Sri Lanka Institute of Information Technology
4th Year Research Project

Analysis & Development of Dynamic TLPs in the use of Smart Mobile Device Communication

Project ID: 15-019

Research Review Document

Date of Submission:

Supervisor: Mr.Dhammika H De Silva

Table of Contents

Tables Catalogue

Figure Catalogue

1 Introduction

1.1 Purpose

This document delivers the study of TCP and UDP protocols when transmitting data as per the opted areas of the research. We are looking into the change of header as per the data transferred from the application layer, different methods to be invoked when data to be transferred and analysis of the connection establishment.

1.2 Scope

This document will be looking into three different areas of the transport layer protocol TCP

1. Study of the header of TCP to suite the data coming from the application layer and possibility of advancing it according to the type of data.
2. Study of the possibility of sending out request via the SYN and receive the data from severs through the ACK via the 3 way handshake without having to wait for the connection establishment
3. Study of different methods to be invoked in TCP when transferring data according to their type ie.GPS, WiFi.

- OUTCOME:

Design a protocol suitable for a wireless environment which has a faster, efficient and reliable at flagged transmissions than other transport layer protocol.

- HOW:

Deep study about existing protocols.
Algorithm design for protocols.
Code for algorithms.
Simulate protocols.
Do a comparison between the expected dynamic protocols against existing protocols.

1.3 Overview

1.3.1 Main Objective

To design an enhanced dynamic transport layer protocol which adapts itself according to the data that's being transmitted across a network. For an example if there's any urgent data that needs to be transmitted immediately, a different approach will be used rather than using the conventional transport layer protocol, TCP, which is proven to be inefficient in such situations.

1.3.2 Sub Objectives

- Analysis & classification of application layer data in order to determine which method/functionality to be used in transmitting those data.

- Examining the currently available transport layer protocols & analyzing their behavior to identify their drawbacks.

- Handover method analysis to significantly improve the connection establishment process & to reduce the no of times a particular client has to establish a connection.

- Analyzing the modifications that can be applied to the currently available protocols.

1.3.3 Tasks

The tasks needed to be carried out for the completion of this research are as follows.

- Analyze data transfer behavior
- Evaluate performance of TLP for small Data
- Specify requirements of TLP
- Analyze classification criteria
- Identify the connection type

- Simulate the protocol algorithms and analyse outputs of each aspect of the protocols via Network Simulator 3 software.
- As the final outcome, design an efficient wireless transport layer protocol.

1.3.4 Users

This research will benefit the everyday users of technology for various purposes. A few categories are given below.

- Disabled
- Police
- Surgeons
- Engineers

Availability of info when we need and where we need as per the mobile applications designed to cater the demanding needs of any calibre that is to be found

2 Statement of the work

2.1 Background information and overview of previous literature survey

In computer networking, the transport layer or layer 4 provides end to end communication services for applications within a layered architecture of network components and protocols. The transport layer provides convenient services such as connection oriented data stream support reliability, flow control and multiplexing. A comparison of prominent transport layer protocols used in this research is shown below. [1]

	TCP	UDP	DCCP	SCTP
Connection Oriented	Yes	No	No	No
Reliable	Yes	No	No	Yes
Congestion Control	Yes	No	Yes	Yes
Ordered/Unordered delivery	Ordered	Unordered	Unordered	Both
Retransmission flow	End to End	End to End	End to End	End to End

Table 1.Comparison of transport layer protocols

The project team conducted a thorough literature survey focusing on research papers such as “Impact of Mobile Wireless Links on TCP Performance” by Haowei Bai *et al* [2] which states that TCP will still be the dominant end-to-end reliable transmission control protocol at least in the near future; However, TCP was initially designed to perform well in networks with reliable wired links and stationary hosts, where packet losses are mainly due to network congestion. TCP assumes that all packet losses are due to network congestion.

“Internet Accessibility In High-Speed Vehicles” by Hala Eidaw *et al* [3] was another paper analysed which mentioned that computer and wireless communication require internet accessibility at anytime and anywhere, this includes in a high-speed mobile station such as in speedy trains, fast moving cars as vehicle-to-infrastructure communication. This increased the development of numerous schemes concerning the need of smooth handover of the mobile nodes.

Moreover focusing on the issues in these protocols, Ibtissam El Khayat *et a*l [4] states in his research "Improving TCP in wireless networks with an adaptive machine-learnt classifier of packet loss causes" that nowadays, many applications use TCP as their transport protocol and hence pass through wireless links, which become common in the Internet. Over these links, packet losses are not due anymore only to overthrows but can also be caused by link errors. TCP, which has no mechanism to distinguish packet loss causes, reduces its rate at each packet loss. As a solution the researchers propose to apply a particular learning algorithm called decision tree boosting to automatically design a model for discriminating the two possible packet loss causes and then use this model at best to improve the performance of TCP in wired/wireless networks.

In "Transport Layer Fairness and Congestion Control in Multihop Wireless Networks" it states that experiments and research showed that TCP's congestion control algorithm performs very poorly over wireless ad hoc networks with degraded throughputs and severe unfairness among flows. This paper studies TCP's fairness and throughput issues in wireless ad hoc access networks, and designs an improved congestion control algorithm based on the characteristics of the wireless ad hoc access networks. The protocol is designed as extension to DCCP (datagram congestion control protocol) with a new congestion control component.
The research was done by Kunz.T *et al* [5].

"Improving TCP Startup Performance using Active Measurements:Algorithm and Evaluation" by Ningning Hu *et al* [6] is another such study analysed which mentions that TCP slow start exponentially increases the congestion window size to detect the proper congestion window for a network path. This often results in significant packet loss,while breaking off Slow Start using a limited slow start threshold may lead to an overly conservative congestion window size. This problem is especially severe in high speed networks. In this paper we present a new TCP startup algorithm, called Paced Start that incorporates an available bandwidth probing technique into the TCP startup algorithm.

