

PERFORMANCE ANALYSIS OF THE TRANSMISSION CONTROL
PROTOCOL OVER LOW EARTH ORBIT SATELLITE COMMUNICATION
SYSTEMS

A Thesis Presented To

The Faculty of the

Fritz J. and Dolores H. Russ
College of Engineering and Technology
Ohio University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Rahul Sangal

August, 1999

THIS THESIS ENTITLED
“PERFORMANCE ANALYSIS OF THE TRANSMISSION CONTROL
PROTOCOL OVER LOW EARTH ORBIT SATELLITE COMMUNICATION
SYSTEMS”

by Rahul Sangal

has been approved

for the School of Electrical Engineering and Computer Science
and the Russ College of Engineering and Technology

Dr. Shawn Osterman
Assistant Professor of Computer Science

Warren K. Wray, Dean
Fritz J. and Dolores H. Russ
College of Engineering and Technology

ACKNOWLEDGMENTS

I would like to thank Dr. Shawn Osterman for giving me an opportunity to work with him. I would like to acknowledge his assistance in providing thoughtful insights and wonderful working environment. I would like to thank my classmates and labmates, most notably Issam, Naseef, Eric, Vitaly, Kunjan, Usman and Matai for helping me out of all the problems I encountered in two years of my course work. I'd like to acknowledge the funding granted to me by the School of Electrical Engineering and Computer Science. I would like to thank my friend Nayana for her constant support.

DISCARD THIS PAGE

TABLE OF CONTENTS

	Page
LIST OF TABLES	iv
LIST OF FIGURES	v
1. INTRODUCTION	1
1.1 Internet Over LEO Satellite Network	2
1.2 Transmission Control Protocol	6
1.2.1 Round Trip Time and Retransmission Time Out	8
1.2.2 Effects of RTO Estimate on the Performance of TCP	9
1.3 Motivation for this Thesis	10
1.4 Organization of this Thesis	12
2. SOFTWARE EMULATOR FOR LEO SATELLITE NETWORK	13
2.1 Ohio Network Emulator (ONE)	13
2.2 Communication Model	14
2.3 Delay Analysis	17
2.3.1 Calculation of T_{uplink}	18
2.3.2 Calculation of T_{cross_n}	19
2.3.3 Calculation of T_{sat_n}	21
2.3.4 Calculation of $T_{downlink}$	21
2.4 Effect of Handoff on End-to-End Delay	21
2.5 Simplifying Assumptions and Their Implications	24
2.5.1 Effect of Earth's Rotation	24
3. EFFECT OF ROUND TRIP TIME VARIATION IN A LEO SATELLITE NETWORK ON THE PERFORMANCE OF TCP	25
3.1 Experimental Setup	25
3.2 RTT Variations due to Changes in Propagation Delays	26

	Page
3.2.1 Experiment Number 1	27
3.2.2 Experiment Number 2	30
3.2.3 Summary	35
3.3 Traffic Model for a LEO Satellite Network	35
3.3.1 Traffic Simulation for LEO Satellite Systems	36
3.4 Problem with Retransmission Time Out Algorithm	37
3.4.1 Experiment Number 1	37
3.4.2 Experiment Number 2	43
3.5 Problem with Idle Connections	46
3.5.1 Experiment No. 1	47
3.6 Problem with Fast Retransmit Algorithm	49
4. CONCLUSIONS AND RECOMMENDATIONS	51
4.1 Conclusions	51
4.2 Future Work	53
BIBLIOGRAPHY	54
 APPENDIX	
A. RETRANSMISSION TIME OUT AND ROUND TRIP TIME CALCULATION FOR SUNOS	56
B. FILE FORMAT FOR ONE	62
C. HOW TO READ A TCP TIME SEQUENCE GRAPH	64
D. EXPLANATION FOR THE PERIODIC VARIATION IN RTT	66
ABSTRACT	69

LIST OF TABLES

Table	Page
2.1 Configuration Parameters of a LEO Satellite Network to Study the Variables involved in Calculating End-to-End Delay	22
3.1 Configuration Parameters of a LEO Satellite Network for Experiment Number 1 to Show Variations in RTT Due to Change in Propagation Delay.	27
3.2 Configuration Parameters of a LEO Satellite Network for Experiment Number 2 to Show Variations in RTT Due to Change in Propagation Delay.	30
3.3 Configuration Parameters of a LEO Satellite Network for Experiment Number 1 to Show the Effects of Increase in User Population.	38
3.4 Configuration Parameters of a LEO satellite network for Experiment Number 2 to Show the Effects of Increase in User Population.	44
3.5 Configuration Parameters of a LEO satellite network for the Experiment Showing the Effect of Restart of Idle Connection.	47

LIST OF FIGURES

Figure	Page
1.1 Packet Transfer between a User Terminal and Internet Backbone	4
1.2 Packet Transfer between Source and Destination User Terminals	5
2.1 ONE Software Emulator Setup	13
2.2 Spherical Coordinate System	15
3.1 Experimental Setup	26
3.2 RTT vs Time Graph to Show Variations Due to Changes in Propagation Delay for Data Transfer between Locations $(15^\circ, 1^\circ)$ and $(15^\circ, 73^\circ)$	28
3.3 Time-Sequence Graph for the Data Transfer between Locations $(15^\circ, 1^\circ)$ and $(15^\circ, 73^\circ)$	29
3.4 RTT vs Time Graph to Show Variations Due to Changes in Propagation Delay for Data Transfer between Locations $(15^\circ, 20^\circ)$ and $(135^\circ, 44^\circ)$	31
3.5 RTO vs Time Graph to Show Calculated RTO based on the Samples of RTT for Data Transfer between Locations $(15^\circ, 20^\circ)$ and $(135^\circ, 44^\circ)$	32
3.6 $(RTO - RTT)$ vs Time Graph for the Calculated RTO based on the Samples of RTT for Data Transfer between Locations $(15^\circ, 20^\circ)$ and $(135^\circ, 44^\circ)$	33
3.7 Time-Sequence Graph for Data Transfer between Locations $(15^\circ, 20^\circ)$ and $(135^\circ, 44^\circ)$	34
3.8 RTT vs Time Graph to Show the Increase in RTT due to Increase in User Population from 200 to 6000.	39
3.9 RTO vs Time Graph to Show Calculated RTO based on the Samples of RTT	40

Figure	Page
3.10 ($RTO - RTT$) vs Time Graph for the Calculated Values of RTO	41
3.11 Time-Sequence graph to Show the Effect of Increase of User Population from 200 to 6000 on the Packet Loss.	43
3.12 RTT vs Time Graph to Show the Increase in RTT due to Increase in User Population from 20000 to 60000.	45
3.13 Time-Sequence Graph to Show the Effect of Increase of User Population from 20000 to 60000 on the Packet Loss.	46
3.14 RTT vs Time Graph to Show the Increase in RTT after Restart of an Idle Connection.	48
3.15 Time Sequence Graph to Show the Effect of Restart of an Idle Connection.	49
A.1 RTT vs Time Graph Showing an Increase of 100ms in RTT.	57
A.2 Time-Sequence graph to Show the Effect of Increase in RTT by 100ms on Packet Loss.	58
A.3 RTT vs Time Graph Showing an Increase of 500ms in RTT.	59
A.4 Time-Sequence Graph to Show the Effect of Increase in RTT by 500ms on Packet Loss.	60
C.1 Time Sequence Graph	65
D.1 Time Sequence Graph	67
D.2 RTT vs Time Graph	68

1. INTRODUCTION

The Internet has the potential to serve as the common underlying technology for a whole set of applications ranging from connection-less applications, such as paging or database access, to connection oriented applications, such as file transfer or multi-point multimedia communications[CCJ98]. Private companies are striving to provide truly seamless global access to the Internet for the public. For more than three decades, geostationary satellites have been providing commercial space-based communications. Due to high power requirements and signal delay caused by the high altitude of geostationary satellites, a large number of applications, including essential Internet technologies such as the World Wide Web, are adversely affected[Koh98]. This global approach has sparked the development of new satellite communication systems called Low Earth Orbit (LEO) satellite systems. LEO satellite networks provide Internet access to regions of the world where there is little or no communication infrastructure [PRJT99]. We are looking at a solution in terms of a high capacity broadband network that combines global coverage and low latency of a LEO satellite constellation with the flexibility and robustness of the Internet.

Communication over LEO satellite networks shows different characteristics compared to the traditional satellite or wired networks. The effect of such a communication environment on the working of Transmission Control Protocol(TCP), which forms the backbone of Internet protocol communication, forms the basis of research being conducted as a part of this thesis work. The focus of this thesis work is to study and analyze the variations in Round Trip Time(RTT) encountered by TCP/IP packets in a LEO satellite network.

1.1 Internet Over LEO Satellite Network

A LEO satellite network consists of a ground segment (user terminals, network gateways and, network control systems) and a space segment (the satellite-based switch network that provides communication links among terminals) [tel99].

Ground Segment - The ground segment of the satellite network consists of :

User Terminal - The user terminal in a network communicates directly with the satellite network and supports a variety of data rates. In order to communicate with the satellite, a user terminal is equipped with packet radio modems and radio transmitter/receiver hardware, permitting it to send and receive packets from the satellites. For most of the LEO satellite networks, standard terminals will support configurations that operate at multiples of the 16kBps¹ basic channel payload up-to 2.048MBps². Basic channels (each 16kBps channel) can be aggregated to provide higher data rates. The user terminal can support multiple simultaneous network connections [Koh98].

Network Gateway - The Network Gateway acts as the interface between the satellite network and the terrestrial Internet. Network Gateways are the gateways to the public switched networks. For most of the LEO satellite networks, these gateways communicate with the satellite networks using gigalink terminals that operate at OC-3 rate (155.52MBps) [Koh98] .

Network Control Systems - Some gateways also act as a source of control signals for the satellites for altitude control, station keeping and internal house keeping functions. They are called Network Operations and Control Centers (NOCCs) and Constellation Operations Control Centers (COCCs) [Woo95].

¹1 kBps = 1024 bytes/second

²1 MBps = 1024*1024 bytes/second

Space Segment - The space-based network uses fast packet switching. To achieve global coverage, the space segment consists of a constellation of satellites, which are arranged in equally spaced orbits. Each of these satellites is providing a service area in the form of a cell for the users under the coverage of this satellite. Each satellite is a node in the fast-packet-switch-network and has inter-satellite communication links with the satellites in the same and adjacent orbits. From the network viewpoint, a large constellation of inter-linked switch nodes offers a number of advantages in terms of service quality, reliability, and capacity.

When a user terminal connected to a LEO satellite network tries to establish an Internet connection, it sends the control signals to the satellite available to receive the transmission. Once the connection is established, the user terminal starts sending packets to the satellite. Since the LEO satellites are revolving with a high orbital velocity with respect to the motion of earth, the maximum in-view time of a satellite with respect to a fixed point on earth is not more than 10-20 minutes. When a connection is transferred between satellites to maintain a continuous flow of traffic it is called a handoff.

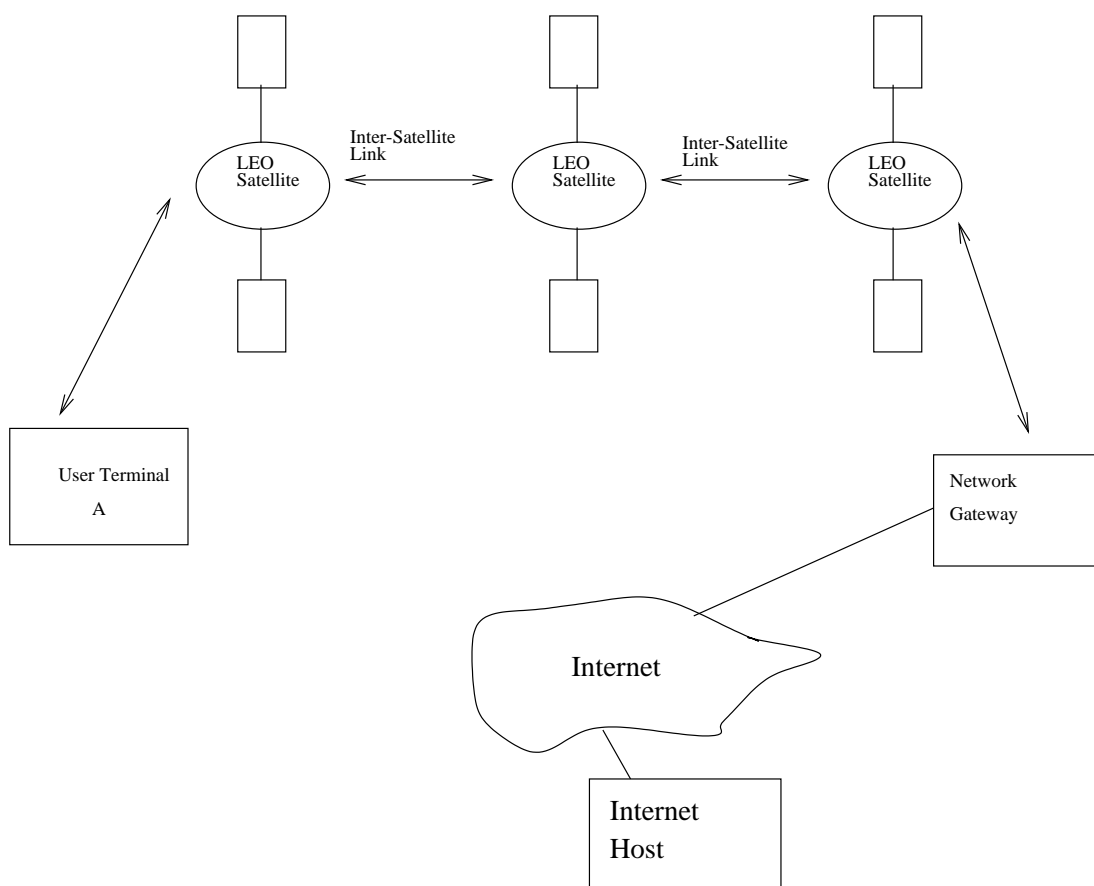


Figure 1.1 Packet Transfer between a User Terminal and Internet Backbone
 This figure shows how the packets are transferred between the user terminal connected to the LEO satellite system and host connected to the terrestrial Internet.

If the destination address of the packets is one of the hosts that is connected to the terrestrial Internet, then the packets are routed to the network gateway by employing inter-satellite links, as shown in Figure 1.1. Similarly, the packets from the Internet hosts that are addressed for the users connected to the LEO satellite network are routed to the network gateways through the public switched network first and then the network gateway transmits the packets to a connecting satellite where the packets are transferred to the appropriate satellite, covering the addressed

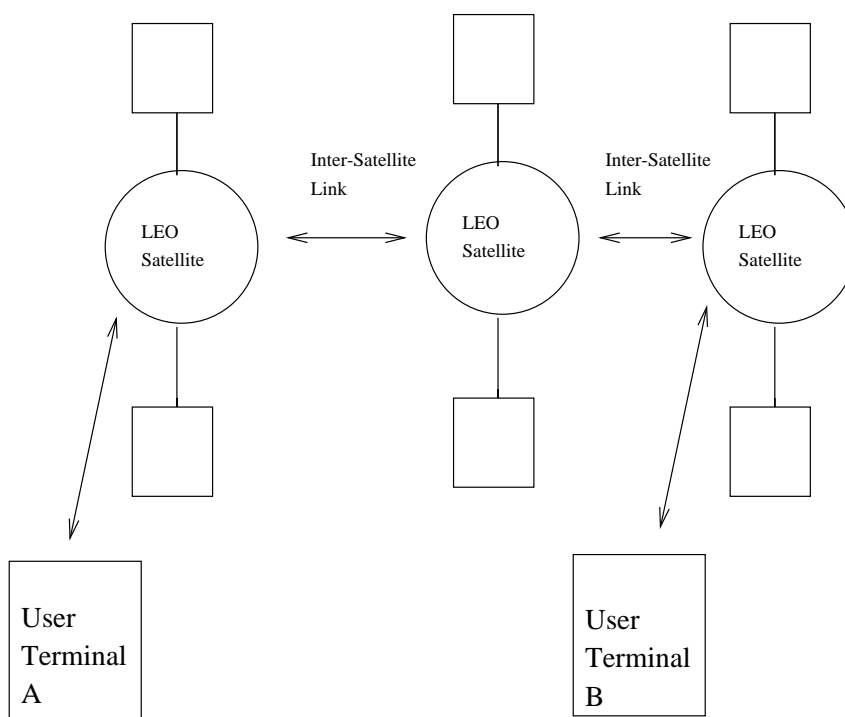


Figure 1.2 Packet Transfer between Source and Destination User Terminals
 This figure shows how the packets are transferred between the source and destination user terminals connected to LEO satellite system.

user terminal. In this manner, the user terminals connected to the LEO satellite network are able to remain a part of the Internet. The Network Gateway routes the packets to the appropriate host using the public switched network. If the packets are destined for another user terminal connected to the LEO satellite network, then the packets are transferred to the destination by forwarding them from satellite to satellite, using inter-satellite links, as shown in Figure 1.2.

1.2 Transmission Control Protocol

The Transmission Control Protocol(TCP) is a connection oriented, end-to-end reliable protocol[Com95]. TCP is used by a number of major Internet applications to provide reliable data delivery (e.g. SMTP[Pos82], HTTP[BLFN96], and FTP[PR85]).

