FACULTAD DE INFORMÁTICA
UNIVERSIDAD POLITÉCNICA DE MADRID

UNIVERSIDAD POLITECNICA DE MADRID
FACULTAD DE INFORMÁTICA

TRABAJO FIN DE CARRERA

Study and Performance Evaluation of TCP versions over GEO and LEO Satellite Links between Performance Enhancement Proxies

Madrid, abril de 2008

AUTOR: Ángel Salamanca Cardona
TUTOR: Marinela García Fernández

Estudio comparativo de diferentes versiones TCP en redes de satélites GEO y LEO entre proxys con mejora de prestaciones.

Resumen: El Protocolo de Control de Transmisión (TCP) es uno de los protocolos fundamentales en Internet. Sin embargo, este protocolo y sus mecanismos de control de la congestión fueron originalmente desarrollados para medios guiados y su aplicación en medios no guiados como son las redes de satélites degrada notablemente su rendimiento. Pese a la creciente demanda de los sistemas inalámbricos que cada vez disponen de más velocidad de transmisión y reducen los costes en la instalación frente a un sistema cableado, la tasa de error sigue siendo muy elevada en comparación con un entorno cableado.

El objetivo de este proyecto es hacer un estudio detallado sobre el proxy Snoop y comparar el rendimiento de diferentes versiones TCP en redes de satélites LEO y GEO. Se ha trabajado con el simulador de redes Ns-2 (*Network Simulator*) para la simulación de los escenarios de nuestras pruebas. Para el caso de redes satelitales LEO se implementó una constelación de órbitas polares, mientras que en el caso de GEO se implementó una red VSAT (*Very Small Aperture Terminals*) en estrella. En ambos escenarios se han utilizado los satélites en modo 'bent-pipe'.

El proxy Snoop es uno de los representantes de un grupo de proxys para la mejora de prestaciones en entornos inalámbricos denominados PEPs (*Performance Enhancement Proxies*). En nuestros escenarios de simulación utilizamos el proxy como separador de medios guiados y no guiados, obteniendo notables mejoras en la eficiencia y funcionamiento del TCP.

Existen numerosas versiones de algoritmos TCP con el objetivo común de alcanzar el máximo rendimiento. En este proyecto se han comparado cinco versiones: New Reno, Hybla, Vegas, DVegas y Westwood+ aplicadas a enlaces satelitales GEO y LEO. Según nuestros resultados TCP Hybla ha resultado ser el más eficiente, mejorando los enlaces satelitales mediante la separación de los errores de corrupción de datos de aquellos debidos a la congestión de la red.

Dedicatorias

A mis padres, a mis abuelos, a los compañeros
de facultad, a las estrellas de Beihang y en especial
a mi novia Nebaí por todo este tiempo en el
que me han estado apoyando.

Agradecimientos

A mis dos tutores:
Marinela García Fernández (UPM – Facultad de Informática) y
Peng Fei (Beijing University of Aeronautics & Astronautics, BUAA),
por su dedicación y disposición para la realización de este Trabajo de Fin de Carrera.

A la UPM por su programa de movilidad Hispano Chino,
a Ángel Álvarez por animarme a iniciar esta aventura y
a David del Val por la financiación de esta beca.
.

ABSTRACT

Abstract: The Transmission Control Protocol (TCP) is the mostly used protocol for data transmission over networks. However, TCP's congestion control algorithm wasn't created to match the special characteristics of satellite links, resulting in a drastically performance degradation. The employment of TCP split connections coupled with link level retransmissions and satellite link tailored TCP versions have been proven effective in hiding corruption losses from TCP, improving end-to-end performance. TCP appears in different algorithm versions, all with different features but with maximal throughput as their main objective. The aim of this paper is to evaluate and compare five TCP versions, which are New Reno, Hybla, Vegas, DVegas and Westwood+, when applied over GEO and LEO satellite links in split TCP connections, using Ns-2 simulation. Suitable scenarios based on a GEO bent-pipe satellite and a polar orbiting LEO constellation, have been implemented. We evaluate the performance of the proposed TCP versions over LEO- and VSAT-based satellite links and compare it with the performance enhancements offered by the Snoop proxy and the Explicit Control Protocol (XCP). Simulation results show that TCP Hybla outperforms the rest of TCP versions, improving the usage of satellite links that are affected by losses not due to congestion. The use of the Snoop proxy is proven to improve end-to-end performance of the satellite link.

Keywords: Transmission Control Protocol (TCP), Satellite, Performance Enhance Proxy (PEP), Very Small Aperture Terminals (VSAT), Snoop, TCP Reno, TCP Sack, TCP New Reno, TCP Hybla, TCP Vegas, TCP DVegas, TCP Westwood+ , Explicit Control Protocol (XCP)

PREFACE

Preface: This thesis is the final stage of the Engineer's degree in Computer Science at The Technical University of Madrid (Universidad Politécnica de Madrid, UPM), Spain. The project is given by the Computer Science School at Beihang University, (Beijing University of Aeronautics & Astronautics), and the work was carried out between September 2007 and March 2008 in Beijing, China.

I want to give a special thank to my supervisor, Prof. Peng Fei fir his first-class guidance and valuable counseling throughout the project period. I will also thank my main supervisors in Spain, Prof. Marinela García and Prof. Ángel Álvarez for starting the cooperation and making this project possible.

Beijing, March 2008
Angel Salamanca

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SECTION I
INTRODUCTION

1. INTRODUCTION

1.1 MOTIVATION

In the next generation Internet, satellite links will play an important role by providing broadband Internet access and high-speed backbone network connectivity between remote networks through easily scalable architecture and multicast capabilities [1]. Besides, they are the only way Internet can reach desolated locations, covering large areas of the Earth.

Satellite networking began with the use of individual satellites in Geostationary Orbit (GEO). However, requirements for lower propagation delays and propagation loss, in conjunction with the coverage of high latitude regions for personal communication services, have pushed the development of new satellite communication systems called Low Earth Orbit (LEO) satellite systems. Differences between LEO and GEO satellites will be discussed later in the paper.

The Transmission Control Protocol (TCP) is the mostly used protocol for data transmissions over networks. However, TCP's congestion control algorithm wasn't created to match the special characteristics of satellite links, resulting in a drastically performance degradation. TCP performs poorly in heterogeneous networks with satellite links because of their characteristics: long propagation delays, large bandwidth-delay products, high bit error rates and bandwidth asymmetry.

Long propagation delays affect TCP's slow start and RTO mechanism. As TCP needs to receive an ACK of every sent segment to increase its *CWND*, the long propagation delays over the satellite unable slow start to reach the maximum achievable throughput. Secondly, large bandwidth delay product affects TCP's window size. The offered window size field in the TCP header is only 16-bit long [2], which restricts its value to 64K. The third problem is that the high bit error rates affect the congestion control protocols. TCP misinterprets packet losses as an indication of congestion and reduces the transmission rate, thus leading to TCP throughput degradation. Finally, the bandwidth asymmetry problem results in traffic burstiness, throughput degradation and data transmission rate reduction. The low bandwidth uplink path may become easily congested leading to delay of ACKs, which causes a TCP sender to transmit data segments in bursts.

1.2 SURVEY OF PROPOSED SOLUTIONS

In light of the above problems, many researchers have proposed solutions to improve TCP performance over satellite networks. The possible solutions can be classified into three categories: *link layer solutions, end-to-end solutions and performance enhancement proxy (PEP) solutions.*

1.2.1 Link Layer solutions

Link layer solutions include link layer retransmissions and forward error correction [3], to mitigate packet losses over satellite links due to transmission errors. However, these methods cannot solve the problems experienced by TCP over satellite links caused by latency.

1.2.2 End-to-end solutions

Concerning end-to-end solutions TCP Reno and TCP New Reno [4] are the most commonly implemented algorithms. SACK support is very common and is an extension to Reno/New Reno. The others are competing proposals which still need evaluation. The following protocols were designed specially to enhance performance in the satellite environment.

TCP Fast Start [5] employs cached values of the most recent past TCP connection state variables (*CWND*, *SSTHRESH*, *SRTT*, and *RTTVAR*) for the new connections starting after an idle period. Its drawback is that it requires sender-side modifications and packet prioritization mechanism at intermediate routers to drop low priority segments when congestion occurs.

The IPv6 datagram allows additional fields to be included in extension headers for specific purposes. The extension headers are placed before the encapsulated TCP payload. Bandwidth Aware TCP (BA-TCP) [6] is an end-to-end solution that employs the IPv6 extension headers with fields for round-trip propagation delay (RTPD) and available bandwidth (ABW) to relay explicit network conditions to a TCP receiver. However, this proposal requires IPv6 hosts to be present in the network.

Sharing TCP (STCP) [7] was proposed to mitigate the effect of long propagation delays by sharing TCP state information (*SSTRESH* and *CWND*) among sequential and concurrent TCP connections. The disadvantage of STCP is that it requires an additional data structure in order to store the shared information.

TCP Peach [8] and its variant TCP Peach+ [9] introduce the sudden start and rapid recovery and the jump start and quick recovery algorithms based on the use of low priority segments (dummy and nil segments, respectively) to probe the network for available bandwidth. The *CWND* is set based on the estimated available bandwidth. TCP Peach and TCP Peach+ require packet prioritization mechanism at every intermediate router along the data transmission path.

TCP Westwood (TCPW) [10] is an end-to-end sender side modification of TCP congestion control algorithms for estimating the available bandwidth in the computation of the *CWND*. The value of the *CWND* during congestion avoidance and after a packet loss is computed using the bandwidth estimate (BWE). Adaptive start (Astart) [11] is a satellite network modification to the slow start algorithm for adaptively resetting the *SSTHRESH* based on the BWE of TCPW. Astart prevents premature ending of the slow start phase and enables the *CWND* to grow rapidly without incurring the risk of buffer overflow and multiple losses. It assumes that the *RWND* is always large so that the sending rate depends only on the *CWND*.

TCP Westwood+ [12] is an evolution of TCP Westwood. Its main novelty is the algorithm used to estimate the available bandwidth end-to-end. This improves TCPW's original estimation algorithm, which didn't work well in the presence of reverse traffic due to ACK compression (aliasing effects).

TCP Bulk Repeat [13] improves TCPW performance in the presence of heavy losses due to link errors.

High Speed TCP [14] and Scalable TCP [15] are variants proposed for large BDP networks such as Gigabit Ethernet WANs. High Speed TCP adaptively increases or decreases the *CWND* as a function of the current *CWND* when an ACK is received or when a segment is lost, respectively. Scalable TCP adjusts the *CWND* by a factor $\alpha = 0.01$ upon receipt of an ACK and by a factor $\beta = 0.125$ when segment loss is detected. Both High Speed TCP and Scalable TCP require sender-side modifications. However, the parameters used in adjusting the *CWND* are not optimized for GEO satellite networks [16].