TCP implementation in Linux

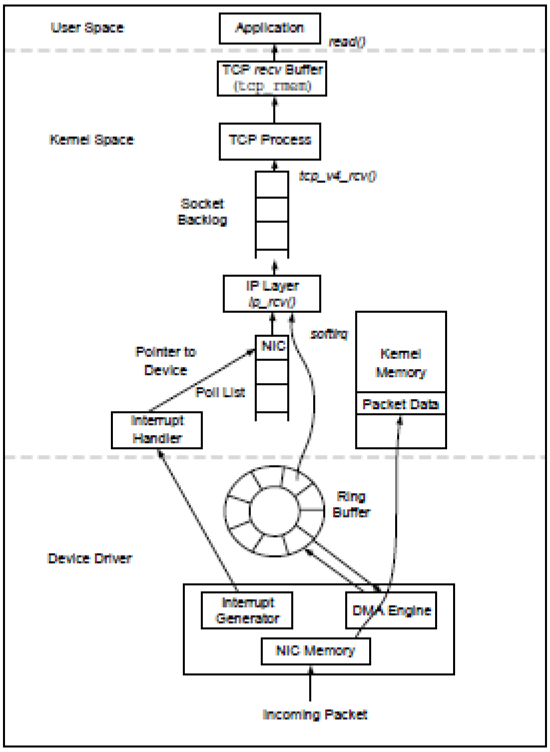


Figure 1.1 Packet Reception 1

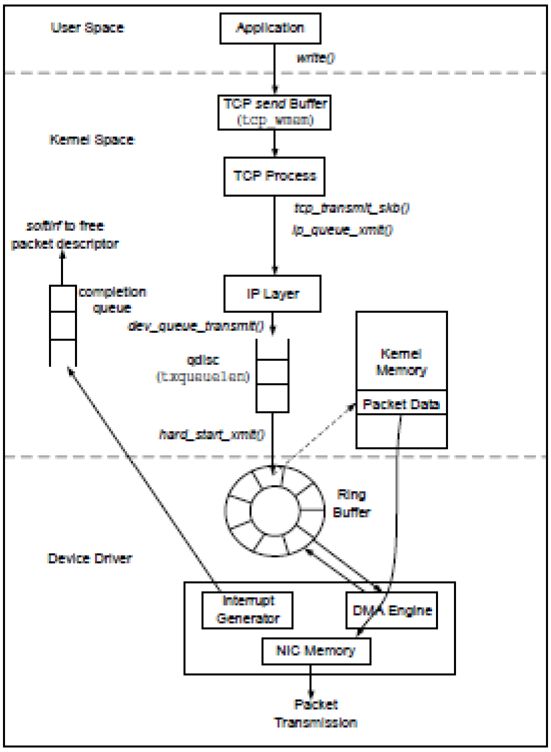


Figure 1.2 Packet Transmission 2

Figures 1 and 2 show the internals of the TCP implementationin Linux kernel. Fig. 1 shows the path taken by a new packet from the the wire to a user application. The Linux kernel uses an *sk buff* data structure to describe each packet.When a packet arrives at the NIC, it invokes the DMA engine to place the packet into the kernel memory via empty *sk buffs* stored in a ring buffer called *rx ring*. An incoming packet is dropped if the ring buffer is full. When a packet is processed at higher layers, packet data remains in the same kernel memory, avoiding any extra memory copies. Once a packet is successfully received, the NIC raises an interrupt to the CPU, which processes each incoming packet and passes it to the IP layer. The IP layer performs its processing on each packet, and passes it up to the TCP layer if it is a TCP packet. The TCP process is then scheduled to handle received packets. Each packet in TCP goes through a series of complex processing steps. The TCP state machine is updated, and finally the packet is stored inside the TCP *recv* buffer.

2.2 Behaviour of Data Transfers

In the research “An investigation in to Dynamic TLP’s for Smartphone Communication” It is said, The way finding application designed for Vision Impaired People requires efficient data transfer in between Building Information Model (BIM) and the person with disability - Infrastructure Network and among the peers (Ad-hoc network). This research identified the size of data, data transfer frequency, reliability, transfer direction and type of the network. Table I shows the summery of identified data transfers and behaviour of data in this system [1]

The mobile device of the vision impaired person will capture the GPS, Wi-Fi and sensor data depending on availability and processed data will be send client to server. This data has very small payload. Due to the very high data transfer frequency the end-to-end reliability is not required. If one segment lost, there is another within very short bounded time. [1]

In the case of a change in the existing map (Obstacle-Fixed), mobile device will be triggered to transfer data to the server. [1]

Data	Size	Transmission Frequency	Delivery Reliability	Direction	Network
GPS - Processed	Very Small	High	No	C → S	I
Wi-Fi - Processed	Very Small	High	No	C → S	I
Obstacle - Fixed	Medium	Triggered	Yes	C → S	I
Obstacle – non stationary	Small	Urgent	Yes	C → S C → C	A, I
Map Data	Large	Triggered	Yes	S → C, C → S	I
User Location	Very Small	High	No	C → S C → C	A, I
Sensor data -Processed	Very Small	High	No	C → S	A, I
Instruction	Small	Urgent	Yes	S → C C → C	A, I

Table 2.Comparison of different data types over network

C- Client, S-Server, A – Ad-hoc, I-Infrastructure

As a summary of the literature review conducted formerly, it was understood that these transport layer protocols were originally developed for wired networks. However due to many reasons such as security, stability, convenience and unlimited roaming and range the protocols were used in wireless networks as well. Yet many shortcomings began to occur and as alternatives, modified versions of these same protocols were developed such as TCP-Reno, UDP-lite.

2.3 Identification and significance of the problem

- **Packet losses during transmission**

TCP has been deployed in the 1980s. Its congestion control is based on the fact that packet losses are mainly due to buffer overthrows and it works quite well in such situations. However, nowadays, many applications use TCP as their transport protocol and hence pass through wireless links, which become common in the Internet. Over these links, packet losses are not due anymore only to overthrows but can also be caused by link errors. TCP, which has no mechanism to distinguish packet loss causes, reduces its rate at each packet loss. This reduction is not justified when there is no congestion and the consequence is that the throughput of TCP over wireless link is lower than what it could be. Research conducted by Ibtissam El Khayat [7].

- **Congestion control in data transmission**

TCP (Transmission control protocol) is a reliable, end-to-end transport protocol, which is widely used for data services and is very efficient for wired networks. However, experiments and research showed that TCP's congestion control algorithm performs very poorly over wireless ad hoc networks with degraded throughputs and severe unfairness among flows by Dr. Thomas Kunz [8].

In fact, the simplest way to solve congestion is to employ the principle of packet conversation in which the sender stops sending a new packet until the previous one is successfully delivered to the receiver. For this reason, the TCP protocol, one of the core protocols of the Internet protocol suite, employs the concept of congestion control which dynamically controls the flow of packet inside a network, and prevents network performance collapse. Research conducted by van Jacobson [9].

- **Error control**

This paper we propose a new scheme that can efficiently deliver multimedia over wireless Internet. Packet losses due to congestion or random error can be distinguished in our scheme. Proper congestion control and error control are performed, which are suited for multimedia applications. Featured with retransmission mechanism, our scheme can also reduce the packet loss ratio caused by fading and random errors, conducted by Fan Yang [10].

- **Improve wireless TCP performance**

This paper, they describe the design and implementation of a simple protocol to alleviate this degradation and present the results of several experiments using this protocol. Their aim is to improve the end-to-end performance on networks with wireless links without changing existing TCP implementations at hosts in the fixed network and without recompiling or relinking existing applications. They achieve this by a simple set of modifications to the network-layer (IP) software at the base station. These modifications consist mainly of caching packets and performing local retransmissions across the wireless link by monitoring the acknowledgments to TCP packets generated by the receiver. Their experiments show speedups of up to 20 times over regular TCP in the presence of bit errors on the wireless link. They have also found that their protocol is significantly more robust at dealing with multiple packet losses in a single window as compared to regular TCP. Conducted by Hari Balakrishnan [11].

This paper makes two important contributions. First, they design a network-based solution called the Window Regulator that maximizes TCP performance for any given buffer size at the congested router. Second, they present a scheduling and buffer sharing algorithm that reduces the latency for short flows while exploiting user diversity, thus allowing the wireless channel to be utilized efficiently [12].

This research, explore a new way to make TCP adapt to frequent route changes without relying on feedback from the network. It is based on TCP detecting out-of-order delivery events and inferring route changes from these events. They call it Detection of Out-of-Order and Response (DOOR). Their study has shown that this approach can significantly improve TCP performance over mobile ad-hoc networks [13].

- **Flow Control**

The performance of TCP degrades over wireless links due to high rate of data losses, which are falsely perceived as network congestion state. TCP performance metrics also diminish due to low data rate, since large delays may occur in last link i.e. wireless link. Similarly in heterogeneous wireless network, packet loss may also occur due to mobility-events that can cause burst-losses, service-disconnection. This motivates to reevaluate TCP control operations and embed some mobility related services to optimize its performance for new generation of wireless networks. TCP for mobile, wireless environment with the help of link-layer triggers which shall be standardized by IEEE 802.21 standard, through its media independent handover (MIH) services. These services are used to detect link layer events in the TCP control operations, and act accordingly to adjust flow and congestion control in a way that does not seriously hamper TCP performance by Muhammad Saeed Akbar. [14]

- **Performance Enhancement of TCP**

Nodes in are distributed and can be statics or mobile. The main advantage of Ad-hoc Networks is that its nodes can be self-organize allowing nodes to connect to each other. The connections between the nodes do not need to establish preexisting infrastructure like other networks. Each node can work as a router to route data to its neighbor nodes. Ad – hoc networks can be used in places that can be difficult to prepare it with infrastructures like open areas. [15]

- **Improving Transport Layer Performance in Multihop Ad Hoc Networks**

TCP congestion control has an implicit assumption, i.e., any packet loss is due to network congestion. However, this assumption is no longer valid in the ad hoc networks as packet losses may well be due to channel errors, medium contention, and route failures.