1. TCP is connection oriented because two hosts must setup and establish a connection before any data exchange takes place.
2. TCP provides reliability by recovering from data that is damaged, lost, duplicated, or delivered out of order by the Internet communication system. This is achieved by assigning a sequence number to each octet transmitted and requiring a positive acknowledgment(ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. TCP uses a separate algorithm to compute this timeout interval, it is described in Section 1.2.1. At the receiver, the sequence numbers are used to correctly order segments that may be received out of order and to eliminate duplicates.
3. TCP utilizes *sliding window protocol*[Com95] to achieve flow control. A sliding window protocol makes efficient use of network bandwidth as it allows the sender to transmit multiple packets before waiting for an ACK. When an ACK is received by the sender, the window advances to allow one or more segments to be transmitted.
4. TCP uses a set of congestion control algorithms[APS99] that defines the behavior of TCP in the event of network congestion. Network congestion usually occurs when the sending application is injecting an inappropriate amount of data into the network. This causes the queuing buffers of the intermediate router to be flooded with excessive data, resulting in a packet loss. If the sending rate of the application is not controlled, it can lead to a congestion collapse[All97]. The various congestion control algorithms[APS99] used by TCP are :

Slow Start and Congestion Avoidance - The slow start and congestion avoidance algorithms[APS99] allow TCP to increase the data rate without overwhelming the intermediate routers. To implement these algorithms, two variables are added to the TCP connection state:

- (a) Congestion Window (*cwnd*): It is sender side limit on the amount of the data that it can send before receiving an ACK.
- (b) Slow Start Threshold (*ssthresh*): It is used to determine whether the slow start or congestion avoidance algorithm is to be used to control data transmission.

Slow start is used in the beginning of a TCP connection to slowly increase the amount of unacknowledged data TCP injects into the network. In the beginning of slow start, the *cwnd* is initialized to 1 segment size. For each ACK received, the *cwnd* is increased by 1 segment size. This process of exponential growth of window size continues until *cwnd* equals or exceeds *ssthresh* or a segment loss is detected. A segment loss is indicated when the TCP's retransmission time out (RTO) timer expires, in which case the *cwnd* is again initialized to 1 segment size and *ssthresh* is reduced. The sender again enters into slow start until *cwnd* meets or exceeds *ssthresh*. After *cwnd* meets or exceeds *ssthresh*, the sender enters into the congestion avoidance phase. During the congestion avoidance, *cwnd* is incremented by 1 segment size per RTT. It continues until congestion is detected, in which case the connection again enters into slow start.

Fast Retransmit/ Fast Recovery - TCP generates duplicate acknowledgments when a segment arrives out of order. The fast retransmit algorithm[APS99] uses the arrival of 3 duplicate acknowledgments as an indication that a segment has been lost. TCP connection at the senders side retransmits the lost segment without waiting for the RTO to occur. TCP connection

at the senders side also uses the lost segment as an indication of network congestion as it enters into fast recovery phase and reduces the transmission rate. In the fast recovery phase, the *cwnd* is reduced to half and *ssthresh* is set to the new value of *cwnd*. Now TCP artificially inflates the *cwnd* by three segment size. For each additional duplicate ACK the *cwnd* is inflated by 1 segment size. When the ACK acknowledging all the new data arrives, the *cwnd* is set to the *ssthresh*. Now the fast recovery phase is over and connection enters into congestion avoidance mode.

1.2.1 Round Trip Time and Retransmission Time Out

Networking protocols have to cope with the problem of determining when the sender should retransmit a packet that has not been acknowledged by the receiver[Jai86]. The timeout algorithm plays an important role in the overall stability of networking protocols. Dynamically estimating the RTT, the interval between the sending of the packet and receipt of the ACK, is a key function of TCP[KP87]. If a packet remains unacknowledged for too long, it is assumed to be lost and is to be retransmitted. Estimated RTT is used to determine when the retransmission should occur. The interval for which TCP waits before retransmitting an unacknowledged segment is Retransmission Time Out (RTO). In simple words, RTO is the estimate on the part of the sender on how much time it should take for a segment to be acknowledged.

TCP uses adaptive retransmission algorithm to arrive at a reasonable value for RTO. This algorithm is basically an estimation process[Mil83]. TCP maintains a current estimate of RTT. To collect the samples of RTT, TCP records the time at which each segment is sent and the time at which the ACK for that particular segment is received. Subtracting the two values, TCP obtains the current RTT estimate³.

³The exact implementation and accuracy of these calculations is operating system dependent. The exact implementation of RTO estimation and timer granularity for SunOS is discussed in Appendix A

TCP software maintains an estimated value of RTT, which is smoothed and updated after the collection of each of the new RTT sample, according to following equations[Com95]:

$$Smoothed_RTT = Old_RTT + (\delta * (New_RTT_Sample - Old_RTT)) \quad (1.1)$$

TCP calculates the value of RTO based on the current estimate of *Smoothed_RTT* according to the following equations:

$$DEV = Old_DEV + \rho * (|New_RTT_Sample - Old_RTT| - Old_DEV) \quad (1.2)$$

$$RTO = Smoothed_RTT + \eta * DEV \quad (1.3)$$

where

DEV is estimated mean deviation.

δ is a fraction between 0 and 1 that controls the effect of new samples on weighted average.

ρ is a fraction between 0 and 1 that controls how quickly DEV is affected.

η is a fraction controlling the effect of DEV on the estimation of RTO.

1.2.2 Effects of RTO Estimate on the Performance of TCP

The correct estimate of the RTO plays an important role in the overall stability of the network. Values of RTO smaller than the average RTT may flood a network

with duplicate copies of the packets and can lead to unwanted congestion. On the other hand, values of RTO higher than the average RTT lead to the waste of network capacity and decrease throughput.

It has been observed that, if the Internet is running at 50% of its capacity, we expect a RTT to vary by a factor of 4 and when it is running at 80%, we expect a variation by a factor of 10[Com95]. This suggests that there is already a great amount of variation in RTT for TCP packets in existing terrestrial network. In a terrestrial network, major factors that contribute to this variation are queuing delay and network congestion.

Some of the experts in the area of TCP research believe that

TCP will become confused if the jitter for a transmission exceeds the time required to send, receive and acknowledge the transmission. If variable latency exceeds the round-trip timing, TCP may interpret jitter as packet loss and begin a re-transmittal. If this happens, large data transfer would be significantly slowed [jit99] .

1.3 Motivation for this Thesis

1. Most of the analysis work in the area of LEO satellite networks has been conducted in the form of various simulations on various software packages like Satlab, Matlab etc.[RD]. Network simulators are useful for testing the performance of networks in term of parameters internal to the communication network such as capacity, fault tolerance, etc., but they make it difficult to test the performance of actual protocols. As a part of this thesis work, Ohio Network Emulator(ONE)[ACO97] is used for all the performance analysis. ONE alters real network traffic between nodes in a physical network to model the characteristics of a LEO satellite network.

2. After a thorough literature survey in terms of research and development taking place in the area of LEO satellite based Internet communication, it has been observed that there is little evidence of experiments being conducted to test the performance of TCP. Therefore, one of the guiding factors behind this thesis work is to modify ONE so that characteristics of various kinds of LEO satellites network can be configured and ONE can act as a testbed for the testing of TCP protocol over LEO satellite networks.
3. As described in Section 1.2.2, the correct estimation of RTO plays an important role in the efficient working of TCP. After studying the architecture of a LEO satellite network, there is a strong indication that the TCP/IP packets will encounter high variations in delay, otherwise known as variable latency or jitter. This variation will compound the problem of already existing jitter in RTT over terrestrial networks. As a part of this thesis work, two main factors that contribute to this high variation are explored :
 - (a) It has been observed that random and constantly changing distances between satellites in a LEO constellation mean that

TCP's window duration measurement algorithms will be constantly rubber-banding and snapping[jit99].

Since the LEO satellites are moving with respect to earth at high orbital velocities, the propagation distance between a fixed user terminal on earth and the satellites that route the packets is constantly changing. This creates a variation in the propagation delay.
 - (b) The LEO satellites provide global connectivity to the Internet and cover many diverse regions in various parts of the globe. Many of these regions are urban areas with numerous users connecting to these satellites, while some of the areas can be vast oceans with no traffic at all. This issue is referred to as *geographical traffic non-uniformity*[Jam98]. The effect of this

geographical traffic non-uniformity is that the *processing delay* and *queuing delay* for each of the satellite nodes involved in routing the packets from the source and destination terminals is constantly changing, at a far greater rate than the terrestrial networks. This introduces an additional variation in the delay encountered by the TCP packets traversing the LEO satellite network.

1.4 Organization of this Thesis

This thesis is organized as follows:

The first step in this research is to develop a communication model encompassing all the vital characteristics of a LEO satellite network. This communication model forms the basis of all the subsequent software development. Details of this communication model are mentioned in Section 2.2.

The second step is to modify the existing code of ONE so that it can be configured to emulate the delay characteristics of a LEO satellite network. Details of the procedure used to calculate the packet delays is discussed in Section 2.3.

The experimental setup used to conduct the experiments is described in Section 3.1.

Details of performance analysis and study of RTT variations for the transfer of TCP packets, using LEO satellite constellations is discussed in Chapter 3.

A summary of the collected information is described in Chapter 4.

2. SOFTWARE EMULATOR FOR LEO SATELLITE NETWORK

The objective of this thesis work is to study the effects of delay characteristics of a LEO satellite network on the working of TCP. The approach that has been adopted for this work is to alter the real traffic between communicating applications. For the purpose of making the TCP packets between two communicating nodes encounter delay that the packets would encounter in a LEO satellite network, Ohio Network Emulator (ONE) was used.

2.1 Ohio Network Emulator (ONE)

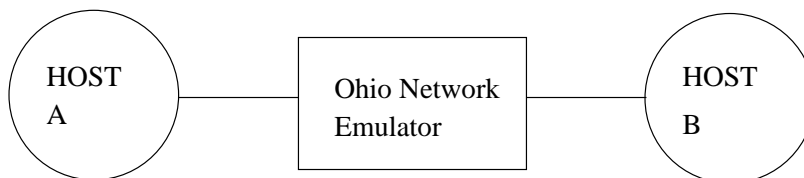


Figure 2.1 ONE Software Emulator Setup

ONE run on a machine that has two network interfaces. The hosts on these two network interfaces communicate via the emulator.

ONE [ACO97] is a software program that runs on a machine with two Ethernet interfaces. The code of ONE was written as part of a project to design an emulator that can be configured to act as a satellite link between two points of communication. A network topology is set, as shown in Figure 2.1, where two hosts are connected to the two separate Ethernet interfaces of the machine running ONE. ONE collects the packets from one of the interfaces, queues them for a particular amount of calculated

time, and forwards them to the other Ethernet interface. The amount of time a packet is queued before being forwarded to the other Ethernet interface is the delay the packet would encounter if it were passed through a real LEO satellite network. In this manner, by calculating the required time delay and making ONE forward the packets between two hosts that are connected to those Ethernet interfaces, we are able to make TCP packets encounter the same kind of delay characteristics as they would have encountered in a LEO satellite network.

To come up with a procedure to calculate the delay time that each packet will encounter, a communication model that exemplifies the characteristics of all major LEO satellite networks is programmed in ONE. The procedure reads a set of parameters from the input file¹. When ONE is running, for each packet that arrives at one of the Ethernet interfaces, the procedure calculates the amount of delay and ONE queues that packet for that particular amount of time before forwarding it to the other Ethernet interface.

2.2 Communication Model

In order to obtain a framework to model diverse kinds of LEO satellite communication systems, a thorough literature survey was conducted. After the completion of study of the research and development in the area of design of LEO satellite networks, the following features were found to be best suited to characterize the design of most of the LEO satellite networks:

1. The first step towards designing a LEO satellite network that will provide coverage to all parts of the globe is to select a geometric constellation where all the satellites act as nodes in the constellation. In our design, we are studying LEO satellite networks with a circular polar orbit. In circular polar orbit constellation, to achieve global coverage the network consists of a total of T satellites,

¹Detail description of the file format is attached in Appendix B

with m satellites placed in each of n orbital planes, so $T = m * n$ [Ste97]. The ascending nodes of the orbital planes are evenly spaced at interval of $360/n$, in the reference plane. The m satellites are also evenly spaced at intervals of $360/m$ within each orbital plane.

2. In the design, we use a spherical co-ordinate system to represent the position of various satellites and the location of source and destination user terminals on the earth surface. Figure 2.2 shows the representation of any point in a 3D space using a spherical co-ordinate system. Any point in a 3D space can be represented by a combination of three dimensions (r, ϕ, θ) , where:

r - is the distance from the center of the earth.

ϕ - is the horizontal angular displacement ($0 \leq \phi \leq \pi$).

θ - is the vertical angular displacement ($0 \leq \theta \leq 2\pi$).

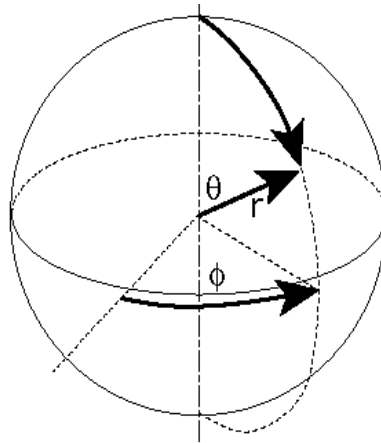


Figure 2.2 Spherical Coordinate System

This figure shows the representation of any point in a spherical coordinate system.

3. In terms of coverage area of each LEO satellite, the coverage of the whole globe is divided equally among all the satellites at all points in time.
4. Each satellite in the constellation stores and forwards the packets to one of the adjacent satellites with which it has inter-satellite links [MHJC95]. Each satellite has a fixed queue size.
5. As a part of the configuration for a LEO satellite network, the following variables are taken as input from the user:

Height of the Satellite is the height of LEO satellites in km.

Rotation Time is the total time it takes for a satellite to cover one complete revolution around earth surface. It is measured in seconds.

Number of Satellites per orbit.

Number of Orbits.

Source and Destination Up-link Bandwidth.

Source and Destination Down-link Bandwidth.

Inter-Satellite Bandwidth.

Location of Source and Destination user terminals.

6. In terms of routing algorithm, a path specifying the satellites to be used for transferring the packets from the source to the destination user terminal is calculated. This path remains static through out the connection. The path is calculated on the basis of minimum hops, so that the minimum number of satellites are used for forwarding the packets between the source and destination user terminals.

2.3 Delay Analysis

The purpose of the communication model described above is to be able to calculate the end-to-end delay, a packet traveling from the source to destination terminal will encounter in a LEO satellite network. The following equation was used to calculate end-to-end delay:

$$T_{end-to-end} = T_{uplink} + T_{cross_1} + \dots + T_{cross_{n-1}} + T_{sat_1} + \dots + T_{sat_n} + T_{downlink} \quad (2.1)$$

When the ONE is started, the position of all the satellites in the LEO satellite network is initialized. Since the height of all the satellites in the constellation is considered to be constant, the position of each satellite in the constellation is specified by (ϕ, θ) . Here ϕ represents the horizontal displacement or the angular displacement of the orbit in which a particular satellite is positioned. Also θ represents the vertical displacement or the angular displacement of the satellite in particular orbit. As soon as a packet arrives at any one of the Ethernet interfaces of the machine running ONE, the procedure to calculate the $T_{end-to-end}$ is called. This procedure updates the position of θ for all the satellites in the LEO satellite network. This update of θ is based on the rotation time of each of the satellite, as:

$$\theta_{new} = \theta_{old} + (2 * \pi * time_diff) / (Rotation_time) \quad (2.2)$$

where $time_diff$ is the time interval between the arrival of the last packet when θ was last updated and the arrival of new packet. In this manner the position of all the satellites in the network are updated. Now the calculations of all the components of $T_{end-to-end}$ is done.

2.3.1 Calculation of T_{uplink}

T_{uplink} component of the total delay is a combination of two factors:

Queuing Delay - T_{uplink} includes the queuing delay encountered by packets at the source terminal before being transmitted to the LEO satellite network. Queuing Delay depends on the uplink bandwidth of the source terminal. Queuing Delay is calculated based on the Equation 2.3.

$$Queuing_Delay = (Bytes_in_Queuing_Buffer + Packet_Size) / (Uplink_Bandwidth) \quad (2.3)$$

Propagation Delay - T_{uplink} also includes the propagation delay between the source terminal and the connecting satellite. This component of T_{uplink} depends on the relative distance between the source terminal and the connecting satellite. Since the connecting satellite is always moving relative to the source terminal, this relative distance is also changing. Each time a packet arrives at one of the Ethernet interfaces, the position of all the satellites is updated according to Equation 2.2. After updating the network, the position of the satellite covering the source terminal is determined. After the position of the satellite covering the source terminal is determined the distance between this satellite and the source terminal is calculated using the formula:

$$x_{sat} = (R + h) * \cos\theta_{sat} * \cos\phi_{sat} \quad (2.4)$$

$$y_{sat} = (R + h) * \cos\theta_{sat} * \sin\phi_{sat} \quad (2.5)$$

$$z_{sat} = (R + h) * \sin\theta_{sat} \quad (2.6)$$

$$x_{source} = R * \cos\theta_{source} * \cos\phi_{source} \quad (2.7)$$

$$y_{source} = R * \cos\theta_{source} * \sin\phi_{source} \quad (2.8)$$

$$z_{source} = R * \sin\theta_{source} \quad (2.9)$$

$$Distance = \sqrt{(x_{sat} - x_{source})^2 + (y_{sat} - y_{source})^2 + (z_{sat} - z_{source})^2} \quad (2.10)$$

where:

R is the radius of the earth.

h is the height of the satellites.

