Multi-Link Iridium Satellite Data Communication System

by

Mohammad Abdul Jabbar

Bachelor of Engineering, Electrical Engineering
Osmania University, Hyderabad, India, 2001

Submitted to the Department of Electrical Engineering and Computer Science and the Faculty of the Graduate School of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

Thesis Committee

Chairperson: Dr. Victor S. Frost

Dr. Glenn Prescott

Dr. Christopher Allen

Date Accepted: ___________________

To my caring parents who are my inspiration

To my graduate advisor who is a remarkable mentor

Acknowledgements

I would like to extend my sincere and utmost gratitude to Dr. Victor Frost, my advisor and committee chair for his invaluable guidance and inspiration throughout my research. I would like to thank him for giving me the opportunity to work on this project, which is his brainchild. I would also like to thank Dr. Glenn Prescott and Dr. Chris Allen for being on my committee and helping me solve some of the technical problems faced during my research.

I would like to take this opportunity to thank Tory Akins and the entire Greenland field team for their support during the field experiments, without which this work would not have been possible. Thanks are due to Dennis Sundermeyer who helped me with the packaging of the system and carrying out field experiments. I would like to thank Brett Becker for his help with the network design and his support throughout my research. Thanks are also due to Dan Depardo, who helped me with the setup in the wireless lab and provided valuable suggestions.

I extend a special note of thanks to my friends, Nandish Chalishazar and Radha Krishna Mukkai for providing the much needed help during the Greenland experiments. I would like to express my gratitude to all my friends at KU for their moral support.

Abstract

Data and Internet access are integral for the success of any field research. The current state-of-the-art communication facilities, besides their wide penetration are still not available in many geographically remote regions including the Arctic and Antarctic regions. Research in the Polar Region involving data collection and telemetry has grown significantly over past few years. The commercial broadband satellite systems do not provide coverage at higher latitudes. The current geo-synchronous NASA satellites providing broadband access to Polar research have a limited visibility window at poles and need extremely large field equipment. Hence the need for a reliable, portable and easily available data/Internet access system is clearly evident.

In this thesis, we present an Iridium based data communication system that can provide round the clock coverage from pole-to-pole. Since the Iridium satellite system provides a low bandwidth that is not sufficient to support most of the data applications, inverse multiplexing technique is used to combine multiple satellite links into a single logical channel of aggregate bandwidth. Multi-link point-to-point protocol is used to implement packet level inverse multiplexing. This technique effectively increases the available bandwidth per application to useable limits. Field experiments conducted at NGRIP, Greenland showed that the system is highly efficient and reliable. TCP performance analysis showed that the system has high throughput efficiency and high round trip time. An analysis of the observed TCP behavior is presented.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modern telecommunication facilities have grown tremendously over the last few decades. While the local area network speeds exceed hundreds of Mbytes/sec, wide area networks too have advanced from dial-up 56Kbytes/sec connections to T1 lines operating at 1.5 Mbps to fiber links at several Gigabits per second. But, these state of the art technologies have evolved mainly in developed areas. To this day, there are places where these technologies have not penetrated for one reason or another. Some are developing nations; where as other regions are climatically challenged and geographically remote. Arctic, Antarctic and other remote regions are such places where data and Internet access still remains an issue.

Though broadband commercial satellite systems have helped to solve the problem in some of the populated regions, they offer little coverage in oceans and none in the Polar Regions. On the other hand, research in Polar Regions involving data collection and telemetry has grown significantly over the years. Numerous field expeditions are being conducted at various locations year round. The telecommunication requirements of the Polar science community are continuously increasing. In 1999 the bandwidth requirement of the South Pole station alone is estimated at 14Gbytes/day [1]. Given the fact that NASA satellites, currently providing access to these regions (for few hours/day), need extremely large ground terminals, the need for a compact, easily portable and field deployable data communication system is evident. Also, the need for a round the clock lifeline data/internet connection has been strongly felt [2]. Iridium is the only commercial satellite system with true global coverage. However, it provides low bandwidth of 2.4 Kbps, which cannot adequately support the most basic applications desired by the science community.

This thesis develops, implements and field tests a new technique to combine multiple satellite channels to increase the available bandwidth per application. A multi-channel Iridium satellite communication system based on multilink point-to-point protocol (MLPPP) is developed that can provide reliable data/internet access throughout the globe, including Polar Regions. Finally, field tests were conducted in Greenland and the performance analysis of the system is presented.

1.1 Motivation

The Polar Radar Ice Sheet Measurement Project (PRISM) [3] is conducting state-of-the-art research to understand the melting of Polar ice that may lead to rise in sea levels. Though the consequences of this phenomenon may be serious, there is not sufficient scientific data available to characterize it and make reliable predictions [3]. PRISM is developing intelligent sensor systems integrated with robotics and communication systems to study the snow accumulation rate, ice thickness and the ice – bedrock interface, all of which will contribute to a better understanding of the ice balance in Polar Regions.

A sensor web consisting of intelligent rovers and tracked vehicles will be deployed at multiple locations in Greenland and Antarctica to collect data independently on the various ice parameters. This information along with other data shall be sent back to and analyzed by experts using high end computing resources, which clearly cannot be made available on the field. Additionally, to increase the interest of the K-12 student community in scientific research, outreach data such as pictures and video will be sent from the field site to be uploaded over Internet. Field data including weather information, camp pictures, video/audio clips can help teachers to prepare interactive lesson on Polar science.

The data communication requirements of such an endeavor are two folds. While a long range inter–rover communication channel is needed for synchronized measurements by multiple rovers (vehicles), a data link from the field site back to the Internet is required for data telemetry, outreach and Internet access. Such a data link could provide field team with meaningful feedback/help from the scientists and technical experts (not present at site) in real time. A reliable data connection between the field sites and mainland research facilities would not only help field participants with data/Internet access, but would also make it possible for the science community to virtually participate in polar science expeditions; an important component of outreach. Further, videoconference and virtual field tours in real-time or near real time besides video/audio updates are great attractions to the student community. In order to satisfy the data communication requirement of Polar field research in general and the PRISM project in particular, a reliable, mobile data communication system is desired.

1.2 Satellite Data Communications

Due to the climatic conditions and the scarcity of population in Polar Regions, there is a lack of conventional communication facilities like telephone, cable, fiber, microwave links, etc. Hence, the only means of communication at such high latitudes is based on wireless communication. Ad-hoc wireless and wired facilities are being installed on a very small scale to support permanent field camps (like South Pole station). These communication networks enable data communication between the field participants of the camp and to some extent between field sensors. But, the communication with external world relies mostly on satellite communication.

1.2.1 Commercial Satellite Systems

During the past decade number of satellite systems were developed and deployed, while many others are still being developed. Geo-stationary/geo-synchronous system like Inmarsat, Intelsat and PanAmSat consist of a few (2-8) large satellites located along the equator at 35800 km above the earth. Each satellite covers up to one thirds of Earths surface and has high transponder capacity to be shared by users in its footprint. They are easier to maintain, have a long life and are very reliable. Both geo-stationary and geo-synchronous satellites orbit earth at an angular velocity that equals the angular velocity of earth's rotation.

Geo-stationary satellites orbit along equator and remain stationary with respect to a point earth, whereas the geo-synchronous satellites have an inclined orbit, resulting in a short visibility window at Poles. User equipment of both geo-stationary and geo-synchronous consists of parabolic antennae of varying sizes pointed at an angle (called elevation angle) towards the satellite. Though parabolic antenna provides good signal reception leading to a higher bandwidth connection, Mobile platforms would need tracking antennas, which may not be desirable in the Polar weather conditions. Further, some of the user equipment might be too heavy for field applications. The elevation angle and the received signal strength of these satellites decreases with increasing latitude. The connectivity is intermittent at higher latitude and ceases to exist beyond certain latitude. As shown in figure 1.1, Polar Regions are not serviced by these satellite systems. Though Inmarsat does have partial coverage in Greenland, most systems are not accessible beyond 70 degrees N or S latitude.

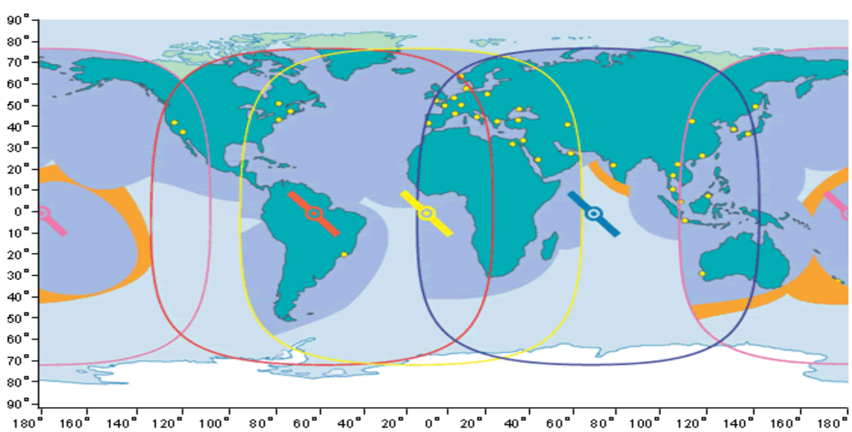


Figure 1.1: Coverage map of Inmarsat (Source: [4])

In recent years, a different system of satellite clusters known as Low Earth Orbiting satellites or LEOs has been launched. Examples of LEOs include ORBCOMM, Iridium, Globalstar, ICO and Teledesic. The orbital period of these satellites is much lower than that of earth. They are visible for a very short time from any given point on earth. Hence, a number of satellites (48, in case of Globalstar) are needed to provide continuous coverage to users.

These LEO satellites are located at a relatively low distance (700-1000 km) from earth. Each satellite covers a small area on the earth and provides connectivity for approximately 10-30 minutes. A user on the land can place a successful call only if both the user and a land gateway lie in the coverage area of an overhead satellite. Hence the coverage is limited both by the number of satellites as well as number of gateway. As seen in figure 1.2, most of the commercial LEO systems do not have footprint/coverage in Polar Regions. The only exception is Iridium satellite system, which does provide coverage in Polar Regions and is dealt with later in this thesis.

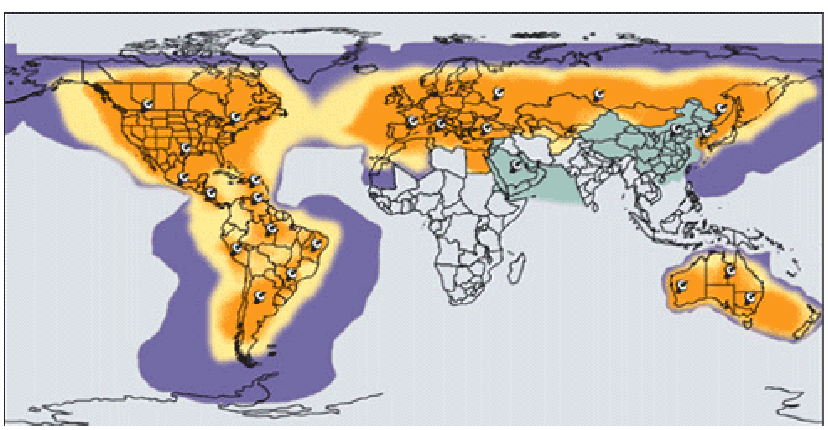


Figure 1.2: Coverage map of Globalstar (Source: [5])
(Orange-primary, Yellow-extended, blue-very weak signal: sporadic connectivity)

1.2.2 Special Purpose Satellite Systems

The commercial satellite systems provide partial or no coverage in deep Polar Regions. So, the current data communication requirements are being met with educational and research satellites. Special NASA and other government satellites are the only broadband systems that can provide coverage in these regions. The current satellite systems being used by US Antarctic Program (USAP) at Poles [6] are ATS3, LES9, GOES, TDRS 1, and MARISAT2, which are either geo-stationary or geo-synchronous satellite systems. ATS 3 and LES 9 are aging systems launched in 1967 and 1976 respectively; ATS 3 is a voice only system where as LES 9 has limited data capability and small visibility window of 5 hours (figure 1.3). Most of the data and Internet access from the science stations and field camps thus relies on special NASA satellites GOES, TDRS 1 and MARISAT 2, which provide data connection with speeds as high as 1 Mbps [1]. Since these satellite systems are geo-synchronous, they have a limited visibility window at Poles. Figure 1 shows various satellite visibility periods at South Pole Station during a typical 24-hour period.

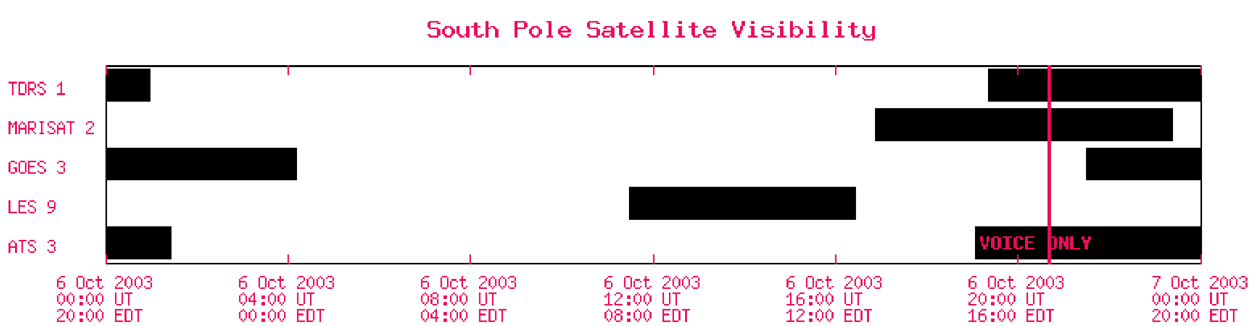


Figure 1.3: Typical South Pole satellite visibility (Source: [7])

It is seen that total connectivity window is approximately 13 hours/day. Clearly, there is lack of round the clock, lifeline data connection.

Some of the other issues with the geo-stationary systems are that they have a very low elevation angle (about 1-4 degrees) from Poles, which combined with high altitude of the satellites results in extremely large field equipment, e.g. antennas, that has to be properly pointed towards the satellite. The GOES and TDRS 1 installations at South Pole [8] use parabolic antennas of 10-meter radii built on large platforms. Though such installations provide high bandwidth, they cannot be used for numerous small field camps and science expeditions conducted each year at various locations in Arctic and Antarctic. NASA has demonstrated the use of lighter mobile PORTCOMM (Portable Communications) terminals with TDRS satellite system at North Pole in April-May 1999 [9]. With an elevation angle of 0.8 degrees they were able to achieve data rate of 4.8 kilobits per second. Higher bandwidths up to 1 Mbps are possible with heavier TILT (TDRS Internet Link Terminal) systems that consume higher power.

1.2.3 Iridium Satellite System

Iridium is a low earth orbiting satellite system with 66 satellites in operation and 14 spares in orbit. It has a true pole-to-pole coverage. The user equipment (phone/modem) is compact and light, comparable to that of an “old cell phone”. Detailed modem specifications are given in Appendix A. It is commercially available and is easily accessible. Further, being a low earth orbiting system, Iridium requires less power and smaller (3.5-inch diameter) patch antennas, which do not need to be pointed in any particular direction. This is especially useful for mobile platforms as in case of PRISM project.

Being the only satellite system that covers the entire globe, Iridium has received wide support from numerous government organizations like DoD and NSF. In a special contract with Iridium, DoD bought an Iridium gateway in Hawaii and is experimenting with the system for use in various research, education and defense projects. Ocean.US is coordinating a project with Omnet, Inc to explore the use of Iridium based data system for collecting ocean data from location where other satellites do not provide coverage[10]. Due to the high costs and other technical problems involved in laying fiber or installing high-speed satellite stations, the South Pole User's Committee [2] has strongly recommended the immediate installation of low-bandwidth Internet over Iridium for high priority data access and as a life line system at South Pole station.

Iridium was initially designed as a voice only system. It can only provide a low data rate of 2.4 Kbps, which is not sufficient for any useful data transfer. Hence, there is a need to develop a data communication system based on Iridium satellites, but with an increased effective system capacity.

1.3 Project Goals

Given that the maximum bandwidth of an Iridium link cannot exceed about 2.4 Kbps, one of the feasible options is to combine multiple channels to increase the effective capacity of the system. The goal of the project is to develop a moderate bandwidth solution that can meet the communication needs of Polar research camps in general and that of PRISM in particular. The specific objectives are as follows.

1. Develop a data communication system to provide reliable data communication from Greenland and Antarctica to the Internet.
2. System level design of a multi-channel Iridium data communication system to increase the available bandwidth per application.
3. Implement the system and conduct field experiments in Greenland.
4. Evaluate the general performance and reliability of the system.
5. Characterize the TCP/IP performance over a multi-channel Iridium system.

1.4 Accomplishments

- A system level design of a scalable multi-channel system was developed.
- Multi-link point-to-point protocol was used to inverse-multiplex (combine) multiple Iridium satellite links in to a single logical channel of aggregate bandwidth.

- Link management software that ensures fully autonomous and reliable operation is developed. Also, an end-to-end network architecture providing Internet access to science expeditions in Polar Regions is demonstrated.
- A 4-channel system was implemented and field-tested at North GRIP, Greenland and experiments were conducted to determine the performance and reliability of the overall system.
- Based on the analysis of data collected in the field experiments, the 4-channel system was found to be over 90% efficient providing an average throughput of 9.2 Kbps.
- The system had an average system up time of 95% and a full capacity up time of 80% and thus is reliable. Also, the mobile performance of the system is very similar to that of stationary systems.
- The system's minimum round trip time of approximately 2 seconds was found to impair real time interactions.
- Hardware and software issues experienced in the field were captured and corrective measures suggested.

1.5 Thesis Organization

Since the work done is primarily based on the Iridium satellite system, it is discussed in detail in Chapter two along with the point-to-point protocol (PPP) and its extension multi-link point-to-point protocol (MLPPP), both of which are crucial to the project.

Chapter three presents the complete system design, individual components of the system and its operation. The requirements, network architecture and the protocol stack are also discussed.

The actual implementation of the system (in Linux) is detailed in Chapter four. Flow control diagrams explaining the overall operation of the system are given. Individual software blocks used for modem control and user authentication are also explained. The optimum link parameters and user authentication is also dealt here.

Chapter five explains the tests conducted on the field (in Greenland) and the results obtained after data analysis. The system performance and reliability are clearly characterized. Further, an evaluation of TCP/IP performance over Iridium is given.

Chapter 2

Background

2.1 Iridium Satellite System

The idea of a global satellite telephone network, involving a constellation of 77 LEO satellites was first conceived in 1987 and named "Iridium" after the 77^{th} element in the periodic table. The final system was completed in May 1998 with an enhanced design consisting of 66 satellites only. The satellites are located at about 780 km from ground, well above the residual atmosphere and below the Van Allen radiation environment, thus preventing it both from atmosphere and radiation effects without any shielding [17].

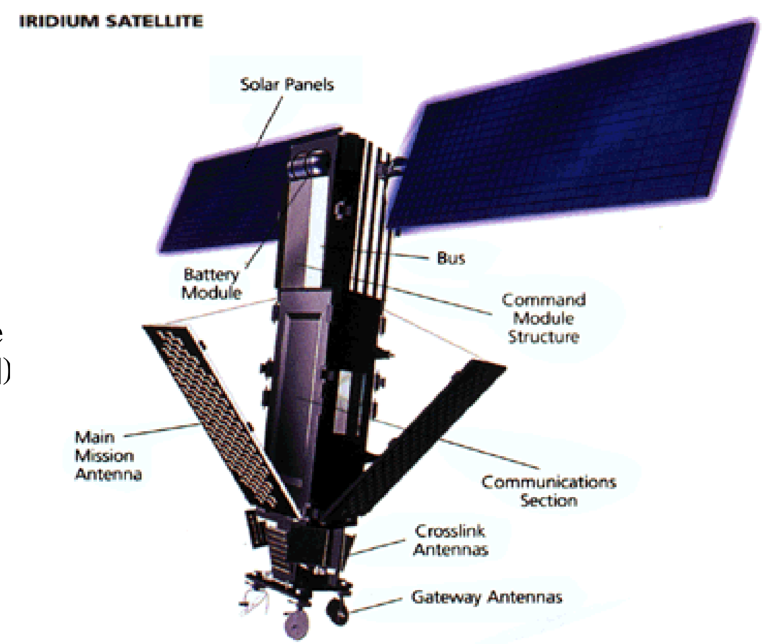


Figure 2.1: Iridium satellite (Source: [14])

The 66 satellites are arranged in six planes with near circular orbit inclined at 86.4 degrees and containing 11 satellites each. These near polar orbits help to provide a true pole-to-pole coverage. Iridium satellite as shown in figure 2.1 weigh about 1500 lb and have an expected lifetime of 5 to 8 years.

2.1.1 Architecture

The LEO satellite system consists of a constellation of satellites that communicate with users on one end and terrestrial gateways on the other end to complete a call. There are two distinct ways of designing a satellite-terrestrial network. Satellite systems like Globalstar are designed on “bent pipe” or “single hop” architecture [15], where each satellite is just a radio repeater. The user’s signal bounces of the satellite to a land station in the same footprint (beam spot) of the satellite. This architecture has an inherent disadvantage that both the user and the gateway must be in the footprint (coverage) of the same satellite for a successful data/voice call. Such systems need large number of costly earth stations all over the world for extensive coverage. Hence such system usually do not have true global coverage, instead, they concentrate on major commercial landmasses. Seas, Oceans and Polar Regions are not serviced.

Another type of architecture, like that of Iridium, involves intelligent satellites with on board switching technology. In this case each satellite maintains inter-satellite links (ISL’s) with other satellites (in case of Iridium, with 4 other satellites). If the satellite directly overhead the mobile user does not have a land station in its visibility (beam spot), it can route the call through multiple satellite hops to the nearest gateway, which eventually completes the call to its destination. Similar routing technique is used for mobile-to-mobile voice calls where the signal hops over multiple satellites and is down-linked directly to the mobile receiver. Each satellite maintains a routing table (analogous to routing in terrestrial networks), which is updated every 2.5 minutes [16], and uses a proprietary dynamic routing protocol to route/switch calls. Since Iridium implements this type of architecture, it can provide global coverage with fewer numbers of gateways. But, the satellites need steerable antennas to maintain (Inter-Satellite Links) ISLs besides other complex features. Thus the price paid is in terms of the satellite complexity, which directly affects the weight of the satellite. Though a single gateway would theoretically provide global coverage, Iridium has 13 gateways located in US, Italy, India and other countries for ease of management, reliability and other technical reasons [17]. A typical gateway is shown in figure 2.2.

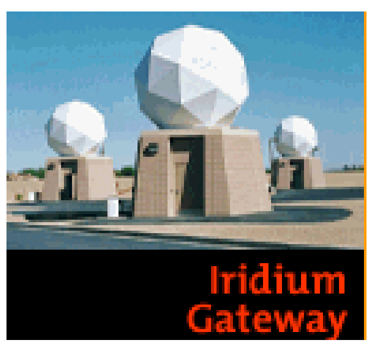


Figure 2.2: An Iridium gateway (Source: [13])

Each gateway [13] has a 17-foot diameter radome that houses a 3-meter diameter auto-tracking antenna operating in the range of 20-30 GHz. The inter-satellite links operate in the ka band of frequencies between 22.55 GHz and 23.55 GHz at a rate of

25 Mbps. The communication between the satellites and the gateways also occur in the ka band. While the uplinks span a frequency range of 29.1 to 29.3 GHz, the down links use 19.1 to 19.6 GHz range. The user (phone) to satellite link operates in L-band between 1616 MHz and 1626.5 MHz and is detailed in section 2.1.5.

