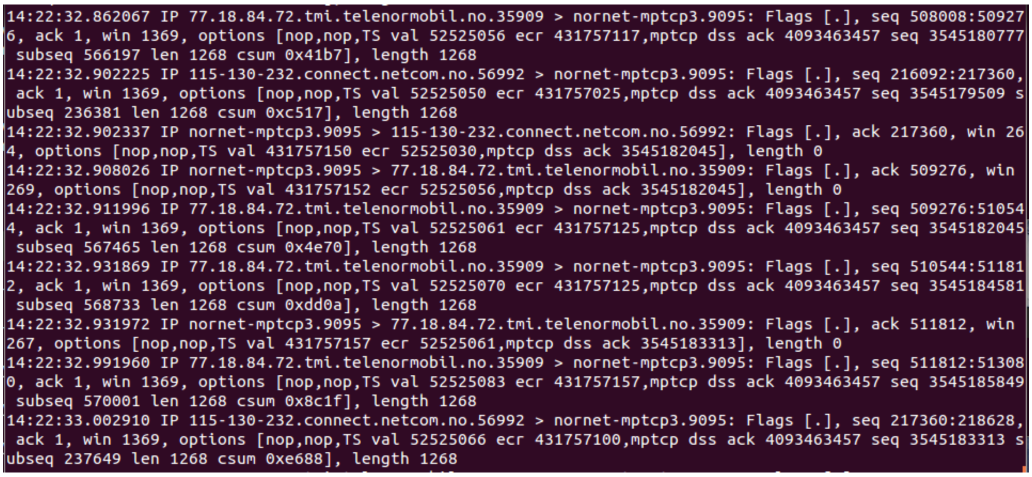
To check the list of MPTCP connection, we can execute netstat -m command 4.3.

In this experiment, we execute receive_send_iface.py[9] (main) script in node, which allows the transmission of packets with respective interface.

[9] https://github.com/grungerambow/test/blob/master/receive_send_iface.py

We have modified the kernel of node's MPTCP configuration and made a special function in the main script to transmit the packets through desired interfaces. We have mention and check packet ID and send the packets through specific interface. Application 1 with packetID 1 goes through the interface ppp0 (Telenor) and the application 2 with packetID 2 goes via interface ppp1 (Netcom). Server receives packets in random order and separates the packets with specific packetID and adds received timestamps for further analysis.

Scenario 3 can be visualized in the Figure 4.7. Figure 4.7 demonstrates the transmission of data between client (Node) to server, it is collected in the server side by tcpdump command. It clearly shows that two links are established during packets transmission as our requirements. We have set different transmission rates while sending packets from client to server in different links (in this case ppp0 (Telenor) and ppp1 (Netcom)). To



```
14:22:32.862067 IP 77.18.84.72.tmi.telenormobil.no.35909 > nornet-mptcp3.9095: Flags [.], seq 508008:50927
6, ack 1, win 1369, options [nop,nop,TS val 52525056 ecr 431757117,mptcp dss ack 4093463457 seq 3545180777
 subseq 566197 len 1268 csum 0x41b7], length 1268
14:22:32.902225 IP 115-130-232.connect.netcom.no.56992 > nornet-mptcp3.9095: Flags [.], seq 216092:217360,
 ack 1, win 1369, options [nop,nop,TS val 52525050 ecr 431757025,mptcp dss ack 4093463457 seq 3545179509 s
ubseq 236381 len 1268 csum 0xc517], length 1268
14:22:32.902337 IP nornet-mptcp3.9095 > 115-130-232.connect.netcom.no.56992: Flags [.], ack 217360, win 26
4, options [nop,nop,TS val 431757150 ecr 52525030,mptcp dss ack 3545182045], length 0
14:22:32.908026 IP nornet-mptcp3.9095 > 77.18.84.72.tmi.telenormobil.no.35909: Flags [.], ack 509276, win
269, options [nop,nop,TS val 431757152 ecr 52525056,mptcp dss ack 3545182045], length 0
14:22:32.911996 IP 77.18.84.72.tmi.telenormobil.no.35909 > nornet-mptcp3.9095: Flags [.], seq 509276:51054
4, ack 1, win 1369, options [nop,nop,TS val 52525061 ecr 431757125,mptcp dss ack 4093463457 seq 3545182045
 subseq 567465 len 1268 csum 0x4e70], length 1268
14:22:32.931869 IP 77.18.84.72.tmi.telenormobil.no.35909 > nornet-mptcp3.9095: Flags [.], seq 510544:51181
2, ack 1, win 1369, options [nop,nop,TS val 52525070 ecr 431757125,mptcp dss ack 4093463457 seq 3545184581
 subseq 568733 len 1268 csum 0xdd0a], length 1268
14:22:32.931972 IP nornet-mptcp3.9095 > 77.18.84.72.tmi.telenormobil.no.35909: Flags [.], ack 511812, win
267, options [nop,nop,TS val 431757157 ecr 52525061,mptcp dss ack 3545183313], length 0
14:22:32.991960 IP 77.18.84.72.tmi.telenormobil.no.35909 > nornet-mptcp3.9095: Flags [.], seq 511812:51308
0, ack 1, win 1369, options [nop,nop,TS val 52525083 ecr 431757157,mptcp dss ack 4093463457 seq 3545185849
 subseq 570001 len 1268 csum 0x8c1f], length 1268
14:22:33.002910 IP 115-130-232.connect.netcom.no.56992 > nornet-mptcp3.9095: Flags [.], seq 217360:218628,
 ack 1, win 1369, options [nop,nop,TS val 52525066 ecr 431757100,mptcp dss ack 4093463457 seq 3545183313 s
ubseq 237649 len 1268 csum 0xe688], length 1268
```