Several works have pointed out that greedy TCP can result in severe congestion in ad hoc networks and hence performance degradation. Subsequently the TCP source will reduce congestion window size before it becomes excessively large. To avoid congestion, Chen et al. dynamically adjusted the congestion window limit according to path length of TCP flows, a neighborhood RED scheme was proposed to alleviate TCP fairness problem by adjusting marking/dropping probability in light of observed channel information. [16]

In terms of how novel the research topic is it could be considered different and unique since currently such protocols designed for wireless networks within the transport layer are rarely used.

One such protocol is the Wireless Application Protocol (WAP) which is a technical standard for accessing information over a mobile wireless network. A WAP browser is a web browser for mobile devices such as mobile phones that uses the protocol.

The WAP standard described a protocol suite allowing the interoperability of WAP equipment, and software with different network technologies, such as GSM and IS-95(also known as CDMA). [8]

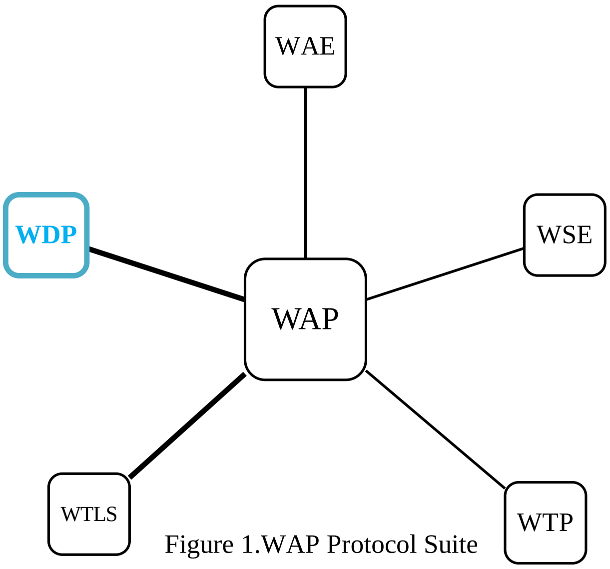


Figure 1.WAP Protocol Suite

However the only transport layer protocol within this suite is WDP (WAP Datagram Protocol).

Wireless Datagram Protocol defines the movement of information from receiver to the sender and resembles the User Datagram Protocol in the Internet protocol suite. It is a protocol in WAP architecture, covers the Transport Layer Protocols in the Internet model. By letting only the transport layer deal with physical network-dependent issues, global interoperability can be acquired using mediating gateways.

Disadvantages of the WAP architecture:

- Thin client architecture
- Normal web technology cannot be used in WAP client
- WAP clients are limited to handheld wireless devices such as mobile phones
- Slow speed of access
- Limited availability

In the research done by Sankara Krishnaswamy on “Wireless Communication Methodologies & Wireless Application Protocol” it is termed that during the study 70% of the users rejected the idea of WAP enabled phones. Some of the disadvantages of WAP clearly made the users to decide not to like this. Because of the misguided use of design principles from traditional Web design, the usability of the **current WAP services is reduced considerably**. WAP is facing the same problem as WEB designs faced in 1994 during the evolution of the Internet. For example, some of the WAP designs that use more screens to display information could have been displayed in a lesser number of screens. This kind of design may work on the Web if users have a big-screen PC, but on a small-screen device, designers must cut short each service down to its essence and show much less information. The time taken to perform a query on the Internet through the WAP is also not acceptable by the users.

Thereby it is vital that an accurate wireless protocol suited for the transport layer is designed. It should be able to avoid the existing drawbacks of the above mentioned wired and wireless protocols. This would be the height of significance this research project is aimed at.

Specific research areas that will be looked into when analysing the Dynamic TLP

2.3.1 Connection Establishment

Connection establishment should be originated by the client. It sends a connection request message called SYN to the server. The server accepts the request and sends an Acknowledgement to the client. Then the client to the server connection will be established. At the same time server sends a SYN to the client to establish a connection from server to the client direction. The client accepts the request and sends the Acknowledgement to the server. Then the server to client direction connection will be established. This is the normal process of the connection establishment.

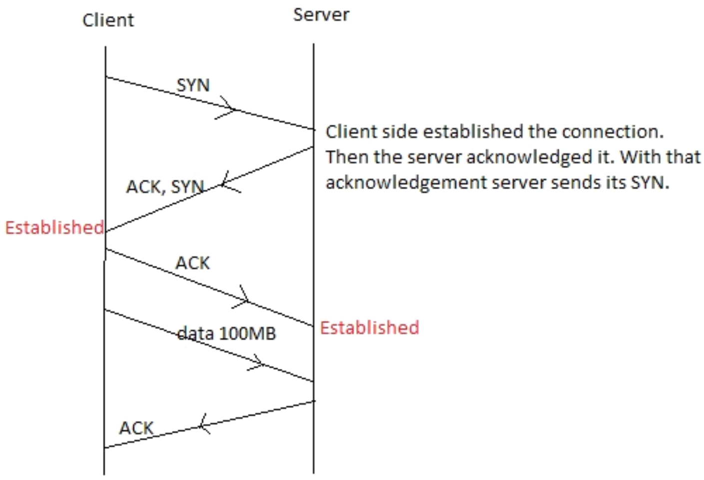


Figure 2. Connection Establishment 3 way handshake

In our research we are trying to find a way to analyze and improve the connection establishment process & to reduce the number of times a particular client has to establish a connection. Here we are hoping to get the request of the client along with the SYN message. Then according to the request the data will be passed to the client from the server along with the ACK. We are trying to analyze the TCP header file and have to go through the code to find out the specific changes that we want to do in order to make the connection establishment a successful one. This will increase the performance.

In order to make this modifications a success we have to,

1. How often the modified connection establishes and terminates itself

2. Time taken to finish the establishment

3. Analyze data transfer speeds

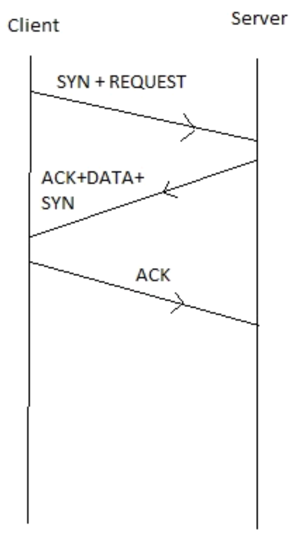


Figure 3. Connection Establishment – Possible Improved version to support the DTL protocol

2.3.2 Header Size

TCP is the currently used Transport Layer Protocol. TCP datagram has a header of 20 bytes no matter what its data content size is. If a small amount of data comes from the application layer & if it needs to be transmitted urgently then there's a huge impact on transmission time of that particular packet if we use TCP and add 20 bytes of data as the header. Therefore dynamic protocol has to be developed which adjust its header size according to the data size.

