TCP PERFORMANCE IN A GEOSTATIONARY SATELLITE ENVIRONMENT

Daniel R. Glover
Project Engineer
NASA Lewis Research Center

Hans Kruse
Director, McClure School of Communication Systems Management
Ohio University

ABSTRACT

Geostationary satellite links have characteristics that can degrade the performance of TCP. The propagation time for a geostationary satellite link is on the order of 0.25 seconds. Thus a satellite can add approximately 0.5 seconds to the round trip time experienced by a TCP packet and the corresponding acknowledgement. This delay time is the primary cause of TCP inefficiency over satellites. There are some things that can be done to enable TCP to better utilize satellite channels. These practices include using the fast retransmit and fast recovery algorithms, the LFN extensions to TCP, and the use of selective acknowledgements. This paper briefly describes the satellite environment, the problems arising from that environment, along with best common practices for mitigating the problems.

1. INTRODUCTION

A large portion of Internet traffic uses the Transmission Control Protocol (TCP) for reliable data delivery. TCP is just one protocol in the family of Internet protocols known as the TCP/IP protocol suite maintained by the Internet Engineering Task Force (IETF). TCP is used for World Wide Web data transport as well as by the File Transport Protocol (FTP). TCP started out as a research protocol in the early 1970s [CerfKahn74], but has evolved substantially over the years. In October 1986, the Internet experienced congestion collapses and in response, several algorithms were developed to prevent or control network congestion [Jac88]. These algorithms were introduced into the 4.3BSD (Berkeley Software Distribution) operating system and that version of TCP became known as Tahoe (after the name of the machine that 4.3BSD was being ported to at the time). TCP-Reno was released in 1990 with additional changes and is the minimum benchmark for TCP today.

Satellites were a part of the ARPANET and TCP experiments using Satnet were performed in the 1980s. The explosive growth of the Internet took place over terrestrial lines without much concern for satellite users. However, as the Internet continues to increase in importance for world-wide communications, satellite connections are again being used to deliver Internet traffic. Users in remote areas, users in regions without well-developed terrestrial networks, and mobile users are all potential beneficiaries of satellite services. Given this environment, there has been a lot of interest in getting the TCP/IP protocols, especially TCP, to work more efficiently over long delay paths. There are Internet services available over satellite today, so TCP obviously works over satellite, but efforts are underway to enable better utilization of satellite links.

The long propagation delay of a geosynchronous satellite path can be a fundamental problem for some applications. Interactive applications, such as a telnet session or a game of Quake, are going to be frustrating for many satellite users. Even a basic TCP implementation will perform well for low-volume applications, including web browsing, especially when the satellite link is shared among several users. On the other hand, high-volume data transfers such as medial images or large web pages will require the implementation of high-performance TCP options as described in this paper.

1.1 Background

Communications satellites were developed in the 1960's to provide international telephone service. The geosynchronous orbit (GSO), at an altitude of 22,300 miles, allows ground station antennas to remain pointed in one direction. The resulting 250 ms delay due to the propagation time up and back proved not to be a problem for telephone conversations (once echo canceling was introduced). In the 1970s, satellites were used primarily for telephony and for television program distribution. Today, 45Mbps satellite services at bit error rates of 1 in 10^7 are commercially available. Very small aperture terminal (VSAT) satellite systems provide 64kbps or fractional-T1 data channels. Satellite frame relay services are also being offered.

New GSO satellite systems are being designed which will use higher frequencies (Ka-band) to offer services from low rates up to 18 Mbps at bit error rates less than 1 in 10^{10}. Low Earth orbit (LEO) satellite systems are also being built that will provide low data rate services, but with delays comparable to, or less than, terrestrial systems. NASA's Advanced Communications Technology Satellite (ACTS) has pioneered the use of technologies which these systems will use. The concatenated error correction coding used by ACTS has provided a link quality approaching fiber, that is, the link is either error free or not available. As error correction coding is used more frequently in satellite systems, the problems due to errors should be mitigated. However, there is a large installed base of satellite ground stations that do not use advanced error correction, many of which are in underdeveloped countries which might be slow to implement technology upgrades.