Figure 4.7: Visualization of data transmission through two different interfaces (Netcom and Telenor)

give more details about our proposed solution, figure 4.8 highlights the transmission of different packets rates in different links. Here two clients with IP 192.168.2.1 with different ports generates the packets with rate 10 and 25 packets/sec, and both applications are transmitted through different links, 10pkts/s through Telenor and 25pkts/s through Netcom. This is just one sample example for 10 and 25pkts/s. Similarly, we have done for other rates (2 and 5 pkts/s) in experiments. All the experiments are done with two different packet size (1024 and 500 Bytes) and different rates for packet transmission (2,5,10 and 25 packets/sec). After successfully

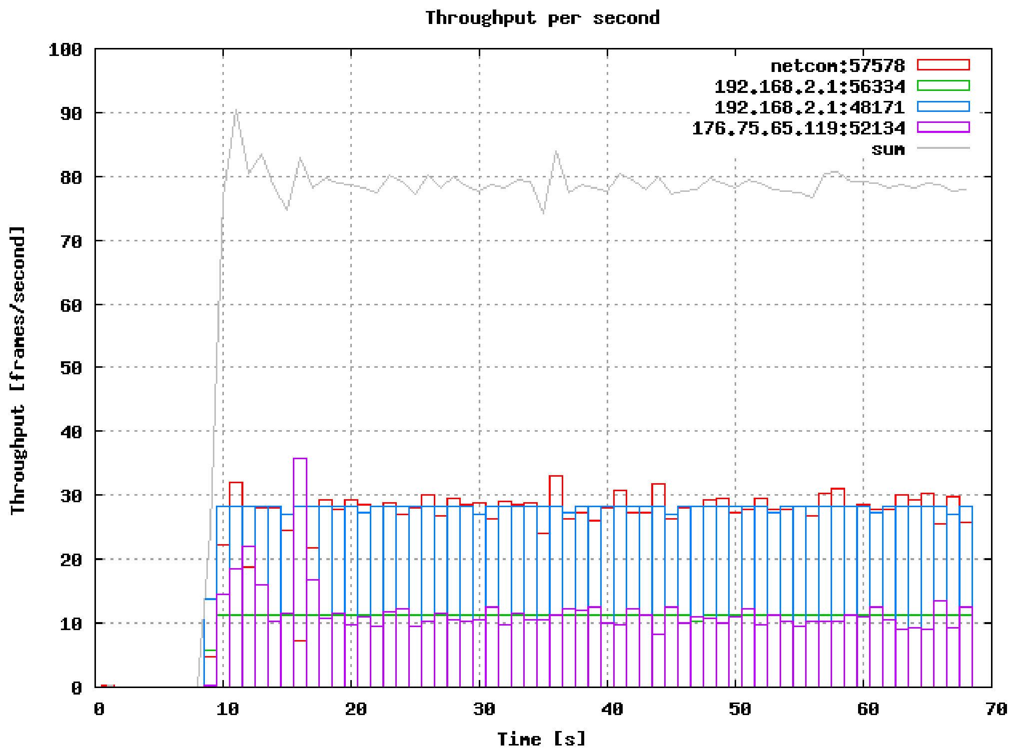


Figure 4.8: Throughput graph for packet transmission utilizing MPTCP enabled with selected interfaces

packet transferred, in server side one-way delay (see section 3.4.1) is calculated according to the receiving packet timestamps and sending packet timestamps. OWD is calculated by taking difference between receiving timestamps and sending timestamps and given by equation 1:

OWD= Tsent-Treceived ....*(1)*

where, Tsent is timestamp when the packet is sent from client side and Treceived is timestamp when the packet is received in server, both in seconds. While taking timestamps in two different nodes at the same time, time synchronization issue may arise. To minimize the clock offset i.e. difference between the local times of two different nodes, time synchronization is required to adjust clock time between two nodes. For this, we maintain one script timesyn.py that synchronize both client and server with same NTP server. Even we use NTP server to synchronize time there might be some difference in time between two nodes. To avoid this, we compute the OWD normalized and given by equation 2:

OWDnormalized= (OWD-OWD_min)/OWD_min................*(2)*

where, OWD is one-way delay calculated from equation 1 and OWD_min is the minimum OWD obtained in each set of experiments. Its unit is in number and indicates that how much far away from minimum one-way delay. For eg. If OWDnormalize is 0 it means minimum OWD; if 1 it is twice as much as the minimum OWD and so on. It is used to minimiz the time synchronization difference between two nodes. For each experiments and observations, we have calculated the OWDnormalize and plot them into a boxplot to visualize the behavior of the system.

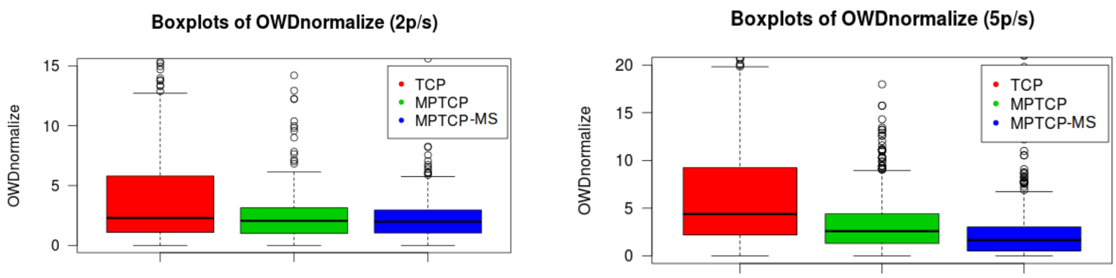
4.2 Results

4.2.1 Packet Size 1: 1024 Byte

First of all, results and analysis for packet size of 1024 Byte is given below with 3 different sets based on packet transmission rates:

Set 1: 2 (ppp0) and 5 (ppp1) packets per second

The transmission rates in this set are selected to be low: 2 and 5 packets per second for each application. The experiment is carried out 10 times



