PROTOCOL SUPPORT FOR A NEW SATELLITE-BASED AIRSPACE COMMUNICATION NETWORK

Yadong Shang, Michael Hadjitheodosiou, John Baras
Center for Satellite & Hybrid Communication Networks
Institute for Systems Research, University of Maryland,
College Park, MD 20742, USA
shangyd@glue.umd.edu, michalis@isr.umd.edu, baras@isr.umd.edu

ABSTRACT

We recommend suitable transport protocols for an aeronautical network supporting Internet and data services via satellite. We study the characteristics of an aeronautical satellite hybrid network and focus on the problems that cause dramatically degraded performance of the Transport Protocol. We discuss various extensions to standard TCP that alleviate some of these performance problems. Through simulation, we identify those TCP implementations that can be expected to perform well. Based on the observation that it is difficult for an end-to-end solution to solve these problems effectively, we propose a new TCP-splitting protocol, termed Aeronautical Transport Control Protocol (AeroTCP). The main idea of this protocol is to use a fixed window for flow control and one duplicated acknowledgement (ACK) for fast recovery. Our simulation results show that AeroTCP can maintain higher utilization for the satellite link than end-to-end TCP, especially in high BER environment.

1 INTRODUCTION

World airline passenger traffic growth is currently approximately 6% per annum and this figure is likely to drop only slightly in the next fifteen years. The current National Airspace System (NAS) is quickly becoming overburdened by increases in air traffic because of the old technologies and legacy systems [1]. The current systems do not support improvements needed for increased capacity, safety and security, not to mention the demands of new data link applications.

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 [2]. 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 other economic benefits.

Several Companies (e.g., Inmarsat, Iridium, Boeing) have announced plans to use satellite technologies to provide commercial broadband data services for airline passengers [3, 4]. These systems are expected to offer Internet access and to support virtual private networks (VPN) for passengers in flight. However, the performance of data communications protocols and applications over such systems is the subject of heated debate in the research community, especially the transport protocol in the Internet TCP/IP protocol suite [5]. Some researchers insist that TCP will work suitably in a satellite environment, while others have suggested satellite specific protocol options for improved performance, and still others claim that TCP cannot work effectively over satellite channels.

In this paper, we will evaluate how well TCP performs in aeronautical satellite networks. The remainder of the paper is organized as follows: in section 2, we discuss the future aeronautical satellite systems that plan to provide Internet access for passengers, focus on the special characteristics of aeronautical satellite environment that impact transport layer protocol performance. In section 3, we describe various TCP flavors and TCP extension that alleviate some of TCP performance problems in satellite environment. Through analysis and simulation, we identify those TCP implementations that can be expected to perform reasonably well for Internet applications in satellite networks. Based on the observation that it is difficult for an end-to-end solution to solve the performance problems effectively in the satellite hybrid networks, we next investigate in section 4 the improvement by using TCP splitting protocol in satellite gateway. We proposed a new splitting based TCP protocol for satellite connection, which we called AeroTCP. This new protocol uses a fixed window for congestion control and flow control, use one duplicated ACK for fast recovery. Based on the simulation result, we conclude that our splitting protocol has significant improvement over end-to-end TCP solution in term of link utilization and response time. This AeroTCP is the suitable transport protocol for the future aeronautical satellite networks.

2 AERONAUTICAL SATELLITE NETWORK

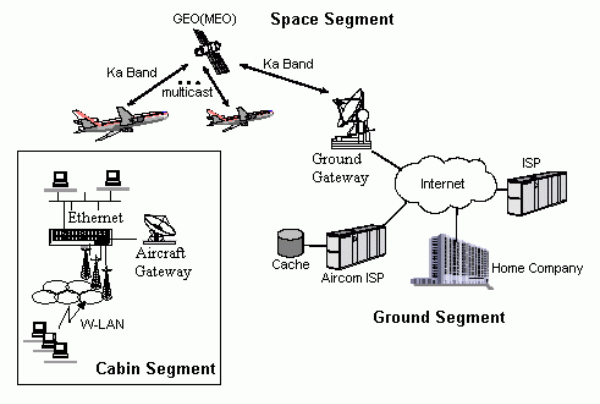


Figure 1 Aeronautical Satellite Network

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 [6]. Here, we consider architecture based on packet switching that is fully compatible with the TCP/IP protocol suite. Figure 1 illustrates the general topology, in which users on aircraft access the Internet via the satellite system.

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.

For the near-term future and any evolutionary approach towards aeronautical multimedia communications, a broadband network based on GEO satellites seems to be the first option. However, a GEO solution for the purpose of future broadband communications to aircraft in flight reveals several problems, such as the coverage problems at higher latitudes and the extreme antenna steering requirements at lowest elevation angles. With a LEO or MEO solution, in particular, potential system capacity limitations and latency for real-time communications could be reduced. On the other hand, besides system costs, especially networking complexity tend to increase while moving to lower orbits. Satellite handover will become a major issue, and inter-satellite links may be necessary at least for LEO constellations to provide connectivity over large ocean areas. Since we are focus on the transport protocol and for simplicity, we will use GEO bent-pipe satellite for our analysis and simulation.

The service scenario considers travelers in aircraft on the move. Since people are becoming more and more used to their own communications equipment, such as mobile phones and laptops with Internet connection, either through a wired or wireless network interface. Thus the cabin segment will consist of wireless access technologies (Such as Blue tooth, Wireless LAN) as well as conventional IP fixed wired networks.

The main characteristics of the end-to-end path that affect transport protocol performance are long propagation delay, large bandwidth delay product, high bit error rate, and bandwidth asymmetry. If part of the communication path includes a satellite channel, these parameters can vary substantially from those found on wired networks.

Long propagation delay: The three main components of delay are propagation delay, transmission delay, and queuing delay. In the broadband satellite case, the dominant portion is expected to be propagation delay. GEO satellite is about 36,000km above the earth. The propagation delay from the earth up to the satellite and from the satellite down to the earth is about 125ms. Therefore a typical round trip time (RTT) for two-way system is about 500ms plus the delay for terrestrial networks.

Large bandwidth-delay product: We assume here that the GEO satellite operates on K/Ka band to provide broadband connection for aircraft. The bandwidth delay product (e.g., 20Mbps*580ms RTT) in this system is very large. To fully utilize such channel, we need to put that much of data (equal to bandwidth-delay product) into the link. Since TCP will send new data until it receives the ACKs for old data, that means TCP window should be at least the bandwidth delay product.

High bit error rate: In satellite channel, bit error rate of the order of 10^{-6} are often observed. This is primarily because the existing systems with legacy equipment and many existing transponders were optimized for analog voice and video services. New modulation and coding techniques, along with higher-powered satellites, should help to make bit error rate very low for GEO system.

Bandwidth asymmetry: Satellite networks can be asymmetric in several ways. Some satellite networks are inherently bandwidth asymmetric, such as those based on a direct broadcast satellite (DBS) downlink and a return via a dial-up modem line. For purely GEO or LEO system, bandwidth asymmetries may exist for many users due to economic factors.

In addition, an aeronautical satellite network has some specific network characteristics.

Mobile Aircraft: In this network, passengers inside the aircraft connect to the aircraft gateway using their own equipments via wired or wireless connections. The aircraft gateway connects to the ground network via a bent-pipe satellite. The mobility of the aircraft will put addition track requirements for the antennas on the aircraft and the satellite.

