TCP OVER SATELLITE...
The Final Frontier

Christopher Metz • Cisco Systems • chmetz@cisco.com

Communication satellites have been in commercial use for more than three decades. Driving their early deployment were military communications, international telephony, and broadcast TV. Now these same satellites are being used to transport TCP/IP traffic between distant locations, and to offer Internet access. Satellites have thus become the celestial link of the global Internet, an "instant" infrastructure in the sky. Data communications over satellite has a clear appeal in areas where a terrestrial communications infrastructure is either not feasible or not present. Satellites also possess a natural broadcast capability that enables a single sender to direct a communications stream up to satellite, and then have it "reflected" down to a large downstream population. Many commercial Internet service providers now use satellite links in their networks as a less costly backbone link to land-based alternatives, especially when great distances or an ocean is involved, or as a means of bypassing a backbone or ISP network to deliver an advanced service to the edges of a network. Home Internet access via satellite is also now available: DirecPC, a satellite downlink, is capable of delivering Internet traffic at up to 400 kbps to the home PC.

The rapid growth of satellite communications is evolving the TCP/IP protocol suite in positive ways. In particular, enhancements to the Transmission Control Protocol (TCP) to address the challenges of satellite transmission will benefit *all* high-bandwidth TCP communications. TCP, as you will remember, is the predominant unicast transport protocol used by Internet applications such as Telnet, FTP, and HTTP.[1]

A TCP sender uses acknowledgments transmitted by the receiver to clock its sending rate and to ensure reliable data delivery. TCP also employs a window-based flow control mechanism to prevent buffer overruns in the receiver and network. The ability of TCP to maximize the link utilization of a satellite channel is being challenged by the inherent delays associated with space communications and some of TCP's own behaviors.

In this column, we will help you understand the basics of using TCP for satellite transmission and describe the changes you can expect to see in the TCP protocol itself as a result of the increase in use of satellites for TCP/IP traffic.

Satellite Basics

A satellite acts as an overhead relay or repeater for communications between two geographically remote locations. As illustrated by the sample configuration in Figure 1, a router-1 is connected to a ground (earth) station that takes the incoming traffic, converts it to a microwave signal, and transmits over a specific frequency up to the satellite. The satellite receives the signal, amplifies it, and then transmits over the downlink on a separate frequency. The ground station dish then receives the signal, converts it to a terrestrial link format, and passes it on to router-2. The bandwidth (in bps, or bits per second) supported over the satellite channel depends on factors that include the allocated frequency range, the modulation technique, error correction control, and the link-layer protocol. The data rates therefore that can be supported range from the very low (for example, 9.6 kbps) up to speeds commensurate with high-bandwidth land links (for example, NASA's Advanced Communications Technology Satellite [ACTS] Gigabit Satellite Network is demonstrating 622 Mbps communications links). For a comparison of these speeds with other deployed technologies, see the sidebar, "Speed Limits on the Internet."

Satellite configurations differ based on the altitude of their orbit. A satellite in geostationary (GEO) orbit operates at approximately 22,000 miles above the earth and orbits once every 24 hours. To always remain visible to the ground stations below, a communications GEO satellite remains fixed above a point on the equator and orbits in the equatorial plane moving at the same angular velocity and in the same direction as the earth. GEO satellites used for weather forecasting or spying will still orbit once every 24 hours but because their orbit is not in the equatorial plane, they will move relative to a fixed point on the earth. Because of their altitude, GEO satellites can illuminate a large portion of the earth's surface and are the most common communications satellite technology in use today. Examples of GEO satellite constellations are Spaceway from Hughes Corporation and Astrolink from Lockheed Martin. However, the speed of light combined with the long distance between ground station and satellite introduces approximately 250 ms of one-way propagation delay for each satellite link in the path. Low earth orbit (LEO) satellites, on the other hand, operate at several hundred miles above the earth, but provide significantly less coverage. A constellation of LEO satellites must be deployed to provide complete coverage, and the ground station must switch over from one LEO satellite that is falling out of range to another coming into range. The primary advantage of LEO satellites is their lower propagation delay of about 10-20 ms. Well-publicized LEO projects include the Teledesic and Iridium satellite networks.