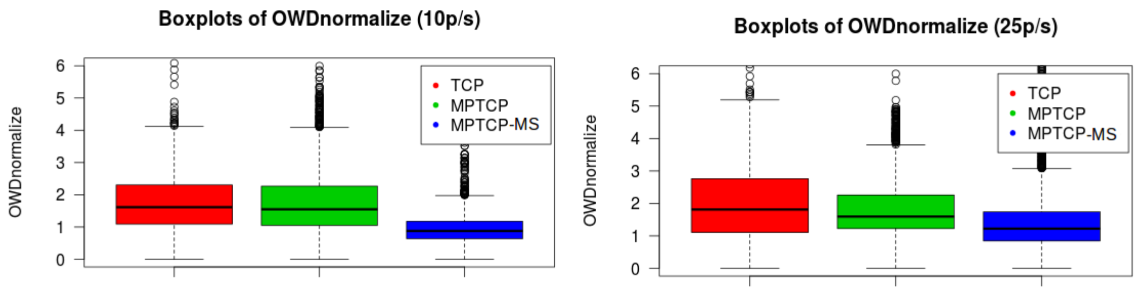
(a) OWDnormalize of 2 packets per second

(b) OWDnormalize of 5 packets per second

Figure 4.9: OWDnormalize of 2 and 5 packets per second

and each for one minute of time interval, it means that the 1200 sample of OWDnormalize are collected for 2 pkts/s and 3000 samples for 5 pkts/s. Individual output file is generated in server side for 2 pkts/s and 5pkts/sec and calculated OWDnormalize from equation 2. The boxplot of the results are illustrated in Figure 4.9 a and 4.9 b, and they clearly show that MPTCP-MS outperforms the other two setups.

As per result, in 2 pkts/s the distribution of OWDnormalize for normal TCP communication seems to have higher variance than other two techniques, which is illustrated in figure 4.9 a. OWDnormalize distributed from nearly 1 to 6 in normal TCP communication, where both MPTCP and MPTCP with selected interface (MPTCP-MS) communication is less than 5 or nearly equal to 2 and 3. In all three cases, the median of OWDnormalize are all at same level i.e. nearly 2. However, the boxplots in these cases shows very different distributions of OWDnormalize for normal TCP communication. In addition, MTCP and MPTCP-MS gives the better result than normal TCP communication as expected in this case. The distribution of the OWDnormalize for 5 pkts/s is shown in figure **??** b. In this case, packet transmission rate is increased to 5pkts/s. In this result, OWD distribution for normal TCP communication is distributed more than other two techniques (figure 4.9 b). The median values also varies with MPTCP-MS even it has a low median value where TCP has higher median OWDnormalize value nearly 5 and MPTCP has nearly 3. As per result, increasing transmission rates from 2 to 5 pkts/s the OWDnormalize is increased compared as 2 pkts/s.



(a) OWDnormalize of 10 packets per second (b) OWDnormalize of 25 packets per second

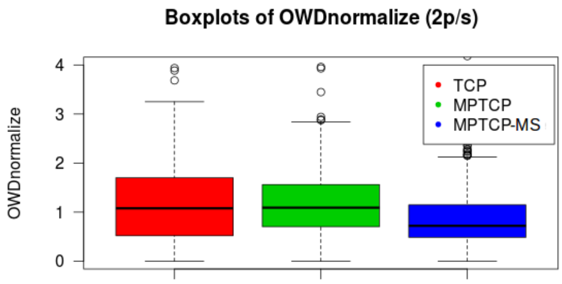
Figure 4.10: OWDnormalize of 10 and 25 packets per second

Set 2: 10 (ppp0) and 25 (ppp1) packets per second

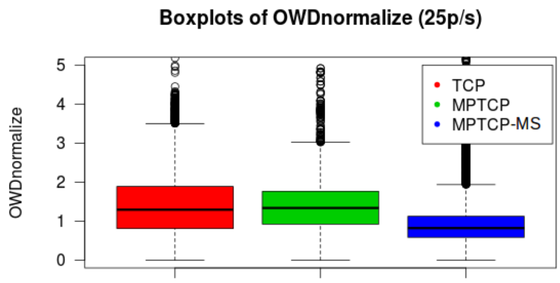
Now, the experiments and results are drawn for different packet rates: 10 and 25 pkts/s. The first application with ID 1 sends 10 packets per second and next application with packetID 2 sends 25 packets per second at the same time. The OWDnormalize for normal TCP and MPTCP communication techniques in 10 pkts/s give the similar distribution, both contain same level of median. The medians of both cases are distributed equally through 75 percentile and 25 percentile. While median of OWDnormalize for MPTCP-MS is nearly half as compared to others (figure 4.10 a).
But in the case of increasing transmission rate to 25pkts/s in same setup with different application, the OWDnormalize is distributed more in normal TCP communication. The behavior of MPTCP remains unchanged even increasing rates. As per result, it shows MPTCP-MS is also slightly affected from increasing rates. It distributes OWDnormalize values in large area. Outliers in MPTCP and MPTCP-MS techniques are more compared to TCP but less in OWDnormalize value (figure 4.10 b).

Set 3: 2 (ppp0) and 25 (ppp1) packet per second

This set of results are observed with two different packets rates: 2 and 25 pkts/s. In the case of 2 packets per second the results show that the OWDnormalize value is distributed less compared to above results. The median of the OWDnormalize value for both TCP and MPTCP communication technique is similar nearly equal to 1. But the 75 percentile distribution of OWDnormalize value for MPTCP-MS lies nearly at 1. The median is less than 1 and near to the 25 percentile of whole data sets (figure