TCP-Swift [17] replaces the slow start and fast recovery algorithms with speedy start and speedy recovery algorithms. The speedy start algorithm enables the *CWND* to open rapidly within two RTTs while the speedy recovery algorithm sends outstanding segments instead of dummy segments (used in TCP-Peach) to probe the network for available bandwidth.

TCP priority-based congestion control strategy (TCP PBS) [18] introduces accelerative start and expeditious recovery algorithms. The accelerative start is similar to the speedy start but sets the IW to min (4 × SMSS, max (2 × SMSS, 4380 bytes)) instead of one SMSS. The expeditious recovery employs explicit error notification (EEN) to distinguish congestion losses from error losses. Both TCP-Swift and TCP PBS require priority mechanisms at all intermediate routers in the data transmission path.

TCP Star [19] implements the following three new mechanisms: congestion window setting (CWS), lift window control (LWC), and acknowledgment error notification (AEN). The CWS is employed to determine the cause of losses in order to adjust the *CWND* accordingly. The LWC increments the *CWND* during slow start and congestion avoidance phases based on available bandwidth estimation mechanism of TCP Jersey [20]. The AEN prevents unnecessary retransmission of segments caused by ACK losses or delays.

TCP Hybla [21] employs a time-scale modification algorithm to increment *CWND* independent of RTTs during the slow start and congestion avoidance phases. However, it assumes that the transmission rate does not depend on the *RWND*. The TCP Hybla algorithm employs the SACK and timestamp options to recover multiple losses and prevent delays in RTO timer update, respectively.

TCP New Vegas [22] is a variant of TCP Vegas [23] proposed for GEO satellite networks. It employs packet pacing and rapid window convergence to improve the performance of TCP during slow start. TCP New Vegas also implements packet pairing to reduce the negative impact of delayed ACK [24] on networks with large RTTs. Some optimizations to TCP New Vegas have been proposed involving improving the Rapid Window Convergence (RWC) algorithm [25]. RWC algorithm aims to rapidly increase the window between the slow-start and congestion-avoidance phases, reducing the time taken by the window to converge to an optimal value. Based on TCP Vegas, TCP DVegas is applied over the satellite links in split TCP connections, and employs active queue management at the medium access control layer for immediate cross-layer congestion feedback to the TCP virtual sources at the PEPs. This method substantially shortens congestion feedback delay and enables transmission losses to be effectively differentiated from congestion losses. Simulation results show that it improves the performance of TCP Vegas when traffic load is heavy and transmission losses increase [26].

TCP Veno [27] proposes an enhancement of TCP Reno congestion control algorithm by using the estimated state of a connection based on TCP Vegas (to deduce what kind of packet loss–congestion loss or random loss—is most likely to have occurred). This scheme significantly reduces "blind" reduction of TCP window regardless of the cause of packet loss. The advantage feature of TCP Veno is that it only needs simple modification at sender side of Reno protocol stack without changing the receiver-side protocol.

1.2.3 Performance Enhancement Proxies solution (PEPs)

The PEP approach has been widely considered as an effective and practical solution to improve TCP performance over satellite links. The advantage of deploying PEPs is that they can act on behalf of end systems without changing their TCP configurations. Two kinds of TCP PEPs have been proposed in the literatures: TCP spoofing PEPs [28] and TCP split connection PEPs [29]. When a PEP ‘splits’ a connection, the proxy pretends to be the opposite endpoint of the connection in each direction, literally splitting the connection into two. When a PEP ‘snoops’ into a connection, the proxy controls the transmissions of the TCP segments in both directions, by ACK filtering and reconstructing the existing connection. By locally acknowledging TCP segments, spoofing proxies solve the problem of slow start up speed of TCP over networks with long delays. However, the problems of long congestion control feedback delay and inability to differentiate between packet losses due to congestion or transmission errors are still present but transferred to the spoofing proxies. TCP split connection proxies can be customized to compensate for specific link characteristics that would otherwise cause poor performance.

As the line between intranets and the Internet continues to fade, the implications of using PEPs start to become more significant for VSAT networks. It is important to notice that the PEP approach has two relevant drawbacks: breaking end-to-end semantics of data connections and its inability to handle IP security (IPSec). However, these drawbacks can be avoided. Some of the proposed solutions are outlined next.

Breaking end-to-end semantics means that the reliability of transmissions is achieved on a hop-by-hop basis as the sender node is not able to infer data delivery information from its actual destination but only from its near PEP. This drawback can be avoided by implementing the networking architecture called Delay-Tolerant Networking (DTN). Its main idea is considering the subdivision of an heterogeneous network into homogeneous sub-regions and relies upon the introduction of a new layer, “bundle layer” between transport and application layers [36]. In contrast to PEP, DTN modifies the protocol stacks, not only at bridge routers, but also at sender and receiver nodes, through the introduction of this additional layer. Moreover, it increases computational processing requirement for DTN router implementing routing functionality on top of the protocol stack.

PEPs were thought to be incompatible with IPSec because IPSec encrypts the IP packet, including application and transport headers, into the IPSec Encapsulation Security Payload. Due to the IPSec, it was impossible for the intermediate PEP to examine the application and transport header and the PEP may not function optimally. Thus the possibility of using IPSec was restricted or sometimes absent in cooperation with PEP. However, recent publications have shown that IPSec can coexist with PEPs. Using hash values and sequence numbers [37], the PEP is able to match packets and corresponding acknowledgements to regulate the flow. Another approach is to use a mapping table at the PEP agent that maps the flow identification data and sequence number corresponding to every session that needs performance enhancement [38].

Besides trying to mitigate or solve PEP's drawbacks, there are many other enhancement mechanisms that can be applied to PEPs. The proposed protocols for enhancing performance of satellite links can be classified into three main groups: *TCP PEPs, Non-TCP PEPs and Onboard satellite PEPS.*

1.2.3.1 TCP Performance Enhancement Proxies (PEPs)

The mechanisms based on TCP versions are unable to distinguish between packet losses due to congestions or transmission errors. As an example, proposals like [30] and [31] use extensions of TCP Reno to enhance performance.

1.2.3.2 Non-TCP Performance Enhancement Proxies solution (PEPs)

The protocols from this group employ standard TCP algorithms and/or satellite specific algorithms. Besides at sender and receiver, they require modification at intermediate nodes. Their best results are obtained when used in satellite segments of split TCP connections.

The Space Communications Protocol Standards-Transport Protocol (SCPS-TP) [39] employs extensions and enhancements to TCP such as selective negative ACK (SNACK) and explicit Internet control message protocol (ICMP) messages for corruption induced losses and link outages. It also employs timestamps and modified delayed ACK options to compute ACK delays based on the estimated RTT. SCPS-TP, with the TCP Vegas option for slow start, improves throughput and is less sensitive to link delays than TCP [40]. Simulation results show that SCPS-TP has better performance in noisy asymmetric satellite networks than TCP.

Satellite transport protocol (STP) [32], is based on an Asynchronous Transfer Mode (ATM) link layer protocol known as service specific connection orientated protocol (SSCOP) [41]. STP does rely on timeout mechanisms. It employs an automatic repeat request (ARQ) mechanism that uses SNACK for retransmission. The ACK polling cycle of STP is used for probing and early error detection [42] and, thus, it performs well in determining the cause of losses. The rate control scheme (RCS) [43] employs low priority dummy packets to probe the network for available resources and requires all routers to support priority schemes for discarding these packets when congestion occurs. PETRA [33] enhances STP with a proxy congestion control protocol. It introduces a new layer over STP to preserve end-to-end semantics. Most of these protocols have shown effectiveness in overcoming problems stemming from a large bandwidth delay product and improving slow start performance. However, due to a very high RTT over satellite, and end-to-end feedback congestion control taking one RTT after congestion is detected before it becomes effective at the point of congestion, the buffer size employed at a split connection PEP has to be as large as the order of the bandwidth delay products [34].

Explicit control protocol (XCP) [44] and its enhanced variant P-XCP [45] employ explicit feedback to determine network conditions and decouple utilization and fairness control. This decoupling of congestion control's efficiency and fairness policies enables routers to quickly make use of available bandwidth while conservatively managing the allocation of bandwidth to flows. XCP is built upon a new principle: carrying per-flow congestion state in packets. XCP packets carry a congestion header through which the sender requests a desired throughput. Routers make a fair per-flow bandwidth allocation without maintaining any per-flow state. Thus, the sender learns of the bottleneck router's allocation in a single round trip. Simulation results show that an XCP PEP is able to utilize available bandwidth faster than TCP variants such as TCP Reno or TCP New Reno. The scheme substantially outperforms TCP in terms of efficiency in high bandwidth–delay product environments [46]. However, the main drawback of this protocol is that it assumes a pure XCP network and requires significant modifications at the end-system.

Stream Control Transmission Protocol (SCTP) [47], [48] employs the TCP congestion control algorithms and other unique features such as multi-streaming and multi-homing.

1.2.3.3 Onboard satellite PEPs

PEPs can also be deployed onboard satellites [35] to further subdivide the connection into earth-to-space and space-to-earth connections and further reduce the RTT over each segment by approximately one half, thus reducing the buffer size over each segment. However, this alternative requires satellites with advanced onboard processing capabilities, which are not commonly available.

1.3. CONTRIBUTION

This paper aims at evaluating and comparing five control algorithms, which are New Reno, Hybla, Vegas, DVegas and Westwood+, using Ns-2 simulations over GEO and LEO satellite links between PEPs.

We have chosen these TCP versions among the best performance candidates available from the literature. TCP New Reno is always interesting to look at and compare to, since it is the most common implementation of TCP. TCP Westwood+ is a similar and also a widely recognized version of TCP. TCP Vegas and TCP Hybla, are specialized for long delay error prone networks. Finally, TCP DVegas improves TCP Vegas by adding an immediate cross-layer congestion feedback to the TCP virtual sources at the PEPs. TCP Reno and SACK are included as baselines for our comparisons.

1.4. PAPER OUTLINE

The rest of the paper is organized as follows. Section 2 presents the system architecture, Section 3 presents the performance evaluation and simulation results. Finally, Section 4 presents the conclusion and future work.

SECTION II

SYSTEM ARCHITECTURE

2. SYSTEM ARCHITECTURE

2.1 LEO- AND VSAT- BASED NETWORKS

Both, the LEO- and VSAT-based networks can work only when the terminal and the satellite "see" each other. The very small aperture terminals (VSAT) communicate with geostationary satellites at 36.000 km distance, which can be seen only from a part of earth surface. VSATS were designed to allow a number of satellite hosts to share a satellite channel in a thin route network configuration to counteract the expensive price of satellite bandwidth. GEO satellite’s strong points are: the simple base station working mechanism (as GEO satellites are always on the same position, no tracking and handover procedures are needed), their higher reliability, long operational lifetime and lower development risks. Their main weaknesses are: onboard complexity, expensive cost and the large number of users affected if malfunction is experienced.

