NettGain Technology -White Paper

Introduction

The use of satellite links for a corporate and ISP network infrastructure has many advantages. However, as applications shift from the traditional transaction exchange to Internet/Intranet access, TCP and its performance over satellite links come into scrutiny. Through experience (see [1,2,3,4,5]), it is apparent that TCP over satellite links is very limited in terms of potential data transfer speed, and is relatively expensive in resource consumption. Flash Networks has developed a proprietary protocol that addresses and overcomes these obstacles. BST, Boosted Session Protocol, is the solution to the limits of TCP. NettGain technology replaces the TCP protocol with the fully optimal BST Protocol. Users can now experience the advantage of optimized bandwidth conditions that increase five fold from the usual TCP connection.

BST addresses two main issues for Satellite service providers:

- **Speed -** Terrestrial alternatives raise user expectations and pose challenges to Satellite systems. ***True*** broadband is difficult to achieve over satellite, and most operators are under constant pressure to improve performance.
- **Capacity -** Terrestrial bandwidth costs are decreasing. Satellite spectrum is a costly and scarce resource. Reasonable ROI requires operators to overload subscribers on the network or to invest in optimization systems.

This paper will discuss the roots of the problem, and describe how TCP over satellite links can be made more efficient using NettGain technology.

TCP Over Satellite – What are the issues?

The underlining factor for the problematic use of TCP over a satellite link is that TCP was not designed for the satellite environment. TCP was specifically designed as a flexible protocol that can operate on various links such as dial-up to Gigabit Ethernet LANs, however it is not optimal on any of these links. Satellite links, in particular, prove to be a medium where TCP provides a marginal service. Experience with TCP usage over satellite links proves that the performance and resource consumption of TCP over a satellite link is unacceptable [1, 2, 3, 4, 5].

In this paper, we will address only the most prominent causes for TCP's inefficiencies over satellite connections: [1] We will separately examine issues limiting Throughput (i.e. Transfer speed) and Resource consumption.

[1] A more detailed description can be found in the form RCC 2488 from the Internet Engineering Task Force (IETF) widely available on the Internet, e.g. at www.ietf.org .