2.1.2 Constellation Design

Iridium satellites are located at an altitude of 780 km (GEO satellites are 35800 km) above the ground. In order to determine the land coverage and the view time of a satellite from a location on ground, consider the relative velocity of the satellite with respect to earth. The velocity of the LEO satellite [15] with respect to the earth (V_l) is given by the equation:

$$V_l = \frac{\omega R_g^{3/2}}{\sqrt{R_l}}$$

where

R_l – radius of the LEO satellite orbit

R_g – radius of the GEO satellite orbit

ω – Angular rotational speed of Earth

$$\omega = \frac{2\pi radians}{24 hours} = 0.2618 rad / hour$$

The radius of a satellite orbit is the sum of the radius of earth (6378 km) and satellite altitude. The radius of GEO orbit (R_g) = 6378 + 35800 = 42178 km

The radius of the Iridium orbit (R_l) = 6378 + 780 = 7158 km

Using the above values, the relative velocity of the Iridium satellite is found to be 26,804 km/hr. Since this is a very high value, the performance of mobile users with speeds as high as that of an airplane should be the same as that of stationary users. Further, the orbital period of the Iridium is given as $2\pi R_g / V_l$ [15], which is found to be 100.13 minutes.

Due to the low altitude of the Iridium satellites, the minimum elevation angle is 8.2 degrees [15]. Further, the average connection/access window to satellite from point on the ground is in the range of 9-10 minutes [18]. As the connection window comes to an end, the hand-off algorithm is used to transfer the call from the outgoing satellite on one horizon to an incoming satellite on the other horizon. Thus, a user experiences a call hand-off roughly about every 10 minutes. This hand-off occurs between two satellites that are at an angle of 8.2 degrees to the user on opposite horizons as illustrated in figure 2.3. The user, hence, should ideally have an unobstructed 163.6 degrees view and enough signal strength to connect to the incoming satellite on the horizon. This forms one of the potential reasons for call drops.

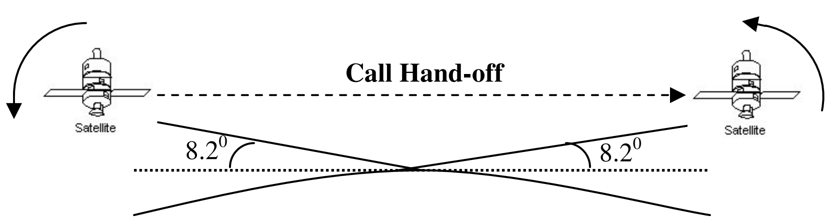


Figure 2.3: Call hand-off between two satellites

The coverage area of a single satellite [15] can be determined using the following equation.

$$A = 2\pi R^2{}_e(1-\cos\theta)$$

As shown in figure 2.4 [19], the earth's radius, Re is 6378 km and the earth's central angle θ is given be the equation

$$\theta = \left[\cos^{-1}\left(\frac{R_e \cos E}{R_e + h}\right)\right] - E$$

where,

E – minimum elevation angle = 8.2^0

h– altitude of the satellite = 780 km

With the above values, the coverage area of a single satellite (A) is found to be 15,299,900 km^2. This results in a footprint radius of 2209 km per satellite.

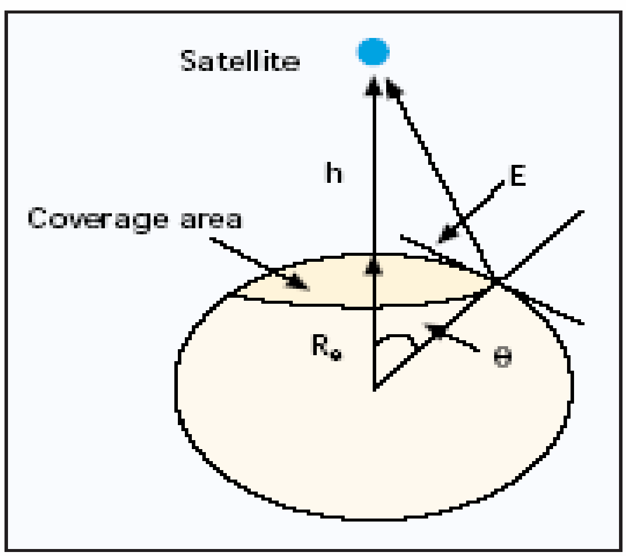


Figure 2.4: Coverage area of a satellite (Source: [19])

2.1.3 Multi-access scheme (TDMA/FDMA)

The Iridium system uses a combination of TDMA and FDMA along with frequency reuse to maximize the capacity of the system. In this section, the multi-access scheme is discussed in detail.

The user-satellite link uses a frequency range of 1616 MHz to 1626.5 MHz, leading to an aggregate bandwidth of 10.5 MHz. A FDMA scheme is used to divide this bandwidth into 240 channels, each with a bandwidth of 31.5 KHz. as shown in figure 2.5 [19]. The channels are spaced 41.67 KHz apart to provide guard bands to minimize the inter-modulation effects and to allow for Doppler frequency shifts [19].

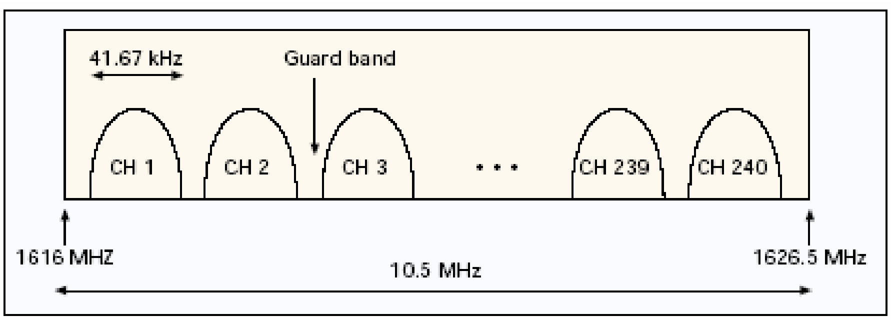


Figure 2.5: Iridium FDMA scheme (Source: [19])

Each satellite has 3 phase array antennas spaced 120^0 apart, as shown in figure 2.1. A close up of the three phased array antennas is shown in figure 2.6. Iridium system divides the satellite footprint into number of small cells called spot beams. Each antenna provides 16 spot beams, leading to a total of 48 spot beams per satellite. If there were no frequency reuse then the total number of frequency channels per spot beam (or cell) is given by

$$\frac{240\ channels\ per\ satellite}{48\ cells\ per\ satellite} = 5\ frequency\ channels\ per\ cell$$

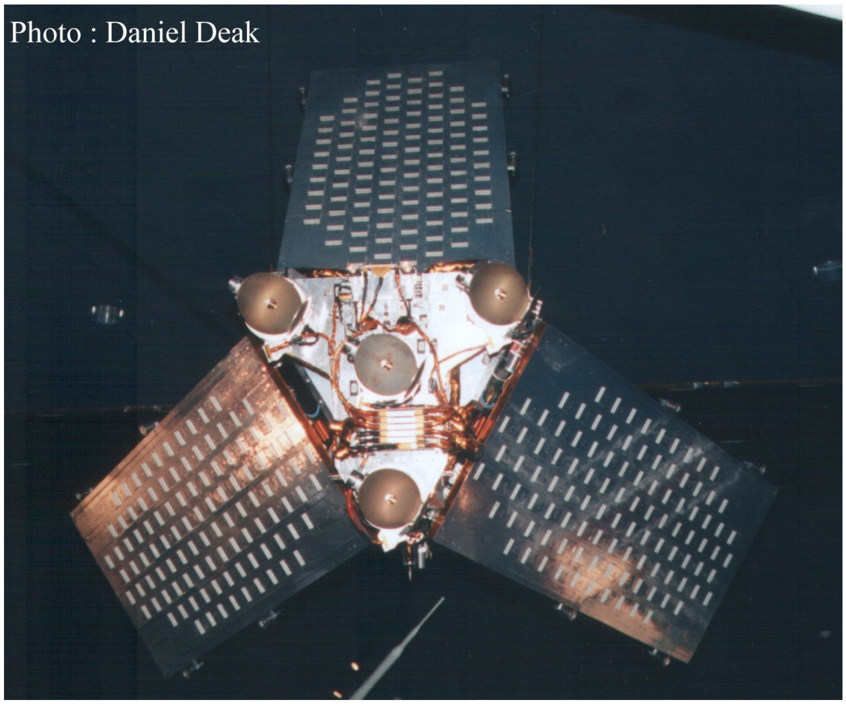


Figure 2.6: The three panels of an Iridium phased array antenna (Source: [12])

To increase the capacity, Iridium implements frequency reuse such that cells far apart can use the same frequency channels; the interface between such cells would be negligible. The frequency reuse factor, in case of Iridium is 12 [19]. This ensures that the carrier-to-interference ratio is high enough to meet the requirements of signal quality, fade margin and handle signal level fluctuations. Hence each cluster consists of 12 cells as shown in figure 2.7. Cluster, in this scenario, is defined as the group of cells, which should use unique frequency channels.

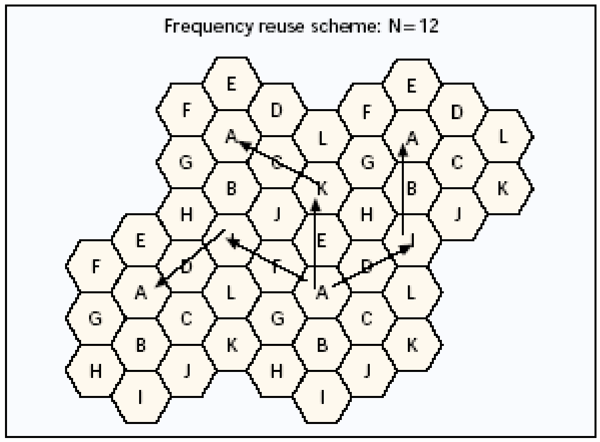


Figure 2.7: Frequency reuse in Iridium (Source: [19])

The frequency reuse factor is defined as $N = I^2 + I.J + J^2$, where I and J are arbitrary integers that represent the number of cells in I and J directions that are 60^0 apart. For Iridium, I=J=2. Hence the frequency reuse factor is 12. Each cluster can utilize full 240 channels. In order to find cells that use the same frequency, move "I" cells from a given cell in one direction, then turn 60^0 and move "J" cells to arrive at the new cell that uses the same frequency as that of the initial cell.

With frequency reuse the number of frequency channels per cell is given by

$$\frac{240\ channels\ per\ cluster}{12\ cells\ per\ cluster} = 20\ frequency\ channels\ per\ cell$$

In order to further increase the available capacity, a TDMA scheme is implemented per frequency channel. The Iridium TDMA frame (at a burst data rate of 50 Kbps) is of 90 ms duration and consists of 4 uplink and 4 downlink slots separated by small guard times as shown in figure 2.8.

The data rate achieved with such a structure is discussed in section 2.1.4.

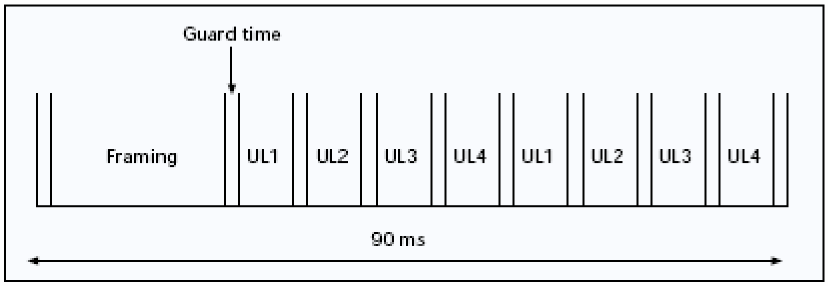


Figure 2.8: Iridium TDMA scheme (Source: [19])

2.1.4 System Capacity

As seen in section 2.1.4, there are 20 channels per cell. Further, each of the 20 channels in a cell has 4 user slots. Hence the capacity of the system would be 80 users/cell. The total number of cells possible with the Iridium constellation is

$$48\frac{cells}{satellite} * 66\ satellites = 3168\ cells$$

But due to cell overlap at the poles, it turns out that only 2150 cells are needed to provide global coverage. An algorithm implemented on the satellites determines which of the spot beams should be shut down in the regions where multiple beams overlap, such that only one beam services a particular area.

Therefore, the total capacity of the system is

$$80\frac{users}{cell} * 2150\ cells = 172{,}000\ users$$

Hence, at any given point of time Iridium can support 172,000 simultaneous users, which is the highest in all the LEO satellite systems. The data rate experienced by users will be discussed later in section 2.1.4.

Further, in each cell, Iridium dedicates 81.25 % of the total 80 channels for full duplex (FDX) voice channels and the remaining 18.75% are data channels. Hence each cell can support 65 FDX voice channels and 15 data channels. Globally, there are 65 * 2150 = 139,750 FDX voice channels and 15*2150 = 32,250 data channels. The allocation of 81.25 % to voice and 18.75 % to data is believed to be customizable.

Maximum number of users that can be co-located at a give place

Though, the total number of data/voice channels is large, the total number of users supported per cell as discussed above is 80. From section 2.1.2 it is clear that each satellite covers an area of 15,299,900 sq km, which comprises of 48 cells.

$$Area \text{ cov} ered\ by\ one\ cell = \frac{15{,}299{,}900\ sq\ km\ per\ satellite}{48\ cells\ per\ satellite} = 318{,}748\, km^2 \big/ cell$$

Hence the maximum number of users that can be co-located in an area of about 318 km radius is 80, in which there are 65 Voice channels and 15 data channels.

2.1.5 Network Performance (Throughput and Delay)

In order to determine the maximum data rate that is supported by the Iridium system let us again look at the TDMA/FDMA structure of the access network. Though the exact specifics of the TDMA/FDMA schemes used in Iridium are not published in literature due to its competitive nature, a reasonable analysis can be made on the most likely configuration.

Theoretical Throughput

The total 10.5 MHz of the available user-satellite bandwidth is divided in to 240 channels each with a bandwidth of 31.5 KHz and spaced 41.67 KHz apart with appropriate guard bands. Each TDMA frame is transferred at a burst data rate of 50 Kbps. Iridium uses QPSK modulation [14] with raised cosine filtering, in which case the occupied bandwidth (1.26 * 50 kbps/2 = 31.5 kHz), matches with the original assumption [17].

The TDMA frames are 90 ms long and have 8 user slots in all. One possible scheme is to have a user slot of 8.29 ms duration and use the rest to provide the guard bands and the packet header. In this case, assuming the system uses 3/4 forward error correction, the maximum information bit rate is = 0.75 *(8.29msec/90msec) * 50 kbps = 3.45 kbps. The proprietary voice/data coder supports information bit rate at 2.4 Kbps, with possible non-uniform error protection. Further, the actual data rate of the system may vary depending on the antenna efficiency, atmospheric conditions [14]

Theoretical Delay

The total delay experienced depends on the performance of the routing algorithm implemented on satellites, along with the transmission and propagation delay. Given the lack of information of the routing algorithm and its performance, the end-to-end packet delay due the other factors can be estimated for an Iridium-to-Iridium call as shown below[19].

$$T_{packet} = T_{transmission} + T_{uplink} + (N-1)T_{cross} + N.T_{sat} + T_{downlink}$$

Where,

$T_{transmission}$ is the transmission time for a given packet size

T_{uplink} is the propagation delay from ground to the overhead satellite

T_{cross} is the propagation time between satellites cross links

$T_{downlink}$ is the propagation time from the satellite to the ground and

T_{sat} is the processing and queuing delay per satellite node

N is the number of satellite nodes in the path

$$T_{uplink} = T_{downlink} = \frac{satellite\ altitude}{speed\ of\ light} = \frac{780\ km}{3x10^5\ km/\sec} = 2.05\ ms$$

The average distance between satellites is about 4000km. Hence,

$$T_{cross} = \frac{crosslink\ dis\tan ce}{speed\ of\ light} = \frac{400\ km}{3x10^5\ km/\sec} = 13.33\ ms$$

For a 100 byte packet, the transmission delay can be calculated as

$$T_{transmission} = \frac{packet\ size}{average\ throughput} = \frac{100\ bytes}{2.4\ Kbits/\sec} = 333\ ms$$

While the value of T_{sat} is not known, the current switching technology suggests that this value should be in the range of few hundred microseconds. The average end-to-end packet delay for a 100 byte packet that goes through, say, 3 satellites cross links should be approximately 380 msec. This was measured and the RTT was found to be 2.5 seconds. While the exact reason for this difference is not clear, possible sources of delay are discussed in chapter 5.

2.1.6 Hardware

Figure 2.9 shows an Iridium data modem, manufactured by Motorola. These modems respond to modified Hayes AT command set [37] and provide a serial interface for data connectivity. They operate in the L-band providing a data rate of 2.4 kbps and link margin of 16 dB at channel BER of 10^{-5}. An external patch antenna, also seen in figure2.9, is connected to the LBT using a TNC connector. As mentioned in the section 2.1.5, this antenna should have a clear 160^0 view.

Figure 2.9: Iridium data modem and antenna (Source: [20])

When the trans-receiver goes off hook, it requests access with the satellite and after a brief moment in assigned one of the available frequency channel from the 20 channels of cell in which it is located. Further, it is also assigned transmit and receive slots in the TDMA frame.

2.1.7 Types of data services

The data services of Iridium can be classified primarily into two types, the Iridium-to-Iridium (or mobile-to-mobile) service and Iridium-to-PSTN service. The Iridium-to-PSTN service can again be further divided into dial-up data and direct Internet service.

Iridium-to-PSTN service

In this mode of service, one end of the communication link is the Iridium user and the other end is a PSTN user or an ISP. As shown in figure 2.10, the Iridium modem uplinks the user call to the satellite directly overhead.

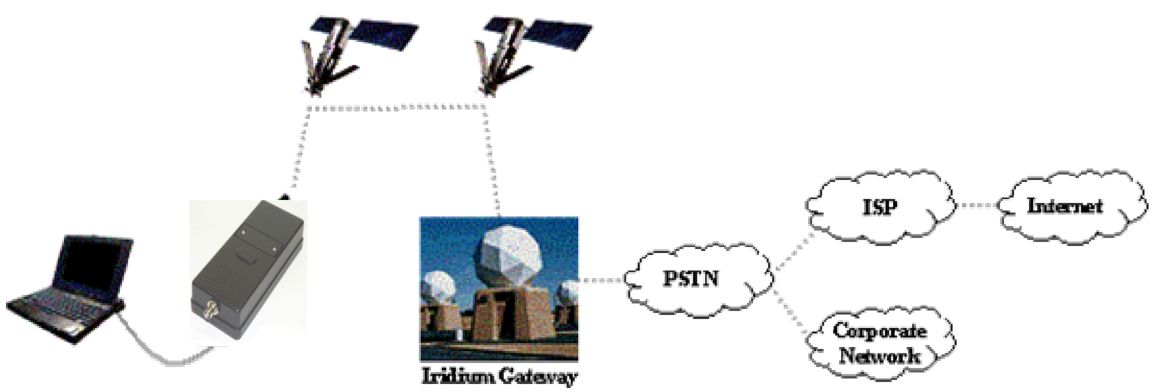


Figure 2.10: Iridium-to-PSTN data connection (Source: [21])

This call hops through a number of satellites and is down linked to one of the Iridium gateways. If the user has requested a dial-up data connection, the call is switched via the PSTN to the destination user or a corporate network. On the other hand if direct-internet connection is requested, then the call is switched via the PSTN to an ISP provider or the gateway itself will act like an ISP and the user gets connected to the Internet. In this case data traverses the satellite earth segment four times per round trip.

Iridium-to-Iridium service

In this case, both the ends of the communication link are Iridium users as shown in figure 2.11. In this case the call is up linked by the modem, routed through number of Iridium satellites and is downlinked to the gateway. The gateway determines its destination and uplinks it back to the satellite network. The call hops through few more satellites before it is down linked to its final destination. In this case, data travels the satellite-earth segment eight times per round trip, increasing the delay significantly.

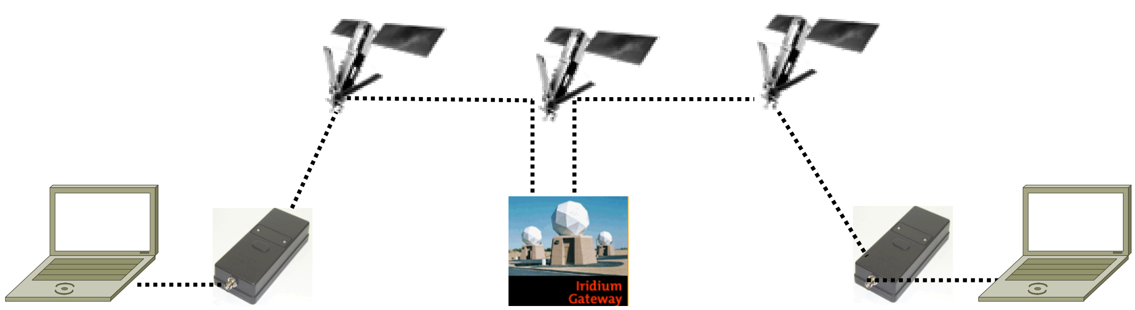


Figure 2.11: Iridium-to-Iridium data call

In order to reduce the large delays associated with an Iridium-to-Iridium call, a new data service called "Data after Voice" has been launched that treats the data as a voice call and eliminates the extra 4 satellite-gateway hops per round trip.

2.1.8 Summary of Iridium

Based on the previous sections, a brief summary of the important parameters of the Iridium satellite system are given below.

Satellite Type	LEO
Application	Voice, Data
Satellite altitude	780 km
Expected Life	5-8 years
Number of Satellites in orbit	66
Switching Technology	On-board
User-Satellite link	L-band, 1616 – 1626.5 MHz
Inter satellite link	Ka-band, 22.55 – 23.55 GHz
Gateway – satellite up link	Ka-band, 29.1 – 29.3 GHz
Satellite-gateway down link	Ka-band, 19.1 – 19.6 GHz
Relative Velocity	26,804 km/hr
Minimum elevation angle	8.2^0
Average satellite view time	9-10 minutes
Coverage radius of a single satellite	2209 km
Access scheme	FDMA and TDMA
Frequency reuse factor	12
Total system capacity	172,000 simultaneous users
Maximum number of located users	80 users in a radius of 318 km
Theoretical throughput	2.4 – 3.45 Kbps
Type of data services	Iridium-to-Iridium, Iridium-to-PSTN

Table 2.1: Summary of Iridium system

2.2 Supporting Technologies

In this section, the concept of inverse multiplexing and its need is discussed. In order to understand inverse multiplexing, first consider the widely know traditional multiplexing, where multiple streams of data are combined into a single high-speed data pipe. As shown in figure 2.12, the data from multiple applications or users is sent over a single high bandwidth link.

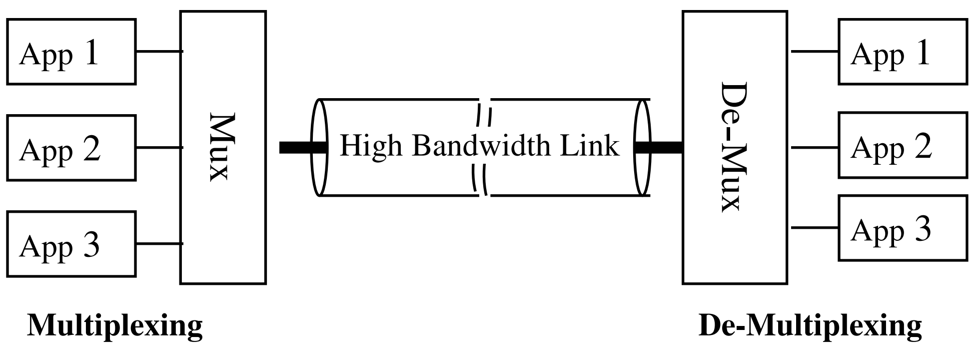


Figure 2.12: Traditional multiplexing

This method is commonly used to aggregate Internet traffic over a campus/corporate network, which has a single high-speed link Internet link. In cases where the available link speed is moderate to low and the bandwidth requirement is high, a number of links are then used to share the load from different applications. An example of Load Sharing is a situation, where the video/audio and the internet/data traffic are routed/sent on separate T1 links resulting in an effective bandwidth that is aggregate of the two links.