(a) OWDnormalize of 2 packets per second



(b) OWDnormalize of 25 packets per second

Figure 4.11: OWDnormalize of 2 and 25 packets per second

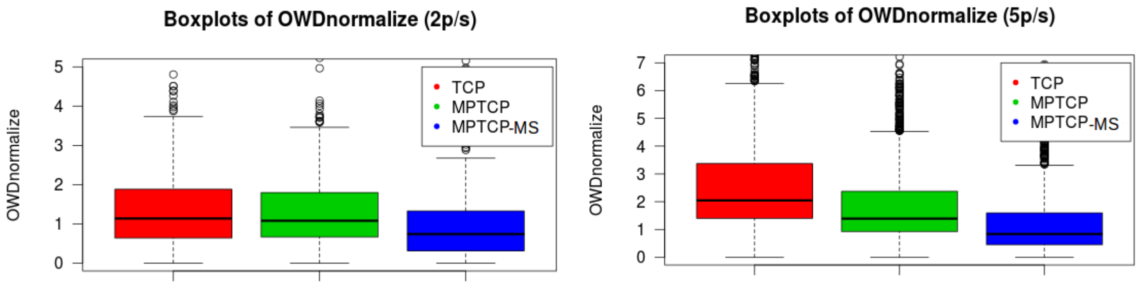
4.11 a).
The next application at same setup sends packets with 25 pkts/s and the result is obtained as figure 4.11 b. However, the rates of transmission is increased to 25 pkts/s, it gives quite similar results as above 10 pkts/s. The distribution of OWDnormalize data is also similar as 10 pkts/s for all of the three techniques.

4.2.2 Packet Size 2: 500 Byte

Now to observe different behavior of the system setup, we have tested our system with packets size 500 Byte with different packets rates. Same procedure as above is applied to collect the data. Three sets of results are obtained with different rates:

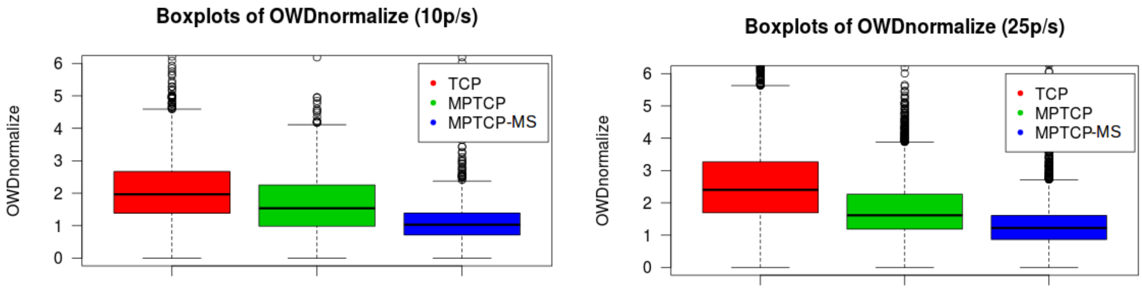
Set 1: 2 (ppp0) and 5 (ppp1) packets per second

This set of results is observed from the two applications with different packet rates 2 and 5 pkts/s and packets size of 500 Byte. As per result, the median of the OWDnormalize value for MPTCP-MS is less than 1 and other both techniques have greater than 1. The OWDnormalize value is also distributed in same manner for TCP and MPTCP but less in MPTCP-MS (figure 4.12 a).
But, in same setup with another application with 5 pkts/s gives the little bit different results. Figure 4.12 b illustrates the results obtained for 5 pkts/s. The median of OWDnormalize value for TCP communication technique is approx. 2; in MPTCP technique median is nearly equal to 1.5 and in MPTCP-MS median is approx. 1. All the medians are near to the 25 percentile of whole datasets.



(a) OWDnormalize of 2 packets per second (b) OWDnormalize of 5 packets per second

Figure 4.12: OWDnormalize of 2 and 5 packets per second

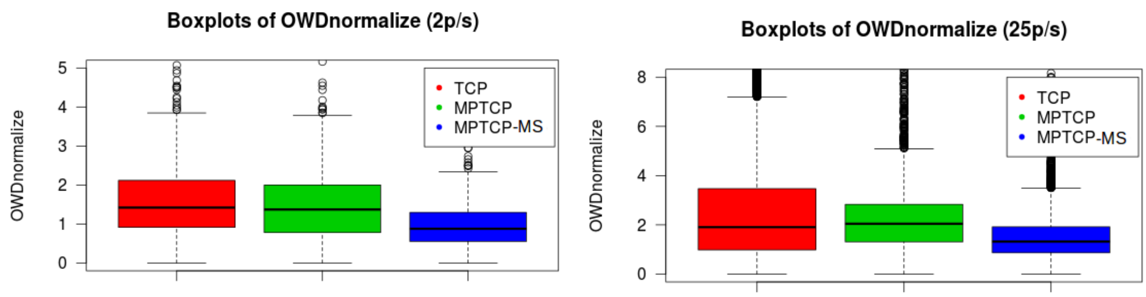


(a) OWDnormalize of 10 packets per second (b) OWDnormalize of 25 packets per second

Figure 4.13: OWDnormalize of 10 and 25 packets per second

Set 2: 10 (ppp0) and 25 (ppp1) packets per second

In this set of observation two applications with different rates of transmission 10 and 25 pkts/s are utilized. In the case of 10pkts/s, the value of OWDnormalize is distributed almost equal to upper quartile and lower quartile from median in all three techniques. The median value of OWDnormalize for MPTCP-MS is nearly equal to 1, where for TCP communication technique it is double i.e. approx. 2 and 1.5 for MPTCP technique (figure 4.13 a).
After increasing the transmission rates to 25 pkts/s in another application, the results are obtained as shown in figure 4.13 b. Here all three techniques have different OWDnormalize values distributed for normal TCP is high and for MPTCP-MS is low. The median OWDnormalize value also varies accordingly. TCP have more than 2 where MPTCP is less than 2 and MPTCP-MS is slightly greater than 1.