En-route Low BER: When the aircraft is en-route, there are no obstacles (such as rain, cloud) between the aircraft and the satellite. It is possible that the satellite link maintains connectivity and achieves "fiber-like" quality (BER of 10^{-10}) most of the time.

FIFO Satellite Channel: The bent-pipe satellite link is a FIFO channel and there is no out-of-order delivery between satellite gateways. Congestion over the satellite link is impossible if the packets are sent at the rate of the satellite bandwidth.

Intermittent connectivity: Due to handover and relative geometric position changes of aircraft and satellites, intermittent connectivity are expected in aeronautical network with typical during of several seconds to several minutes.

Variable Round Trip Times: Because the aircraft fly around the world, the propagation delay to and from the satellite varies over time. The variance of the RTT can be as large as 80ms. The impact of the variation to TCP is an open problem in literature.

In summary, we assume the aeronautical satellite networks characterized by high BER, high bandwidth delay product, long delay. Although the K/Ka band satellite can provide higher bandwidth than Ku band, 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.

3 END-TO-END TCP PERFORMANCE

In this section, we describes basic TCP operation, identifies protocol options helpful to improve TCP performance in satellite environment. We also quantify how well different TCP implementations perform in a satellite environment for Internet services.

3.1 TCP OPERATION

TCP is a connection-oriented, end-to-end, process-to-process reliable transport protocol [7]. TCP source and destination port combined with IP source and destination addresses uniquely identity each TCP connection. There are several mechanisms in TCP to ensure those functions, such as flow control, congestion control and error control.

Flow control: TCP uses a sliding window to achieve flow control. The TCP receiver sets the receive window (RCVWND) field in the acknowledgement to its free buffer size so that the sender will never overflow the receiver's buffer.

Congestion control [8]: The TCP sender maintains a state variable CWND for congestion window size. While RCVWND is used to guard that the sender will not overload the receiver buffer, the CWND is used to guard that the sender will not overload the network. The TCP sender can send at most the minimum of RCVWND and CWND window worth packets without receiving any ACK. In the most popular TCP Reno, there are four algorithms used for congestion control, which are slow start, congestion avoidance, fast retransmit, and fast recovery. Slow start is used upon the start of a new connection to probe the network bandwidth, it increases the CWND by one Maximum Segment Size (MSS) when an ACK is received, which results in increasing congestion window exponentially. TCP stays in slow start until its CWND is greater than the slow start threshold. After that TCP gets into congestion avoidance, it increases CWND about one MSS per round trip time (RTT). Fast retransmit algorithm is triggered when a fixed number of duplicate acknowledgements (usually 3) are received. TCP retransmits the potential lost packet indicated by the acknowledgement and cuts its CWND to half. After that, it inflates its CWND by one MSS when a duplicate acknowledgement 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 duplicate acknowledgement is received if allowed by the RCVWND. Finally it will send half a window new packets when it receives the first non-duplicate acknowledgement. TCP Reno doesn't like TCP Tahoe, which does not have the fast recovery algorithm and sends half a window packets in burst after the loss has been recovered.

Error control: Error control is the main component of reliable protocols, which includes error detection and error recovery. TCP uses acknowledgement packet, timer and retransmission to achieve error control. TCP uses cumulative acknowledgement, which means when a packet gets lost, it prevents the acknowledgement from being advanced and the window cannot slide until the lost packet is recovered. The sliding window mechanism actually ties the flow control, congestion control and error control together and it becomes vulnerable when there are losses due to congestion loss and packet corruptions in the network.

3.2 TCP FLAVORS AND EXTENSION

As originally specified, TCP did not perform well over satellite networks (or high latency networks in general) for a number of reasons related to the protocol syntax and semantics. Over the past decade, a number of TCP flavors and extensions have been specified which improve upon the performance of the basic protocol in satellite environments [9, 10].

Large Initial Window [11]: For a connection with large RTT, TCP spends a long time in slow start before reaching the available bandwidth. The time taken by TCP slow start to reach the satellite bandwidth (SatBW) is about RTT * $\log_2$ (SatBW * RTT) when every TCP segment is acknowledged. For short transfers, they could be finished in slow start, which obviously does not use the bandwidth efficiently. Some researchers propose to use a large initial window up to 4380 bytes (or a maximum of 4 segments) rather than 1 segment. Thus Files less than 4K bytes (many web pages are less than this size) can finish their transfers in one RTT rather than 2, 3.

Window Scaling [12]: In TCP protocol syntax, the receiver advertised window in the TCP header cannot be more than 64K bytes, which limits the two-way throughput to roughly 1Mbps in GEO satellite networks. Window Scaling is proposed to solve this problem, which significantly increases the amount of data that can be outstanding on a connection by introducing a scaling factor to be applied to the window field. This is particularly important for the satellite links, which require large windows to fully utilize their high bit rate.

Selective Acknowledgements (SACK) [13]: Because TCP Reno (popularly used in many systems) treats all losses as congestion in the network. The link layer error can causes TCP to drop its window to a small size and leads to poor performance. TCP SACK can convey non-contiguous segments received by the receiver in the acknowledgements so that the sender can recover error much faster than TCP Reno, which well know can recover only on loss per RTT.

Path MTU discovery [14]: This option allows the TCP sender to probe the network for the largest allowable Message Transfer Unit (MTU). Using large MTUs is more efficient to reduce the overhead and helps the congestion window to open faster.

Forward Error Correction (FEC): FEC is usually used in satellite communication to reduce the bit error rate. However, FEC consumes some bandwidth by sending redundant information together with the data and transforms the original random error nature to one with burst error.

In this work, we are interested in quantifying the performance of TCP implementations with those standard enhancements. Note that even though some of these options have been specified for over several years, not all implementations use them today. It is important to emphasize that all of the above implementations would be regarded as conformant to the TCP standards; in practice, many more variants of TCP exist.

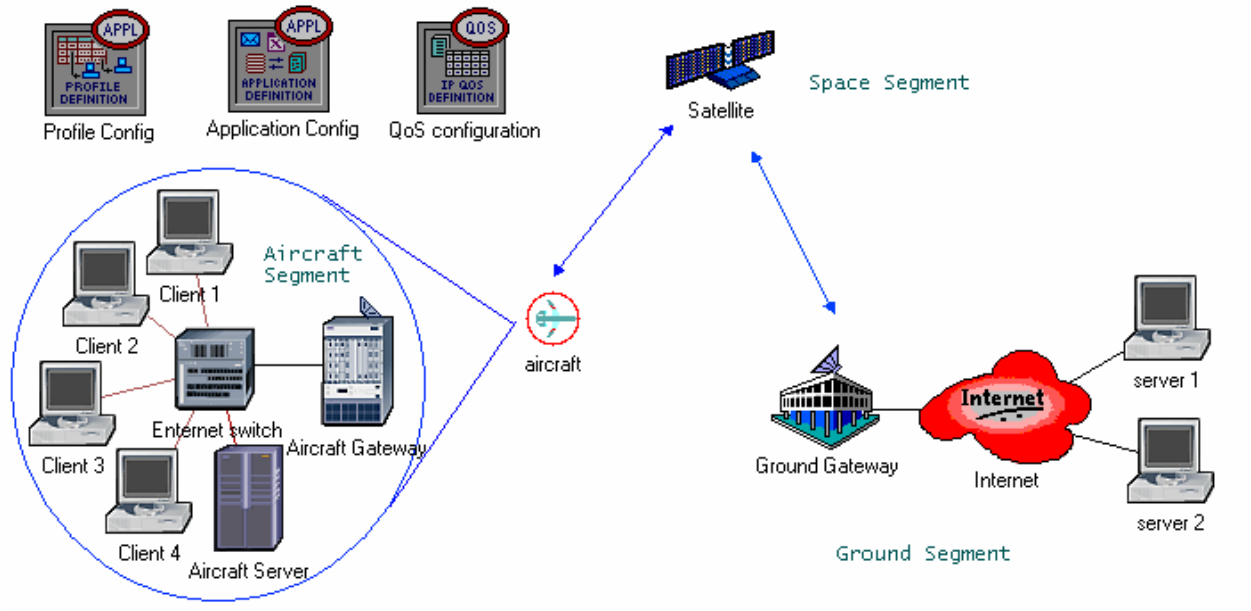


Figure 2 Experiment setup for End-to-end TCP solutions

3.3 END-TO-END TCP PERFORMANCE

In this section we study how well different TCP implementations perform in aeronautical satellite network for Internet services. The detail of the simulation setup can be found in [15].