2.2.1 Inverse Multiplexing

Load Sharing has the inherent disadvantage that the bandwidth per application is equal to the bandwidth of a single link, even when more than one link is being used. This is especially problematic in cases where the available link speed is too low (of the order of few kilobits) to support most of the regular data applications. Typical example is the Iridium system with an effective bandwidth of 2.4 Kbps. This problem can be solved using Inverse multiplexing technique, which combines multiple links in to single logical link that is the aggregate of the individual link bandwidths, thereby increasing the available bandwidth per application significantly. As shown in figure 2.13, a high-speed data stream from a single application is divided into partial data streams that are transmitted at the same time across separate channels and are recombined at the other end in to the original data stream.

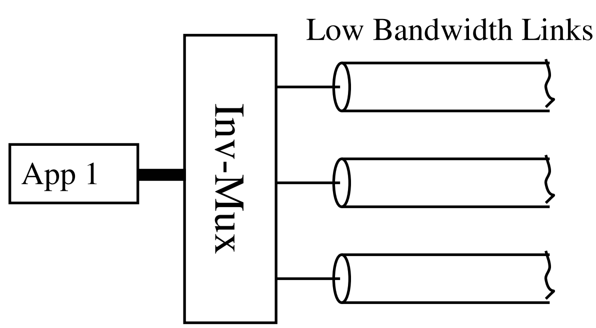


Figure 2.13: Inverse multiplexing

From the application point of view, this setup looks like a single logical data pipe, thus providing an higher aggregate bandwidth that can be effectively used by many applications that are otherwise not supported due to low bandwidth. Further, inverse multiplexing has a better ability to recover from link failures. In case of load sharing a link failure will cause the application running on that link to be dropped, where as in case of inverse multiplexing, the data channel remains active with a reduced bandwidth as supported by the remaining channels. This results in the reduction of the effective bandwidth during the period of the link failure.

Inverse multiplexing uses less equipment and in many cases, pure software solution exists. Also, different links (like T1, frame relay, dial-up, etc) at different data rates can be inverse multiplexed. Further, inverse multiplexing is scalable, i.e., capacity can be easily added. Depending on the bandwidth requirements, more channels can be added or removed from the bundle seamlessly. Typical installations of inverse multiplexing include corporate networks that use multiple T1 links to meet the bandwidth requirements and Internet users that combine two dial-up lines to obtain ISDN like speeds. The latter case is particularly important in remote places where modern high-speed facilities like ISDN, Cable Modem and DSL do not exist.

There are different ways to implement inverse multiplexing. While both software and hardware solutions are available, software solutions are cheap as they do not need additional hardware. Inverse multiplexing can be done at bit level (bonding) or at packet level, latter being more popular due to widely available standards and protocols.

2.2.2 Point-to-Point Protocol (PPP)

One of the software implementations of inverse multiplexing is Multilink extension to the well know point-to-point protocol. To understand the Multilink, first consider the important features of PPP protocol.

Point-to-point protocol, as explained in RFC 1661 [22], "is a standard method for transporting multi-protocol datagrams over point-to-point links". There are 3 main components of PPP [22].

1. A method for encapsulating multi-protocol datagrams
2. A link control protocol (LCP) for establishing, configuring and testing the data link connection
3. A family of network control protocols for establishing and configuring different network-layer protocols.

PPP uses 8 octets to encapsulate different network layer protocols simultaneously over the same link. The network layer datagrams are encapsulated in to one or more PPP frames. Thus a PPP frame is an encapsulation of IP, IPX and other network layer packets. A link control protocol is used to establish and terminate a point-to-point link between two nodes and handles various link parameters like encapsulation formats, size of packets, user authentication and failure/error detection. In order to support various network layer protocols and handle their requirements a family of network control protocols are defined. As both the ends communicate their capabilities and requirements, most of the link parameters are automatically configured, like the assignment of IP addresses, etc.

In order to understand the link operation of the point-to-point protocol, consider the phase diagram as shown in figure 2.14.

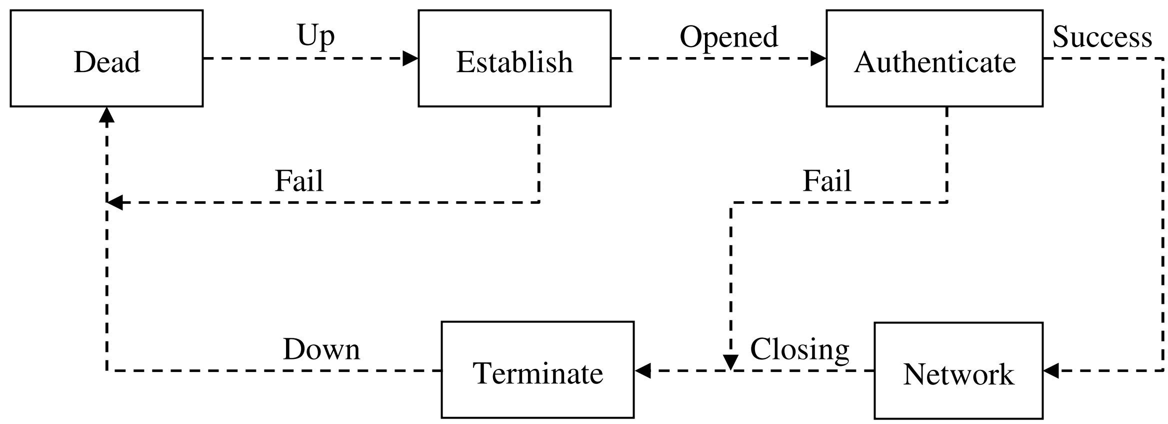


Figure 2.14: PPP phase diagram (Source: [22])

A PPP link starts and end in the dead state. When the physical medium is ready, the link enters the establishment phase where it negotiates a PPP connection and link parameters with its peer using link control protocol. Failure to negotiate will send the link to dead state. Once the link is successfully configured, peer authentication is done using the specific authentication protocol negotiated during the link establishment phase. Though authentication is not mandatory, if used, the peer must successfully authenticate before the link proceeds to the network layer protocol phase. Individual network-layer protocols (like IP, IPX, etc) are configured by separate network control protocols (NCP's) in the network state. After configuring the network layer protocols, the link is ready for data transfer. Finally, the link may enter termination phase due to authentication/network phase failure, physical link failure or an administrative request to close the link. Link control protocol is used to terminate the link gracefully or a timer is used in case of physical link failure. In either case the link returns to the dead state.

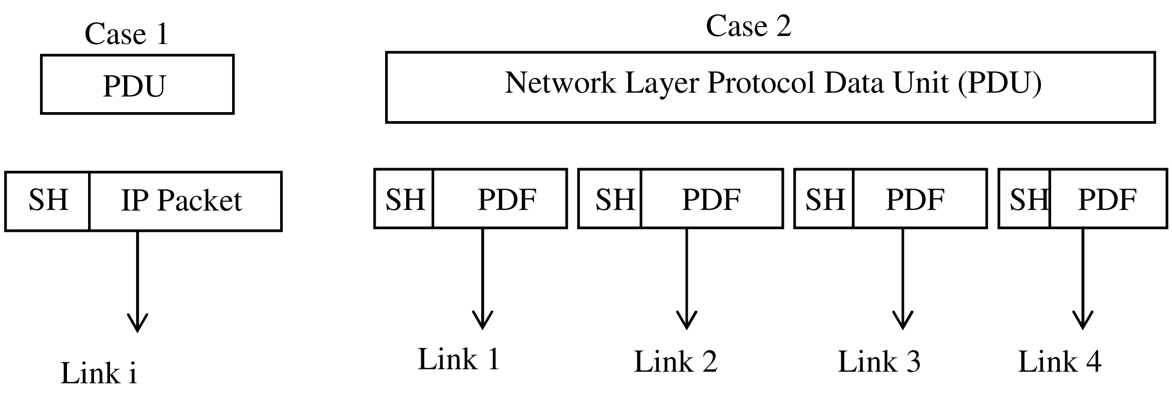
2.2.3 Multilink point-to-point protocol (MLPPP)

Multi-link point-to-point protocol (MLPPP) is a software implementation of packet based inverse multiplexing. As mentioned before, it is an extension to commonly known point-to-point protocol (PPP). It is most commonly used to combine two dial-up links to achieve ISDN like speeds in regions where ISDN/Cable does not exist. As detailed in RFC 1990 [23], “Multilink PPP is a method for splitting, recombining and sequencing datagrams across multiple logical links”.

Multilink is an extension to the original PPP and can be configured during the link establishment phase of a point-to-point connection using LCP option negotiation [23]. It can be used only if both the ends of the connection agree to support multilink. During the negotiation, the end nodes decide the network layer protocols that will be allowed on the multilink bundle as well as the maximum size of the protocol data unit (PDU) that will be sent on the bundle. This is called maximum received reconstructed unit (MRRU). The first link established between the two end nodes creates in a new bundle and the links that are established later are added to the bundle.

When a network layer packet is to be transferred over a multilink channel, it is first encapsulated as per the regular PPP framing procedure [23] described in section 2.2.2. Then, depending upon the number of available links and MRRU size, the data can either be sent as a single frame or can be fragmented in to a number of smaller units that are encapsulated using new PPP header consisting of the multilink protocol identifier and the multilink header [23]. These fragments are then transferred over separate links simultaneously and are recombined at the other end to obtain the original PPP frame. In order to keep track of the order of the fragments, a sequence number is used in the multilink header. While sequence numbers are used to arrange the fragments received out of order, the end and beginning bits in the multilink header indicate the start and end of a frame. The complete header format is described in Appendix B.

For example, consider a MLPPP bundle of 4 links implemented using a Linux system. At any given instant of time during the data transfer, the fragmentation of the protocol data unit depends on the length of the data packet and the number of links that can accept a fragment at that moment. Links can accept a fragment if there are no queued fragments on that link [24].



a. High Load **b. Low Load**

Figure 2.15: Multi-link PPP operation

Figure 2.15a depicts a heavy load situation where packets are sent as single fragments with segment header (SH) on the next available link, there by reducing the fragmentation overhead. On the other hand figure 2.15b show a light load situation where all the links are able to accept a fragment, in which case, an incoming protocol data unit is fragmented into 4 smaller fragments and sent simultaneously over all the links thereby reducing the latency.

Chapter 3

System Design

3.1 System Design Requirements

The design requirements of the system are as follows.

1. The multi-channel implementation should maximize the throughput.
2. To support real-time interactions, the system should minimize the end-to-end delay.
3. The overall system should be reliable and have autonomous operation so as to handle call drops and system/power failures in remote field deployment.
4. Since the typical field implementation of the system will be in harsh polar climates, the number of hardware components equaling number of (POFs) points of failures should be minimized.
5. The overall system should be compact, lightweight and support mobile platforms.

3.2 System Design

The design of a multi-channel Iridium system to meet the above mentioned requirements is discussed in this section. The block diagram of multi-channel Iridium based point-to-point data communication system is shown in figure 3.1. The overall system can be divided into three sections: Remote subsystem, Logical link and the

Local subsystem. The individual components of each subsystem will be discussed in detail in chapter 4.

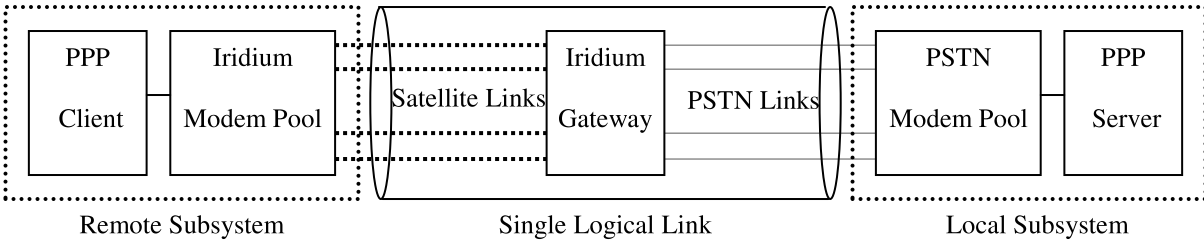


Figure 3.1 Multi-channel data communication system

The system is configured in Iridium-to-PSTN mode (as explained in section 2.1.9) wherein the remote end uses Iridium satellite modems and the local side uses regular PSTN modems as seen in figure 3.1. Remote data is transmitted through Iridium satellite modem to the Iridium gateway via the satellite network. The gateway then routes it through the regular PSTN network to the receiving modem connected to a PSTN phone line. As the data traverses the satellite-earth segment four times per round trip, the round trip delay is less than the Iridium-to-Iridium mode where data traverses the satellite-earth segment eight times per round trip. Since the data traverses less number of satellites, the inter-satellite processing delay is reduced in the Iridium-to-PSTN mode.

The remote end is configured as a PPP client and the mainland end as a PPP server. The link parameters are optimized for Iridium system and will be discussed in detail in section 4.3. Each Iridium modems dials a PSTN modem at the mainland facility to establish a point-to-point link operating at 2.4 Kbps. The multiple point-to-point links are logically combined using multi-link point-to-point protocol (as discussed in section 2.2.3) to form a single logical channel of aggregate bandwidth.

3.3 Protocol Stack

Consider the protocol stack of the two end systems as shown in figure 3.2.

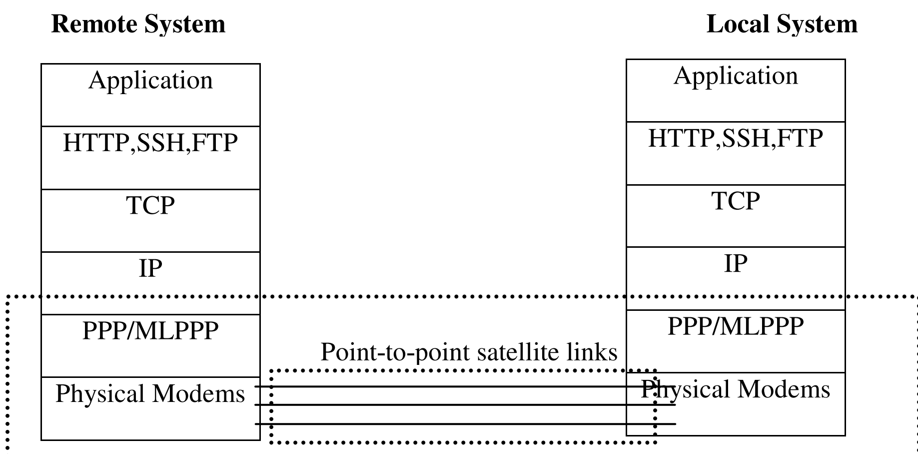


Figure 3.2 Protocol stack of the remote and local system

The IP packets of a single application from the remote system are fragmented at the MLPPP layer into smaller segments depending upon the packet size, the link MTU and the availability of the links (section 2.2.3). These segments are then sent simultaneously over multiple satellite links. The MLPPP layer at local system on the other end combines the received segments into the original data packet and checks for errors or segment loss. If the packet is successfully reconstructed, it is presented to the IP layer; else a packet error is reported. Any errors/packet losses that occur are handled by TCP/IP layer in a standard manner. The same procedure is followed for packets going in the other direction (local system to remote system).

The overall process of splitting the datagrams over multiple channels is abstracted from the IP layer. From the upper layers (TCP/IP) point of view the communication channel looks like a single PPP link with aggregate bandwidth and one interface (PPP0).

3.4 Network Architecture

In this section we discuss the end-to-end network architecture that could be used to provide data and Internet access to Polar field research camps/sites.

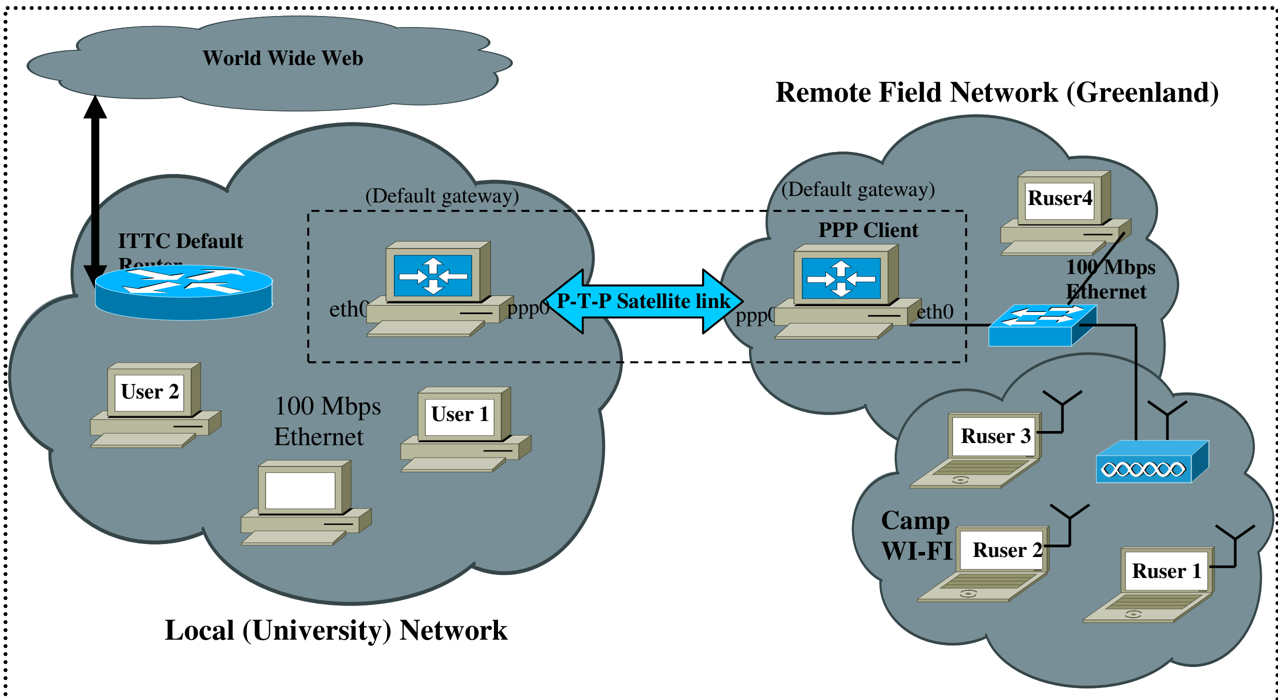


Figure 3.3 Network architecture

A specific implementation of the system between the Polar (Greenland) Field camp and University of Kansas (as shown in figure 3.3) is considered to illustrate the network architecture. This architecture could be generalized to provide Internet access to any remote field site from a mainland facility or to provide data connection between two remote field sites.

The remote field site is configured as a subnet of the local network; University of Kansas in the above example. The PPP client (system with multi-channel Iridium system) is configured as the default gateway of the Greenland subnet. Data from wired (Ruser4) and wireless users (Ruser1-Ruser3) is routed by the PPP client over the satellite link (through the PPP0 interface) to the PPP server at the University of Kansas. PPP server being a part of the university's network routes the data packets through the University router to the World Wide Web.

Similarly, PPP server is configured as the default gateway to forward packets going from the University network and Internet to the remote (Greenland) network over the satellite channel. A static route on the University router forwards all the traffic intended for the remote subnet to the PPP server.

The advantages of this type of architecture are:

1. It provides Internet access to the remote site with the University's existing Internet connection. No additional ISP services are needed.
2. Since the remote site is basically an extended subnet of the University network, field users can access the University resources including databases or file systems.

3. No additional hardware (like routers, etc) is needed. This increases the reliability of the system as the number of points of failure is reduced.
4. The link from the field site to the university is a direct point-to-point connection and hence has the lease delay possible. This architecture is particularly useful for Polar field experiments, where most of the data transfer/communication is between the field site and the local facility.

Chapter 4

System Implementation

This chapter presents the implementation of a 4-channel Iridium communication system based on the design discussed in chapter 3. This system was implemented and tested at NGRIP, Greenland in summer 2003. While the current chapter explains the implementation, tests and results will be discussed in chapter 5.

4.1 Four channel Iridium communication System

The functional diagram of the implemented system is shown in figure 4.1.

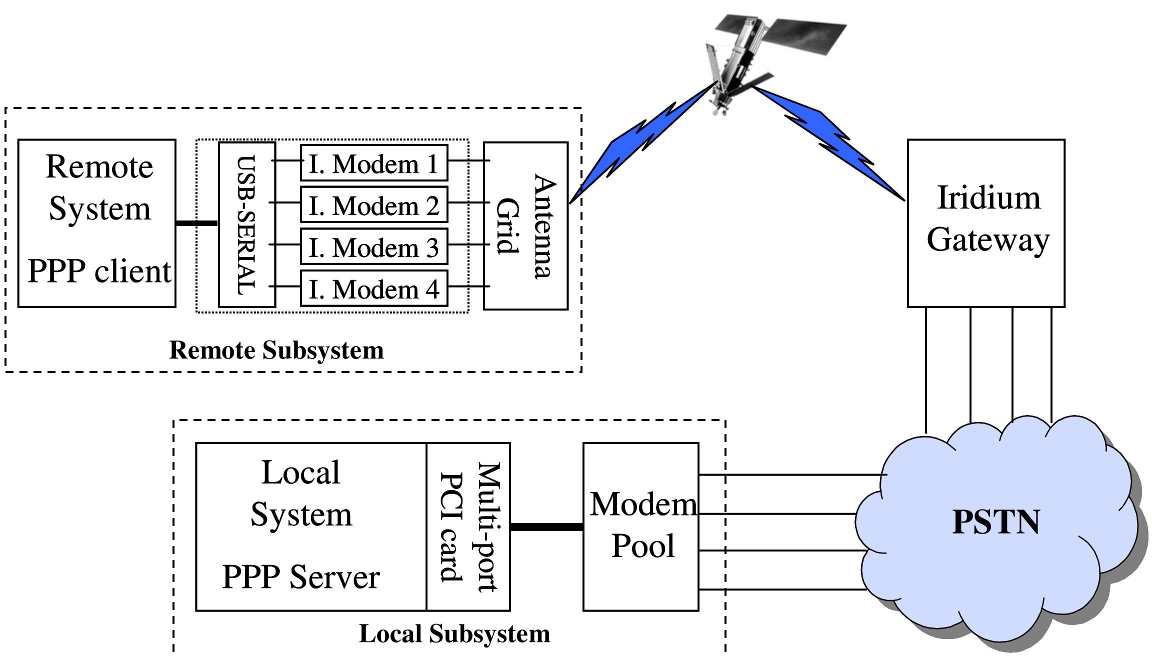


Figure 4.1 Four-channel Iridium communication system

The remote subsystem system consists of 4 Motorola-Iridium modems connected to a rugged laptop with an USB-to-Serial converter (figure 4.2 a).

(a) (b)

Figure 4.2 Four-channel Iridium communication unit and its antenna array

The antennas of these modems are installed on a metal plate of 1 sq ft that forms the ground plane. These four antennas are then mounted on a frame such that they are separated from each other by 2 ft in order to reduce the effects of interference (figure 4.2b). The PPP daemon (PPPD) on the remote terminal is configured as a PPP client so as to connect to the mainland terminal. The computer terminal at the mainland has four PSTN modems connected to it via a multi port serial card and an octopus cable. This terminal is configured as a PPP server to receive call from the remote Iridium system through 4 PSTN phone lines.