Distance is the distance between the source terminal and the connecting satellite.

Now propagation delay between the source terminal and the connecting satellite is given by Equation 2.11

$$Delay = (Distance)/(Speed_of_Light) \quad (2.11)$$

2.3.2 Calculation of T_{cross_n}

After a packet arrives at a Ethernet interface and the procedure is called to calculate the end-to-end delay, a route in terms of satellites forwarding the packets between the source and destination user terminals is calculated. After this route is calculated

and the position of each satellite in the network is updated, one of the factors adding to the end-to-end delay is the propagation delay between the satellites which use intersatellite links to forward the packets. Since the satellites are also moving at high orbital velocities, this delay is also variable. To calculate the propagation delay between satellites $n - 1$ and n in the routing path, the distance between these satellites based on their positions in the spherical co-ordinate system is calculated. If the position of n_{th} satellite in the routing path is represented by (ϕ_n, θ_n) , the distance is calculated based on Equation 2.18.

$$x_{sat_{n-1}} = (R + h) * \cos\theta_{n-1} * \cos\phi_{n-1} \quad (2.12)$$

$$y_{sat_{n-1}} = (R + h) * \cos\theta_{n-1} * \sin\phi_{n-1} \quad (2.13)$$

$$z_{sat_{n-1}} = (R + h) * \sin\theta_{n-1} \quad (2.14)$$

$$x_{sat_n} = (R + h) * \cos\theta_n * \cos\phi_n \quad (2.15)$$

$$y_{sat_n} = (R + h) * \cos\theta_n * \sin\phi_n \quad (2.16)$$

$$z_{sat_n} = (R + h) * \sin\theta_n \quad (2.17)$$

$$Distance = \sqrt{(x_{sat_{n-1}} - x_{sat_n})^2 + (y_{sat_{n-1}} - y_{sat_n})^2 + (z_{sat_{n-1}} - z_{sat_n})^2} \quad (2.18)$$

Now T_{cross_n} between the two satellites is given by Equation 2.19

$$T_{cross_n} = (Distance)/(Speed_of_Light) \quad (2.19)$$

2.3.3 Calculation of T_{sat_n}

T_{sat_n} is the queuing delay the packet encounters at each of the satellite nodes. It depends on the inter-satellite bandwidth. After the route from the source to the destination user terminal is calculated, the end-to-end delay is calculated based on the queuing delay encountered by the packet at each of the satellite node. It is calculated for each of the satellite nodes and added to $T_{end-to-end}$. T_{sat_n} is calculated based on the Equation 2.20

$$T_{sat_n} = (No_of_Bytes_in_Queuing_Buffer + Packet_Size) / (Inter_Satellite_Bandwidth) \quad (2.20)$$

2.3.4 Calculation of $T_{downlink}$

$T_{downlink}$ is calculated in exactly the same manner as T_{uplink} . It includes the queuing delay encountered at the last satellite involved in routing the packets. This queuing delay is calculated using Equation 2.21.

$$Queuing_Delay = (No_of_Bytes_in_Queuing_Buffer + Packet_Size) / (DownLink_Bandwidth) \quad (2.21)$$

It also includes the propagation delay between the last satellite involved in routing the packet and the destination user terminal. It is calculated in the same manner as the propagation delay in T_{uplink} .

2.4 Effect of Handoff on End-to-End Delay

To study the effect to handoff on the values of the variables used to calculate the end-to-end delay, the following example is considered. Table 2.1 shows the values

of different parameters used in the experiment. Let us consider a source terminal located at $(15^\circ, 15^\circ)$ and the destination terminal located at $(45^\circ, 30^\circ)$. Now we are considering a constellation of satellites with 6 orbits and 12 satellites per orbit.

Table 2.1 Configuration Parameters of a LEO Satellite Network to Study the Variables involved in Calculating End-to-End Delay. :

This Table gives the configuration parameters of a LEO Satellite constellation. These parameters are used to study the effect of configuration on calculations of $T_{end_to_end}$

Number of Satellites	12
Number of Orbits	6
Height of Satellites	1000 km.
Rotation Time	113 min.
Source ϕ	15°
Source θ	15°
Destination ϕ	45°
Destination θ	30°

The following sequence of events follow in the transfer of packets:

1. The position of all the satellites is initialized.
2. It is determined that the source terminal is under the coverage area of the satellite located at $(15^\circ, 15^\circ)$ and the destination terminal is under the coverage area of satellite located at $(45^\circ, 15^\circ)$. Since each satellite is supposed to have intersatellite links with the satellite just next to in the adjacent orbit, now there are only two satellites involved in routing the packets. Satellite1 located at $(15^\circ, 15^\circ)$ and Satellite2 located at $(45^\circ, 15^\circ)$. Neglecting the queuing delay for the sake of simplicity, the Total delay encountered by the packets is given

by Equation 2.22.

$$T_{end_to_end} = T_{uplink} + T_{cross_1} + T_{downlink} \quad (2.22)$$

where :

T_{uplink} is the propagation delay between $(15^\circ, 15^\circ)$ for the source terminal at earth surface and $(15^\circ, 15^\circ)$ for the connecting satellite. It is equal to 3.33 ms.

T_{cross_1} is between the two satellites located at $(15^\circ, 15^\circ)$ and $(45^\circ, 15^\circ)$. It is equal to 12.28ms.

$T_{downlink}$ is the propagation delay between the satellite at $(45^\circ, 15^\circ)$ and $(45^\circ, 30^\circ)$ for the destination terminal is equal to 6.8ms.

Therefore $T_{end_to_end} = 3.33 + 12.28 + 6.8 = 22.41$ ms.

3. Now since the satellites are constantly moving after 4.71 minutes the position of the Satellite1 one would be $(15^\circ, 30^\circ)$. This is the point at which the handoff of the connection at the source terminal will take place. Now the connection for the source terminal will be transferred to the coverage of the satellite located at $(15^\circ, 0^\circ)$. Due to the limited intersatellite linkage, now the packets have to be routed with the help of 3 satellites, located at $(15^\circ, 0^\circ)$, $(45^\circ, 0^\circ)$ and $(45^\circ, 30^\circ)$. Now the Total delay encountered by the packets is given by Equation 2.23.

$$T_{end_to_end} = T_{uplink} + T_{cross_1} + T_{cross_2} + T_{downlink} \quad (2.23)$$

T_{uplink} is the propagation delay between source terminal at $(15^\circ, 15^\circ)$ and connecting satellite at $(15^\circ, 0^\circ)$. It is equal to 6.82 ms.

T_{cross_1} is between the two satellites located at $(15^\circ, 0^\circ)$ and $(45^\circ, 0^\circ)$. It is equal to 12.71 ms.

T_{cross_2} is between the two satellites located at $(45^\circ, 0^\circ)$ and $(45^\circ, 30^\circ)$. It is equal to 12.71 ms.

$T_{downlink}$ is the propagation delay between destination terminal at $(45^\circ, 30^\circ)$ and the connecting satellite at $(45^\circ, 30^\circ)$ and is equal to 3.33ms.

Therefore $T_{end_to_end} = 6.82 + 12.71 + 12.71 + 3.33 = 35.57\text{ms}$

In this described manner the calculation of various parameters of the $T_{end_to_end}$ is done according to the positions of various satellites in the constellation and the location of source and destination user terminals.

2.5 Simplifying Assumptions and Their Implications

2.5.1 Effect of Earth's Rotation

Low earth orbit satellites revolve around the earth making a complete revolution in about 113minutes. Since the Earth is also rotating and it takes Earth 24 hours to complete one complete rotation, the effect of this rotation can be safely neglected for most of the connections. In the worst case scenario, the effect of Earth's rotation is that another handoff of connection will take place between the satellites in the adjacent orbits. This will lead to an additional variation in RTT encountered by the packets. For the experiments conducted as a part of this thesis work, the effect of Earth's rotation is ignored.

3. EFFECT OF ROUND TRIP TIME VARIATION IN A LEO SATELLITE NETWORK ON THE PERFORMANCE OF TCP

This chapter describes all the experimental work that has been conducted to study the variations in RTT TCP packets encounter in a LEO satellite network. This chapter outlines the possible problems that current implementations of TCP might encounter, due to the delay characteristics that are particular to a LEO satellite network.

3.1 Experimental Setup

Figure 3.1 describes the experimental setup that was used to conduct the experiments. ONE runs on a Sun workstation (*thrawn*) running the Solaris Operating system. *thrawn* has two Ethernet interfaces: *le0* and *le1*. The Ethernet interface *le0* is connected to another Sun workstation (*jarok*) running the Solaris Operating system. The Ethernet interface *le1* is connected to an Intel 486 machine (*mork*) running NetBSD1.1. ONE running on *thrawn* passes the data segments between two physical networks connected to *le0* and *le1*. The data segments are subjected to a time delay based on the calculations described in Section 2.3.

All the data for the experiments was collected using the *tcpdump*¹ packet capture utility software. Time Sequence and RTT graphs were generated and analyzed with the *tcptrace*² utility.

¹Available at <http://www-nrg.ee.lbl.gov>

²Available at <http://jarok.cs.ohiou.edu>

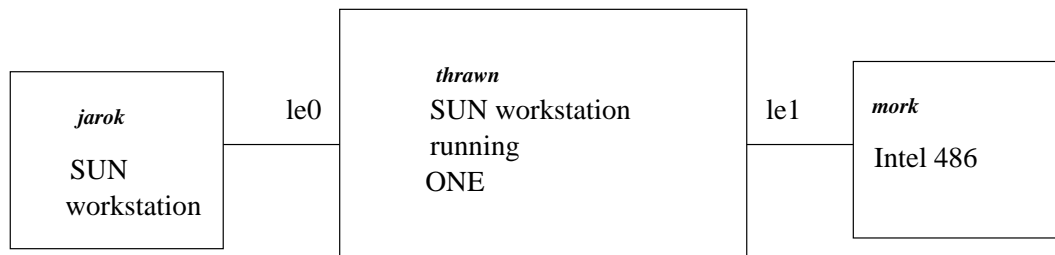


Figure 3.1 Experimental Setup

This figure shows the experimental setup where *thrown* is the machine running ONE. The transfer of data segments between *jarok* and *mork* takes place using ONE running on *thrown*.

3.2 RTT Variations due to Changes in Propagation Delays

In this section, the effect of the relative movement between the satellites and source and destination terminals is explored. Due to the relative movement of the satellites, the propagation delay the packets encounter while traveling from source to destination user terminals, using LEO satellites as forwarding nodes, is constantly varying. The effect is more evident when a handoff for a particular connection takes place. In the case of a handoff, a completely new set of satellites is used for routing the packets, and the distances of these satellites from the source and destination user terminals may be completely different compared to the previous set of satellites. This will cause a sudden, significant change in propagation delay, which in turn causes a wide variation in RTT. In this section an analysis has been made by setting up various configurable parameters, such as source and destination location. At the end of this section, a summary of the effects of such a variation on the working of TCP is attached.

3.2.1 Experiment Number 1

Table 3.1 shows the configuration parameters of a LEO satellite network. There are a total of 72 satellite with 12 satellites in each of 6 orbits. In this experiment, a data segment of 512 bytes is sent after every 1 second interval. These data segments experience the delay they would have encountered in a LEO satellite network with the design parameters described in Table 3.1.

Table 3.1 Configuration Parameters of a LEO Satellite Network for Experiment Number 1 to Show Variations in RTT Due to Change in Propagation Delay.: This table gives the configuration parameters of a LEO satellite network. As seen in the table the data transfer is taking place between terminals located at locations $(15^\circ, 1^\circ)$ and $(15^\circ, 73^\circ)$.

Number of Satellites	12
Number of Orbits	6
Uplink Bandwidth	2.048 MBps
Downlink Bandwidth	2.048 MBps
Inter-satellite Bandwidth	2.048 MBps
Source ϕ	15°
Source θ	1°
Destination ϕ	15°
Destination θ	73°

As shown in Figure 3.2, there is an abrupt variation in the RTT at certain sections of the graph. This sudden variation is due to the fact that whenever there is a handoff of connection, either at the source or destination user terminals, there is a completely new set of satellites handling the connection. Therefore, there is a change in the distance between the satellites and user terminal and, hence, there is a sudden change in the propagation delay. This change in propagation delay causes the variation in the RTT.

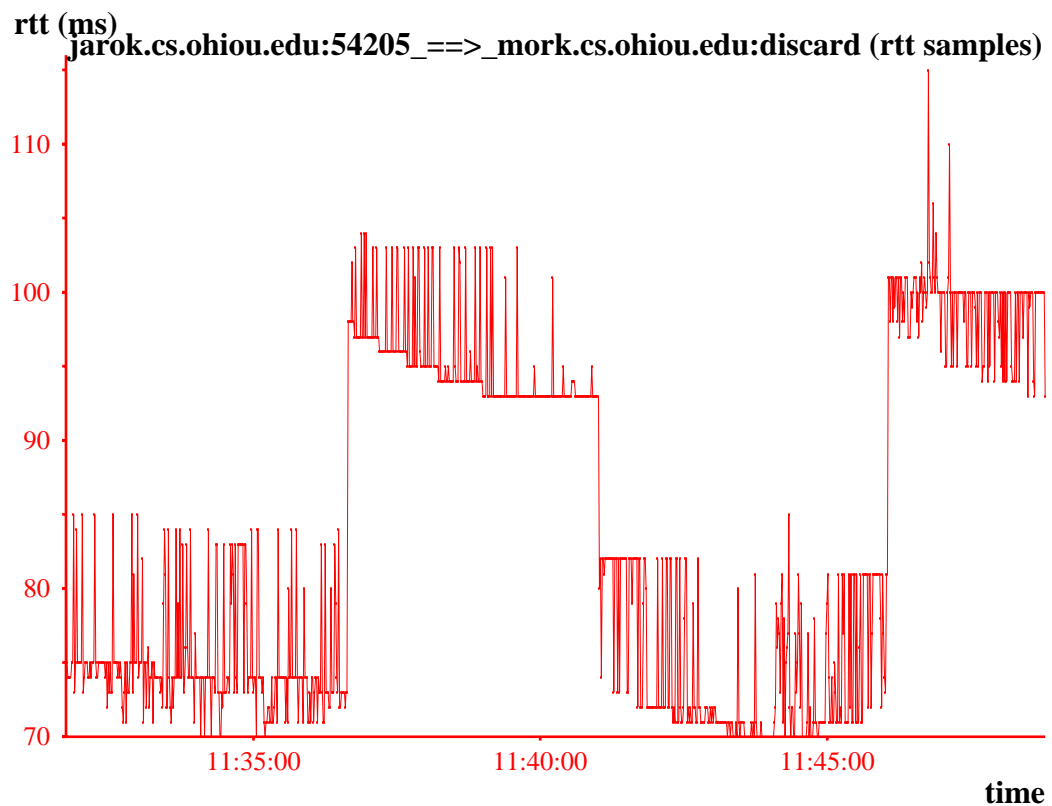


Figure 3.2 RTT vs Time Graph to Show Variations Due to Changes in Propagation Delay between Locations $(15^{\circ}, 1^{\circ})$ and $(15^{\circ}, 73^{\circ})$

This Figure gives the RTT vs time axis graph for the data transfer between *jarok* and *mork*. This data transfer is taking place for source location of $(15^{\circ}, 1^{\circ})$ and the destination location of $(15^{\circ}, 73^{\circ})$.

As seen in Figure 3.3, there is no retransmission of packets during the whole session of data transfer. There is a sudden increase in RTT by 40ms, but this does not trigger the retransmission. The reason for this behavior is that RTT is measured in multiples of 500ms. Any sample of RTT which is below 500ms is counted as 1 tick³. Therefore, all the values of RTT are being measured in denominations of 500ms, which prevents any retransmission for values of RTT below 500 ms unless there a packet loss. The details of RTO measurement and the timer granularity are discussed in Appendix A.

