International Conference on Communications (IEEE ICC 2011), 05-09 June 2011, Kyoto, Japan.

SatERN: a PEP-less solution for satellite communications

Tuan Tran Thai[1], Dino Martin Lopez Pacheco[2], Emmanuel Lochin[1,3], Fabrice Arnal[4]
[1] Université de Toulouse ; ISAE ; Toulouse, France
[2] I3S Lab - CNRS UMR 6070 University of Nice, Sophia-Antipolis, France
[3] CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France
[4] Thales Alenia Space

***Abstract*—In networks with very large delay like satellite IP-based networks, standard TCP is unable to correctly grab the available resources. To overcome this problem, Performance Enhancing Proxies (PEPs), which break the end-to-end connection and simulate a receiver close enough to the sender, can be placed before the links with large delay. Although splitting PEPs does not modify the transport protocol at the end nodes, they prevent the use of security protocols such as IPsec. In this paper, we propose solutions to replace the use of PEPs named SatERN. This proposal, based on Explicit Rate Notification (ERN) protocols over IP, does not split connections and is compliant with IP-in-IP tunneling solutions. Finally, we show that the SatERN solution achieves high satellite link utilization and fairness of the satellite traffic.**

I. Introduction

In current IP networks, TCP New Reno (denoted standard TCP in the rest of the paper) is the main protocol in charge of providing congestion control, fair share and full utilization of the network resources. Standard TCP provides good performance in terms of link utilization in networks with short propagation delay (only a few ten of milliseconds) and low bandwidth (less than 100Mb/s). However, its performance is poor in large bandwidth$\times$delay product (LBDP) networks such as network with satellite links.

To solve the problem of standard TCP in LBDP, high speed variants have been proposed such as CUBIC TCP [1], Compound TCP [2], Hybla TCP [3] and High Speed TCP [4]. However, it has been shown in [5], [6] that these TCP variants potentially lead to congestion states and intra/inter-protocol unfairness. The intra-protocol and inter-protocol fairness indicate, respectively, the fairness between flows using either the same or different protocols. Others high speed TCP variants, known as delay-based protocols such as FAST TCP [7], consider an increase of the round-trip time (RTT) as a congestion indicator. Thus, they monitor the RTT at the sender side to prevent congestion state. However, delay-based protocols do not solve the problem of intra/inter-fairness [6]. High speed TCP variants and delay-based protocols belong to the class of end-to-end (E2E) protocols since they control the congestion in an end-to-end basic.

Basically, in the case of networks with very large delay due to satellite links, the use of splitting Performance Enhancing Proxies (PEPs) has been proposed to improve the performance of standard TCP. PEPs break the end-to-end connection and simulate a receiver. When a PEP is implemented before the link with large delay, the sending rate is sensibly increased. Furthermore, between the PEP and the receiver, other transport protocols, more aggressive than standard TCP, are often used. One of the main barrier of this architecture is the use of security protocols. In the context of privacy protection such as IPsec and Secure Socket Layer (SSL), PEPs can not be used without introducing complex modifications [8]. In addition, PEPs might require both high memory capacity to keep connection states and complex fault tolerant mechanisms.