Future satellite systems are being designed which will provide personal communication services. There are LEO systems being designed and built that will offer mobile voice, messaging, and low-rate, remote Internet services. The LEO systems have low delay as well as lower power requirements compared to GSO systems, but require a larger number of spacecraft to provide coverage. Several new high frequency (Ka-band) GSO satellite systems are scheduled for launch in the next few years. These systems will offer services at data rates from 16 kbps to nearly 20 Mbps.

Internet services are available to the end user today over satellite systems. Some of these are closed systems which can be designed for optimal transfer over high delay links and can take advantage of the asymmetric nature (that is, higher transfer rates in one direction than the other) of most Internet traffic. One satellite system is offering end users today a 400 kbps peak incoming data rate, which is much higher than the typical 28.8 kbps modem. In this paper, high data rate means over 1 Mbps

TCP is a windowed protocol that uses acknowledgements from the receiver to clock the sender's output. New data cannot be introduced into the network until an acknowledgement has been received by the sender indicating that data has been removed from the network. The windows control the amount of data that can be transmitted at any particular time. A maximum window size (generally corresponding to the receiver's buffer size) is negotiated at the connection setup and a congestion window is used to control the amount of data introduced into the network. Two algorithms, slow start and congestion avoidance, make TCP behave like a good neighbor and prevent data from being blasted out onto the network at the beginning of a connection or after a loss. These algorithms gently probe the network to find the proper level of data transmission. TCP assumes any loss is due to congestion of the network and throttles back the transmission rate severely.

1.2 Advantages of satellites

Satellites have not played a major role in the Internet up to this time, so one might question why any effort should be spent on making TCP work more efficiently over satellite. Satellites have some potential advantages that should be considered by architects of future networks. One advantage is that satellites, being wireless, do not require an infrastructure of cables and therefore can bring the network to hard-to-reach areas (e.g., in the middle of the ocean) or to economically disadvantaged areas that cannot afford a wired infrastructure. Another advantage is that a satellite is a natural for broadcasting the same data to many locations. There may be some future opportunities to take advantage of the natural broadcast characteristics of satellites to make multicast more efficient in the overall network.

1.3 Fooling TCP

It is possible to get around some of the inefficiencies of TCP by going around the protocol. This must be thought out carefully, because the congestion control in TCP is there for a reason. However, in some situations, such as at the edge of the network, it may be possible to translate to another protocol for a satellite hop or to spoof TCP by acknowledging packets before the satellite hop. It is possible to use multiple TCP connections to improve overall throughput. This can be done by splitting a large file up and sending the pieces over separate connections. It is also done by some browsers, but using multiple connections has been frowned on as an unfair practice by some computer scientists.

While bypassing TCP's slow start and congestion control algorithms may seem attractive in some cases, care must be taken to make sure that the network as a whole is not abused. This is most easily done at the edge of the network where the last leg of the connection is controlled. It might make sense in some cases to disable congestion control completely (for a deep space probe, for example, if it were using TCP for some reason). On the Internet, if one person disabled their congestion control, they might notice an improvement in performance, but if everyone disabled their congestion control, there would be no more Internet. In this paper, we are concerned with how to make TCP work better for everyone including satellites, not with satellite-specific solutions.

2. SATELLITE LINK CHARACTERISTICS

Satellite links have characteristics that can degrade the performance of TCP. These include:

1) Long delay paths (long feedback path)
2) Large delay*bandwidth product
3) Transmission errors (as opposed to congestion loss)
4) Limited bandwidth
5) Asymmetric use
6) Variable Round Trip Times (for some constellations)
7) Intermittent connectivity (handoffs and outages)

By far, the most common type of communications satellite today uses the geostationary orbit. Such satellites have an altitude of 22,300 nautical miles and orbit the Earth once a day, thus appearing to be stationary in the sky. These satellites do not normally suffer from characteristics number 6 and 7 above. Characteristic number 5 is a result of the way satellite systems are typically configured for end users. For the rest of this paper, we will concentrate on geostationary orbit satellites and characteristics 1 through 3. The term "satellite" will be taken to mean "geostationary satellite" below.

2.1 Long Delay Path

The first characteristic, long delay path, is the primary problem addressed in this paper. It leads to the second, large delay*bandwidth product, even though satellites typically have limited bandwidth. It also exacerbates the problems arising from channel errors.