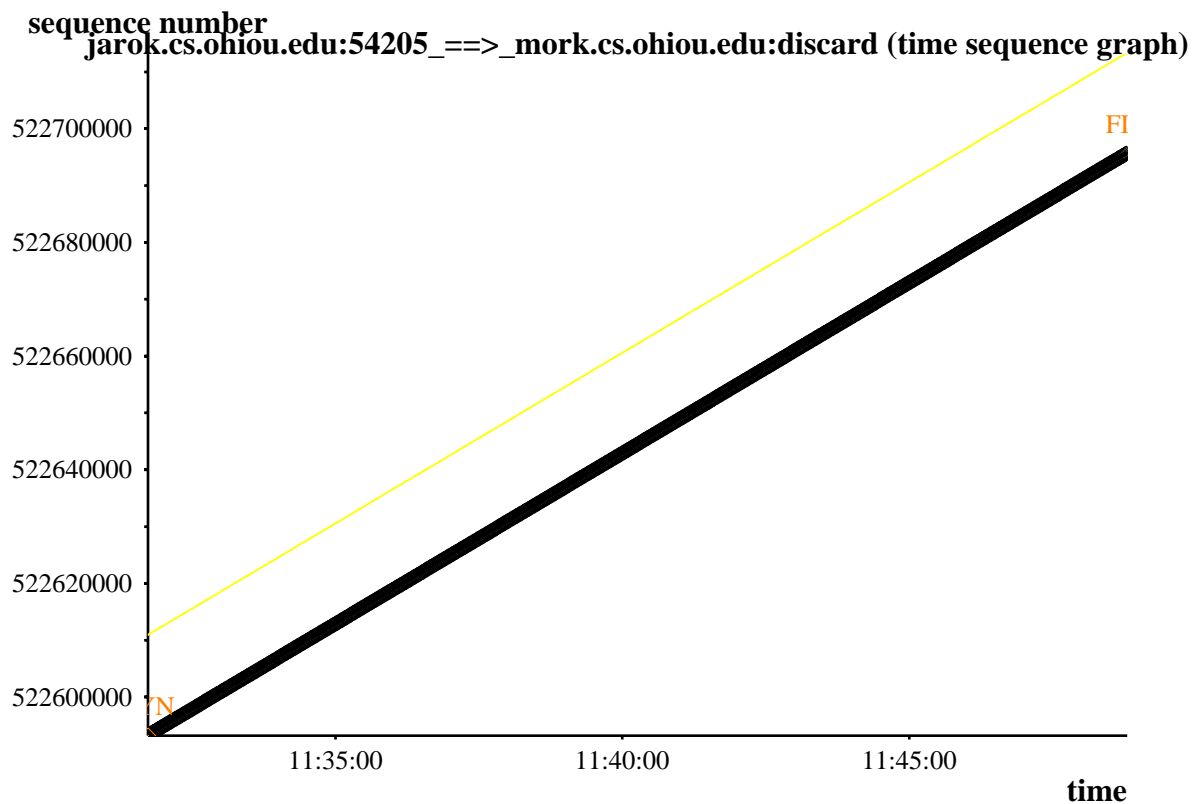


Figure 3.3 Time-Sequence Graph for the Data Transfer between Locations ($15^\circ, 1^\circ$) and ($15^\circ, 73^\circ$)

This figure describes the Time Sequence Graph for the data transfer between *jarok* and *mork*. As clear for the time-sequence graph there is no retransmission of segments.

³1tick = 500ms.

3.2.2 Experiment Number 2

Table 3.2 shows the configuration parameters of another LEO satellite network. There are a total of 72 satellites with 12 satellites in each orbits. In this experiment, 4000 bytes of data are sent every 1 sec. This results in transmission of 3 segments between *vger*⁴ and *mork* every second. The packets experience the same delay as they would have encountered in a LEO satellite network.

Table 3.2 Configuration Parameters of a LEO Satellite Network of a LEO Satellite Network for Experiment Number 2 to Show Variations in RTT Due to Change in Propagation Delay.:

This table gives the configuration parameters of a LEO satellite network. As seen in the table the data transfer is taking place between terminals located at locations ($15^\circ, 20^\circ$) and ($135^\circ, 44^\circ$).

Number of Satellites	12
Number of Orbits	6
Uplink Bandwidth	2.048 MBps
Downlink Bandwidth	2.048 MBps
Inter-satellite Bandwidth	2.048 MBps
Source ϕ	15°
Source θ	20°
Destination ϕ	135°
Destination θ	44°

Figure 3.4, gives the RTT variation for the data transfer between *vger* and *mork*.

As seen from Figure 3.4 there is a sudden increase in RTT, in certain sections

⁴*vger* is a machine connected in the same segment ad *jarok*. *vger* is running SunOS 5.7 as the operating system.

of the graph . This is due the connection handoff at the source terminal. There is a certain periodic variation in RTT observed in the graph⁵.

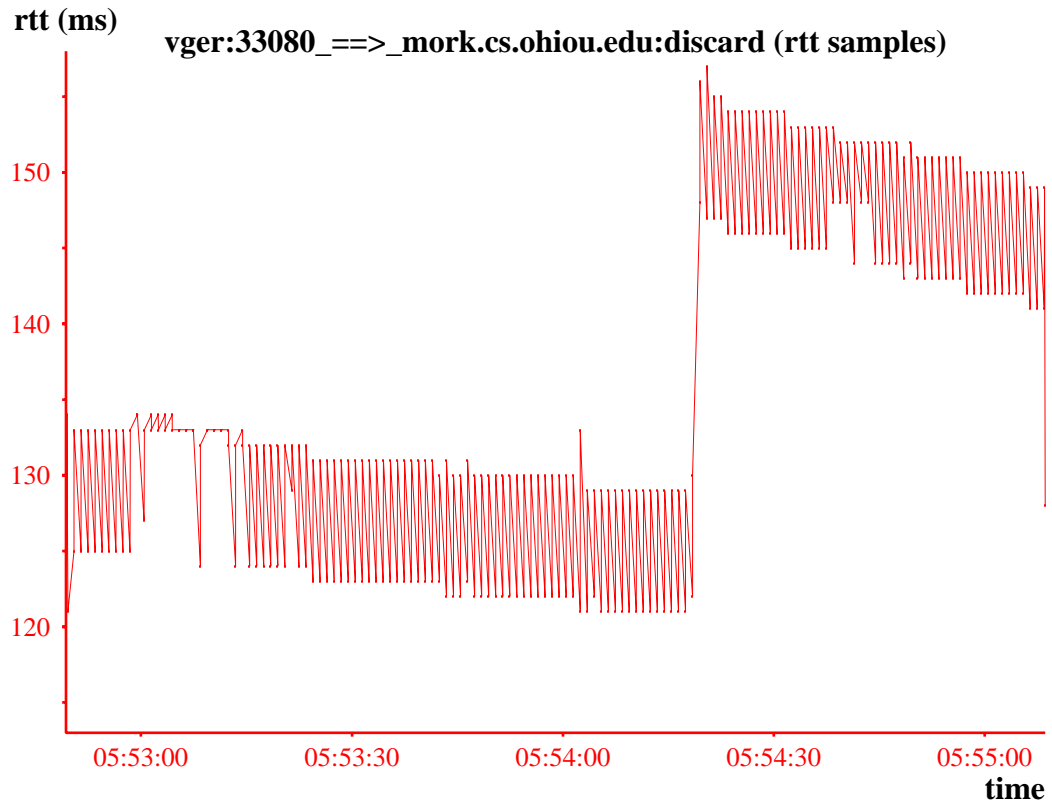


Figure 3.4 RTT vs Time Graph to Show Variations Due to Changes in Propagation Delay between Locations $(15^{\circ}, 20^{\circ})$ and $(135^{\circ}, 44^{\circ})$

This figure gives the RTT vs time graph for the data transfer between *vger* and *mork*. This transfer of data is taking place for the source location $(15^{\circ}, 20^{\circ})$ and destination location $(135^{\circ}, 44^{\circ})$. Sudden change in RTT in certain section of graphs indicates a handoff of connection between satellites.

⁵The explanation for this variation is given in Appendix D

Figure 3.5 gives the graph of measured RTO (for the various samples of RTT) with respect to the time axis.

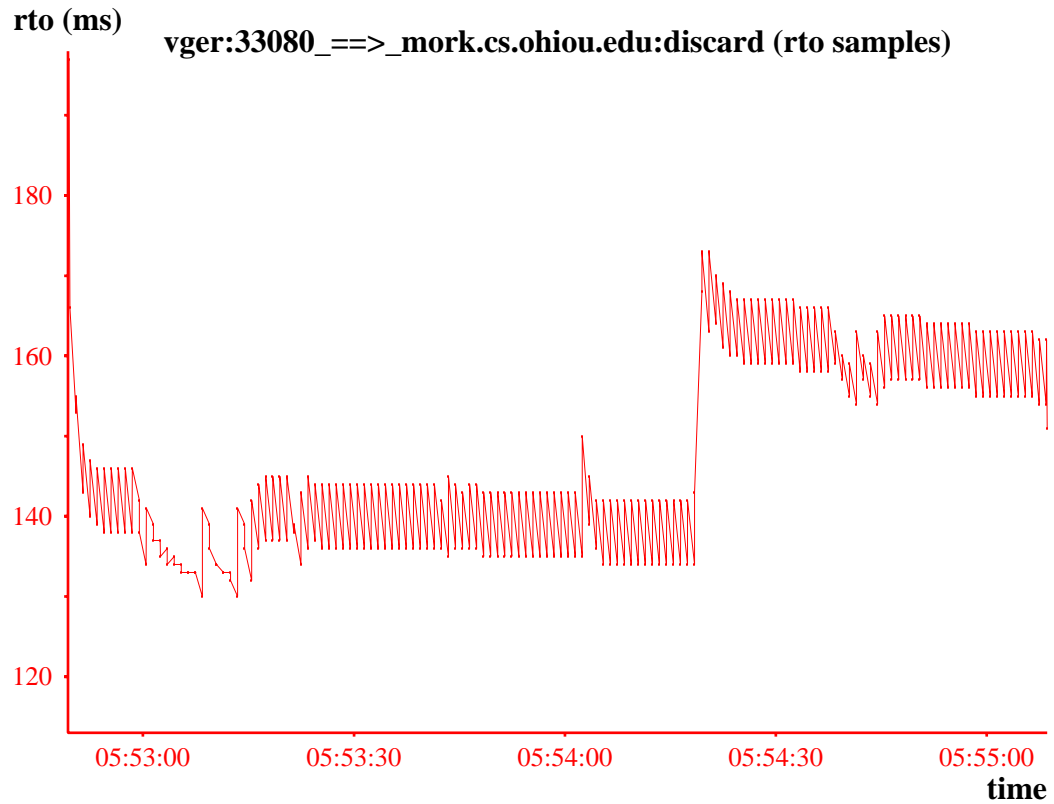


Figure 3.5 RTO vs Time Graph to Show Calculated RTO based on the Samples of RTT for Data Transfer between Locations $(15^\circ, 20^\circ)$ and $(135^\circ, 44^\circ)$

This figure gives the RTO vs Time graph for the data transfer between *vger* and *mork*. The calculation of RTO is based on Equation 1.3.

Figure 3.6 gives the graph of $(RTO - RTT)$ vs the time axis. This graph is particularly important as for all the points with negative value of $(RTO - RTT)$ indicate the instances where the value of RTT of the transmitted packets was less than the calculated value of RTO. The calculations of RTO is based on Equation 1.3. The actual implementation of RTO is dependent on the operating system and is described in Appendix A.

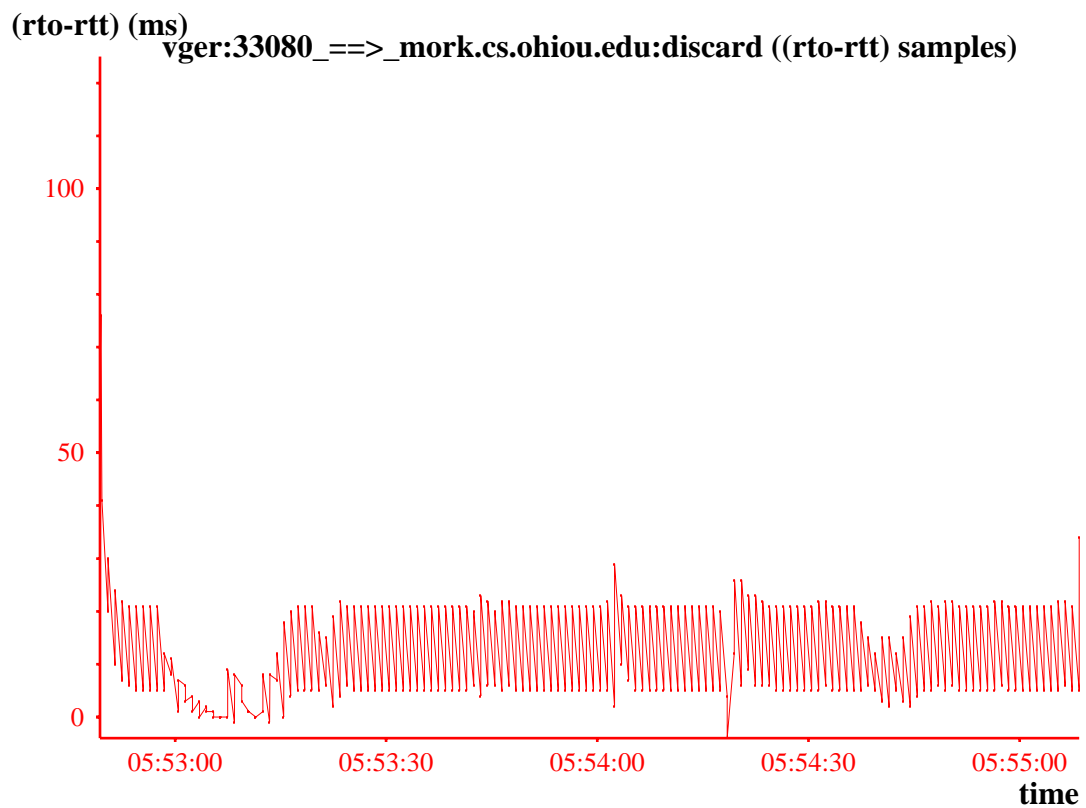


Figure 3.6 (RTO-RTT) vs Time Graph

This figure gives the $(RTO - RTT)$ vs Time graph for the data transfer between *vger* and *mork*. The calculations of RTO is based in the equations described in Section 1.2.1.

It is observed, after making RTO calculations based on the RTT samples, that the value of RTO before the handoff was 140.72ms while the RTT for the subsequent packet was 148ms. Since the new value of RTT is greater than the calculated value of RTO, it should have resulted in the retransmission of packets. As seen in Figure 3.7, there is no retransmission of packets for the entire session of data transfer. As discussed in Appendix A, the exact calculation of RTO and RTT depends on timer granularity of the implementation and thus prevents retransmission of segments.

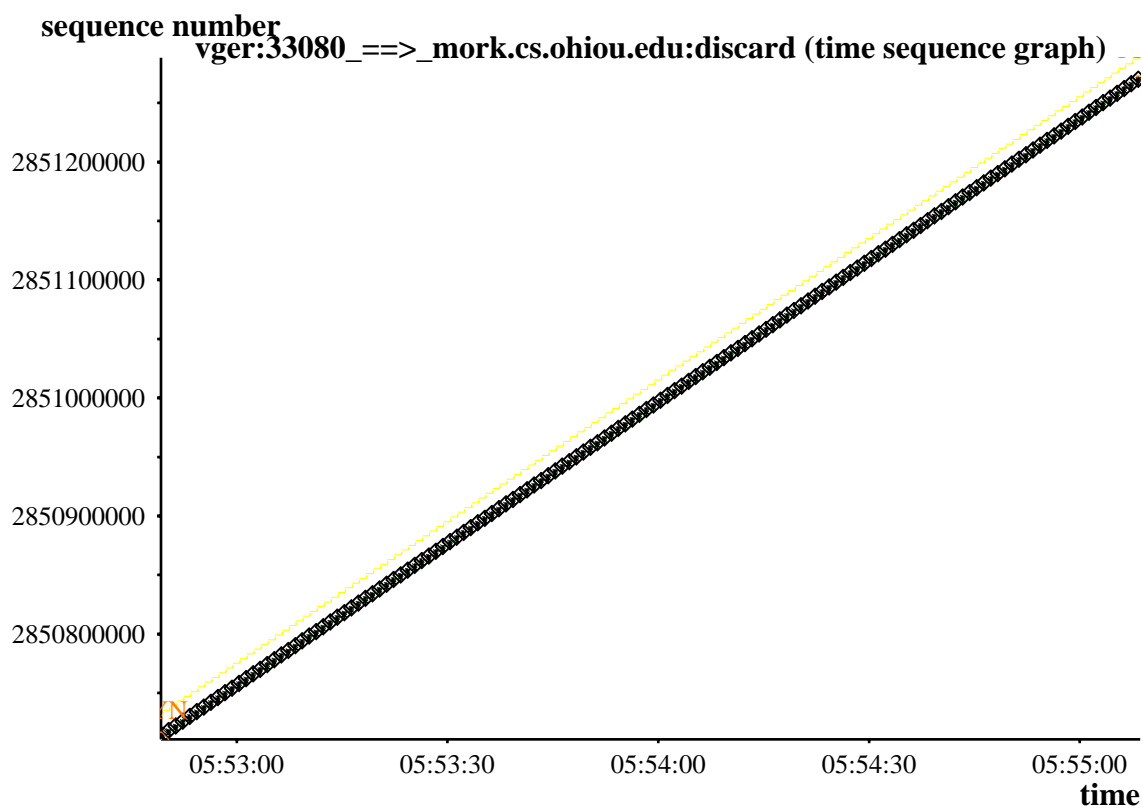


Figure 3.7 Time-Sequence Graph for Data Transfer between Locations ($15^\circ, 20^\circ$) and ($135^\circ, 44^\circ$)

This figure describes the Time Sequence Graph for the data transfer between *vger* and *mork*. It is clear from the graph that there is no retransmission of packets.

3.2.3 Summary

As seen in the experiments, the handover of connection between the satellites can result in a sudden variation in RTT encountered by the TCP packets involved in the data transfer. After conducting many experiments, it has been observed that there can be an increase of up to 50ms in RTT due to the change in the propagation delay. As discussed in Appendix A, for most of the implementations of TCP the timer granularity for the RTT is 500ms. Since for any combination of source and destination user terminal location the maximum RTT caused by the effect of the propagation delay is in the order of 100ms, a variation in RTT of about 50ms would not cause retransmission. There are certain implementations of TCP which use more accurate RTT calculations, a variation of 50ms can trigger a retransmission.