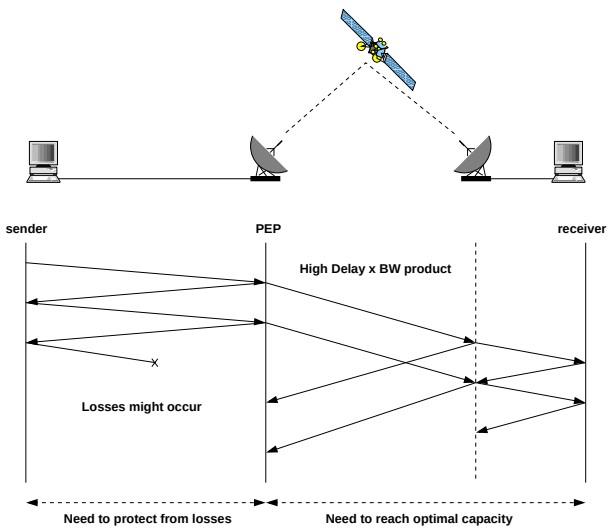
More recently, a new family of protocols known as Explicit Rate Notification (ERN) protocols shows high intra-protocol fairness and performance in terms of link utilization, buffer occupancy in full ERN-capable networks. Full ERN-capable network implies the network in which all routers support ERN capabilities. In the ERN approach, ERN routers inform the sender about the optimal sending rate. However, they do not implement any mechanism to deal with networks where non-ERN protocols (e.g., standard TCP) and non-ERN equipments (e.g., DropTail routers) are present [9]. Although ERN protocols cannot be gradually deployed in heterogeneous networks (e.g., the Internet), this approach and particularly the use of eXplicit Control Protocol (XCP) [10] received a particular attention by the satellite community. Indeed, a satellite topology can be seen as a bounded network where the edges are defined by the PEPs. As an illustration, the authors in [11] propose the use of splitting PEPs which maps TCP flows to XCP flows thus targeting the use of XCP to provide a faster access to satellite links. Some efforts have also been done to assess the benefits and to improve the behavior of XCP in a satellite context. In [12], the authors propose a revisited version of XCP (named P-XCP) especially designed to enhance XCP performance over satellite links and a more recent paper provides a study of TCP over XCP in a satellite context [13].

This motivates our study to propose Satellite-ERN (SatERN) solution which allows senders benefiting from ERN capabilities in non-fully ERN-capable networks. We show that our SatERN solution achieves high performance in terms of link utlization and fairness of flows in satellite IP-based networks (and, in general, in any kind of LBDP networks). The SatERN solution, which does not introduce any complex and heavy mechanisms either at the end hosts or forwarding

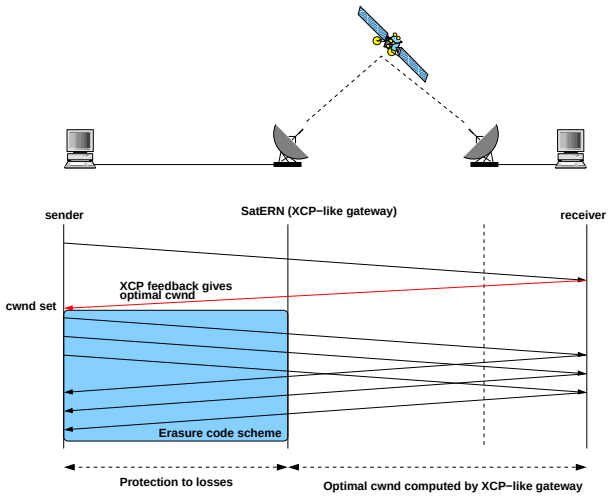
devices, are applicable even in presence of non-ERN protocols (e.g., standard TCP) and equipments. The proposed solution does not split end-to-end connections and thus, is compliant with IP-in-IP tunneling solutions.

This paper is organized as follows. Section II presents the global view of satellite communications at transport level and related issues. Section III describes the rationale of our SatERN solution. In Section IV, we present the simulation results and analysis. We conclude and provide the future work in Section V.

II. The Big Picture



(a) Standard PEP architecture



(b) Proposed SatERN architecture

Fig. 1. PEP versus SatERN architecture

The SatERN proposal introduces a novel gateway scheme that does not break the end-to-end connectivity. Basically, the goal is to offer a transparent connection establishment and data exchange between two Internet hosts connected via a satellite link.

In the context of standard satellite communications, when an Internet host uses a satellite link to access to the Internet, the connection between the Internet and the satellite link is usually realized by a PEP as illustrated in Figure 1(a). The rationale behind is that this host, which generally uses TCP to browse the Internet, is faced with two major issues that directly impact on the overall performance of the standard TCP: the long delay inherent to satellite links and the losses due to congestion or poor quality of transmission (e.g., wireless connection) over the Internet link.

