ABSTRACT

Title of Document:	TRANSPORT PROTOCOL AND FLOW CONTROL FOR IP-BASED BROADBAND AERONAUTICAL SATELLITE NETWORKS
	Yadong Shang, Master of Science, 2007
Directed By:	Professor John S. Baras, Dept. of Electrical and Computer Engineering

The IP-based broadband aeronautical satellite network will provide numerous new applications and services for both airspace system operations and passenger communications. However, the interoperation between a satellite system and the exiting terrestrial Internet infrastructure introduces new challenges. In this thesis, we recommend suitable transport protocols for an aeronautical network supporting Internet and data services via satellite. We study the future IP-based aeronautical satellite hybrid network and focus on the problems that cause dramatically degraded performance of the Transport Protocol. Based on the observation that it is difficult for an end-to-end TCP solution to solve the performance problem effectively, we proposed a new splitting based transport protocol, called Aeronautical Transport Control Protocol (AeroTCP). The main idea of AeroTCP is the fixed window flow control, adaptive congestion control, and super fast error control. Simulation results showed that AeroTCP can achieve high utilization of satellite channel and fairness.

TRANSPORT PROTOCOL AND FLOW CONTROL FOR IP-BASED BROADBAND AERONAUTICAL SATELLITE NETWORKS

By

Yadong Shang

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2007

Advisory Committee:

Professor John S. Baras, Chair
Professor Richard La
Professor Nuno Martins

Dedication

To my wife, Joy, and our daughter, Jessica.

Acknowledgements

I would like to take this opportunity to express my gratitude to all those who have given me advice, help and support throughout my work. Foremost among these is my advisor Professor John S. Baras.

I am very grateful to Dr. Michael Hidjitheodosiou for his help and advice with this work.

I am also grateful to the Hybrid Network Center (former the Center for Satellite and Hybrid Communication Networks) for the support of my research. This work was supported under NASA cooperative agreement NCC8-235 and NASA cooperative agreement NAG3-2844.

Acknowledgement also goes to my friends and co-workers for their help and encouragement, especially Hui Zeng, Nicolas Rentz, George Papageorgiou, Vahid Tabatabaee.

Table of Contents

List of Figures

Chapter 1: Introduction

World's aviation industry is soaring into the 21st Century with projected increases in business, recreation, and personal travel. The current airspace systems are quickly becoming overburdened by increases in air traffic coupled with the use of old technologies and legacy systems [1]. These systems must be maintained to ensure safety and efficiency while also transitioning to future systems. In addition, airplanes seem to be the last remaining islands where mobile communications and Internet access is not available. The demand for making air travel more pleasant, secure and productive for passengers also demonstrated the need for major improvements and new initiatives in aeronautical communications.

New Internet infrastructure and technologies capable of providing high-speed and high-quality services are needed to accommodate multimedia aeronautical applications. Inspired by the big market and business opportunity, many investigations and commercial activities are being developed to establish broadband aeronautical communication networks. A Satellite communication system, distinguished by its global coverage, inherent broadcast capability, bandwidth-on-demand flexibility, suitability to free flight concepts, and the ability to support mobility, is an excellent candidate to provide broadband integrated services for aeronautical communications. [2]

Several companies (e.g., Boeing, Hughes, Loral Space) have announced plans to use satellite technologies to provide commercial broadband data services for airline passengers [3][4]. The future aeronautical satellite systems will offer Internet connections at up to broadband (tens of Mbps) data rates via networks of GEO or LEO satellites. Figure 1 illustrates the IP-based network topology of aeronautical satellite networks. This system will be composed of three major segments: cabin segment with on-board networks, space segment for interconnection of the cabin with the terrestrial networks, ground segment which provides the interconnection to the terrestrial personal and data networks as well as the Internet backbone. In this study, we focus on the GEO satellites because of their stationary relative to earth, large coverage, and significant reduction in system complexity comparing to LEO satellite systems. In our study, the GEO satellites are bent pipe satellites, which are simply signal repeaters in the sky. They are physical layer devices and no switching is performed on board.

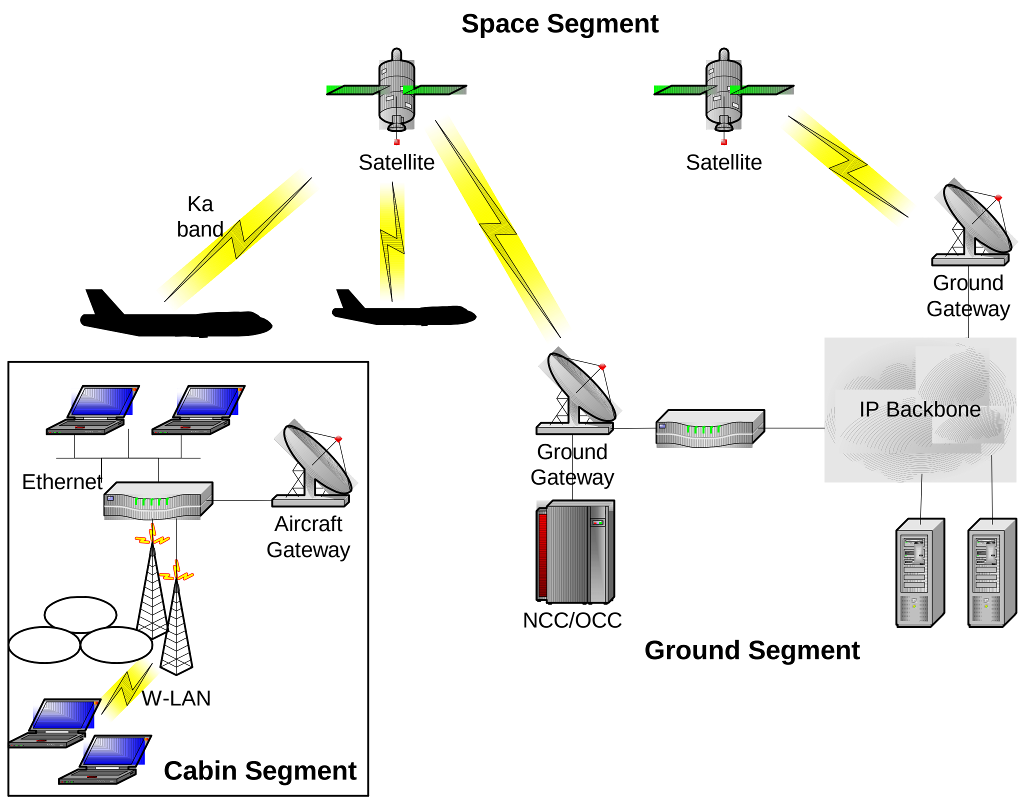


Figure 1 Aeronautical Satellite Networks

The aeronautical satellite networks will provide numerous new applications and services for both airspace system user operations and secure air traffic management. Those applications could include the System Wide Information Management (SWIM), Controller Pilot Data Link Communications (CPDLC), regular downloading of the aircraft's flight data and surveillance video, better enhanced weather information, voice over IP, telemedicine, and electronic flight bag applications. It also provides airlines with new revenue generating services (e.g. entertainment services, Internet access, directed advertising, and telephone service) for passengers [5]. In this thesis, we focus on the integration of satellite networks with the terrestrial networks to provide data services to and from aircraft, specifically TCP/IP traffic.

Application of commercial off the shelf (COTS) technologies and techniques has the potential to make network operations economically and technically realizable. However, the performance of data communications protocols and applications over aeronautical satellite systems is the subject of heated debate in the research community, especially the Internet TCP/IP protocol suite [6]. Our goal is to design an efficient and fair transport protocol for aeronautical satellite networks.

Chapter 2: Aeronautical Communications

2.1 Applications and services for Aeronautical Satellite Network

The types of applications, which must be supported in the aeronautical communications, can be divided into two categories: safety communications and not-safety communications. The aeronautical satellite communication system is normally used for communications related to the safety and efficiency of flight, but non-safety communications could be permitted on a non-interference basis, when priority and preemption can guarantee the precedence of the safety communications. [7]

2.1.1 Safety and non-Safety Communications

The safety communications are currently performed by the National Airspace System, which consists of both ground-to-ground and air-to-ground communication systems. Ground-to-ground communication systems interconnect all ground facilities to each other. Air-to-ground (A/G) communication systems provide pilot-to-controller communications. The safe separation of aircraft during flight is the essential task performed by air traffic control (ATC). Currently ATC services depend on air/ground voice communication between pilots and the air traffic controllers established principally via ground based VHF and UHF radios. These links support all phases of flight including ground movements; departures and arrivals; and en route. Furthermore, A/G communications are used to transmit instructions and clearances, provide weather services and pilot reports.

The safety communications also include new application scenarios, which make air traveling more secure for the passengers. Video, audio and avionic data transmission may be useful to prevent or analyze aircraft accidents. Flight data, cabin and cockpit video can be sent to ground and stored for a certain time. In case of aircraft disaster, these data can give helpful information for resolving hijacking or analyze aircraft failures faster and more precisely, before the “black box” is found.

Another important application is logistics and aircraft maintenance information, which is not observable to the passenger, but can reduce on-ground time and ease maintenance of the aircraft. For example, when the cabin crew or automated sensors recognize faulty equipment, maintenance crew on ground can prepare the repair and organize replacements parts in advance, based on detailed fault identification data being transmitted immediately.

Non-safety communications can make air traveling more pleasant, secure and productive for passengers. Today’s in-flight entertainment (IFE) systems only include a limited number of pre-recorded movies or music channels, short screen “news” and rudimentary travel info. All these one-way services are come from an on-board storage medium and presented at a fixed time. In recent years, some airline companies introduce new in-flight entertainment, such as direct TV, Internet applications and so on. But those services are limited in access (e.g., only in some particular airline and for first/business class). In the other end, modern users can get various entertainments at home or while moving on ground. Currently, Internet access for web applications and email seems to be the most attractive and fashioned feature to be provided to aircraft passengers, but the list of services is manifold. Moreover, IFE is only one of the driving applications for high data rate links to airliners.

Non-safety communications are more important for the business traveler. The time those travelers spend on board an aircraft can be made more productive. Design studies show that airlines are thinking of a new kind of office class. Almost one half of aircraft passengers are business travelers. Over 70 percent of them carry a mobile computer and over 80 percent a mobile phone [8]. The aircraft office for this user group raises some other design and technical challenges. While Internet access for passengers being on a vacation trip has to be available on installed terminals, e.g. in seat, the business user on board wants to connect his/her own equipment to the communication network, and power for this equipment has to be provided. Although a standardized in-seat terminal would ease electromagnetic compatibility problems, the need for a private workspace supporting the connection of own equipment will prevail from the airline customers' view. This brings about the interesting question of applicable protocols. Mobile IP may provide not only the possibility of getting access with personal equipment to Internet and work on the familiar desktop, it could also serve to extend the "personal network", for instance a company's VPN, to everywhere in the sky.

Based on the previous discuss, these two application categories, safety and non-safety communications, include a range of particular communication services. Table 1 assigns to each application category respective key services. Some services fit into more than one category. Moreover, not all services will be permanently required. In case of an emergency, for instance, the shutdown of passenger services for the benefit of flight security applications is acceptable. From a system design viewpoint, this immediately relaxes the worst-case data rate demand of the aircraft communication system.

Table 1 Categorized Service

Category	Services
Safety Communication	ATC, Weather services, pilot reports, Cabin and cockpit surveillance video, flight recorder data, aircraft logistics and maintenance data
Non-safety communication	WWW, email, live TV, phone, fax, video-conferencing, file transfer, intelligent travel information, gambling

2.1.2 Services Requirements and Dimensioning

The next step is to derive the individual traffic statistics for the identified service categories. Table 2 contains a list of traffic parameters for possible communication services. The usage parameters are estimated currently. The second column shows how frequently an application may be used. The numbers apply for business travelers. It is assumed that the video conferencing services will only apply to dedicated corporate aircraft. The third column shows the average duration of the usage of an application. The fourth and fifth columns show the bit rates required by the applications. The last column indicates the burst which is defined as peak bit rate divided by the average bit rate. [8]

Table 2 Traffic Characteristics

Service, Application	Application frequency	Mean Holding time	Data rate return link	Data rate forward link	Burst rate
Video surveillance	Permanent	Unlimited	64 kb/s	-	1.0
Aeronautical Surveillance	Permanent	Unlimited	100 bps	100 bps	1.0
Video conference	0.01/flight	15 min	16+384 kb/s	16+384 kb/s	3
Telephony	2/h	3 min	9.6 kb/s	9.6 kb/s	2.857
Video telephony	0.01/flight	5 min	16+64 kb/s	16+64 kb/s	1.0
Shared Applications	0.01/flight	15 min	384 kb/s	384 kb/s	2.5
Email service	5/h	0.25 s	16 kb/s	16 kb/s	1.0
File transfer	5/h	4 s	144 kb/s	144 kb/s	20
Internet	2/h	30 min	16 kb/s	144 kb/s	20

The traffic generated and received by a single aircraft is a function of the distribution of passengers among first, business and economy class, the duration of the flight, the physiological flight time, and the set of available services. The traffic should be described as superposition of the traffic generated by each passenger according to the characteristics of the desired services in terms of data rate and QoS parameters.

When different types of flights are concerned, short and medium haul flights should be focused on needs for business and information type of services. Long haul flights should include also entertainment type of services, in order to offer a complete set of services. The dimensioning of a satellite system providing aeronautical services requires an in-depth analysis of the airline passenger traffic with the region of coverage. Global trends in air-traffic have been identified which allow system-dimensioning activities to be performed such as spot-beam allocation. Europe is and will continue to be the world's largest market for international passenger traffic. Traffic between European, East Asian and North American is and will remain to be a dominant market route. The north Atlantic corridor between the UK and North America is identified as being an important route regarding European international passenger traffic.

The system dimensioning of the satellite system is beyond the scope of this work. We will focus on the traffic management for one aircraft. However, the same scheme and solution can be extended to more complicated aeronautical satellite networks.

2.2 Air Traffic Control and Current Systems

2.2.1 Air Traffic Control

The safety of air travel is ensured by many mechanisms working precisely and cooperatively. Aircraft use navigational equipment and aides and follow Visual Flight Rules and/or Instrument Flight Rules (VFR/IFR) to precisely follow their flight path. Air Traffic Control ensures that no two aircraft have conflicting flight paths. The FAA has established federal airways where the necessary air traffic control is provided for safe air travel. These federal airways consist of necessary ground navigational aids for precise navigation of the aircraft, flight service stations for weather advisories, and radio communication facilities for the air traffic controller-to-pilot communications.