The experiment setup of this simulation is shown in figure 2. A K/Ka-band GEO satellite operates in the microwave switch mode, in which it behaves as a bent-pipe transponder. Its spot beam is used to establish the communication link between the satellite and the fixed ground terminal, which provides the interconnection to Internet backbone. The aircraft, which includes on-board network, communicates with satellite by its tracking antenna. The client on the aircraft will download files from the ground server by using FTP during the flight. For each scenario, the forward and return links have data rate 5Mbps and 1Mbps, respectively. Both the client and the server have TCP buffer size of 65536 bytes.

To maintain high throughput for large file transfers, the TCP congestion window must be large. This implies that the congestion avoidance and loss recovery mechanisms are very important in determining performance. 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 Window Scaling (SACK + Window Scaling) [16].

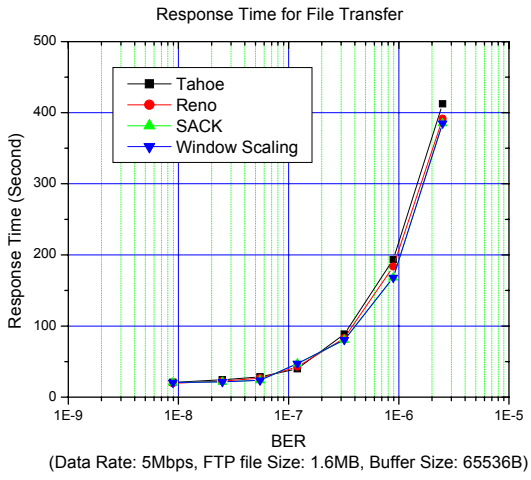


Figure 3. TCP performance for satellite link (1.6MB)

Figure 3 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 recovery quickly when there are lots of packet losses (high BER). For same BER, the Windows Scaling and SACK have better performance than Reno, Tahoe, where Tahoe need the largest response time to download the same file. The differences of response time are more obvious when the BER becomes large.

We see that Window Scaling and SACK have almost same performance. That is because in these scenarios, both the client and the server use TCP buffer size of 65536 bytes. The congestion window of TCP connection cannot be larger than the buffer size. Window scaling will have better performance than SACK when we have large buffer size. Figure 4 shows this effect. Here the client will download file of size 1.6MB. When the receiver's buffer size is less than 65536 bytes, the Window scaling and SACK have same performance. Actually window scaling is not used in this case. When the buffer size is larger than 65536 bytes, window scaling needs less time than SACK to download the same file. The difference in the response time will become more obvious in low BER and for big file.

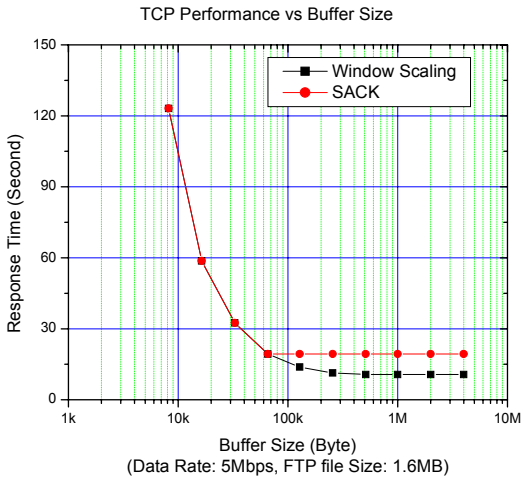


Figure 4. TCP performance with different buffer size

From previous results, we can conclude that by 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 with window scaling option 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. We will discuss this problem in next section.

4 TCP SPLITTING PROTOCOL

Although TCP can work well over GEO satellite links under certain conditions, there are cases for which even the best end-to-end modifications cannot ensure good performance. Furthermore, in an actual network with a heterogeneous user population, users and servers cannot all be expected to be running satellite-optimized versions of TCP. This has led to the practice of splitting transport connections. In this section we describe unsolved problems by use end-to-end TCP solutions, the design of our splitting protocol, and the performance comparison of the splitting protocol and end-to-end solutions.

4.1 Unsolved problems

Despite the progress on improving TCP, there remain some vexing attributes of the protocol that impair performance over satellite links. The end-to-end enhancements cannot solve these problems, or not very effectively.

Small operational window: 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 [17]. 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 even if the window scaling option is available, it is difficult for satellite TCP connections to actually operate with large windows.

Asymmetric link: With respect to transport protocols, the forward throughput achievable depends not only on the link characteristics and traffic levels in the forward path but also on those of the reverse path. The congestion in reverse path could lead to poor performance in the forward link because TCP uses ACKs to clock out data. To alleviate this problem, ACK filtering was proposed to drop the ACKs in the front of the IP queue by taking advantage of the cumulative acknowledgement strategy in TCP [18]. The situation is even worse for two-way transfers. When the users are sending data and browsing the web at the same time, a lot of data packets could be queued in front of ACKs in a FIFO queue, which increases the ACKs delay dramatically. In this case, a priority queue can be used to schedule the ACK to be sent first.

TCP fairness: For bulk transfer, TCP throughput is inverse proportional to RTT, so TCP connection with large RTT does not get its fair share of the bandwidth when it competes with the connections with shorter RTT [19]. It is difficult for end-to-end solutions to solve this fairness problem. Using the Constant-rate additive increase policy can correct this bias. However, it is difficult to implement in a heterogeneous network.

Because the feedback information of the satellite is either delayed too long or too noisy or both, end-to-end schemes cannot solve these problems very effectively. An alternative to end-to-end schemes is to keep the large window of packets in the network such as at the satellite gateway between the satellite and aircraft platform. Considering the interoperability issue, we propose a connection splitting based scheme to solve those problems.

4.2 Splitting protocol

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 5 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, another one is from the ground gateway to the aircraft gateway, and the last 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.

The goal of splitting connections is for end users to be unaware of the presence of an intermediate agent, other than improved performance. From the perspective of the hose in the wide-area Internet, it is communicating with a well-connected host with a much shorter latency. For the satellite link between the ground gateway and the aircraft gateway, a satellite optimized transport protocol can be used.