3.3 Traffic Model for a LEO Satellite Network

In the Section 3.2, the only traffic that was loading the network was the packets flowing between the source and destination user terminals. The idea was to study the effect of only the geometric configurations of LEO satellites on the variation of end-to-end delay. In practice, there are thousands of other simultaneous connections between various user terminals around the globe. The effect of which is that the satellites involved in routing the packets from the source to the destination user terminals under consideration are also involved in handling the traffic from numerous other sources. This means that the packets from a number of other user terminals arrive at the satellite and are queued in the memory buffer before being forwarded to other satellites. This traffic from numerous other sources effects the queuing delay that the packets traveling between the source and destination user terminals will encounter at each of the satellites involved in routing the packets. The network traffic, effecting the queuing delay from numerous other sources, will change the end-to-end packet delay and RTT for the TCP packets for the connection under consideration. The goal

of this section is to model such network traffic for a LEO satellite network and study its effects on RTT variations.

One of the unique features of network traffic in a LEO satellite network is geographical traffic non-uniformity. Since the network traffic generated by user terminals is a function of the density of user terminals trying to communicate via the LEO satellite network, the effect is that the packets between the source and destination terminals experience large variations in queuing delay when the connection is transferred between the satellites. When the connection is transferred to a new satellite, there is also a change in the routing path, and the packets now are forwarded using a new set of satellites. The new satellites handling the packets might be involved in handling a far larger or smaller amount of traffic compared to the previous set of satellites which means the queuing delay in each of these satellites may be far larger or smaller compared to the previous set of satellites. This change in the number of the user population effects the total end-to-end delay the packets encounter between source and destination user terminals. The effect of such variation is explored in the rest of the chapter.

3.3.1 Traffic Simulation for LEO Satellite Systems

In order to accommodate the effect of the amount of traffic that is loading the network apart from the source and destination user terminals, the queue length of each of the satellites participating in routing decision is varied in direct proportion to the number of users that particular satellite is covering.

The number of users under the coverage of a particular satellite is a continuous function of the location of the satellite. Also, in a particular coverage area the user population is distributed all over the geographical region. In our analysis, each user in a particular coverage area contributes one packet to the satellite queuing buffer, where each packet is 1024 bytes. If there are n users under the

coverage of a particular satellite, then the satellite queue length will be equal to $(n * 1024) / (InterSatelliteBandwidth)$ bytes.

3.4 Problem with Retransmission Time Out Algorithm

One of the possible scenarios in a LEO satellite communication system is that when a handoff of the connection takes place, then the new set of satellites that are forwarding the packets are now handling a larger number of user population compared to the previous set of satellites. There is an abrupt increase in queuing delay now encountered by the packets that are transmitted after the handoff.

Since TCP maintains a current estimate of RTT and it uses this estimate of RTT to calculate RTO, the current estimate of RTO, which is based on the RTT delay of the packets before the handoff will be lower than the new value of RTT, which the packets transmitted after the handoff will encounter. This will result in the expiration of the RTO timer and a retransmission of all the unacknowledged packets transmitted immediately after the handoff. Also, the TCP connection will enter a congestion control phase as TCP takes expiration of the RTO timer as an indication of network congestion. Therefore, the effect of this sudden increase in end-to-end packet delay after the handoff is that the packets which were already transmitted and are not lost are again retransmitted due to expiration of RTO interval. This results in a waste of network capacity. This also results in the decrease in throughput.

3.4.1 Experiment Number 1

This experiment studies the effect of increase in the number of users, using the LEO satellite network described in Table 3.3. This experiment is conducted by transferring a 5MB file between *jarok* and *mork*, machines which act as the end points to the machine running ONE. The packets experience a delay variation they would have

encountered in a LEO satellite network. There is an increase in user population from 200 to 6000 after the handoff.

Table 3.3 Configuration Parameters of a LEO Satellite Network for Experiment Number 1 to Show the Effects of Increase in User Population.:

This table gives the configuration parameters of a LEO satellite network. As seen in the table there is an increase in user population from 200 to 6000.

Number of Satellites	12
Number of Orbits	6
Uplink Bandwidth	2.048 MBps
Downlink Bandwidth	2.048 MBps
Inter-satellite Bandwidth	155.52 MBps
Source ϕ	15°
Source θ	353°
Destination ϕ	15°
Destination θ	73°
Initial Users	200
Final Users	6000

As seen in Figure 3.8, there is a sudden increase in the RTT of the packets, because of the fact the queuing delay has increased after the handoff due to the increase in the number of users being handled by the satellites involved in routing the packets.

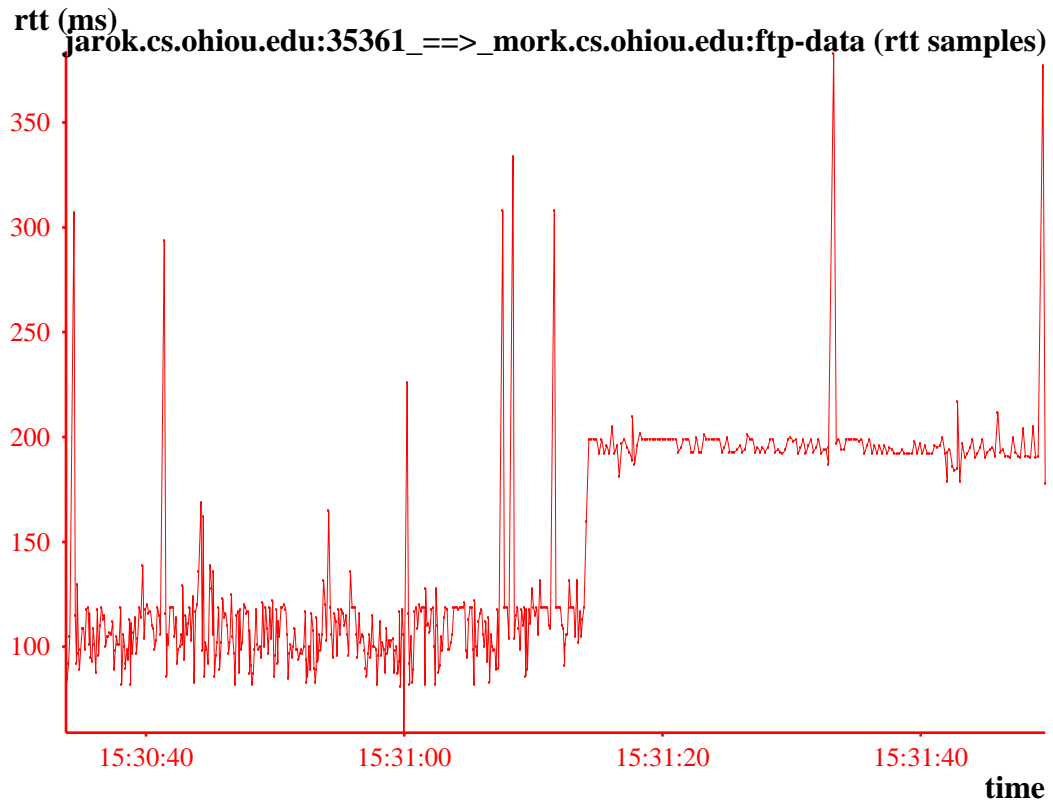


Figure 3.8 RTT vs Time Graph to Show the Increase in RTT due to Increase in User Population from 200 to 6000

This figure gives the RTT vs time graph for the data transfer between *jarok* and *mork*. This data transfer is taking for the source location ($15^{\circ}, 353^{\circ}$) and destination location ($73^{\circ}, 15^{\circ}$). As seen in the figure there is abrupt changes in RTT due the effect of handoff and change in propagation delay.

Figure 3.9 gives the calculated values of RTO, based on the sample values of RTT. The calculations are based on the equations mentioned in Section 1.2.1.

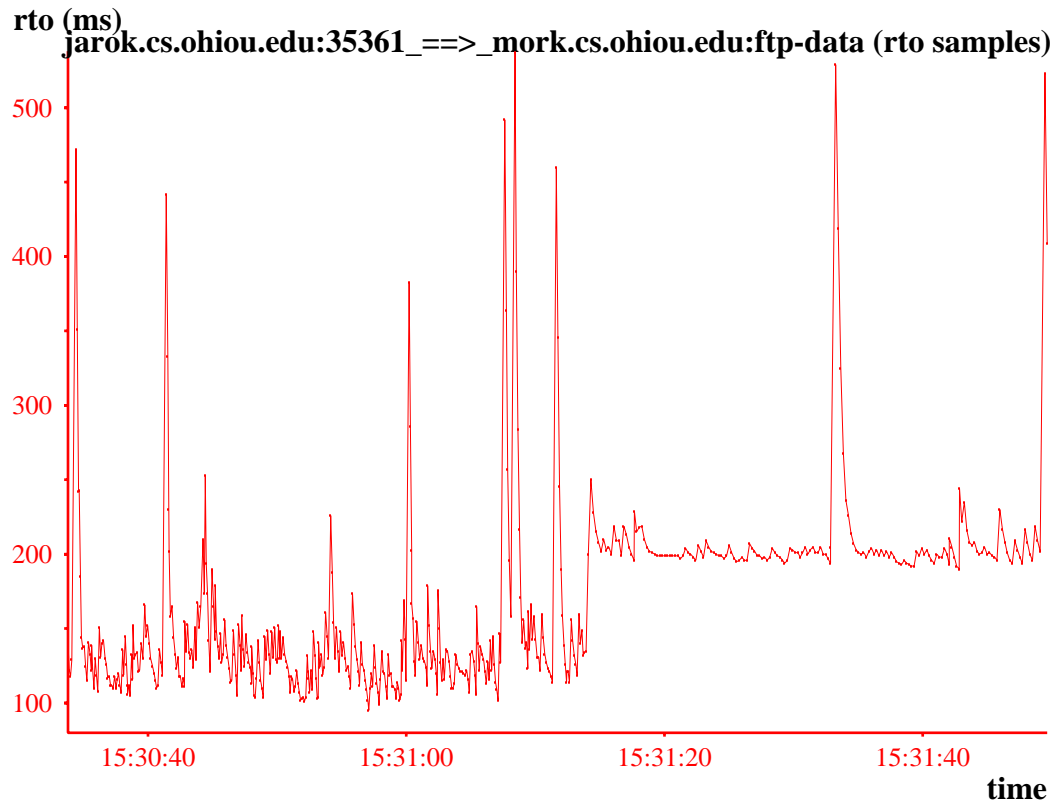


Figure 3.9 RTO vs Time Graph to Show Calculated RTO based on the Samples of RTT

This figure gives you the graph of the various values of calculated RTO vs Time. These values of RTO are calculated based on RTT sample. Equation 1.3 are used for calculations.

Figure 3.10 gives the graph of $(RTO - RTT)$ vs the time axis. As clear from the graph that there are lot of places in the data transfer where the points of graph show negative value of $(RTO - RTT)$. Which means that at these points during the data transfer there could have been a retransmission of segments.

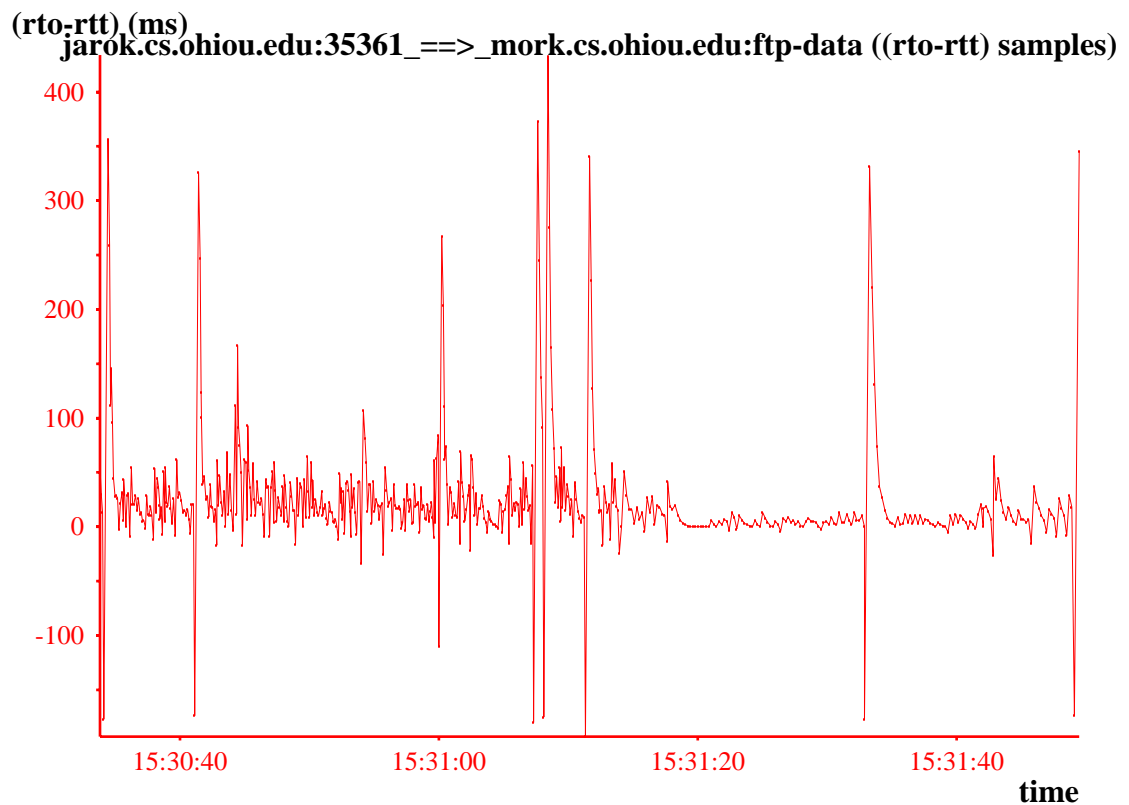


Figure 3.10 $(RTO - RTT)$ vs Time Graph for the Calculated Values of RTO
 This figure gives you the graph of the various values of calculated $(RTO - RTT)$ vs Time. These values of RTO are calculated based on RTT sample. Equation 1.3 are used for calculations. All the points for $y < 0$ gives the points in the data transfer where calculated value of RTO was less than the RTT of the transmitted packets.

Figure 3.11 gives the time-sequence graph for the entire session of data transfer. As seen from the graph there is no instance of packet retransmission. After making the calculations of RTO based on the sample values of RTT, it is observed that the value of RTO is 135.11ms before the handoff, while the value of RTT for the next packet is 160ms. This should have resulted in retransmission. As discussed in Appendix A, there is no retransmission of packets for RTT of less than 500ms unless there is a packet loss. Also, as seen in the graph there is an change in slope after the *handoff* of the connection. This change of slope is due to the increase in RTT encountered by TCP packets. As the RTT encountered by the TCP has increased, this has resulted in the decrease in throughput and there is an decrease in the slope of the graph, which represents the throughput of the connection.

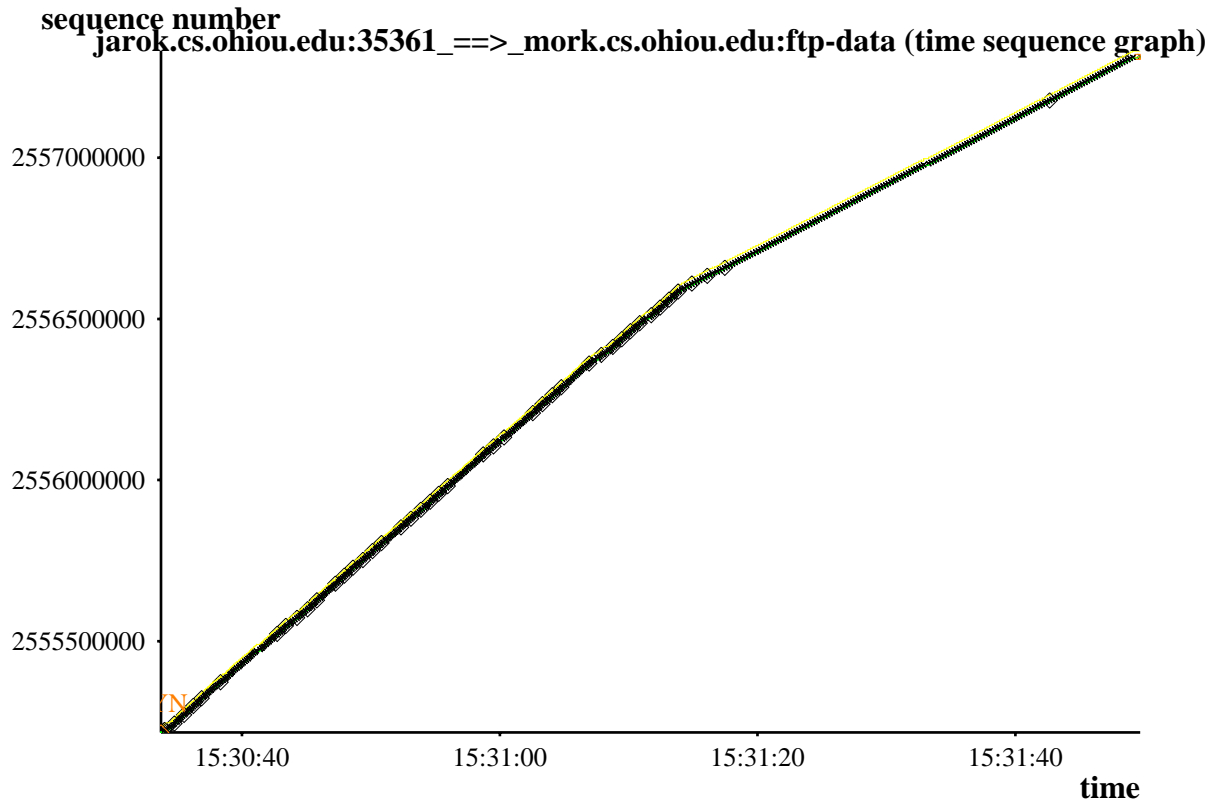


Figure 3.11 Time-Sequence graph to Show the Effect of Increase of User Population from 200 to 6000 on the Packet Loss

This figure gives you the Time-Sequence graph for the data transfer between *jarok* and *mork*.