Because each satellite in the path will contribute about 0.5 seconds to the round trip time (RTT), connection establishment will take longer over satellite. The use of TCP for Transactions (T/TCP) [RFC 1644] can help make connection setup quicker by maintaining a connection for use by multiple transactions (request/response sequences). HTTP1.1 should provide benefits for satellite channels for the same reason. Care should be taken in network design with regard to DNS server location to avoid DNS lookups over the satellite.

To avoid congestion in the network, TCP ramps up the connection rate using the "slow start" algorithm [Jac88]. Because slow start is directly proportional to the round trip time (RTT), the long RTT makes slow start over a satellite much slower than for terrestrial connections. Slow start time (the time to reach full connection speed after connection establishment in the absence of loss) is given by:

$$\text{Slow Start Time} = \text{RTT} \log_2 (W)$$

Where W is the window size in packets [Jac88].

When using the delayed acknowledgements algorithm (which acknowledges two packets with one ACK), the slow start time increases to roughly [PS97]:

$$\text{Slow Start Time} = \text{RTT} \log_{1.5} (W)$$

Thus, avoiding the use of delayed ACKs during slow start would appear to help in a satellite channel in the case of small file transfers (however, this requires further study). For large file transfers, delayed ACKs can help avoid congestion loss by controlling the burstiness from large bunches of ACKs and so is beneficial. Delayed ACKs may also be helpful in highly asymmetric channels where ACK bunching may occur which can lead to burstiness and congestion loss. However, delayed ACKs also cause slower congestion window growth during congestion avoidance. This is still a topic for research, however it is clear that for small transfers delayed ACKs hurt, while for large transfers they can help. An obvious approach is to turn delayed ACKs on only after slow start has gotten the connection up to a reasonable rate.

The use of selective acknowledgements (SACK) [RFC 2018] can allow better handling of the retransmission queue in the event of multiple packet losses in a single window.

2.2 Large Delay*Bandwidth Product

The delay*bandwidth product is a measure of the capacity of a link in terms of the maximum amount of data that can be in flight at one time. This is a critical parameter in windowed protocols such as TCP. The window size must be equal to the delay*bandwidth product to fully utilize the channel at steady-state.

The largest window size in TCP without options is 65,535 bytes. For RTT = 0.5 seconds, this limits the throughput to around 1 Mbps per connection. To reach a steady-state condition where a large delay*bandwidth channel (such as one that includes a satellite) is utilized as fully as possible, the use of long fat network (LFN) extensions [RFC 1323] is needed. These options include window size scaling (or large windows), timestamps, and protection against wrapped sequence numbers.

Many computer scientists assume that the only thing to worry about on a satellite connection is using large windows. They assume the problem was solved many years ago (the large window option has been around since 1990) [RFC 1185]. Using LFN options is something that most users will have to do someday, but they are not a solution for all the problems faced by satellites. The large windows option allows TCP to reach a steady state that uses the full bandwidth available, but does nothing for the long feedback path problems.

TABLE 1 Window Sizes for a Single TCP Connection

Rate	RTT	B*D (window)	LFN needed?

(bps)	(sec)	(bytes)	(Yes, No)
33.6k	0.1	420	N
33.6k	1	4200	N
33.6k	2	8400	N
128k	0.1	1600	N
128k	1	16000	N
128k	2	32000	N
1.55M	0.1	19375	N
1.55M	1	193,750	Y
1.55M	2	387,500	Y
10M	0.1	125,000	Y
10M	1	1,250,000	Y
10M	2	2,500,000	Y
155M	0.1	1,937,500	Y
155M	1	19,375,000	Y
155M	2	38,750,000	Y

Table 1 shows the results of window size calculations for selected data rates and round trip times. Round trip time for a coast-to-coast terrestrial network is represented by 0.1 seconds. A network including a single satellite link should come in under 1 second. Some remote sites (e.g., Alaska) may experience RTT of 2 seconds. Note that for a 2 sec RTT, you don't need large windows for a channel with bandwidth less than 262,140 bps. For a 0.1 sec RTT you will need large windows when the rate is a little greater than 5.2 Mbps. These numbers are for individual TCP connections. If multiple TCP connections are in use on the satellite link, each individual connection can utilize data rates up to the limit dictated by the window size; the utilization of the satellite link will be the sum of the individual connections. As stated earlier, no action may be required if many users are sharing the circuit, and each users expects to use only a modest data rate. If a single TCP connection is expected to fully use the available bandwidth, the considerations in this section apply.