To overcome both problems, the PEP architecture optimizes the transfer to the satellite link by using a transport protocol suited for long-delay links (e.g., SCPS-TP [14], TCP-Hybla) while performing the retransmissions of lost packets only on the Internet link as in clear weather condition, a satellite link is considered as mostly error free (following DVB-S2 standard, $BER \approx 10^{-10}$). However, this architecture is not transparent for the host as the PEP splits the connection. As already explained in the introduction, this splitting prevents the use of a secure protocol such as IPSec; the establishment of an encrypted VPN and more generally any connection using the Secure Socket Layer (e.g., https).

SatERN prevents the use of PEP in order to solve these two major issues discussed above (losses over Internet and high delay link over the satellite link). The challenge is then twofold: we first have to propose a method to efficiently reach the high bandwidth delay product resulting from the satellite link; then, we must avoid as possible losses on the Internet link to prevent end-to-end restransmissions as illustrated in Figure 1(b). We believe that one of the main challenge is first to achieve the satellite capacity while losses can be mitigated with an erasure code mechanism such as Forward Error Correction (FEC) as illustrated in Figure 1(b). Thus, in this paper, we specifically focus on the method to achieve the capacity over the satellite link.

We propose to use XCP as ERN protocol and particularly the XCP signalling mechanism to announce the optimal TCP window size of the satellite link to the source. The optimal window size is computed in the SatERN gateway (which acts as an XCP-like router) then put inside the feedback messages. The sender retrieves this information and takes the minimum between the congestion window given by the ERN feedback and the actual TCP congestion window (see next Section III for further details). This architecture does not need a full deployment of XCP routers on the entire path as the satellite link is the only one subject to XCP congestion window computation. We have designed an adapted version of the XCP transport protocol to emulate at the sender side an XCP-like service that would only respond to XCP-like gateways connected to satellite links. This service XCP would allow a host to transparently use an Internet connection like DSL or satellite without having to save a specific configuration for each connection context.

SatERN has been developed and evaluated using ns-2 simulator. An implementation proposal of the core ERN protocol with exchange messages specification is detailed in [15]. The

resulting SatERN framework allows to use different TCP flavors (e.g., FAST TCP, CUBIC TCP). In the following we drive experiments with different TCP variants to assess the benefits brought by our proposal.

III. Rationale of SatERN proposal

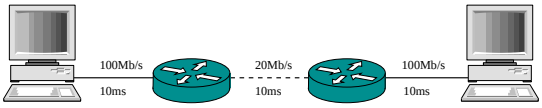
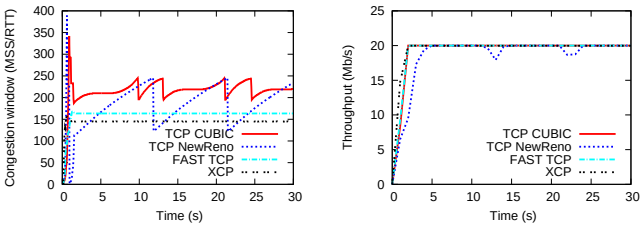


Fig. 2. Topology used to present the rationale of the idea



(a) Congestion window (b) Throughput

Fig. 3. Comparison between E2E and ERN protocols

Before presenting our evaluations with satellite links scenario, we drive a simple experiment to assess the rationale of the idea.

Delay-based TCP versions (TCP Vegas [16], FAST TCP [7]) try to keep a certain number of packets in the network pipe, mostly in the bottleneck, when the connection remains stable. For instance, FAST TCP tries to keep between α and β packets in the network. While standard TCP and its high speed variants frequently push the network to congestion state by increasing the TCP sending window size (congestion window or cwnd) even though the network capacity has been reached. Congestion window, which is interpreted by the sending rate, indicates the number of packets sent without acknowledgement. On the other hand, ERN protocols use the optimal congestion window by minimizing the buffer utilization since they have the explicit feedback from the ERN routers.