(a) OWDnormalize of 2 packets per second (b) OWDnormalize of 25 packets per second

Figure 4.14: OWDnormalize of 2 and 25 packets per second

Set 3: 2 (ppp0) and 25 (ppp1) packets per second

In this set of results, transmission rates are varies from 2 to 25 packets per second, one application with 2 packets per second and another with 25 packets per second. For 2 pkts/s transmission rate the result for different three techniques is shown in figure 4.14 a. As per result, the OWDnormalize value is equally distributed for normal TCP and MPTCP communication technique. The median value is also in same level nearly 1.5. In addition, the result for MPTCP-MS is, as from all other results it distributes the OWDnormalize value less than others and median lies in approx. 1.
On the other side, another application with 25 pkts/s transmission rates gives the results that are illustrated in figure 4.14 b. In this set of results, we can clearly say that the OWDnormalize value is more distributed than above 2pkts/s experiments mainly in TCP communication techniques. The median value is more compared with above experiments. The median OWDnormalize value of TCP and MPTCP technique is also in same level but little more i.e. 2. Same for MPTCP-MS, the median value is more than above set i.e nearly equal to 2.

4.3 Analysis

For our performance analysis, we examine the basic setup scenarios with respect to parameter One-way delay. As the definition describes that there are two kinds of transmission protocols as default MPTCP and MPTCP-MS. First, default MPTCP where all the packets from different applications are multiplexed and scheduled over multiple links. And the second one,

proposed algorithm, takes specified paths for packet transmission (in our case two paths ppp0 and ppp1) benefiting the QoS parameter (OWD).
The same experiment has been also observed in the TCP technique, which was only undertaken for the reference purpose. This is because it uses a single packet transmitting connection link with the lowest metric value. This technique is solely dependent upon the single connection link (ppp0) and can have effect on the QoS parameters if the different links is used. Since, this may not be a reliable source for quality assessment, this finding will not be used for the comparison purpose.
From our observations, in the case of experiments using packet size 1024 bytes by two applications one with 2 pkts/s and another with 5 pkts/s; as expected TCP´s performance was the worst with more than double OWD-normalize figure than the other two. This is may be because of TCP uses higher delay path for transmission. The MPTCP-MS performance was the best in every set of results with around 30% less then the default MPTCP. This is due to the less reordering of the packets coming from different links. Even after reducing the size of packets (i.e. 500 bytes), same kinds of results were observed. But there is comparatively same distribution of OWD-normalize value in all the cases of 2 pkts/s with a lower median value of MPTCP-MS. Where as, in 5 pkts/s, a significant difference is observed.
Another variation of transmission observation with two applications one with 10 pkts/s and other with 25 pkts/s, For 10 pkts/s, in the case of 1024 bytes packet size, the performance shown by the default MPTCP is quite similar to that of TCP. In theoretical approach, the default MPTCP should have shown better performance than that of TCP but due to some constraints the result seems to be affected. Even in such constraints, the performance shown by the MPTCP-MS is remarkably good with less than half of the OWD normalize value. In the second test with 25 pkts/s with same packet size, the exact difference can be seen. TCP´s OWDnormalize value is largely distributed. Default MPTCP shows a little better performance than that of TCP and even though MPTCP-MS shows better performance value but still is more that of the previous one (i.e. 10 pkts /s). With these findings, we can suggest that MPTCP-MS performs better with less transmission rate. Similar results were observed while decreasing the packet size with similar transmission rates. The performance values for all the transmission protocols seems to be identical as of the 1024 bytes and 500 bytes of packet size used by both applications in 10 pkts /s and 25 pkts/s

simultaneously.
As to observe or replicate different behaviors in completely different set of transmission rate, we use 2pkts/s for one application and completely larger rate i.e. 25 pkts/s in another application. Commonly, as obvious 2 pkts/s in both cases OWDnormalize is minimally distributed and median value is lower than the 25 pkts/s. For both transmission rates, OWDnormalized is more distributed in 500 bytes packet size and others case (i.e. 1024 bytes packet size) are similar to each other. The medians of OWDnormailze value comparatively are more in 500 bytes of packet size.
In all above observations, some delay values are obviously very high that can be considered as outliers and shown in all boxplots. All our experiments show that in the scenarios used, the customized MPTCP strategy to create a selected paths among the available interfaces performs significantly better than the default MPTCP and normal TCP communication techniques.

Chapter 5

Discussion and Future Work

5.1 Discussion

This particular section of the report is focused on providing a general summary regarding the procedures and protocols followed during the thesis along with the practical implementation of the project, the issues faced during the period of the project including the implementation phase and last but not the least, the outcome and the further possibilities regarding the future usage of this project as a base line for other projects as well.

TCP has been an industry standard in the packet transmission technology especially within the realm of World Wide Web and Internet, primarily focusing on the reliable transmission of packets of data. With the TCP technology as a foundation, MPTCP has been an ongoing development towards the optimum usage of multiple paths and resources to provide a better transmission service utilizing the concept of sub-flows.
Along with the increase in access devices with multiple network interfaces, the availability of these resources and the maximum and simultaneous usage of this multiple network interface technology in order to increase efficiency for data transaction and transmission, this field of research is both exciting and demanding. With this particular scope at hand, the project has been focused on the selective transmission of data through multiple interfaces as per the requirement of the user and the size of data. TCP has been an industry standard for the general transmission of data where a single connection is established between the source and destination points. With MPTCP there lies an option of using multiple paths using multiple inter-

faces for varied connection purposes between end points. MPTCP with selected interface (MPTCP-MS) was the proposed technology for this project, which enables the functionality of MPTCP; additionally support the selection of specific path during transmission of data. It is possible by modifying the kernel of NorNet edge node to control over which path we want to send packets based on packetID.

In the initial stage of the thesis, the focus was particularly in developing a Telemedicine System prototype including the user interactive interface along with the data transmission aspect as well. However, adhering to the suggestions provided during discussions with faculty members regarding the project; the project and thesis has now been directed towards the communication and transmission factor as the premium agenda. With this said, it has been exciting and challenging to venture into this territory where TCP and MPTCP has been the technological basis of the project where the technology has been utilized with the assistive support of NorNet edge node for practical implementation of the technology using multiple mobile broadband networks provided by Netcom and Telenor. One of the major issues that was faced during the implementation phase of data transmission between the client and the node was particularly focused on the IP conflict problem where the client possessed a private IP and when the client was connected to the node with the help of crossover cables (LAN), the connection could not be established with the server as the node also received the private IP from the client in eth0 which caused the transmission failure as the eth0 was the primary transmission path between the client and server via node. In order to rectify this problem, a public IP interface was set as the de facto standard i.e. lower metric for transmission purpose instead of eth0. This alternative method of correction has been able to solve the aforementioned problem.