The FAA has established procedures to be followed in these federal airways for air traffic control. Prior to a flight, the aircraft files a flight plan to its departure Air Route Traffic Control Center (ARTCC). The flight plan consists of the requested flight route, the duration of the flight, the requested altitude, etc. The local ARTCC gives clearance to the flight with possible amendments, and changes. When the flight is cleared, the local airport tower controls the departure of the airplane. The tower is responsible for the safe landing and takeoff of the aircraft, as well as safe taxiing on the ground. The airport control tower is also responsible for safe separation of the aircraft within a 5-mile radius of the airport. When the airplane takes off, it communicates its flight information to the departure Terminal Radar Approach Control (TRACON), which relays this information to the local ARTCC. This information is further disseminated to other ARTCC's that are en route of the airplane. The TRACON is responsible of safe separation of aircraft within a 50-miles radius of the airport (also called the terminal area). The air traffic control of an aircraft flying outside a 50-miles radius of its arrival and departure airports is the responsibility of the ARTCC's.

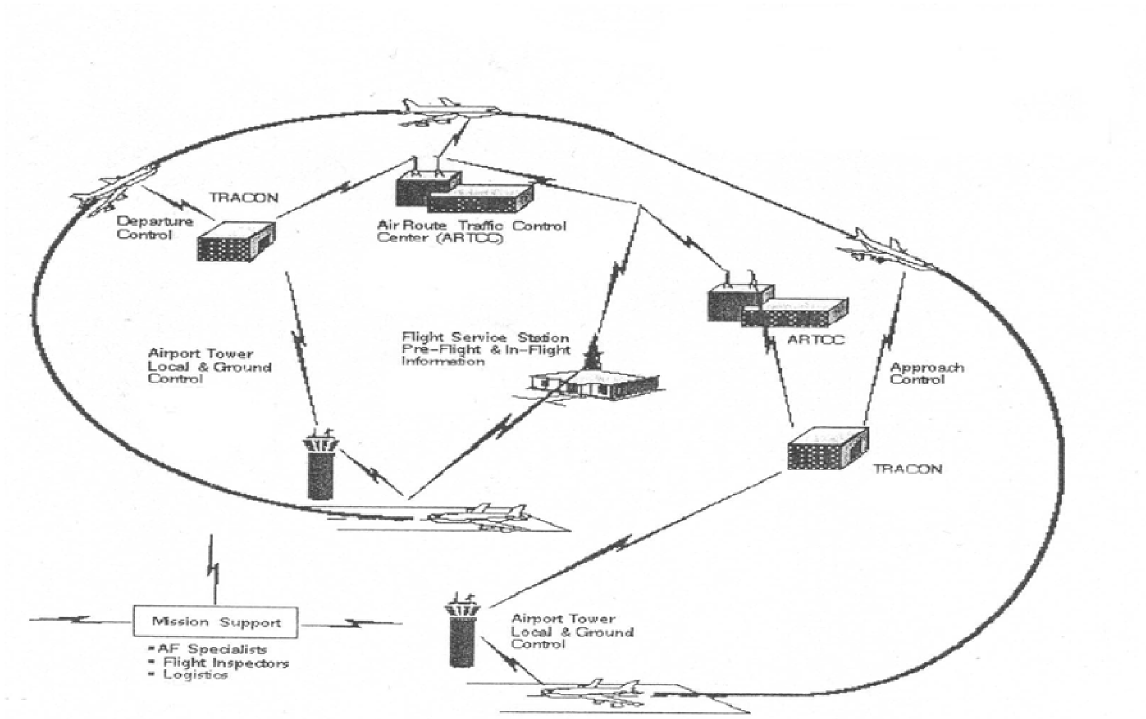


Figure 2 Current NAS Air Traffic Control Structure [7]

Thus, the air traffic control of an aircraft is performed in three stages: Ground and departure control by Air Traffic Control Towers (ATCT), terminal area control by TRACON, and en-route control by ARTCC. There are 21 ARTCC's covering the flight routes in the continental United States. Each ARTCC is responsible for a portion of the airspace. This airspace is further divided into sub-portions, called sectors. There is a controller and fixed radio frequency assigned to each sector. The controllers communicate with the aircraft in their sector via this fixed multi access voice channel, and provide altitude, heading information to the pilots, as well as weather advisories for flight safety. The current communication system is an analog, voice-only system. Communication between the controller at the ARTCC and the pilot is relayed via Remote

Communications Air/Ground (RCAG) stations located throughout the USA. As the aircraft flies on its path, it changes sectors. At every sector change the control of the aircraft is handed over to the receiving controller. The transferring controller provides the aircraft with the new frequency to be used for communication with the receiving controller. As the aircraft approaches its destination airport, the transfer of responsibility is reversed, and the control of aircraft is first transferred to the destination TRACON and then to the airport control tower.

2.2.2 The Current Communication System

The NAS Air-to-Ground communications is supported by:

- 21 Air Route Traffic Control Centers (ARTCC) and 3 Center Radar Approach Controls (CERAP) supported by 793 Remote Communication A/G Facilities (RCAG) and 720 Back-Up Emergency Communication (BUEC).
- 14 Flight Service Stations (FSS) and 61 Automated Flight Service Stations (AFSS) supported by 1854 Remote Communication Outlets.
- 57 Tower Data Link Services (TDLS) air traffic control towers supported by 393 Remote Transmitter Receivers (RTRs).
- 289 non-TDLS air traffic control towers supported by 1029 RTRs.

Every controller is responsible for the separation of the aircraft in his/her sector. The controller of a sector is assigned a fixed 25 kHz AM Double Side Band voice channel for the air traffic control communications with the pilots of the aircraft in his/her sector. The communications between the controller and the pilot of an aircraft is carried out in a party-line/broadcast mode so that all the aircraft in the same sector can monitor all controller-to-pilot communications in that sector. The controllers are located in the

ARTCCs. The A/G communication between the controllers and the pilots is accomplished by over 40,000 radios located at 2,500 different sites. The spectrum that is reserved for air traffic communications is VHF 117.975MHz-137MHz for civilian applications and UHF 225MHz-400MHz for military applications.

The inefficient use of the current radio resources is one of the main reasons for the insufficiency of the current NAS communication system capacity. The inefficiency is caused by both the spectrum inefficiency of the AM system, and the inefficiencies in the operation of this system. In the current voice-only analog system, all airborne and ground users share the same channel. Thus, as the number of users grows, voice congestion increases. Furthermore, the channel may become completely unusable by channel blockage, a problem caused by “stuck microphone” (the switch on the speaker of the radio is left on). The 25 kHz AM-DSB channel is also susceptible to interference, which may cause difficulties for the pilots and the controllers to understand each other. User addressing is verbal through the use of the flight’s call sign. Thus, continuous pilot monitoring is required to identify transmissions directed to the cockpit. The communication structure is also inefficient; for example 1 in 7 ATC messages are hand-off messages exchanged during the change of sectors. All of these problems result in low system message throughput. The AM radio equipment is outdated, and requires high maintenance. The reliability of the overall communications system is high – however, this is mainly due to high redundancy in a system made up of failure-prone components. The high redundancies and maintenance requirements result in a significant financial burden on the part of the FAA, just to keep the system running.

2.2.3 Near Term Plans for Improving ATC A/G Communications

In order to solve the inefficiencies of the current A/G communication system, the FAA is pursuing a modernization program. Under the proposed communications system, ATC A/G communications will evolve from primarily voice to primarily data. Aeronautical VHF radio systems will transition to digital modulation to improve voice quality and to increase channel capacity. VHF resources will be networked to make more efficient use of the resources and to support new capabilities, such as intrinsic backup. Voice communications will continue to be used for some communications such as, emergency or non-routine messages, and for those aircraft that are not data-equipped; AM voice will continue to be supported during the transition period.

The data link is currently envisioned as an extension of the current voice communication system, with applications imitating their voice communication counterparts. Multiple data link standards have been developed: VDL Mode S, VDL Mode 2, and VDL Mode 3. VDL Mode S can provide two-way, domestic ATC communications, cooperative surveillance, and Automatic Dependent Surveillance Broadcasts (ADS-B). VDL Mode 2 uses a Carrier Sense Multi Access (CSMA) 25 kHz channel with 31.5Kbps differential 8-bit Phase Shift Keying (PSK). The capacity of a VDL Mode 2 channel is 2400 bps. This is almost ten times the capacity provided by the current ACARS data link. The VDL Mode 2 standard does not provide priority-preemption-precedence, and cannot guarantee the timely delivery of the message. The FAA plans to replace all aging VHF AM radios with digital NEXCOM radios to increase the radio spectrum efficiency. The data link standard for these radios is VDL Mode 3. NEXCOM radios are based on 25KHz Time-Division-Multi-Access channels that use differential 31.5 kbps 8-bit PSK. The same frequency is used for both uplink and downlink. 3 or 4 time slot schemes may be used, where 3 slots provide long-range interference free communication and 4 slots provides short range interference-free communication. The NEXCOM radios are designed to work both in analog AM and digital TDMA modes. The four time slot scheme may be used both for data and voice in 2 Voice – 2 Data (2V2D), 3V1D, 4V formats. Voice has 4.8 kbps encoding with 250 ms end-to-end propagation delay. Data can be used functionally simultaneously with voice. NEXCOM radios provide priority-preemption-precedence, for both voice and data communications. Voice communication can be provided by push-to-talk action and data communication is done via reservation schemes.

2.3 Satellite Systems

Recognizing the potential for significant improvements in over-ocean coverage afforded by the use of satellite technology for aeronautical communications, the airline industry is developing a design for a global satellite-based communications system to meet the needs of the aviation industry. The expected advantages of the satellite systems for aeronautical communications also include high communication capacity, low message propagation delay, suitability to free flight concepts, and economic benefits.

2.3.1 Satellite System for Aeronautical Communications

In this section we discuss various existing or planned satellite systems and their potential use for aeronautical communications.

INMARSAT-3: Until now Inmarsat has handled the vast majority of satellite-based civil aeronautical traffic, with its four satellites (plus spares) in geostationary orbit around the equator providing the Aero-H, Aero-I and Aero-L range of services. Inmarsat’s equatorial satellite system does not provide polar coverage. Aero-H has 9.6 Kbps, Aero-H+ has 4.8Kbps, and Aero-I has 4.8Kbps RF channel capacity. User data rates of 160-500 bps can be attained for data transfer. TDMA and FDMA multi-access schemes are used for low-rate data communications and high-rate data communications respectively. The cost of the aeronautical Inmarsat equipment exceeds $200,000, and has a calling charge of $5/min. Inmarsat can support 1100 circuits with global beam coverage and 4300 circuits with spot beam coverage.

With its fourth generation of satellites, the Inmarsat I-4, Inmarsat built a Broadband Global Area Network (B-GAN) during 2004. Inmarsat I-4 can deliver Internet and intranet content and solutions, video-on-demand, video conferencing, fax, email, voice and LAN access at speeds up to 432 kbps virtually anywhere in the world via notebook or palm top computers. Interoperability with the current I-3 satellite network is foreseen, thus allowing seamless migration to the new services.

IRIDIUM: IRIDIUM is the first operational LEO system providing narrow-band phone services. It employs a 66 satellite constellation, which can provide 100% coverage. It can provide user data rates of 2400bps without any overhead. IRIDIUM has plans to provide aeronautical service. It has contracted with AlliedSignal for the production of aeronautical terminal equipment, called AIRSAT for large body planes and with Edmo for aeronautical terminal equipment, called SatTalk, which is more suitable for general aviation. The equipment ($3,995 for SatTalk) and the per-minute usage costs (half of Inmarsat as announced by IRIDIUM) are much lower than the current satellite communications equipment. IRIDIUM complies with ICAO AMSS specifications. On August 13th, 1999, IRIDIUM filed for Chapter 11 bankruptcy, since it has failed to promote sales of its satellite phones. Since then, it has been undergoing financial restructuring, and according to these restructuring plans has reduced substantially the price of handsets (from $3500 to $1800) and calls (from $3/min to $2/min).

GLOBALSTAR: Globalstar is the second operational LEO system providing narrow-band phone services. It has officially launched service on October 13th, 1999. Globalstar satellite system has a 48-satellite LEO constellation. The satellites are of bent-pipe type, so global coverage can only be possible with a sufficient number of earth stations. Support for data rates of 9600 bps has been announced. This system has one of the cheapest announced satellite call rates: $0.35/min. The price of terminal equipment for personal users is also quite low: $750. The satellites are not as sophisticated as those of other systems; they do not have on-board processing or inter-satellite links. The call set up delay of Globalstar phones may be as high as 1-2 minutes. It is also stated that, this system can maintain 2000-3000 full duplex circuits per satellite using CDMA. However, Globalstar has not yet announced plans for aeronautical services.

Connexion by Boeing: Boeing recently provided live TV/audio and real-time high-speed Internet (data) services to commercial airlines, business jets and government customers. Rollout started on North American routes on 2001 and expanded to other global flight routes through 2005. Two-way broadband connectivity shall be delivered directly to airline seats to provide passengers with personalized and secure access to the various forms of content via their own laptop. Initially, an asymmetric available bandwidth of 5 Mbps receive and 1.5 Mbps transmit per aircraft is envisaged. Customer airplanes will be equipped with a Boeing proprietary phased array receive and transmit antennas. Connexion by Boeing plans to lease multiple transponders of Loral’s geostationary

Telstar satellite fleet providing C band and Ku band coverage not only over the continental United States, but also over Europe, Asia, South America, northern and South Africa.

Table 3 Key comparison of present and planned systems

	Iridium	Globalstar	Inmarsat-3	Connexion
Satellites	66 LEO	48 LEO	4 GEO	GEO (Leased)
Coverage	Global coverage	No oceans	All major air routes, no polar	All major air routes, no polar
Data Rate	2.4-10kpbs	9.6kbps	4.8-64kbps	1-2,5-10Mbps
System Capacity	<174Mbps	<450Mbps	<500Mbps	
Regulatory Status	L-band, No Certified	L-band, No certified	L-band, FCC certified	Ku-band, FCC exclusion
Call Charge	Half of Inmarsat	$0.35/min	$5/min	
Equipment Cost	$4,000	$750	$200,000	Double of Inmarsat

2.3.2 Research Issues for Aeronautical Satellite Systems

Aeronautical Communications Systems: The future satellite aeronautical communications systems must evolve with the overall NAS architecture. Due to the large variety of the users with different needs, the ground communication infrastructure will have to be supported for the foreseeable future. Therefore, any improvements in the NAS should consider hybrid terrestrial/satellite communications architecture. The primary users of the NAS will not accept rapid large-scale changes in the operations of the air traffic services. It is unreasonable to assume that a plan to totally transition the aeronautical communications infrastructure to satellite system would ever be adopted in the near term. For a possibly very long transition period, air-ground communications will be supported by hybrid terrestrial/Satellite architectures. Researches in this area include definition and analysis of hybrid terrestrial/satellite architectures, investigation of how to seamlessly integrate terrestrial and satellite systems, and analysis of transition strategies.