TABLE 2 Round Trip Times Corresponding to the Maximum Standard Window Size At Various Rates

Rate (bps)	RTT for 64KB B*D (seconds)
33.6 k	15.6
128 k	4.096
1.55 M	0.340
10 M	0.050
155 M	0.003
622 M	0.0008

Table 2 shows the results of a similar calculation in reverse, the RTT that gives a delay*bandwidth product equal to the largest possible window in TCP without using window scaling options. Note that a 128kbps connection doesn't need large windows even for several satellite hops, while a 155Mbps or 622Mbps connection would need large windows routinely even for terrestrial channels. The low rate numbers in these tables may seem familiar today, but in a few years a reader might wonder why such ridiculously low rates are being examined. On the other hand, limits are being reached as to what can be done with the existing phone system and cost/benefit considerations may keep many consumers using their current modems at home. As data rates increase, more connections will need LFN extensions even over terrestrial networks because of the growth in bandwidth.

2.3 Transmission Errors

The long feedback time discussed in section 1.1 also affects the response of congestion avoidance, which is an even slower growth phase than slow start. In congestion avoidance, the congestion window can be increased by no more than one segment per RTT (and when delayed ACKs are used has an increase rate of only one segment per two RTTs).

A loss event in TCP is always interpreted as an indication of congestion and always causes TCP to enter congestion avoidance. If a loss occurs during slow start, then slow start is terminated and TCP enters congestion avoidance. Premature termination of slow start by losses other than congestion losses should be avoided because the congestion avoidance algorithm increases the transmission rate at a slower pace than slow start. A loss during steady-state operation will usually result in a drastic reduction of the effective window size at the sender, followed by congestion avoidance. Modern TCP implementations should use the techniques know as fast retransmission and fast recovery. The mechanisms attempt to recognize a case where a single TCP segment has been lost (often due to a transmission error); in this case TCP can bypass the congestion avoidance stage. To improve the handling of a loss of multiple TCP segements, the selective acknowledgement (SACK) option for TCP is needed. While this option and its performance characteristics have been extensively studied by network researchers, it is not yet widely available in end-user operating systems.

For TCP to operate efficiently, the channel characteristics should be such that all loss is due to network congestion. The use of forward error correction coding (FEC) on a satellite link should be used to bring the performance of the link to at least fiber quality. Because of the effect of long RTT, errors on a satellite link have more severe repercussions than on a lower RTT terrestrial channel, so an even better error performance is indicated, if possible [PS97]. There are some applications, such as military jamming, where FEC cannot be expected to solve the noise problem. Also, there may be a trade-off between coding complexity and delay.

3. IMPLEMENTATION ISSUES

Satellite users should keep abreast of updates to pertinent documents and ensure that the implementations they use are up to date (see STD-001). They should also follow the efforts of groups such as TCP Implementation working group. Efforts are currently underway to update RFC 1323 and RFC 2001. The TCP Over Satellite working group in the IETF has been working on identifying the problems with current versions of TCP and recommending best common practices for satellites.

There is a large installed base of TCP that does not necessarily support the options that can improve performance over satellites. Different implementations also vary in the degree to which they follow the TCP specifications. These specifications can be difficult to understand because in the past they have not been clearly written or available in one place. The TCP Implementation working group is working on correcting these problems.

There has been some talk about working on the next generation TCP, or TCPng. A decision was made not to work on TCPng when work on Ipv6 was begun, but now there is growing interest in fixing TCP. TCP has evolved through the use of extensions that make use of the option field in the header. It may begin to make sense to migrate some of these optional extensions to a more regular position. Resistance to changing TCP could be due, in part, to a wish to avoid the congestion collapse of 1986. The Internet has become so complex that it is not possible to predict perfectly how it will react to a change in TCP.

4. RECOMMENDATIONS

Satellite users should make sure that the TCP implementation that they are using meets the Internet Standards for host requirements and for TCP [STD-001][STD-003][STD-007].

The TCP implementation should also meet the RFC for congestion control [RFC 2001]. Specifically, a TCP implementation should include slow start, congestion avoidance, fast retransmit and fast recovery.