On the other hand, LEO satellite constellations are at lower distances of 1.500 km, and are visible practically from any point on the earth. LEO satellite’s strong points are: its lower propagation delay, lower RTTs and its ability to communicate with small devices. Because the signal travels a relative short distance, mobile handheld devices are able to pick up the signal sent by a LEO satellite. Thus, for the same transmission power, LEO signal is stronger than GEO. The main weaknesses of LEO satellites are: the complex tracking procedures and handover mechanisms needed at the base station and the fact that a whole constellation has to be developed in order to provide broad coverage.

2.2 SPLIT CONNECTION WITH PERFORMANCE ENHANCEMENT PROXY

A typical scenario for PEP implementation is a satellite link along the data path. The PEP agent is installed at the edge router bridging terrestrial and satellite networks. As a result, the end-to-end communication flow is split into two sections: a low-delay terrestrial link using standard TCP and a high-delay satellite link using a protocol specifically designed for satellite networks. For the satellite link, suitable scenarios based on two satellite configurations were considered: GEO satellite configuration and polar LEO configuration. Suitable scenarios based on these configurations were implemented.

When working with Ns-2 simulator, we decided to follow a divide and conquer strategy to accomplish our final system architecture simulation. For the GEO satellite configuration, “bent-pipe” satellite architecture was modeled. Nowadays, VSAT networks are implemented with geosynchronous satellites [49], most of them using a star topology. This way, our first step was to implement a star topology as shown on Figure 1. A large hub earth station is located at the centre of the star with VSATs used at the remote sites of the network.

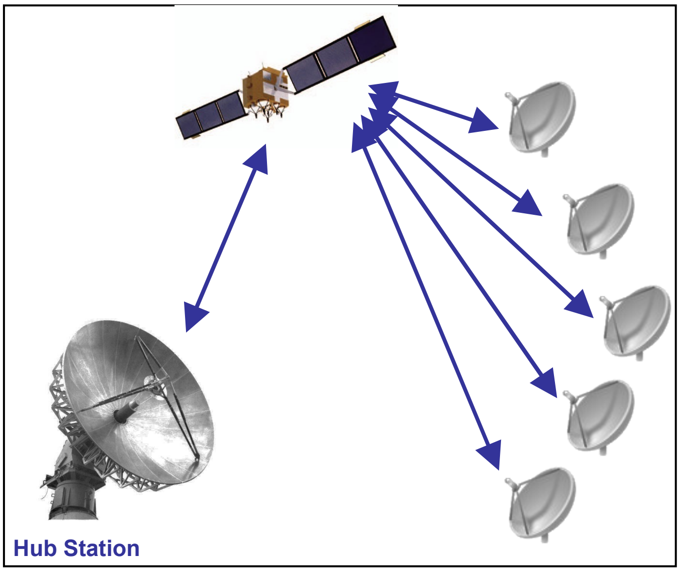

Figure 1. Star topology network.

For the polar LEO configuration, Ns-2 simulator offers two network models: Iridium and Teledesic. Table 3 shows the differences between them. For this configuration, we decided to model an Iridium polar orbiting LEO constellation.

	Iridium	Teledesic
Altitude	780km	1375
Planes	6	12
Satellites per plane	11	24
Inclination (deg)	86.4	84.7
Interplane separation (deg)	31.6	15
Seam separation (deg)	22	15
Elevation mask (deg)	8.2	40
Intraplane phasing	Yes	Yes
Interplane phasing	Yes	No
ISL per satellite	4	8
ISL bandwidth	25 Mb/s	155 Mb/s
Up/downlink bandwidth	1.5 Mb/s	1.5 Mb/s
Cross-seam ISLs	No	Yes
ISL latitude threshold (deg)	60	60

Table 3. Simulation parameters used for modeling a broadband version of the Iridium and Teledesic polar orbiting constellation systems.

Secondly we had to decide which kind of PEP we wanted to use in our simulation. Ns-2 simulator offers the possibility to work with either spoofing or splitting PEPs. The latest modules available for Ns-2 simulator are Snoop (spoofing) and SaTPEP (splitting). On the one hand, SaTPEP splitting proxy introduces specialized congestion control and loss recovery mechanisms based on link utilization measurements [50]. On the other hand, the Snoop proxy is implemented in the data link layer but is capable of monitoring the flow of TCP connection packages in both directions. It stores on its cache all the TCP segments that have not yet been recognized by the receiver. The main advantage of this is that the proxy retransmits the lost segments locally. This way the sender doesn't have to recognize duplicated segments and no longer has to use congestion control mechanisms and fast recovery. There is not a dominant PEP implementation used with VSAT or LEO networks. For this simulation we decided to implement a Snoop network model as shown on Figure 2.

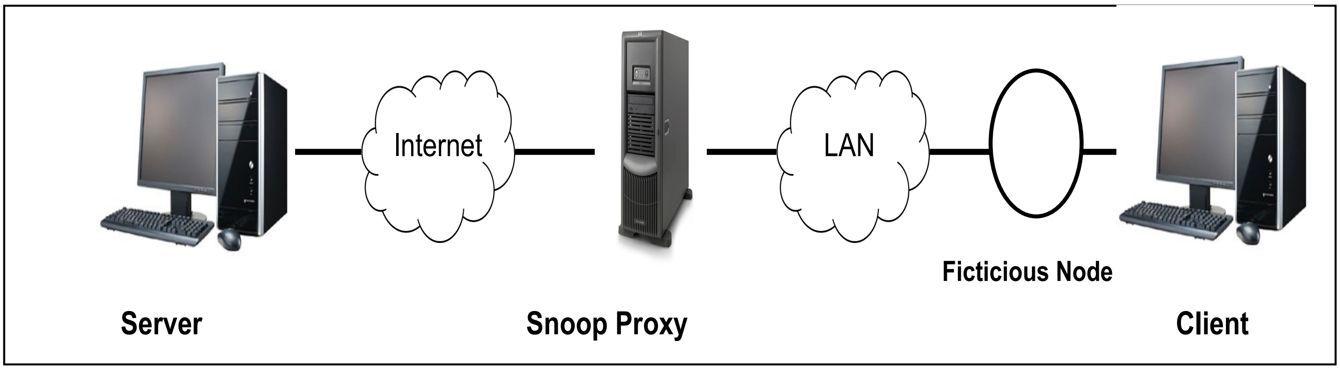

Figure 2. PEP simulation network.

Finally, Figure 3 illustrates the final system architecture of our VSAT network for Internet access. The test bed is treated as two different segments separated by a PEP. Once the VSAT network was completed, the LEO-based network shown on Figure 4 was easier to model. In both network architectures a number of VSATs/LEO GW terminals (GEO and LEO configurations, respectively) are located at the subscriber premises, which enable end hosts to access the Internet via satellite. In order to efficiently share satellite bandwidth, the VSATs employ a MAC protocol to access a shared uplink. In both GEO and LEO split TCP configuration, one Snoop proxy is located at the gateway earth station (GES), connected to the GES through a LAN. The GES is used to connect the satellite network to local web servers or the global Internet. When an end host wants to set up a TCP connection, the Snoop PEP will intercept this request and set up cascading TCP connections for it. The gateway is granted fixed bandwidth in a dedicated uplink channel. The Snoop PEP is responsible for local acknowledgments and retransmission over the satellite network.

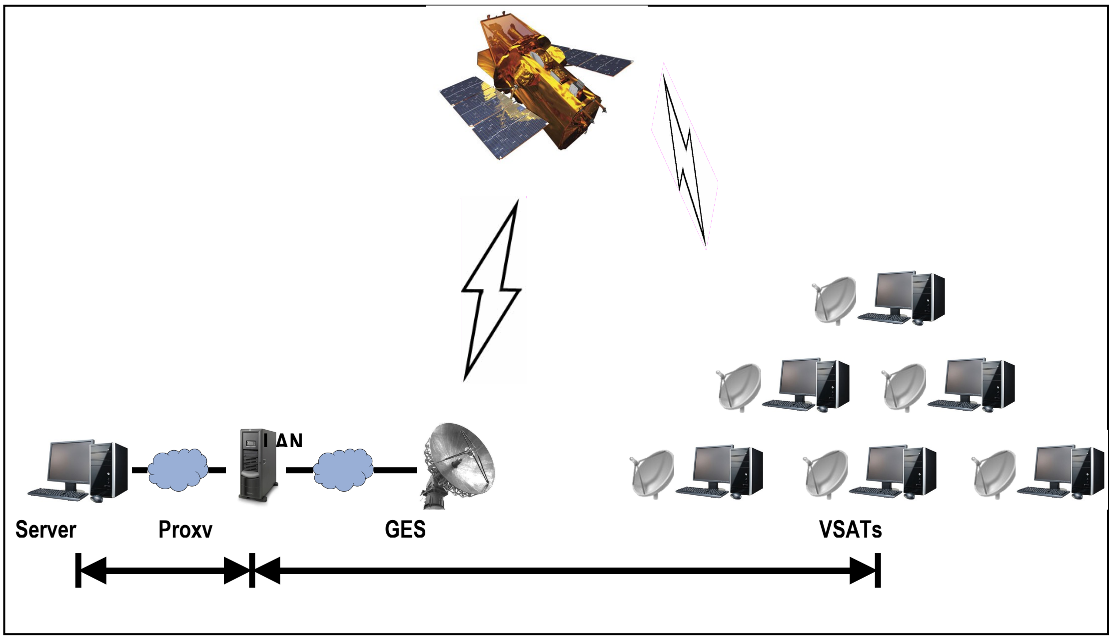

Figure 3. System Architecture I: GEO “bent pipe” satellite architecture.