Next Generation Satellite Systems: The satellite systems have evolved from analog transmission modes to using digital messaging techniques for communication. They use such sophisticated techniques as spot-beams, frequency reuse, inter-satellite links, on-board processing, TDMA and CDMA techniques and their constellations lie not only in geosynchronous orbits but also in low and medium orbits. LEO satellite systems offer significant advantages over GEO systems for the delivery of mobile satellite services. GEO satellite systems are best suited for their missions of high-speed data, television transmission and other broadcast applications and various broadband applications. Currently various existing or planned satellite systems are now used for aeronautical communications on a limited basis. They are INMARSAT, BOEING, IRIDIUM, GLOBALSTAR, ICO-TELEDESIC, and so on. Researches in this area include system and overall architecture design, estimation of cost and spectrum requirements, and providing reliable communications for remote/oceanic areas.

Region of Interest: The main congestion problem for air-ground communications is experienced at the terminal areas. The need for new channel assignments will persist as the number of flight increase. En route communications is currently supported by a very expensively maintained and geographically dispersed large ground infrastructure. Terminal area communications capacity could be enhanced by a significant diversion of en route communications to satellite systems. An aircraft could be interested in three different region of interest (ROI): a tactical ROI, a near-term strategic ROI and a far-term strategic ROI. At certain times during a flight the pilot and aircraft control systems could be interested in any one of the three ROIs, while at other times, there would be only one of interest. We need to model the requirements of each separately. Fundamental questions include: how big is each region of interest? What are typical aircraft densities in each region? What is traffic load?

Communication requirement: Although currently the risk of an aircraft accident is quite low, as air traffic continues to increase, the expected number of accidents could reach unacceptable levels even though the underlying accident risk remains constant. Worse yet, it is possible that as traffic levels increase, the accident risk also increases due to increased congestion. Thus, the underlying communication, navigation and surveillance systems, which support the future NAS, must provide for greater capacity, but at the same time satisfy stricter safety performance criteria.

Future ATC regimes envision new forms of air traffic such as free flight and the use of smaller aircraft that utilize many smaller airports widely dispersed around the country. Such changes are seen as necessary to increase the capacity of the NAS. Ensuring passenger safety and system wide performance will therefore require new and better forms of communications, navigation and surveillance. Each aircraft embarking on such a flight path must have sufficient information regarding the flight paths of other aircraft as well as access to relevant weather information. The goal of aeronautical communications includes Safety, Accessibility, Flexibility, Predictability, Capacity, Efficiency, and Security. Research is needed to define communication requirements of aeronautical systems, investigate the performance of satellite systems.

Layered protocol support: Since ATN was developed specifically for aeronautical communications it provides the necessary QoS in terms of priority, precedence and preemption. On the other hand, because it is highly specialized, it appears that ATN-based products will be very expensive. The Internet protocols appear to provide most of the required features necessary to support aeronautical communications and it represents a much more cost effective solution. However, the significant signal propagation delay of satellite link could pose problems, and if protocols such as TCP/IP need to be supported, appropriate modifications are needed for their operation to be more efficient. The goal of research in this area is to investigate the possibility of utilizing Internet protocols for purposes of the NAS in addition to the ATN system, analyzing the performance of TCP over satellite systems.

Chapter 3: Transport Protocol for Aeronautical Satellite Communications

3.1 TCP operations

A common way to characterize the performance of an access network is in terms of the throughput observed by the applications running above the TCP layer. The throughput achieved depends on three facts: the bandwidth available for data and acknowledgments, the packet loss rate, and the specific TCP implementation. In this section, we introduce a brief description of the TCP functions relevant to our work. Then we discuss the two main problems by running TCP over aeronautical satellite networks.

Current TCP implementations, TCP-Reno, contain a number of algorithms aimed at controlling network congestion. These algorithms include slow start, congestion avoidance, fast retransmit and fast recovery [9][10]. Together they define the congestion window, *cwnd*, as an estimate of the maximum number of packets that can be sent without receiving any acknowledgement (ACK). While the receiver's advertised window, *rwnd*, is used to guard that the sender will not overflow the receiver buffer, the *cwnd* is used to guard that the sender will not overload the network. The TCP sender never sends more than the minimum of *cwnd* and *rwnd* window worth packets without receiving any acknowledgement.

The TCP sender operates in one of two modes: slow start or congestion avoidance. The sender determines its mode based on the values of *cwnd*. As long as the *cwnd* is smaller than a slow start threshold, *ssthresh*, the sender works in slow start mode. When *ssthresh* is reached, the sender switches to congestion avoidance. During slow start the sender starts with a congestion window of one packet and grows it by one with every acknowledgement received. Assuming an acknowledgement is sent for every data packet received, which is not the case when the receiver uses delayed ACKs, this results in doubling *cwnd* every round trip and increasing *cwnd* exponentially. While in congestion avoidance mode, the sender increases the value of *cwnd* by 1/*cwnd* for every data packet received, which is approximately equivalent to an increase of one packet every round trip time (RTT), yielding linear growth.

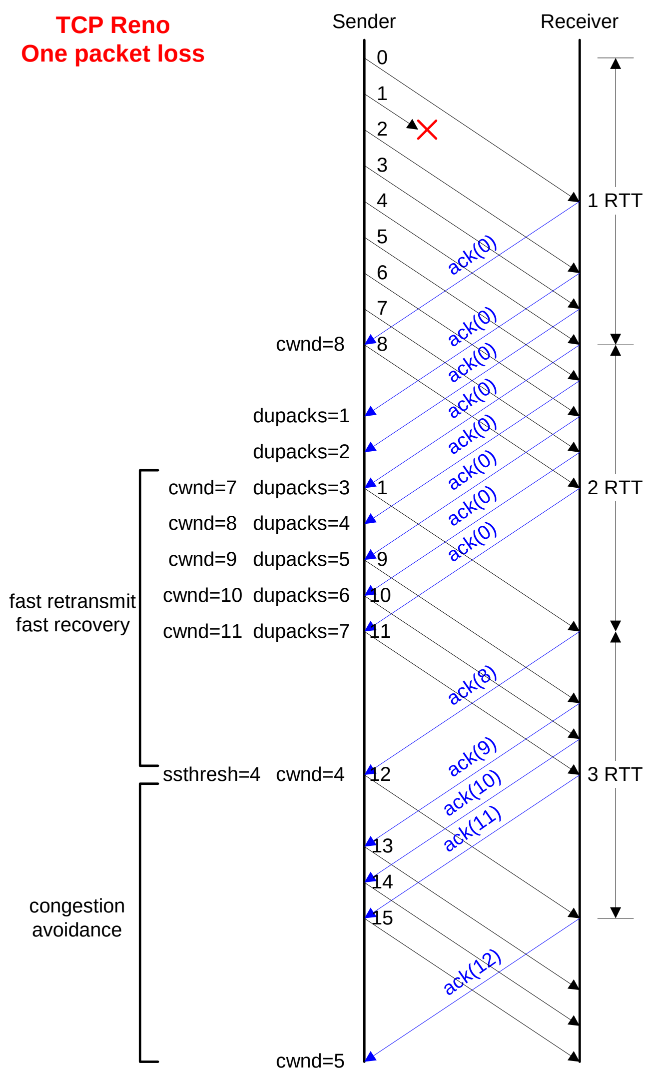


Figure 3: TCP-Reno in the presence of one packet loss

When a packet is lost, the subsequent packets are received out of order. An out of order packet triggers a duplicate ACK (dupACK), which carries the same sequence number as a previous acknowledgement. When the third dupACK is received, the sender assumes a packet has been lost and enters fast retransmit mode. TCP transmits the potential lost packet indicated by the ACK and cuts its *cwnd* to half, as depicted in Figure 3. After that it inflates its *cwnd* by one packet when a dupACK is received. If there is one and only one packet lost in a single window, the inflation can increase the *cwnd* to the original *cwnd* before the loss after about half RTT. After that TCP can send a new packet when each dupACK is received if allowed by *rwnd*. Finally it will send half a window new packets when it receives the first non-duplicate ACK. After receiving an ACK for the retransmitted packet, the sender performs a procedure called fast recovery by shrinking *cwnd* to half and entering congestion avoidance mode.

The fast retransmit mechanism is not always triggered when a packet is lost. As a simple example, consider the case where the window size is only 4, in which case only two dupACKs can be received. To detect a packet loss even in this case, the sender maintains a retransmission timer. This timer is set when TCP sends data, but only if the timer is not currently enabled. The retransmission timer is turns off when an ACK for all outstanding packets is received. If only part of the data is acknowledged, the timer is restarted. When the timer expires, the oldest packet for which an ACK has not been received is retransmitted. The time the sender is idle, waiting for the timer to expire, is called a timeout. After a timeout, the sender retransmits the lost packet, shrinks *cwnd* to 1 and enters slow start. In addition, it sets *ssthresh* to half the value *cwnd* had when the loss was detected. Thus timeouts have a significant impact on throughput both because they introduce a period of idle time and because they shrink the window.

3.2 TCP Problems in Satellite Networks

Aeronautical satellite networks have several characteristics that differ from terrestrial channels. Two main characteristics may degrade the performance of TCP: long round trip delay and burst losses.

The first problem is long round trip delay. A Geosynchronous Earth Orbit (GEO) satellite is about 36,000km above the earth. At this altitude the orbit period is the same as the earth's rotation period. Therefore, each ground station is always able to "see" the orbiting satellite at the same position in the sky. The propagation time for a radio signal to travel twice that distance is about 240ms (corresponding to a ground station directly below the satellite). Therefore, the propagation delay for a round trip time (for a message and the corresponding reply) is about 560ms, including 80ms RTT for typical terrestrial Internet delay. The RTT will be increased by other factors in the network, such as the transmission time and propagation time of other links in the network path and queuing delay in networks.

In the beginning of a new TCP connection, the sender executes the Slow Start algorithm to probe the availability of bandwidth along the path. The time taken by TCP slow start to reach the satellite bandwidth, *SatBW*, is about [11]

$$t_{SlowStart} = RTT * (1 + \log_2 (SatBW * RTT / l))$$

Where RTT is the round-trip time and l is the average packet length expressed in bits. This equation is satisfied when every TCP segment is acknowledged. For a connection with large RTT, it spends a long time in slow start before reaching the available bandwidth. For short transfers, they could be finished in slow start, which obviously does not use the bandwidth efficiently.

Large RTT also introduces large bandwidth delay product (BDP) for satellite link. The BDP defines the amount of data a protocol should have “in flight” (data that has been transmitted, but not yet acknowledged) at any time to fully utilize the available channel capacity. However, the receiver advertised window, *rwnd*, is 16 bits in the TCP header. This window cannot be more than 64K bytes, which limits the two-way system throughput to 64KB/560ms, 117KBps. Window Scaling [12] is proposed to solve this problem. But when the window is large, it is more likely that multiple packets are lost in one window caused either by congestion or link layer corruption. The multiple losses will trigger TCP congestion control algorithms and lead TCP to actually operate with a small average window.

Large RTT also lead to large timeout value because the timeout value is calculated dynamically, based on the round trip time measurements the sender performs throughout its operation [6]. If a loss can not detected by fast retransmit, the sender is idle waiting for the timer to expire, and operates below its optimum speed a few round trip times. Afterwards during slow start mode, the sender works with a window smaller than 4 for 2 round trip times, even a single packet loss may cause a timeout.

The second problem is burst losses. Communications over satellite links are often characterized by sporadic high bit error rates and burst losses. This is especially true when working in the Ka band (30/20 GHz), where weather conditions greatly affect link availability.

TCP uses all packet drops as signals of network congestion and reduces its window size in an attempt to alleviate the congestion. In the absence of knowledge about why a packet was dropped (congestion or corruption), TCP must assume the drop was due to network congestion to avoid congestion collapse. Therefore, packets dropped due to corruption cause TCP to reduce the size of its sliding window, even though these packet drops do not signal congestion in the network. After any retransmission, whether following a timeout or following fast transmit, the sender shrinks its transmission window to one or to half its original size, respectively. Thus following a loss the sender operates below its optimum speed for a few round trip times. If losses occur at the time the window is growing back towards its optimal size, they lower the window yet again. Moreover if a loss occurs while the window grows in slow start, the growth rate turns from exponential to linear, and it takes even longer for the window to reach the optimal value.

Burst losses are more likely lead timeouts. The probability of a timeout increases with the packet loss rate because for high loss rates, the probability of losing several packets in the same window, which usually leads to timeouts, increases. TCP does not perform well under burst losses. The mechanism for efficient recovery of lost packets, e.g. fast retransmit, fails when several consecutive packets are lost, drastically affecting the throughput. For TCP Reno, in order for the fast retransmit algorithm to recover the loss, the congestion window size has to be greater than four for single packet loss and has to be greater than ten for two consecutive losses in one window. While for three or more consecutive losses in one window, the TCP sender has to wait for timeout to recover the loss [13]. TCP New-Reno [14] can avoid many of the retransmit timeouts of Reno when a large number of packets are dropped from a window of data. However, New-Reno can only recover one lost packet during each RTT. TCP SACK [15] can convey information about non-contiguous segments received by the receiver in the acknowledgements so that the sender can recover error much faster than TCP Reno and New-Reno.

3.3 Related work of TCP over satellite network

The proposed TCP solutions for satellite environment can be categorized into four classes: End-to-end enhancements, TCP connection splitting, Rate based solution, and link layer solution. All these proposals are not independent of each other. A better solution may combine some of them and comes up with a new protocol. Some solutions are designed for specific networks and may not work well in other networks. We will discuss some of the proposed solutions related to our work, specially the TCP splitting solutions.

TCP enhancements TCP enhancements include large initial window [16], delayed ACKs after slow start [17], TCP for transaction [18], selective acknowledgement [15], and forward acknowledgement [19]. All these enhancements are end-to-end solutions. They only need to be implemented at the end nodes, rather than at every route in the network. However, based on the simulation on [20], it is difficult for an end-to-end solution to solve these problems in the hybrid satellite networks effectively.