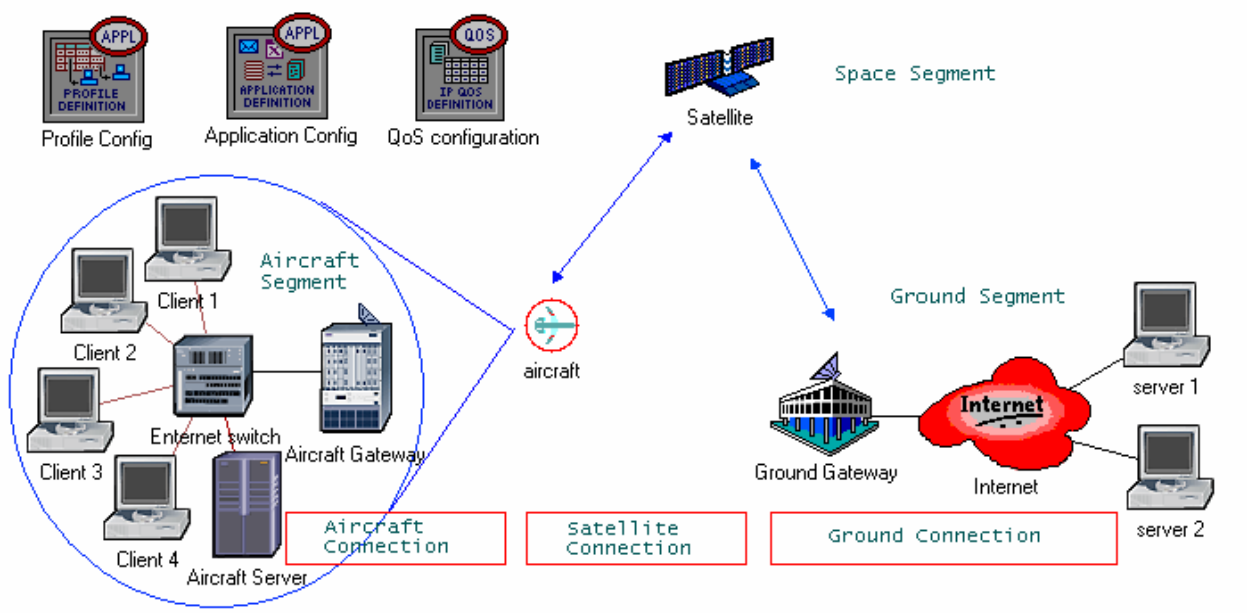


Figure 5 TCP Splitting protocol for Aeronautical Satellite Networks

Because GEO satellite channel is a FIFO channel, there is no out-of-order routing. And congestion over the satellite link is impossible if the packets are sent at the rate of the satellite bandwidth. The above observations motivate us to decouple the congestion control and error control in TCP first and then design more efficient and effective congestion and error schemes with our specific network characteristics in mind. We design a new TCP splitting protocol, which we called Aeronautical Transport Control Protocol (AeroTCP), for the satellite connection. The main idea is to use one duplicate ACK to trigger the fast retransmission at the satellite gateway and to use a fixed window size for the satellite TCP connection. This implementation of this idea will be discussed in detail in the following.

Flow Control: We still use a sliding window for flow control, however, here the window are fixed for each satellite connection. For a normal router, they only have the functions up to IP layer, while the satellite gateway is able to process TCP packets. All the TCP packets received from the servers are forwarded to the TCP received buffer of the ground gateway and they are moved from the received buffer to send buffer for transmission. The receive buffer and send buffer can be implemented by one physical buffer.

The buffer size assigned to each connection at the satellite gateway has a direct impact on the end-to-end TCP throughput. Consider the traffic from Internet server to the client on aircraft, assume there is only one connection in this system, the buffer size assigned to the TCP connection is Buff and the effective satellite bandwidth is SatBW. The date in the satellite pipe is SatWin and the advertised receiver window for the server is RecvWin. The round trip time for the satellite connection is SatRTT and for the ground connection is GndRTT. Then 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, i.e., RecvWin / GndRTT = SatWin / SatRTT. The throughput of the connection is min (SatBW, Buff / (SatRTT + GndRTT)) and the backlog packets are max (o, Buff – SatBW * (SatRTT + TerrRTT)) [20]. 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. However when the buffer size is greater than the bandwidth delay product, there are packets backlogged at the satellite gateway and these backlogged packets cannot contribute to the throughput and only increase the queuing delay.

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. For simplicity, we assign each connection a static peak rate, which is the maximum bandwidth it can achieve and is much smaller than the total satellite bandwidth, and the buffer size is set corresponding to that peak rate.

We assume large but not infinite buffer is available at the client and the TCP flow control is still enforced so that the gateway will not overflow the receiver's buffer.

Congestion Control: For the satellite connections, the satellite link bandwidth to be shared among them is fixed and known. Besides 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 all the congestion control algorithms in TCP. The gateway can send packets as long as there are packets in the buffer and the receiver's window allows sending. Also there is no need to exponentially back off the timer after timeout because congestion is impossible over the satellite link. Timer is used only for error recovery. As long as there are packets buffered at the satellite gateway, the satellite link can be fully utilized. When the traffic load increases, the buffers begin to be filled up and the congestion is back pressured to the sources through the advertised receiver windows. When the traffic load decreases, the buffers begin to be emptied and larger advertised receiver windows are sent to the source so the sources can speed up. This way satellite link efficiency is achieved.

Error Control: TCP depends on duplicate acknowledgements and timer for error control because out of order packet arrivals are possible in the wide area networks, the fast retransmit algorithm is triggered after three rather than one or two duplicate acknowledgements are received. The three duplicate acknowledgements requirement puts a high burden on the return channel bandwidth. The high bit error rate of the satellite link can cause multiple packet losses in one RTT and may lead to timeout. When the retransmitted packets are lost, timer could be the only means for error recovery. However, timer has to be conservative and is usually set much larger than the round trip delay to make sure the packet does leave the networks. These conservative loss detection and recovery schemes in TCP are not effective in satellite networks and should be enhanced.

In our scheme, we explore the specific characteristics of our network. Firstly, because congestion is impossible for the satellite connections and any loss must be caused by the link layer corruption. So 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 design a scheme for error control by using one duplicated ACK for fast recovery. We keep track of the packets in sequence space of all acknowledged packets. Whenever a duplicated acknowledgement is received, we just assume that packet is lost and retransmit it. During the recovery, we use the same idea as in TCP NewReno [21]. We use partial ACKs to calculate the burst loss gap and send all the potentially lost packets beginning from the partial acknowledgement number. Although it is possible that the sender could retransmit packets that have already been correctly received by the receiver, it is more effective in recovering burst errors (popular in satellite environment). Timer is still used as the last resort for loss recovery, however, after timer expires, two copies of the lost packet are sent to increase redundancy.

4.3 TCP Splitting Protocol Performance

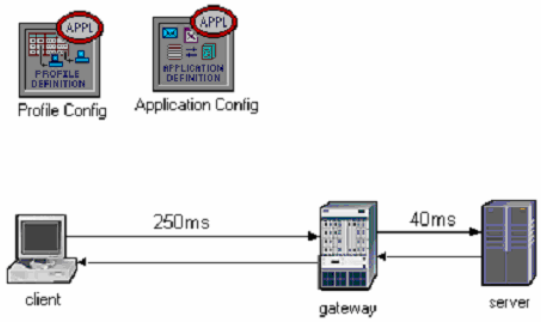


Figure 6 Experiment setup for TCP splitting Protocol

To test our splitting scheme, we setup a simulation model as shown in figure 6. A client downloads a file of 1.6M bytes from a server via a satellite link. The link delay between the hybrid gateway and the client is 250ms and the link delay between server and gateway is 40ms. Therefore the RTT between the server and the client is 580ms. The satellite link bandwidth is DS1 (1.544Mbps) and the ground link is 10Mbps. The transport protocol for the ground connection is normal TCP SACK, while we use our TCP splitting scheme for the satellite connection. We assume the satellite link between the gateway and the client is noise channel with various bit error rate, while the ground link has no error. The statistics we collect is link utilization and link throughput of the two downstream links.