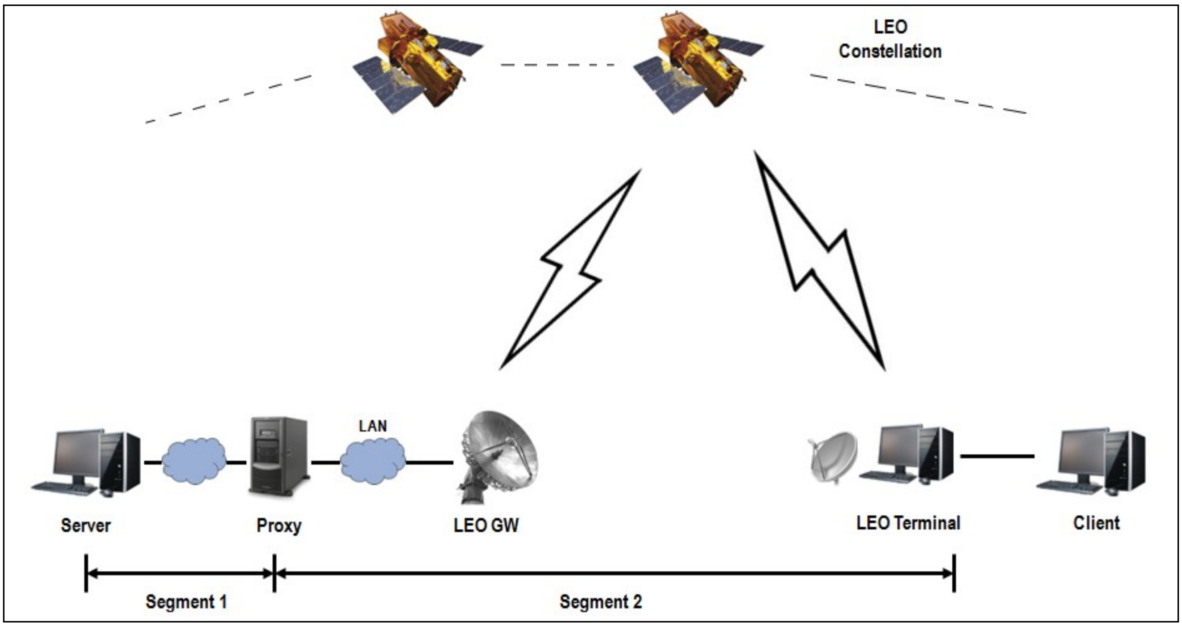

Figure 4. System Architecture II: polar orbiting LEO constellation architecture.

2.3 MAC PROTOCOL

The Combined Free/Demand Assignment Multiple Access (CF-DAMA) [51] is a hybrid MAC protocol that offers excellent delay and throughput performance for data traffic via satellites. It combines the free assignment of data slots, as well as demand assignment, which can achieve low access delay when traffic load is light, and efficient statistical multiplexing over the shared satellite channel when traffic load is heavy. It has been shown that the CF-DAMA protocol is superior to slotted ALOHA over virtually the entire range of channel utilization [52].

Recently, a Ns-2 TDMA DAMA patch for Geo satellite network simulation has been released [53]. With the installation of this patch, Ns-2 simulator's supported MACs are Unslotted-Aloha (with or without collision detection) and TDMA. Using this patch makes our simulations more realistic because we consider that the technique used to assign time slots in a TDMA satellite frame can affect significantly the dynamic of TCP startup and its response to bandwidth changes.

SECTION III
PERFORMANCE EVALUATION

3. PERFORMANCE EVALUATION

We use Network Simulator version 2.31 [54] as the simulation tool for performance evaluations. The different TCP versions available are compared when applied over the proposed satellite link between PEPs. Comparisons of the TCP versions with end-to-end TCP protocols are not included in the paper since it is well-known that a PEP scheme will out-perform end-to-end TCP over a satellite network [55]. Since TCP Reno is the most popular version implemented in current networks, we use it as baseline for the comparison with PEP schemes. The CFDAMA protocol is implemented at MAC layer over the satellite link. The comparison of different MAC protocols over satellite links is beyond the scope of the paper.

As shown on Figures 3 and 4, the implementation of the VSAT-based network includes 5 VSATs at the user side, while the LEO-based network includes one LEO terminal connected to the potential client. The TCP throughput is defined as the number of received data bits divided by the simulation time. Table 1 and 3 list the simulation parameters used in simulations, except otherwise noted. Considering that system bandwidth should be shared by bursty traffic to take advantage of statistical multiplexing, instead of long-lived traffic, we use a traffic model that approximate World Wide Web traffic [56] as the web represents the most popular application that transmits data over TCP. The comparison also takes into account FTP and UDP as background traffic for our simulations.

Our performance evaluation simulation results can be divided into five groups, depending on which situations and characteristics are studied. These evaluation groups are the following: BER, Uplink bandwidth, Traffic load, Snoop proxy and XCP as TCP PEP.

3.1 IMPACT OF BER

We test the impact of the BER on TCP performance in both scenarios, varying the BER of the VSATs and LEO terminal from 10-7 to 10-5. The former scenario is common in satellite networks where rain attenuation affects a subset of the earth stations. The other simulation parameters have the same values as shown in Table 1.

Parameter Items	Parameter Values
Bandwidth for Terrestrial Link	10 Mbps
Propagation Delay for Terrestrial Link	50 ms
Bandwidth for Gateway over Satellite	2/6/10 Mbps
Packet length for downstream (including TCP/IP headers)	1024 Bytes
Packet length for upstream (including TCP/IP headers)	128 Bytes
Propagation delay for satellite link	250 ms
Bit Error Rate (BER)	10^{-7} /10^{-6}/ 10^{-5}
Simulation time	25 s

Table 1. Simulation parameters for VSAT model.

In the GEO “bent pipe” satellite scenario the overall throughput does not suffer much degradation due to the adaptive congestion mechanism of TCP. Other TCP connections take bandwidth from those suffering from BER degradation. TCP Hybla is specifically designed to work with high packet error rates, so its good performance was expected. DVegas ranked second because of his successful decoupling of congestion control and error control. As shown on Figure 7, TCP Vegas offers a better performance than TCP Reno because TCP Vegas only cuts its congestion window by one fourth if a packet is lost for the first time. On the figure, TCP Hybla's results have been reduced in order to appreciate the other TCP variants in the graph.

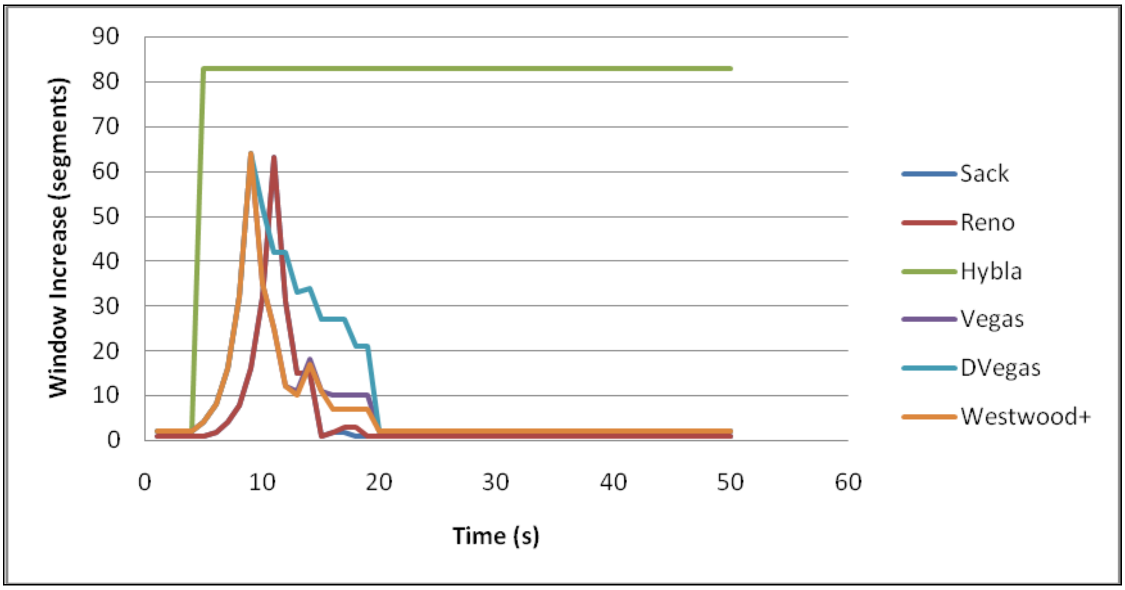

Figure 7. Impact of BER on Congestion Window increase of VSAT terminals experiencing BER=10^{-7} NOTE: TCP Hybla's results have been reduced in order to appreciate the other TCP variants in the graph.

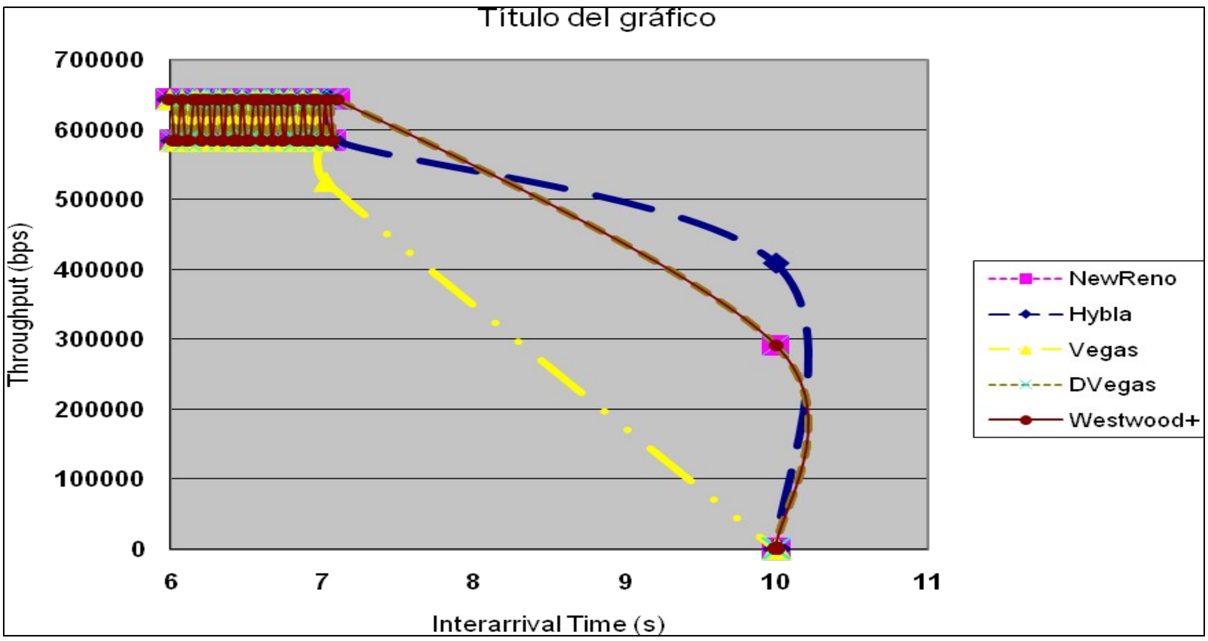

Figure 8. Impact of BER on Throughput of LEO terminals experiencing BER=10^{-7}

In the polar orbiting LEO constellations scenario, Figure 8 shows that when working in prone error satellite networks, TCP Hybla once more outperforms the rest of TCP variants. However, when we drastically reduce the error rate as shown on Figure 9, TCP Hybla hands over its first place to TCP Westwood+. The reason for this change is that while TCP Hybla was specifically designed to counteract RTT-unfairness, TCP Westwood+ was mainly conceived to tackle random error networks. TCP Westwood+ adaptively reduces the *CWND* and SSTRESH by taking into account an estimation of the available bandwidth. This reduction mitigates the impact of random errors not due to congestion that provoke premature reductions of TCP New Reno and TCP Vegas's control windows.