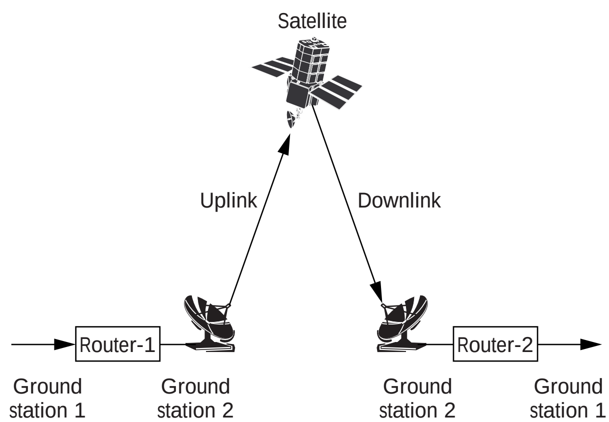

Figure 1. Inter-router satellite link.

Satellite Network Performance

There are three factors that most affect throughput for TCP/IP over a satellite channel.[2] They are

- *Long Feedback Delay*. A TCP sender is dependent on timely network feedback for rate adjustment, congestion avoidance, and error recovery. A delay of 0.5 seconds imposed by the RTT (round-trip time) on a GEO satellite channel will delay the execution of these functions and affect throughput.
- *Large Bandwidth-Delay Product*. The product of bandwidth * delay (BD) determines how much unacknowledged data a TCP sender should transmit into the network to fully utilize the capacity of the link. The delay in this equation is the RTT, and the bandwidth is the maximum bandwidth of the slowest link in the path. For satellite channels with long RTTs, as well as terrestrial links with very large bandwidths, the BD product can be quite high, which means that the TCP sender and receivers must be capable of handling larger amounts of data in a single transfer window.
- *Transmission Errors*. A satellite channel may exhibit high bit-error rates (BER) due to factors including atmospheric conditions, RF interference, a weak signal, and so on. When, due to corruption, a packet is not successfully delivered to the destination and acknowledged, a TCP sender interprets this as network congestion and enters into a congestion avoidance state that can substantially reduce overall throughput. Unfortunately, there is no way for TCP to know that corruption and not network congestion caused it to reduce its sending rate.

Additional factors that serve to reduce throughput include asymmetric routing and variable RTTs. The former occurs when, due to the high cost of ground station equipment, an alternate, slow-speed terrestrial back channel (for example, dial-up modem connection) is used to communicate in one direction, while the higher speed satellite channel is used for the other. The resulting bandwidth asymmetry can lead to ACK (acknowledgment) starvation, in which TCP acknowledgments flowing back to the sender over the slow-speed back channel arrive too slowly to allow the sender to fill the satellite channel in a timely fashion. Variable RTTs can occur in a LEO satellite configuration when the ground station is handed off to a different satellite in the constellation. If not properly performed, packet loss can result, causing the TCP sender to reduce its transfer rate.

Speed Limits on the Internet

The thousands of concatenated networks that form the global Internet contain a wide variety of transmission technologies that vary in distance, topology, cost and speed. How do satellite transmission speeds, which range from 9.6 kbps up to 622 Mbps, stack up against other popular technologies? A dial-up analog circuit provides a single residential subscriber with a convenient and simple 28.8 kbps (kilobit per second) access connection into the Internet. Local area networks (LANs) such as Ethernet offer unicast and multicast connection services to applications and wide area network (WAN) routers in "campus" proximity to each other at speeds up to 1 Gbps (gigabits per second). Routers in turn are connected to each other by WAN point-to-point links covering hundreds and thousands of miles at speeds ranging from 64 kbps up to OC48 (2.4 Gbps). Now emerging technologies such as Distributed Wave Division Multiplexing (DWDM) in conjunction with the existing but growing fiber-optic installed base will make it possible to support link speeds up to OC192 (10 Gbps) and beyond.

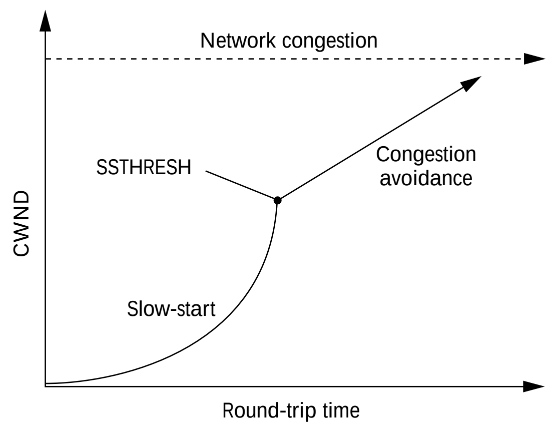

Figure 2. Slow start and congestion avoidance schemes as supported in standard TCP implementations.