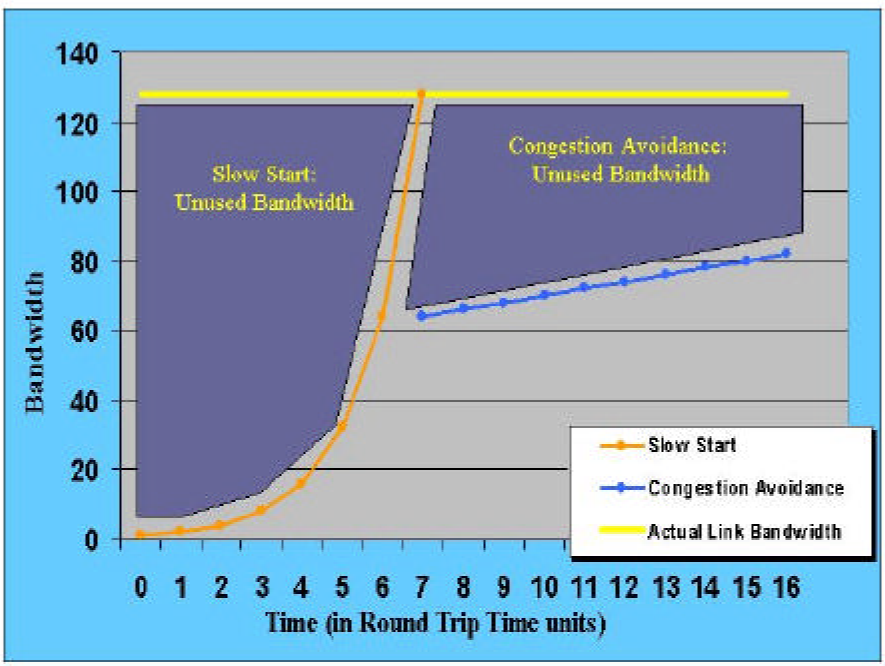
Speed Limiting Effects

Slow Start

TCP performs a detection process called "Slow Start" whenever a connection is made because TCP is unaware of the link bandwidth. TCP calculates the lowest possible bandwidth and increases exponentially until a packet loss is detected. In the satellite environment, the long round trip time implies a long period (depends on available bandwidth) during which the link is under-utilized and perceived by the user as slow. This is true for every new TCP connection in the satellite system. Seven to ten RTTs are typically required before switching from Slow Start to Congestion avoidance mode, and then an additional - RTTs may be required to get back to near link-capacity rate again. This process leaves a significant amount of bandwidth simply unused. The figure below illustrates this point for a 128 Kbps satellite link that after 16 RTTs, the sending rate is still only about 80 Kbps.

Congestion Avoidance

TCP assumes all packet loss[2] is caused by congestion. Hence, there is a reduction in the current transmission rate by half. Also, the "slow start" algorithm is changed into a much slower acceleration. If the packet loss occurred due to link noise, or other transient conditions, TCP will still react in this manner. The result is unused link capacity which means a slower data transfer.



TCP connection bandwidth (Kbps) utilization

[2] Most of today's satellite system are very reliable, however satellite link also suffer a 1%-2% packet loss For more information, you can refer to: http://www.internettrafficreport.com/[3] The satellite RTT is 540ms and the terrestrial side is about 150ms, thus the total RTT for the TCP connection is 700ms.

Receive Window

An application that sends data using TCP is only allowed to send as many bytes that are explicitly allowed in the last packet seen from the intended destination. (This is to make sure the destination has enough space to store the bytes about to be sent.) This value is known as the "Advertised Receive Window." Typically, this is set to about 8 Kbytes, which severely limits the top speeds that a TCP connection can transfer data over to a satellite. The actual bandwidth can be calculated according to the RTT and the window size using the following formula:

Max BPS = RWIN / RTT

For example, consider a 4 Mbps satellite link with a 500ms delay and 8Kbytes window size -- only 128 Kbps will be sent on a 4 Mbps link! Even if the "Advertised Receive Window" is set to the theoretical maximum of 64 Kbytes, it will still limit TCP's speed for satellite links to less than 1 Mbps.

Path Asymmetry

Satellite networks are frequently set up in an asymmetric manner, either in terms of allocated bandwidth, or even using a different technology for each direction (i.e. a hybrid satellite/dial-up network). This is especially inherent in Internet/Intranet environments, where traffic patterns typically exhibit intrinsic asymmetry.
Unfortunately, asymmetry has been shown to be a problem for TCP. TCP's sending rate is constrained by the rate at which ACKs arrive from the receiver. A ratio of 47:1 is about the maximum amount of asymmetry that will allow full speed transfers. Given a dial-up link of 33.6 Kbps it will exceed the limit if more is needed than about a T1 rate in the forward direction.

Too Many Concurrent Connections

A significant finding for Internet/Intranet via satellite for a large user population (e.g. a large Intranet network, or an ISP) is presented in "TCP Behavior with Many Flows" [8]. Briefly stated, when there are more concurrent connections in the pipe than the bandwidth-delay product, TCP forces a packet loss rate approaching 50% while the link utilization stays high and response time to the users suffers large variations. Packet loss rates are a source of both throughput degradation and resource waste. Adding buffering can delay the effect however a complete solution requires the use of a protocol conversion system as described in the *NettGain - Satellite Link Optimization Solution* section below.

3-way Handshake

The TCP three-way handshake requires three RTTs before there is an initiation of sending data. This is about 2 seconds for each connection on a satellite system.