Let us show a simple simulation with one ERN protocol (XCP) and different E2E protocols (TCP New Reno, CUBIC TCP, FAST TCP) in short BDP network as depicted in Figure 2. The router queue is XCP in case of XCP and DropTail for other cases. In Figure 3(b), both protocols achieve the bottleneck capacity of 20Mb/s. Specifically, XCP uses the smallest congestion window (Figure 3(a)). This means that XCP maximizes the link utilization while minimizing the buffer occupancy. Indeed, when the TCP New Reno's congestion window is smaller than the XCP's congestion window, TCP New Reno's throughput is slightly lower than 20Mb/s. On the other hand, when the XCP sender receives misleading congestion window in non-fully XCP-capable network, XCP's congestion window might be higher than E2E's congestion window but XCP's goodput is lower than E2E's [9].

These observations arise an idea of mixed E2E-ERN protocols which use the minimum value between E2E's cwnd and ERN's cwnd. Our SatERN solution uses mixed E2E-ERN protocols in conjunction with an ERN satellite gateway to replace the use of PEP. To validate our SatERN proposal, we choose CUBIC TCP as an E2E protocol since it is used by default in several Linux distributions and XCP as ERN protocol.

In the following, we briefly describe the way CUBIC-XCP works with XCP satellite gateway. The sender updates the XCP header before sending data to the network. Upon reception of the data packet, XCP satellite gateway computes the feedback based on the XCP header and updates the XCP header according to XCP algorithm. XCP satellite gateway does not update the header in ACK packets. When the receiver receives the data, it sends back an ACK with the header contained in the data packet. Upon reception of the ACK, the sender calculates the congestion window according to CUBIC algorithm (*cwnd_cubic*) and the congestion window according to XCP algorithm (*cwnd_xcp*). Then, the sender uses the minimum value according to the equation (1). This routine is executed until the connection is closed.

$$cwnd = min\{cwnd\_cubic, cwnd\_xcp\} \quad (1)$$

IV. Results and analysis

Our simulations use the ns-2 network simulator. The network topology which simulates the network with satellite link (Figure 4) has a base RTT of 640ms and a satellite capacity of 1Mb/s.The buffer size of satellite gateway is fixed to 20 packets and the packet size is 1000 bytes.

We propose to demonstrate the capability of the proposed approach within three scenarios. The first one verifies that TCP variants used with our SatERN framework behave like XCP protocol when possible and are able to correctly grab the available bandwidth. The second one tackles the intra-fairness of flows in order to assess whether arriving supplementary flows do not disturb the previous ones. Then, we evaluate the inter-fairness between flows when non-XCP flows share the link capacity with XCP flows. Finally, we evaluate our solution over a dynamic network.

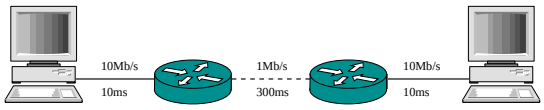
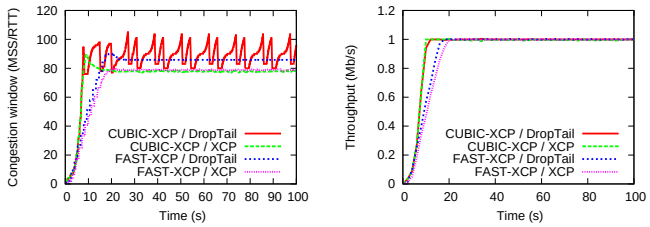


Fig. 4. Satellite network topology

A. Correctness of the SatERN solution

We have implemented the SatERN solution in ns-2 with CUBIC TCP and FAST TCP protocols. With the network settings as in Figure 4, we perform four different experiments. In the first and second experiments, we run a CUBIC-XCP flow when bottleneck router is, respectively, XCP and DropTail. For the third and fourth experiments, FAST-XCP

