

# Optimal Configuration for Satellite PEPs using a Reliable Service on Top of a Routers-Assisted Approach

Dino Martin Lopez-Pacheco<sup>1,2</sup> and Emmanuel Lochin<sup>1,2</sup>

<sup>1</sup> CNRS ; LAAS ;

7 avenue du colonel Roche, F-31077 Toulouse, France

<sup>2</sup> Université de Toulouse ;

UPS, INSA, INP, ISAE ; LAAS ;

F-31077 Toulouse, France

firstname.lastname@isae.fr

**Abstract**—Routers-assisted congestion control protocols, also known as Explicit Rate Notification (ERN) protocols, implement complex algorithms inside a router in order to provide both high link utilization and high fairness. Thus, routers-assisted approaches overcome most of the end-to-end protocols problems in large bandwidth-delay product networks. Today, routers-assisted protocols cannot be deployed in heterogeneous networks (e.g., Internet) due to their non-compliance with current network protocols. Nevertheless, these approaches can be deployed in satellite networks in the context of splitting PEPs. In this work, as routers-assisted protocols can use TCP algorithms to enable reliability, we aim at understanding and providing a detailed view of the impact of such algorithms on the performance obtained by routers-assisted protocols over satellite links. In particular, we both study XCP and P-XCP proposals over long delay, lossy and asymmetric links and propose a ns-2 implementation of the P-XCP protocol to the satellite community. To the best of our knowledge, this study is the first one which tackles the impact of TCP internal mechanisms on top of XCP protocol. Our main conclusion is that TCP New Reno Slow But Steady variant on top of P-XCP is to date, the most optimal configuration for satellite proxies.

## I. INTRODUCTION

End-to-End (E2E) protocols implement algorithms to control congestion inside the end hosts (i.e. at the sender side and at the receiver side). The complete independence of E2E protocols to the network infrastructure allow them to be incrementally deployed in any kind of networks, like Internet. The commonly used E2E protocols is TCP NewReno [1] as it provides the best performance in terms of link utilization, fairness and congestion control in networks with low bandwidth (lower or equal than 100Mbps) and low latency (RTTs smaller than 500ms).

However, in scenarios with large bandwidth and/or large delay, TCP and general E2E approaches are not able to efficiently control congestion in order to correctly share in a fair manner the available resources [2].

The main problem is that E2E approaches cannot exactly assess the congestion level of core networks. Thus, a new

strategy, where both congestion and fairness algorithms are placed in routers, have been proposed. In this novel approach, routers provide to senders the optimal rate. For this reason, they are known as Explicit Rate Notification (ERN) protocols. Among several proposals, some of the most promising ERN proposals are XCP [3], JetMax [4], QuickStart [5], PIQI [6].

Theoretical as well as experimental studies have shown that ERN protocols provide high performances in a wide range of network configurations [7], [8]. However, it has also been proved that ERN protocols are not compliant with current network protocols, limiting their spectrum of use to experimental and private networks [9], [10].

Thus, in order to provide a faster access to satellite links, the authors in [11] proposed the use of splitting PEPs which map TCP flows to XCP flows thus targeting the use of XCP to the satellite or highest delay link.

Some efforts have been done to assess the benefits and improve the behavior of XCP in a satellite context. Indeed, the authors in [12] propose a revisited version of XCP (named P-XCP) especially designed to enhance XCP performances over satellite links and a recent paper provides a preliminary study of XCP in a geostationary context [13].

As the use of satellites in IP networks is progressively increasing and following the interest of big companies such as Google in the deployment of satellite topology<sup>1</sup>, this paper is motivated by studying the advantage that might bring ERN protocols and aims at carefully analyzing the behavior of XCP and P-XCP in a satellite context. In particular, the purpose of this study is to provide to satellite vendors and operators a better view of the benefits and shortcomings of running an ERN protocol.

In this paper, we firstly introduce XCP and our implementation of the P-XCP protocol<sup>2</sup> based on the study in

<sup>1</sup>See "Google goes after 3 billion with super satellite": [http://www.theregister.co.uk/2008/09/09/other\\_three\\_billion/](http://www.theregister.co.uk/2008/09/09/other_three_billion/)

<sup>2</sup>This code is available for download at <http://dmi.ensica.fr/~dlopez/xcp-sat/pxcp.html>

[12]. We then propose to focus on the impact of TCP's Fast Retransmit/Fast Recovery algorithm through simulation experiments and study both variants of TCP New Reno namely Impatient and Slow But Steady. To the best of our knowledge, no study have specifically tackled the impact of internal TCP mechanisms on XCP. We detail the performance obtained by both protocols and conclude that in a context where losses due to factors others than congestion occur (typically the case over a network fully XCP), P-XCP on top of TCP New Reno, using the Slow-but-Steady variant is the best setting. We finally present possible extension and measurements.

## II. GENERAL DESCRIPTION OF XCP AND P-XCP

This section aims at giving to the reader XCP and P-XCP basic principles. We recall that XCP is not a stand-alone protocol and is used on top of other transport protocols such as TCP, DCCP, UDP. Indeed, XCP can be seen as a signalling protocol allowing to assess the congestion level of the network in order to help transport protocols to efficiently fills the available bandwidth [3].