The result obtained from the project was inline to the desired output we required from this particular project, where the major focus of the project was to probe our prototype concept against the pre formulated TCP, MPTCP and MPTCP-MS. After a thorough probing in this matter, the expected result has been achieved which has also been shown from figures 4.7 and 4.8 in results section of this report as well. The figures as mentioned above, show that the if two clients send two different sized packets of data

in different rates, the script running within the node will separate the transmitted data in terms of packet ID and the path on which the packets flow will be determined within the node in terms of the packet ID to the server. In this project, one way delay factor was taken as a threshold parameter and this project has been able to succeed in achieving one way delay factor as minimum as possible compared to aforementioned technologies.
In multi-path transfer, the most serious issue is the reordering of the packets, which was caused by dissimilarity delays path. In our experiments, we have used two different MBB providers (Telenor and Netcom) with different characteristics. Each provider may have different characteristics in-terms of delays or bandwidths. While packets are transferred through multiple paths, packets transmission using low delay path reach the destination faster than high delay path. This may cause the reordering of packets in the receiver side. If more packets have been reordered, it will create more complex to restoration for the receiver and also required more buffer size. To reorder packets in receiver side in its original form, it may take time so that delay may be increased. In the default MPTCP mechanism, this issue may arise. To solve the reordering of packets, we can assign specific path and send packets with same packetID on the same path. This will minimize the reordering issue and give the better result in-terms of delay.
In all of our scenarios, results obtained using normal TCP connection, shows that, it utilizes more delay than other two mechanisms. This is due to, TCP only use one path for packets transmission and may be it uses higher delay path. In the case of default MPTCP connection, it seems better than TCP. It is obvious because it uses multi-path for packet transmission. It is also reliable because if we loose one link we have another link to continue data transmission. Even it uses multi-path it performs quite less than MPTCP-MS, this is because of reordering issue specified earlier. Finally our proposed method MPTCP-MS shows the best results in terms of delay, because it separates the incoming packets from two clients with respective packetID and transmits through the selected interface. The packets in receiver, receives in its original form and don't need to reorder, so it reduce the delay factor.
One particular astonishing factor realized during the tenure of this project was that, while experimenting with transmission of the data packets, there were several experiments carried out in order to find out any performance issues relating to OWD parameter. While doing these repetitive experi-

ments on MPTCP-MS, OWD was observed to be higher in the first few repetitions however during the progression of repetitions it was observed that the OWD decreased in an explicit pattern where the decrement was openly noticeable. In comparision to TCP and default MPTCP, MPTCP-MS has proved its robustness and OWD factor was found to be minimum as compared to the other two technologies.

The project has been able to meet the agendas set forth, although there were a few hindrances during the course of the project. One major problem faced during the practical implementation of the project was the use of NorNet Edge node, which was not a familiar device for me. The basic working of the device was theoretically understandable however, the practical usage of the device was a bit of a problem as the device was absolutely foreign to my knowledge. Therefore, I needed to invest some time in order to familiarize myself with this device before I was able to operate the device in a capable manner.
As mentioned before, initially the project was structured to design a full-fledged Telemedicine system with transmission capabilities. However, during the course of time, the project was substantially minimized in its scope to focus particularly on the transaction and transmission of data. This transition of the structure of the project was one limiting factor in the proper completion and implementation of the project as there were time constraints due to the fact that the focus of the project was shifted which caused considerable loss of time in terms of research which focused on the full fledged Telemedicine system. In the background section of the report, I had explained a lot regarding the full-fledged Telemedicine system. However, with the shift in focus the information mentioned in the background might not all remain relevant to the project at hand. Even though, this was a limiting factor the project remained on course and within the time limit, which guaranteed timely completion of the project as well as the results obtained from the project has been quite satisfactory as well.

The major focus of the project was to compare the delay factors related to TCP and MPTCP technologies where we conducted several experimental probes in order to find out one-way delay related to both the technologies where we were more focused on MPTCP-MS. The project is practical in its approach, where a prototype device such as Nor-Net Edge node was

used to transmit delay sensitive data and these transmission details were recorded and analyzed in this project in order to generate an unbiased conclusion generated from the information gathered from several experiments with the device. As per my view, it is paramount that a prototype device such as Nor-Net Edge containing multiple interfaces was necessary for the proper completion of this project. In light of this fact, I believe this was the best approach for the project and it would be wise for any other such projects to embrace this approach.
One major issue that is necessary to discuss regarding this project is the issue of variety of time zones (time synchronization), which may cause disparity in time delay calculation within the project. In order to solve this issue of disparity in time delay calculation due to the variety of time zones, one particular solution that I think is creation of a script that will enable both client and server devices to sync from the same NTP server which has somewhat resolved the issue of time difference. However, in order to further minimize this issue, I have also devised and used the OWD normalization concept.

Our proposed system, which has also been implemented practically to a greater extent, has been able to provide valuable answers related to transmission of different kinds of data in the path desired by the user. This particular outcome will impact the functionalities of transmission of data using TCP and MPTCP, which are the forth-set standards up and until now. This will undoubtedly be a major contributing factor for MPTCP-MS.

5.2 Limitations

In our experiment setups, we have used thin packets transmission rates i.e. 2, 5,10 and 25 pkts/s with two packets size 1024 bytes and 500 bytes. It is very few packets per second for transmission, which is similar to chat traffic. To transmit or download large files we need to manipulate large rates of packets per second for e.g. 100 or 500 packets per second. This is the limitations observed with defining large rates for packet transmission or bulk data transmission. In our implementation, we were unable to send more than 50 packets per second. With TCP, it is harder to control exactly what we want because we are limited by the feedback coming from the client side. We cannot just send how much packets we want. There is some imprecision in the OS if we put the rate too high. To overcome this problem UDP socket is more appropriate or we can use traffic generator tool like netperfmeter, iperf, etc. Due to time constraints we cannot test this methods for our experiments.