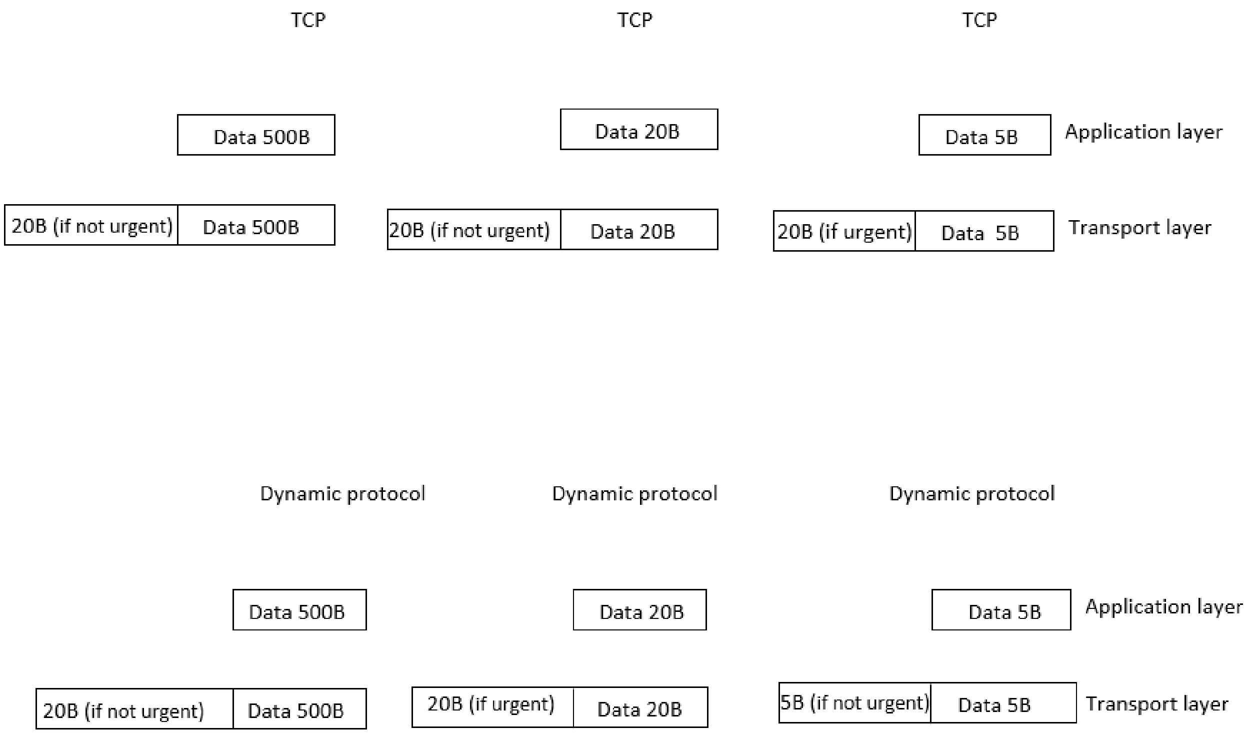


Figure 4. Change of the Header as per the request from Application layer

2.3.3 A single ACK for several requests

Without having to acknowledge each request whenever they are to be received we will be looking into the possibility of enabling a single ACK for several requests of similar nature. We are looking into the “Batch Acknowledgement” mechanism where a single ACK is generated for several requests. However, this way it is more likely to not to receive an ACK if at least one of the requests fails to reach the destination. Since we are not solely looking into the reliability aspect of it, the intention is faster communication within the transport layer.

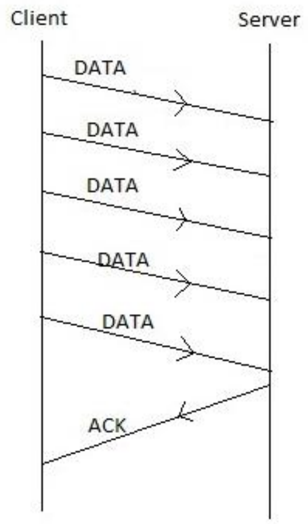


Figure 5. One ACK for 5 data requests/transfers

2.3.4 Altering the order of the execution of Methods in TCP protocol

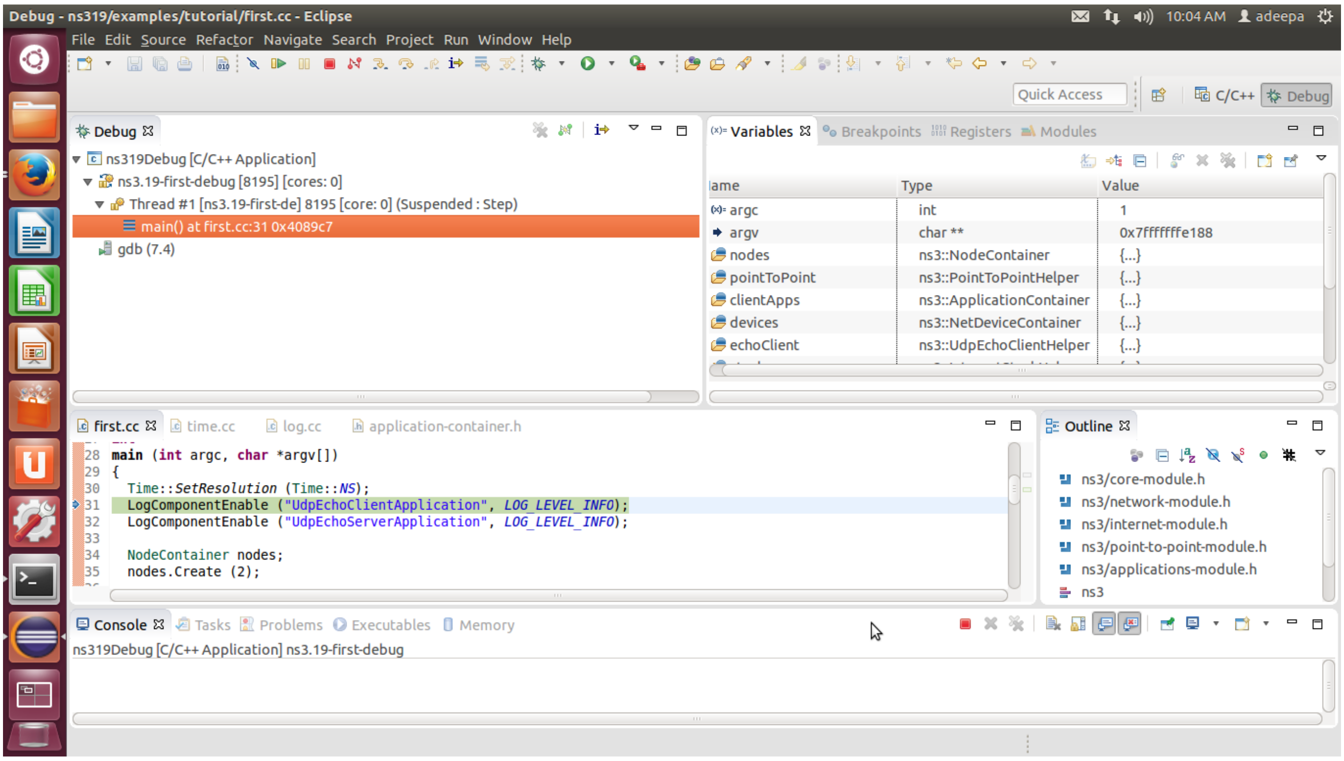
We have found different methods ie. Traffic, Reliability, Congestion control to improve certain aspects of TCP. A way has to be identified where we can change the order of these method executions happening in the tcp protocol in order to facilitate the requirements of the DTL protocol such as avoiding executions of the methods which satisfy the congestion control, reliability, etc. in a situation where urgent data transfer is needed & vice versa.