### A. XCP

XCP [3] (eXplicit Control Protocol) uses router-assistance schemes to accurately inform the sender of the congestion network level. In order to realize this task, XCP data packets carry a congestion header, filled in by the source, that contains the sender's current congestion window size ( $H\_cwnd$ ), the estimated RTT and a feedback field  $H\_feedback$ . The  $H\_feedback$  field is the only one which could be modified at every hop (XCP router) based on the value of the two previous fields. Basically, the  $H\_feedback$  field can take positive or negative values representing the amount by which the sender's congestion window size is increased or decreased.

On reception of data packets, the receiver sends back to the source the congestion header (modified accordingly by the routers) into ACK packets. On the reception of ACK packets, the sender would update its congestion window size as follows:

$$cwnd = \max(cwnd + H\_feedback, packetsize)$$

with  $cwnd$  expressed in bytes. The core mechanism resides in XCP routers that use an efficiency controller (EC) and a fairness controller (FC) to update the value of the feedback field over the average RTT which is the control interval. The EC has the responsibility of maximizing link utilization while minimizing packet drop rate. The EC basically assigns a feedback value  $\phi$ , proportional to the spare bandwidth  $S$ , deducted from the difference monitored between the input traffic rate and the output link capacity and the persistent queue size  $Q$ .

The authors in [3] proposes the following EC equation:

$$\phi = \alpha.rtt.S - \beta.Q$$

with  $\alpha = 0.4$  and  $\beta = 0.226$ . Then the FC translates this feedback value, which could be considered as an aggregated increase/decrease value, into feedback for individual packets (put in the data packet's congestion header) following fairness

rules similar to the TCP AIMD principles. However, this feedback value is decoupled from drops because only the difference between input traffic rate and output link capacity ( $S$ ) is used instead in the EC.

The original XCP proposition did not mention any mechanism for handling severe congestion situations as it was assumed that such situations should not occur with the XCP kind of control laws. However, some work have shown that severe congestion do happen and that it is desirable to keep the TCP mechanism [7], [8].

### B. P-XCP

In a satellite context, XCP has two main weaknesses :

The first one is due to the fact that XCP inherits the TCP Fast Retransmit/Fast Recovery algorithms. Therefore, in presence of losses, XCP halves its congestion windows size. In a satellite context, if we consider XCP to be the only congestion control protocol used, losses will be only caused by the medium and not by congestion events (recall the main goal of XCP is to suppress losses due to congestion by computing the optimal emission rate). We agree that current satellites implement high reliable layer 2 protocols that offer BER smaller than  $10^{-7}$ . However, we can note that in some areas with severe weather conditions such as in tropical zones, satellites might experience BER higher than 1% [14]. The second weakness lies on the fact that the Fairness Controller can under-utilize the bandwidth when rate limited XCP flows and non rate limited XCP flows share the same link.

P-XCP [12], the XCP version for satellite networks, tries to solve both problems. For the first one, P-XCP proposes to suppress the Fast Recovery algorithm from XCP. Therefore in case of losses, XCP is still able to resend data at the reception of 3 duplicate acknowledgements (DUPACK) without halving its congestion window. This allows to keep the same rate during loss events. Concerning the second problem, P-XCP introduces a new equation to compute the number of rate limited and non rate limited flows. In this way, P-XCP operates to a redistribution of the bandwidth between non rate limited flows only.

## III. LINK UTILIZATION IN HIGH ASYMMETRIC CONFIGURATIONS

In order to test XCP and P-XCP in this scenario, we propose to observe the sharing behavior of two flows over a 100Mbps link capacity. One of them (Flow 0) is limited by a bottleneck of 0.5Mbps placed elsewhere, while the other one (Flow 1) can reach the full link capacity (i.e. 100Mbps). When XCP is used, Flow 0 grabbed all the available bandwidth while Flow 1 only grabbed around 90% of the bottleneck available bandwidth. However with the modifications proposed by P-XCP, Flow 1 obtains more than 95% of the available bandwidth.

From this experiment, we can conclude that P-XCP increases link utilization in high asymmetric networks. However, P-XCP is not able to provide a full link utilization. However, we note that in this case, the available bandwidth for Flow 0 represents around 0.5% of the available bandwidth for Flow

1. We believe we cannot compare this configuration with a real network scenario because of the presence of multiple concurrent flows which avoid this kind of disparity. Indeed, since we are considering XCP between splitting PEPs, the network configuration used here remains exceptional. However, this highlights the potential increase of performances obtained by P-XCP.

#### IV. PERFORMANCE OF XCP AND P-XCP IN LOSSY LINKS

This section compares the performances of XCP and P-XCP in presence of losses due to bit errors. Figure 1 presents the testbed topology used. Since we only focus on the fraction of the network bordered by XCP PEPs, the propagation delay observed by XCP or P-XCP flows in both forward and reverse directions are 250ms or 500ms. To simulate BERs, we introduced a PLR of 0.1% with a uniform distribution.