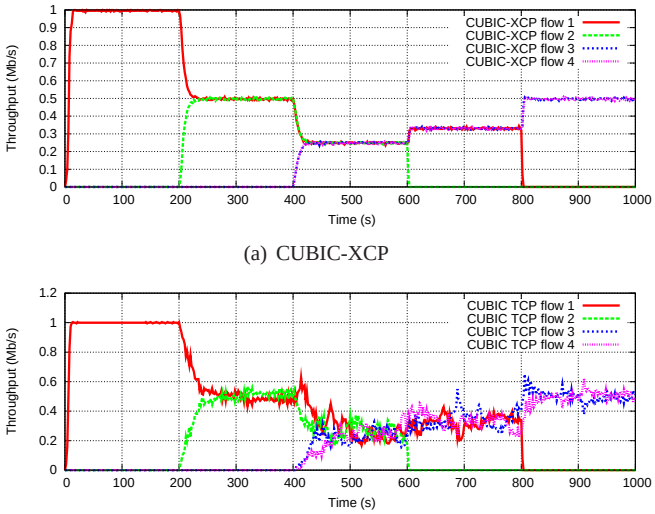


(a) Congestion window (b) Throughput

Fig. 5. CUBIC-XCP and FAST-XCP with DropTail and XCP queue

flow is sent when the bottleneck router is, respectively, XCP and DropTail. By taking the minimum value between E2E and ERN congestion windows, we switch between ERN or E2E behaviour depending on the network conditions. As a result, CUBIC-XCP and FAST-XCP can be considered as an E2E-ERN protocols which behave like ERN in the possible cases where the bottleneck router is ERN-capable and only ERN flows are present in the bottleneck. Otherwise, E2E-ERN protocols use their E2E capability to compete against other flows. The results in Figure 5 show that FAST-XCP and CUBIC-XCP protocols behave like FAST TCP and CUBIC TCP, respectively, in case the bottleneck router is DropTail. They act as XCP protocol in case the bottleneck router is XCP. This simulation shows that when the sender receives the misleading ERN information, it uses its E2E capability for the connection where the misleading information in pure ERN protocol causes poor performance [9]. As CUBIC TCP is currently enabled by default in Linux, we only present simulation results with CUBIC-XCP from now.

B. Intra-fairness



(a) CUBIC-XCP

(b) CUBIC TCP

Fig. 6. Intra-fairness

The aim of this simulation is to show the intra-fairness of CUBIC-XCP protocol. In the experiment, four CUBIC-XCP flows start and terminate at different time. In Figure 6(a), when new flows enter or old flows leave the bottleneck, the remaining CUBIC-XCP flows quickly converge to the new fairness line and are stable since then. In the presence of only CUBIC-XCP flows and the satellite gateway with XCP capabilities, CUBIC-XCP behaves like XCP protocol which provides good intra-fairness property as shown in [10]. For comparison purpose, Figure 6(b) gives the result obtained without SatERN and clearly highlights both the slow convergence of CUBIC TCP flows and their oscillating behavior.