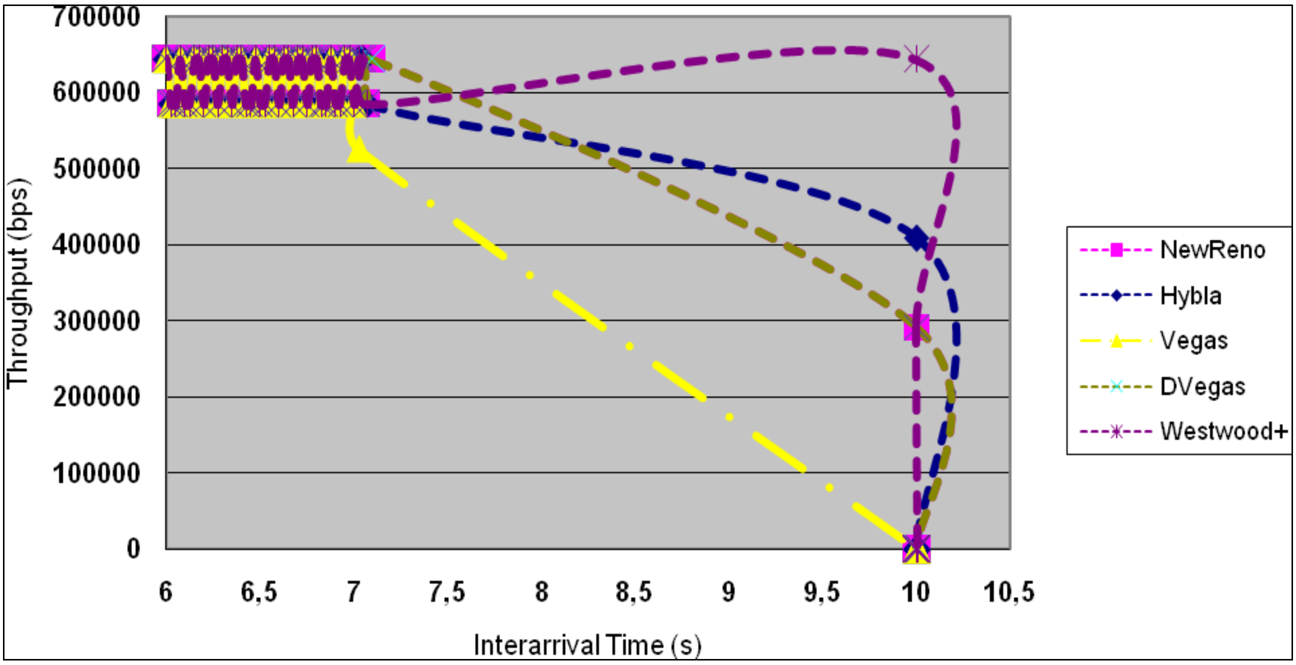

Figure 9. Impact of BER on Throughput of LEO terminals experiencing BER=10^{-5}

3.2 IMPACT OF UPLINK BANDWIDTH

In the polar orbiting LEO constellations scenario, the throughput results do not suffer much impact when the bandwidth is modified. This can be explained by the lower propagation delay of the Inter Satellite Link (ISL) of LEO networks. This way, in this section we will focus on evaluating the results from the GEO "bent pipe" satellite scenario.

We investigate the impact of the uplink bandwidth (from VSAT to proxy) on the TCP throughput as well as the round trip delay (satellite segment only). While the downlink bandwidth (from proxy to VSAT) is kept constant at 6 Mbps, the uplink bandwidth changes from 2Mbps to 10Mbps.

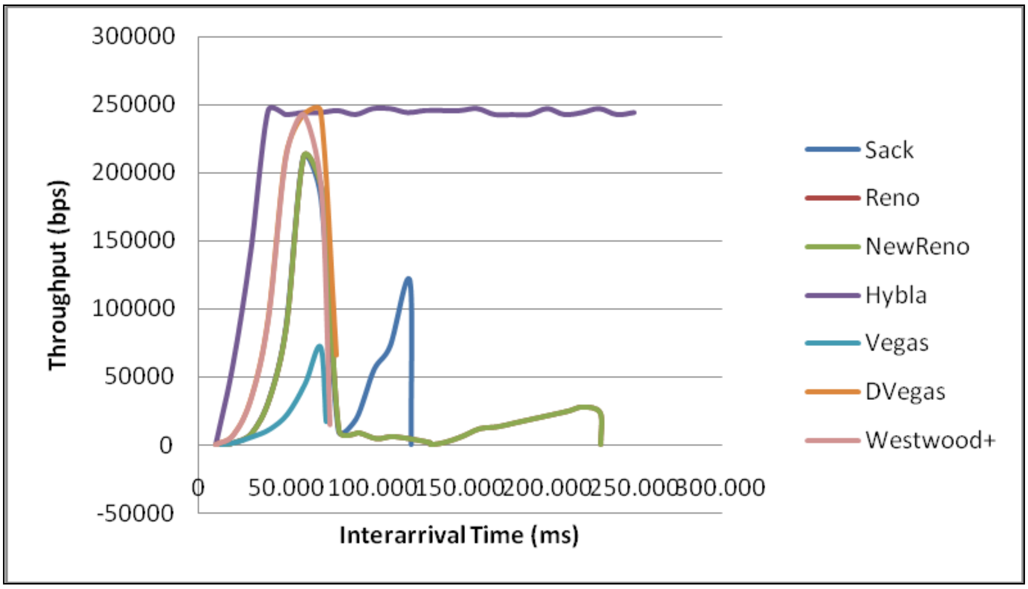

Figure 10. Impact of Uplink bandwidth. GEO "bent pipe" satellite scenario. Throughput. Uplink bandwidth = 2Mb, Download bandwidth = 6Mb, BER = 10^{-7}

Figures 10 and 11 show the TCP throughput evolution when the uplink bandwidth is increased keeping the BER and downlink bandwidth constant. In both figures, TCP Hybla is the fastest in exploiting the link bandwidth, reaching the steady state before the other TCP variants. This is because TCP Hybla's congestion control algorithm allows a fast opening of the *CWND* to counteract the effects of large RTTs. When the uplink bandwidth is 2Mb (Figure 10) all the TCP protocol suffer a clear degradation in throughput, and the results are more levelled between the TCP versions.

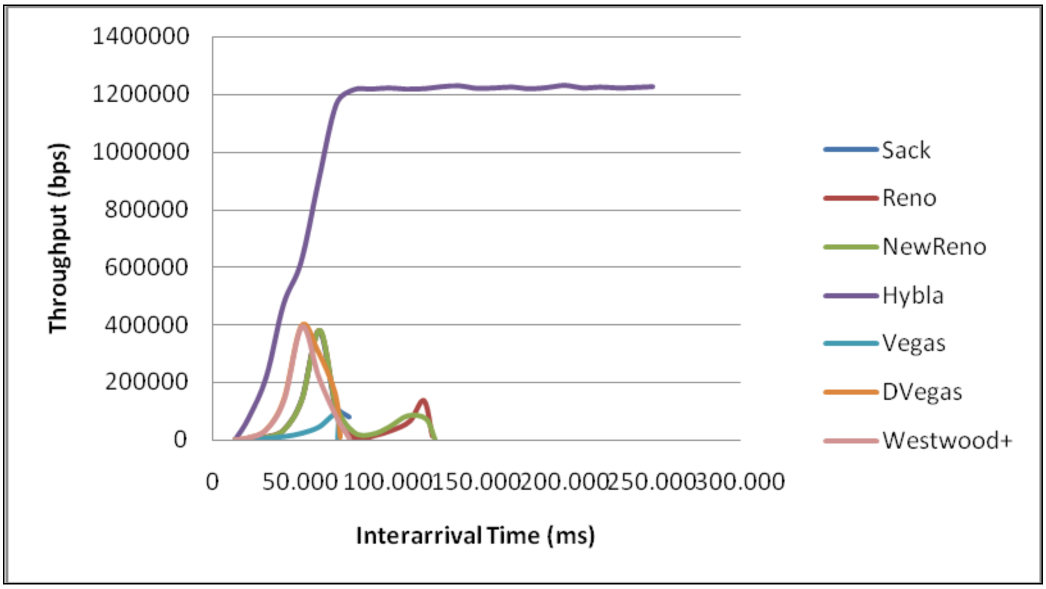

Figure 11. Impact of Uplink bandwidth. GEO "bent pipe" satellite scenario. Throughput. Uplink bandwidth = 10Mb, Download bandwidth = 6Mb, BER = 10^{-7}

Figure 12 shows the impact of VSAT uplink bandwidth on throughput. TCP Hybla performs best, followed by DVegas and TCP Westwood+. Packet spacing and initial bandwidth estimation, allows TCP Hybla to avoid the early buffer overflow produced when packets first arrive to the router. This buffer overflow affects all other TCP variants, except DVegas and TCP Westwood+. After this fast explosion, TCP Hybla's performance levels with that of DVegas and TCP Westwood+. The good performance of DVegas is due to its dynamic congestion control. TCP Westwood+

doesn't have RTT fairness so its performances is slightly worse than DVegas, as it has to continuously update to a correct *SSTHRESH* value. TCP Vegas performed last, unable to succeed in bandwidth share when it coexists with TCP Reno because of its RTT-based congestion detection mechanism implying α and β thresholds reduction.

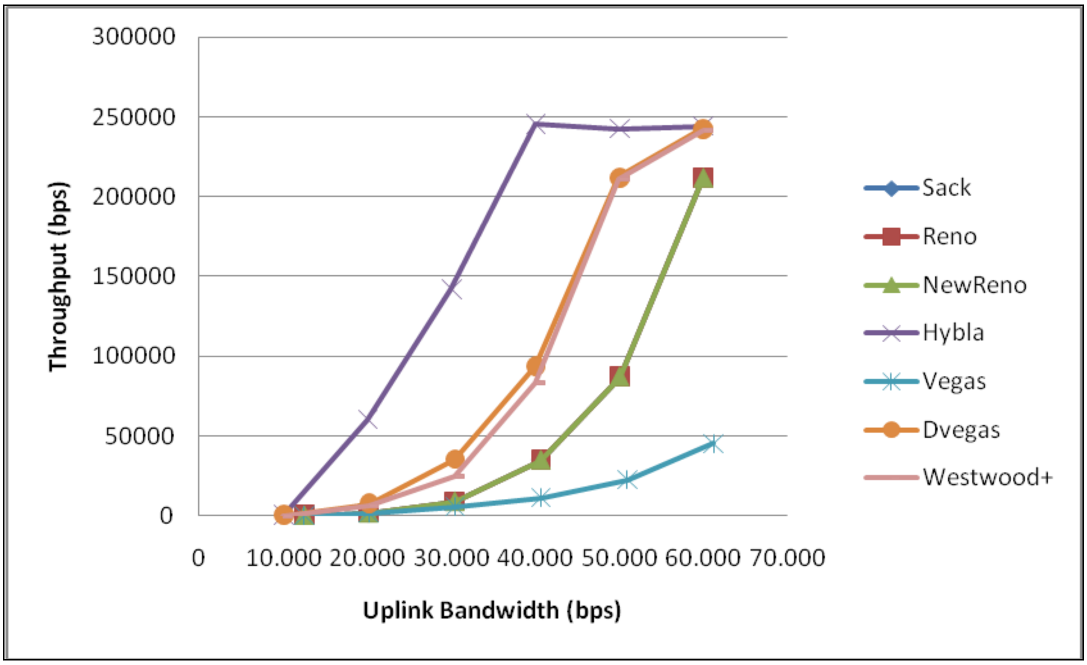

Figure 12. Impact of VSAT Uplink bandwidth on Throughput.