TCP Spoofing TCP spoofing for Internet over satellite was first conceived, developed, implemented, and commercialized by Hughes Network System in a series of papers by Baras and his group [21][22][23][24][25][26]. A router near the source sends back ACKs for TCP segments in order to give the source the illusion of a short delay path. TCP spoofing improves throughput performance but has some problems. The router must do a considerable amount of work because it becomes responsible for the correct delivery of the TCP segments it acknowledges to the source. Spoofing requires ACKs to flow through the same path as data. On contrary, in Internet it is very common that ACKs flow through a different path than data. If the path changes or the router crashes, data may get lost. If IP encryption is used, this scheme cannot be applied.

I-TCP [27]. I-TCP stands for Indirect TCP and is mainly designed for mobile Network. The basic idea of indirect TCP is that the end-to-end TCP connection is now divided into two connections, one is from the server to the base station and another one is from the base station to the mobile users. The base station sends premature acknowledgements to the server and takes responsibility to relay the data to the mobile host reliably. The advantages are the separation of flow control and congestion control of wireless and wired network, resulting in faster reaction to link layer loss.

Super TCP [13] Because satellite channel is a FIFO channel, out-of-order routing and congestion on the satellite link are impossible. Super TCP uses one duplicate ACKs to trigger the retransmission at the base station and to use a fixed window size for the satellite TCP connection. It also proposes a new sender algorithm using the same idea as in TCP new Reno. It uses partial ACKs to calculate the burst loss gap and sends all the potential loss packets beginning from the partial acknowledgement number. It is possible that the sender could retransmit packets that have already been correctly received by the receiver.

SCPS-TP [28]. Space communication protocol standards-transport protocol (SCPS-TP) is a set of TCP extensions for space communications. This protocol adopts the Timestamps and window scaling options in RFC1323. It also uses TCP Vegas low-loss congestion control mechanism. SCPS-TP receiver doesn’t acknowledge every data packet. ACKs are sent periodically based on the RTT. The traffic demand for the reverse channel is much lighter than in the traditional TCP. However it is difficult to determine the optimal acknowledgement rate and the receiver may not respond properly to congestion in the reverse channel. Because there is no regular acknowledgement-driven clock, it uses an open-loop rate control mechanism to meter out data smoothly. To transmit data continuously in the presence of link layer loss rather than congestion loss is especially important. SCPS-TP uses selective negative acknowledgement (SNACK) to address this problem. SNACK is a negative acknowledgement and it can specify a large number of holes in a bit-efficient manner.

RWBP [29]. Receiver Window Backpressure Protocol (RWBP) is a connection splitting based solution with new congestion control and error control algorithms for direct-to-user hybrid satellite networks. RWBP cancels all the congestion control algorithms in TCP and uses per-flow queuing, round robin scheduling and receiver window backpressure for congestion control. The round robin scheduler at the satellite gateway is used to send packets for all TCP connections to achieve fairness and efficiency. In RWBP, error control uses the same idea as TCP SACK, where multiple packet losses in one window can be fast retransmitted based on the SACK information. However, the round robin scheduler as a centralized controller puts extra processing overhead on the satellite gateway since the gateway need to handle numerous simultaneous connections. Also the buffer allocation scheme for flow control in RWBP is depended on the link bit error rate, which may not be available or not accurate at the time of connection setup.

3.4 Motivation and basic idea

The related work above tries to solve some of the TCP problems in the satellite data networks. But a scheme that solves all the problems does not exist yet and the proposed solutions may not work well in some other networks. In addition, all TCP proposals in the literature are not independent of each other. A better solution can combine some of them and come up with a new protocol for specific satellite networks. The problems of Internet over satellite are far from being solved. In this study, we propose a scheme, which takes into account the characteristics and requirement of aeronautical satellite networks. Our scheme shows significant improvements in terms of efficiency and link utilization.

In the aeronautical satellite network as in Figure 1, the client (passenger) on the aircraft accesses an Internet server over bent pipe GEO satellite. We focus on the transport layer protocol design and assume the point-to-point link between ground gateway and satellite gateway. The terrestrial link between the server and the ground gateway is actually a path through routers in the Internet with typical Internet delay and very low bit error rate. The aircraft link between the client and the aircraft gateway are wired/wireless LAN connection with very small delay and very low bit error rate. In order for this hybrid TCP/IP network to be commercially deployable, it must seamlessly interoperate with existing TCP/IP networks. The following two requirements must be satisfied. First, both the Internet servers and clients on aircraft must use standard TCP/IP protocol. Most of the passengers (especially business travelers) want to connect to the communication network with their own equipments (such as laptops, PDAs) because they are used to from their daily life. Also it is not possible to implement or change the protocol at every server in the Internet. Second, we need to provide high utilization of the satellite channel. Satellite bandwidth is still a scarce resource compared to the bandwidth provided by optical fibers in the terrestrial networks. Therefore we assume the satellite link is the bottleneck of the system and the terrestrial networks have enough bandwidth.

As stated in section 3.1, satellite TCP connections need large windows to fully utilize the available bandwidth. However it takes much longer for satellite TCP connections than for terrestrial TCP connections to reach the target window size because of the large propagation delay and the slow start algorithm in TCP. And the window multiplicative decrease strategy makes the hard gained large TCP window very vulnerable to congestion. The misinterpretation of link layer corruption as congestion makes this situation even worse. In the best case, the packet loss does not cause timeout and TCP can stay in congestion avoidance phase rather than in slow start, the additive increase strategy makes the window to grow very slowly. From the above observations, we can see that it is difficult for satellite TCP connections to actually operate with large windows.

Based on the fact that the end-to-end schemes cannot solve these problems very effectively, we propose a connection splitting based scheme. The idea behind split connections is to shield high-latency or noisy network segments from the rest of the network, in a manner transparent to applications. Figure 4 illustrates the general split case, in which an end-to-end TCP connection is split into 3 connections at the aircraft gateway and ground gateway. One connection is from the Internet server to the ground gateway, the second one is from the ground gateway to the aircraft gateway, and the third one is from the aircraft gateway to the client in aircraft. We consider the data transfer from the Internet servers to the client in aircraft. Ground gateway sends premature acknowledgements to the Internet servers and takes responsibility to relay all the acknowledged packets to the aircraft gateway reliably. The aircraft gateway does the same job to relay the data to the client. For the satellite link between the ground gateway and the aircraft gateway, a satellite optimized transport protocol can be used.

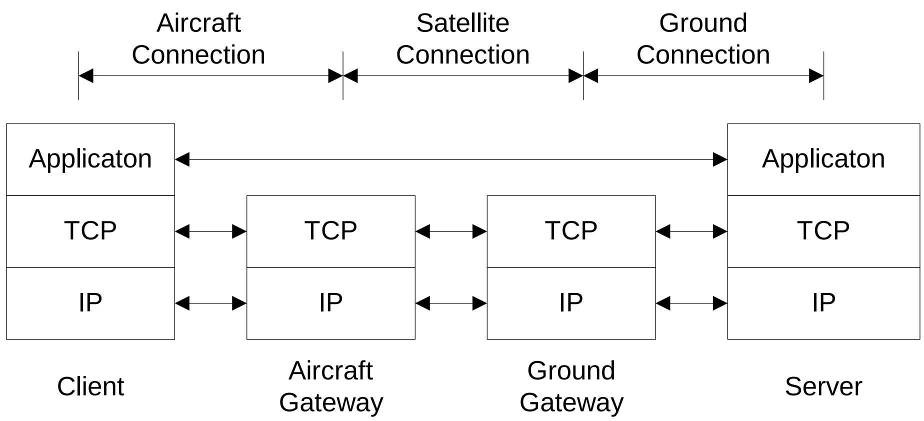


Figure 4: Protocol stack for a split connection configuration

One advantage of the split connection approach is that it separates the losses on the satellite link from the losses on the Internet, allowing local recovery of lost packets. Another advantage of the split connection approach is that it allows tailoring the TCP implementation on each of the connections to best suit the characteristics of the underlying channel. The disadvantage is that the splitting scheme violates the end-to-end semantics of TCP. In splitting TCP, it is possible the sender receives an acknowledgement of a data packet while the data packet has not reached the destination rather is buffered at the gateway. However, many applications such as FTP and HTTP use application layer acknowledgements in addition to end-to-end TCP acknowledgements. Using splitting protocol for these applications does not comprise end-to-end reliability.

3.5 TCP splitting Protocol

3.5.1 Connection Splitting

As we discuss above, the end-to-end TCP connection between the client on aircraft and the Internet server on ground is split into three connections: the ground connection between the Internet service and the ground gateway, the satellite connection between the ground gateway and the aircraft gateway, the aircraft connection between the client and the aircraft gateway. Both the ground connection and the aircraft connection have much large link bandwidth with very low bit error rate. Therefore the satellite link is the bottleneck. A satellite optimized transport protocol can be used for satellite TCP connection, while standard TCP protocol are used for other two connections. In this way, high utilization of the satellite link can be achieved, while there is no any change to the protocol stacks at end hosts.

The advantage of splitting the TCP connection is that the satellite channel is isolated from the rest of the Internet. This channel has two unique properties, which differentiate it from the rest of the Internet. The first property is that packets sent on the satellite channel cannot be routed out of order. The second property is that congestion is not possible for the satellite channel if we design the congestion avoidance schemes for the gateways carefully. Therefore the only reason for packet losses is transmission errors. Both properties are attributable to the fact that there are no any other routers on the link between the ground gateway and the aircraft gateway.

The above observations motivate us to design more efficient and effective congestion and error schemes with our specific network characteristics in mind. We design a new TCP splitting protocol, called Aeronautical Transport Control Protocol (AeroTCP), for the satellite connection. The main idea is to design specific flow control, congestion control, and error control mechanisms for satellite TCP connections based on the properties of the satellite channel. This implementation of this idea will be discussed in details in the following sections.

3.5.2 Flow Control and Buffer Allocation

TCP uses flow control to ensure that the sender will not overflow the receiver's buffer. For every TCP connection as in Figure 5, all packets waiting for transmitting or received are buffered at the send buffer or receive buffer, respectively. Consider the traffic from the Internet server to the client on the aircraft (all data packets flow on the upper half path on Figure 5, the low path is only for ACKs), all the TCP packets received from the server are forwarded to the TCP received buffer of the ground connection and they are moved from the receive buffer to the send buffer in sequence at the ground gateway. Then the packets are sent from the send buffer to the aircraft gateway over the satellite link. At the aircraft gateway, the packets are moved from receive buffer of satellite connection to send buffer of aircraft connection in sequence. Finally, the packets are sent from the send buffer to client over aircraft connection.

Flow control is done between the ground gateway and the aircraft gateway at the transport layer by using the receiver's advertised window, *rwnd*. For each satellite TCP connection, the aircraft gateway advertises a receiver window based on the available receive buffer space for that connection just as in TCP. Window scaling can be used here to advertise large windows. At the ground gateway, when packets are moved from ground connection receive buffer to satellite connection send buffer in sequence, a blocking write is performed so that the send buffer will not overflow. For ground TCP connection and aircraft TCP connection, standard TCP flow control is used so that the sender will not overflow the receiver's buffer. In this way, the traffic load at the satellite connection is back pressured to the receive buffer of the ground connection. When the traffic load on the satellite connection increases, the buffer begin to be filled up and a smaller receive window is going to be sent to the server. When the traffic load decreases, the buffers begin to be emptied faster and larger advertised receiver windows are sent to the server so that the server can speed up.

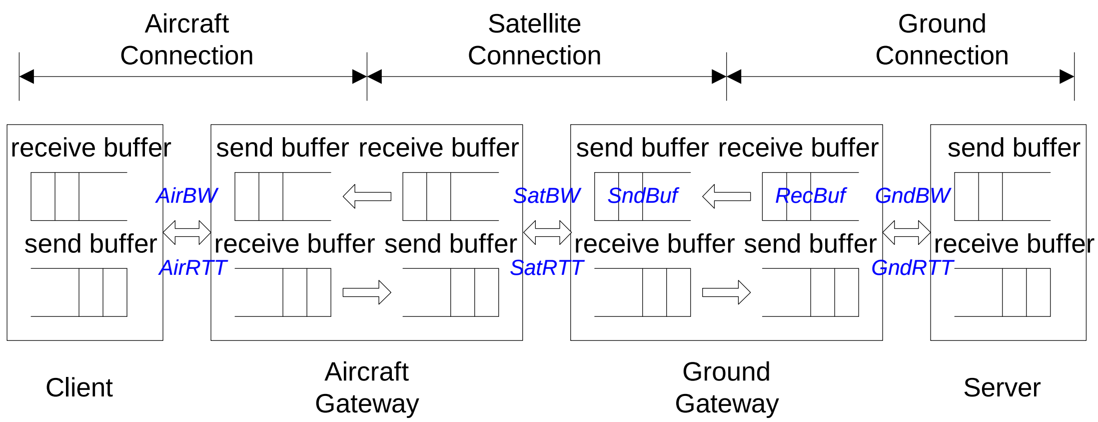


Figure 5: TCP flow control and buffer allocation

Buffer allocation algorithm for Iternet over satellite gateway was investigated in [30]. The buffer size assigned to each connection at the satellite gateway and aircraft gateway has a direct impact on the end-to-end TCP throughput. Although memory is cheap, infinite buffer for each connection cannot be assumed because the satellite gateway is designed to support a large number of connections. Based on the observation that the number of TCP connections is small at the client and the server compared to that at the gateways, we assume large buffer is available for each TCP connection at those end hosts. Following we will discuss the buffer allocation at those gateways.

Assume that the traffic is from Internet server to the client on aircraft. The bandwidth of the ground link and the aircraft link is much larger than the satellite link. Assume there is only one connection in this system and the satellite link is error free. At the ground gateway, the send buffer of the satellite TCP connection is *SndBuf* and the receive buffer of the ground TCP connection is *RecBuf*. The effective satellite bandwidth, which is the raw satellite bandwidth deducted by the protocol headers, is *SatBW*. The round trip time is *SatRTT* for the satellite connection and is *GndRTT* for the ground connection (refer to Figure 5). When the system reaches the steady state, the input rate of the queue at the ground gateway should be equal to the output rate of the queue. The maximum achievable throughput of the end-to-end connection is [29]

$$Throughput_{\max} = \min(SatBW, \frac{SndBuf}{SatRTT}, \frac{\mathrm{RecBuf}}{GndRTT})$$

From the above analysis, we can see that the buffer size can become the bottleneck of the end-to-end TCP performance if it is less than the bandwidth delay product (BDP). However when the buffer size is greater than the bandwidth delay product, that is

$$SndBuf >= SatBW * SatRTT$$

$$RecBuf >= SatBW * GndRTT$$

There are packets backlogged at the satellite gateway and these backlogged packets cannot contribute to the throughput and only increase the queuing delay. The same analysis applies to the aircraft gateway.