TCP Behaviors to Avoid Congestion

TCP employs several intertwined congestion control mechanisms that enable a TCP sender to adjust its information transfer rate on the basis of available network capacity.[3] These algorithms were engineered into the TCP protocol over time to support several notions. First, a TCP sender initially has no idea of the available network capacity. Second, blasting a bunch of packets into the network following connection establishment is a bad idea that could lead to congestion, packet loss, retransmissions, and degraded throughput. And third, to best utilize network capacity and achieve optimal application throughput, TCP should make all fair and reasonable attempts to fill the link with as much data as possible.

The Slow-Start Algorithm. Slow-start, as the name implies, causes a TCP sender to gradually increase the amount of data injected into the network following connection establishment, the restart of an idle connection, or a TCP connection time-out. Two state variables are defined and maintained by the TCP sender: CWND and SSTHRESH. CWND limits the amount of data a sender can transmit before an ACK is required; SSTHRESH is the threshold of the slow-start process.

Initially, the TCP sender transmits one segment (or packet) into the network and waits for an ACK. For each subsequent ACK the sender receives, the CWND is increased by one segment, resulting in an exponential increase in the amount of data sent into the network. The slow-start phase is terminated when CWND equals the value of SSTRESH, the receiver's advertised window, or when congestion occurs. TCP thenceforward—in what is termed the congestion avoidance phase—takes a more conservative approach in probing for network capacity. In this phase, the CWND is incremented by at most one segment per RTT for each ACK received until congestion is detected, or the receiver's advertised window is reached. Congestion avoidance thus increases the TCP sending rate in a linear fashion. The graph in Figure 2 provides a relative illustration of the slow-start and congestion avoidance schemes as supported in standard TCP implementations. Slow-start and congestion avoidance were first suggested by Van Jacobson in his famous 1988 paper on the subject following a series of congestion collapses.[4]

The Fast Retransmit Algorithm. Fast retransmit enables a TCP sender to rapidly recover from a single lost packet, or one that is delivered out of sequence, without shutting down the CWND. When a TCP receiver detects the loss of a packet, it acknowledges subsequent packets with the ACK number of the last correctly received packet. When the TCP sender receives three duplicate ACKs, it then retransmits the lost packet. The receiver responds with a cumulative ACK for all packets received up to that point. Fast recovery is based on the notion that, since the subsequent packets generating the duplicate ACKs were successfully transmitted through the network, there is no need to enter slow-start and dramatically reduce the information transfer rate. CWND should therefore stay open.

Several issues arise when TCP slow-start and congestion avoidance are deployed over a satellite link. First is the negative impact of slow-start on performance for transmission of small files. This is rooted in the fact that the rate by which the sender will increase the information transfer rate (CWND) is proportional to the RTT.[5] Indeed, the duration of the slow-start phase can be calculated by the value of RTTlog2W, where W is the receiver's advertised window size. A long RTT (0.5 sec for GEO satellite links) will result in a longer period spent in slow-start, when the TCP sender is not fully utilizing all available link capacity. This adversely impacts the duration for transmitting small files because the transmission usually completes before the CWND can be opened up all the way.

A similar problem occurs in the behavior of congestion avoidance over satellite links. Again, the increase in transfer rate is linearly proportional to the RTT—one segment per RTT. Therefore, it may take quite a few lengthy RTTs to reach an optimal steady-state transfer rate.

The Maximum Throughput Window. The primary objective of the TCP behaviors noted above is for the sender and receiver to make full use of the available capacity on the satellite link or, in other words, to achieve maximum throughput. This in turn is determined by the quotient of the receiver's buffer size (or advertised window—W) and the RTT. Therefore, the maximum throughput for a standard TCP window size of 65 Kbytes and a typical RTT of 500 ms is a little over 1 Mbps. This means that without large windows (greater than 65 Kbytes), a TCP sender can only transmit 1Mbps of data before it must wait for an acknowledgment from the sender. When multiple TCP connections are sharing a satellite link, or the satellite link itself is less than, say, a T-1 (1.55 Mbps), the impact of this may be minimal. However, when a single TCP connection is using a 2 Mbps satellite channel, then the link cannot be fully utilized. Of course, this would also hold true for large bandwidth-delay terrestrial links.