3.4.2 Experiment Number 2

This experiment studies the effect of an increase in the number of users using the LEO satellite network described in Table 3.4. This experiment involves the transfer of a 5MB file between *jarok* and *mork*, machines which act as the end points to the machine running ONE. The packet experience a delay variation they would have encountered in a LEO satellite network. The purpose of this experiment is the study the effect of an increase in RTT beyond 500ms. For this purpose, the user population is considered to be high, as the initial user population is 20000 and the final user population is 60000.

Table 3.4 Configuration Parameters for Experiment Number 2 to Show the Effects of Increase in User Population.:

This table gives the configuration parameters of a LEO satellite network. As seen in the table there is an increase in user population from 20000 to 60000.

Number of Satellites	12
Number of Orbits	6
Uplink Bandwidth	2.048 MBps
Downlink Bandwidth	2.048 MBps
Inter-satellite Bandwidth	155.52 MBps
Source ϕ	15°
Source θ	353°
Destination ϕ	15°
Destination θ	73°
Initial Users	20000
Final Users	60000

Figure 3.12 gives the RTT vs time graph for the data transfer. As seen in Figure 3.12, there is an abrupt increase in the value of RTT. Increase in RTT is 480 ms. The RTT increases from 390ms to 870ms.

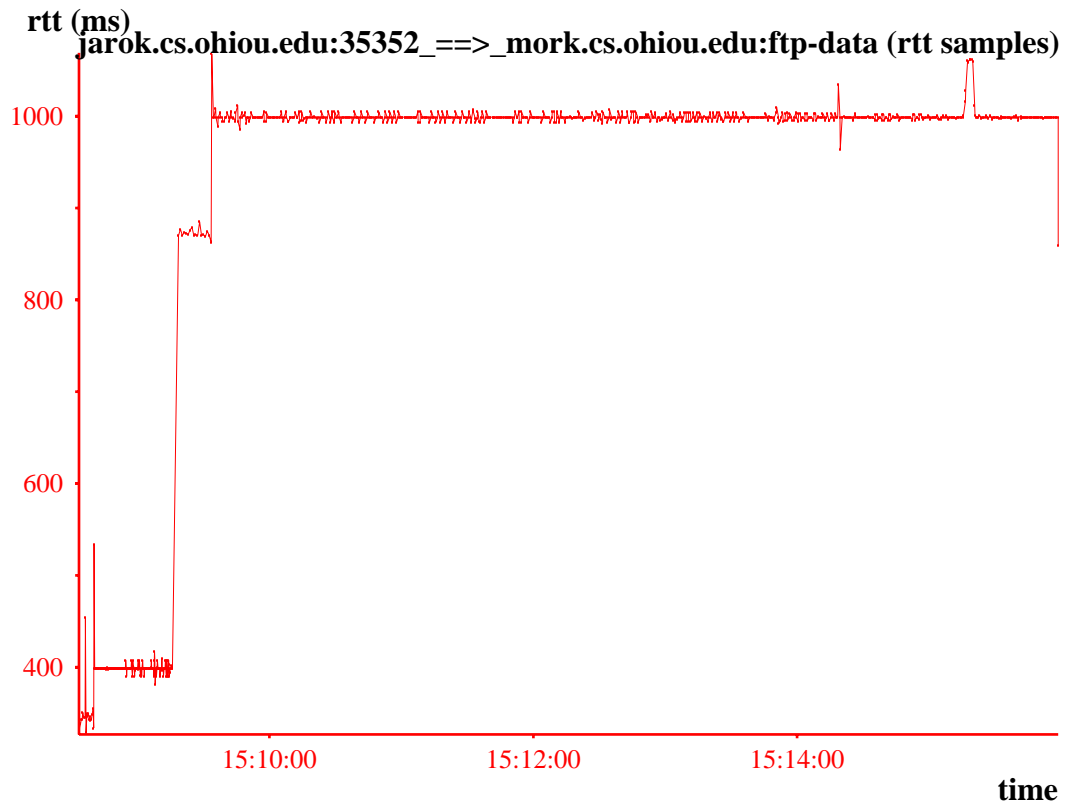


Figure 3.12 RTT vs Time Graph to Show the Increase in RTT due to Increase in User Population from 20000 to 60000

This figure gives the RTT vs Time Graph for the 5MB file transfer between *jarok* and *mork*.

Figure 3.13 gives the Time-Sequence graph for the experiment, which shows the retransmission of packets at the time of the handoff. This is due to the increase in the RTT. It is verified from the trace files containing the raw information of the data transfer that the retransmission is due to the expiration of RTO timer and not due to the loss in packets.

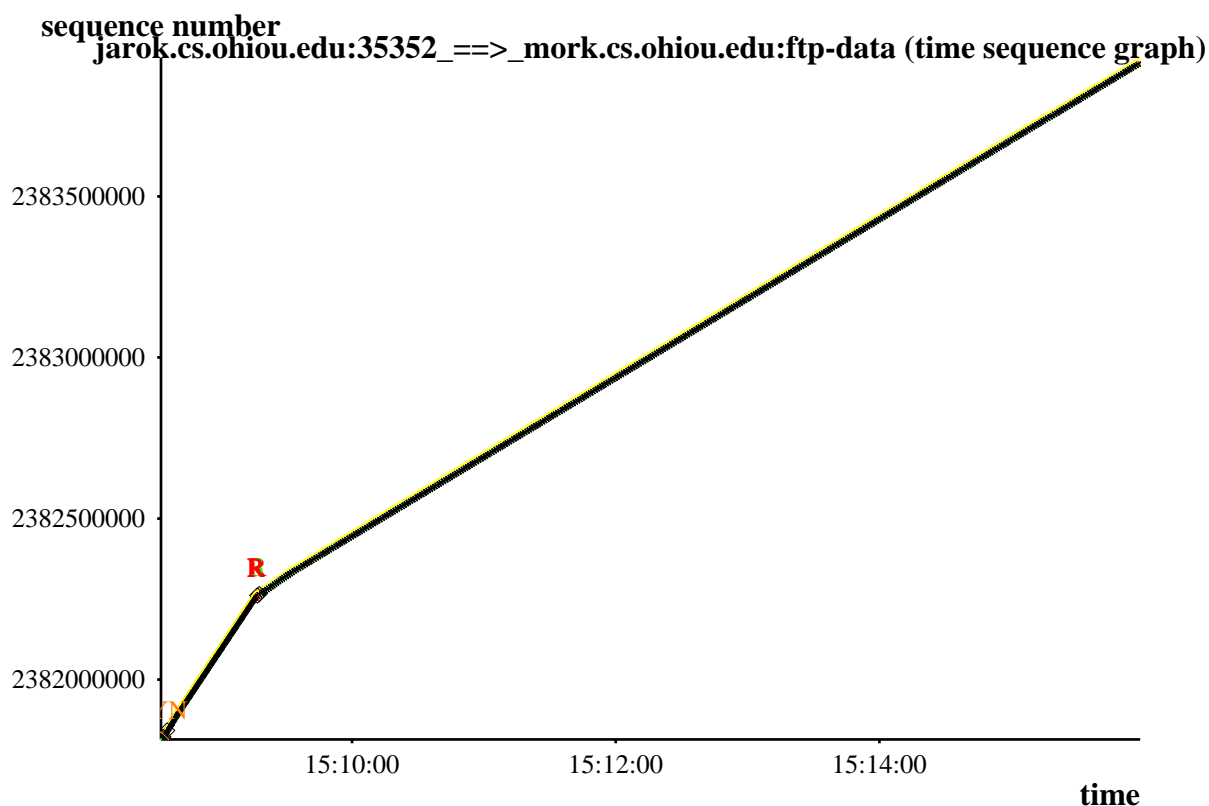


Figure 3.13 Time-Sequence Graph to Show the Effect of Increase of User Population from 20000 to 60000 on the Packet Loss. :

This is the time sequence graph for the data transfer between *jarok* and *mork*.

3.5 Problem with Idle Connections

After a TCP connection has been idle for an extended period of time it can potentially allow a burst of traffic to be transmitted into the network after it restarts sending the data. This can result in a large amount of data loss[VH97][APS99]. This creates a problem of excessive data loss in a scenario when there is an increase in the RTT of the connection after the handoff. The following experiment has been conducted to show the nature of the problem.

3.5.1 Experiment No. 1

The configuration parameters used for the LEO satellite network are shown in Table 3.5. In this experiment, 4000 bytes of data are written after every 1 second. After sending data for a period of 100 seconds, the connection is set on idle and it again starts sending data after a period of 1 min. Meanwhile, in this idle time of 1 min., the handoff of connection between the satellites take place. This results in an increase in the number of users from 20000 to 60000. As an effect of the increase in the number of users, the queuing delay is increased.

Table 3.5 Configuration Parameters of a LEO satellite network for the Experiment Showing the Effect of Restart of Idle Connection.:

This table gives the configuration parameters of a LEO satellite network.

Number of Satellites	12
Number of Orbits	6
Uplink Bandwidth	2.048 MBps
Downlink Bandwidth	2.048 MBps
Inter-satellite Bandwidth	155.52 MBps
Source ϕ	15°
Source θ	22°
Destination ϕ	45°
Destination θ	42°
Initial Users	20000
Final Users	60000

As seen in Figure 3.14, there is an increase in the RTT after idle time of the connection.

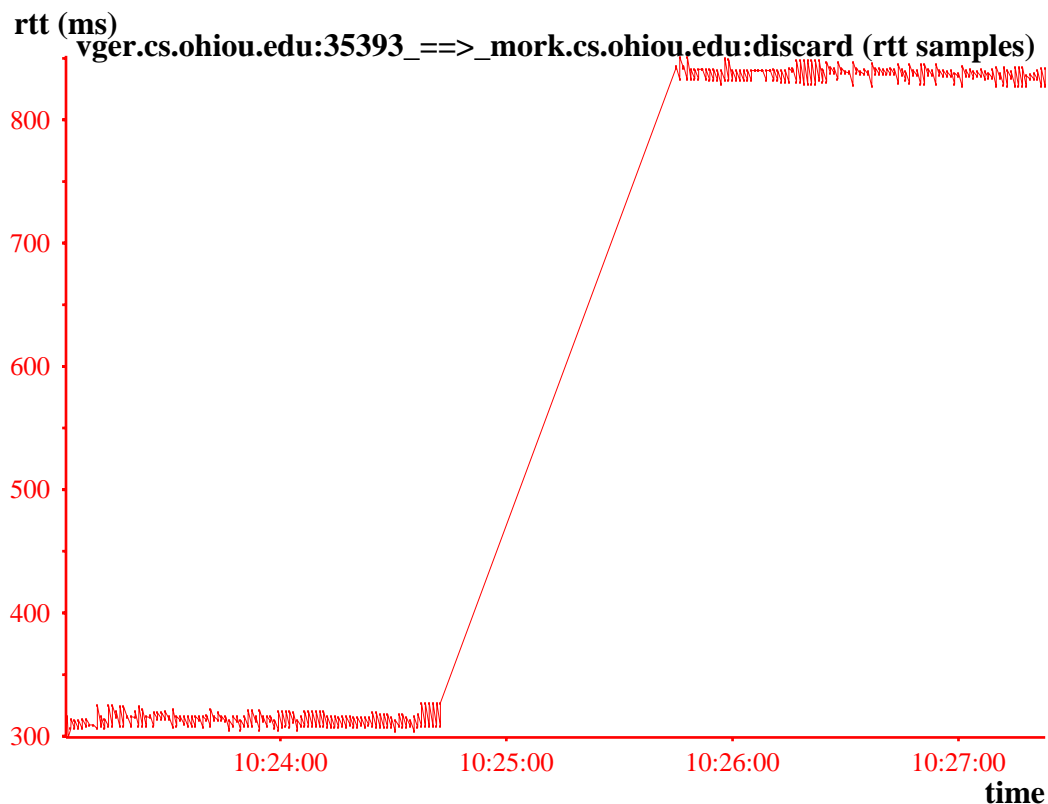


Figure 3.14 RTT vs Time Graph to Show the Increase in RTT after Restart of an Idle Connection

This figure gives the RTT vs Time Graph for the data transfer between *vger* and *mork*.

Figure 3.15 shows the time sequence graph for the data transfer. As seen in the graph, there is a retransmission of segments after the restart of the idle connection. As verified from the raw data file for this session of data transfer, all the packets that were transmitted immediately after the idle time are to be retransmitted, due to an increase in RTT and expiration of RTO timer. It is recommended that if TCP does not receive a segment for more than one retransmission time out interval, then the value of *cwnd* should be reset to 1 segment size and the connection should again start from slow start phase. This will minimize the loss of segments due to the increase in RTT.

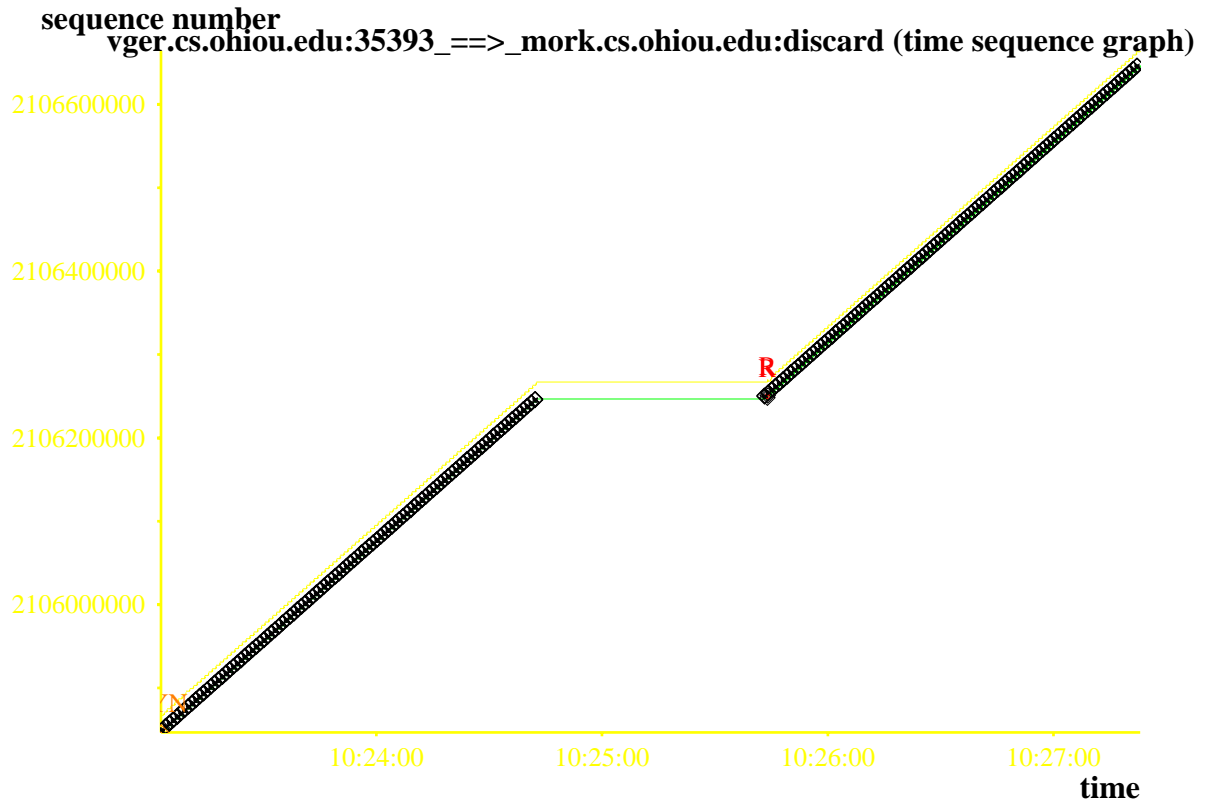


Figure 3.15 Time Sequence Graph to Show the Effect of Restart of an Idle Connection.:

This is the time sequence graph for the data transfer between *vger* and *mork*.

3.6 Problem with Fast Retransmit Algorithm

After the handoff there can be a decrease in the number of users being handled by the new set of satellites forwarding the packets from source to destination user terminals. There is a decrease in queuing delay and a decrease in RTT encountered by the packets transmitted after the handoff, the effect is that the packets transmitted after the handoff will experience a lesser amount of delay as compared to the packets that are queued up with the previous set of satellites handling a larger user population. This can result in a scenario where some of the packets transmitted immediately after the handoff will reach the destination user terminal before the packets that were

transmitted before the handoff. This will result in the generation of duplicate ACKs. If the sender side of TCP connection receives 3 duplicate ACKs, it will retransmit all unacknowledged packets without waiting for the RTO timer to expire and will enter the fast recovery phase. Therefore the effect of the decrease in RTT is that there is again a retransmission of already transmitted packets which again results in a waste of network bandwidth. Also now TCP will start with the congestion avoidance phase after the completion of fast recovery which will result in the decrease of throughput.