Resource consumption effects

Thus far, we have presented various throughput-limiting issues. Resource consumption issues merit a separate discussion. TCP is not designed to optimize the amount of traffic over the link and some of the protocol features lead to unnecessary bandwidth consumption.

Variable Response Time

TCP is heavily dependent on measuring the round trip time between the sender and the receiver. For example, on multiple-access satellite networks such as TDMA or DAMA based access methods; the time needed to set up the link or solve channel contention can cause TCP to perceive a variable round trip time. This causes TCP to assume lost packets when, in fact, they were correctly received. These packets are then re-transmitted needlessly. In one TDMA network experimental trial, as much as 13% of the traffic sent was redundant and therefore unnecessary.

TCP Acknowledgment Content and Frequency

A significant factor for resource waste is the sheer volume of ACK packets generated by TCP. TCP standards state that instead of sending ACK for each packet received, it is possible to send it on constant time slots. This reduces the amount of the unnecessary ACKs however it does not totally eliminate it.

HTTP over Satellite: Performance Issues

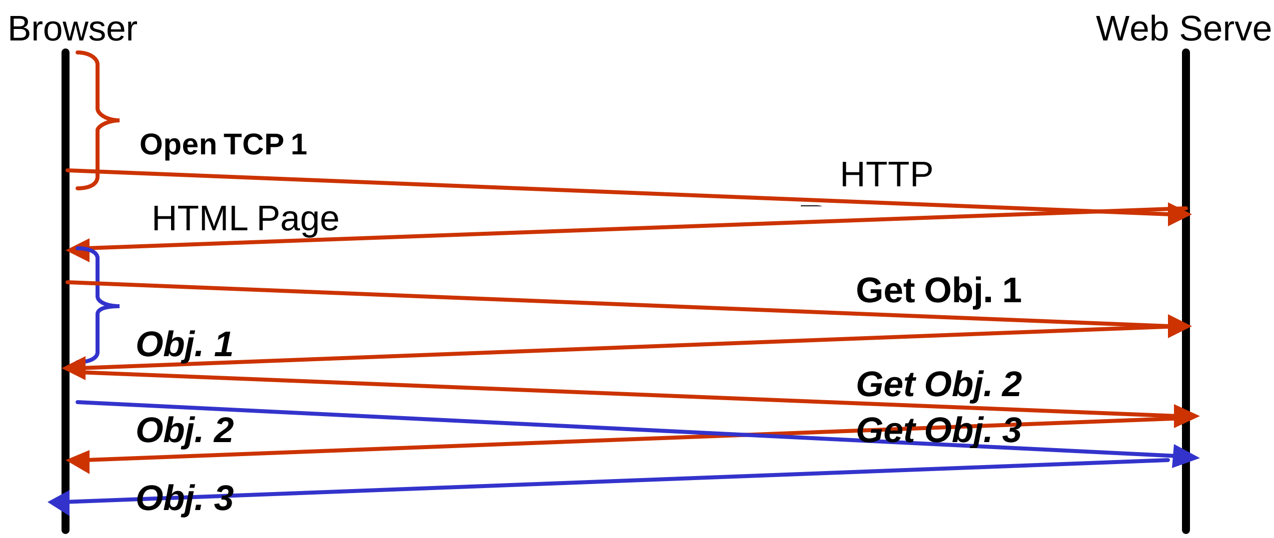
One of the major protocols that need to be considered when planning Internet/Intranet via satellite is HTTP, which is the basis for browser based applications. As it turns out, HTTP has several side effects that have implications for network designers. HTTP/1.0 uses a separate TCP connection for each object to be retrieved. This causes throughput degradation mainly due to performing slow-start at frequent intervals and typically terminating the connection long before "cruising speed" is reached. The 5 packets exchanged in connection setup and termination are essentially overhead. The more connections that are used means that more bandwidth is wasted. On DAMA based satellite networks, careful design is needed to avoid setting up a circuit and tearing it down repeatedly.