The developed link management software configures the remote terminal as a PPP client, dials the Iridium modem to establish the serial connection with the mainland terminal and handles PPP negotiation to complete a point-to-point data connection.

Once the basic connection is established, it dials the remaining modems and seamlessly attaches them to the first PPP connection, forming a single high bandwidth pipe. Standard Internet protocols, TCP/IP are then used to provide end-to-end connectivity. This MLPPP system developed in Linux also monitors the satellite connection, detects any call drops during satellite hand-offs and immediately reconnects dropped modems. Further, it handles power failures and system resets providing a reliable and fully autonomous data link. The following sections detail how each of the above functionalities is achieved.

4.2 Modem Control

As seen in figure 4.1, the remote unit uses Motorola-Iridium modems while the mainland facility uses regular PSTN modems. The PSTN modems are controlled using the MGETTY [25] software tool in Linux. Mgetty is configured such that it locks the serial lines to which the modems are connected, initializes the modems by running the init scripts from the configuration file and awaits the arrival of any incoming call. The modem is continuously monitored by Mgetty to ensure the readiness of the modem to receive an incoming call.

On the other hand, the Motorola-Iridium modems do not have Linux drivers. But, they do respond to modified Hayes AT command set. In order to communicate and control these modems, system scripts are written using the Linux CHAT facility. The CHAT scripts written with AT commands (as given in Appendix C) can be used to communicate with the modem to obtain its serial number, signal level measurements and request data service from the Iridium network. Once a frequency and time slot is acquired from the Iridium network, the chat script dials the PSTN number of the receiving modem and completes the serial connection when the other end accepts the call. Following a successful call, it hands over the line control to the PPP daemon.

4.3 PPP link parameters and user authentication

Multi-link point-to-point protocol is generally used to combine two dial-up links to obtain ISDN like speed or to combine multiple T1 lines to obtain T3 speeds [26]. These terrestrial links have very short and fixed delays. On the other hand, Iridium being a satellite network with intelligent, inter-satellite switching has a huge end-to-end delay which varies with time, weather, location and packet size. Due to this large and inherently variant delay, PPP link negotiation often fails leading to increased number of unsuccessful connection attempts. Hence, the link parameters should be fine tuned for this system. A number of simulations and experiments were conducted to determine the optimum link parameters for this system. PPP provides a number of options through the PPP daemon to fine tune the link. The PPP daemon (PPPD) parameters of primary concern are:

1. ***lcp-restart***: It sets the Link Control Protocol (LCP) restart interval (retransmission timeout). A value of 10 insures that the during the link establishment phase, the system waits 10 seconds between LCP packet retransmissions. This avoids the confusion that arises when client restarts the

LCP negotiation and then receives the delayed response of the server to the previous LCP negotiation packets; a situation often leading to link failure.

2. ***pap-restart***: It sets the password authentication protocol (PAP) restart interval (retransmission timeout). Again, a value of 10 ensures proper operation with Iridium system. A small value on this parameter leads to number of retransmissions and eventually failure of the link.

3. ***lcp-max-configure***: It sets the maximum number of LCP configure-request transmission. This determines the number of times the system transmits configure-request packets before giving up on the link. A value of 10 is best suited. A very high value of this parameter could potentially lock the system when there is a physical link failure. The system would keep trying to establish a failed link for a long time.

4. ***connect-delay***: This option implemented on the remote end means: wait for up to n milliseconds after the connect script finishes the establishment of the serial link to start the PPP negotiation on that link. After this time is elapsed or on the receipt of a valid PPP packet from the peer, the remote system will send its first LCP configure-request packet. A value of 5000 milliseconds gives enough time for the server to transfer the line control from Mgetty to the PPP daemon, check the link speed and get ready to handle the PPP packets from the client.

5. ***lcp-echo-interval***: This option is used on both sides of the link to monitor the status of the link. It sets the time interval between sending lcp echo-request

packets. When a PPP client/server receives this packet, it replies with echo-reply packet. This helps to detect failed links in case the hardware flow control does not detect the link failure. A value of 30 seconds does not load the link considerable and at the same time is effective.

6. ***lcp-echo-failure***: This option goes along with the lcp-echo-interval parameter and sets the number of consecutive lcp-echo-request packets the system should send without receiving a echo-reply to assume the link to be dead. A value of 2 quickly detects a failed link so that it can be redialed.

User authentication is implemented by login/password mechanism. To avoid passing of text passwords and to automate the authentication procedure, password-authentication-protocol (PAP) supported by PPP is used. The permissible users and their passwords are stored in pap-secrets file of the PPP server. During the link establishment phase, the PPP client passes the username and password to the server in an encrypted from. If a match is found, the authentication is successful; else the link is immediately terminated.

4.4 Client-Server Configuration

This section deals with configuration of the remote machine as a PPP client and local system as a PPP sever. The PPP client configuration of the remote system and its operation is illustrated with the software flow control diagram in figure 4.3.

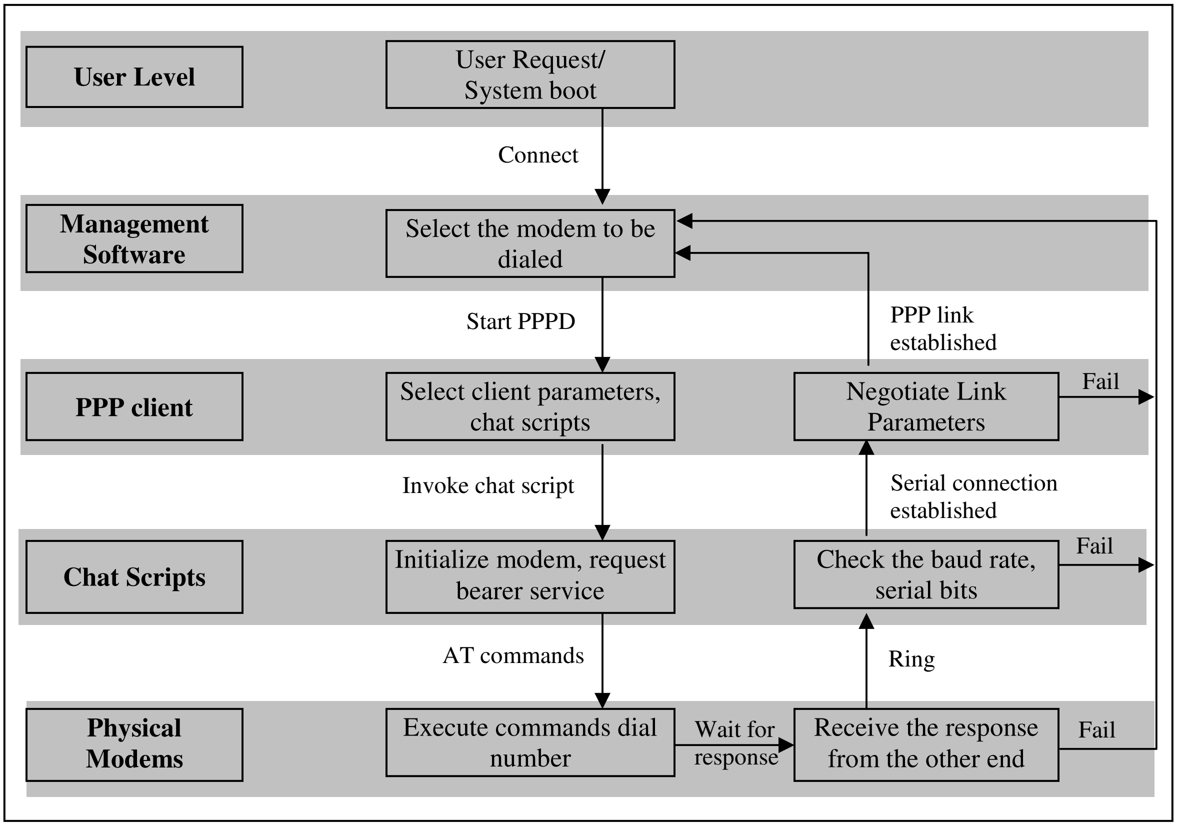


Figure 4.3 Software flow control diagram of the remote system

The remote system in the field is configured as a PPP Client and the management software controls the system operation. Upon user request or system boot up, the management software is started. The management script first selects a single Iridium modem and initializes PPP daemon. The PPP client script then configures serial link speed, multilink operation, flow control, link parameters and invokes the appropriate chat script on modem. The chat script reads the chat-configuration file and initializes the modem using AT commands. Finally, modem dials the destination number and waits for the other end to respond. When the other side of the link accepts the incoming call, modem sends a “Ring” signal to the chat script, which then checks for the link speed. If the speed of the link is 19200 baud then it transfers the control of the link to the PPP daemon; else the connection is terminated and the management script takes over the control. After a successful serial connection, the PPP daemon negotiates (with the peer on the other end) PPP link parameters including MTU, multilink operation, MRRU, etc. These parameters are initialized in the configuration file of the dialing modem. This is followed by user authentication, as discussed in section 4.2. The PPP link is established and the PPP0 interface is created if the user authentication is successful. A failure at any stage would transfer the control to the management script. Once the first link is established, the management script repeats the procedure on the rest of the links and attaches them to the first link, forming a single logical connection of aggregate bandwidth.

The configuration of the local system is shown in the software flow diagram (figure 4.4). In this case the operation is similar to that of client configuration with few important differences. The modem control is handled by Mgetty software tool instead of chat scripts and the PPP daemon is configured as a server instead of client. A link failure during any stage results in the transfer of control to the Mgetty.

4.5 Autonomous System Operation

As discussed in section 2.1.2, it is inherent in a LEO satellite system like Iridium to experience frequent call drops. In order to have a reliable and seamless operation for remote deployments it is essential to a have an autonomous system that can recover form call drops and system/power failures. It should be noted that if a call drop occurs on any modem except the first one, the overall connection still remains up with reduced bandwidth. The dropped line is redialed and attached back to the bundle. On the other hand if the call drop occurs on the first modem to which the interface PPP0 is attached, the entire session fails and the system must be completely rest. The management software discussed in the previous sections is also programmed to perform this operation. In order to understand the functionality, consider the flow control diagram of connection setup as shown in figure 4.5

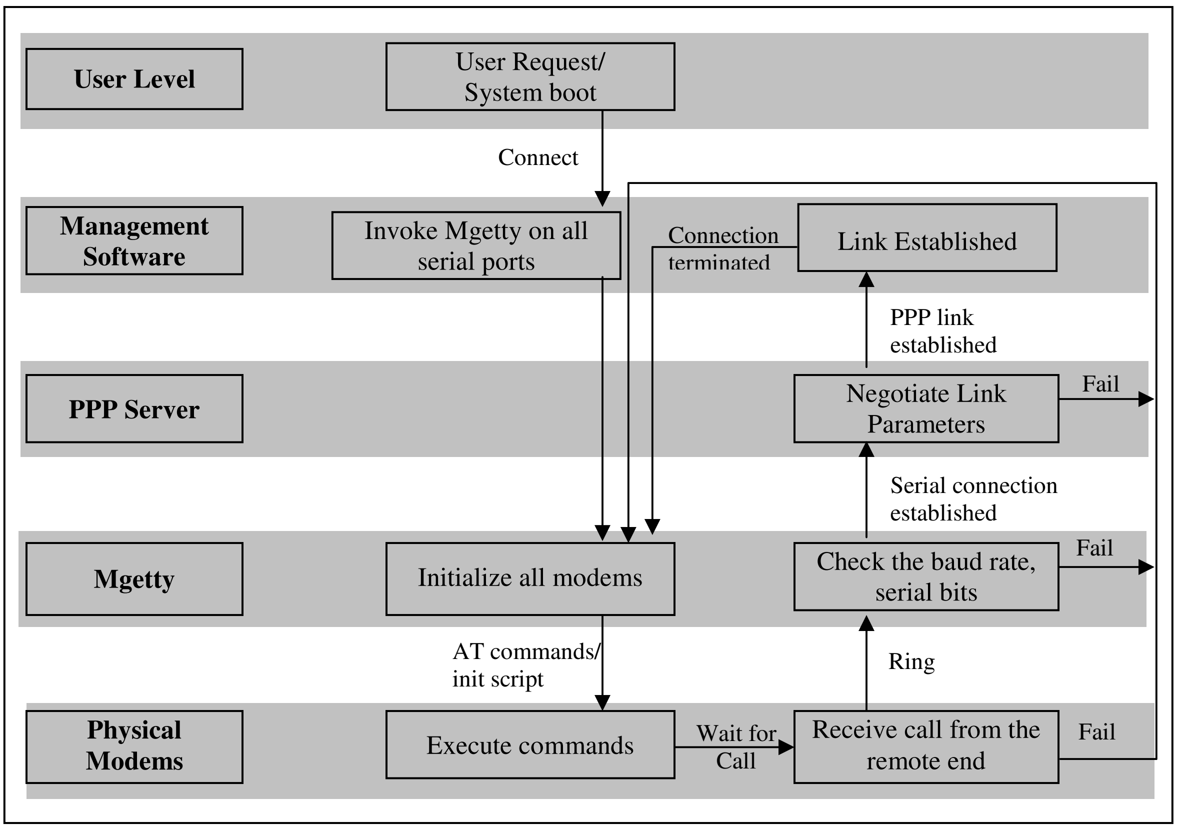


Figure 4.4 Software flow control of the local system (server)

The management script establishes a connection with the other end using the first modem. Only after the first modem connection is connected and an interface formed, other modems are dialed. The links established thereafter are attached to the first link forming a multi-link bundle. Each connection is monitored individually by the script.

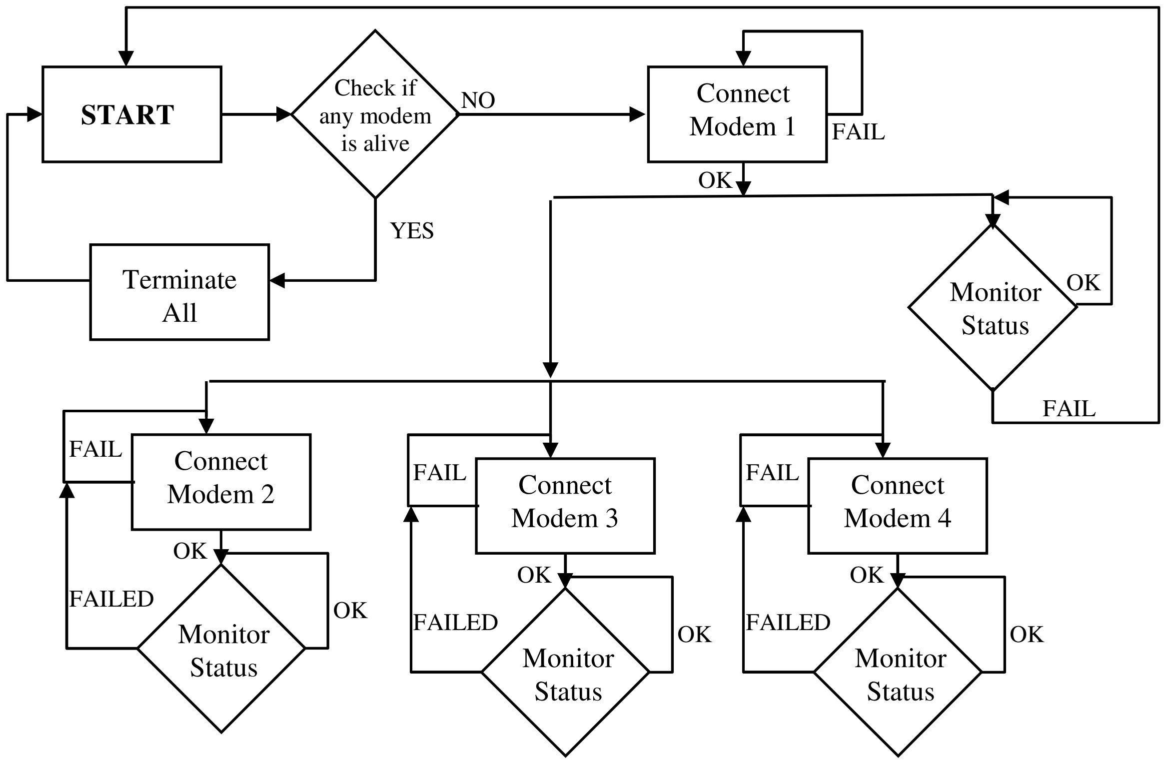


Figure 4.5 Modem flow control diagram

A call drop is detected if modem sends a SIGHUP signal. Further if the system does not receive a reply to a predetermined number of consecutive echo-request packets, the link is assumed to be dead. In figure 4.5, when a call drop on the modems 2-4 is detected, the individual dropped link is redialed and attached to the bundle restoring the full capacity of the system. On the other hand, when Modem 1 experiences a call drop, the session is reset, all links are terminated and a new connection is started. Finally, the system responds to a power/system failure as a call drop. The failed link is redialed; if a power failure occurs, the connection is restarted. Thus the management software ensures a highly reliable and fully autonomous operation.

Chapter 5

Field Tests and Results

A 4-channel Iridium communication system (discussed in chapter 4) was implemented and tested in the field at NGRIP, Greenland (75^0N, 42^0 W) from June 23 to July 17, 2003. This chapter presents the field experiments conducted, analysis of the data collected and the results obtained.

5.1 Overview of field experiments

The developed 4-channel Iridium communication system was integrated and packaged in a rack mountable unit and installed on the field site (Polar camp) as shown in figure 5.1. In order to characterize the call drops experienced due to the behavior of the Iridium system alone (section 2.1.2), the antenna array was mounted on a scaffold such that it has a clear 180^0 view of the sky as show in figure 5.2. This eliminates call drops that may occur due to obstructions like tents etc.

The overall objective of the field experiments was to determine the reliability and the performance of the communication system. This involves determining the qualitative performance as well as specific performance metrics like system end-to-end delay, packet loss, throughput, system efficiency, call drop rate, call drop interval, etc. Analysis of the collected data is performed to characterize the system behavior.

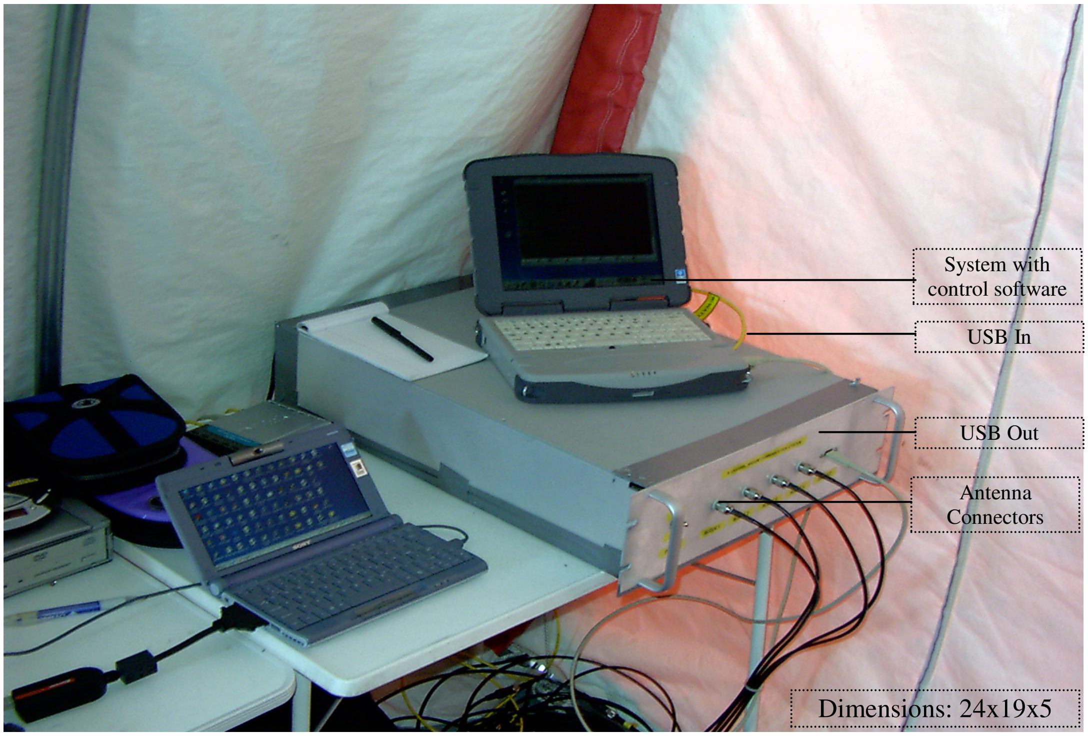


Figure 5.1 Field implementation of multichannel Iridium communication system

Since the system would eventually be implemented on an autonomous vehicle, field experiments were conducted on mobile platforms as shown in figure 5.3 to obtain the performance of the system in such situations. TCP/IP performance over satellite links has been extensively studied [28] to [31]. We conducted tests to evaluate the performance of the TCP/IP protocol suite over Iridium system in general and over multi-channel Iridium system in particular.

The field experiments and data analysis has been classified into:

1. General system performance
2. System reliability
3. TCP/IP performance over Iridium
4. Mobile system performance

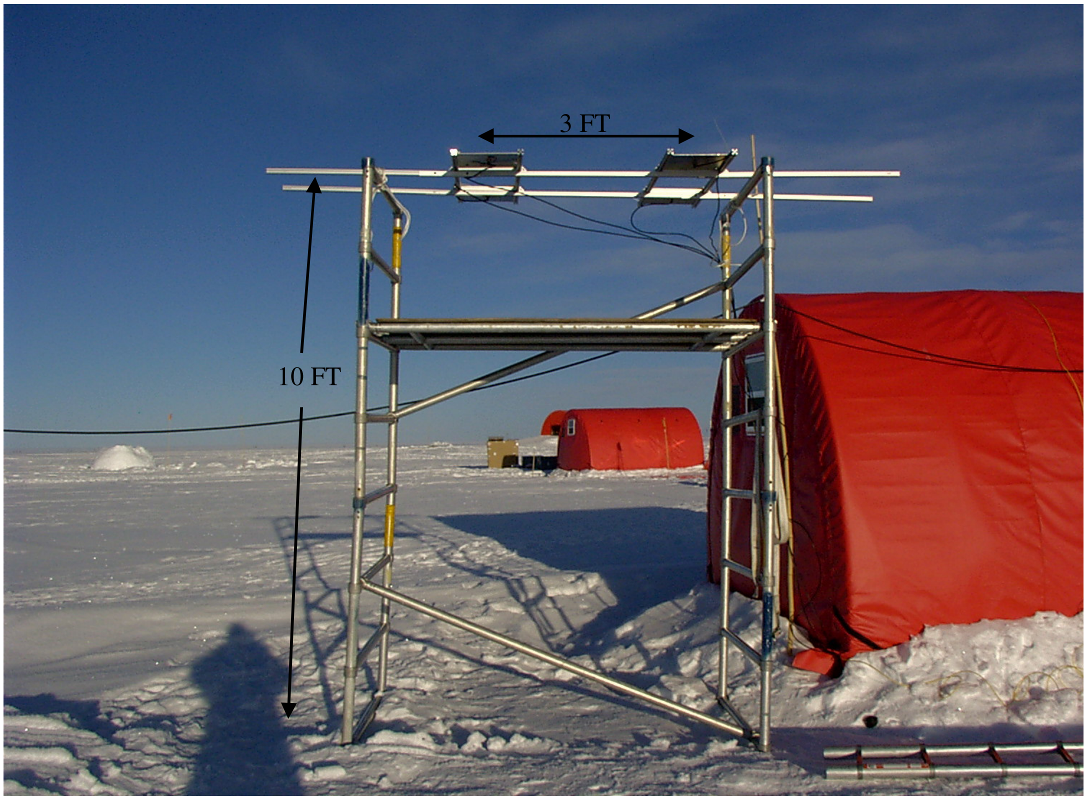


Figure 5.2 Field antenna setup

Figure 5.3 Mobile performance tests

5.2 General System Performance

The performance metrics in this section are: end-to-end delay, packet loss and throughput of the system.