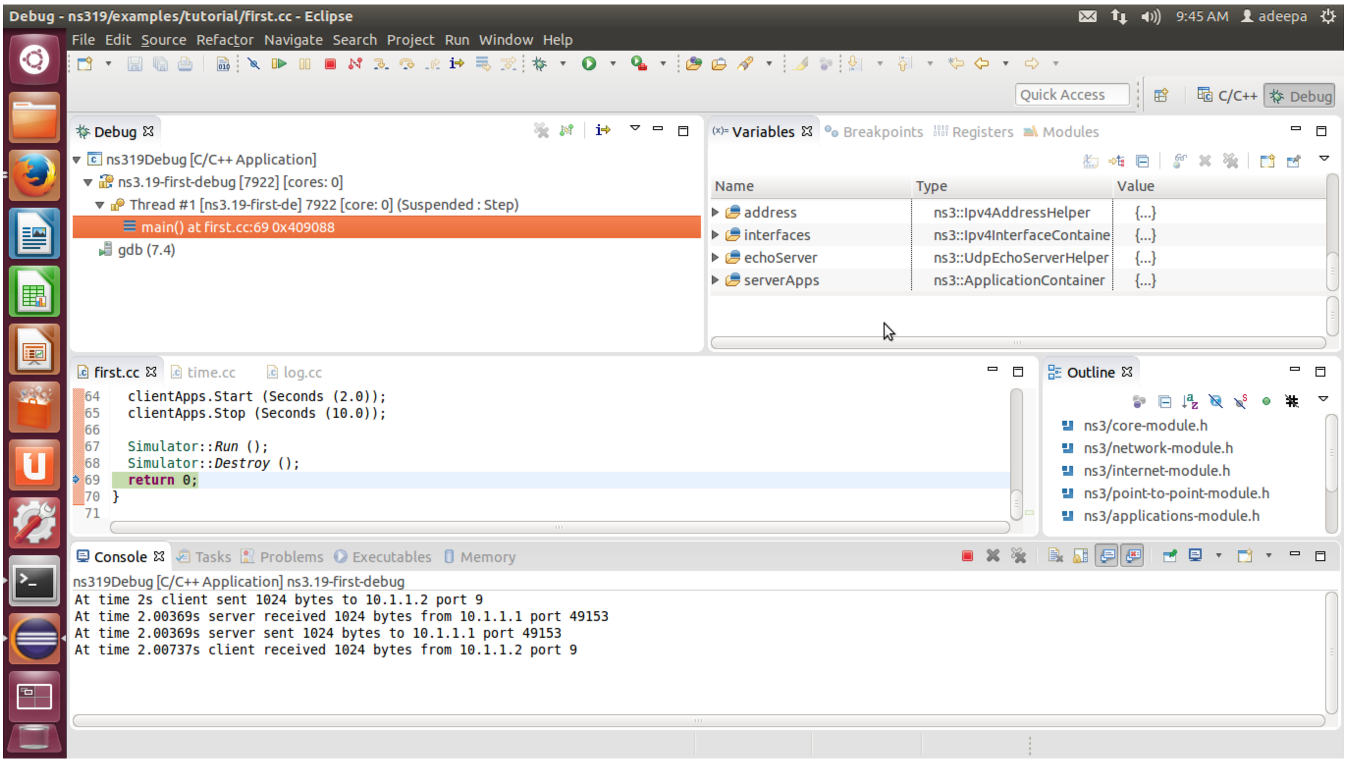
2.4 Technical Objective

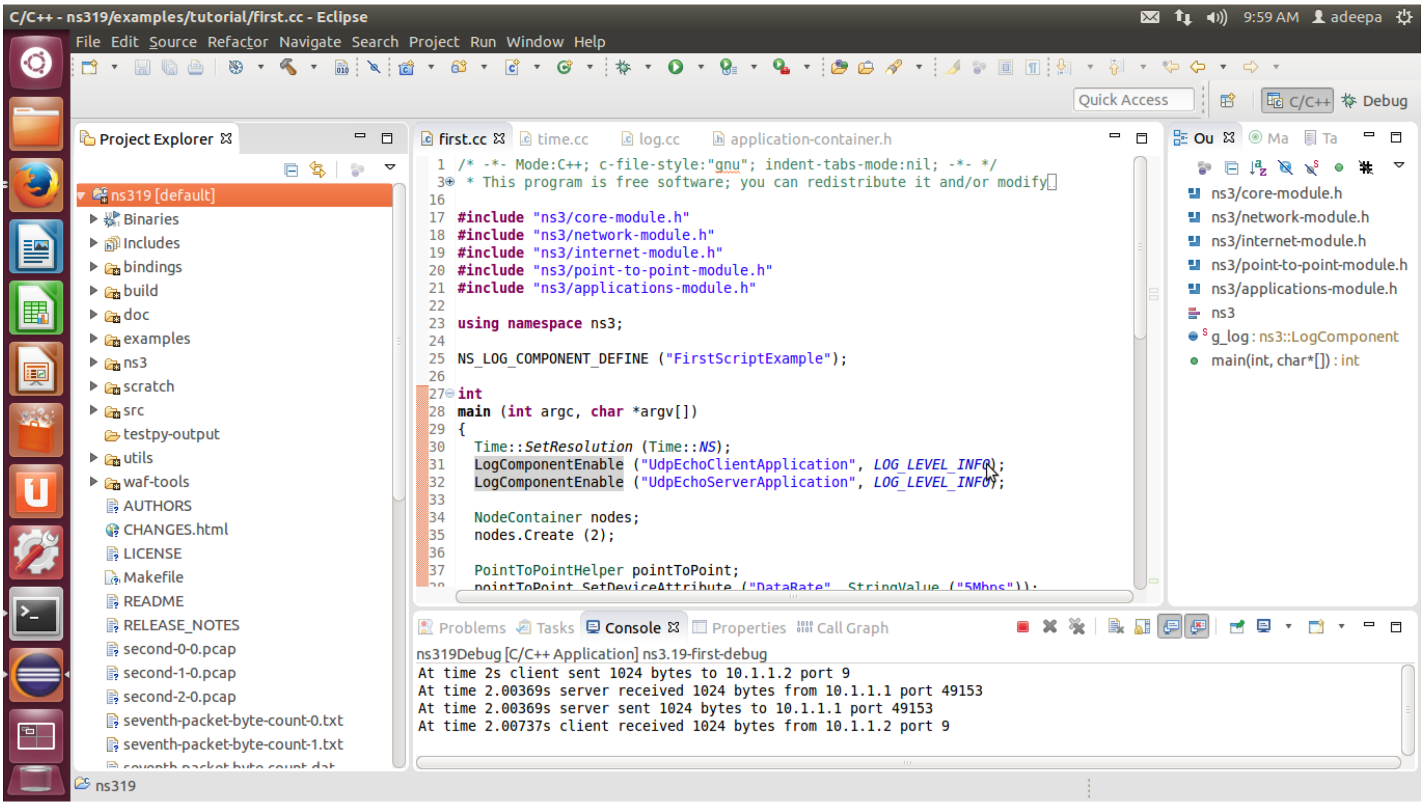
The required software in order to carry out the research is

- Network Simulator 3 (NS3) - A distinct event simulator targeted at networking research. That provides extensive support for simulation of TCP, routing, and multicast protocols over wired and wireless networks.
- Wireshark- A network protocol analyser that conducts thorough inspection of protocol behaviour.
- Ubuntu- A Linux operating system that permits the use of NS3.