Persistent HTTP (P-HTTP) and HTTP/1.1 allow TCP connections to exist for longer periods, and to be used for multiple object retrievals. This cuts down on the connection setup and discard overhead, but on DAMA based satellite networks this can mean circuits are idle for a significant percentage of the time. Finally, HTTP (and sometimes the user applications) can force TCP to send segments that are smaller than the maximum allowed size thereby causing various inefficiencies.

The following image presents a typical (simplified) HTTP session course. The session begins by establishing a TCP connection from the browser to the web server. Following, a request for the HTML page is sent to the web server, which in turn send the HTML response. The browser parses the HTML, and **sequentially** sends requests for all required objects. A new request is sent only after the previous object was fully received. HTTP1.1 enables up to 4 parallel TCP connections, yet most browsers use only 2 TCP connections.

The overall page download time is therefore constituted from the dialog time (the left element) and the transport time.

$$Time \cong RTT \cdot \frac{\#Objects}{\#TCP} + \frac{PageSize}{Bandwidth}$$

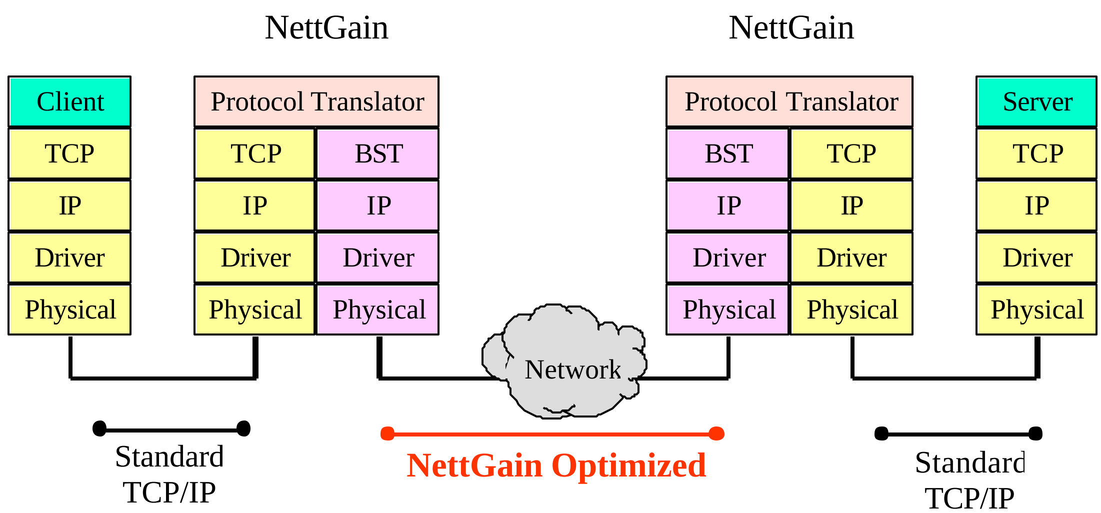


NettGain™ - Satellite Link Optimization Solution

The NettGain products from Flash Networks are protocol gateways for a satellite link. NettGain technology incorporates its full protocol conversion system allowing a TCP connection to utilize the full link capacity, thus speeding up transfer speed by 500%. The heart of the system is Flash Networks proprietary enhanced protocol, BST. The BST protocol provides users with an effective resource that allows enhanced browsing capability with optimized bandwidth conditions.

BST

The Boosted Session Transport (BST) protocol is a TCP-like reliable protocol that replaces TCP over the path where performance optimization is needed. TCP packets from a source location are transformed from TCP to BST and sent to the destination. There they are transformed back into TCP and sent to the originally intended target. NettGain manages both BST and TCP flow control and data queues to allow a full protocol conversion. Protocol conversion is illustrated in the following figure.