4. CONCLUSIONS AND RECOMMENDATIONS

There has been very little previous research to explore the effects of the communication environment of a LEO satellite network on the delay variations that the data segments using these networks encounter. This thesis work has resulted in the modification of the existing code of ONE. Now ONE software can be used to simulate the delay characteristics of a LEO satellite network. Using the simple interface of ONE, the delay characteristics of various configurations of LEO can be analyzed. With the experimental setup mentioned in Section 3.1, a lot of existing implementations of TCP can be tested for their performance on various configurations of LEO satellite networks. After studying the structure of LEO satellite networks, two major factors that can contribute to the variation in end-to-end delay were studied:

Variation in propagation delay - The variation in propagation delay is caused by the changing distances between the satellites and user terminals.

Variation in user population - There can be a large variation in the user population being handled by the set of satellites before and after the *handoff* of connection takes place. This again causes a large variation in end-to-end delay.

After studying these two factors, the effect of these variation on the performance of TCP was analyzed. The results and analysis of this study has been enumerated in Section 4.1.

4.1 Conclusions

The following were the results and inferences of the research conducted as a part of this thesis work:

Variations in RTT - After conducting many experiments to analyze the RTT behavior of a LEO satellite network, it has been observed that during the handoff there can be a sudden variation in RTT. The effect of this variation on the performance of TCP is dependent on the implementation of TCP. For most of the implementations of TCP as discussed in Appendix A, there is a timer granularity of 500ms attached to the calculation of RTT which prevents any retransmission of segments for RTT less than 500ms. If RTT is measured in multiples of 500ms, then most of the variations encountered by the TCP packets in a LEO satellite network would not result in retransmission of packets.

Problems with some congestion control algorithms - Due to the effect of *traffic non-uniformity*, there is sudden change in the network traffic being handled by the satellites involved in routing the packets which could result in the following problems with respect to the existing congestion control algorithms used by TCP :

Problem with Retransmission Time Out - A sudden increase in RTT can lead to the expiration of RTO timer. This can lead to wasteful retransmission of packets.

Problem with idle connection - If a connection has been idle for a long time, the TCP should use slow start with the window size of 1 segment for sending a new set of data or this can result in a loss of a larger number of data segments.

Problem with Fast Retransmit Algorithm - For a connection with a large window size, if the number of data segments in flight is large, the sudden decrease in RTT can lead to the generation of duplicate ACKs. These duplicate ACKs can again result in wasteful retransmission of data.

4.2 Future Work

In order to further extend the research that has been conducted as part of this thesis , following areas can be explored:

Study of different kind of orbital constellation - As part of this thesis, LEO satellite networks with polar circular orbits have been studied. This work can be extended to study the effect of non-polar constellations.

Study the effects of various kinds of routing algorithms - The algorithm that was used to calculate the route form the source to the destination user terminals was based on simple heuristics of calculating the minimum number of hops. In practice, there are many complex routing algorithms that can be implemented, and the effect of these routing algorithms on the end-to-end delay can be studied.

Effect of LEO satellite networks on other implementations of TCP - The operating system SunOS 5.7 was used to study the variations in RTT delay. In its implementation of TCP, RTO did not occur for the values of RTT of less than 500ms. There are many other implementations of TCP which might use some other timer granularity for the calculations of RTT. The effect of the variations in RTT and performance of RTO algorithms for the various other implementations of TCP is definitely one of the most interesting areas to be investigated.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [ACO97] Mark Allman, Adam Caldwell, and Shawn Osterman. ONE: The Ohio Network Emulator. Technical report, Ohio University, 1997.
- [All97] Mark Allman. Improving TCP Performance Over Satellite Channels. Master's thesis, Ohio University, August 1997.
- [APS99] Mark Allman, V. Paxson, and W. Stevens. TCP Congestion Control, 1999. RFC 2001.
- [BLFN96] Tim Berners-Lee, R. Fielding, and H. Nielsen. Hypertext Transfer Protocol - HTTP/1.0, May 1996. RFC 1945.
- [CCJ98] Ramon Caceres, Li Fung Chang, and Ravi Jain. Wireless Internet and Intranet Access, 1998.
- [Com95] Douglas E. Comer. *Internetworking with TCP/IP, Volume I, Principles, Protocols, and Architecture*. Prentice Hall, 3rd edition, 1995.
- [Jai86] Raj Jain. Divergence of Timeout Algorithms for Packet Retransmissions. *Proceedings Fifth Annual International Phoenix Conference on Computers and Communications*, pages 174–179, 1986.
- [Jam98] Abbas Jamalipour. *Low Earth Orbital Satellites for Personal Communication Networks*. Artech House, 1998.
- [jit99] Leo dance the jitterbug, 1999. <http://www.nwc.com/905/905f2side1.html>.
- [Koh98] Daniel M. Kohn. Providing Global Broadband Internet Access Using Low-Earth-Orbit Satellites. Technical report, Teledesic Corporation, 1998.
- [KP87] Phil Karn and Craig Patridge. Improving Round-Trip Time Estimates in Reliable Transport Protocols. *Proceedings of ACM SIGCOMM*, pages 2–7, 1987.

- [MHJC95] M.N.Allery, H.E.Price, J.W.Ward, and R.A.Da Silva Curiel. Low Earth Orbit Microsatellites for Data Communications Using Small Terminals. Technical report, Surrey Satellite Technology Ltd, UK, 1995.
- [Mil83] D.L. Mills. Internet Delay Experiments, 1983. RFC 889.
- [Pos82] Jon Postel. Simple Mail Transfer Protocol, August 1982. RFC 821.
- [PR85] Jon Postel and Joyce Reynolds. File Transfer Protocol (FTP), October 1985. RFC 959.
- [PRJT99] Stephan R. Pratt, Richard A. Raines, Carl E. Fossa Jr., and Michael A. Temple. An Operational and Performance Overview of IRIDIUM Low Earth Orbit Satellite System. *IEEE Communications*, 1999.
- [RD] Richard A. Raines and Nathaniel J. Davis. Personal Communications Via Low Earth Orbit Satellite Communication Networks.
- [Ste97] Douglas K. Stenger. Survivability Analysis of the Iridium Low Earth Orbit Satellite Network. Master's thesis, Air Force Institute of Technology, 1997.
- [tel99] Technical Overview of the Teledesic Network. Technical report, Teledesic Corporation, 1999.
- [VH97] Vikram Visweswaraiyah and John Heidemann. Improving restart of idle TCP connections. Technical Report 97-661, University of Southern California, November 1997.
- [Woo95] Lloyd Wood. Network Performance of Non-geostationary Constellations Equipped with Intersatellite Links. Master's thesis, University of Surrey, 1995.

APPENDIX

A. RETRANSMISSION TIME OUT AND ROUND TRIP TIME CALCULATION FOR SUNOS

In actual implementation of TCP for SunOS, the measurement of RTT is based on the accuracy of the timer being using to measure RTT. In practice a timer of accuracy 500ms is used to measure RTT. The timing is done by incrementing a counter every 500ms. As a segment is sent the counter is set to 0. The starting sequence number of the segment whose RTT is being measured is also recorded. When an ACK that include this sequence number is received, the timer is turned off and the count of the counter multiplied by 500ms is the new RTT. Therefore, if the ACK for the segment is received within a period of 500ms then it counted as 1count or 1tick. If the ACK is received after 500ms then the count is again incremented by 1. This means that the RTT is measured in multiples of 500ms.

To illustrate the effect of this timer calculation following experiments were conducted :

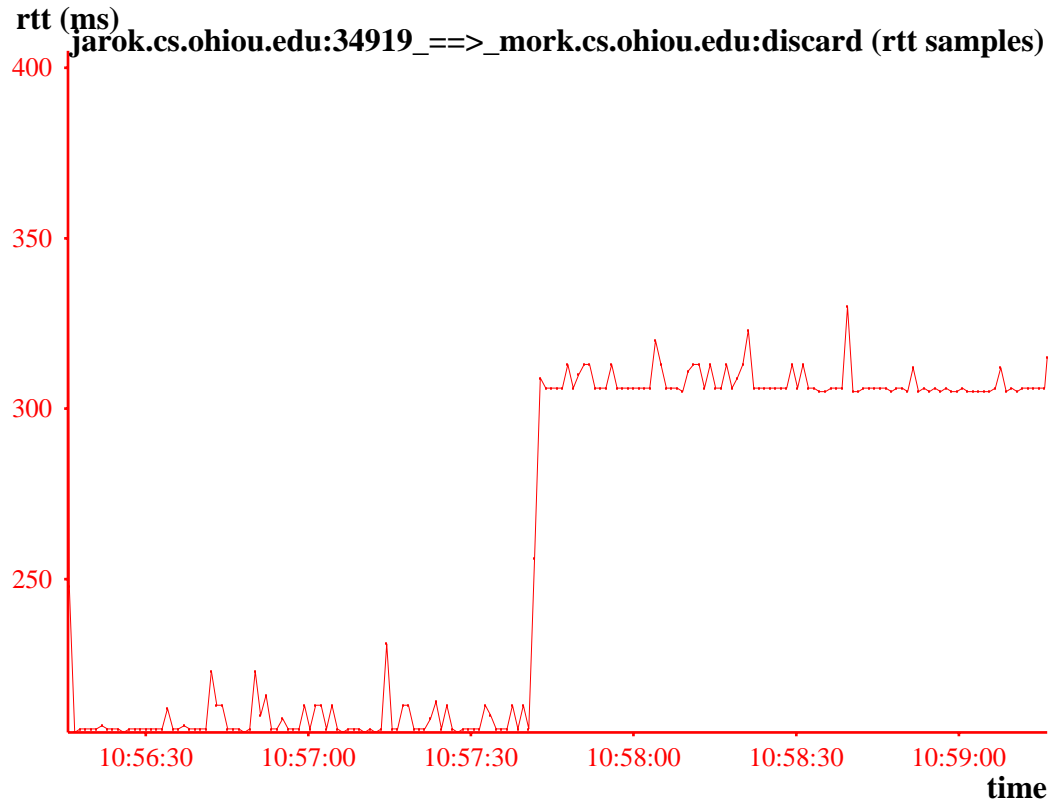


Figure A.1 RTT vs Time Graph Showing an Increase of 100ms in RTT
 This figure gives the RTT vs the time graph for the data transfer between *jarok* and *mork*.

Experiment no. 1 - In this experiment the data transfer is initiated between two machines transferring TCP packets using ONE. Initial time delay for forwarding the packets is set to 100ms and after a certain period of time the delay is increased to 150ms. As seen in Figure A.1, there is an increase in RTT from about 200ms to 300ms.

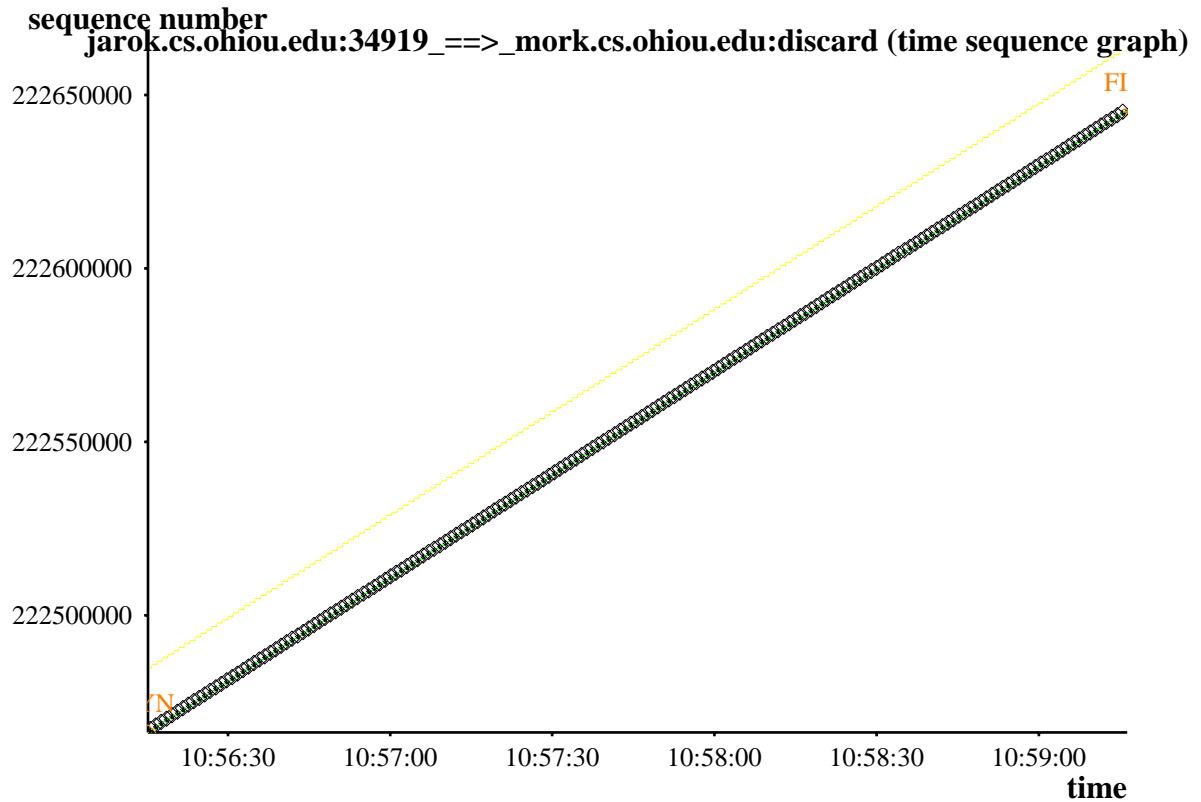


Figure A.2 Time-Sequence graph to Show the Effect of Increase in RTT by 100ms on Packet Loss

This figure gives you the Time-Sequence graph for the data transfer between *jarok* and *mork*.

Figure A.2 shows the time sequence graph for the data transfer. As seen from Figure A.2 there is no retransmission of segments. The reason for no retransmission is the fact the RTT for the whole session was below 500ms. This was always counted as 1 tick or 500 ms irrespective of the actual value of RTT.

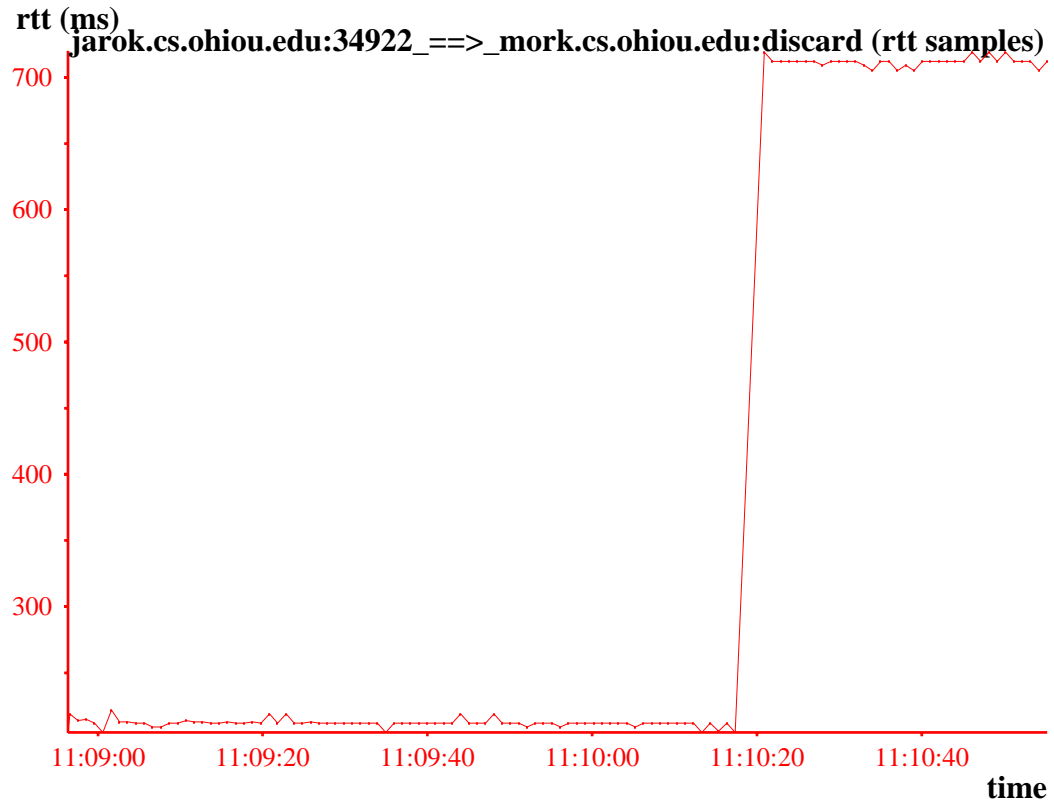


Figure A.3 RTT vs Time Graph Showing an Increase of 500ms in RTT

This figure gives the RTT vs the time graph for the data transfer between *jarok* and *mork*.

Experiment no. 2 - In this experiment the data transfer is initiated between two machines transferring packet using ONE. Initial time delay for forwarding the packets is set to 100ms and after a certain period of time the delay is increased to 350ms. As seen in Figure A.3, there is an increase in RTT from about 200ms to 700ms.