5.3 Future Work

As we discussed earlier, the initial interest for this project is to build and implement a full fledge Telemedicine system with nice user interaction interfaces using our proposed communication mechanism. Due to time constrains, we can only focus on communication system using new and modified MPTCP technology that can select the interfaces during transmission. So, designing and implementing the full fledge Telemedicine system in a real time will be a future work. To verify our proposed system in a good way, we have measured and computed OWD as a one parameter, this can be done with other different QoS parameters such as throughput, bandwidth, error rates etc. in future to get more realiability of our system. We have tested our system with only two mobile broadband providers (Telenor and Netcom) enabling 3G, in the future, number of providers can be increased and test same functionality with 4G aswell.

Chapter 6

Conclusion

The project has been carried out in partial fulfillment of the Masters degree in Network and System Administration at University of Oslo, where the major focus of the project remained within the arena of designing a prototype of telemedicine system utilizing multiple mobile broadband networks, where a refined focus was within the communication and transmission patterns of the system. The project was partly supervised and carried out under Simula Research Laboratory. The major technology support provided with the help of NorNet Edge node device provided by Simula Research Laboratory lies vital to the project which enables multiple path streaming as the node contains multiple interfaces which can be used to transmit data from multiple mobile broad band providers (Telenor and Netcom) simultaneously and provide optimum path selection as per the need of the user.

The project particularly focused on the transmission issues and the project has remained vital to drawing up a comparison in terms of OWD parameter where the competing technologies were TCP and MPTCP. MPTCP with selected interface was the proposed technology of the project, where the experiments ran within this project has sufficed to provide a sustained result, which showed that MPTCP with selected interface was capable to transmit different types of data through selected interfaces (paths) using the NorNet Edge node device. In order to provide the selective functionality for selection of path, I have modified the Kernel of MPTCP as the modification was required for this particular project.

In this regard, the project can be deemed successful in obtaining the desired outcome, as the transmission of a variety of data through selected paths was clearly observed during repetitive experiments, which have been discussed in detail in discussion as well as results and analysis sections of this report. Several issues related to the project from the shift in focus of the project from a full fledged Telemedicine system to the communication and transmission aspect as well as issues related to time synchronization problems are notable and have been explained in detail in the discussion section as well. These issues however, were also solved during the completion of this project which were some major milestones achieved while completing the project. Therefore, the project has been able to attain its relevant outcome and in future, the project can provide contribution towards understanding the transmission issues related to different types of data through different links as required by the users. Another major contribution is the solution provided in terms of time synchronization, which can be beneficial when it comes to time zone variations, which can cause severe malfunction of the system if time synchronization functionality is not put into consideration.

Bibliography

[1] Strehle EM, Shabde N. One hundred years of telemedicine: does this new technology have a place in paediatrics?, *Archives of Disease in Childhood*, 2006, 91(12):956–959.

[2] What is Telemedicine?, [www-document], http://www.americantelemed.org/about-telemedicine/what-is-telemedicine, Last visited on 12/03/2014

[3] WHO, A health telematics policy in support of WHO's Health-For-All strategy for global health development: *report of the WHO group consultation on health telematics*, 11–16 December, Geneva, 1997.

[4] Telemedicine: Norway is leader of the pack, [www-document], http://esante.gouv.fr/en/the-mag-issue-1/telemedicine-norway-leader-the-pack, Last visited on 12/03/2014

[5] Jacob Hygen, Managing Director, Norwegian Centre for Informatics in Health and Social Care (KITH), Health Informatics and Telemedicine in Norway, *This article has appeared in 10th anniversary bulletin, Norwegian Centre for Telemedicine*

[6] About NST, [www-document], http://telemed.no/about-nst.5108462-258955.html, Last visited on 12/03/2014

[7] Craig J, Patterson v. Introduction to the practice of telemedicine., *Journal of Telemedicine and Telecare*, 2005, 11(1):3–9.

[8] Einthoven W. Le télécardiogramme [The telecardiogram]. Archives Internationales de Physiologie, 1906, 4:132–164.

[9] Currell R et al., Telemedicine versus face to face patient care: effects on professional practice and health care outcome*Cochrane Database of Systematic Reviews*, 2000, Issue 2. Art. No.: CD002098.

[10] E. A. Viruete Navarro, J. Ruiz Mas, J. Fernández Navajas, C. Peña Alcega, Performance of a 3G-Based Mobile Telemedicine System, *Consumer Communications asnd Networking Conference*, 2006. CCNC 2006. 3rd IEEE (Volume:2)Page(s):1023-1027

[11] A. Panayides , M.S. Pattichis , C. S. Pattichis , C. N. Schizas , A. Spanias , and E. Kyriacou, An overview of Recent End-to-End Wireless Medical Video Telemedicine Systems using 3G, *Annual International Conference of the IEEE*, 2010,Page(s):1045-1048

[12] NeYu e c h u n C h u and A u r a G a n z, Mobile Telemedicine Systems Using 3G Wireless networks , *P h D Multimedia Networks Laboratory, Department of Electrical and Computer Engineering*, University of Massachusetts, 2005

[13] Clemensen J, Larsen S B and Ejskjaer N., Telemedical treatment at home of diabetics [J]., *Journal of Telemedicine and Telecare*,2005,11(2):14-16.