Figure 7 shows the utilization for our scheme and for TCP connection splitting scheme. The TCP connection splitting scheme uses TCP SACK for both the satellite connections and terrestrial connections. When the bit error rate is very low, both schemes can achieve very high throughput since the TCP can actually operate with large window. 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 link utilization is very low. 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 utilization is drop when the BER increases to 10^{-5}, which is because lots of packets are lost due to layer error corruptions.

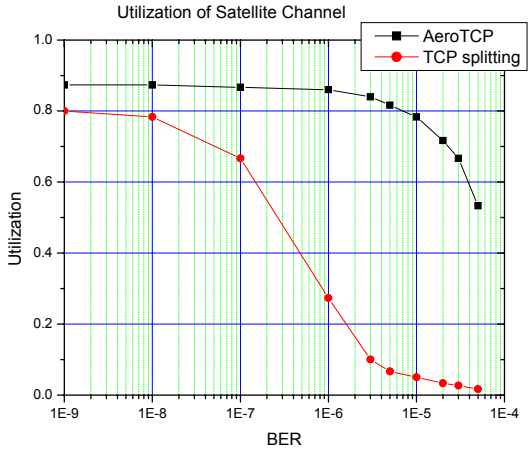


Figure 7 Utilization for different bit error rates

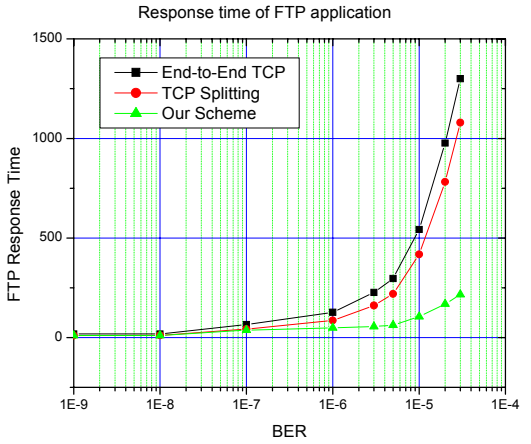


Figure 8 Response time for FTP application

It is important to compare our scheme with the end-to-end TCP solutions. Figure 8 shows the FTP response time for end-to-end TCP, TCP splitting, and our scheme to download a file of 1.6M bytes. The TCP splitting uses TCP SACK for both the satellite connections and terrestrial connections. We can see that the TCP splitting has 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 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}.

In summary, we have described the design and performance of a satellite-optimized transport protocol, AeroTCP. AeroTCP inherently incorporates many of the features that have been proposed or adopted as TCP options for improved satellite performance. AeroTCP allows for the use of rate-based congestion control and is well matched to satellite networks.

5 CONCLUSIONS

In this paper, we have investigated the performance of IP-Compatible transport protocols over satellite links from several perspectives. We observed degradation in TCP performance for large bandwidth-delay product networks such as aeronautical satellite systems. If the right TCP options are used and congestion is light, TCP can work well for large file transfers even over GEO links. Because it is difficult for an end-to-end TCP solution to solve the problems in the aeronautical satellite networks, we propose a connection splitting based solution, AeroTCP, which is designed for the satellite connections by taking advantage of the specific characteristics of the satellite networks. Our simulation results show that our scheme can maintain high utilization of the satellite link and has better performance than end-to-end solutions.

Acknowledgements: *This work is supported by the Center for Satellite and Hybrid Communication Networks, under NASA cooperative agreement NCC8-235 and NASA cooperative agreement NAG3-2844.*

REFERENCES

1. Oagur 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
2. M. Werner, M. Holzbock, "System Design for Aeronautical Broadband Satellite Communications", In Proceedings Int. Conference on Communications (ICC'02), Paper # J03-3, 2002
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", Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE, Volume: 2, 2001 Page(s): 755 -758 vol.2.
5. W. Stevens, "TCP/IP Illustrated", Volume 3, Addison Wesley, 1996.
6. Walter J. Gribbin, "Aeronautical Satellite Networks", IEEE 1988, CH2674-0-11/88/0000-135
7. J. Postel, "Transmission Control Protocol", Internet RFC 793, 1981.
8. M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control", RFC 2581, 1999
9. M. Allman, D. Glover, and l. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", RFC 2488, 1999
10. 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
11. M. Allman, S. Floyd, C. Partridge, "Increasing TCP's Initial Window", Internet RFC 2414, September 1998.
12. V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High performance", Internet RFC 1323, 1992
13. M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement Options", Internet RFC 2018, 1996.
14. J. Mogul and S.Deering, "Path MTU discovery", Internet RFC 1191, 1990.
15. Yadong Shang, Michael, Hadjitheodosiou, and John Baras, "Using Broadband Satellite Systems to Support Aeronautical Communications", Proc. 21st International Communications Satellite Systems Conference and Exhibit, 15-19 Apr. 2003, Yokohama, Japan.
16. K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP", ACM Computer Communications Review, Vol. 26, no. 3, pp. 5-21, July 1996.
17. Xiaoming Zhou, Xicheng Liu, and John Baras, "Flow Control at Satellite Gateways", Technical Research Report, ISR, University of Maryland, CSHCN TR 2002-19, http://www.isr.umd.edu
18. H. Balakrishnan, V. Padmanabhan, and R. Katx, "The Effects of Asymmetry on TCP performance", Proceedings of Third ACM/IEEE MobiCom Conference, pp. 77-89, Sept. 1997.
19. T. Lakshman and U. Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss", IEEE/ACM Transactions on Networking, vol. 5, no. 3, pp. 336-350, June 1997.
20. Xiaoming Zhou and John S. Baras, "TCP over GEO satellite hybrid networks", in Proc. IEEE MilCom Conference, 2002.
21. S. Floyd and T. Henderson, "The New Reno Modification to TCP's Fast Recovery Algorithm", Internet RFC 2582 (Experimental), April 1999.

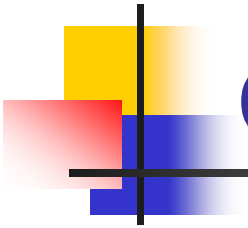
UNIVERSITY OF MARYLAND

CENTER FOR SATELLITE AND HYBRID COMMUNICATION NETWORKS

PROTOCOL SUPPORT FOR A NEW SATELLITE-BASED AIRSPACE COMMUNICATION NETWORK

Yadong Shang, Michael Hadjitheodosiou, John Baras

Center for Satellite & Hybrid Communication Networks
Institute for Systems Research, University of Maryland,
College Park, MD 20742, USA
shangyd@glue.umd.edu, michalis@isr.umd.edu,
baras@isr.umd.edu