When there are multiple connections in this system, the bandwidth available to each connection is a function of the number of connections and their activities. Although the average number of active connections is large, the variance is small. The bandwidth available to each connection does not vary dramatically. For simplicity, we assign each connection a static peak rate (PR), which is the maximum bandwidth it can achieve and is much smaller than the total satellite bandwidth. The buffer size is set to peak rate delay product (PRDP).

When the satellite link is error free, the buffer sizes allocated above are enough to achieve the target peak rate. However when the satellite link is not error free, changes need to be made at both ground gateway and aircraft gateway. When a packet is corrupted, the aircraft gateway has to buffer the out of order packets because the receiver of the satellite TCP connection only forwards in sequence packets. In order to keep the advertised receiver window open so that the sender of satellite connection can send new packets during the error recovery, the aircraft gateway needs a buffer size larger than the peak rate delay product (PRDP) to achieve the peak rate. The error control algorithm (will be discussed below) can recover multiple packet losses within one window in one RTT. If the error rate of the satellite link is low so that corrupted packets can be recovered in one RTT, receive buffer size about two times of the peak rate delay product should be provided. If the error rate is very high, retransmissions of the corrupted packets can be lost again. In this case, receiver buffer size should be about three to four times of the peak rate delay product. The same buffer allocation scheme should be used for send buffer of satellite TCP connection. For the send buffer at the aircraft gateway, only one peak rate delay product is enough since the aircraft TCP connection has very short delay and very low bit error rate. For the same reason, the receive buffer at the ground gateway only need one peak rate delay product buffer size to achieve the target transfer rate. However, we set this buffer to twice of peak rate delay product. This is because 1), the ground link has some round trip delay and low bit error rate. Although they use standard

TCP protocol, large buffer can ensure the data flow when there are packet losses. 2), we want the server to send little more data packets to the satellite gateway so that there are some packets backlogged at this buffer. Whenever the satellite TCP connection recovered from packet losses, it will not get starve for new packets to send.

The buffer allocation scheme for both the ground gateway and the aircraft gateway are summarized in Table 4. Please note that last row is for the normal case with multiple TCP connections and with some link error.

Table 4: Buffer Allocation for gateways

Sat. Link	# of Conn.	Aircraft Gateway		Ground Gateway	
		SndBuf	*RecBuf*	*SndBuf*	*RecBuf*
No Error	One	*SatBW*AirRTT*	*SatBW*SatRTT*	*SatBW*SatRTT*	*SatBW*GndRTT*
	Multi	*PR*AirRTT*	*PR*SatRTT*	*PR*SatRTT*	*PR*GndRTT*
Error	One	*SatBW*AirRTT*	*2~4*SatBW*SatRTT*	*2~4*SatBW*SatRTT*	*2*SatBW*GndRTT*
	Multi	***PR*AirRTT***	***2~4*PR*SatRTT***	***2~4*PR*SatRTT***	***2*PR*GndRTT***

3.5.3 Congestion Control and Bandwidth Allocation

The above flow control and buffer allocation scheme can ensure that each TCP connection will be able to achieve its target peak rate. However, without additional congestion control algorithm, it cannot guarantee network stability and fairness among TCP connections. There are two reasons: 1) It is difficult to allocate the buffer size for each TCP connections based on link conditions and number of connections because the link bit error rate is difficult to measure or estimate accurately. 2) The buffer size corresponding to that peak rate is set when the connection is initialized. They are difficult to change for active connection when the number of connections increases or decreases. Thus the buffer allocation could cause problems such as underutilization of the satellite link, unfairness and un-scalability.

Ott in his 1996 paper [30] derives the stationary distribution of the congestion window size for idealized TCP congestion avoidance. The main results are that if every packet is lost with a small probability p, average window size and long range throughput are of the order of $1/\sqrt{p}$. Consider transfer a large file from the server on the ground to the client on the aircraft with packet loss rate of p and p' for satellite link and ground link, respectively, the end-to-end throughput is limited due to large delay and packet loss:

$$ETE\_r = \frac{Const * MSS}{(SatRTT + GndRTT) * \sqrt{p + p' - pp'}}$$

Use TCP splitting doesn't help that much since the satellite link has long delay and high bit error rate. The throughput in satellite link will limit the end-to-end throughput. It is possible to achieve high utilization if we have multiple TCP connections with throughput ETE_r_i for connection i. However, it is still difficult to achieve the link bandwidth. In addition, since the RTT and packet loss rate may be different for each connection; the end-to-end throughput is different, which means they cannot share the bandwidth fairly.

$$ETE\_r_i = \frac{Const * MSS}{(SatRTT + GndRTT_i) * \sqrt{p_i + p_i' - p_i p_i'}}$$

TCP congestion control algorithms can guarantee network stability and fairness among TCP connections in terrestrial fiber networks, but it is not efficient and effective in satellite networks. In our aeronautical networks, assume that the satellite link bandwidth to be shared among them is fixed and known. Also assume that the number of connections and the traffic arrival pattern are known. All this information is available at the satellite gateway. Therefore there is no need to use slow start to probe the bandwidth and use additive increase and multiplicative decrease congestion avoidance to guarantee fair resource sharing as in the distributed case. In our scheme, we cancel the congestion control algorithms in TCP. However congestion window is still used to guide the transmission of each TCP connection.

If there is only one connection, this TCP can send packets to IP layer at the rate corresponding to the satellite bandwidth. As long as there are packets in the send buffer of the ground gateway and the receiver's window allows, the satellite link can be fully utilized. However, when there are multiple connections, without a fair queuing scheduler, the congestion window is used to allocate satellite bandwidth and achieve fair sharing. We will discuss the static bandwidth allocation and adaptive bandwidth allocation scheme as following. More advanced queuing management and congestion control scheme for integrated services will be left as future work.

The simplest scheme of bandwidth allocation is static bandwidth allocation. Each TCP connection is assigned the same congestion window. If there are N connections in the system with satellite bandwidth of *SatBW*, each connection is allocated *SatBW/N* to ensure the fair sharing. The congestion window is set as *SatBW/N*SatRTT*. The conservative scheme is to use the maximum number of connections for the congestion window. However, when the number of connection is much smaller than the maximum number, the satellite link is underutilized. Another scheme is to use the peak rate delay product (PRDP) for congestion window based, however, this may cause buffer overflow to the lower layer of gateways at heavy load.

The congestion window, thus the bandwidth allocation for each connection, can be set adaptively. To achieve fairness, each connection will need *SatBW/N*SatRTT* for its congestion window. This window size will increase when the number of connection N decrease. When some connections are closed, the total number of connection is decreased. Other TCP connection will increase its congestion window and speed up. However, this scheme requires that the number of connections is small and changes very slow such that other connections could catch up with the change in time. This is perfect for video or ftp applications with long connection duration. Another adaptive bandwidth allocation is set the congestion window based on the measurements of the traffic characteristics and the target satellite link utilization. The target bandwidth for each connection may be changed from time to time, but it could be computed and stored in a predefined table. Whenever a new connection is initialized, it could setup its congestion window by looking up the table. This way, we can guarantee fair bandwidth sharing without a fair queuing scheduler.

3.5.4 Super Fast Error Control

The satellite link in Ka band is often characterized by high bit error rate and burst losses because the weather conditions greatly affect link availability. The burst losses normally cause TCP retransmission timeouts and significant throughput degradation. In this section, we first discuss the behavior of TCP Reno and TCP New-Reno in the presence of burst losses and then present our super fast error control algorithm for AeroTCP. This error control algorithm can recover multiple packets in a burst in one RTT while Reno and New-Reno can only retransmit at most one dropped packet per RTT.

TCP depends on duplicate acknowledgements and timer for error control. As we discussed in section 3.1, the TCP Reno [10] does not perform well under burst losses. The fast retransmit algorithm is triggered after three duplicate acknowledgements are received. In order for the fast retransmit algorithm to recover the loss, the congestion window size has to be greater than four for single packet loss and has to be greater than ten for two consecutive losses in one window. While for three or more consecutive losses in one window, the TCP sender has to wait for timeout to recover the loss.

Figure 6 shows TCP Reno with two consecutive losses. In the following discussion, we assume the sender uses its window fully. We also assume that the receiver's advertised window, *rwnd*, is large and the transmission of packets is controlled only by congestion window, *cwnd*. At the beginning, packets 1-8 are sent as the sending TCP's *cwnd* is 8. However, packets 1 and 2 are lost. After receiving the first ACK for packet 0, the sender receives 6 additional ACKs for packet 0 corresponding to the receiver's successful receipt of packet 3-8. The third duplicate ACK of the sequence (the fourth ACK for packet 0) meets the duplicate ACK threshold of three, and triggers a retransmission of packet 1. Then the sender goes into fast recovery and reduces its *cwnd* and *ssthresh* to 4. During fast recovery, receipt of the fourth dup ACK brings the usable *cwnd* to 7, and by the 6th dup ACK, the *cwnd* reaches 10. The "inflated" window from the last 2 dup acks allows the sender to send packets 9-10. Upon receiving the ACK for packet 1, the sender exits fast recovery and continues in congestion avoidance with a *cwnd* of 4. The sender is unable to send data because nine packets (2-10) are still unacknowledged. During congestion avoidance, the sender receives two dup ACK corresponding to the receipt of packet 9-10. At this time, the sender is stalled and the "ACK clock" is lost, implying Reno is unable to employ fast retransmit and must await a retransmission timeout. The timeout for packet 2 expires, causing a retransmission and putting the sender into slow start. The ACK for packet 10 corresponds to the arrival of packet 2 at the receiver. The sender increases *cwnd* to 2 and continues in slow start.

A modified version of TCP, called TCP New-Reno [14], aims at avoiding timeouts when multiple packets are lost in the same window by changing the behavior of the TCP sender during fast retransmit as follows. The protocol defines a fast retransmit phase as the time between the receipt of 3 dup ACKs and the time when an ACK arrives for all the packets that were outstanding when the phase started. This is in contrast to the regular Reno implementation, where the fast retransmit mode lasts until the ACK for the retransmitted packet arrives. When multiple packets are lost in the same window, the retransmission of the first lost packet triggers a partial ACK, an ACK that acknowledges some but not all the packets what were outstanding at the start of fast retransmit. A partial ACK is treated as a signal that the packet whose sequence number is indicated has been lost and should be retransmitted. In this pattern, the TCP New-Reno recovers one lost packet during each RTT.

Consider the same example as in Figure 7, New Reno's operation is similar to Reno above until the receipt of the first ACK for packet 1. This ACK is a partial ACK and causes New-Reno to retransmit packet 2 immediately and not exit fast recovery. The dup ACK counter is reset to zero and later increased by the number of dup ACKs matching the partial ACK. The *cwnd* is not affected. With the arrival of two dup ACKs for packet 1, the sender cannot send new data since packets 2-10 are unacknowledged. The ACK for packet 10 causes the sender to exit fast recovery with a *cwnd* of 4 and continue in congestion avoidance.