BST Protocol use by NettGain

The BST protocol offers numerous enhancements over TCP:

- **Pre-Tuned Parameters:** BST consistently operates at a high data rate in order to maximize the use of bandwidth resources. It can be optimized to the specific link in use to take into account bandwidth, delay, bit error ratio, and Maximum Transmission Unit (MTU) size.

- **Immediate Transmission at Full Speed**: Unlike TCP, the BST protocol does not need to perform Slow Start in order to discover available bandwidth. Transmission begins at the full rated link speed from the first byte.

- **Smart Congestion Avoidance**: Unlike TCP, which slows data transfer dramatically after a lost packet is detected, BST always assumes lost packets are due to BER and not congestion. Thus, it will always recover from loss packets using immediate and selective retransmission without reducing speed or bandwidth utilization. As a result, BST can fill any satellite "pipe" regardless of latency or BER.

- **Intelligent Bandwidth Allocation:** Globally manages all the available bandwidth to ensure efficient and fair usage.

- **Smart Rate Control:** BST guards against buffer overflows to prevent packet loss. Implements rate control for both BST and TCP.

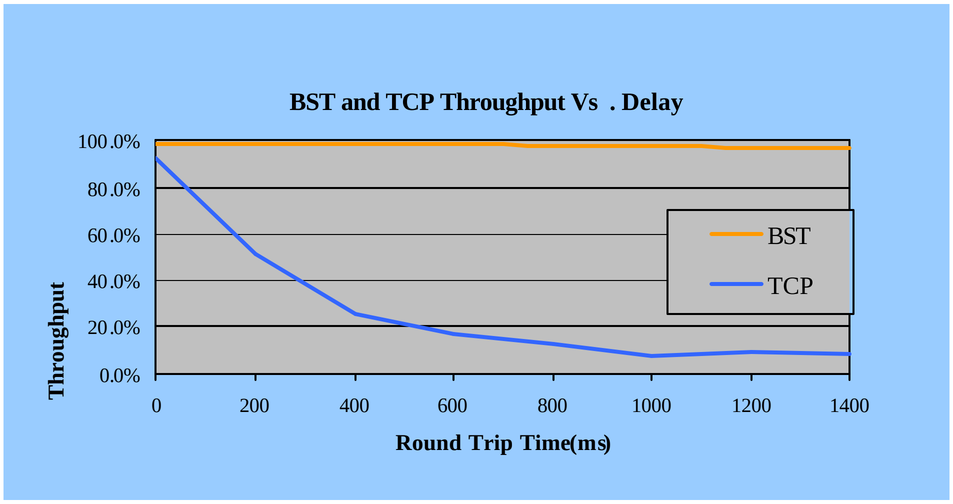
- **Intelligent Acknowledgements:** BST uses selective acknowledgements and negative acknowledgements (ACKs and NACKs) to convey packet transmission results to the sender. The sender can then selectively retransmit lost packets. This results in faster, more efficient loss detection/recovery and significantly less (up to 75%) overhead than with TCP. This is very useful in an asymmetrical satellite system where the uplink bandwidth is significantly low.

- **Unlimited Window Size:** NettGain internally buffers data allowing TCP to use the full link bandwidth by removing the window size limitation by advertising a window size that is not related to the high RTT on the satellite link.

- **No 3-way Handshake**: The BST protocol keeps a tunnel open between the Client and the Server, eliminating the need for the standard TCP 3-way handshake and cutting 1 second off of any transaction. (This is a software-controlled feature that can be turned off if needed.)

- **Packet Loss Recovery**: BST recovers from lost packets using a resend algorithm. Its smart queue management and disorder handling allows it to minimize the number of packets retransmitted over the satellite link.

The following shows a throughput comparison test for traffic transferred over a satellite link between TCP and a NettGain enhanced case.