[14] Maglaveras N, Koutkias V, Chouvarda I, Goulis D.G, Avramides A, Adamidis D, Louridas G and Balas E.A., Home care delivery through the mobile telecommunications platform: the Citizen Health System(CHS) perspective.,*International Journal of Medical Informatics* 68 (2002) 99-111

[15] Pin Zhang, Bo Gao, Weiyi Yiang, Shengfeng Qin, Requirement Study of Telemedicine System and Product, *3rd International Conference on Biomedical Engineering and Informatics (BMEI 2010)* 2010

[16] Brett Harnett, Telemedicine systems and telecommunications, *Journal of Telemedicine and Telecare* 2006;12:4–15

[17] Sylvia Ratnasamy, Andrey Ermolinskiy, and Scott Shenker, Revisiting ip multicast, *SIGCOMM Comput. Commun. Rev.*, 36(4):15–26, 2006.

[18] George Coulouris, Jean Dollimore, and Tim Kindberg., Distributed Systems - Concepts and Design., *Addison-Wesley*, 2005.

[19] Kristoffer Høegh Mysen, RBC: A Relevance Based Caching Algorithm for P2P Access Patterns, University of Oslo, Oslo, Norway, 2007

[20] Jochen Schiller, Mobile Communications 2nd Edition., 2003

[21] H. Inamura, Ed. NTT DoCoMo, Inc.G. Montenegro, Ed.Sun Microsystems Laboratories Europe R., Ludwig Ericsson Research, A. Gurtov Sonera, F. Khafizo, TCP over Second (2.5G) and Third (3G) Generation Wireless Networks, *Nortel Networks*, February 2003

[22] Jacobson, V., Braden, R. and D. Borman, TCP Extensions for High Performance, *RFC 1323*, May 1992.

[23] Allman, M., Balakrishnan, H. and S. Floyd, Enhancing TCP's Loss Recovery Using Limited Transmit, *RFC 3042*, January 2001.

[24] Mathis, M., Mahdavi, J., Floyd, S. and R. Romanow, TCP Selective Acknowledgment Options, *RFC 2018*, October 1996.

[25] About FFmpeg, [www-document], http://ffmpeg.org/about.html, Last visited on 12/03/2014

[26] Welcome to the FFmpeg Bug Tracker and Wiki, [www-document], http://trac.ffmpeg.org/wiki, Last visited on 12/03/2014.

[27] TCP vs UDP, [www-document], http://www.diffen.com/difference/TCP_vs_UDP, Last visited on 12/03/2014

[28] Ramakrishnan, K., Floyd, S. and D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, *RFC 3168*, September 2001.

[29] Cisco Systems, Inc. Cisco visual networking index: Global mobile data traffic forecast update, 2012 - 2017, February 2013

[30] Rajiv Laroia, Sathyadev Uppala, and Junyi Li, Designing a Mobile Broadband Wireless Access Network An integrated approach based on cross-layer optimization, *Signal Processing Magazine, IEEE (Volume:21,Issue:5)*, Page(s):20-28, 2004

[31] Dahlman,E., Parkvall,S., Skold,J. and Beming,P., 3G evolution: HSPA and LTE for mobile broadband, 2nd Edition 2008.

[32] Rysavy research/4G Americas, Mobile Broadband Explosion, *The 3GPP wireless evolution*, August 2012

[33] 3G Defination, [www-document], http://ajayannext.blogspot.no/2011/07/3g-definition.html, Last visited on 12/03/2014

[34] Wang Z Q., Long Term Evolution of TD-SCDMA, *Shanghai: LTE Summit.* 2008

[35] Amund Kvalbein, Dziugas Baltrunas, Kristian Evensen, Jie Xiang, Ahmed Elmokashfi and Simone Ferlin-Oliveira, The Nornet Edge Platform for Mobile Broadband Measurements, Simula Research Laboratory 2013

[36] Tsung-Yu Tsai,Yao-Liang Chung, and Zsehong Tsai, Introduction to Packet Scheduling Algorithms for Communication Networks, *Source: Communications and Networking, Book edited by: Jun Peng*

[37] Multi-homing, [www-document], http://computer.howstuffworks.com/nat6.htm, Last visited on 12/03/2014

[38] What is NorNet?, [www-document], https://www.nntb.no/home/, Last visited on 12/03/2014

[39] Scrase, Adrian., A future-proof evolution for 3GPP TDD mode, *Shanghai: LTE Summit.* 2008

[40] MultiPath TCP - Linux Kernel implementation, [www-document], http://multipath-tcp.org/pmwiki.php/Main/HomePage, Last visited on 11/05/2014

[41] A. Ford, C. Raicu, M. Handley, S. Barre and J. Iyengar, Architectural Guidelines for Multipath TCP Development, *RFC 6182* March 2011

[42] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, TCP Extensions for Multipath Operation with Multiple Addresses, *draft-ietf-mptcp-multiaddressed-08*, November 2012

[43] Sander Boele , Freek Dijkstra , Artur Barczyk , Gerben van Malenstein , Jim Hao Chen and Joe Mambretti, Multipathing with MPTCP and OpenFlow, 2012

[44] HowToInstallMPTCP?, [www-document], http://multipath-tcp.org/pmwiki.php/Users/HowToInstallMPTCP?, Last visited on 12/05/2014

[45] Manual Configuration, [www-document], http://multipath-tcp.org/pmwiki.php/Users/ConfigureRouting, Last visited on 12/05/2014

[46] A. Hernandez and E. Magaña, One-way delay measurement and characterization, *in Proc. 3rd ICNS*, pp. 114–120, Jun. 2007

[47] Luca De Vito, Sergio Rapuano, Member, IEEE, and Laura Tomaciello IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, One-Way Delay Measurement: State of the Art, VOL. 57, NO. 12, DECEMBER 2008

[48] G. Almes, S. Kalidindi, and M. Zekauskas, A One-Way Delay Metric for IPPM, *RFC 2679*, Sep. 1999. IETF.

[49] Compilation guide Ubuntu, [www-document], https://trac.ffmpeg.org/wiki/CompilationGuide/Ubuntu, Last visited on 19/05/2014

[50] Decoupled from IP, TCP is at last able to support multihomed hosts., [www-document], http://queue.acm.org/detail.cfm?id=2591369, Last visited on 12/05/2014