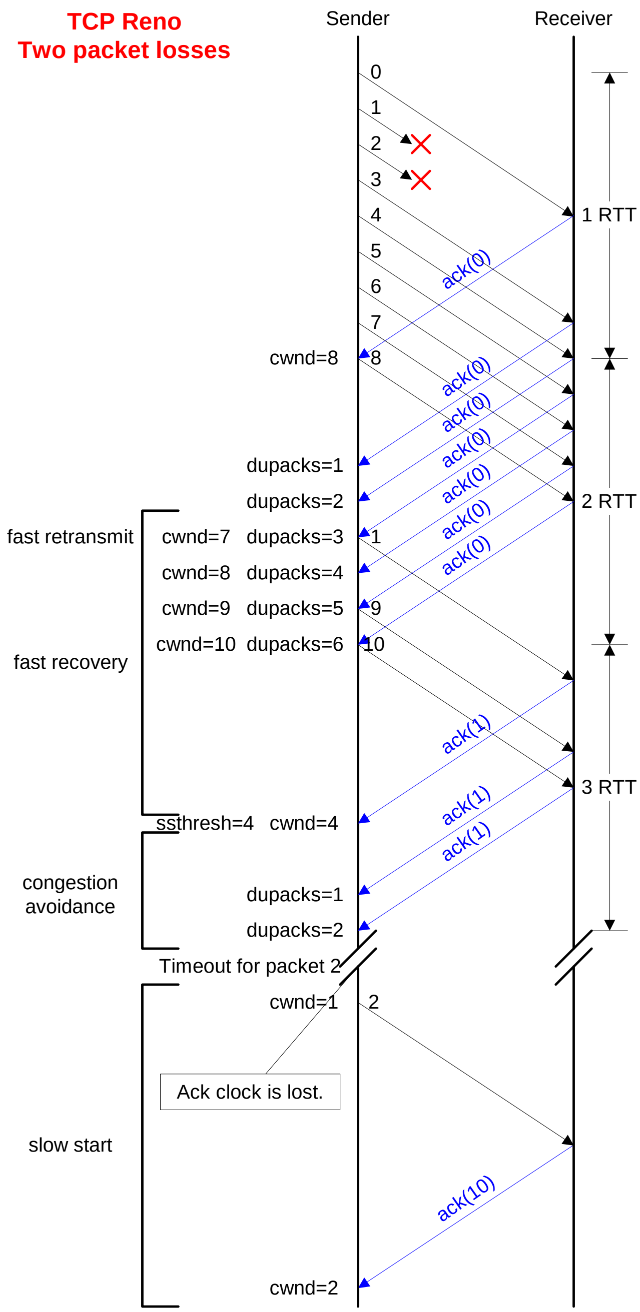


Figure 6 TCP Reno with two packet losses

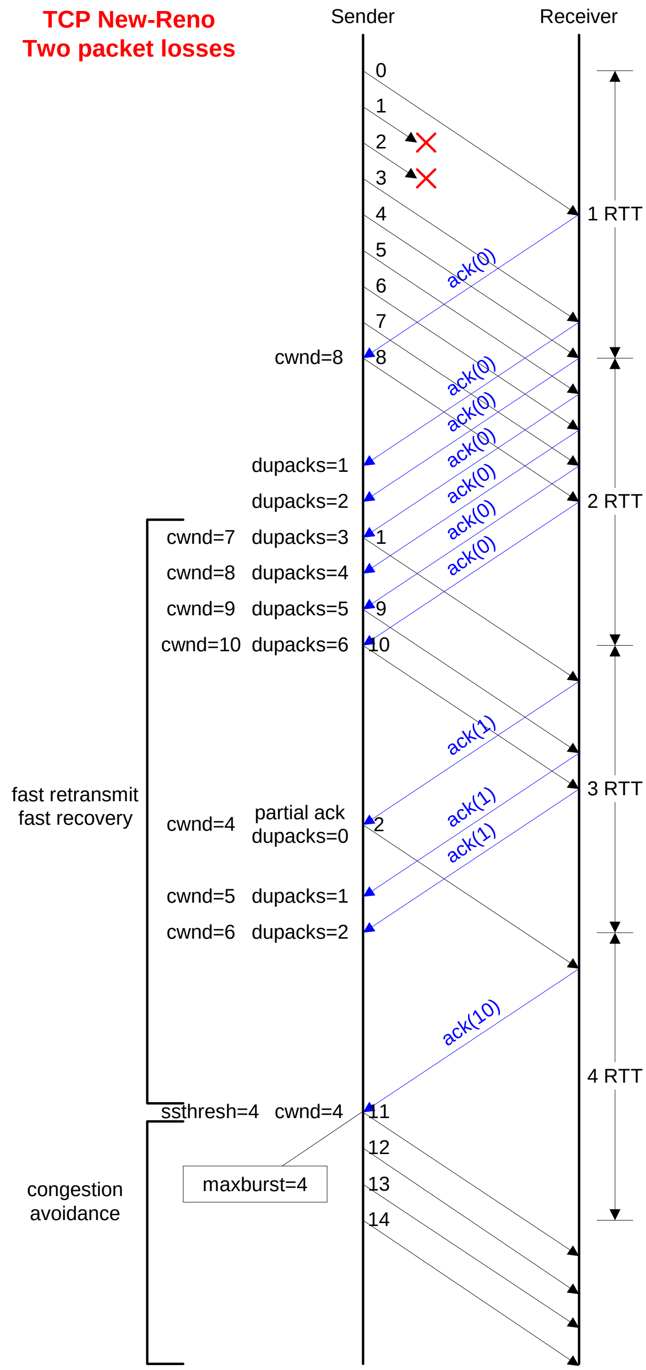


Figure 7: TCP New-Reno with two packet Losses

In AeroTCP, we explore the specific characteristics of our network. Firstly, the congestion is impossible for the satellite connections and any loss must be caused by the link layer corruption. The error recovery scheme can operate independently with the congestion control scheme. Secondly, the satellite link is a FIFO channel and out of order packet arrivals are impossible. We propose a new error control algorithm based on SACK TCP. The basic idea is to use one dup ACK for fast retransmit, use fixed window for satellite connection, and recover packet losses based on SACK information.

In SACK TCP [15], the SACK option field contains a number of SACK blocks, where each SACK block reports a non-contiguous set of data that has been received and queued. The first block in the SACK option is required to report the data receiver's most recently received segment, and the additional SACK blocks repeat the most recently reported SACK blocks. In our research each SACK option is assumed to have room for three SACK blocks. The congestion control algorithms implemented in the SACK TCP are a conservative extension of Reno's congestion control, in that they use the same algorithms for increasing and decreasing the congestion window. However, in AeroTCP, we change the congestion control algorithm as discussed in previous section. The AeroTCP implementation preserves the properties of TCP SACK of recovery multiple packet losses from one window of data and uses retransmit timeouts as the recovery method of last resort.

In AeroTCP, the sender enters super fast recovery when it receives one duplicate acknowledgement since out-of-order delivery in satellite channel is impossible. The sender retransmits a packet but will not cut the congestion window in half. During fast recovery, the sender maintains a variable called *pipe* that represents the estimated number of packets outstanding in the path. The sender only sends new packets or retransmits packets when the estimated number of packets in the path is less than the congestion window. The variable *pipe* is incremented by one when the sender either sends a new packet or retransmits an old packet. It is decremented by one when the sender receives a dup ACK packet with a SACK option reporting that new data has been received at the receiver.

Use of the *pipe* variable decouples the decision of when to send a packet from the decision of which packet to send. The sender maintains a data structure, the *scoreboard*, which remembers acknowledgements from previous SACK options. When the sender is allowed to send a packet, it retransmits the next packet from the list of packets inferred to be mission at the receiver. If there are no such packets and the receiver's advertised window is sufficiently large, the sender sends a new packet.

The sender exits super fast recovery when a recovery acknowledgement is received acknowledging all data that was outstanding when fast recovery was entered. When a retransmitted packet is itself dropped, the AeroTCP implementation detects the drop with a retransmit timeout. However, the timer has a finer granularity. After timeout, two copies of the dropped packet are sent to increase redundancy.

The AeroTCP sender has special handling for partial ACKs. For partial ACKs, the sender decrements *pipe* by two packets rather than one. When super fast recovery is initiated, *pipe* is effectively decremented by one for the packet what was assumed to have been dropped, and then incremented by one for the packet what was retransmitted. However, for partial ACKs, *pipe* was incremented when the retransmitted packet entered the pipe, but was never decremented for the packet assumed to have been dropped. Thus when a partial ACK arrives, it does in fact represent two packets that have left the pipe: the original packet (assumed to have been dropped), and the retransmitted packet.

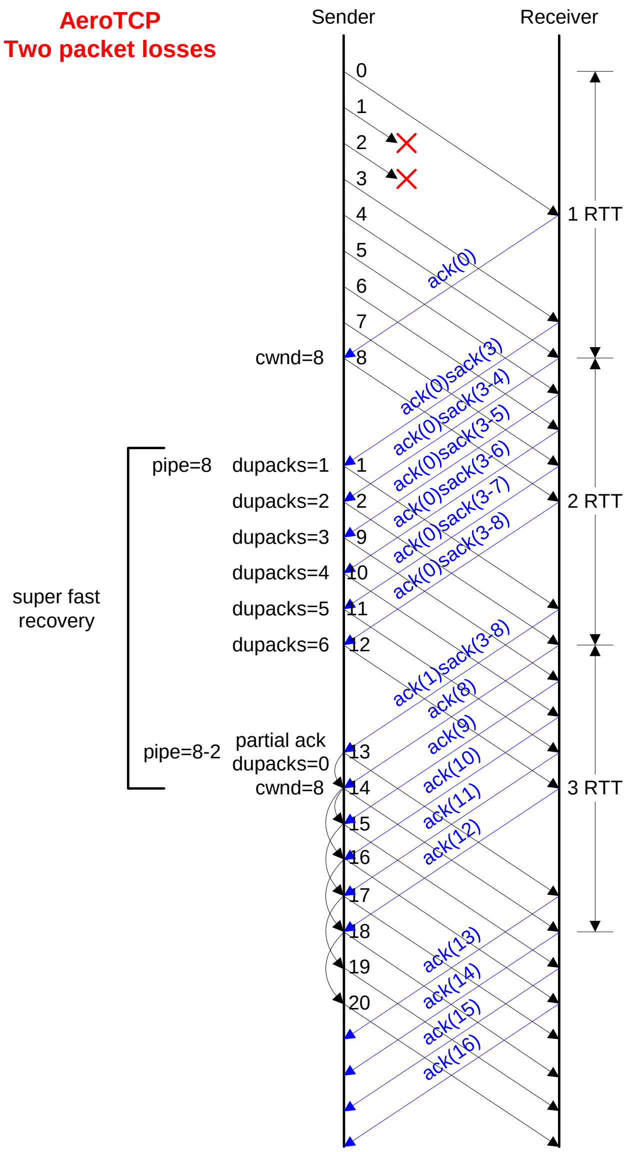


Figure 8: AeroTCP with two packet losses

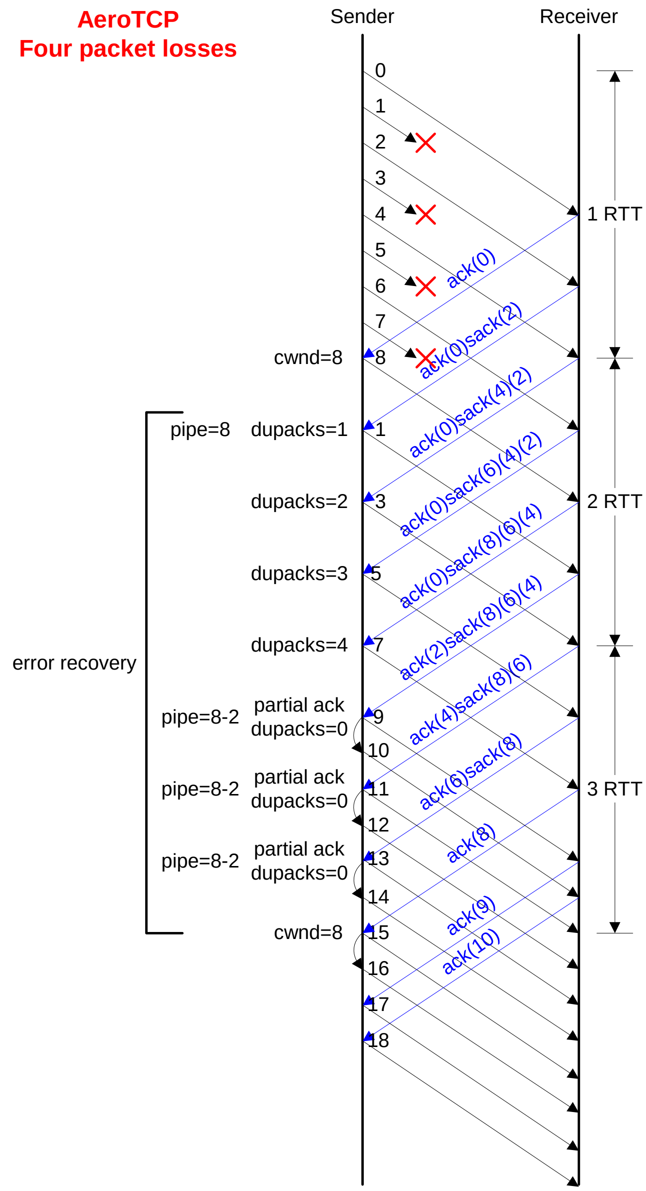


Figure 9: AeroTCP with 4 packet losses

Figure 8 shows the sequence of events for AeroTCP with two packet losses. At the beginning, packets 1-8 are sent while packets 1 and 2 are lost. After receiving the first duplicated ACK for packet 0, the sender goes into the super fast recovery, retransmits the lost packet 1, and initializes the *pipe* as *cwnd*. The second dup ACK causes the value of *pipe* to become 7 and contains SACK information, indicating a hole at packets 2. Packet

2 is retransmitted. Next 4 dup ACKs allow 4 new packets to be sent. From *scoreboard*, no holes remain to be filled and the sender may send new packets 9-12. The next ACK arrives corresponding to the receipt of retransmitted packet 1. It is a partial ACK, causing *pipe* to be decremented by two and allowing the sender to send packets 13-14 (packet 14 is scheduled to be sent in next time slot). The next ACK received corresponds to the receipt of retransmitted packet 2 and brings the sender out of super fast recovery with a congestion window of 8. After packet 20, the TCP returns back to its normal sliding window transmission pattern.

Figure 9 is another example of AeroTCP with 4 packet losses. Here packets 1, 3, 5, and 7 are lost. After receiving the first duplicated ACK for packet 0, the sender goes into super fast recovery with *pipe* initialized to 8. Packet 1 is retransmitted. Next three dup ACKs contains SACK information indicating a hole at packets 3, 5, 7 and those packets are retransmitted without delay. The next three ACKs correspond to the receipt of retransmitted packets 1, 3, and 5. They are partial ACKs and causes *pipe* to be decremented by 2 three times. Since there is no hole to be filled in the *scoreboard*, packets 9-10, 11-12, and 13-14 are sent corresponding to these ACKs. The ACK for packet 8 brings the sender out of super fast delivery with *cwnd* of 8.

Chapter 4: Simulation Results

4.1 The Simulation Scenario

We evaluate the performance of our protocol with OPNET. The metrics we are interested in are end-to-end throughput, satellite link utilization, fairness, and application response time. The simulation scenario is shown in Figure 10. A K/Ka-band (20/30GHz) GEO satellite operates as a bent-pipe transponder. There are two sets of transceivers on the satellite to provide connection between ground gateway and the aircraft in both directions. One is fixed and used to establish the communication link between the satellite and the fixed ground gateway, which provides the interconnection to the Internet backbone. The other is used to communicate with the aircraft, which includes on-board network. In our scenario, both the aircraft and the satellite need to track with each other. We assume that the tracking information is available through additional control channel. Future aeronautical satellite networks will have multiple satellites and multiple spot beams for each satellite to cover the whole world and more aircraft can be supported. The special handover procedure between spot beams and multi-access control between aircraft need to be implemented, while the aircraft still need to track the satellite during flight.

The link delay between the ground gateway and satellite gateway is about 240ms, While the link delay between the ground gateway and the Internet server and between the aircraft gateway and the client on aircraft are 40ms and 10^{-4}ms, respectively. Therefore the RTT between the server and the client is about 560ms. For satellite channel, the forward link (from ground to aircraft) and the return link (from aircraft to ground) have bandwidth of 5Mbps and 1Mbps, respectively. The bandwidth for ground link and aircraft link is 100Mbps. The bit error rate for satellite channel is determined by link budget and pipeline stages for radio link. It is uniformly distributed with a range from 10^{-9} to 10^{-4} in our simulation.

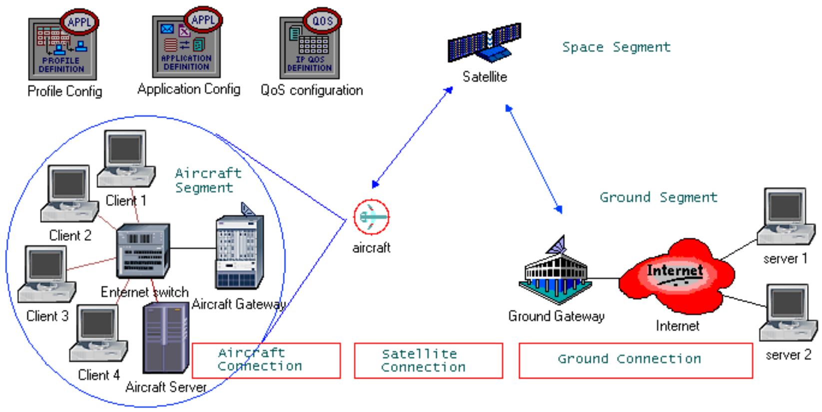


Figure 10: Simulation Topology

4.2 Web User Behavior Model

Web traffic continues to increase and is now estimated to be more than 70 percent of the total traffic on the Internet [35]. A good web traffic model is essential for simulations and experiments to investigate the performance of the network protocols and algorithms. Measurements of real traffic indicate that web traffic shows self-similarity, which means that significant traffic variance (burstiness) is present on a wide range of time scales and it can exhibit long-range dependence.