C. Inter-fairness

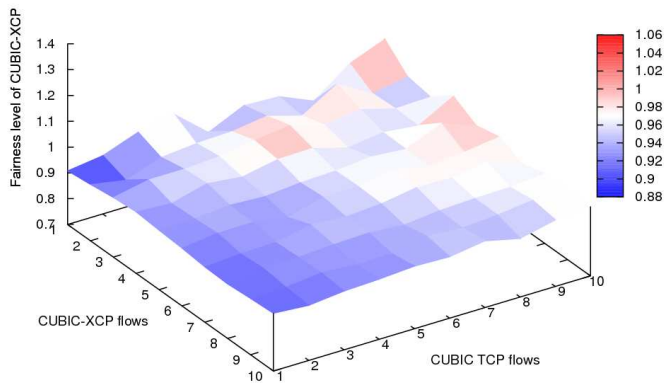


Fig. 7. Inter-fairness between CUBIC-XCP and CUBIC TCP

In order to show the inter-fairness, we perform 100 experiments. We remark that the observed link utilization is 100% for all experiments). In each experiment, we let CUBIC-XCP flows (ranged from 1 to 10) compete against CUBIC TCP flows (ranged from 1 to 10) with duration of 1000 seconds. CUBIC-XCP and CUBIC TCP flows start at the same time with the same conditions. We define the fairness level of CUBIC-XCP as a ratio beween the actual aggregate throughput and the expected aggregate throughput of CUBIC-XCP flows. For instance, in the experiment with four CUBIC-XCP versus eight CUBIC TCP flows, the expected aggregate throughput of CUBIC-XCP is $4/(4+8) = 0.33$ of the link utilization. The actual aggregate throughput of CUBIC-XCP flows is the averaged aggregate throughput observed during simulation of 1000 seconds. The fairness level, which is greater than one, implies that CUBIC-XCP flows take more resources than CUBIC TCP flows in a fairness point of view and vice versa. The fairness increases when the fairness level is close to 1 and vice versa. With this definition, the fairness level ranged from 0.9 to 1.1 shows good fairness and specifically, the range from 0.95 to 1.05 shows very good fairness. In Figure 7, most of experiments have the fairness level greater than 0.9. Specifically, in Table I, 52 experiments have the fairness level between [0.95-1.05] and the fairness level between [0.9-0.95] and [1.05-1.1] has 45 experiments. Our experiments show that CUBIC-XCP and CUBIC TCP achieve good fairness level regardless of number of flows.

TABLE I
STATISTIC OF FAIRNESS LEVEL

Fairness level	< 0.90 or > 1.10	0.90 - 0.95 and 1.05 - 1.10	0.95 - 1.05
# of experiments	3	45	52

D. Dynamic scenario

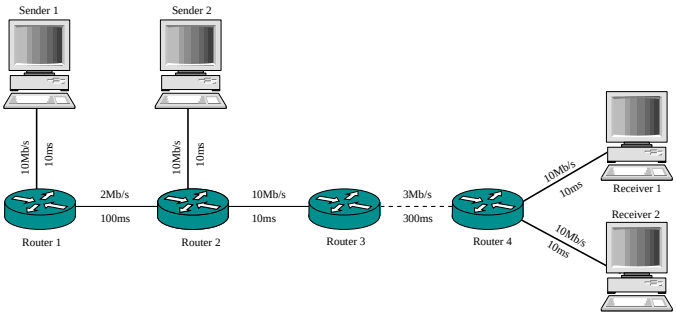
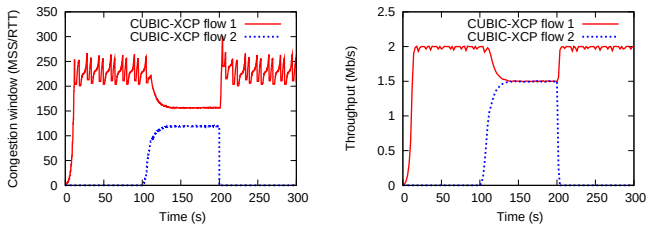