5.2.1 System Delay and Loss

End-to-end delay involves transmission time, propagation time, switching time (on satellites) and the router queuing time. It depends on a number of factors including the data packet size, link capacity, distance traveled, number of intermediate hops, etc. As seen in the theoretical calculation of the Iridium system delay presented in section 2.1.4, the important parameter needed to calculate the end-to-end delay is the number of satellite links traversed by the data packet. Since this is not directly available, an alternative method to calculate the theoretical end-to-end delay between the Greenland and the University of Kansas is given below.

Consider the end-to-end delay observed by a 64 byte packet between Greenland and University of Kansas.

Data Route: Satellite uplink (from Greenland) – Iridium satellite channel – downlink at Hawaii gateway – PSTN link at Kansas.

Distance traveled = 800+8000+800+6000 = 15600 Km

Propagation time = distance traveled/speed of light = 15600 Km/ (3e5)Km/sec= 52msec

Transmission time for a 64 bytes@2.4Kbps = 64*8/2400=213msec

Unknown parameters = inter satellite switching time + switching time at gateway

Theoretical end-to-end delay = 265 msec + unknown value

The PING tool in Linux was used to send ICMP packets of 64 bytes from one end. The other end responds to a ping packet by sending a ping reply packet of the same size. Further, any packet losses that occur are also recorded.

Table 5.1 shows the round trip time and percentage loss for 64 byte ping packets. The test was repeated number of times with varying number of data packets.

Packets Sent	Packets Received	% Loss	RTT (sec)			
			Avg	Min	Max	Mdev
50	100	0	1.835	1.347	4.127	0.798
100	100	0	1.785	1.448	4.056	0.573
100	100	0	2.067	1.313	6.255	1.272
200	200	0	1.815	1.333	6.228	0.809

Table 5.1 Round trip time (RTT) of Iridium data link

It is seen that the average round trip delay is 1.8 seconds. The end-to-end delay, which is half of round trip time, is 0.9 seconds. But, the theoretical delay was calculated as 265 msec. The additional delay could be attributed to the inter-satellite switching and processing at the gateway. Due to unavailability of information on the details of inter-satellite switching and processing at gateway, the observed delay cannot be accounted for. Further, it should be noted that the system RTT has very large mean deviation, which means that the delay will vary by large amounts. Reason for this random variation in delay includes:

a. Changing distance between the fixed user and the moving satellites. Further the distance of ISL's (inter-satellite links) through which the data traverses also changes with time.

b. Call hand-off resulting in new satellite links (hops) that route the call. The new route could have different propagation and queuing delay, latter being caused due to non-uniform traffic distribution.

This random variation in the delay could lead to performance issues with TCP/IP protocol and will be discussed later in this chapter. No packet losses were observed for small data transfers. Hence the system has low bit error rate and the end-to-end system is reliable. Figure 5.4 shows the round trip delay observed during different periods of the day and night.

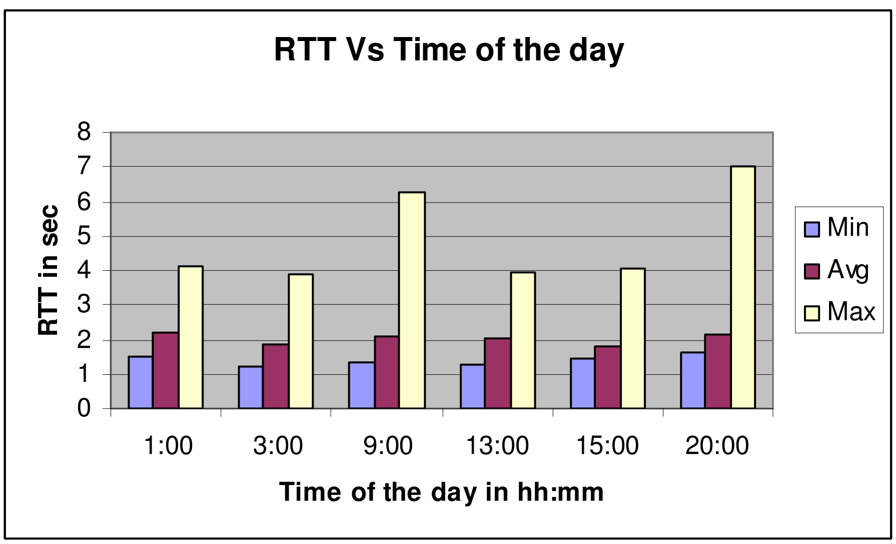


Figure 5.4 Round trip time vs. time of the day

It is seen that the Round trip time not only has a high mean deviation but also varies with time of the day. Since it does not follow any pattern, it may be stated that the variation of delay is random with respect to time. In order to determine the variation of the delay with packet size, ping measurements were done with varying packet size. While figure 5.5 shows the variation of delay with packet size, the normalized delay in msec/byte is shown in figure 5.6.

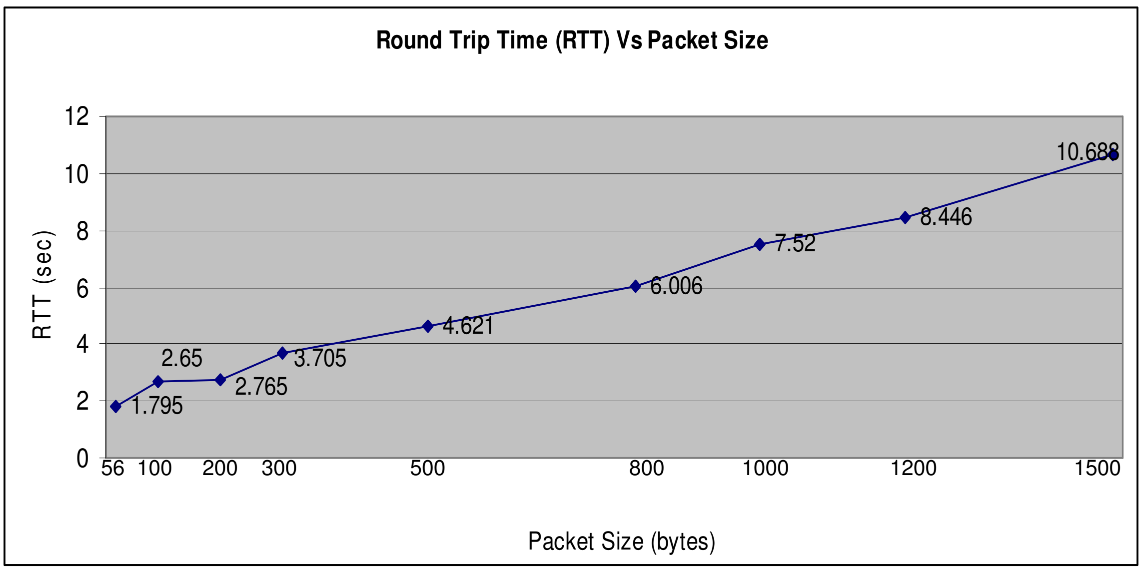


Figure 5.5 Variation of RTT with respect to packet size

It is seen that the delay increases almost linearly with packet size except for low packet sizes. This is clearly evident from the normalized RTT where the msec/byte is essentially constant for packets sizes beyond 800 bytes.

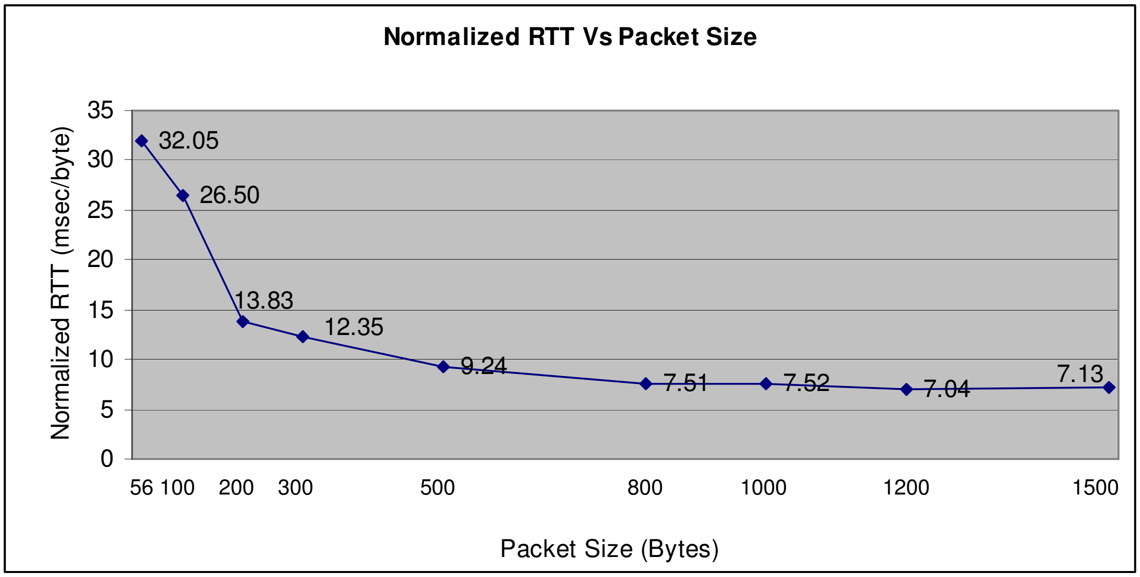


Figure 5.6 Variation of normalized RTT with respect to packet size

5.2.2 Throughput

Iridium system has an average link capacity of 2.4 Kbps with instantaneous capacity varying from 2.2 to 3.2 Kbps as explained in section 2.1.4. The throughput is usually less than the link capacity (physical bits/sec) due to overhead such as TCP/IP headers, PPP/MLPP headers, data fragmentation and encoding.

$$Throughput = \frac{Data\ Paylaod}{Time\ Taken}$$

$$Efficiency\ \eta = \frac{Throughput}{Link\ Capacity}$$

In order to determine the throughput of the system, a fixed amount of data is transferred between the two systems using a reliable transport mechanism and the time taken is recorded. In our field experiments we used TCP/IP as the underlying reliable transport mechanism. Linux tools IPERF [33] and TTCP [34] were used to determine the throughput. Further, FTP file transfers were also used to determine the effective long time system efficiency.

Method	1 Modem	2 Modems	3 Modems	4 Modems
Iperf	2.1	4.0	7.0	9.6
Iperf	1.9	3.9	7.0	9.3
Iperf	1.7	4.5	6.8	9.7
Ttcp	2.29	4.43	6.6	8.9
Ttcp	2.48	4.40	7.0	8.78
Average	2.1	4.25	6.88	9.26

Table 5.2 Variation of throughput with number of modems

Table 5.2 summarizes the throughput obtained with one, two, three and four modems. The measurements were done both using IPERF and TTCP. It is observed that the throughput also varies to some extent with time, which can be attributed to the random delay variation as discussed in the previous section (5.2.1).

A plot of throughput vs. number of modems is shown in figure 5.7. The throughput efficiency of the multi-channel Iridium communication system is observed to be over 90%, considering the average throughput of an Iridium link to be 2.4 Kbps. Thus using 4 modems, we were able to achieve a throughput of 9.26 Kbps which was used to support transfer of data to and from the field camp.

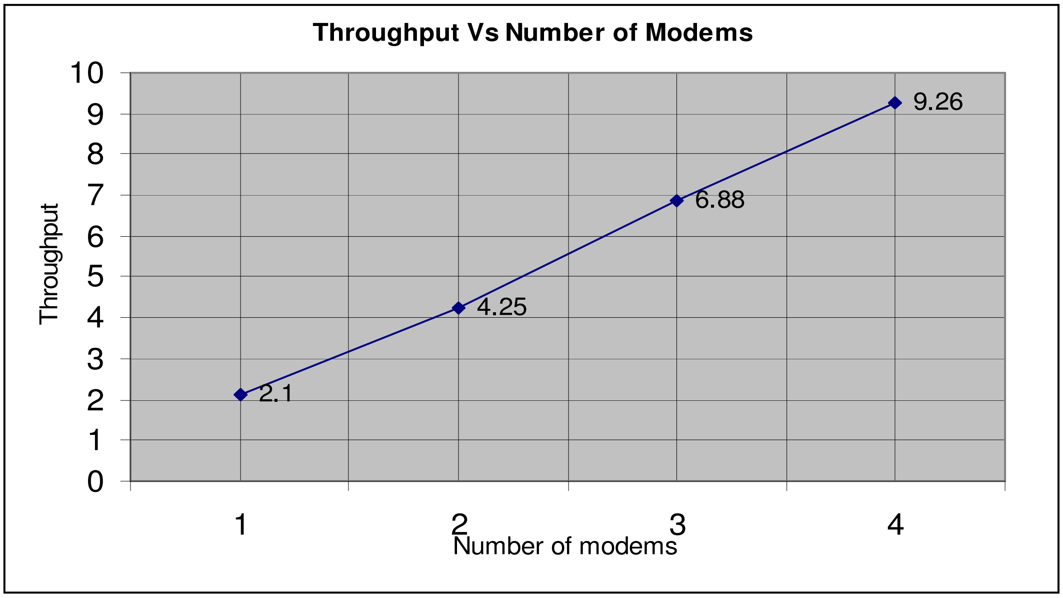


Figure 5.7 Plot of Throughput vs. Number of modems

File Size (MB)	Upload Time (min)	Throughput (bits/sec)
0.75	11	9091
3.2	60	7111
1.6	23	9275
2.3	45	6815
1.5	28	7143
2.5	35	9524

Table 5.3 Effective throughput for large file transfers

Table 5.3 shows the effective throughput obtained while transferring large video files ranging from 0.75 to 3.2 MB from field to the University of Kansas using FTP. Modem call drops resulted in the throughput being less than expected in some cases. However, it is important to note that these large files were successfully transferred automatically even in the presence of call drops, indicating that the new link management software properly executed. Further, this demonstrates that the developed system can support the data requirements of field research.

5.3 System Reliability

As discussed in section 2.1.2, a low earth orbiting satellite system like Iridium experiences more number of call drops as compared to a GEO or a terrestrial cell system. In order to determine the overall reliability of the system it is essential to determine the call drop pattern. Initial studies on Iridium call drops [27], [2] have reported a call drop rate of 6-18% based on call duration of 10-15 minutes. This means 6-18% of the calls were dropped with in the first 15 minutes of the call. But, it should be noted that if the remaining 94-82% of the calls were to be continued for long periods of time, it is likely that they would eventually experience a call drop. Hence the performance criterion is not how many calls are dropped, but the interval between the call drops.

To determine the reliability of the system we conducted two 24-hour tests. During these tests the management software controlled the 4-channel communication system. The management software detects any call drops/link failures, logs the event, automatically redials the dropped link and attaches it to the multilink bundle.

July 10, 2003 – 24 hour test

Figure 5.8 shows the call drop pattern on the first modem, which defines the basic connection itself. Since a call drop on the first modem results in the termination of the entire session, these call drops represent the system interruptions. The time interval between the successive call drops is given in Table 5.4. During the 24 hour test 14 call drops were recorded. The average connection time between the call drops was observed to be approximately 100 minutes. The overall percentage up time on the first modem was about 96%. The longest up time without a call drop was observed as 618 minutes.

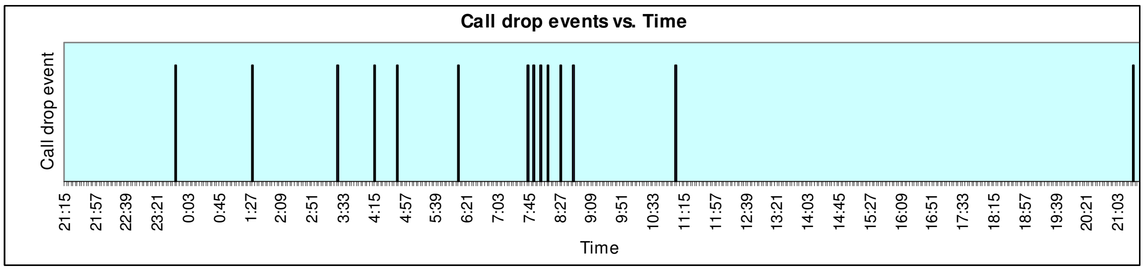


Figure 5.8 Call drop pattern of the first modem during the 10th July 24 hour test

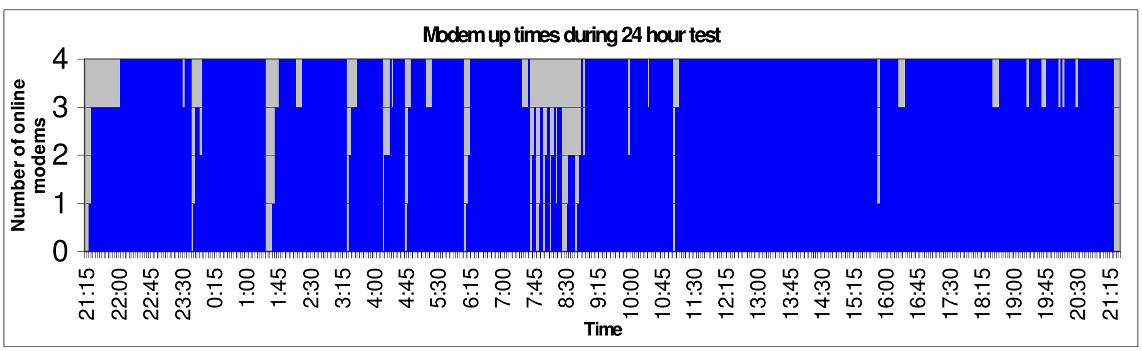


Figure 5.9 Availability of modems during the 10th July 24 hour tests

Time interval between call drops	146	106	114	50	25	84	89	8	7	7	17	11	137	618

Table 5.4 Time interval between successive call drops during the 10th July 24hour test

It should be noted that a drop on the first modem results in a complete loss of the communication link, whereas call drops on the other 3 modems result in a brief reduction in the available bandwidth before the management software reconnects the dropped link and attaches it to the bundle. In order to determine the available bandwidth during the same 24-hour test period, the number of online modems vs. time is plotted in figure 5.9. The statistics obtained from this graph are shown in Table 5.5.

Number of online modems	Up time (min)	% Up time
All the 4 modems	1161	80.6
At least 3 modems	1323	91.8
At least 2 modems	1365	94.7
At least 1 modem	1395	96.8

Table 5.5 Statistics of the July 10, 2003 24 hour test

It is seen that during 80% of the test time all the 4 modems were running providing full bandwidth of 9.6 Kbps. On the other hand we had at least one modem connected for 96% of the time.

July 12, 2003 – 24 hour test

A second reliability test was conducted for another 24 hours. The system operation and test procedure was same as that of first test. The call drop pattern is shown in figure 5.10 and the time interval between successive call drops is given in table 5.7. There were 16 call drops on the first modem during the 24 hr test period. The average time interval between call drops is observed to be 87.5 minutes and the maximum interval without a call drop was 386 minutes. The percentage uptime of the first modem was 96%.

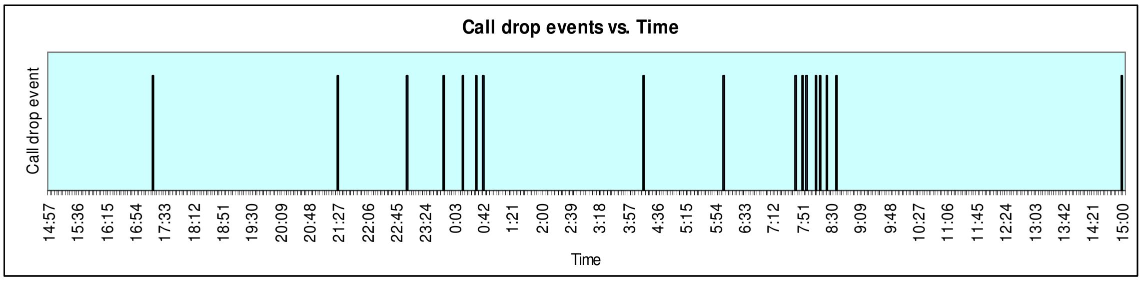


Figure 5.10 Call drop pattern of the first modem during the 12th July 24 hr test

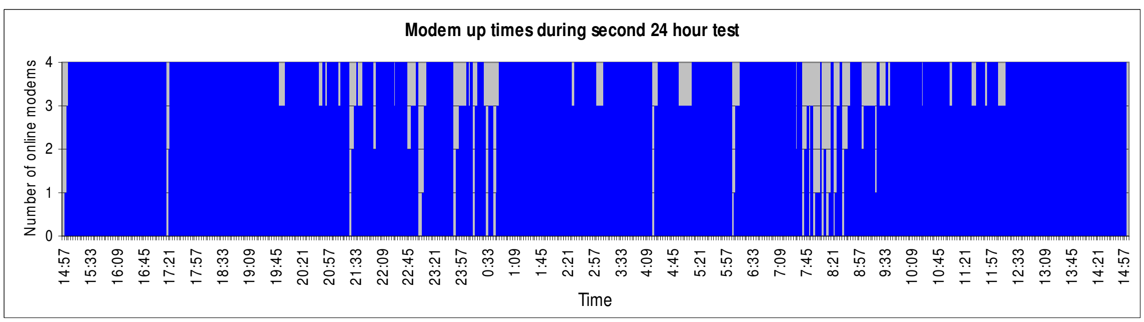


Figure 5.11 Availability of modems during the 12th July 24 hour tests

Time interval between call drops	135	248	93	40	26	16	8	211	108	91	8	5	6	5	8	7	386

Table 5.6 Time interval between successive call drops in the 12th July 24 hr test

Figure 5.11 shows the plot of number of online modems vs. time during this test period. The statistics from the graph are summarized in table 5.7. For 80% of the test time, all the 4 modems were up providing full capacity.

Number of online modems	Up time (min)	% Up time
All the 4 modems	1161	80
At least 3 modems	1323	92
At least 2 modems	1365	95
At least 1 modem	1401	96

Table 5.7 Statistics of the 12th July 24 hour test

5.4 Performance of transmission control protocol over Iridium

Transmission Control Protocol (TCP) is the most common and widely used transport layer protocol for reliable data transfer. Several applications like FTP, HTTP, SSH, etc, are based on TCP. Since its invention in 1974, TCP has been modified to support various networks and evolving applications. TCP has developed from TCP Tahoe to TCP Reno to TCP new Reno to TCP SACK, each with modifications geared towards the improvement of the overall performance. In order to provide reliable data communication over satellite links, a number of TCP modifications have been suggested.

5.4.1 Related Work

Performance of TCP over satellite links has previously been studied through simulations and emulations. Typical studies [28] to [31] involve modeling of the satellite system, communication channel, effect of urban surrounding and traffic load on the system to determine the various TCP performance metrics.

Alman [28] compared the slow start mechanism of TCP between a high bandwidth satellite link with 560ms RTT and terrestrial link with 80 msec RTT. It was seen that in the time taken for the TCP to reach the maximum congestion window size in a satellite network, TCP could send 22 times more data in a terrestrial network. Similarly in the event of a packet loss, the congestion avoidance algorithm takes longer time to increase the cwnd (congestion window) as compared to a terrestrial network.

The performance of various TCP versions over LEO satellite links [31] showed that for packet error rate up to 0.001% different versions of TCP had similar performance. For higher packet error rates, TCP SACK was observed to have better performance as compared to TCP Tahoe.