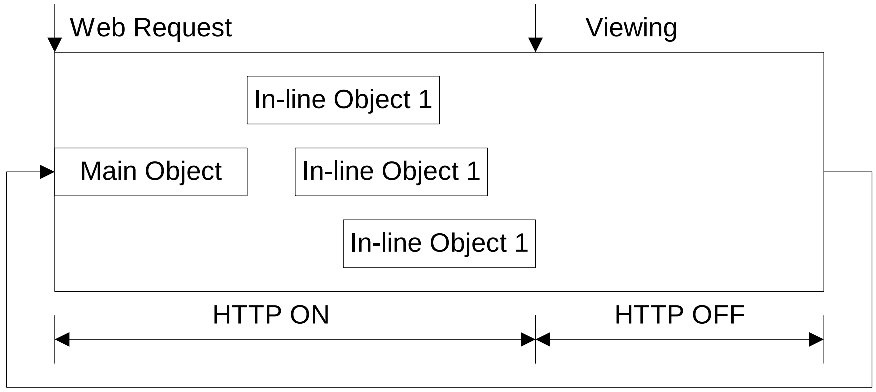


Figure 11: ON/OFF source model for web traffic

A process X is distributional self-similar if the distribution of the aggregated process of X is the same as that of X. A self-similar process has an autocorrelation function $r(k) \sim k^{-\beta}$ as $k \to \infty$ and $0 < \beta < 1$. This autocorrelation function follows power law decay, which is slower than exponential decay exhibited by traditional traffic models. The power spectrum of such a process is hyperbolic, rising to infinity at frequency zero, reflecting the infinite influence of long-range dependence in the data. Self-similar traffic can be constructed by multiplexing a large number of ON/OFF sources, where the ON and OFF period lengths have a heavy tailed distributions. The basic ON/OFF source model for web traffic is shown in Figure 11. A typical web page consists of a hypertext document with links to other objects that make up the whole page. An object is an entity stored on a server as a file. This model simulates an ON/OFF source where the ON state represents the activity of a web-request and the OFF state represents a silent period after all objects in a web-requests are retrieved. A new web-request is immediately generated after expiration of the viewing period.

Crovella [32] shows evidence that a number of file distributions on the web exhibit heavy tail distributions, including files requested by users, files transmitted through the network, transmission times of files, and files stored on the servers. A random variable X follows a heavy tailed distribution if

$$P[X > x] \sim x^{-\alpha}, as\, x \to \infty, \quad 0 < \alpha < 2$$

That is, regardless of the behavior of the distribution for small values of the random variable, if the asymptotic shape of the distribution is hyperbolic, it is heavy tailed. The simplest heavy tailed distribution is the Pareto distribution, with probability density function

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1}, where\, \alpha, k > 0, \quad x \geq k$$

and cumulative distribution function

$$F(x) = P[X \leq x] = 1 - (k / x)^{\alpha}$$

The parameter k is the location parameter and it represents the possible smallest value of random variable X. For Pareto distribution, when $\alpha \leq 2$, it has infinite variance; if $\alpha \leq 1$, it has infinite mean. For $1 < \alpha < 2$, the mean of Pareto distribution is $\alpha / (\alpha - 1) * k$.

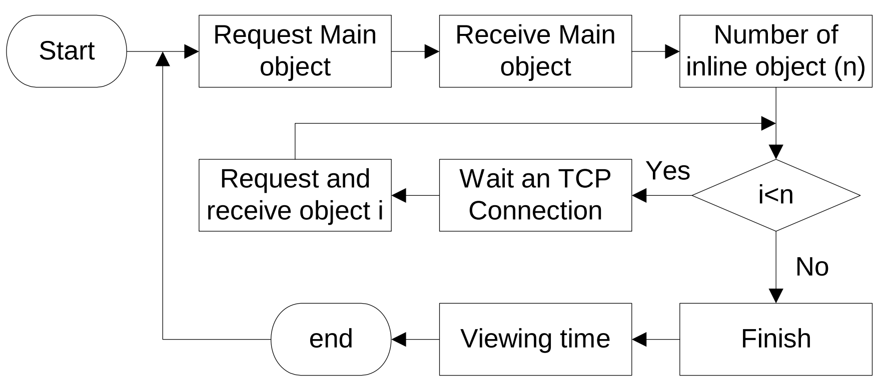


Figure 12: Web user behavior model

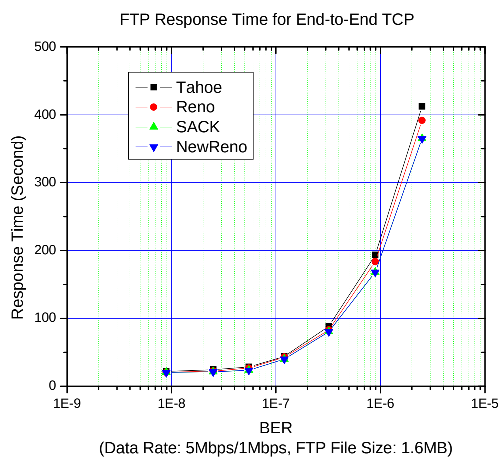
HTTP is a request-response based protocol. There are several empirical web traffic models in the literature[33][34][35]. The elements of a HTTP model are: 1) HTTP request length; 2) HTTP reply length; 3) number of inline objects per page; 4) user think time between retrieval of two successive pages. We will use the model shown in Figure 12 to generate the web traffic at the application layer for our experiments. First the web browser requests the HTML main object. Once the main object is received, the browser figures out how many inline objects are in the page and begins to request the inline objects. After all the inline objects are received, the user viewing the page for some time and starts to retrieve another web page. This model can model HTTP 1.0 with one or multiple TCP connections as well as HTTP 1.1 with a persistent connection. If there are TCP connections available whether it is one of the parallel connections in HTTP 1.0 or the persistent connection in HTTP 1.1, the browser can send new request through that connection.

The HTTP request length [33] will be modeled by a bimodal distribution with one large peak occurring around 250 bytes and another, a smaller one around 1KB. Mah [33] argues that the short requests correspond to simple file retrieval; the long requests correspond to complex requests such as those generated HTML forms. The reply file sizes will be modeled by Pareto distribution with $\alpha = 1.04$ to $\alpha = 1.14$ and $k = 1KB$. The number of inline objects per page will be modeled by a Gamma distribution [35] with mean of 5.55 and standard deviation of 11.4. The user think time will be modeled also by Pareto distribution [34] with $k = 1sec$ and $\alpha = 1.5$.

<u>4.3 End-to-end TCP performance</u>

First we investigate the End-to-End TCP performance. The client on the aircraft will download files from the ground server by using FTP during the flight. The forward channel and the return channel of satellite link have a bandwidth of 5Mbps and 1Mbps, respectively. Both the client and the server have TCP buffer size of 65536 bytes. In this simulation, we examine the performance of four variants of TCP loss recovery and congestion control: Tahoe (Fast Retransmission), Reno (Fast Retransmission and Fast Recovery), SACK (Reno + Selective ACK), and New Reno.

Figure 13 shows the TCP performance for the satellite link with FTP file size of 1.6MB. We can see that the response time to download a file increases exponentially with the BER. That's because the TCP congestion window cannot recover quickly when there is lots of packet losses (high BER). For same BER, the TCP New Reno and SACK have better performance than Reno and Tahoe. The differences of response time are more obvious when the BER becomes large. Figure 14 shows the satellite channel throughput, which is the file size divided by the total transfer time. Comparing to the satellite bandwidth of 5Mbps, this TCP connection can't fully utilize the satellite channel, especially for high BER.



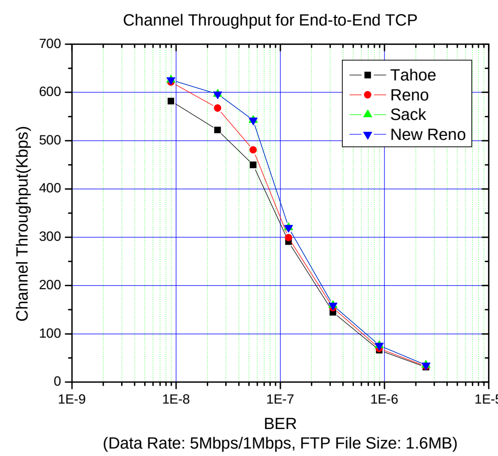
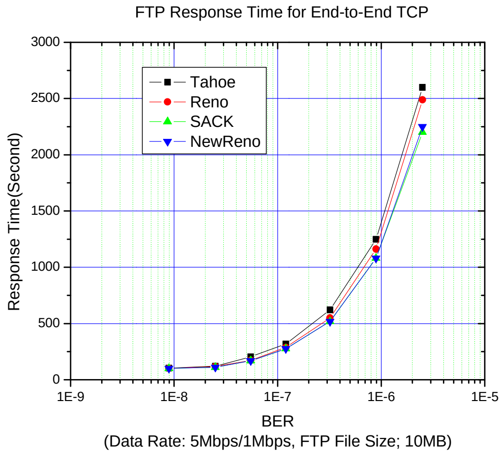


Figure 13: Response Time for 1.6MB file   Figure 14: Throughput for 1.6MB file



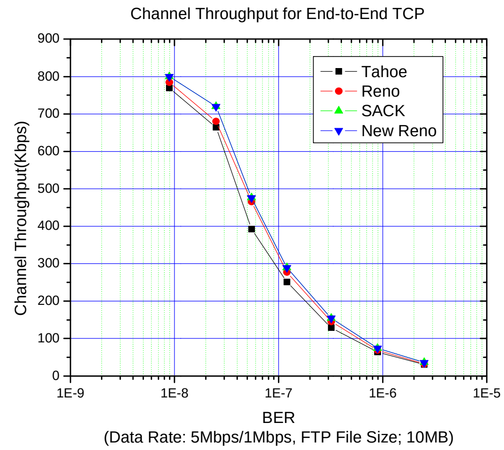
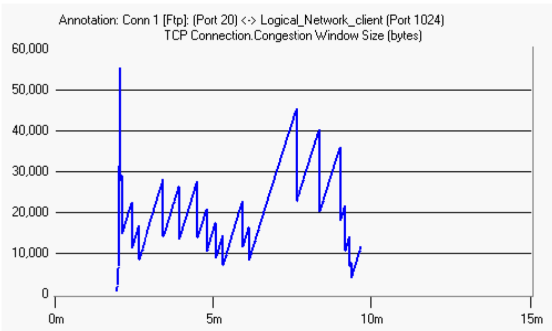


Figure 15: Response Time for 10MB file Figure 16: Throughput for 10MB file

Figure 15 and Figure 16 shows the response time and channel throughput for the file size of 10MB. The differences of the response time for those TCP flavors are more obvious especially in high BER.

To maintain high throughput for large file transfers, the TCP congestion window must be large. However, for end-to-end TCP implementations, the congestion window is shriek whenever there is an error (either link layer corruption or congestion). This leads to a small operational congestion window and low channel utilization. Using Window Scaling doesn’t help because the congestion can’t operate fully. Figure 17 shows this effect from our simulation with BER=10^{-6}.



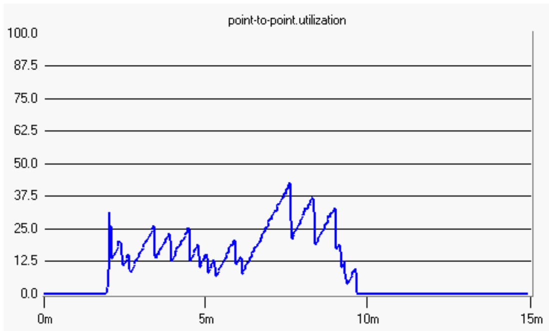


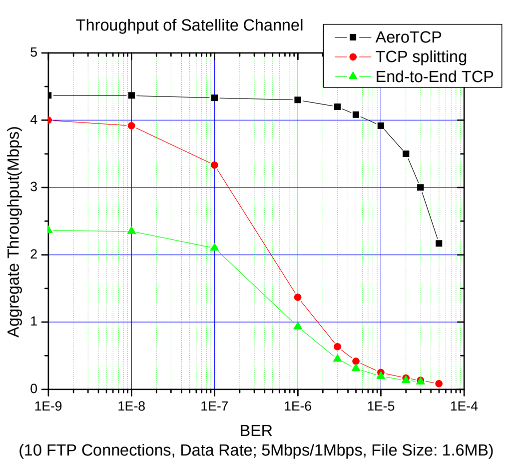
Figure 17: Congestion Window Size and Channel Utilization for BER=10^{-6}

From previous results, we can conclude that by using standard TCP with some enhancements for the satellite communications system with our configuration and system architecture, basic communication requirements can be meet. If TCP/IP protocols are going to be adopted in the future satellite system, some modifications of the protocol stacks will be necessary to achieve better performance. In particular, TCP SACK has better performance than other TCP flavors in our scenario. It also achieves high link utilization when the link BER is relative low. However, when the link BER becomes high, the end-to-end solutions cannot solve those problems effectively.

4.4 AeroTCP performance

To test our splitting scheme, ten clients download a file of 1.6MB using FTP from ten different servers. The round trip delay between the ground gateway and aircraft gateway is about 480ms while the round trip delay for the ground connection is 40+10*i, for connection i (i=1-10). Therefore the end-to-end RTT for connection i is about 520+10*i ms, i.e. in the range from 530ms to 640ms. The satellite link bandwidth is about 5Mbps and the peak rate for one connection is 1Mbps. The flow control and the receiver buffer size are set to 2-4 times peak rate delay product as in Table 4. The congestion window for each connection is set to bandwidth delay product corresponding to 500Kbps (static allocation). The transport protocol for the ground connection and aircraft connection are normal TCP SACK, while we use AeroTCP for the satellite connection. We compare three scenarios in our simulation: our AeroTCP, TCP splitting (with standard TCP SACK for satellite link), and End-to-End TCP SACK.

Figure 18 shows the aggregate throughput for all connections. When the bit error rate is very low, both splitting schemes can achieve very high throughput since the TCP can actually operate with large window, while end-to-end TCP has little difficult to increase its window fast enough to fully utilize the satellite bandwidth. For TCP connection splitting scheme, when the bit error rate increases up to 10^{-6}, the link layer corruption causes the satellite TCP to drop its congestion window, which leads to degraded performance. When the BER increases to 10^{-5}, the retransmitted packets can get lost again and TCP may have to wait for the timeout to recover the error. After timeout, the congestion window is set to one and TCP enters slow start and the channel throughput is very low. For End-to-End TCP solution, the performance gets even worse when the bit error rate increases. While for our scheme, the TCP can send packet at the rate of bandwidth as long as there are packets in the buffer and the receiver has enough buffer. The throughput is drop when the BER increases to 10^{-5}, which is because lots of packets are lost due to layer error corruptions.