Fig. 8. Network topology for dynamic scenario



(a) Congestion window (b) Throughput

Fig. 9. Dynamic property for CUBIC-XCP

The aim of this simulation is to show the dynamic property of CUBIC-XCP using the network topology in Figure 8.The base RTT is 860ms for flow 1 between Sender 1 and Receiver 1 and 660ms for flow 2 between Sender 2 and Receiver 2. Queue type of all routers is XCP except Router 1 which is DropTail. CUBIC-XCP flow 1 starts at 0s, the bottleneck at this time is in Router 1. As shown in Figure 9, CUBIC-XCP flow takes link capacity of Router 1. When CUBIC-XCP flow 2 starts from Sender 2 at 100s, the bottleneck is now moved to Router 3. Since Router 3 is XCP, CUBIC-XCP flow 1 switches now to XCP mode. Both flows behave like XCP and fairly share 3Mb/s. It is noted that flow with larger RTT (CUBIC-XCP flow 1) is not penalized since CUBIC-XCP inherits good intra-fairness property of XCP. At 200s, CUBIC-XCP flow 2 stops, the bottleneck is now moved back to Router 1 and CUBIC-XCP flow 1 switches back to CUBIC mode. It has been shown in this simulation that the CUBIC-XCP automatically switches between CUBIC TCP or XCP modes depending on the network condition. This switch is transparently performed by the minimum comparison without any explicit notification mechanism.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced a solution to prevent the use of PEP in the context of satellite communications. The presented solution, named SatERN, allows to efficiently grab the available resource of the satellite link without splitting the end-to-end connection. Furthermore, we have also demonstrated that our proposal allows E2E-ERN protocols intra-fairness, inter-fairness and does not require fully ERN-capable networks. This preliminary evaluation emphasizes that the SatERN solution is a potential alternative to satellite communication. In a future work, we expect to drive real experimentations with a Linux prototype and compare the performance obtained against various PEPs solutions.

VI. ACKNOWLEDGEMENTS

This research work has been supported by funding from Thales Alenia Space (TAS).

REFERENCES

[1] I. Rhee and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," in *International Workshop on Protocols for Fast Long-Distance Networks*, February 2005.
[2] K. Tan, J. Song, Q. Zhang, and M. Shridaran, "Compound TCP: An Scalable and TCP-Friendly Congestion Control for High-Speed Networks." in *IEEE Infocom*, Barcelona, Spain, April 2006.
[3] C. Caini and R. Firrincieli, "TCP hybla: a TCP enhancement for heterogeneous networks," *International Journal of Satellite Communications and Networking*, vol. 22, 2004.
[4] S. Floyd, "HighSpeed TCP for Large Congestion Windows," RFC 3649 (Experimental), Dec. 2003.
[5] J. Lee and S. B. et all, "A study of tcp fairness in high-speed networks," University of California, Santa Barbara, Tech. Rep., apr 2005.
[6] S. Ha, Y. Kim, L. Le, I. Rhee, and L. Xu, "A step toward realistic performance evaluation of high-speed tcp variants," in *Elsevier Computer Networks (COMNET) Journal, Special issue on*, 2006.
[7] D. X. W. Cheng Jin and S. H. Low, "FAST TCP: Motivation, Architecture, Algortihms, Performance," in *INFOCOM*. IEEE, March 2004.
[8] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations," RFC 3135 (Informational), Jun. 2001.
[9] D. Lopez Pacheco, C. Pham, and L. Lefèvre, "Xcp-i : eXplicit Control Protocol for heterogeneous inter-networking of high-speed networks," in *Globecom 2006*, San Francisco, California, USA, nov 2006, pp. 1 – 6.
[10] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," in *ACM SIGCOMM*, 2002.
[11] A. Kapoor, A. Falk, T. Faber, and Y. Pryadkin, "Achieving Faster Access to Satellite Link Bandwidth," in *INFOCOM*, 2006.
[12] Kaiyu Zhou and Yeung, K.L. and Li, V.O.K., "P-XCP: a transport layer protocol for satellite IP networks," *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, vol. 5, Nov 2004.
[13] D. M. L. Pacheco and E. Lochin, "Optimal Configuration for Satellite PEPs using a Reliable Service on Top of a Routers-Assisted Approach," in *International Workshop on Satellite and Space Communications*, Siena-Tuscany, Italy, Sep. 2009.
[14] C. 714.0-B-2, "Consultative committee for space communications, recommendation for space data system standards, space communications protocol sspecification transport protocol (scps-tp)," Oct. 2006, tech. Rep.
[15] D. M. L. Pacheco, T. T. Thai, E. Lochin, and F. Arnal, "Towards an incremental deployment of ERN protocols: a proposal for an E2E-ERN hybrid protocol," in *PFLDNet*, Lancaster, Pennsylvania, Nov. 2010.
[16] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "Tcp vegas: New techniques for congestion detection and avoidance," *ACM SIGCOMM Computer Communication Review*, pp. 24–35, 1994.