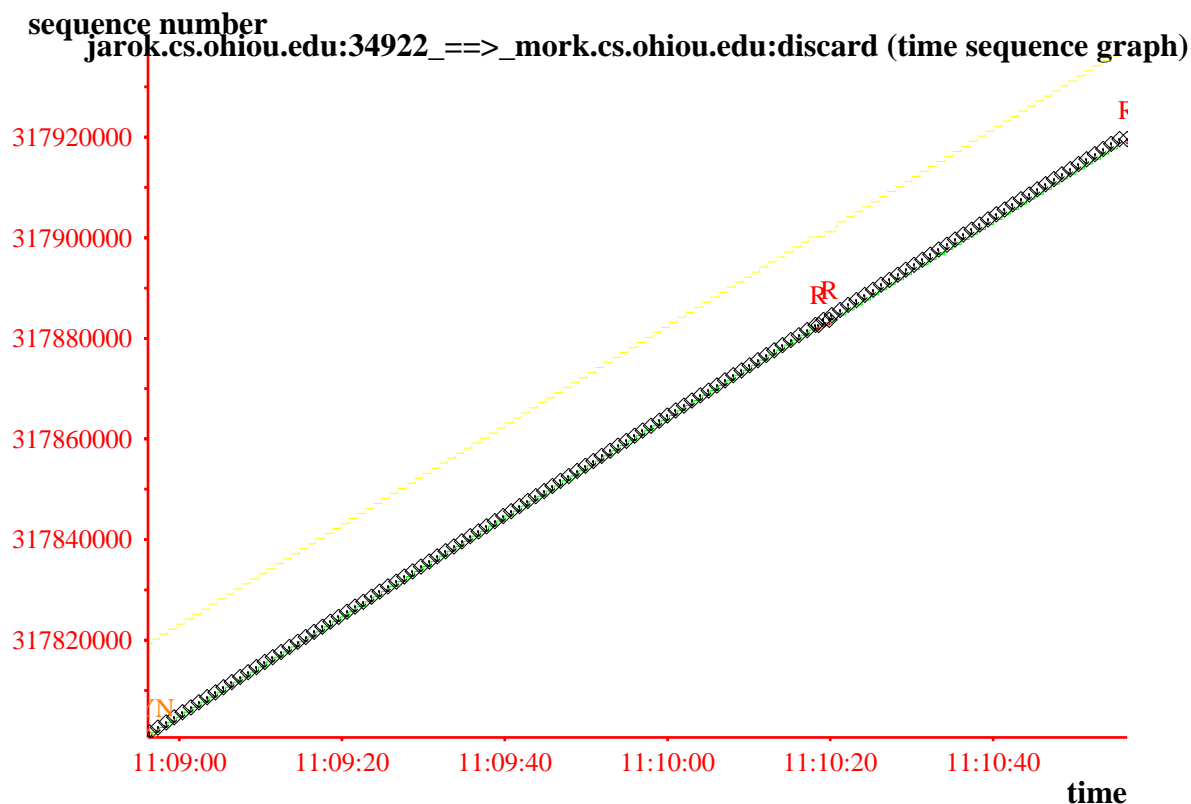


Figure A.4 Time-Sequence Graph to Show the Effect of Increase in RTT by 500ms on Packet Loss

This figure gives you the Time-Sequence graph for the data transfer between *jarok* and *mork*.

Figure A.4 shows the time sequence graph for the data transfer. As seen from Figure A.2 there is retransmission of segments. This time there was a retransmission of segments as the RTT increased from 200ms to 700ms. The reason for this retransmission was the fact that the new value of RTT was more than 500ms and would have been counted as 2 ticks or 1000ms. Therefore effectively there is an increase of RTT from 1tick(500ms) to 2 ticks(1000ms), which resulted in the retransmission of segments. The the previous value of RTO was based on the value of RTT being 500ms. while the new packets are acknowledged after

the period of 500ms resulting in expiration of RTO timer and retransmission of the segments.

B. FILE FORMAT FOR ONE

This section describes the format of the input file which contains all the parameters, required by ONE to calculate the end-to-end delay for the packets that arrive at any one of the Ethernet interface of the machine running ONE. The format of the input file is all follows:

```
#
# Satellite Emulation
#
# propagation:  ms
# the propagation delay in milliseconds
# qsize unit:  size of the outgoing packet queue
# can use units below, no units=bytes
# linespeed unit:  variable
# can use units below, no units=bytes/second
# a linespeed of "infinite" means go as fast as possible
# memunit:  memory allocation size for buffering packets, same units
# as linespeed (1K means alloc packets in 1K allocation units)
# verbose:  number for 0 to 10, 0 is quiet, 10 is noisy
# -----
# values that take args in bits/bytes can use the following units:
# K=1024, k=1000, b=bits, B=bytes, M=1024*1024, m=1000*1000, s=ignored
# so 1 Kbs = 1024 bits/second, and 1 kBs=1000 bytes/second
# initial users 20000
# final users 60000
```

```
verbose: 1
logfile: /export/home/one/bridge.log
pidfile: /tmp/bridge.pid
delayfile: /tmp/bridge.delay
memunit: 1024 B
drawplots: 0
# front side
No_of_orbits: 6
No_of_satellites: 12
I_S_Bandwidth: 2.048 MBs
Source_uplink: 2.048 MBs
Destination_uplink: 2.048 MBs
Source_downlink: 2.048 MBs
Destination_downlink: 2.048 MBs
Source.theta: 22
Source.phi: 15
Destination.phi: 45
Destination.theta: 42
```

C. HOW TO READ A TCP TIME SEQUENCE GRAPH

A TCP Time Sequence graph is a graphical representation of the data transfer for a TCP connection. Each TCP segment has a sequence number to represent the order in stream. The y-axis of the time sequence graph is the TCP sequence number. The x-axis is the time index of the connection in seconds. As the sequence number increases, the graph will increase diagonally. As any point in the graph, the slope of the graph represents the throughput of the connection.

A data segment is represented by a vertical line with arrows at each end. The length of the line is determined by the size of the segments in bytes. The vertical placement of the line on the page represents the position of those bytes in the data stream.

A retransmitted data segment is a vertical line with an arrow on both sides topped with an R. The vertical ticks along the bottom horizontal line represents the duplicate acknowledgments received by the sender. The symbol represents the receipt of 3 duplicate acknowledgments on the sender side, resulting in a retransmission of segments.

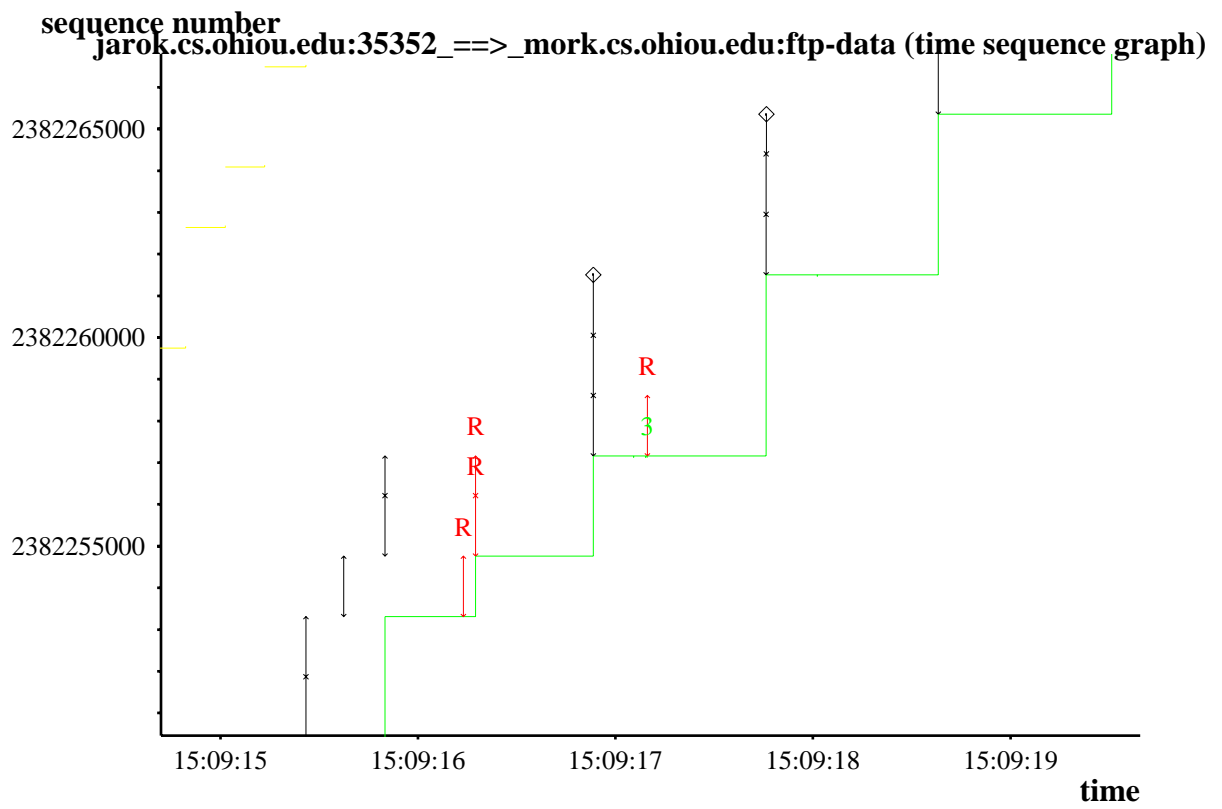


Figure C.1 Time Sequence Graph

This figure shows a part of a time sequence graph with retransmissions and duplicate acknowledgments.

D. EXPLANATION FOR THE PERIODIC VARIATION IN RTT

This appendix is used as an explanation for the periodic variation observed as a part of Experiment 2 in Section 3.2.2. In this particular experiment, 4000 bytes of data is transmitted every 1 second. This results in a transmission of 2 full sized segments(1460 bytes each) and 1 partially full segment (1080 bytes). This is also clear from the Time-Sequence graph in Figure D.1.

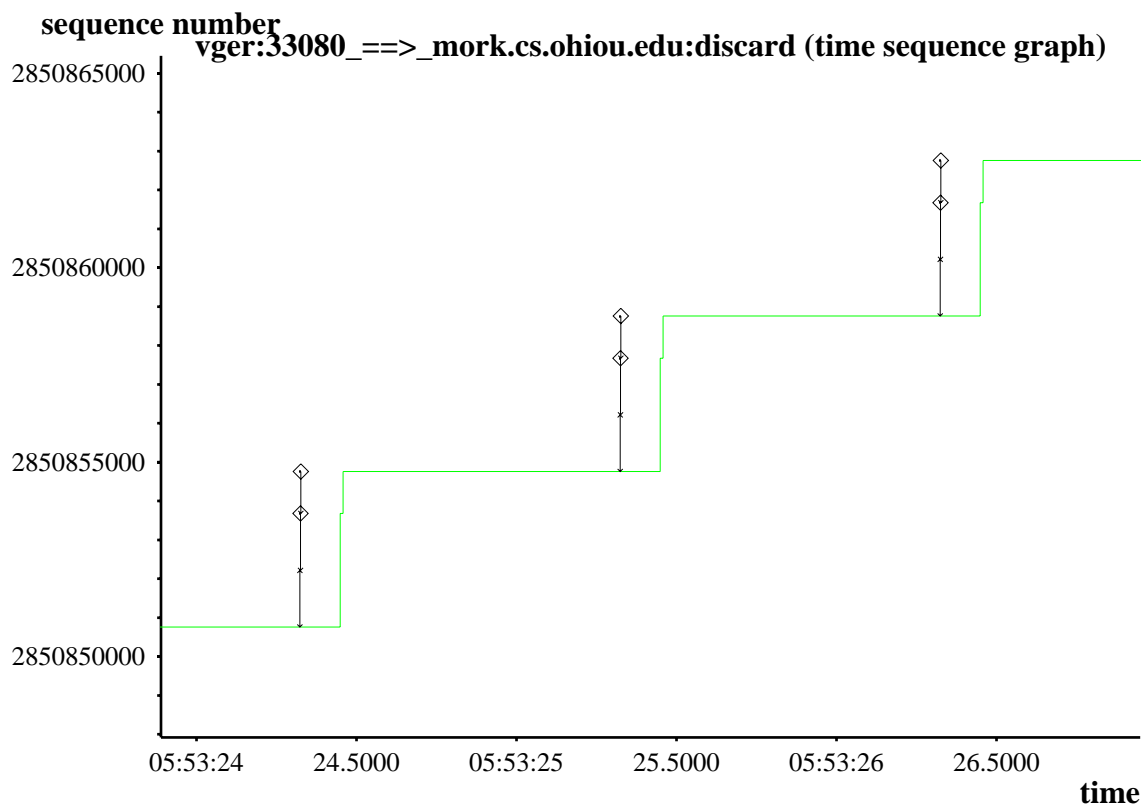


Figure D.1 Time Sequence Graph

This figure shows a part of a time sequence graph showing a transmission of 3 segments after a time interval of 1 second. This graph is a part of the data transfer for Experiment 3.2.2.

As we see in Figure D.2 there is a periodic variation in RTT observed. After studying the trace files generated by the experiment, it has been observed that for the first two segments there is a delayed ACK acknowledging the receipt of the first 2920 bytes of data. Therefore, the calculated RTT for the first transmitted segment (which was transmitted earlier than the second segment) is greater than the actual RTT by the time interval by which the receiver side of the connection waited before sending the delayed ACK. The RTT for the first segment lies on the top of the triangular variation as seen in Figure D.2. Because third segment is not a full sized segment, an instant ACK is generated. The RTT of this third segment is represented

by the point at the bottom of that triangular variation. This behavior is repeated for the whole process of data transfer resulting in a periodic shape of the RTT graph.

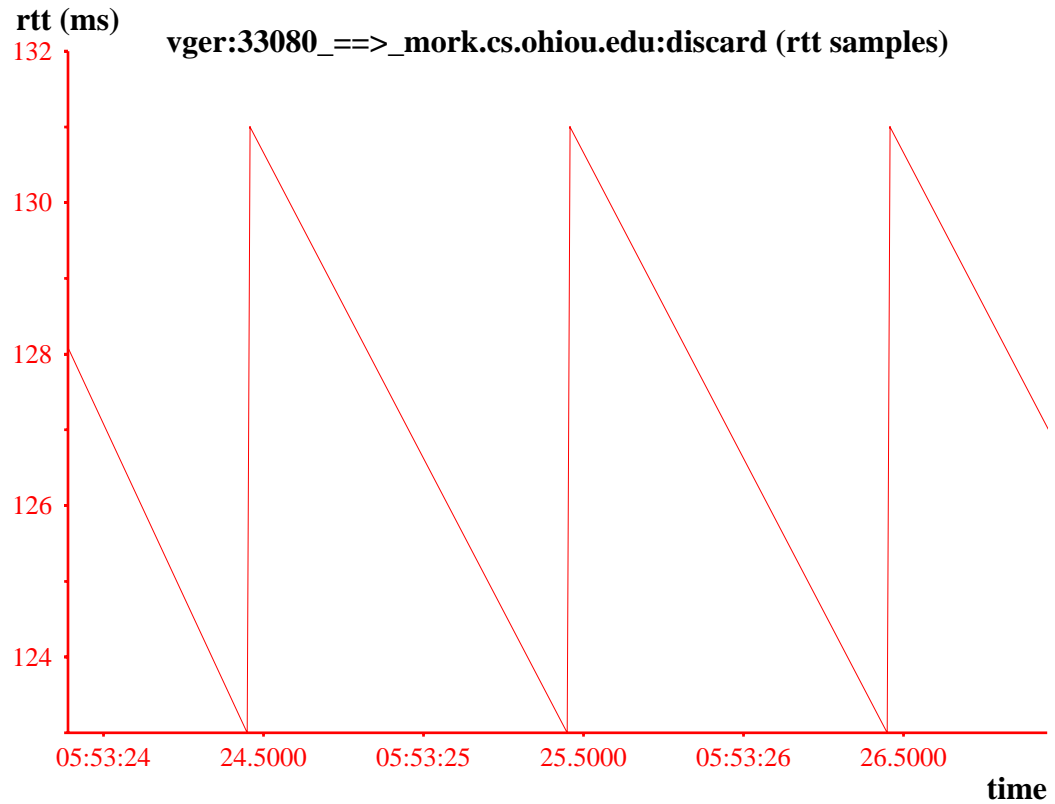


Figure D.2 RTT vs Time Graph

This figure shows a part of RTT vs Time graph for the data transfer. This graph shows a part of periodic pattern in RTT variation.

SANGAL, RAHUL, M.S., AUGUST, 1999
ELECTRICAL ENGINEERING and COMPUTER SCIENCE

Performance Analysis of the Transmission Control Protocol over
Low Earth Orbit Satellite Communication Systems

Director of Thesis: Dr. Shawn Osterman

There has been a proliferation in the development of Low Earth Orbit(LEO) satellite networks to provide seamless global access to Internet. The effect of the variations in propagation delay and user population in a Low Earth Orbit satellite environment on the performance of Transmission Control Protocol(TCP) is an open research question. This thesis examines the variations in Round Trip Time(RTT) that packets in a LEO satellite networks encounter. The variations in RTT due to the relative movement of the satellites is studied and the effect of such variation is summarized. Then the effect of sudden change in user population after the handoff of the satellite connection is studied. The effect of sudden change in user population and its effect on the variation of RTT is analyzed. As a part of this thesis work it has been shown that this sudden change in user population can result in degradation of the performance of Retransmission Time Out and Fast Retransmit Algorithms used by TCP for congestion control.

Approved: _____