Sangal [30] evaluated the effect of variation in RTT variation on the performance of TCP in a LEO network. He reported that satellite hand-off in a LEO network could lead to sudden changes in the RTT due to variation in the propagation delay caused by changing distance between the satellites and the user terminals. Further, following a handoff, the queuing delay could vary due to an increase or decrease in the number of users being serviced by the new satellites involved in routing of the packet (traffic non-uniformity). In summary, a sudden variation in RTT could lead to a timeout causing unnecessary retransmissions and for connections with large window size, sudden variations in RTT could lead to generation of duplicate ACKs and subsequent retransmissions.

While simulations are useful to evaluate the network performance parameters like capacity and fault tolerance, emulations on the other hand help in testing the actual protocols over the satellite network. Both the method, however, make a number of assumptions that may not be accurate. Further, they may not incorporate all the elements of the system that contribute towards its performance. Performance tests conducted on the actual satellite systems, though costly, can provide insight in to the behavior of protocols accurately.

5.4.2 Overview of TCP performance tests

The experiment setup consisted of two systems running RedHat Linux with Kernel-2.4.18 connected by a point-to-point satellite channel. The TCP version used in this Kernel is TCP SACK. Since the experiment was conducted on a point-to-point satellite link, the ambiguity associated with Internet links is eliminated. Further, field tests were conducted at NGRIP, Greenland, where there are no significant obstructions and hence no shadowing/urban canyon effect.

In section 5.2, we saw that the throughput of a single Iridium link is 2.4 Kbps and the average RTT for a 1500 byte packet is 11 seconds.

Bandwidth Delay Product (BDP) = 2400*11/8 = 3300 bytes = 3 segments.

For the maximum end-to-end delay of say 20 seconds, the BDP is 6000 bytes = 4 segments. Though, the system has a high end-to-end delay, due to extremely low bandwidth, the BDP is small.

The performance analysis of TCP over Iridium is decomposed as follows:

1. Performance of TCP over Iridium system in general.
2. Efficiency of the multi-channel satellite (Iridium) communication system
3. Degradation in TCP performance due to packet loss
4. Degradation in the TCP performance due to call drops and link terminations

The experiments were conducted using the bandwidth measurement tools like IPERF [33], TTCP [34] and by transferring test (video) data with FTP. In either case TCPDUMP is used to grab the TCP headers of all the packets. Later, a TCP analysis tool, TCPTRACE [35] is used to determine various performance metrics and analyze the results. Plots of instantaneous throughput, average throughput, RTT variation, congestion window, time segment graph and time line graph are used to assess the overall performance of the system.

5.4.3 TCP performance over single Iridium link

Figure 5.12 shows the throughput as observed at the sender. The red line tracks the average of the last 10 segments TCP data segments and the blue line tracks the running average of the throughput for one modem. The average throughput of the connection is 2.45Kbps.

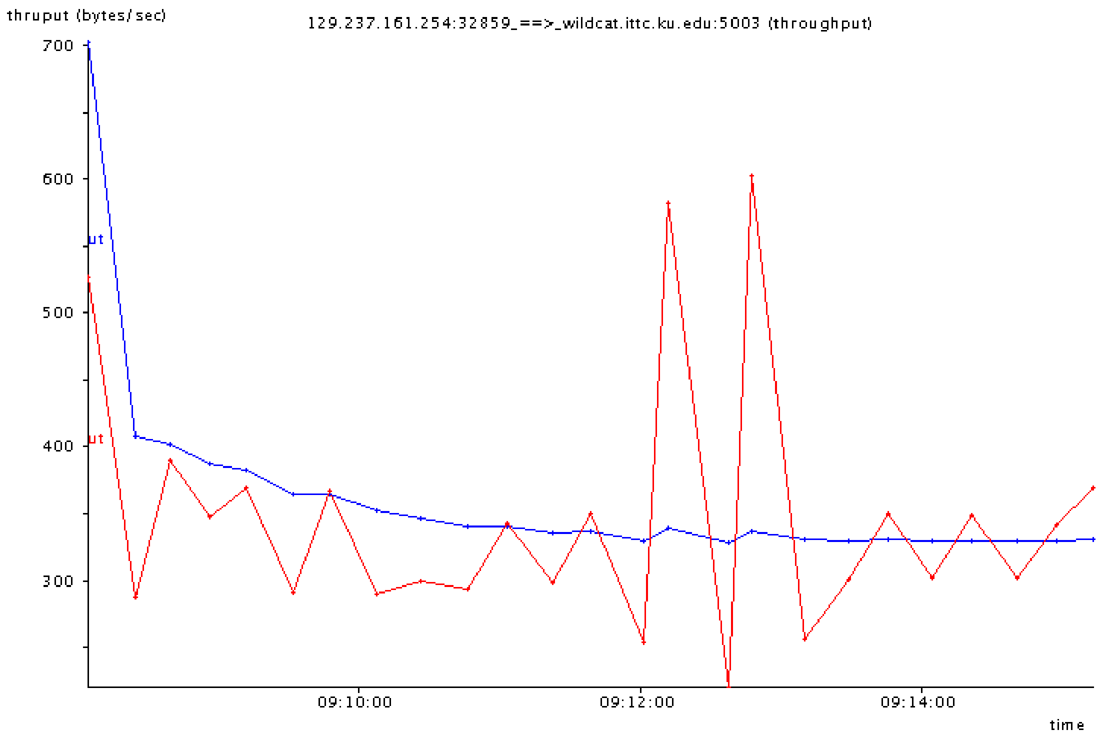


Figure 5.12 Throughput of a single Iridium link

The measured RTT of the system is shown in figure 5.13. The average RTT was found to be 20 seconds. The average RTT measured with ping (sec 5.2) was 11 seconds. Given the random variation of delay of the Iridium system, measurements done during different time periods could give different results. A detailed explanation of increased RTT is given in section 5.4.2

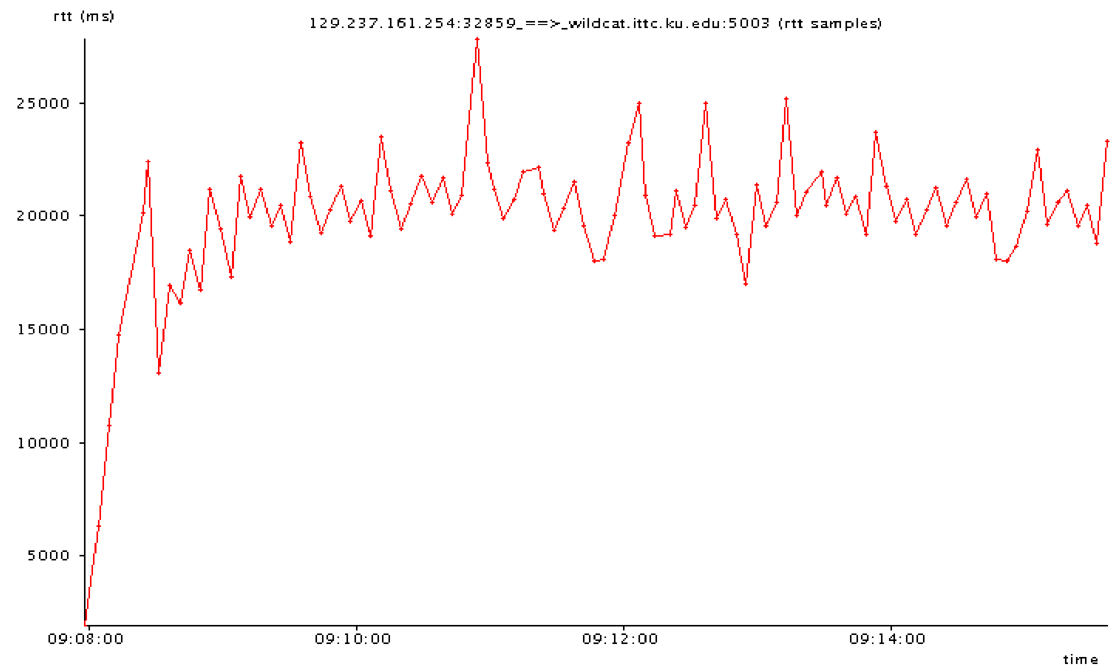


Figure 5.13 Measured RTT of a single Iridium Link

5.4.4 TCP performance over multi-channel Iridium system

In this section, we consider the performance of TCP over multi-channel system using four modems. The throughput of the system is shown in figure 5.14. The average throughput of the connection is 9.4 Kbps. The red line tracks the throughput of the last 10 TCP data segments. The throughput of a segment is defined as the size of the segment seen divided by the time since the last segment was seen (in this direction).

Blue line tracks the average throughput of the connection up to that point in the life time of the connection (total bytes seen / total seconds so far). The RTT measured from the TCP segments is plotted in figure 5.15. RTT is measured as the time interval between the instance when the TCP sends out a segment and the instance of arrival of an ACK, acknowledging the receipt of that segment. The average RTT is found to be 16.6 seconds.

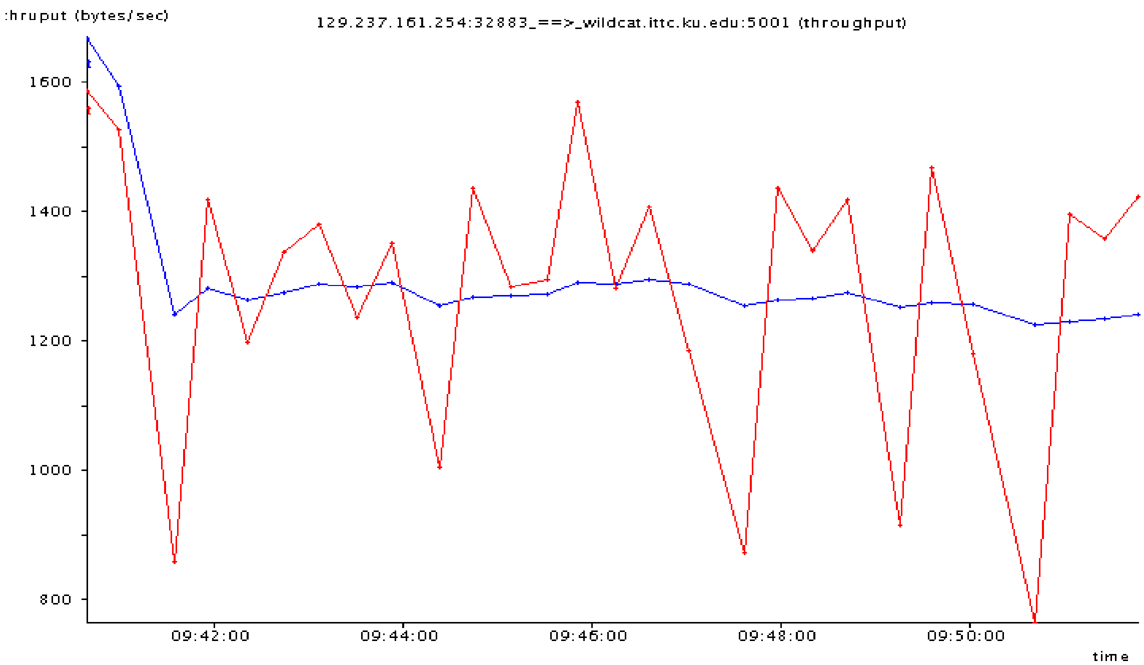


Figure 5.14 Throughput of a 4-channel Iridium communication system

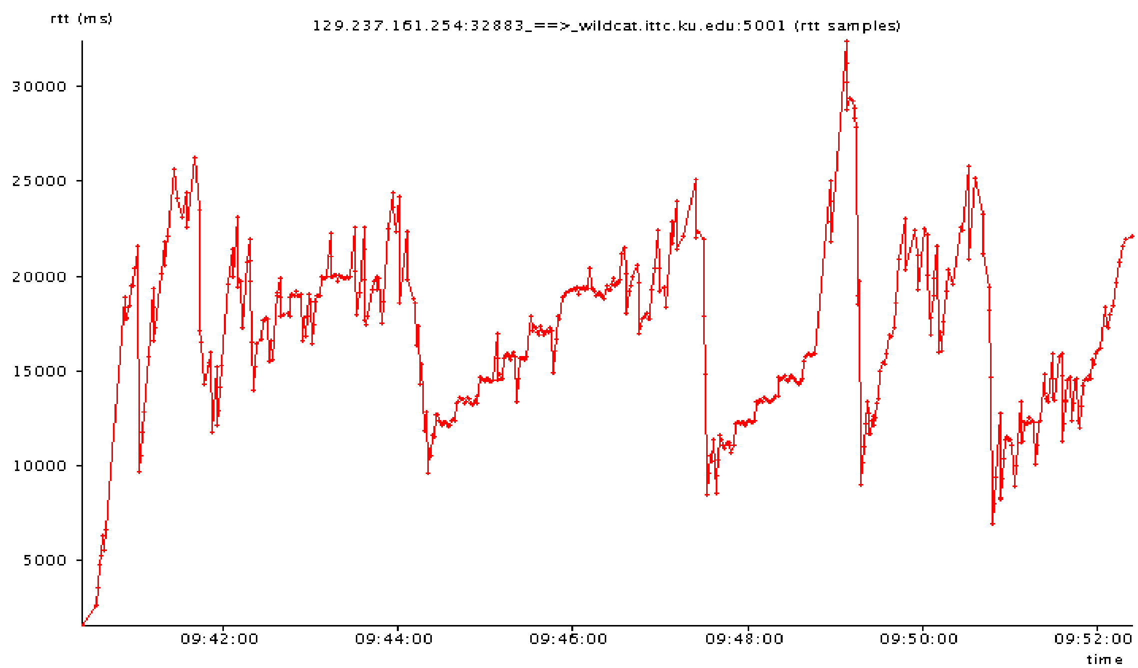


Figure 5.15 RTT of the 4-channel Iridium communication system

Both the variation in throughput and RTT is affected by three important factors.

a. ***TCP Slow start:*** During the slow start period, as shown in the time sequence graph of figure 5.16 (and its close-up in figure 5.17) and the outstanding unacknowledged data of figure 5.18, every ACK (indicated by green line) received increases the congestion window (cwnd) of the sender by one segment. Hence upon receipt of one ACK, TCP sends two segments (indicated by black arrows) almost simultaneously (with in a few millisecond interval). But since the Iridium system has an average capacity of 2.4 Kbps (and a transmission time of 5 secs), the second segment has to wait an additional 5 seconds while the first segment is being transmitted. This adds 5 seconds to the actual RTT of the second segment. TCPTRACE estimates the throughput as the segment size divided by the time since the last segment is transmitted. In this particular case, the second packet will be estimated at a very high bandwidth (as it was transmitted by TCP with in milliseconds of the first segment).

b. Fragmentation by MLPPP: As discussed in section 2.2.3, in a multi-channel system, MLPPP fragments an upper level protocol data unit (PDU) in to a number of fragments depending upon the packet size and the availability of the links amongst other factors. Under low loads (when all links are available to send data packets) each TCP packet is spit into number of fragments (not to exceed the total number of links) that are simultaneously transmitted across multiple satellite links resulting in higher throughput and lower RTT. On the other hand, under heavy loads, data packets are fragmented in to fewer segments or not fragmented at all. Hence, the resulting large data packets experience lower throughputs and higher RTT values.

c. ***Random Delay and cumulative acknowledgements***: Due to call hand-offs and continuous variation of the inter-satellite distances during the lifetime of the connection, the end-to-end delay varies significantly. The sudden variations in RTT cause the data transmission to be bursty. Further, TCP cumulatively acknowledges multiple packets by delaying the acknowledgment of some packets. Especially in a multilink-channel under heavy load, packets are transmitted (without fragmentation) over the multiple links in a round robin fashion. Due to the inherent variation in the end-to-end delay, packets transmitted in different links experience difference delays. The receiver, in this case, is more likely to receive out of order packets and hence generates cumulative ACKs. Though, the cumulative ACKs itself are not used for RTT calculation (by TCPTRACE), they results in TCP transmitting multiple packets (bursty traffic) to fill in the congestion window (as the outstanding unacknowledged data suddenly falls), leading to increased RTT as discussed in (a)

Figure 5.18 shows the outstanding unacknowledged data as a function of time. Since there is no direct method of measuring the congestion window, this provides a close estimate of the progress of the TCP congestion window as shown in figure 5.18.

During the slow start phase, the increase in the window is exponential. Where as during the congestion avoidance phase the widow scales linearly with each RTT.

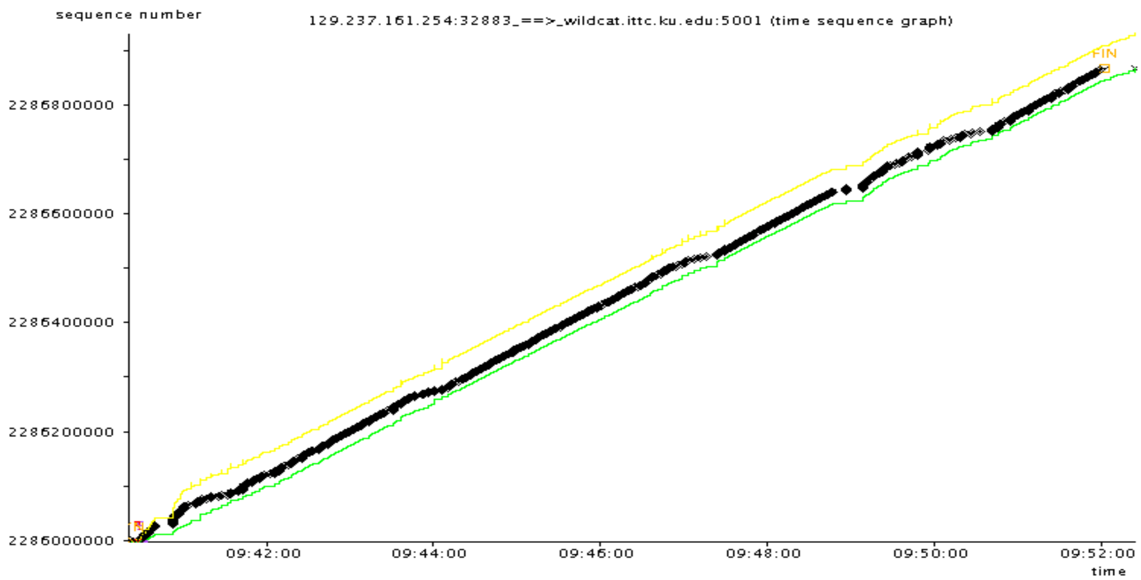


Figure 5.16 Time sequence graph of the 4-channel Iridium system

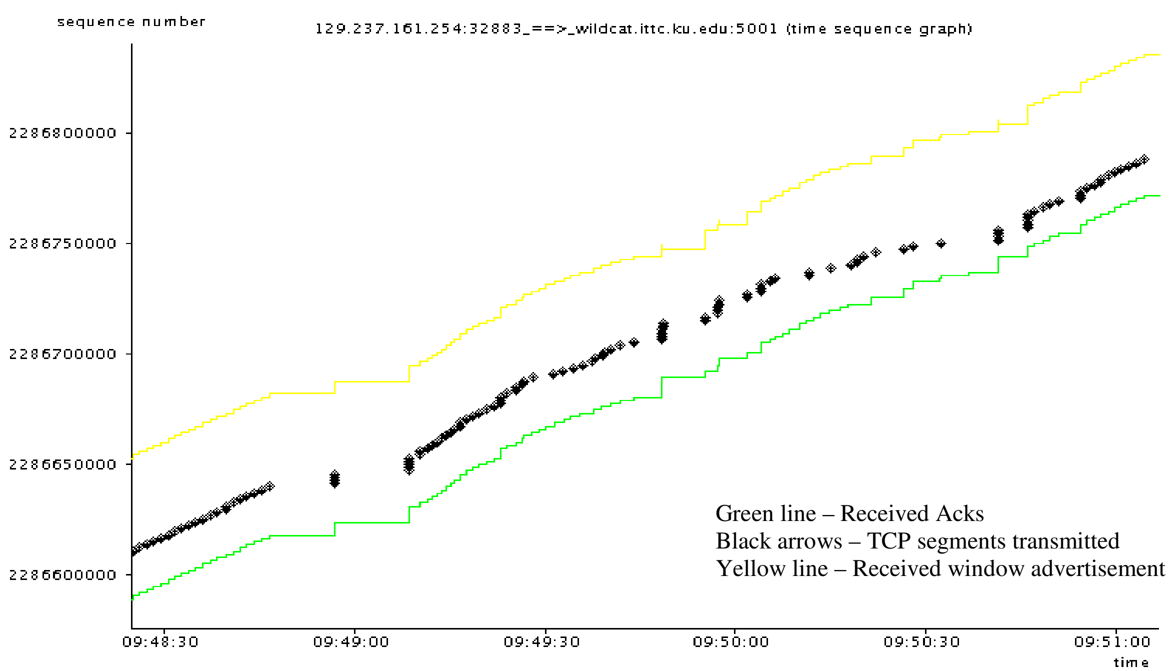


Figure 5.17 Closer look at the time sequence graph of a 4-channel system

It is noticed that due to cumulative acknowledgements, the progress of congestion window is slowed down and hence it never reaches the maximum window size advertised (63712 bytes).

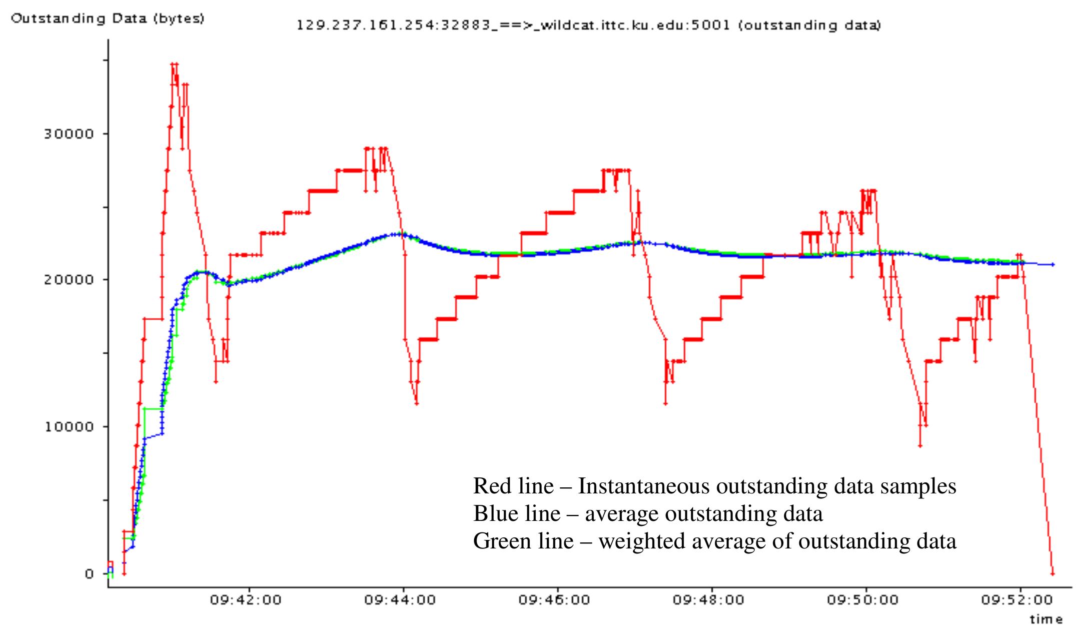


Figure 5.18 Outstanding unacknowledged data during a TCP connection using 4-channel Iridium system

From the trace file of the TCP connection shown in figure 5.16, it was observed that the maximum idle time is 22.116 seconds which is also evident from the time sequence graph of figure 5.17. This phenomenon occurs when the maximum number of allowed in-flight packets has been transmitted; the sender must wait to receive an acknowledgement from the receiver to proceed with further transmission. These long delays in the receipt of an acknowledgement often occur when an out of order packet is received and the receiver later sends a cumulative ACK. Upon receipt of cumulative ACK, the following burst of data packets at the sender result in queuing of packet (due to limited transmission ability of the sender.), which ultimately leads to an increase the RTT. The performance of TCP over 2-channel and 3-channel system is given in Appendix 2.