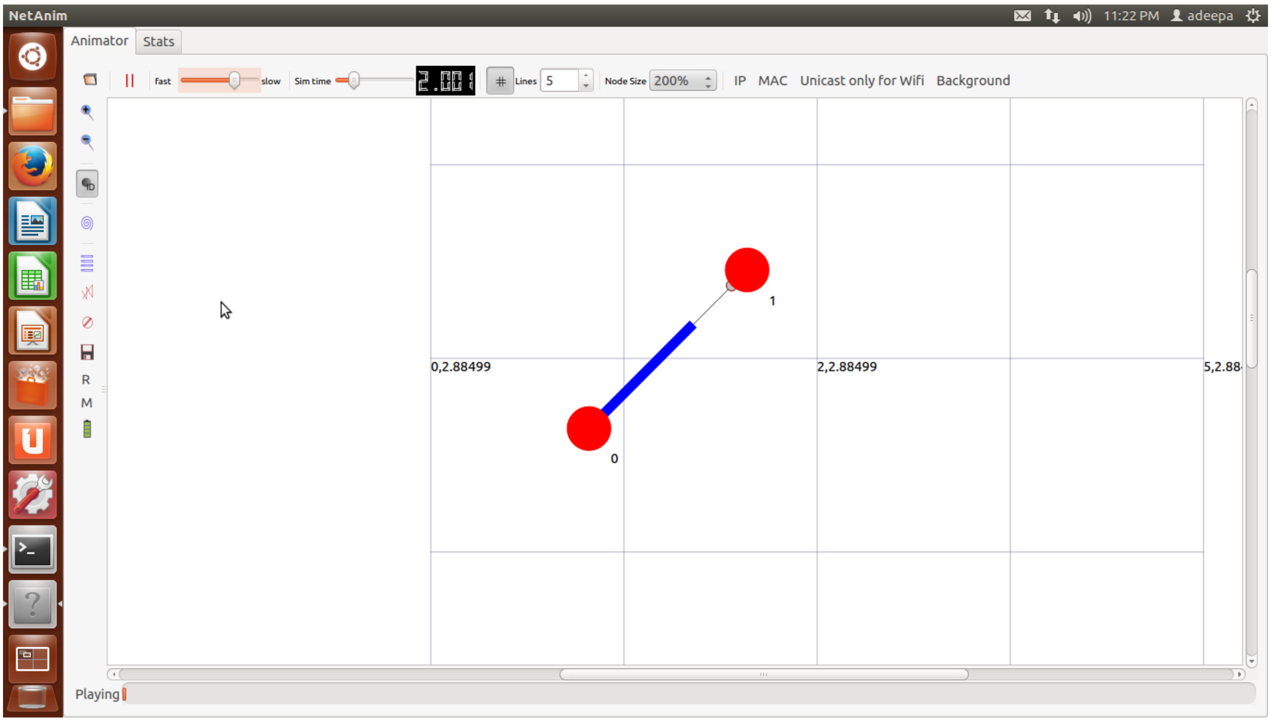
2.5 Detailed Design

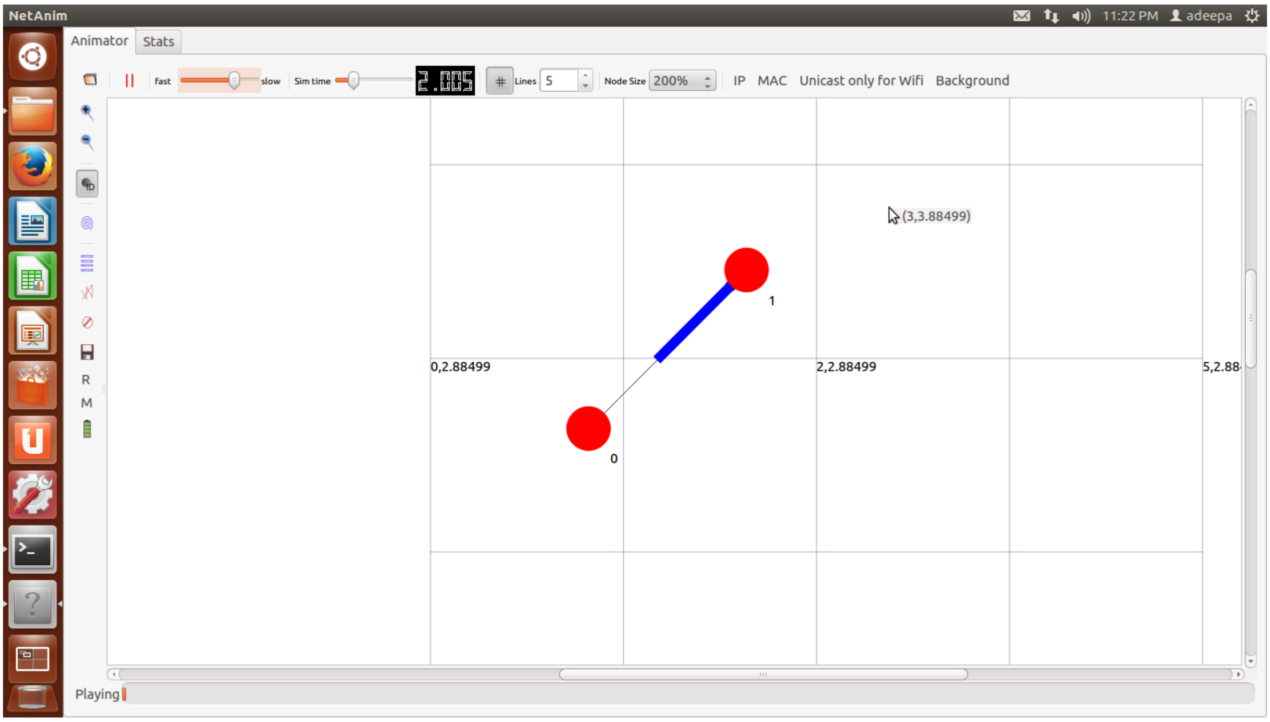




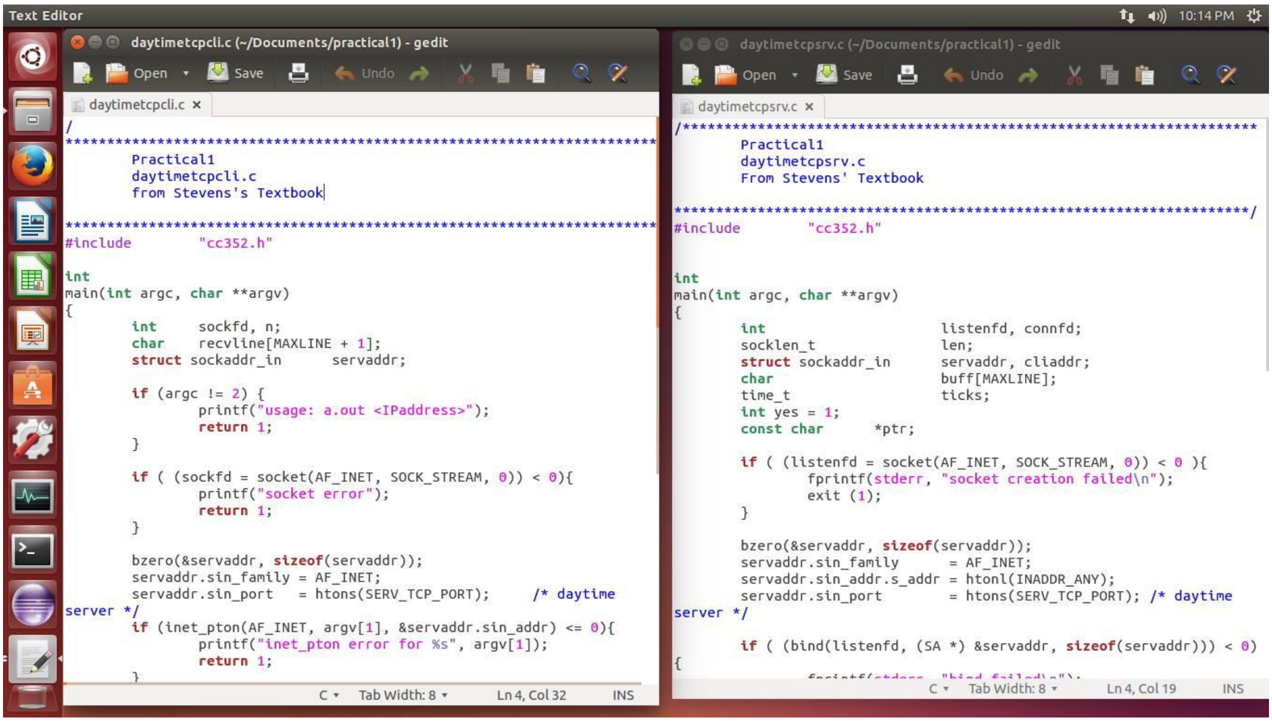


Graphical simulation of the scenario above.