Satellite users with a large delay*bandwidth product should also use implementations that include the proper options, known variously as LFN, high performance, or large windows [RFC 1323].

Satellite users should use selective acknowledgements (SACK) [RFC 2018].

Satellite users should use HTTP 1.1 capable servers and browsers [RFC 2068].

5. CONCLUDING REMARKS

It is important to remember that TCP works over satellite. There are several companies that offer Internet access over satellite today. TCP suffers from some inefficiencies over a long delay path, but it is not unworkable. There are efforts underway to make TCP work better, not only over satellite, but for all users. There are also efforts under way to improve the performance of TCP over satellite at high data rates. The performance issues described above are on a per connection basis. Trunking TCP channels together does not affect the performance of TCP.

To fully utilize the available bandwidth it is necessary in some, but not all, cases to use the TCP large windows option. Although this solves the steady state condition, the dynamics of getting a connection up to speed or recovering after a loss is slow (proportional to the round trip time) over satellites due to the long delay path. Efforts are underway to improve this situation.

6. REFERENCES

[AHKO97] Allman, Mark; Chris Hayes, Hans Kruse, and Shawn Ostermann: "TCP Performance Over Satellite Links." In Proceedings of the 5th International Conference on Telecommunication Systems, March 1997.

[All97] Allman, Mark: "Improving TCP Performance Over Satellite Channels." Master's thesis, Ohio University, June 1997.

[CerfKahn74] Cerf, V. and R. Kahn: "A Protocol for Packet Network Intercommunication," IEEE Transactions on Communications, vol. 22, no. 5, pp. 637-648, May 1974.

[Jac88] V. Jacobson: "Congestion Avoidance and Control," Computer Communication Review, vol. 18, no. 4, pp. 314-329, Aug. 1988. ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z.

[Jac90] V. Jacobson: "Modified TCP Congestion Avoidance Algorithm," end2end-interest mailing list, April 30, 1990. ftp://ftp.isi.edu/end2end/end2end-interest-1990.mail.

[Mar78] James Martin: Communications Satellite Systems. Prentice Hall, 1978.

[PS97] Craig Partridge and Tim Shepard: "TCP Performance Over Satellite Links." IEEE Network, 11(5), pp. 44-49, September/October 1997.

[STD-001] Postel, J.: "INTERNET OFFICIAL PROTOCOL STANDARDS," June 1997. (Also RFC2200)

[STD-003] Braden, R., Ed.: "Host Requirements," October 1989. (Also RFC1122, RFC1123)

[STD-007] Postel, J.: "Transmission Control Protocol – DARPA Internet Program Protocol Specification," September 1981. (Also RFC-793)

[RFC 1185] Jacobson, V.; R. T. Braden, L. Zhang: “TCP extension for high-speed paths.” Oct-01-1990. (Obsoleted by RFC 1323)

[RFC 1323] Jacobson, V.; R. Braden and D. Borman: "TCP Extensions for High Performance, February 1991.

[RFC 1644] Braden, R.: “T/TCP -- TCP Extensions for Transactions Functional Specification,” July 1994.

[RFC 2001] Stevens, W. Richard: “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms,” January 1997.

[RFC 2018] Mathis, Matt; Jamshid Mahdavi, Sally Floyd, and Allyn Romanow: “TCP Selective Acknowledgment Options,” October 1996.

[RFC 2068] Fielding, R.; J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee: “Hypertext Transfer Protocol -- HTTP/1.1,” January 1997.

Daniel R. Glover is an electronics engineer in the Satellite Networks and Architectures Branch of the NASA Lewis Research Center in Cleveland, Ohio. He received a Ph.D. in engineering science from the University of Toledo. Dr. Glover is the author of PlanetQuest published by the Space Telescope Science Institute’s Excellence in Education program. He is currently conducting research on Internet protocols using testbeds, simulation, and experiments over NASA’s Advanced Communications Technology Satellite (ACTS).

Hans Kruse, associate professor of communication systems management at Ohio University, is director of the J. Warren McClure School of Communication Systems Management. He received a Ph.D. in theoretical nuclear physics from Vanderbilt University. Dr. Kruse, who has taught at Ohio University for six years, continues to teach and advise students and engage in scholarly research. He has conducted numerous experiments on Internet protocols using NASA’s Advanced Communications Technology Satellite (ACTS).