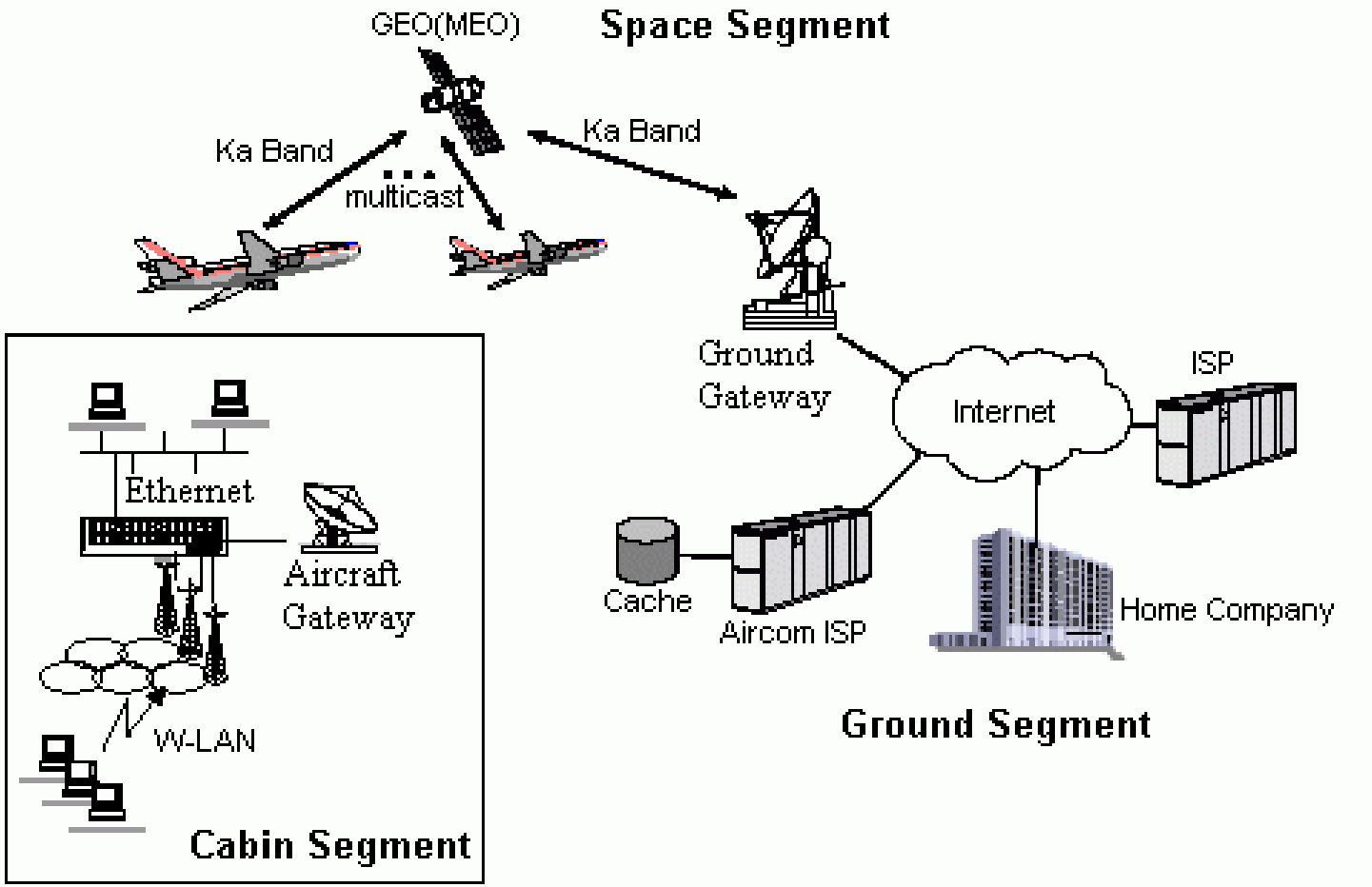
Contents

- Introduction
- Aeronautical Satellite Network
- End-to-End TCP solution
- TCP splitting solution
- Conclusion and Future work

Introduction

- Significance
  - Increased air traffic volume vs. old communication system
  - Use Satellite technology for aeronautical communication
  - Internet data services for passengers on flight
  - TCP/IP protocol support
- Objectives
  - Evaluate TCP performance on aeronautical network
  - Design a better transport protocol

Aeronautical Satellite Network



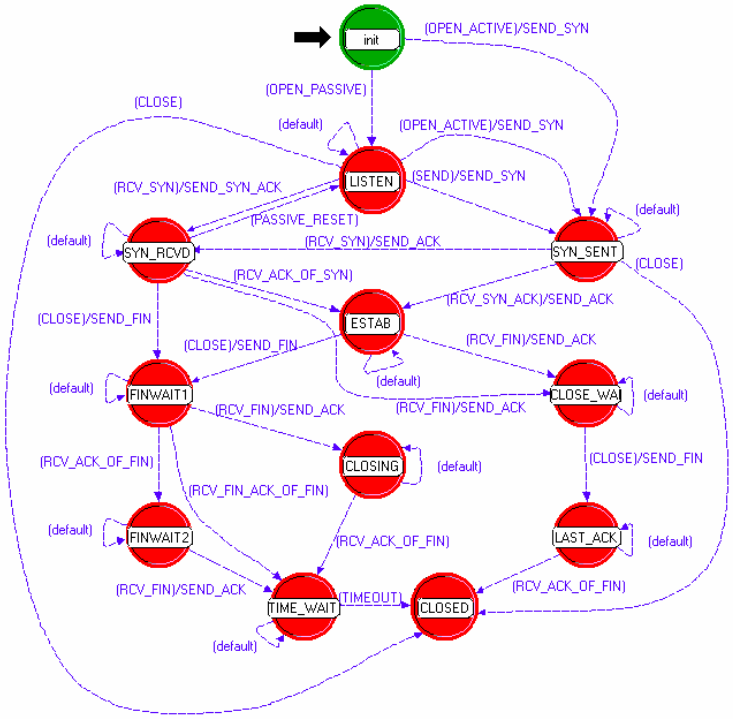
Network Characteristics

- Satellite Channel Characteristics
  - long propagation delay,
  - large bandwidth delay product,
  - occasional high bit error rate,
  - bandwidth asymmetry
- Aeronautical network
  - Mobile Aircraft
  - En-route Low BER
  - FIFO Satellite Channel
  - Intermittent connectivity
  - Variable Round Trip Time

TCP Operation

- Flow Control: Sliding window

  Received window=receiver Buffer size
- Congestion Control: Congestion window
  - Slow start
  - Congestion avoidance
  - fast retransmission
  - fast recovery
- Error Control: acknowledgement, timer, and retransmission