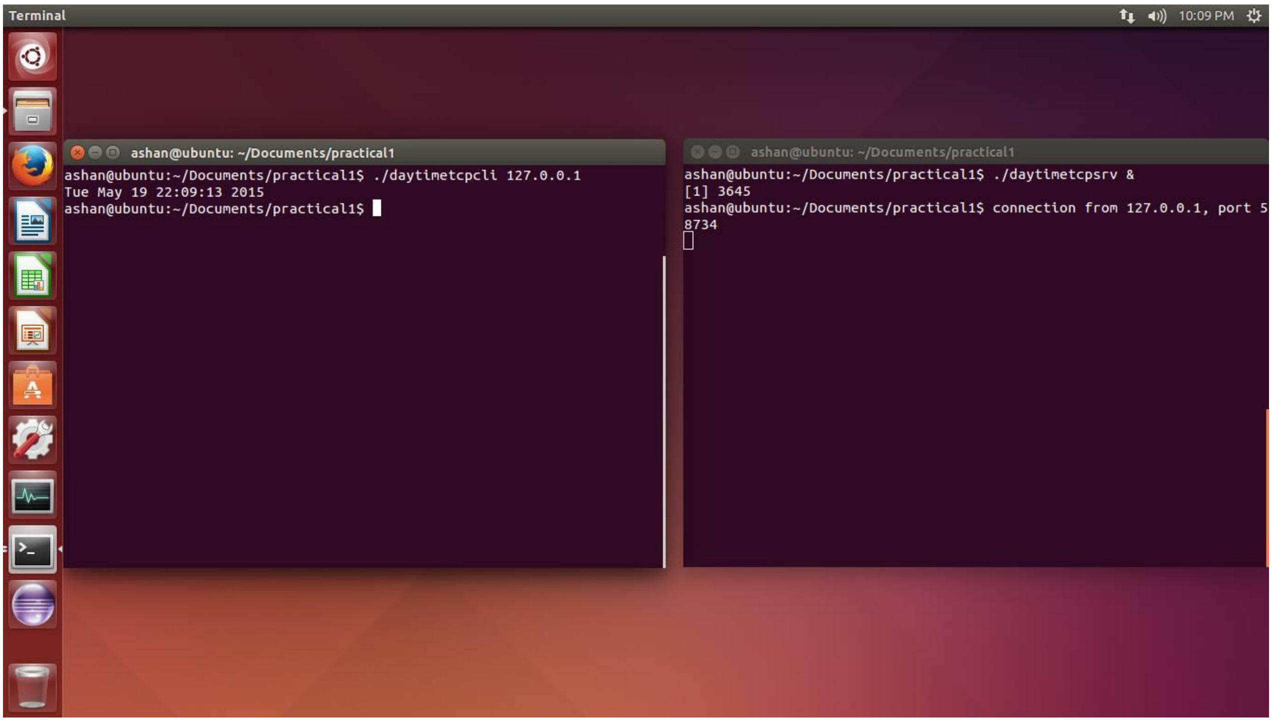




TCP Daytime Client Server Scenario (client on left & server on right)



TCP Daytime Client Server Scenario (client on left & server on right) output



2.6 Sources for test data and analysis

2.6.1 Data collection procedures

first.cc

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");

int
main (int argc, char *argv[])
{
  Time::SetResolution (Time::NS);
  LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
  LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);

  NodeContainer nodes;
  nodes.Create (2);

  PointToPointHelper pointToPoint;
  pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
  pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

  NetDeviceContainer devices;
  devices = pointToPoint.Install (nodes);

  InternetStackHelper stack;
  stack.Install (nodes);

  Ipv4AddressHelper address;
  address.SetBase ("10.1.1.0", "255.255.255.0");

  Ipv4InterfaceContainer interfaces = address.Assign (devices);
```

```
  UdpEchoServerHelper echoServer (9);

  ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
  serverApps.Start (Seconds (1.0));
  serverApps.Stop (Seconds (10.0));

  UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
  echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
  echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
  echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

  ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
  clientApps.Start (Seconds (2.0));
  clientApps.Stop (Seconds (10.0));

  Simulator::Run ();
  Simulator::Destroy ();
  return 0;
}
```

TCP daytime client and server code

daytimetcpcli.c

```
#include	"cc352.h"

int
main(int argc, char **argv)
{
	int	sockfd, n;
	char	recvline[MAXLINE + 1];
	struct sockaddr_in	servaddr;

	if (argc != 2) {
		printf("usage: a.out <IPaddress>");
		return 1;
	}

	if ( (sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0){
		printf("socket error");
		return 1;
	}

	bzero(&servaddr, sizeof(servaddr));
	servaddr.sin_family = AF_INET;
	servaddr.sin_port   = htons(SERV_TCP_PORT);	/* daytime server */
```

```
	if (inet_pton(AF_INET, argv[1], &servaddr.sin_addr) <= 0){
		printf("inet_pton error for %s", argv[1]);
		return 1;
	}

	if (connect(sockfd, (SA *) &servaddr, sizeof(servaddr)) < 0) {
		printf("connect error");
		return 1;
	}

	while ( (n = read(sockfd, recvline, MAXLINE)) > 0) {
		recvline[n] = 0;	/* null terminate */
		if (fputs(recvline, stdout) == EOF) {
			printf("fputs error");
			return 1;
		}
	}
	if (n < 0) {
		printf("read error");
		return 1;
	}

	exit(0);
}
```

daytimetcpsrv.c

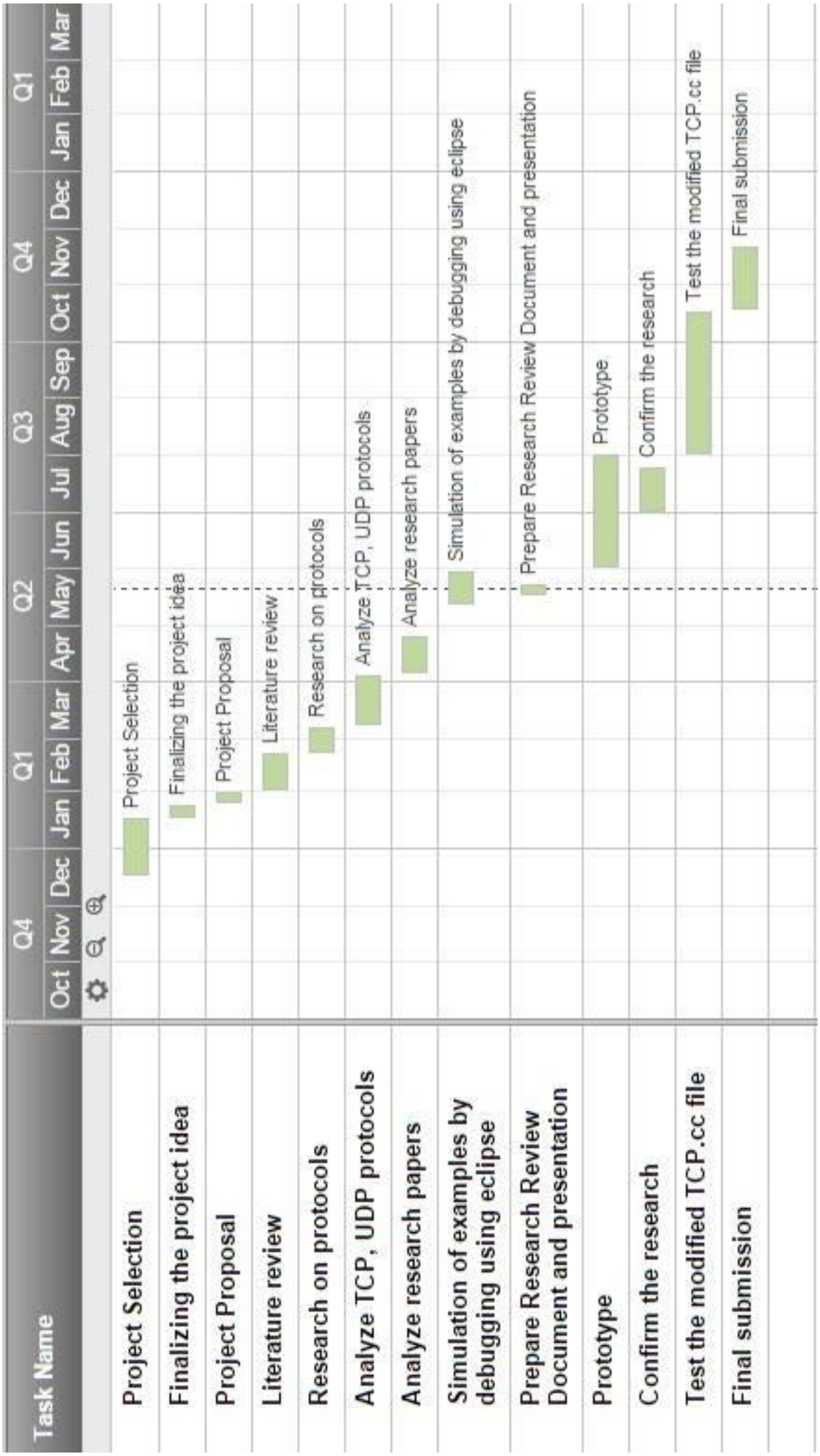
```
#include	"cc352.h"