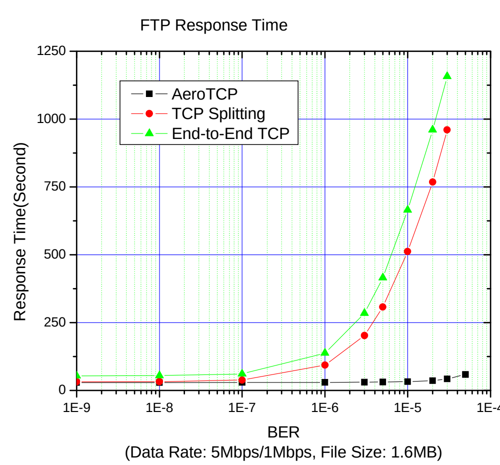


Figure 18: Aggregate Throughput for AeroTCP   Figure 19: Response time for AeroTCP

Figure 19 shows the FTP response time for end-to-end TCP, TCP splitting, and our scheme to download a file of 1.6M bytes. We can see that the TCP splitting has little better performance than end-to-end TCP because in TCP splitting, the terrestrial connection can operate with large window and send packets to gateway faster due to no error. It is interesting that performance is improved if we just use standard splitting protocol, although not noticeable when the BER is high. In the other hand, our scheme has the best performance than both end-to-end TCP and TCP splitting. This is because both the satellite connection and terrestrial connection of our scheme can operate with large window. The response time is more obvious when the BER increase to 10^{-5}. We get similar results when we use different FTP file size. It also shows the improvement for other applications like HTTP, Email, video and audio data.

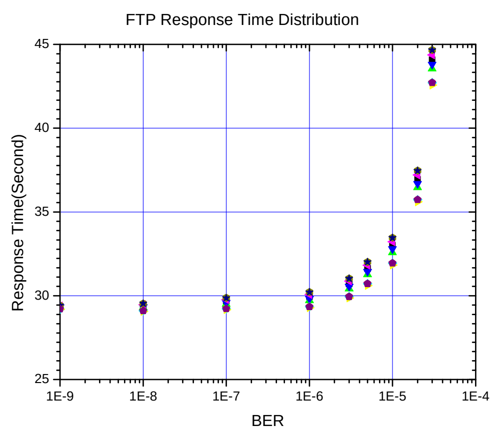


Figure 20: FTP response time for 10 connecitons

Figure 20 shows the FTP response time for those 10 FTP connections with different RTT. The variation of response time is relative small comparing to mean value of response time. That means the 10 connections need almost same time to download the file and get a fair share of the satellite link bandwidth. The variation of response time is mainly due to the RTT difference for terrestrial connections.

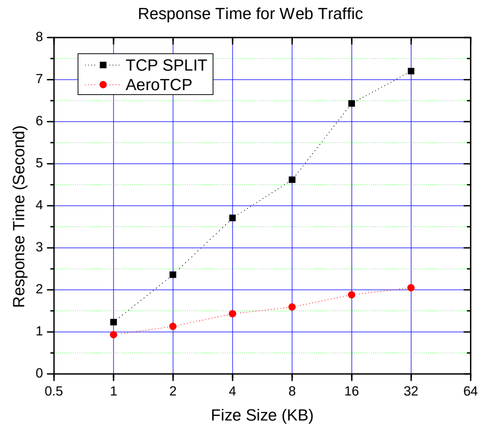


Figure 21: Response Time for Web Traffic

Figure 21 shows the response time for web traffic. The HTTP connection downloads only 1 object with file size of 1-32KB. The BER is 1E-7. We can see that the response time of TCP split is linear proportional to the file size and RTT, while AeroTCP need much less time to download the same file. This is because TCP split use slow start and multiple RTT to download big files. Please note that the mean for web file is 8-10KB, while median 1.5-2KB.

From above results, we can conclude that our AeroTCP protocol has better channel utilization and shorter response time for FTP application than end-to-end TCP and TCP splitting protocol. It also provides fairness among users with different round trip times.

Chapter 5: Conclusions and Future Work

The IP-based broadband aeronautical satellite network will provide numerous new applications and services for both airspace system operations and passenger communications. However, the interoperation between a satellite system and the exiting terrestrial Internet infrastructure introduces new challenges. In this thesis, we have investigated the performance of transport protocols over satellite links from several perspectives.

- ✓ We described the applications and services for aeronautical communications. To support projected air traffic growth, the modernization of current NAS system is required. A Satellite communication system, distinguished by its global coverage, inherent broadcast capability, bandwidth-on-demand flexibility, suitability to free flight concepts, and the ability to support mobility, is an excellent candidate to provide broadband integrated services for aeronautical communications.
- ✓ We defined basic operational scenarios and simulation platforms for aeronautical satellite communications networks. We will focus on the performance of data communications protocols and applications over aeronautical satellite systems, specially the Internet TCP/IP protocol suite.
- ✓ We describe the TCP problems in satellite networks and its possible solutions. We observed degradation in TCP performance for large bandwidth-delay product networks such as aeronautical satellite systems. We studied the performance of standard TCP protocols for aeronautical communications and concluded that it is difficult for end-to-end TCP solutions to solve the problems effectively.

- ✓ We proposed a TCP splitting protocol, AeroTCP, for aeronautical communications, which is designed for the satellite connections by taking advantage of the specific characteristics of the satellite channel. The basic schemes of this protocol are large window flow control, adaptive congestion control, and super fast error control.
- ✓ Simulation results show that AeroTCP can maintain high utilization of the satellite channel and has better performance than TCP split protocol and end-to-end TCP solutions.

Future work remains in the following areas:

- ➢ Evaluate AeroTCP for more complicated traffic and network configurations such as forward and return congestion, large number of users, large satellite bandwidth. Study the scalability problem in term of computation overhead and deployment considerations.
- ➢ The bit error rate for satellite channel is determined by link budget and pipeline stages for radio link. It is uniformly distributed with a range from 10^{-9} to 10^{-4} in our simulation. However, the radio channel in K/Ka band is an error prone channel with non-stationary error characteristics. Bit error rates as bad as $10^{-2\sim-3}$ are reported. To model the error characteristics of such channel, we need to add some statistic error model to the pipeline stages of radio link. The Gilbert Elliot model [37][38] would be a good start point. Those parameters can be chosen based on the results from the satellite propagation experiments.

- The AeroTCP protocol can maintain high utilization of satellite channel and fairness for large file transfer. However, it still has some problems to support other applications with bursty traffic, especially for integrated services with different QoS requirements [36]. In [39], we proposed a new traffic management scheme to provide congestion control in all time scale for integrated services. For web services with bursty traffic, a new random early detection flow control (REDFL) algorithm is proposed. This scheme will try to maintain the average queue size, minimize the packet drop rate, and achieve fairness for the gateways. It also avoids the bias against bursty traffic and global synchronization. Further evaluation and investigation is needed.

Bibliography

[1] Blueprint for NAS Modernization 2002 Update, http://www.faa.gov/nasarchitecture/Blueprint2002.htm

[2] Yurong Hu, Victor O.K. Li, "Satellite-Based Internet: A Tutorial", IEEE Communication Magazine, March 2001, pp 154-162

[3] Peter W. Lemme, Simon M. Glenister, Alan W. Miller, "Iridium Aeronautical Satellite Communications", IEEE AES System magazine, November 1999

[4] W. H. Jones, M. de La Chapelle, "Connexion by Boeing^{SM}-Broadband Satellite Communication System for Mobile Platforms", MILCOM 2001. IEEE Communications for Network-Centric Operations: Creating the Information Force, Vol 2, 2001, pp 755-758

[5] M. Werner, M. Holzbock, "System Design for Aeronautical Broadband Satellite Communications", In Proceedings of Int. Conference on Communications (ICC'02), Paper # J03-3, 2002

[6] W. Stevens, "TCP/IP Illustrated", Volume 3, Addison Wesley, 1996

[7] Ozgur Ercetin, Michael O. Ball, Leandros Tassiulas, "Next Generation Satellite systems for Aeronautical Communications", Technical Research Report of National Center of Excellence in Aviation Operations Research, NEXTOR T.R. 2000-1, ISR T.R. 2000-20

[8] A. Jahn, M. Holzbock, Markus Werner, "Dimensioning of Aeronautical Satellite Services", American Institute of Aeronautics and Astronautics, Inc., ICA-02-M.3.07

[9] J. Postel, "Transmission Control Protocol", Internet RFC 793, 1981

[10] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control”, RFC 2581, 1999

[11] C. Partridge and T. Shepard, “TCP performance over satellite links”, IEEE Network, 11:44-49, September 1997

[12] M. Allman, D. Glover, and l. Sanchez, “Enhancing TCP Over Satellite Channels using Standard Mechanisms”, RFC 2488, 1999

[13] I. Minei and R. Cohen, “High-speed Internet access through unidirectional geostationary satellite channels”, IEEE J. Select. Areas Comm., 17, February 1999

[14] S. Floyd and T. Henderson, and A. Gurtov, “The New Reno Modification to TCP’s Fast Recovery Algorithm”, Internet RFC 3782, April 2004

[15] J. Mahdavi, S. Floyd, M. Mathis, and A. Romanow, “TCP Selective Acknowledgements Options”, RFC 2018, April 1996

[16] M. Allman, S. Floyd, C. Partridge, “Increasing TCP’s Initial Window”, Internet RFC 2414, September 1998

[17] M. Allman(ed), S. Dawkins, D. Glover, J. Griner, D. Tran, T. Henderson, J. Heidemann, J. Touch, H. Kruse, S. Ostermann, K. Scott, and J. Semke, “Ongoing TCP research Related to Satellites”, RFC 2760, 2000

[18] R. Braden, “T/TCP-TCP extensions for transactions, functional specification”, RFC 1644, 1994

[19] M. Mathis and J. Mahdavi, “Forward Acknowledgment: Refining TCP Congestion Control”, in Proc. ACMSIGCOMM, pp. 281-291, Aug. 1996.

[20] Yadong Shang, Michael, Hadjitheodosiou, and John Baras, “Using Broadband Satellite Systems to Support Aeronautical Communications”, Proc. 22nd

International Communications Satellite Systems Conference and Exhibit, Monterey, California, 9-12 May 2004

[21] V.G. Bharadwaj, J.S. Baras and N.P. Butts, "An Architecture for Internet Service via Broadband Satellite Networks", International Journal of Satellite Communications, Vol. 19, Issue 1, pp. 29-50, January/February 2001

[22] A. Falk, N. Suphasindhu, D. Dillon and J.S. Baras, "A System Design for a Hybrid Network Terminal Using Asymmetric TCP/IP to Support Internet Applications", Proceedings of the Technology 2004 Conference, Washington, D.C., November 8-10, 1994

[23] V. Arora, N. Suphasindhu, J.S. Baras, and D. Dillon, "Effective Extensions of Internet in Hybrid Satellite-Terrestrial Networks", Proceedings of the 1st Conference of Commercial Development of Space, Part One, pp. 339-344, Albuquerque, New Mexico, January 7-11, 1996

[24] V. Arara, N. Suphasindhu, J. S. Baras, D. Dillon, "Asymmetric Internet Access over Satellite-Terrestrial Networks", Proceedings of the AIAA: 16th International Communications Satellite Systems Conference and Exhibit, Part1, pp. 476-482, Washington, D.C, February 25-29, 1996

[25] V. Arora, J.S. Baras, A. Falk, D. Dillon and N. Suphasindhu, "Hybrid Internet Access", Proceedings of the 12th Symposium on Space Nuclear Power and Propulsion/Commercialization, pp. 69-74, Albuquerque, NM, January 8-12, 1995

[26] J.S. Baras, M.S. Corson, S. Papademetriou, I. Secka, and N. Suphasindhu, "Fast Asymmetric Internet Over Wireless Satellite-Terrestrial Networks", Proceedings of

the 1997 IEEE Military Communications Conference (MILCOM), Vol. 1, pp. 372-377, Monterey, California, November 2-5, 1997

[27] A. Bakre and B. R. Badrinath, "Implementation and performance evaluation of indirect TCP", IEEE Transactions on Computers, Vol. 46 No. 3, March 1997

[28] R. C. Drust, G. Miller, and E. J. Travis, "TCP extensions for space communications", Proc. ACM Mobicom, '96, Nov. 1996

[29] Xiaoming Zhou and John S. Baras, "TCP over GEO satellite hybrid networks", in Proc. IEEE MilCom Conference, 2002

[30] R. Vaidyanathan, Issues in Design and Resource Allocation of Satellite Gateways,, M.S. Thesis, Electrical and Computer Engineering, University of Maryland, Fall 1999

[31] T. Ott, J. Kemperman, and M. Mathis, "The stationary behavior of ideal TCP congestion avoidance", Tech. rep., Bellcore 1996, ftp://ftp.bellcore.com

[32] M. E. Crovella, M. S. Taqqu, and A. Bestavros. "Heavy-Tailed Probability Distributions in the World Wide Web", Chapter 1 of A Practical Guide to Heavy Tails. Chapman & Hall, 1998

[33] Bruce A. Mah, "A Empirical Model of HTTP Network Traffic", In IEEE INFOCOM, 1997

[34] P. Badford and M. Crovella, "Generating representative Web workloads for network and server performance evaluation", In ACM SIGMETRICS, 1998

[35] Hyoung-Kee Choi and John O. Limb, "A Behavior Model of a Web Traffic", In International Conference of networking Protocol'99, September 1999

[36] J-S. Jou, X. Tan, J. S. Baras, “A Parallel Queue Structure for Active Queue Management”, Proceedings of the 38th Annual Conference on Information Sciences and Systems CISS 2004, pp. 467-472, Princeton, NJ, March 17-19, 2004; also ISR Technical Report TR2003-9

[37] E. O. Elliott, “Estimates of Error Rates for Codes on Burst-Noise Channels”, Bell Syst. Tech. J., Vol. 42, pp. 1977-1997, Sept. 1963

[38] Jeffrey Slack, “Finite State Markov Models for Error Bursts on the Land Mobile Satellite Channel”, Master Thesis, Brigham Young University, August 1996

[39] Yadong Shang, Michael Hadjitheodosiou, John Baras, “Flow Control and Active Queue Management for Integrated Services in an Aeronautical Satellite Network”, Accepted by 24th AIAA Interantional Communications Satellite Systems Conference, 11-15 Jun 2006, San Diego, California