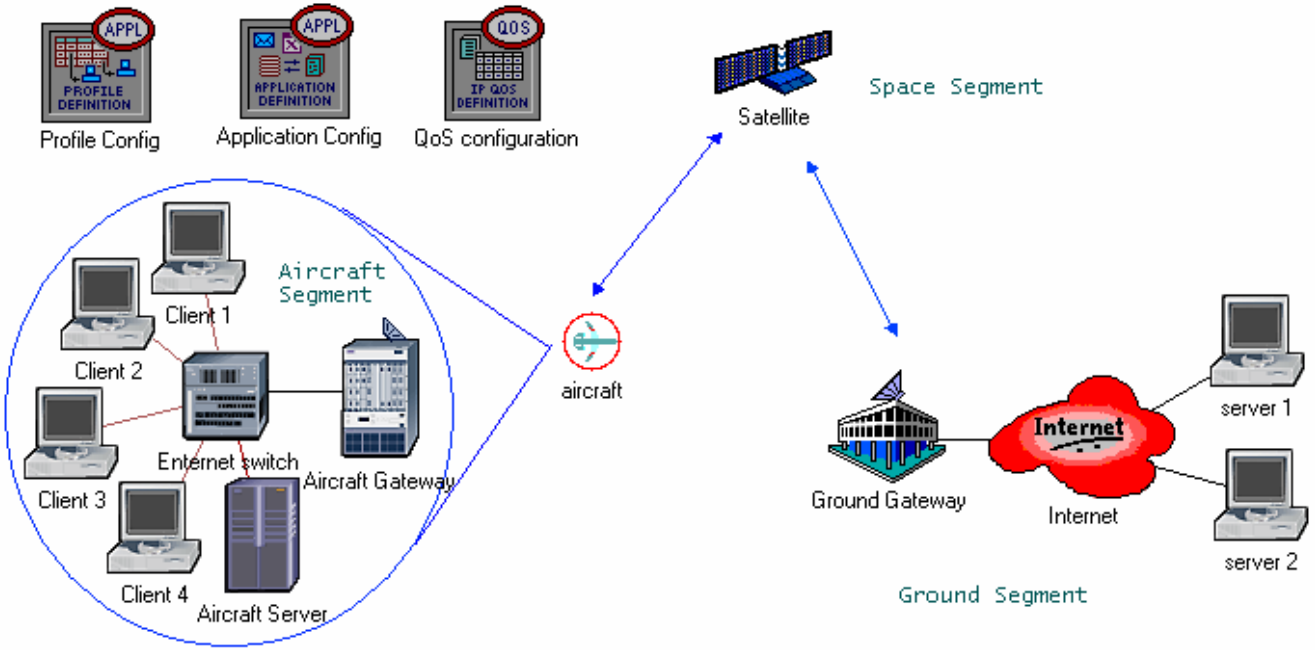


End-to-End TCP solution

Satellite Hybrid Network	*TCP Problems*	*End-to-End TCP Solution (Flavors and Extensions)*
Long propagation delay	Spend long time in Slow Start	Large Initial Window (4 MSS)
Large bandwidth-delay product	16bits Window	Window Scaling (multiple losses in one window)
High bit error rate	Drop its congestion window to a small size	Can not use fix window, TCP SACK for recovery
Bandwidth asymmetry	Increase ACKs delay	Priority Queue

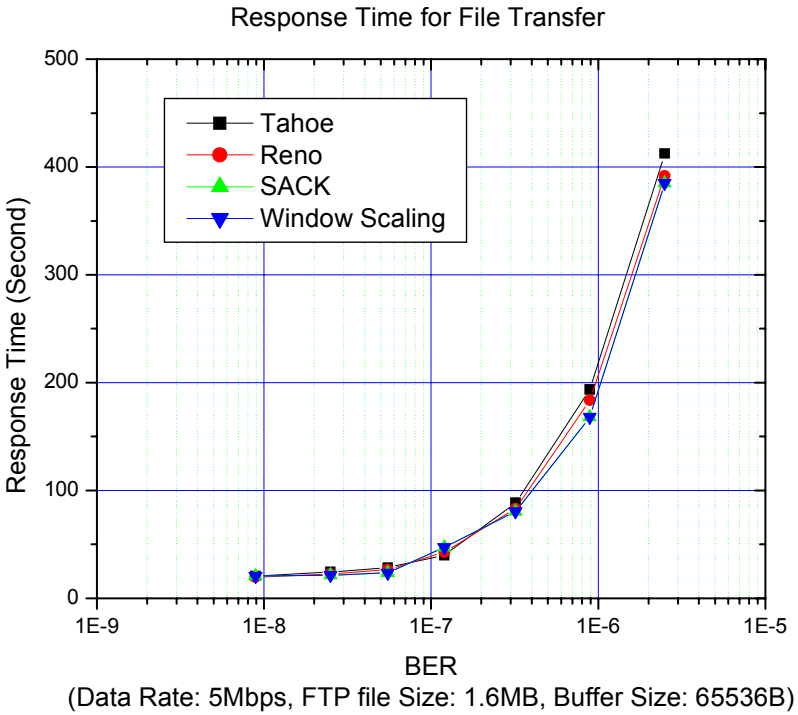
Other mechanisms: Path MTU discovery, Forward Error Correction, Ack filtering.

Experiment setup

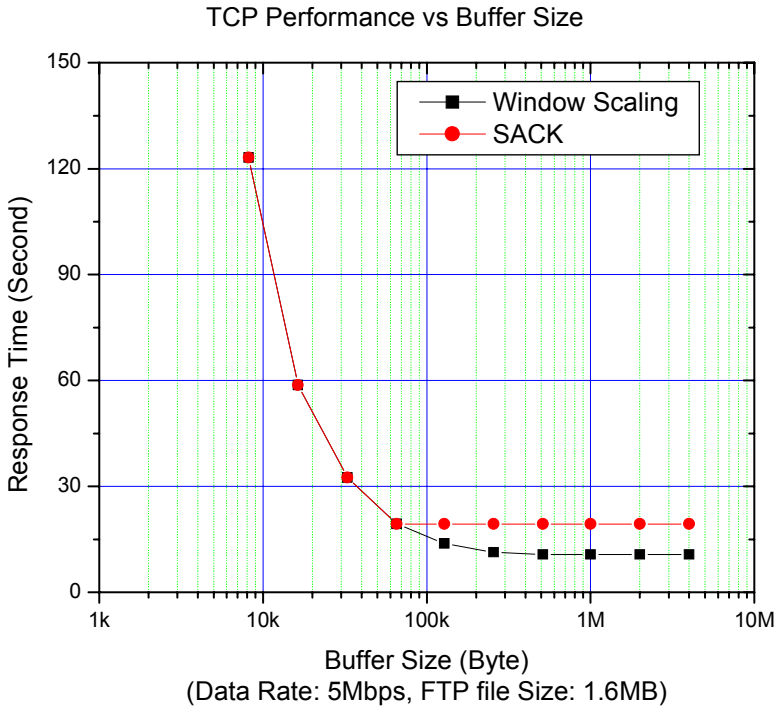


Tahoe	Fast Retransmit
Reno	Fast Retransmit and Fast Recovery
SACK	Selective Acknowledge
Window Scaling	SACK and Window Scaling

End-to-End TCP Performance



(Data Rate: 5Mbps, FTP file Size: 1.6MB, Buffer Size: 65536B)



(Data Rate: 5Mbps, FTP file Size: 1.6MB)

Unsolved problems

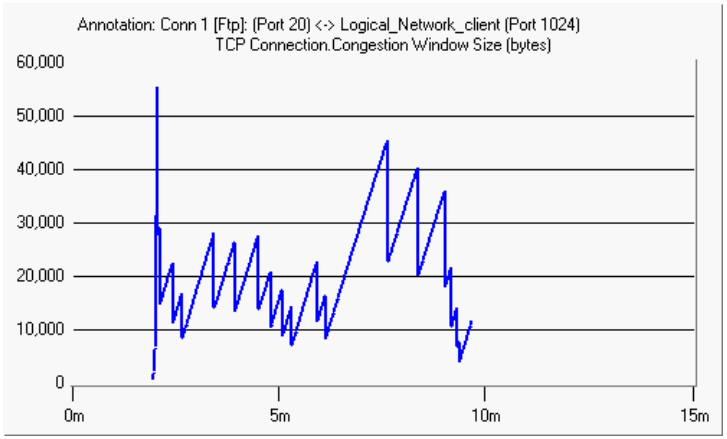
- Small operational window

  Large propagation, slow start and link layer corruption

  RecvWin / GndRTT = SatWin / SatRTT

  Throughput=min (SatBW, Buff / (SatRTT + GndRTT))

  Backlog packets=max(o,Buff–SatBW*(SatRTT+TerrRTT))



BER=1E-7

DS1=1,544,000bps

RTT=580ms

Unsolved problems (cont.)