5.4.5 Performance degradation due to packet loss

In this section we present the effect of packet loss on the TCP performance. Since, the system has low packet loss rate, long duration experiments have to be conducted to observe packet losses.

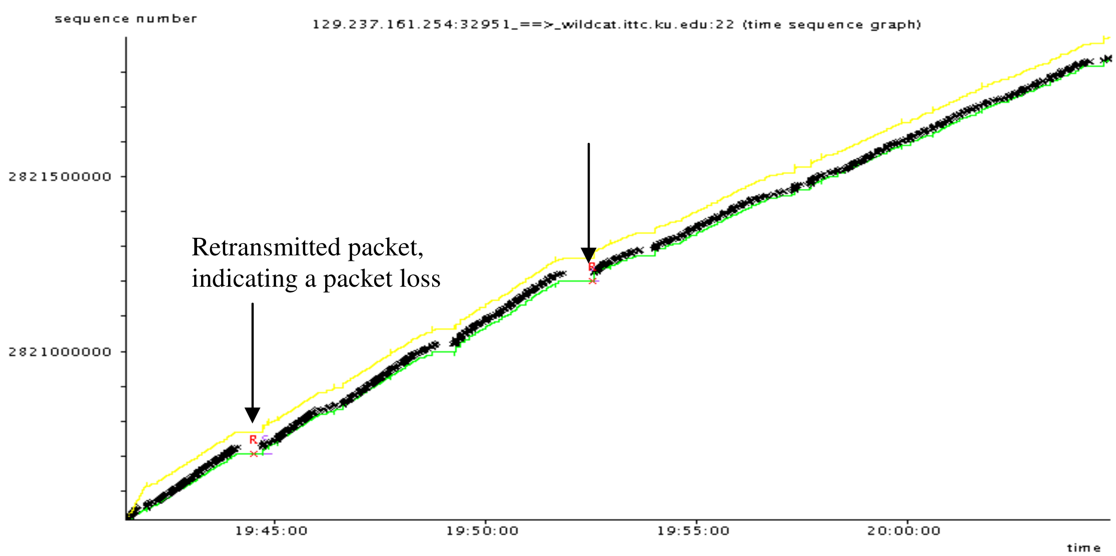


Figure 5.19 Time sequence graph of a video upload

Figure 5.19 shows a part of the time sequence graph of the FTP upload of a large video file. There are two packet losses that were observed in the shown time interval resulting in packet retransmissions (indicated by red arrows). A zoomed in plot of the time sequence graph is shown in figure 5.20. It was verified from the trace files that the loss was detected by the retransmission time out (RTO). The effect of this time out is clearly evident in the outstanding data plot (equivalent to congestion window) of figure 5.21.

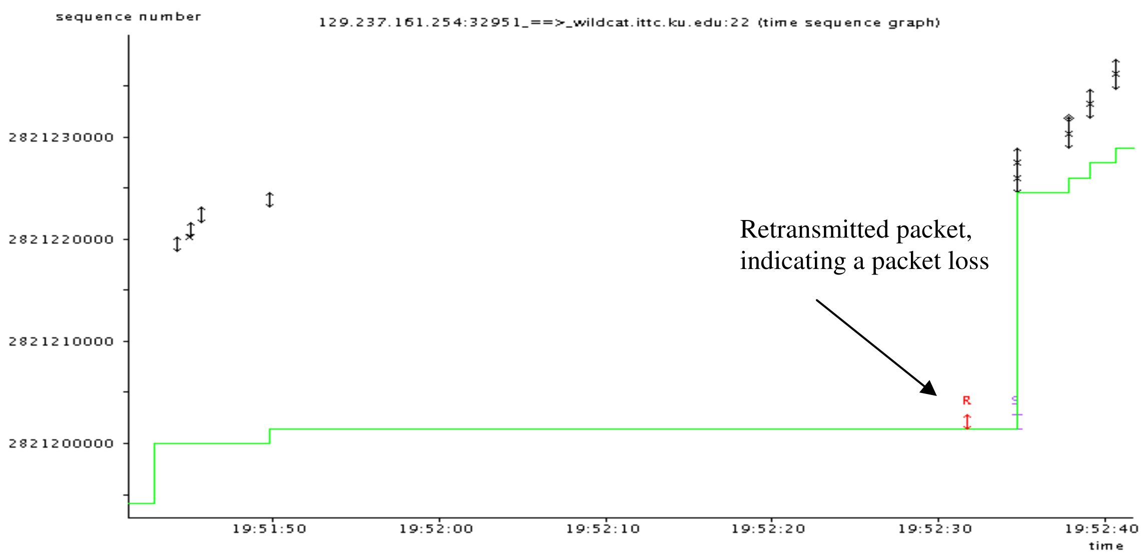


Figure 5.20 Time sequence graph of a video upload (Zoomed in)

The TCP connection was in the congestion avoidance phase when the packet loss occurred. The window size at that instant ware approximately 30,000 bytes. The expiration of the timer caused the slow start threshold (ssthresh) to be set to half of existing ***cwnd*** (30000/2 = 15,000 bytes), the cwnd was set to one segment (1500 bytes) and the connection entered slow start phase. The cwnd was increased one segment per ACK, till it reached the ssthresh set earlier (at 15000 bytes), where upon it entered the linear congestion avoidance phase.

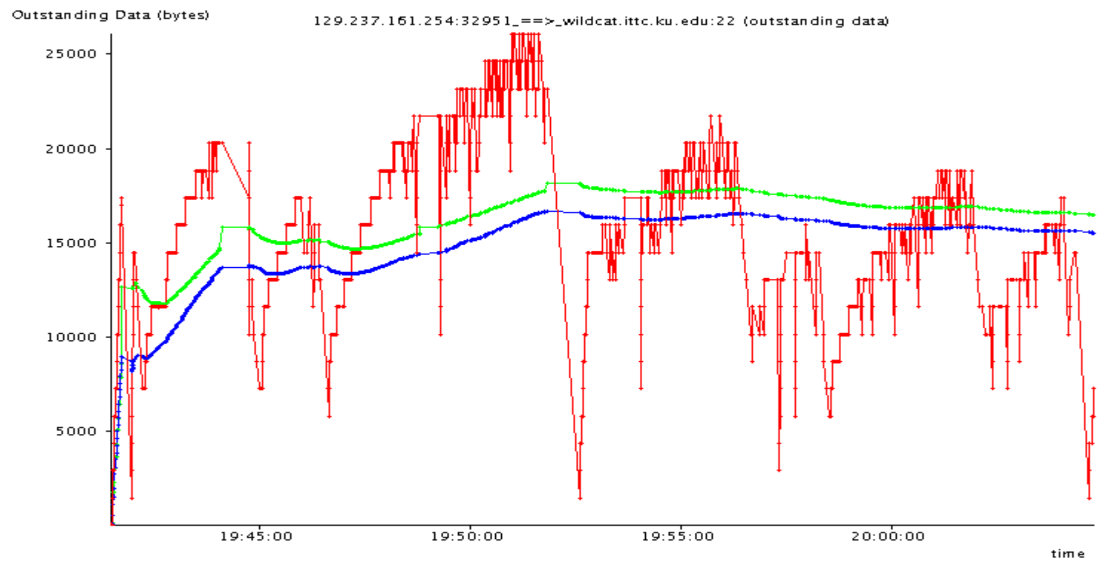


Figure 5.21 Outstanding unacknowledged data of a video upload

This increase in RTT and decrease in throughput of the system is shown in figure 5.22 and 5.23.

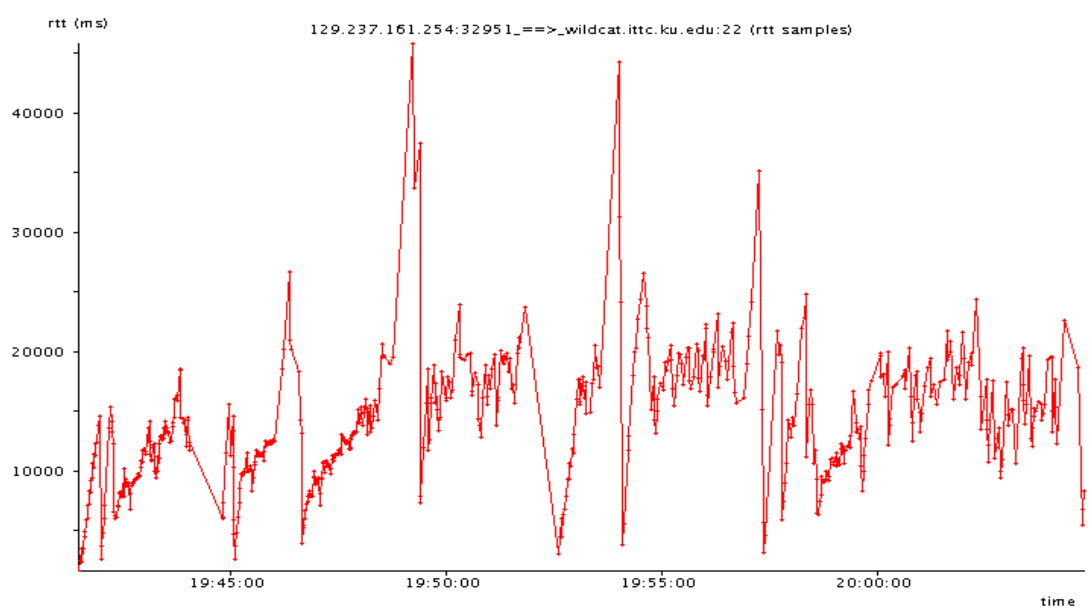


Figure 5.22 Measured RTT during video upload

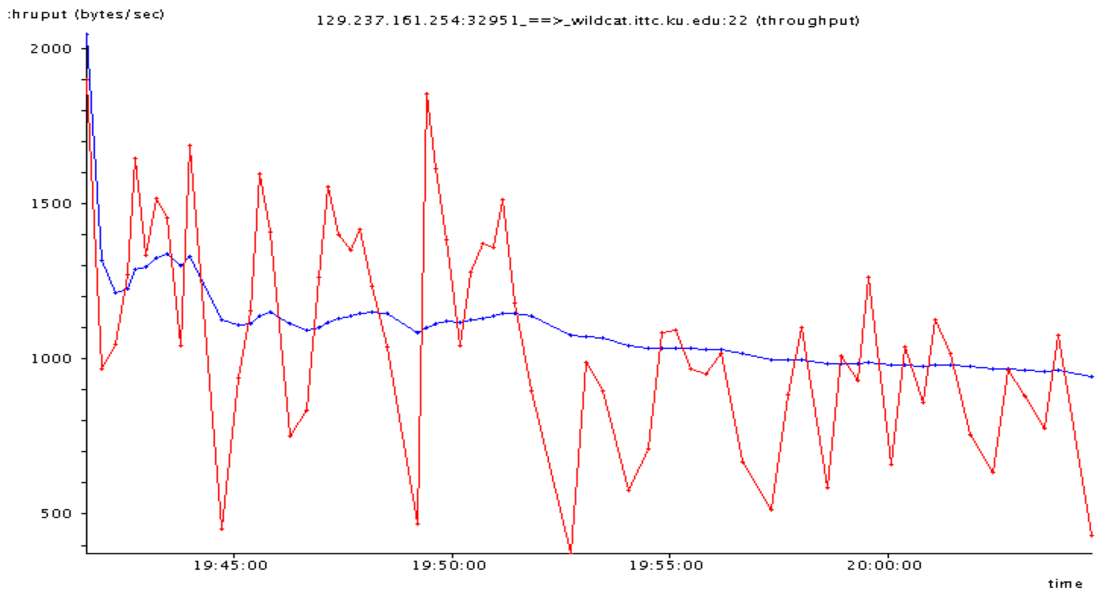


Figure 5.23 Throughput observed during video upload

The average throughput of the connection was observed to be 7.55 Kbps. Immediately following the maximum RTT point is a minimum RTT point. This is due to the fact that the TCP packet in question is fragmented by MLPPP over all the 4 links (which are all idle). Hence the RTT of the first couple of packets following a loss is low. As the number of packets to be transmitted increase, MLPPP fragments the TCP packets in to fewer fragments, thereby increasing the RTT.

5.4.6 Performance degradation due to call drops and link termination

In this section we present the effect of a call drop and link termination on the TCP performance. Figure 5.23 shows the time sequence graph of data transfer using IPERF. A call drop was emulated (by disconnecting the antenna cable) to evaluate the performance hit on TCP. The corresponding cwnd is shown in Figure 5.25.

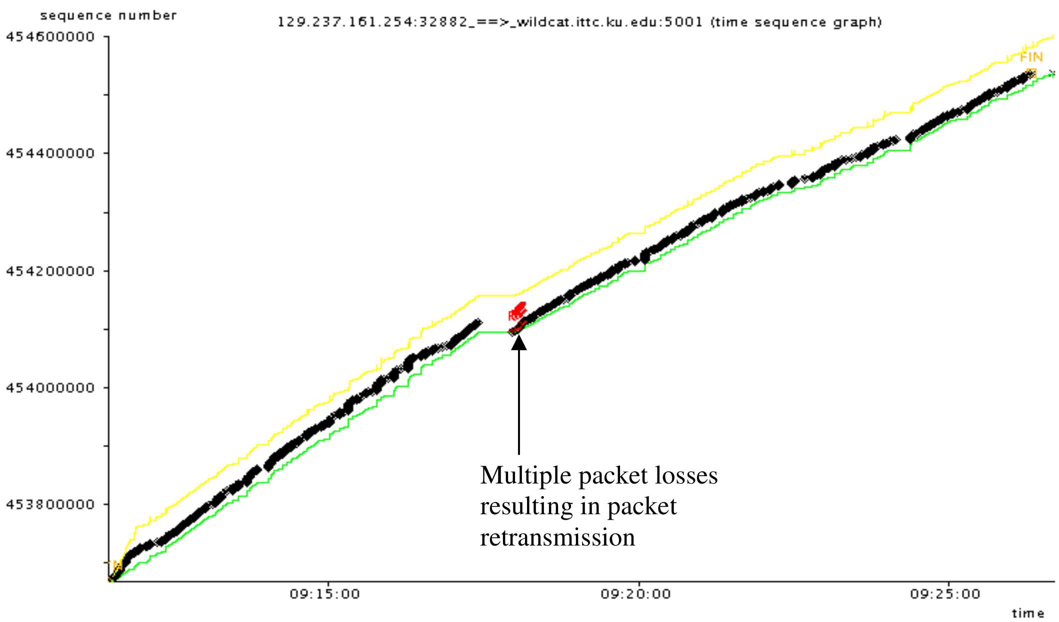


Figure 5.24 Time sequence graph of a TCP connection with a single call drop

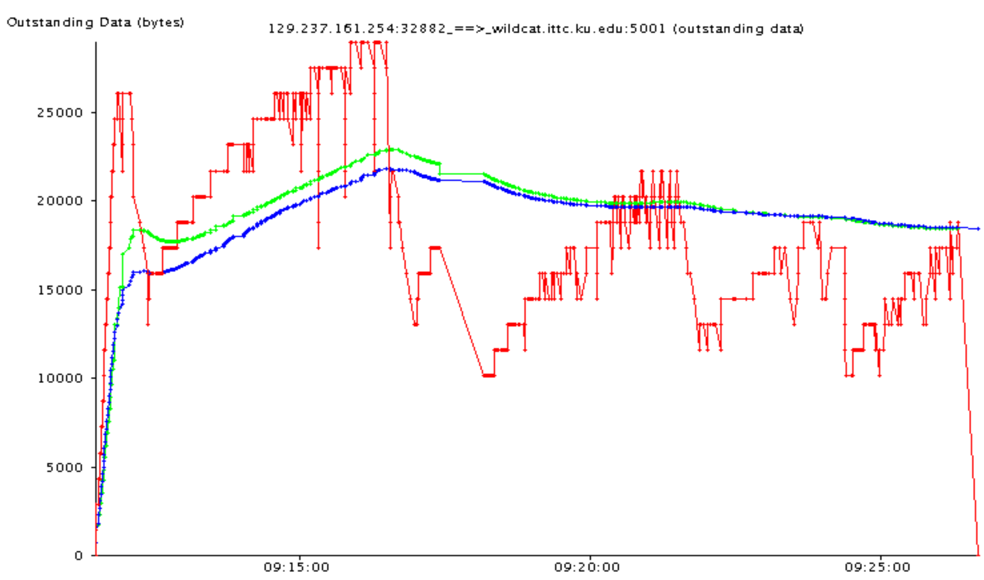


Figure 5.25 Outstanding data window of TCP connection with a single call drop

In this case, it takes a finite amount of time for the data link layer realize the link has failed, during which it sends data packet over an essentially failed link.

As seen in Figure 5.26, the first dotted line indicates the call drop event and the second dotted line indicates the time when the PPP layer realizes the failure. In this case, the entire window of packets that is in flight on that particular link is dropped leading to a large number of packet losses. These losses observed are either due to loss of a packet in flight or acknowledgement in flight. Due to large RTT and mean deviation of the link, the RTO timer is large. Hence, the system wastes the bandwidth until the RTO timer expires upon which it retransmits the dropped packets. In this particular example 12 packets were retransmitted. The reduction in throughput due to this call drop is evident from the throughput plot of Figure 5.27. The overall throughput of the connection was observed to be 7.6 Kbps.

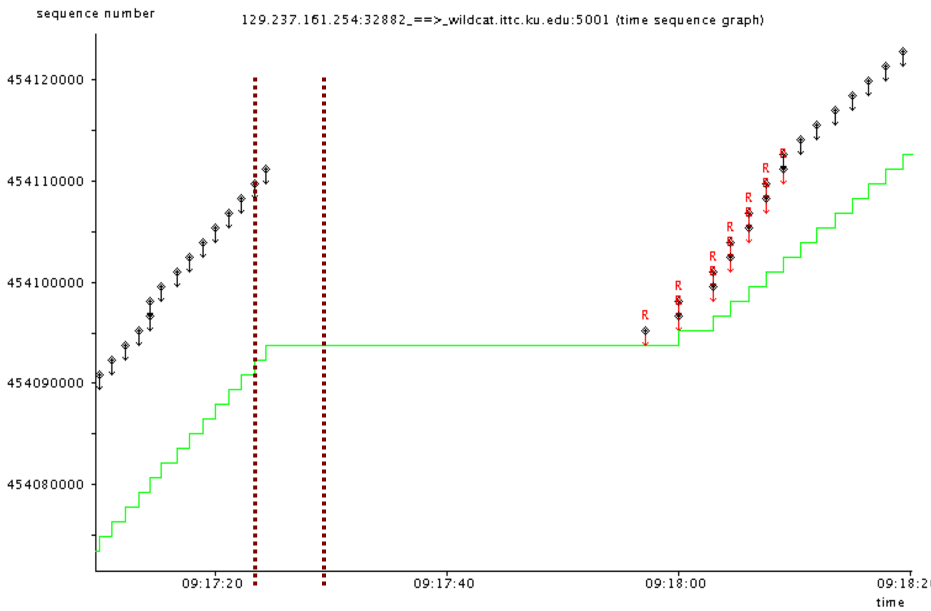


Figure 5.26 Time sequence graph a TCP connection with a single call drop

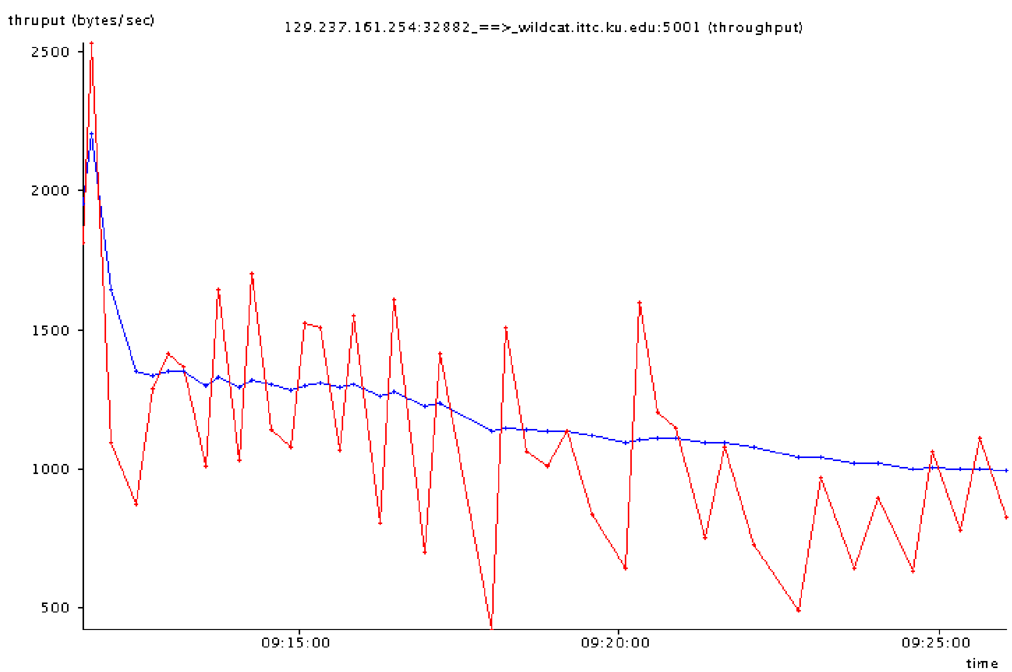


Figure 5.27 Throughput of the TCP connection with a single call drop

Link Termination: Consider a light load case where a link (in a 4-link bundle) is terminated as the current application could be supported by the aggregate data rate of three modems alone. In this case the fourth link will be neatly terminated (software terminated). It was observed that during this link termination, a small number of packet losses occur.

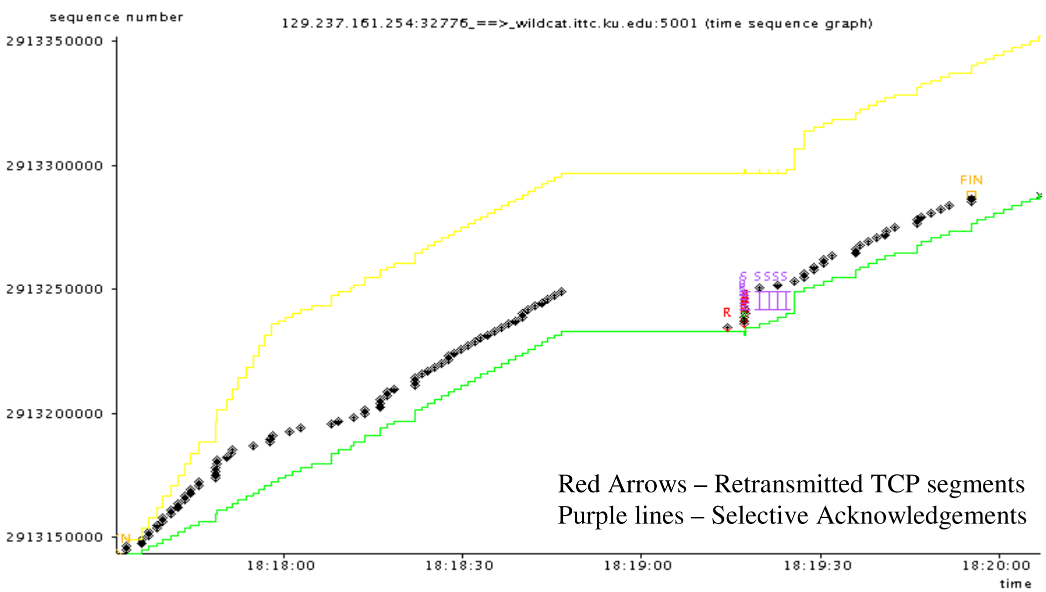


Figure 5.28 Time sequence graph of a TCP connection with a single link termination

This is due to the fact, that when the user starts the link termination phase on any given (PPP) link, any packets or acknowledgements that are in flight on that link are discarded. This is clearly evident from the time sequence graph of Figure 5.28 and 5.28 where 6 packets are lost during the link termination resulting in 6 retransmissions (indicated by red arrows). The remaining packets are selectively acknowledged (purple lines) and the dropped packets are subsequently retransmitted.