int
main(int argc, char **argv)
{
	int			listenfd, connfd;
	socklen_t		len;
	struct sockaddr_in	servaddr, cliaddr;
	char			buff[MAXLINE];
	time_t			ticks;
	int yes = 1;
	const char	*ptr;

	if ( (listenfd = socket(AF_INET, SOCK_STREAM, 0)) < 0 ){
		fprintf(stderr, "socket creation failed\n");
		exit (1);
	}
```

```
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family      = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port        = htons(SERV_TCP_PORT); /* daytime server */

    if ( (bind(listenfd, (SA *) &servaddr, sizeof(servaddr))) < 0) {
        fprintf(stderr, "bind failed\n");
        exit (1);
    }


    if ( listen(listenfd, LISTENQ) < 0) {
        fprintf(stderr, "listen failed\n");
        exit (1);
    }

    for ( ; ; ) {
        len = sizeof(cliaddr);
        if ( (connfd = accept(listenfd, (SA *) &cliaddr, &len)) < 0 ) {
            fprintf(stderr, "accept failed\n");
            exit (1);
        }
    if( (ptr = inet_ntop(AF_INET, &cliaddr.sin_addr, buff, sizeof(buff))) == NULL) {
        fprintf(stderr, "inet_ntop error \n");
        exit (1);
    }
        printf("connection from %s, port %d\n", ptr, ntohs(cliaddr.sin_port));

   ticks = time(NULL);
   snprintf(buff, sizeof(buff), "%.24s\r\n", ctime(&ticks));

    if ( write(connfd, buff, strlen(buff)) != strlen(buff)) {
        fprintf(stderr, "accept failed\n");
        exit (1);
    }


        if ( close(connfd) == -1) {
            fprintf(stderr, "accept failed\n");
            exit (1);
        }
    }
}
```

2.7 Anticipated benefits

This research will be analysing and develop a dynamic transporter layer protocol that supports less packet drops and takes a considerably less retransmission time which is able to transmit data in a quick and efficient manner within a wireless environment. This has been identified as a vital factor in real time systems; specifically in way finding applications in order to gather critical information of patients, evade hazardous situations for disable persons, and receive alternative paths in navigation systems in instances of urgent nature.

3 Project plan or schedule



4 Research constraints

Given the start of the research on fine tuning the TCP to be looked into a dynamic transport layer protocol use of Smart Mobile Device Communication, we have noticed a lot of restrictions when it comes to accessing the Socket API at the point of debugging via eclipse. We find it challenging to come up with solutions to develop a complete dynamic protocol given the above constraints as we need to access these methods to determine its behaviour to move forward.

5 Specified deliverables

Design and develop a new Dynamic transport layer protocol for communication in wireless environment in an effective, efficient and reliable manner.

6 Supporting Information

6.1 References

[1]Dhammika H De Silva, Dr.Ian Murray "An investigation in to Dynamic TLP's for Smartphone Communication", in International Conference on Advances in ICT for emerging regions ICTer, 2013, p.194 – 197

[2]Haowei Bai *et al* "Impact of Mobile Wireless Links on TCP Performance"

[3]Hala Eidaw *et al* "Internet Accessibility In High-Speed Vehicles"

[4]Ibtissam El Khayat *et a*l "Improving TCP in wireless networks with an adaptive machine-learnt classifier of packet loss causes"

[5]Kunz.T et al "Transport Layer Fairness and Congestion Control in Multihop Wireless Networks"

[6]Ningning Hu *et al* "Improving TCP Startup Performance using Active Measurements:Algorithm and Evaluation" by

[7]I. E. Khayat, P. Geurts and G. Leduc "Improving TCP in wireless networks with an adaptive machine-learnt classifier of packet loss causes"

[8]H. Zhang "Transport Layer Fairness and Congestion Control in Wireless Ad Hoc Access Networks". July 2006

[9]V. Jacobson, M. J. Karels "Congestion Avoidance and Control". November, 1988

[10] F. Yang, Q. Zhang, W. Zhu and Y. Zhang "An Efficient Transport Scheme for Multimedia over Wireless Internet"

[11] H. Balakrishnan, S. Seshan, E. Amir and R. H. Katz "Improving TCP/IP Performance over Wireless Networks"

[12] M. C. Chan and R. Ramjee "Improving TCP/IP Performance over Third Generation Wireless Networks"

[13] F. Wang, Y. Zhang "Improving TCP Performance over Mobile Ad-Hoc Networks with Out-of-Order Detection and Response"

[14] M. S. Akbar, S. Z. Ahmed, M. A. Qadir, "*Performance Optimization of Transmission Control Protocol in Heterogeneous Wireless Network during Mobility*", Vol.8 No.8., IJCSNS International Journal of Computer Science and Network Security,2008.

[15] S. Zaher, "*Performance Enhancement of Transmission Control Protocol over Wireless Ad-hoc Networks*", Vol.4, International Journal of Emerging Technology and Advanced Engineering, 2014.

[16] Hongqiang Zhai, Xiang Chen, and Yuguang Fang," *Improving Transport Layer Performance in Multihop Ad Hoc Networks by Exploiting MAC Layer Information*", Vol.6 No.5, IEEE Transactions on Wireless Communications, 2007.

6.2 Appendix

Congestion control: A technique for monitoring network utilization and manipulating transmission or forwarding rates for data frames to keep traffic levels from overwhelming the network medium.

Mobile nodes: A node is either a connection point, a redistribution point communication endpoint

Packet loss: Packet loss is the failure of one or more transmitted packets to arrive at their destination. This event can cause noticeable effects in all types of digital communications

Multi-hop: Multi-hop or ad hoc, wireless networks use two or more wireless hops to convey information from a source to a destination.

Wireless ad hoc networks: The ad hoc network does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data for other nodes, so the determination of which nodes forward data is made dynamically on the basis of network connectivity.