3.3 IMPACT OF TRAFFIC LOAD

We study the effect of traffic load on TCP performance, focusing on throughput and sequence number increase. Both GEO and LEO scenarios produced similar results when traffic load is modified. Figures 13 and 14 show the simulation results when the load is heavy on the LEO scenario. As explained above, TCP Hybla outperforms the rest of the TCP variants. In both cases, TCP Reno performs worst. This can be explained considering the fact that TCP Reno needs more time to recover from packets dropped by transmission errors, while SACK can recover quicker using selective acknowledgments.

Figure 15 shows the simulation results when the traffic load is light on the GEO scenario. In this case TCP Vegas has the worst throughput, being unable to cope with the small thresholds ballast. On the figure, TCP Hybla's results have been reduced in order to appreciate the other TCP variants in the graph.

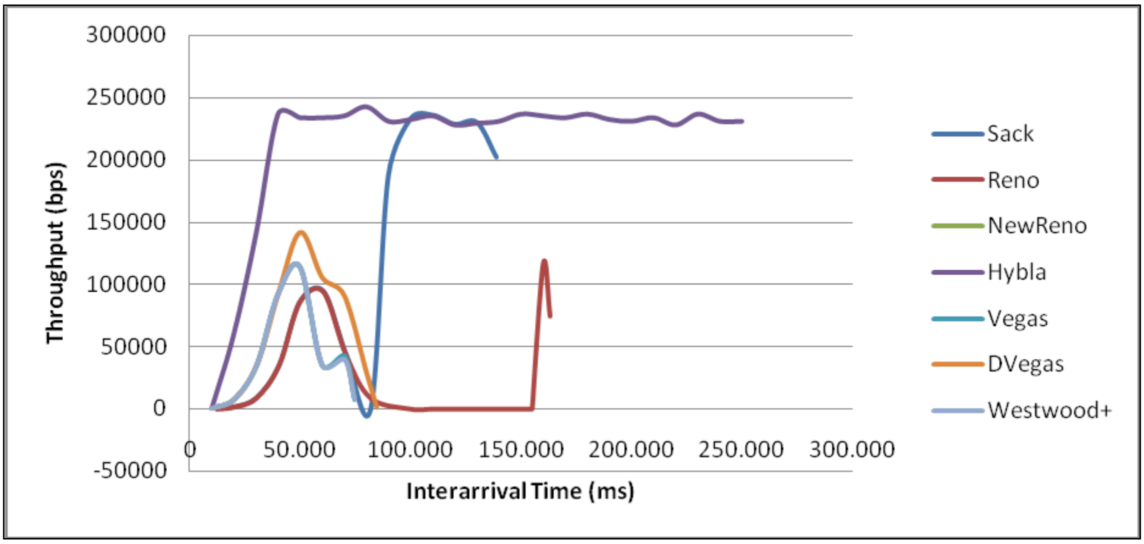

Figure 13. Impact of traffic load on Throughput. LEO Constellation satellite scenario.

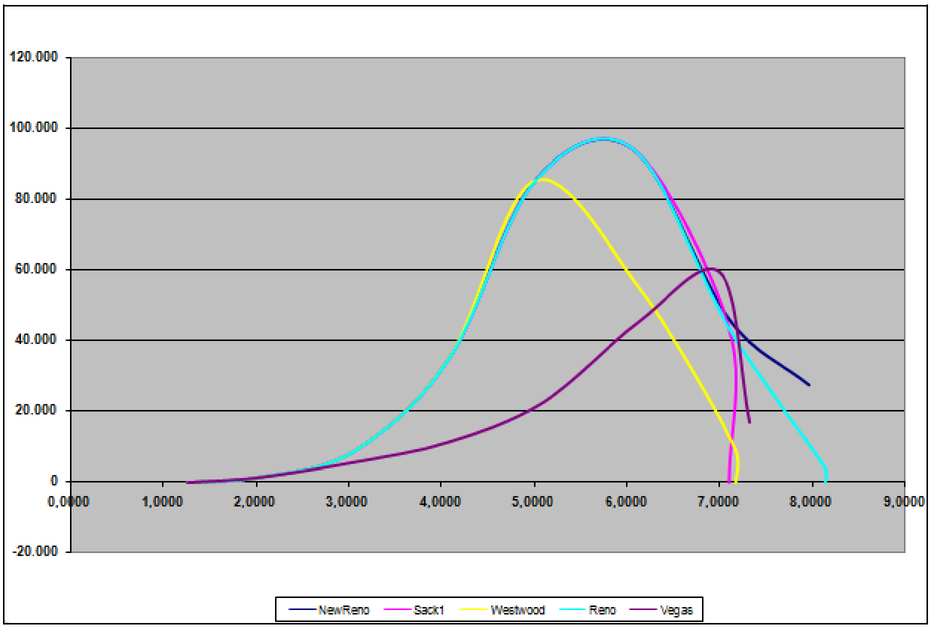

Figure 14. Impact of traffic load on Sequence Number Increase. LEO Constellation satellite scenario.

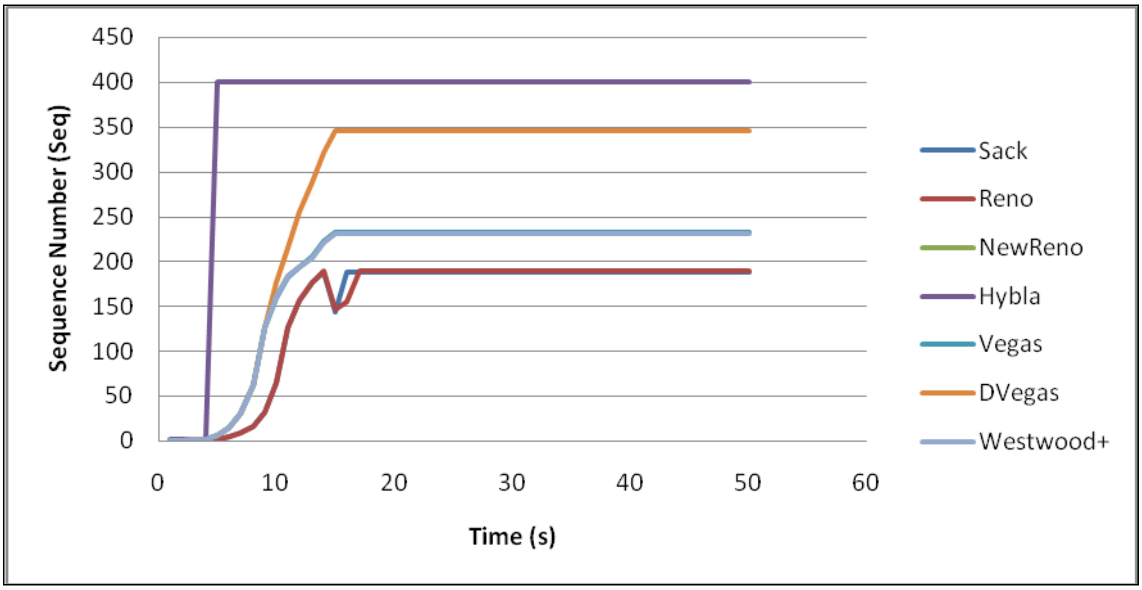

Figure 15. Impact of traffic load on Throughput. GEO "bent pipe" satellite scenario. NOTE: TCP Hybla's results have been removed in order to appreciate the other TCP variants in the graph.

3.4 IMPACT OF SNOOP

In the Segment 1 of our system architecture (see Figure 2) we wanted to test the convenience of using Snoop. We tested proxy with two types of traffic generators: CBR and FTP. When using a constant bit rate (CBR) Snoop improves the average throughput in all our simulations. If we decrease the bandwidth in the links between our simulation nodes, the number of retransmissions also decreases as there's less traffic in the system. Table 2 shows us that when using a FTP server Snoop doesn't always improve the throughput. As an example Figure 5 shows how the presence of Snoop proxy doesn't improve the TCP throughput when tested on low speed links (0.1Mb). As the proxy eliminates all the duplicate acknowledgements from our system, the number of retransmissions from the sender and consequently superfluous traffic are reduced. The routing convergence time of is the amount of time it takes the PEP to adapt to a change. The convergence time due to packet drops gets lower when the bandwidth increases, as shown on Figure 6. Both in high bandwidth and low bandwidth links the proxy reduces the number of retransmissions. However in low bandwidth links the convergence time of the proxy is greater than in the same situation without proxy.

		Bandwidth Sender-Snoop PEP (Mbps)	Average Throughput (bps)
FTP	No Snoop	1	19432,7272
		0,1	18618,1812
	Snoop	1	19432,7272
		0,1	18618,18182
CBR	No Snoop	1	14196,3636
		0,1	1698.0909
	Snoop	1	21527,2727
		0,1	21527,2727

Table 2 Throughputs obtained in Segment 1. Average

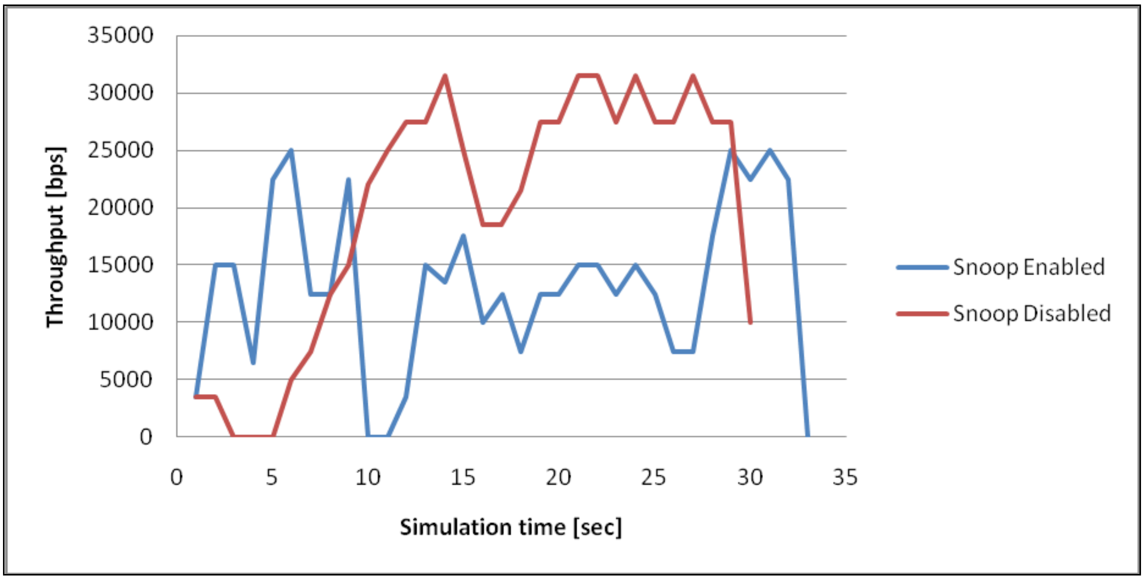

Figure 5. Throughput with and without Snoop. FTP. 1Mbps .Sender – Snoop proxy.