Network Protocol Modifications for Satellite Transmission

Understanding and modifying TCP protocol behavior to account for satellite link delays can go a long way toward improving throughput. However, there are a number of other non-TCP techniques that are used to further enhance satellite link performance and functions. The TCPSAT working group of the IETF has identified and recommends the use of two such techniques: Path MTU Discovery and Forward Error Correction (FEC).

Avoiding IP Fragmentation

Path MTU Discovery (RFC 1191) is used to determine the maximum packet size over a path such that IP fragmentation is avoided. IP fragmentation is the process of segmenting a single packet into several smaller packet fragments. It is performed by a router when the size of a packet received on an inbound link exceeds the maximum allowable packet size on the outbound link. Path MTU Discovery therefore eliminates fragmentation processing on the routers and reassembly processing at the destination host. In addition, the use of larger packets without fragmentation improves the data byte to control byte ratio and therefore improves throughput utilization on the link.

Forward Error Correction

Because satellite links can be noisier than terrestrial links, they often have higher bit-error rates (BER). Unfortunately, TCP reacts to data corruption as if it were network congestion, and reduces its sending rate. Therefore, link-layer protocols used on satellite channels employ FEC techniques to improve the quality of the link. FEC works by transmitting with the data additional control information that can be used to recover or repair corrupted data at the other end of the link. While FEC consumes additional bandwidth, and also requires more processing, it enables satellite channels to approach terrestrial BER rates.

Persistent Connections

Another very useful technique involves the use of the Hypertext Transport Protocol (HTTP) used in Web browsing. HTTP uses TCP as a transport and, until recently, required a separate TCP connection for each object retrieved from a Web page. A satellite link operating in slow-start mode would only exacerbate this very inefficient protocol design. The simple solution is to allow multiple objects to flow over the same TCP connection so that the CWND has an opportunity to open up and fill the channel. This is called persistent connections, and is supported in HTTP 1.1 as defined in RFC 2068.

Unidirectional Link Routing

Asymmetrical channels enable a site to benefit from the high capacity and performance of a unidirectional satellite downlink while using an inexpensive and conventional dial-up (or terrestrial) link as a separate uplink. While asymmetrical paths are quite common on the Internet today, they are formed by routers and hosts exchanging control information over bidirectional links. The UniDirectional Link Routing (UDLR) Working Group of the IETF (http://www.ietf.org/html.charters/udlr-charter.html) is examining various modifications to routing protocols to support network topologies with unidirectional links.

TCP Enhancements

Improving TCP throughput over high-speed networks and satellite links continues to be an active area of research and discussion.[6] Researchers at NASA's Lewis Research Center are working with the ACTS satellites to understand the issues associated with TCP/IP over satellite connections. The IETF TCP over Satellite Working Group was chartered in 1997 to better understand and document the issues related to TCP performance over satellite links. To date, their efforts have served as a repository for discussion on standard TCP mechanisms and some of the outstanding research work in the area. Some of the TCP mechanisms that have been documented by the working group to address TCP throughput over satellite links consist of the following:

Large Windows. RFC 1323 defines a set of window scaling (large windows) options available to TCP implementations that operate over large bandwidth-delay networks such as those containing satellite links.[7] Large windows are required for other large bandwidth-delay networks such as ATM, Gigabit Ethernet, and Packet SONET, so that just about all commercially available TCP implementations now support the large window options.

Delayed ACKs. Instead of generating an ACK for each received segment, a TCP receiver may choose to generate an ACK for every second segment that arrives or, if a second segment does not arrive, wait for a time-out period of up to 500 ms before generating the ACK.[8] The idea here is to reduce the amount of ACK processing in the network. Since the arrival of incoming ACKs at the sender determines how quickly CWND can open, any delay will slow this process and possibly impact performance. Indeed, the first segment sent during slow-start will incur the RTT plus the delayed ACK time-out at the receiver.