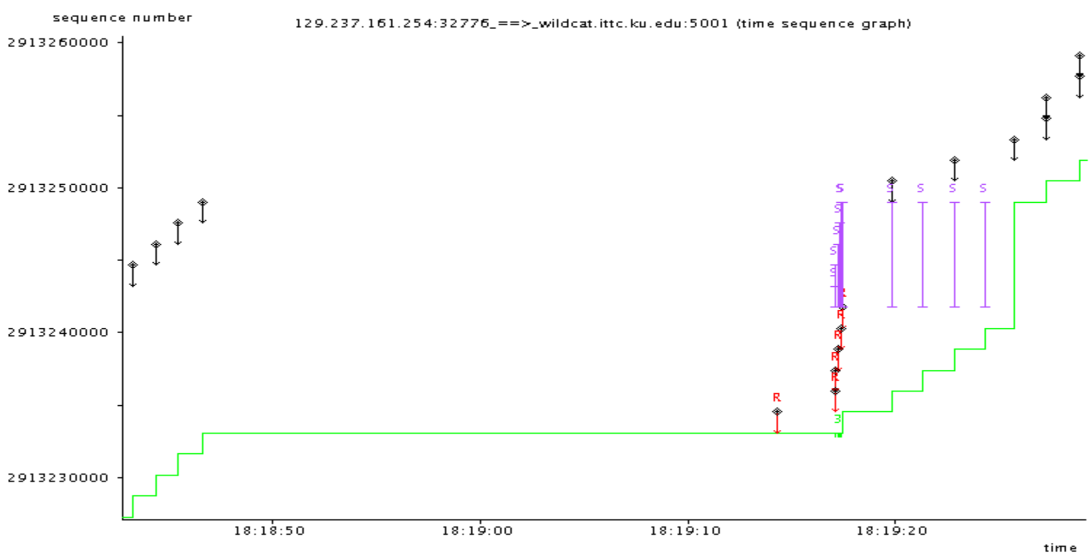


Figure 5.29 Closer look at time sequence graph of a TCP connection with a single link termination

5.5 Mobile System Performance

As discussed in section 5.2 we conducted mobile experiments to determine the performance of the system over mobile platforms. Figure 5.30, 5.30 and 5.31 shows the path taken by the test vehicle, speed and the throughput obtained during the first test.

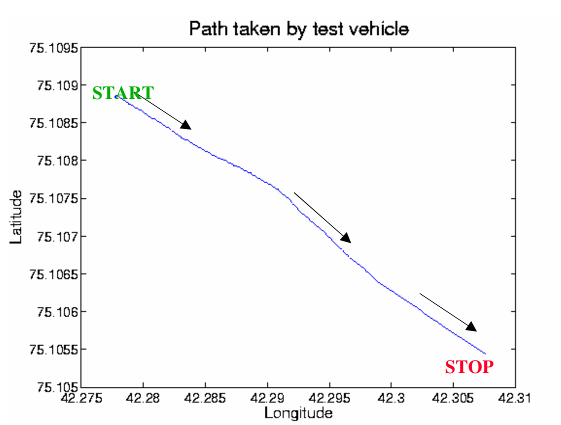


Figure 5.30 Path taken by test vehicle

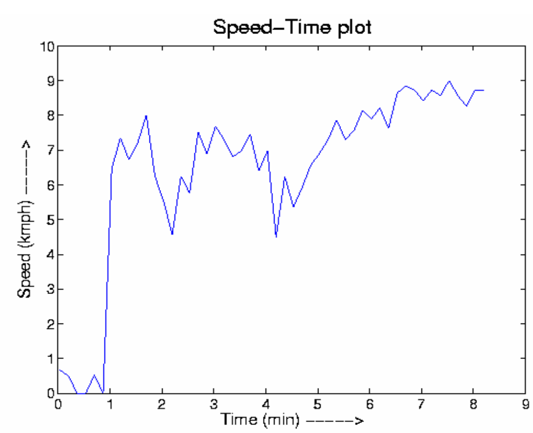


Figure 5.31 Speed of the vehicle

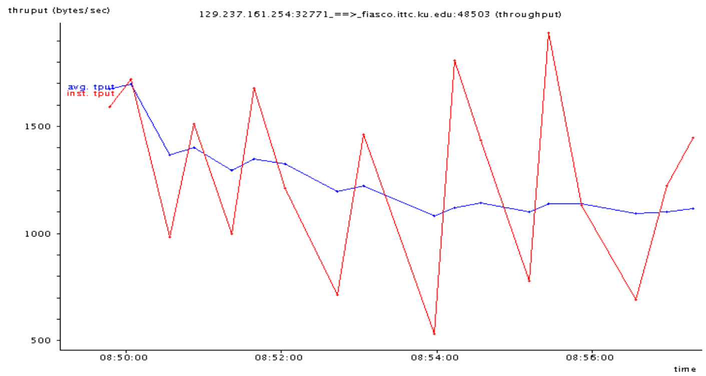


Figure 5.32 Throughput of the mobile 4-channel system

The results of a second test conducted at higher speed are shown in Figure 5.33 to 5.34

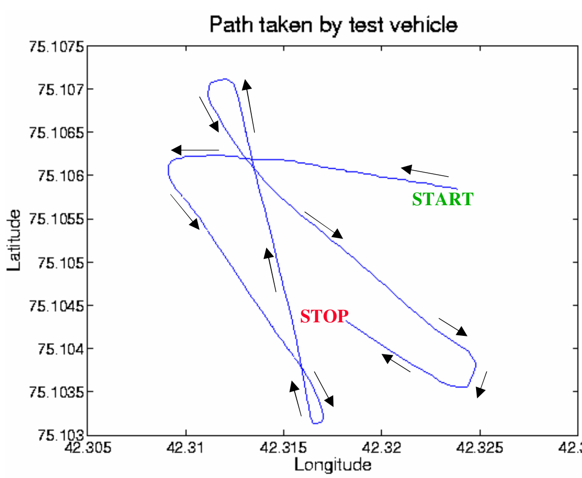


Figure 5.33 Path taken by test vehicle

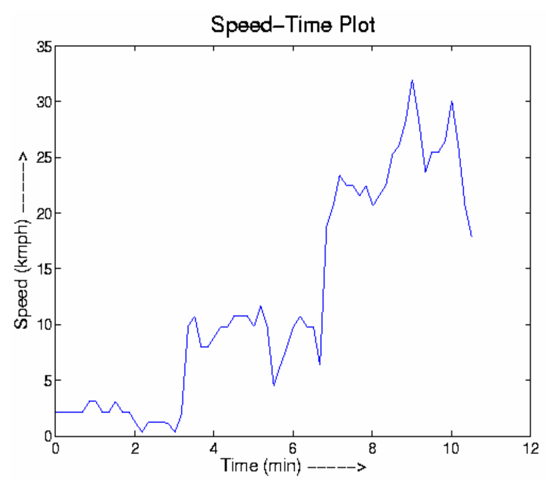


Figure 5.34 Speed of the vehicle

As expected, the performance of the system was similar to that of stationary system. For speeds up to 15 mph an average throughput of 9.4 Kbps was obtained. Also, the instantaneous throughput (red line) of the system is very similar to that of stationary system.

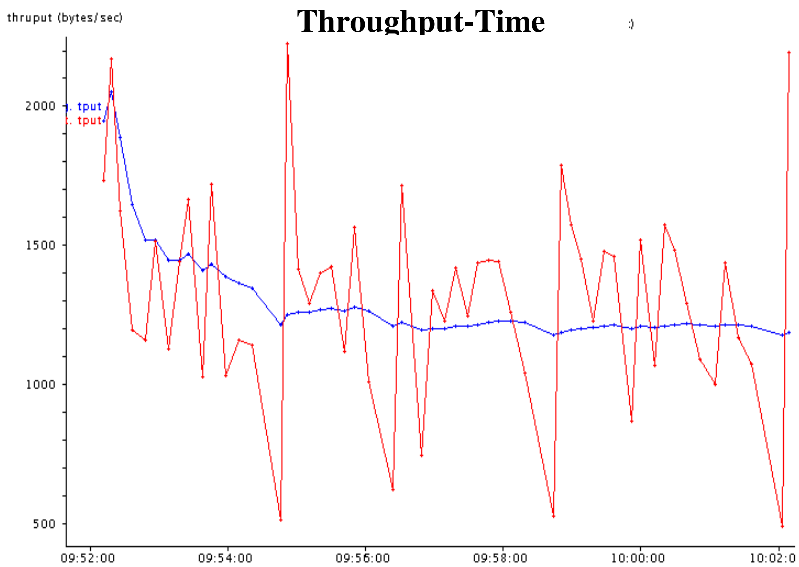


Figure 5.34 Throughput of the mobile 4-channel system during second test

5.6 Applications

Finally, we briefly present the various applications of the system that were tested during the field trip to NGRIP, Greenland. The system was used to download files of various sizes both from the Internet and the University network. Table 5.8 summarizes the HTTP downloads and the importance of the files to researchers obtained through user survey.

Table 5.8 Summary of file downloads from Greenland using the Iridium system

	Title	Downloaded/uploaded	Size	Imp
1	Spectrum Analyzer programmers Manual	Download from Agilent.com	7.2MB	9
2	Matlab Programs	Download from ITTC	500KB	7
3	Voltage regulator data sheet	Download from Fairchild.com	226KB	9

4	GPS software	Download	800KB	9
5	Proposal submission	Upload	600KB	8
6	Access point manager software	Download from Orinoco.com	4.66MB	7
7	Drawing of machine spares to order	Upload to University of Copenhagen	1MB	9
8	Video of core, datasheet	Upload for press release	2MB	8
9	Pictures, press release of longest core in Greenland	Upload to Kangerlussauq for press release	500KB	6

The system was used to conduct outreach activities and communicate with the faculty at the university of Kansas through video conferencing as shown in Figure 5.36 (a). Though both video and audio worked, long delays and low overall bandwidth proved to be a limit on the quality of such interactions.

Figure 5.36 (a) Video conferencing (b) Wireless Internet access

The system was combined with long range 802.11b wireless network to provide internet access throughout the camp. Figure 5.36 (b) shows NGRIP researchers accessing wireless network. Email access kept the researchers in touch with their home institutions. Scientists at NGRIP were able to send information back to their home institutions. The system was also useful for general camp purpose: sending drawings to order spares for a broken caterpillar, excel spreadsheet for food order, general press releases, and downloading weather reports for planning C-130 landings.

6. Conclusions and Future Work

6.1 Conclusions

Satellite communication is the primary means of data/Internet access in many climatically challenged and geographically remote regions. While commercial satellite systems usually do not provide coverage at higher latitudes, especially in Polar Regions, communication based on NASA satellites needs bulky user terminals and is not available round the clock. We have developed a reliable, lightweight, and readily available multi-channel data communication system based on Iridium satellites that provide round the clock, pole-to-pole coverage.

Multi-link Point-to-point protocol is used to inverse multiplex (combine) multiple satellite point-to-point links to obtain a single logical channel with aggregate bandwidth. This technique not only increases the otherwise low bandwidth (2.4 Kbps) to useable limits, but also provides a modular solution that can be easily scaled according to user requirements. The MLPPP link parameters are tuned to operate efficiently over the delay variant Iridium satellite system. Finally, link management software is developed that ensures fully autonomous and reliable operation.

Due to the small bandwidth delay product and low packet error rate, the system has a high TCP throughput efficiency. The 4-channel system, implemented and tested at

Greenland, proved to be over 90% efficient providing an average throughput of 9.2 Kbps.

The system had an average system up-time of 95% and a full capacity up-time of 80% and thus is reliable. Mobile performance results are very similar to that of stationary systems. While the end-to-end network architecture developed to provide Internet access worked well, the minimum system round trip time is significant (~2 seconds), which can impair real time interactions.

The randomly variant end-to-end delay combined with MLPPP behavior of splitting packets on the multiple links of the bundle leads to out of order delivery of TCP packets. This causes cumulative acknowledgements in TCP resulting in bursty traffic and further increase in RTT. A packet loss or call drop in such a case results in significant degradation in the performance of the system. Finally, A call drop or (graceful) link termination results in the loss of the entire window of the packets in flight of that link.

6.2 Future Work

Since, the end-to-end delay of the system was found to be the most detrimental factor, additional research can be done to determine the cause of this delay and develop methods to overcome it. The new Data after Voice (DAV) service of Iridium can be evaluated for this purpose. By using Iridium modem-to-modem configuration the DAV service offers the potential for removing the extra hop through the ground gateway, with the potential to significantly reduce the delay. On the other hand, research can be done on the different versions of TCP, to determine the enhancements in TCP that can effectively handle delay variation and other RTT issues discussed in section 6.1. The user friendliness of the system can be improved by incorporating a graphical user interface for the connection set-up and self-test/diagnostic tools to troubleshoot the system. Finally, spacing and sharing of antennas can be researched to reduce the footprint of the antenna array.

References

[1] Proceedings of National Science Foundation United States Antarctic Program Communication Workshop, March 1999.

[2] Recommendations of the South Pole Users committee meeting (SPUC), Colorado 2002 and 2003.

[3] Polar Radar Ice Sheet Measurement Project. www.ku-prism.org

[4] Inmarsat coverage. www.inmarsat.com

[5] Globalstar coverage. www.globalstar.com

[6] Satellites utilized by the South Pole station. http://amanda.wisc.edu/data/comms-summary.shtml

[7] South Pole satellite visibility. http://adelie.harvard.edu/spole/satellite/riseset_table.html

[8] Nicolas S. Powell, South Pole satellite communications update, United States Antarctic Program, July 2002 .

[9] Demonstration of new communication technology at North Pole, NASA, May 1999. http://earthobservatory.nasa.gov/Study/NorthPole/northpole.html

[10] The Ocean.US Iridium Project. http://www.omnet.com/iridium/

[11] Lloyd Wood, Internetworking with satellite constellations, 2001.

[12] Iridium Satellite in pictures. http://www.obsat.com/irimage_e.html

[13] Iridium Antennas. http://kdishman.home.mindspring.com/work.html#Iridium

[14] Joseph Kabara, Iridium satellite communications tutorial. http://www2.sis.pitt.edu/~jkabara/tele-2100/iridium/iridium_final.html

[15] Fossa, C. E., Raines, R. A., Gunsch, G. H., and Temple, M. A, An Overview of the Iridium Low Earth Orbit (LEO) Satellite System, Proceedings of the IEEE National Aerospace and Electronics Conference,1998, pp. 152—159.

[16] Joseph N. Pelton, et al, Global Satellite Communications Technology and Systems, WTEC Panel Report, December 1998.

[17] Robert A Nelson, Iridium: From Concept to Reality. http://www.aticourses.com/news/iridium.htm

[18] G. M. Comparetto, A Technical Comparison of Several Global Mobile Satellite Communications Systems, Space Communications., vol. 11, no. 2, 1993, pp. 97–104.

[19] Pratt, S. et al, An Operational and Performance Overview of the IRIDIUM Low Earth Orbit Satellite System, IEEE Communications Surveys, Vol. 1, No. 3, 1999, pp. 2-10.

[20] Iridium hardware. http://www.nalresearch.com/satellite_modem_index.html

[21] Iridium Satellite System. www.iridium.com

[22] W. Simpson, The Point-to-Point Protocol (PPP), July 1994, RFC 1661.

[23] K. Sklower, B. Lloyd, G. McGregor, D. Carr, T. Coradetti, The PPP Multilink Protocol", 1996, RFC 1990.

[24] Paul Mackerras, PPP Generic driver and channel interface, Linux-2.4.

[25] MGETTY. http://alpha.greenie.net/mgetty/

[26] T1 Inverse multiplexing. http://www.larscom.com/pdfs/white_papers/wp_t1imux.htm

[27] Frost and Sullivan, "Satellite telephone Quality of Service comparison: Iridium Vs. Globalstar", July 2002.

[28] Mark Allman, Improving TCP Performance Over Satellite Channels, June 1997.

[30] Rahul Sangal, Performance Analysis Of The Transmission Control Protocol Over Low Earth Orbit Satellite Communication Systems, August 1999.

[31] P. Loreti, et al, Throughput and Delay Performance of Mobile Internet Applications Using LEO Satellite Access.

[32] C. Partridge, T. J. Shepard, TCP/IP Performance over Satellite Links, IEEE, Network, September-October 1997, pp. 44-49.

[33] IPERF. http://dast.nlanr.net/Projects/Iperf/

[34] TTCP. http://www.pcausa.com/Utilities/pcattcp.htm

[35] TCPTRACE. www.tcptrace.org

[36] Yvette C. Hubbel and Lokheed Martin Sanders, A comparison of the Iridium and AMPS systems, IEEE network, March/April 1997.

[37] Modified AT command set supported by Iridium modems, http://www.nalresearch.com/pdf/SSP-ISU-CPSW-USER-0005V19.pdf

Appendix A

A.1 Specifications of the Iridium Modem used in the field experiments

Specifications

Length:	7.73" (196.4 mm)
Width:	3.25" (82.6 mm)
Depth:	1.54" (39.0 mm)
Weight:	~1.3 pound (~610 g)
Data Rate:	2.4 Kbits/sec
Hardware/Software Interface:	RS232/Standard AT Commands

Environmental Specifications

Operating Temperature Range	-30°C to +60°C
Operating Humidity Range	< 85% RH
Storage Temperature Range	-40°C to +85°C
Storage Humidity Range	< 85% RH

Physical Connectors

Iridium Antenna	TNC
Multi-Interface Connector	25-Pin D-Type

DC Power Interface Specifications

Main Input Voltage Range:	+4.0VDC to +5.0VDC
Main Input Voltage Nominal:	+4.4VDC
Main Input Voltage Ripple:	40mV peak-to-peak
Power-up Current:	~2.2A @ 4.4VDC
Transmitting/Receiving Current:	~1A @ 4.4VDC average
Standby Current:	~120mA @ 4.4VDC average

RF Interface Specifications

Frequency Range:	1616 to 1626.5 MHz
Duplexing	Time Domain Duplex (TDD)
Multiplex Method:	TDMA/FDMA
Link Margin (w/ external antenna):	12.5 dB average

A.2 Specifications of the Mobile flat antenna used in the field experiment

Specifications	
Diameter	3.5" (8.9 cm)
Height	0.85" (2.2 cm)
Weight	14.3 ounces (405 g)
Operating Temperature	-40°C to +70°C
Connector	TNC Male
Impedance	50 Ohms
VSWR	Less than 2:1
Frequency	1616 - 1625.5 MHz
Polarization	Right hand circular
Axial Ratio	15 dB Max
Radiation Patterns	1.0 dB Zenith to 40° 0.5 dB 40° to 70° -0.5 dB 70° to 80°

Appendix B

Packet format of the multilink point-to-point protocol – RFC 1990

The format of individual fragments, which are the packets in the Multilink protocol is shown below [23].

```
                +---------------+---------------+
PPP Header:     | Address 0xff  | Control 0x03  |
                +---------------+---------------+
                | PID(H)  0x00  | PID(L)  0x3d  |
                +-+-+-+-+-+-+-+-+---------------+
MP Header:      |B|E|0|0|0|0|0|0|sequence number|
                +-+-+-+-+-+-+-+-+---------------+
                |     sequence number (L)       |
                +---------------+---------------+
                |        fragment data          |
                |               .               |
                |               .               |
                |               .               |
                +---------------+---------------+
PPP FCS:        |              FCS              |
                +---------------+---------------+
```

Network Protocol packets are first encapsulated according to normal PPP procedures, and large packets are broken up into multiple segments sized appropriately for the multiple physical links [23]. The Address and Control Field are not included in the logical entity to be fragmented. A new PPP header consisting of the Multilink Protocol Identifier, and the Multilink header is inserted before each section. PPP multilink fragments are encapsulated using the. Following the protocol identifier is a four-byte header containing a sequence number, and two one bit fields indicating that the fragment begins a packet or terminates a packet. The (B)eginning fragment bit is a one bit field set to 1 on the first fragment derived from a PPP packet. The (E)nding fragment bit is a one bit field set to 1 on the last fragment A fragment with both the

(B)eginning and (E)nding fragment bits set to 1 is a complete network protocol data unit. The sequence field (a 24 bit or 12 bit number) is incremented for every transmitted fragment.

Appendix C

The PPP configuration script on the client

```
# Select the serial port to which the modem is connected
/dev/ttyS0
# Enable debugging
debug
# Enable the multilink option
multilink
# Set the serial link speed
19200
# Enable hardware flow control
crtscts
# Set the interface that will be created as the default route
defaultroute
# Set the authentication retransmission timeout
pap-restart 10
# Set the maximum number of pap requests
pap-timeout 10
# Set the LCP link negotiation retransmission timeout
lcp-restart 10
# set the maximum number of LCP configure requests
lcp-max-configure 20
# Set the time interval between echo packet
lcp-echo-interval 30
# Set the maximum number of un replied echo packets to pronounce the
link dead
lcp-echo-failure 2
# Username
user xxxx
# Do not force the peer to authenticate itself
noauth
# Logfile
logfile /etc/ppp/modem1log
# Disable compression
nobsdcomp
noccp
nodeflate
passive
# Set the delay in milliseconds to wait after the establishment of
serial link
connect-delay 5000
```

```
# Call the chat-script to initialize the modem and dial out
connect '/usr/sbin/chat -v -f /etc/ppp/chat-modem1'
SOURCE FILE: /etc/ppp/peers/modem1
```

The modem initialization script on the client

```
# Check for errors
ABORT "NO CARRIER"
ABORT "NO DIALTONE"
ABORT "ERROR"
ABORT "NO ANSWER"
ABORT "BUSY"
ABORT "Username/Password Incorrect"
# Hang up the modem if in use
"" "ATHn"
# Obtain the signal strength measurement
OK "AT+CSQ"
# Request data bearer service with the satellite network
OK "AT+CBST=6,0,1"
# Set the time out to receive a reply
TIMEOUT 100
# Dial the number
OK "ATDT 00697xxxxxxxxxx"
"CONNECT 19200" ""
SOURCE FILE: /etc/ppp/chat-modem1
```

PPP modem configuration script on the Server

```
# Enable debugging
debug
# Enable the multilink option
multilink
asyncmap 0
# Lock the serial port
lock
# Ser the serial link speed
19200
# Enable the hardware flow control
crtscts
# Set the authentication retransmission timeout
pap-restart 10
# Set the max number of pap requests
pap-timeout 10
# Set the lcp request retransmission timeout
lcp-restart 10
# Set the maximum number of LCP-configure requests
lcp-max-configure 20
# Force the peer to authenticate
auth
# Force the user to PAP authentication
require-pap
# Set the IP address of the server and client
xxx.xxx.xxx.xxx:xxx.xxx.xxx.xxx
SOURCE FILE: /etc/ppp/options.ttyR1
```