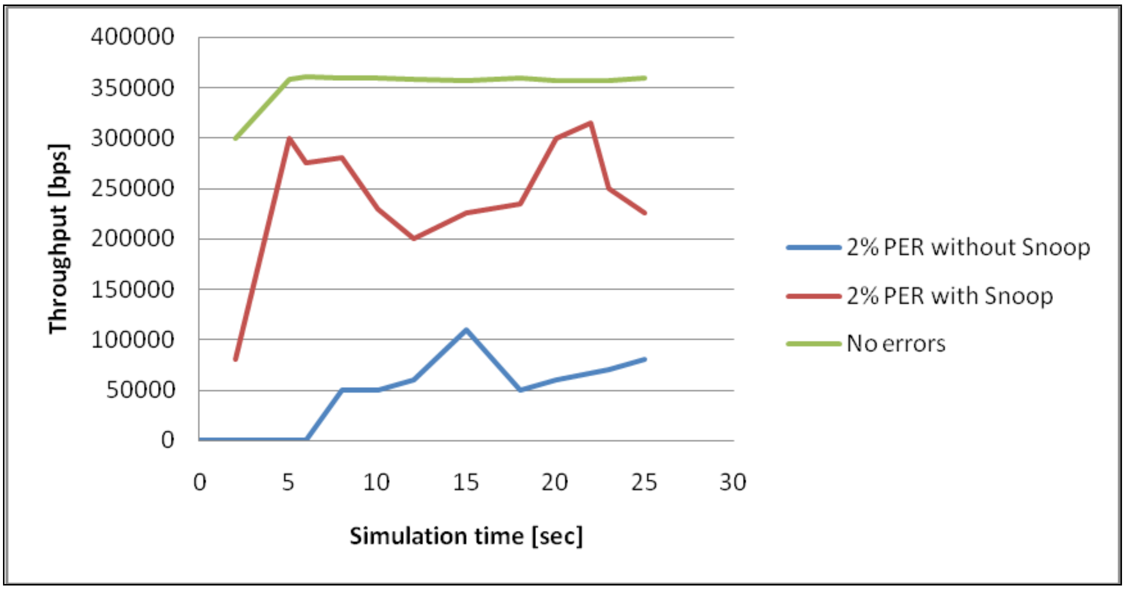

Figure 6. Increase of throughput with Snoop proxy and 2% PER.

3.5 IMPACT OF XCP AS TCP PEP

When considering the study of other protocols that are able to dynamically adapt to the congestion state of the network, XCP (Explicit Control Protocol) seemed the best option. XCP represents a major advance in Internet congestion control, delivering the highest possible application performance over a broad range of network infrastructure, including extremely high speed and very high delay links that are not well served by TCP [44]. In so doing, it achieves maximum link usage and wastes no bandwidth due to packet loss. XCP is novel in separating the efficiency and fairness policies of congestion control, enabling routers to quickly make use of available bandwidth while conservatively managing the allocation of bandwidth to flows. XCP is built upon a new principle: carrying per-flow congestion state in packets. XCP packets carry a congestion header through which the sender requests a desired throughput. Routers make a fair per-flow bandwidth allocation without maintaining any per-flow state. Thus, the sender learns about the bottleneck router's allocation in a single round trip.

XCP may be used in between two TCP PEPs, but its router implementation on Ns-2 simulator is beyond the scope of this paper. However, as a simple approach, we created a bottleneck scenario that reproduced the throughput simulation results obtained by the TCP variants in the GEO satellite bottleneck link and we tested XCP's performance in this scenario. From the results shown on Figure 16, we can state a hypothesis: when used as a TCP PEP, XCP will get high transfer rates and still respond to congestion in a correct and scalable way.

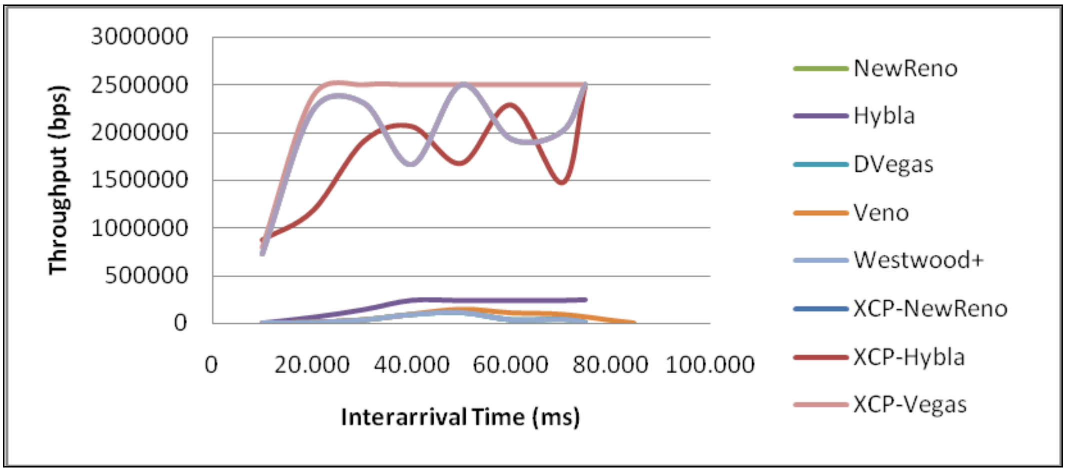

Figure 16. XCP Throughput over bottleneck satellite link.

SECTION IV
CONCLUSIONS

4. CONCLUSIONS

4.1 GENERAL CONCLUSIONS

In this paper our aim was to evaluate and compare five TCP versions, which are New Reno, Hybla, Vegas, DVegas and Westwood+. The obtained results show that TCP Hybla allows a better efficiency in utilizing the available satellite bandwidth, especially for high packet error rates and high satellite bandwidth values. Secondly, from our results we can conclude that Snoop proxy improves end-to-end performance of the satellite link.

The only disadvantage we found when working with Ns-2 simulator is the lack of possibility to create NAM files for satellite networks. NAM files are a typical graphical output available for wired networks. We overcame this problem by using Xgraph program and multiple perl scripts with our traces.

Finally, we have a set of contributions (1) by sharing our experiences on reproducing suitable scenarios based on GEO bent-pipe satellites and polar orbiting LEO constellations with ns2 simulator; (2) on running a satellite network simulation methodology; (3) on presenting our results and gaining understanding of link layer and transport protocols in satellite networks.

4.2 FUTURE WORK

Three research directions appear of major interest for further investigations on the subject: the first one concerns the inclusion of other types of PEPs (SaTPEP) and their enhancements in our system architecture and compare them with the results obtained by Snoop. The second one is to continue working on using XCP as TCP PEP, and the third one is to explore and impulse artificial intelligence approach to satellite networks.

To fulfill the second requirement, we have to take into account that work on using XCP as TCP PEP is just beginning [57]. The goals to improve such mechanism would be: to preserve the end-to-end TCP semantics as much as possible, to enable some form of acknowledge for one PEP to determine whether another PEP was in the data path. Finally, the mechanism should allow some form of recovery in case a PEP fails.

Concerning Artificial Intelligence's approach to satellite networks, the application of knowledge-based and Artificial Intelligence systems to traffic management operations has been an active research area in the last decade [58]. The concept of Intelligent Traffic Management System (ITMS) is developed; a system that embodies a knowledge model of traffic behavior at a strategic level. As this type of knowledge is built from theoretical frameworks together with common sense criteria for making reasonable assumptions, specific methods and techniques have been developed in physical-tangible everyday use examples [59] , [60].

Fuzzy logic based congestion control algorithm which combines ECN and BECN mechanisms [61] and intelligent packet dropping mechanism [62] have been implemented and show efficient simulation results. When trying to apply intelligent systems to improve the congestion control for adaptive satellite communication systems, the main problem encountered is finding the right metrics for the fuzzy logic rules. Satellite-based systems are affected by external factors such as weather and it is difficult to engineer predictable and reliable service metrics such as throughput, delay and jitter when there are uncertainties.

SECTION V

REFERENCES

REFERENCES

[1] A. Jamalipour, M. Marchese, H. Cruickshank, J. Neal, and S. Verma, "Broadband IP Networks via satellites-part I," *IEEE J. Select. Areas Commun.*, vol. 22, no. 2,
pp. 213–217, Feb. 2004.

[2] M. Allman, D. Glover, and L. Sanchez, "Enhancing TCP over satellite channels using standard mechanisms", RFC2488, January 1999.

[3] V. Jacobson and R. Braden, "TCP Extensions for Long-Delay Paths", RFC1072, October 1988.

[4] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno modification to TCP's fast recovery algorithm," RFC 3782, Apr. 2004.

[5] V. Padmanabhan and R. Katz, "TCP fast start: a technique for speeding up web transfers," in Proc. IEEE GLOBECOM Internet Mini-Conf., Sydney, Australia,
Nov. 1998.

[6] M. Gerla, W. Weng, and R. L. Cigno, "BA-TCP: a bandwidth aware TCP for satellite networks," in Proc. Eighth IEEE Int. Conf. on Comput. Commun. and
Netw., Las Vegas, NV, Oct. 1999, pp. 204–207.

[7] J. Peng, P. Andreadis, C. Belisle, and M. Barbeau, "Improving TCP performance over long delay satellite links," OPNETWORK 2001, Washington, DC, Aug.
2001.

[8] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: a new congestion control scheme for satellite IP networks," IEEE/ACM Trans. Netw., vol. 9, no. 3,
pp. 307–321, June 2001.

[9] I. F. Akyildiz, X. Zhang, and J. Fang, "TCP-Peach+: enhancement of TCP-Peach

for satellite IP networks,” IEEE Commun. Lett., vol. 6, no. 7, pp. 303–305, July 2001.

[10] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, “TCP Westwood: end-to-end congestion control for wired/wireless networks,” Wireless Netw., vol. 8, no. 5, pp. 467–479, Sept. 2002.