While it is suggested that all TCP hosts should use delay ACKs, one can easily see that disabling this feature on a satellite link would be useful. Promising techniques such as using delayed ACKs after the slow-start phase completes have been proposed, but would require a change to the sender and receiver's TCP implementation.[9]

Larger Initial Window. Slow-start uses an initial window size of one. Starting off with a larger initial window size of three or four segments will allow more segments to flow into the network, generating more ACKs, and will decrease the time it takes to complete the slow-start process.[10] Because the CWND can open up faster as a result, better performance is gained, in particular for small files transmitted over links with long RTTs. The effectiveness of the use of larger initial windows has been well documented, and it is expected that future TCP implementations will make use of this feature.

TCP SACK. TCP selective acknowledgments enable a TCP receiver to inform the sender of what specific segments were lost so that the TCP sender can retransmit them.[11] When faced with multiple packet losses in a single window, TCP SACK enables a sender to continue to transmit segments (retransmissions and new segments) without entering into a time-consuming slow-start phase. The use of this TCP acknowledgment function is not widespread; however, this will change with the release of Windows 98 and other OSs that include support for TCP SACK.

Other techniques currently under study to enhance TCP performance involve placing some TCP intelligence in the network to provide faster and more accurate feedback to the TCP endpoints about the nature of the satellite connection. An example would be to place a TCP performance-enhancing proxy (PEP) agent in the ground station router (or device). A PEP agent could, for example, terminate the TCP connection with the sender and originate a different, more satellite-friendly transport protocol session over the satellite link. The PEP agent could provide fast ACK feedback to the sender while hiding the management of the satellite connection. Another technique would be to provide explicit congestion notification (ECN) information to the TCP sender when network congestion appears imminent. This would enable a TCP sender to identify when packet loss may be due to congestion versus corruption, and to adjust the CWND accordingly.

Satellites have become an indispensable part of the global Internet to supplement the evolving ground infrastructure and to deliver new services. The requirement to sustain and enhance maximum link capacity over networks with the unique characteristics of satellite channels will lead to further enhancements to the TCP/IP protocol suite. ■

Resources Online

on the Web

Astrolink • www.astrolink.com/

DirecPC • www.direcpc.com/index2.html

NASA's Lewis Research Center Internet Protocols • ctd.lerc.nasa.gov/5610/inetprotocols.html

NASA's ACT Gigabit Satellite Network • mrpink.lerc.nasa.gov/gsnhome.html

Spaceway • www.hcisat.com/SPACEWAY/SPACEWAY.html

TCP Over Satellite Working Group Charter • www.ietf.org/html.charters/tcpsat-charter.html

TCP SACK Commercial/Shipping Implementations • www.psc.edu/networking/all_sack.html#commercial

Christopher Metz is a consulting systems engineer with Cisco Systems. He focuses on advanced and emerging IP protocols and architectures and can be reached at chmetz@cisco.com.

REFERENCES

1. J. Postel, "Transmission Control Protocol," RFC 793, Sept. 1981; available online at http://info.internet.isi.edu:80/in-notes/rfc/files/rfc793.txt.
2. D. Glover and H. Kruse, "TCP Performance in a Geostationary Satellite Environment," *Annual Rev. of Comm. 1998*, Int'l Eng. Consortium, Apr. 1998.
3. W.R. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, IETF Network Working Group, Jan. 1997; available online at http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2001.txt.
4. V. Jacobson, "Congestion Avoidance and Control," *Computer Comm. Rev.*, Vol. 18, No. 4, Aug. 1988, pp. 314-329; also available online at ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z.
5. M. Allman et al., "TCP Performance over Satellite Links," *Proc. Fifth Int'l Conf. Telecomm. Systems*, 1997.
6. C. Partridge and T. Shepard, "TCP/IP Performance over Satellite Links," *IEEE Network*, Vol. 11, No. 5, Sept./Oct. 1997, pp. 44-49.
7. V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High Performance," RFC 1323, IETF Network Working Group, May 1992; available online at http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1323.txt.
8. R. Braden, ed., "Requirements for Internet Hosts," RFC 1122, IETF Network Working Group, Oct. 1989; available online at http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1122.txt.
9. M. Allman, "On the Generation and Use of TCP Acknowledgements," *Computer Comm. Rev.*, Oct. 1998.
10. M. Allman, S. Floyd, and C. Partridge, "Increasing TCP's Initial Window," RFC 2414, IETF Network Working Group, Sept. 1998; available online at http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2414.txt.
11. M. Mathis et al., "TCP Selective Acknowledgment Options," RFC 2018, IETF Network Working Group, Oct. 1996; available online at http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2018.txt.