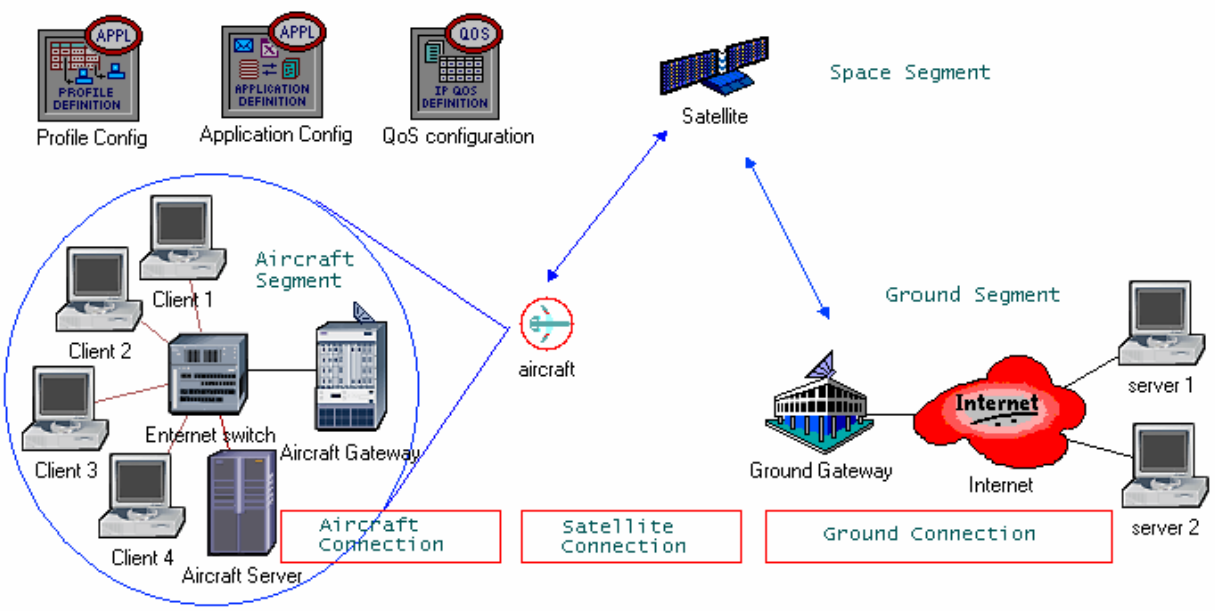
- Asymmetric link

  Congestion in reverse link: ACK filtering, Priority Queue

- TCP Fairness

  TCP throughput is inverse proportional to RTT, so TCP connection with large RTT does not get its fair share of the bandwidth when it competes with the connections with shorter RTT

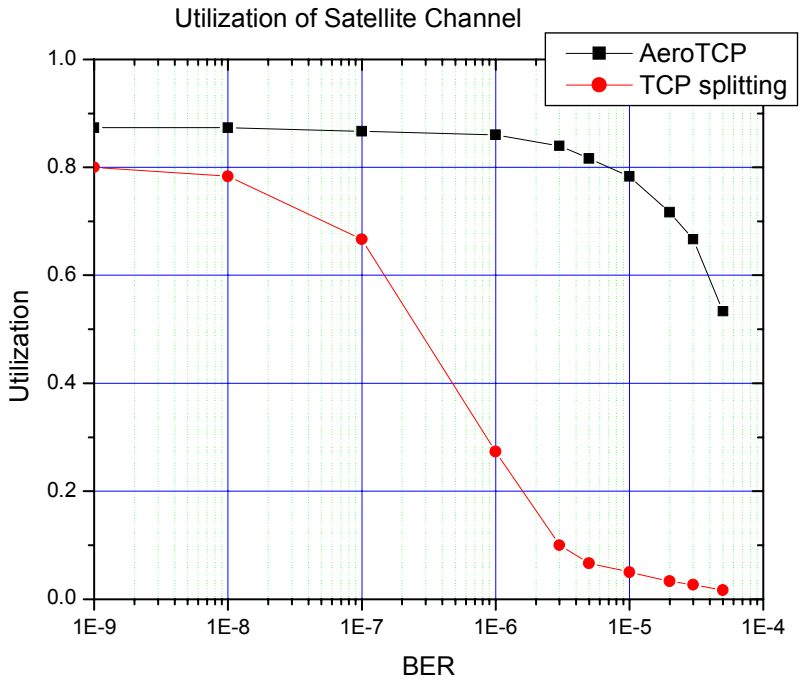
TCP Splitting Protocol

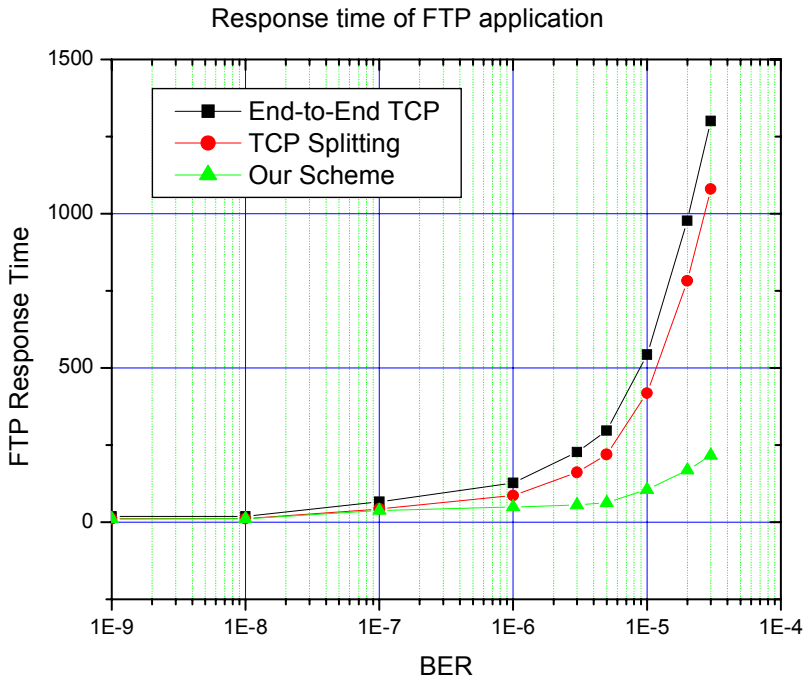


AeroTCP

- Flow Control: Fixed window for each connection
- Congestion Control: FIFO Channel, No congestion
- Error Control: One duplicated ACK for fast retransmission and partial ACK for burst loss recovery

TCP splitting protocol performance





Study Scenario: 2 Connections, TCP/IP/PPP, FTP application, File size=1.6MB, DS1=1,544,000bps, RTT=580ms (500ms for satellite link and 80ms for terrestrial link)

AeroTCP (Our scheme), TCP splitting (TCP SACK for both connections), and End-to-End TCP

Conclusion

- We observed degradation in TCP performance for large bandwidth-delay product networks such as aeronautical satellite systems. If the right TCP options are used and congestion is light, TCP can work well for large file transfers even over GEO links.
- It is difficult for an end-to-end TCP solution to solve the problems in the aeronautical satellite networks, our connection splitting based solution, AeroTCP, can maintain high utilization of the satellite link and has better performance than end-to-end solutions.

Future Work

- Modeling the realistic Ka-band satellite channel (Uniform BER in OPNET, burst error)
- Support other applications and services (FTP, HTTP, TELNET, Email, Telephone, Video)
- Support more aircraft and global coverage (MAC layer protocol, spot beam handover, ISI)