[11] R. Wang, G. Pau, K. Yamada, M. Y. Sanadidi, and M. Gerla, “TCP startup performance in large bandwidth delay networks,” in Proc. IEEE INFOCOM 2004, Hong Kong, China, Mar. 2004, vol. 2, pp. 796–805.

[12] L. A. Grieco, S. Mascolo,”Performance evaluation and comparison of Westwood +, New Reno and Vegas TCP congestion control”, ACM Computer Communication Review, April 2004, Vol. 34(2).

[13] G. Yang, R. Wang, M. Gerla, and M. Y. Sanadidi, “TCP bulk repeat,” Comput. Commun., vol. 28, no. 5, pp. 507–518, Mar. 2005.

[14] S. Floyd, “HighSpeed TCP for large congestion windows,” RFC 3649, Dec. 2003.

[15] T. Kelly, “Scalable TCP: improving performance in high speed wide area networks,” ACM SIGCOMM Comput. Commun. Rev., vol. 33, no. 2, pp. 83–91, Apr. 2003.

[16] G. Giambene and D. Miorandi, “A simulation study of scalable TCP and highSpeed TCP in geostationary satellite networks,” Telecommun. Syst., vol. 30, no. 4, pp. 297–320, Dec. 2005.

[17] K-F. Leung and K. L. Yeung, “TCP-Swift: an end-to-end host enhancement scheme for TCP over satellite IP networks,” in Proc. Ninth IEEE Int. Symp. on Comput. and Commun., Alexandria, Egypt, July 2004, vol. 1, pp. 551–555.

[18] H. Xu and S. Wu, “A priority-based TCP congestion control strategy in satellite IP networks,” in Proc. Int. Conf. on Commun., Circuits, and Syst., Hong Kong, China, May 2005, pp. 402–406.

[19] H. Obata, K. Ishida, S. Takeuchi, and S. Hanasaki, "TCP-STAR: TCP congestion control method for satellite Internet" IEICE Trans. Commun., vol. E89-B, no. 6,
pp. 1766–1773, June 2006.

[20] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for wireless IP communications," IEEE J. Select. Areas Commun., vol. 22, no. 4, pp. 747–756, May 2004.

[21] C. Caini and R. Firrincieli, "TCP Hybla: a TCP enhancement for heterogeneous networks," Int. J. Satellite Commun. Netw., vol. 22, no. 5, pp. 547–566, Sept.
2004.

[22] J. Sing and B. Soh, "TCP New Vegas: improving the performance of TCP Vegas over high latency links," in Proc. Fourth IEEE Int. Symp. on Netw. Comput. and
Appl., Cambridge, MA, July 2005, pp. 73–82.

[23] L. S. Brakmo and L. L. Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet," IEEE J. Select. Areas Commun., vol. 13, no. 8, pp. 1465–
1480, Oct. 1995.

[24] T. Lang and D. Floreani, "The impact of delayed acknowledgements on TCP performance over satellite links," in Proc. ACM First Workshop on Wireless
Mobile Internet, Rome, Italy, July 2001, pp. 56–61.

[25] Joel Sing, Ben Soh, "Optimising delay based congestion control for geostationary satellite networks" IEEE 2006

[26] Fei Peng, Lijuan Wu, Victor C. M. Leung, "Cross-layer Enhancement of TCP Vegas over Satellite Links between Performance Enhancement Proxies", April 4, 2008

[27] Cheng Peng Fu, Soung C. Liew," TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks". IEEE J. Select. Areas Commun, vol.21, no.2, February 2003

[28] J. Ishac and M. Allman, “On the performance of TCP spoofing in satellite networks,” Proc. IEEE Milcom, pp. 700-704, Vienna, Virginia, Oct. 2001.

[29] J. Border, M. Kojo, J. Griner, G. Montenegro and Z. Shelby, “Performance enhancing proxies intended to integrate link-related degradations”, RFC3135, June 2001.

[30] T. R. Henderson and R. H. Katz, Transport protocols for Internet compatible satellite network, IEEE Journal on Selected Areas in Communications, vol. 17, pp. 326-344, Feb. 1999.

[31] M. Filip and E. Vilar, Optimum utilization of the channel capacity of a satellite link in the presence of amplitude scintillations and rain attenuation, IEEE Transactions on Communications, vol. 38, pp. 1958-1965, Nov. 1990.

[32] T.R. Henderson and R.H. Katz, “Transport protocols for Internet-compatible satellite network,” IEEE Journal on Selected Area in Comm., vol. 17, pp. 326-344, Feb. 1999.

[33] M. Marchese, M. Rossi, G. Morabito, “PETRA: Performance Enhancing Transport Architecture for Satellite Communications”, IEEE Journal on Selected Areas in Communications, vol. 22, pp. 320-332, Feb. 2004.

[34] V. Mhatre, C. Rosenberg, “Performance Improvement of TCP-based Applications in a Multi-access Satellite System”, IEEE VTC, Vancouver, Canada, September 2002.

[35] M. Luglio, M.Y. Sanadidi, M. Gerla, J. Stepanek, “On-board satellite "split TCP" proxy”, IEEE Journal on Selected Areas in Communications, vol. 22, pp. 362–370, Feb. 2004.

[36] Carlo Caini, Piero Cornice, Rosario Firrincieli, Daniele Lacamera, Stefano Tamagnini,” A DTN Approach to Satellite Communications” IEEE 2007

[37] Dilek Demirel, Fatih Alagoz, M. Ufuk Qaglayan,”IPSEC over Satellite Links: A New Flow Identification Method”, (ISCN' 06) IEEE 2006

[38] N. Thanthry, M. Deshpande, R. Pendse, “A Novel Mechanism for Improving Performance and Security of TCP Flows over Satellite Links”, IEEE 2006

[39] R. C. Durst, G. J. Miller, and E. J. Travis, “TCP extensions for space communications,” Wireless Netw., vol. 3, no. 5, pp. 389–403, Oct. 1997.

[40] R. Wang, V Bandekodige, and M. Banerjee, “An experimental evaluation of link delay impact on throughput performance of TCP and SCPS-TP in space communications,” in Proc. 60th IEEE Veh. Technol. Conf., Los Angeles, CA, Sept. 2004, vol. 6, pp. 4061–4065.

[41] B-ISDN Signalling ATM Adaptation Layer–Service Specific Connection Oriented Protocol, ITU-T Recommendation Q.2110, 1994.

[42] M. E. Elaasar, M Barbeau, E. Kranakis, and Z. Li, “Satellite transport protocol handling bit corruption, handoff and limited connectivity,” IEEE Trans. Aerosp. and Electron. Syst., vol. 41, no. 2, pp. 489–502, Apr. 2005.

[43] I. F. Akyildiz, O. B. Akan, and G. Morabito, “A rate control scheme for adaptive real-time applications in IP networks with lossy links and long RTTs,” IEEE/ACM Trans. Netw., vol. 13, no. 3, pp. 554–567, June. 2005.

[44] D. Katabi, M. Handley, and C. Rohrs, “Congestion control for high bandwidthdelay product networks,” ACM SIGCOMM Comput. Commun. Rev., vol. 32, no. 4, pp. 89–102, Oct. 2002.

[45] K. Zhou, K. L. Yeung, and V. O. K. Li, “P-XCP: a transport layer protocol for satellite IP networks,” in Proc. IEEE GLOBECOM, Dallas, TX, Dec. 2004, vol. 5, pp. 2707–2711.

[46] A. Kapoor, A. Falk, T. Faber, and Y. Pryadkin, “Achieving faster access to satellite link bandwidth,” in Proc. IEEE INFOCOM 2005, Miami, FL, USA, Mar. 2005, vol. 4, pp. 2870–2875.

[47] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, “Stream Control Transmission Protocol,” RFC 2960, Oct. 2000.

[48] S. Fu and M. Atiquzzan, “SCTP: state of the art in research, products, and technical challenges,” IEEE Commun. Mag., vol. 42, no. 4, pp. 64–76, Apr. 2004.

[49] J. Border, M. Kojo, J.Griner, G. Montenegro, Z. Shelby, “Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations,” RFC3135, June 2001.

[50] D. Velenis, D. Kalogeras, and B. Maglaris, “SaTPEP: a TCP performance

enhancing proxy for satellite link," in Networking: Second Int. IFIP-TC6 Netw.
Conf., Lecture Notes in Comput. Science. Springer, Berlin, vol. 2345, pp. 1233–
1238, 2002.

[51] T. Le-Ngoc and J.I. Mohammed, Combined free/demand assignment multiple access (CFDAMA) protocols for packet satellite communication, Proc. IEEE ICUPC, vol. 2, pp. 824-828, Ottawa, Oct. 1993.

[52] P.D Mitchell, T.C. Tozer and D. Grace, Improved medium access control for data traffic via satellite using the CFDAMA protocol, IEE Seminar on Broadband Satellite: The Critical Success Factors Technology, Service and Markets, pp. 18/1-18/7, London, Oct. 2000.

[53] Raffaello Secchi, "TDMA DAMA satellite support". 2006, http://votos.isti.cnr.it/TDMA%20DAMA%20NS2.html

[54] The Network Simulator ns-2, http://www.isi.edu/nsnam/ns.

[55] M. Allman et al., "Ongoing TCP Research Related to Satellite", RFC2760, February 2000.

[56] E. Anderlind and J. Zander, "A traffic model for non-real-time traffic in wireless radio networks", IEEE Comm. Letters, vol. 2, pp. 37 -39, March, 1997.

[57] Aaron Falk, Ted Faber, Eric Coe, Aman Kapoor, Bob Braden, "Experimental Measurements of the explicit control protocol (XCP)", 2nd International Workshop on Protocols for Fast Long-Distance Networks, February 16-17, 2004, Argonne, Illinois USA

[58] Bielli M., Ambrosino G., Marco B. (Eds.): "Artificial Intelligence Applications to Traffic Engineering", Utrecht 1994, The Netherlands.

[59] Cuena J., Hernández J., Molina M., "A Structure of Problem-solving Methods for Real-time Decision Support in Traffic Control", International Journal of Human and Computer Studies, 1998

[60] Hernandez, J.: "Real-time Traffic Management through Knowledge-based Models: The TRYS approach",ERUDIT Tutorial on Intelligent Traffic Management Models (www.erudit.de) , Helsinki 1999, Finland

[61] Clement N. Nyirenda , Dawoud S. Dawoud.:"A Fuzzy Logic Based Dual Explicit Congestion Notification Mechanism for Satellite TCP/IP Networks". EUROCON 2007 The International Conference on "Computer as a Tool" Warsaw, September 9-12

[62] Banupriya Vallamsundar, Jiesheng Zhu, K.Ponnambalam, Changcheng Huang, Anand Srinivasan, Brian Cheng.:"Congestion Control for Adaptive Satellite Communication Systems Using Intelligent Systems". IEEE 2007