NettGain throughput compared with TCP throughput for various delays

Additional Features to NettGain Technology

Compression

A compression system directly reduces the amount of traffic that needs to be transported, saves bandwidth, speeds up response time and shortens transfer times. NettGain offers lossless compression based on the zlib algorithm (same algorithm used in the popular internet tool gzip) and lossy compression to improve compression ratio on images. Compression gains depend on the content that is transferred. Average overall compression ratios are about 25%-30% for Web traffic.

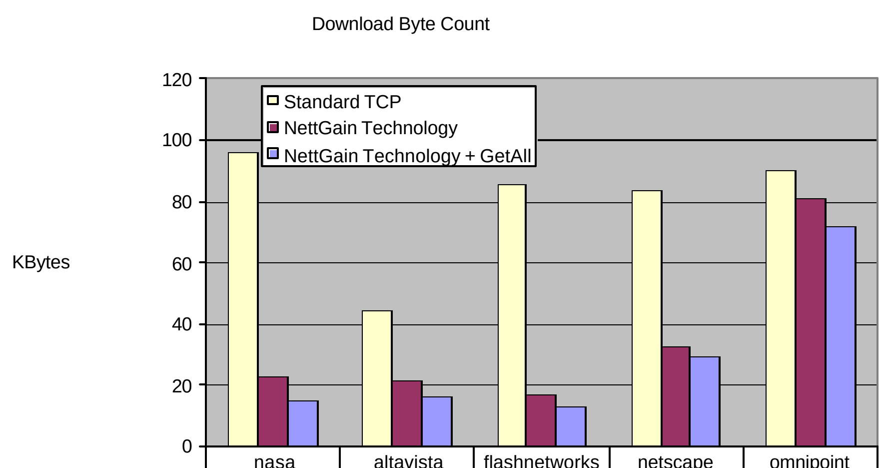
HTTP optimization

NettGain introduces the patented GetAll Protocol. GetAll minimizes the dialog time to zero, passing the browsing bottleneck from the given propagation delay to the page size and bandwidth utilization. These can be dealt with by content manipulations and BST. For each page retrievable with NettGain the browser starts with the unavoidable RTT for the first object. After that the user will receive constant data stream at high rate much like FTP transfer of the object. This results in faster page retrivale and real broadband experience.
The following table lists a few examples of test done on satellite link.

URL	Download time (sec)	
	NettGain	TCP
http://www.cnn.com/WORLD/	7.4	17.4
http://www.goldmansachs.com/	6.3	14.9
http://www.fool.com/	6.4	16.3
http://cnnfn.cnn.com/	5.3	13.1
http://www.amazon.com/	6.6	17.4

Saving Bandwidth

The following diagrams visualize the bandwidth savings achieved with NettGain technology. NettGain realizes savings through the reduction of TCP overhead required by Slow Start , congestion avoidance and HTTP optimization. Note that the savings in download bytes varied widely from site to site. The savings is based on the nature of the site. The NASA (http://www.nasa.gov) site is mostly text and therefore can be compressed very efficiently. The OmniPoint site (http://www.omnipoint.com) is mostly images that are already compressed (GIF or JPEG) and therefore less savings is realized. These tests did not include the Lossy compression feature.



	nasa	altavista	flashnetworks	netscape	omnipoint
Standard TCP	96.2	44.5	85.5	83.6	90.3
NettGain Technology	23	21.4	16.8	32.4	81.3
NettGain Technology + GetAll	15	16.1	13.1	29.4	71.6

Download byte counts for Internet sites using Standard TCP, NettGain Technology, and NettGain Technology with the HTTP optimizations

Note that these results **DO NOT** include the effects of compression that is available as an option.

About Flash Networks

With the increasing global reach of the Internet, satellite networks are growing in popularity as a means of delivering high-speed data connectivity. Typical uses are branch office access to IP connectivity and high speed Internet access to the home.

However, the inherent characteristics of satellite communications - long transmission delays, asymmetric bandwidth, shared channels, and weather effects - combined with the limitations of TCP and applications that were not designed for these conditionscan cause significant network performance degradation. In fact, some applications may not work at all over a satellite link.

