

HTTP Traffic over Satellite

[<draft-pohlmann-http-traffic-satellite-00.txt>](#)

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

To view the entire list of current Internet-Drafts, please check the "lid-abstracts.txt" listing in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

Internet use over satellites (LEO, MEO and GEO) has not received enough attention. Satellites have an inherent broadcasting ability useful for transmitting Internet traffic. They can send information to many locations simultaneously depending on the coverage beam of the satellite. The issue of Internet use over satellites needs to be addressed so that future implementations of networking protocols can be used with this medium. The vast majority of Internet traffic is TCP traffic resulting from the HTTP protocol so this issue will be addressed here. It is hoped that this facilitates the transition from the use of terrestrial networks, to networks incorporating satellite links.

Acknowledgements

The TCP algorithms that are mentioned were developed by Van Jacobsen. An excellent explanation of their use is described in [1].

1. The Issues

Several issues arise when using satellites (LEO, MEO and GEO) for information transmission. The delay, or latency, of a satellite link (sender to receiver) is greater than a terrestrial communications link. This varies depending on the type of satellite. LEO satellites typically have a latency on the order of 100 milliseconds and GEO satellites have a latency of approximately 250 milliseconds. The data traveling over a satellite will also travel over terrestrial links in addition to the satellite link. Consequently the latency from sender to receiver can grow to as much as 400 msec when using a GEO satellite. These latency values raise questions. What effect will the latency have on HTTP traffic, and am I using the communications link effectively?

The other issue in a satellite link is that errors may be bursty rather than random and so lost packets may be an indication of discarding due to errors rather than congestion. Although there are powerful error-correcting codes used on a satellite link, the distribution of errors in packets that have traversed a satellite link are not as uniformly distributed as errors seen on terrestrial connections such as fiber-optic cable. This can cause problems for networks using technology where the error correction mechanisms are not meant to deal with multiple errored bits in a cell. ATM is one such technology. In this case it may be helpful to reduce the bit error rate of the link in order to reduce the probability of grouping errored bits.

2. Consequences for HTTP Traffic

TCP's receive buffer or receive window is very important for good performance of TCP. A rule of thumb is that the optimum receive buffer size is the bandwidth-delay product of the link. The Internet user would like to make efficient use of bandwidth by using a TCP receive buffer approximately equal to the bandwidth-delay product. Assuming an individual is downloading a web page using a 56.6 kbps modem, the bandwidth they would use is around 7 kbytes/sec. Assuming an end to end delay of 1.0 second, an optimum receive window would be 7 kbytes. Since most PC's use receive windows of 8 kbytes or slightly more, they make efficient use of the connection. Doubling the speed of the connection means that a window size of 14 kbytes would be needed. This would mean the user cannot increase their window size enough to make efficient use of the TCP connection.

[RFC1323](#) [2], specifies "large windows", which are windows greater than TCP's maximum theoretical window of 64 kbytes

and practical window of 32 kbytes. Large windows increase the theoretical maximum window size of TCP to 2^{32} bytes. Unfortunately, the adoption and use of TCP with [RFC1323](#) support is not widespread even though its importance is increasing. Newer versions of HTTP state that HTTP will use a single TCP connection rather than multiple connections as in HTTP/1.0. With all of the HTTP data flowing through a single connection, it is critical for TCP to efficiently use the channel bandwidth. TCP is making efficient use of the bandwidth when it is in congestion avoidance mode. Congestion avoidance mode is where the importance of the receive window is greatest since TCP is trying to have many segments "in route" between the sender and receiver. A smaller than optimum window means that all of the available bandwidth is not being used. Recent work [3] shows that the penalty suffered for using "smaller than optimum" window sizes increases with increasing file size and increasing latency. Consequently, the importance of large windows is increasing.

When TCP is in slow-start mode it is not making efficient use of the channel's bandwidth. TCP is initially in slow start mode and falls back into slow-start if no data is sent in a period approximately equal to the round trip time of a segment. HTTP/1.1, which sends data through a single TCP connection rather than multiple TCP connections, will reduce the chances of TCP falling back into slow-start. When slow start begins, TCP starts the sender's congestion window at one segment. This lets TCP determine the bandwidth that is available to it. One technique that is being investigated for using TCP over satellites is to start the congestion window at a value greater than one segment. This would mean that TCP starts out using a larger percentage of the available bandwidth. Since the available bandwidth will usually allow sending many segments per window, it is overly conservative for TCP to start out sending a single segment. Other techniques are implemented in the routers in a network. One such protocol is called RED [4] and it will help in congested networks. RED will allow TCP to prepare itself for increasing congestion in a controlled manner. By discarding a few TCP segments as congestion begins, TCP will stop increasing its congestion window. This will prevent the loss of many segments that could force TCP to severely reduce its congestion window. By preventing a large reduction in the congestion window, the available bandwidth can be better used.

3. References

- [1] W. R. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms" [RFC 2001](#), Jan 1997.
- [2] V. Jacobson, R. Braden, and D. Borman, "TCP

Extensions for High Performance" [RFC 1323](#), May 1992.

[3] Y Zhang, D DeLucia, B Ryu, S Dao, "Satellite Communications in the Global Internet: Issues, Pitfalls, and Potential",
<http://www.isoc.org/isoc/whatis/conferences/inet/97/proceedings/F5/F5-1.HTM>

[4] S. Floyd and V. Jacobson. "Random early detection gateways for congestion avoidance", IEEE/ACM Transactions on Networking, 1(4):397-413, Aug 1993

Author's Address:

Robert Pohlmann
1761 Business Center Drive
Reston, VA 20190

Phone: (703) 438-7805

Email: pohlmann@sed.stel.com