Flash Networks' NettGain technology overcomes these limitations and allows satellite users to achieve significant performance improvement. Flash Networks is recognized as the leader in this market, having relationships with the largest of satellite service providers and enterprise satellite users.

NettGain makes a satellite lnk feel like a fast terrestrial link, virtually eliminating the impact of satellite latency on application performance. Users enjoy accelerated data transfer, full bandwidth utilization, and improved response times.

NettGain Products

NettGain technology is implemented in two industry-leading products. The NettGain 1000 is a proxy based software solution with a redundancy and load balancing mechanism. It is designed to handle up to 10Mbps of traffic. The NettGain 1000 is available for common industry operating systems.

The NettGain 2000 is a hardware-based solution fully transparent and symmetrical that can handle up to 45Mbps on the Server side and up to 10Mbps on the client side. The NettGain 2000 implements a revolutionary redundancy and load balancing algorithm with hot fail over detection and recovery, allowing one server to act as a redundant unit for a full cluster.

For more information regarding NettGain products please refer to the NettGain 1000 and the NettGain 2000 product specification documents.

Conclusion

As applications and traffic over satellite networks moves closer to an Internet/Intranet model, TCP and its behavior in satellite-based networks is becoming increasingly critical. TCP can be made effective and efficient over satellite networks as demonstrated by using various means ranging from tune-ups to protocol conversion. The use of NettGain technology allows users to attain higher standards of optimal performance using a satellite link.

Contact Information:

Flash Networks
2137 Route 35 North
Holmdel, NJ 07733

Sales Email: sales@flashnetworks.com

Tel: +1-732-203-4060
Fax: +1-732-203-4069

References

[1] Enhancing TCP Over satellite Channels using Standard Mechanisms, Request For comments, RFC 2488 M. Allman, D. Glover and L. Sanchez, Internet Engineering Task Force (IETF), 1999
http://www.ietf.org/rfc/rfc2488.txt
[2] HTTP Page Transfer Rates Over Geo-Stationary Satellite Links. Hans Kruse, Mark Allman, Jim Griner, Diepchi Tran.
Proceedings of the Sixth International Conference on Telecommunication Systems, March 1998. http://roland.grc.nasa.gov/~mallman/papers/nash98.ps
[3] HTTP Experiments in Satellite-Based Networks.
M. Allman, Presented at the Telecommunications Industry Association Meeting. March 1999.
http://roland.grc.nasa.gov/~mallman/papers/tia-3.99.ps
[4] TCP Performance Over Satellite Links.Mark Allman, Chris Hayes, Hans Kruse, Shawn Ostermann.
Proceedings of the Fifth International Conference on Telecommunications Systems, Nashville, TN, March, 1997.
http://roland.grc.nasa.gov/~mallman/papers/nash97.ps.
[5] The effects of asymmetry on TCP performance
Hari Balakrishnan, Venkata N.Padmanabhan, and Randy H.Katz.
Proceedings of the third annual ACM/IEEE international conference on Mobile computing and networking , 1997, Pages 77 – 89
http://www.acm.org/pubs/citations/proceedings/comm/262116/p77-balakrishnan/
[6] The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm.
M. Mathis, J. Semke, J. Mahdavi, T. Ott, Computer Communication Review, volume 27, number 3, pp. 67-82, July 1997.
http://www.psc.edu/networking/papers/model_ccr97.ps
[7] Modeling TCP throughput: a simple model and its empirical validation.
Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose.
Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication , 1998, Pages 303 – 314
http://www.acm.org/pubs/citations/proceedings/comm/285237/p303-padhye/
[8] TCP Behavior with Many Flows, Morris, R.
IEEE International Conference on Network Protocols, October 1997, Atlanta, Georgia.
http://www.eecs.harvard.edu/networking/papers/icnp97-web.ps