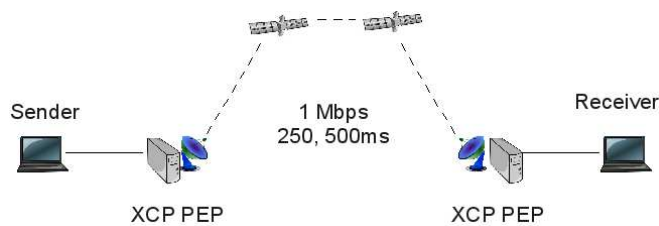


Fig. 1. Satellite network isolated with XCP PEPs.

If the original XCP is used, when losses occur the throughput decreases due to the Fast Recovery action (Figure 2). Later, a fast increase of the throughput can be observed. Since the propagation delay in this case is set to 250ms, we claim that any E2E protocol would improve the performance of XCP. In addition, E2E protocols will also suffer of congestion they might produce. On the other side, when Fast Recovery is not used, as advised by P-XCP, most of the time the throughput evolution remains stable when losses occur (Figure 3). However, it can be observed that before second 60, P-XCP also decreases the rate. Indeed, logs show that P-XCP suffers of retransmission timer expiration (timeout).

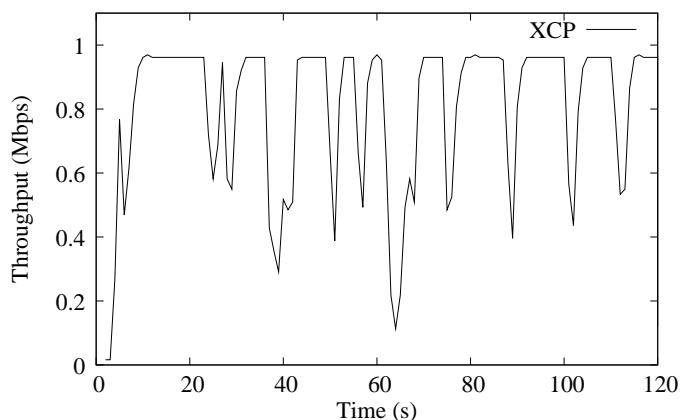


Fig. 2. XCP in presence of a 0.1% LPR

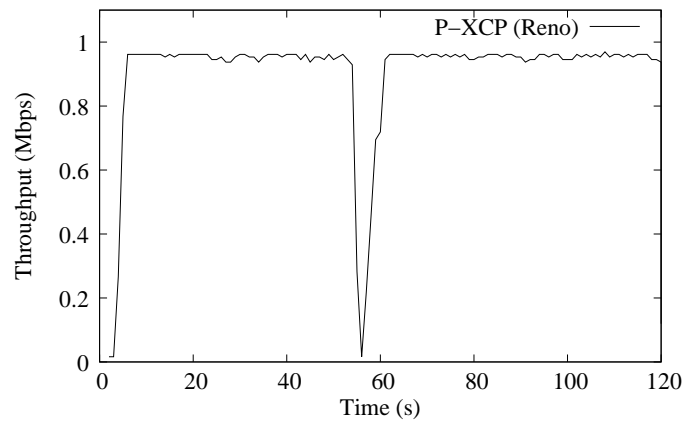


Fig. 3. P-XCP in presence of a 0.1% LPR

In order to correctly recover losses of packets at the reception of 3 DUPACKs, P-XCP must keep the Fast Recovery algorithms inherited from TCP. Since in this case P-XCP (like XCP) is implemented on the top of TCP Reno [15], the Fast Retransmit algorithm used in this case is the one from that TCP version.

Moreover, it has been already proved that TCP Reno is unable to recover from multiple packet losses belonging to a single congestion window. This problem is inherited from TCP Reno to P-XCP / XCP. Thus, one lost packet followed by a couple of packet losses leads to a timeout.

#### V. IMPACT OF TCP NEW RENO MECHANISMS OVER P-XCP

This section studies two TCP New Reno variant namely Impatient TCP and Slow But Steady TCP (SBS TCP). We choose to study both because the impatient variant is enabled by default and we would expect to avoid timeouts in presence of multiple packet losses with the SBS variant.

##### A. XCP/P-XCP over Impatient TCP

In order to avoid Timeout originated by the lack of mechanisms in TCP Reno to recover from multiple losses, we have implemented P-XCP on top of TCP New Reno [1]. In the first simulation result presented in this section, we used the Fast Retransmit Impatient variant. The throughput and congestion window evolution are graphically represented in Figures 4 and 5.

As it can be observed in these figures, a retransmit timeout occurs before second 60 in a similar way than TCP Reno. This timeout is caused by 3 dropped packets in slightly more than 1 RTT interval. Therefore, compared to TCP Reno, TCP New Reno in its Impatient variant does not provide significant improvements to P-XCP. The reason of the timeout suffered by Impatient TCP is that when the RTO is not much larger than the RTT<sup>3</sup>, a timeout can occur when there is a small number of packet dropped [1].

<sup>3</sup>This is the case when XCP is used, since the experienced RTT by the senders is close to the End-to-End propagation delay.

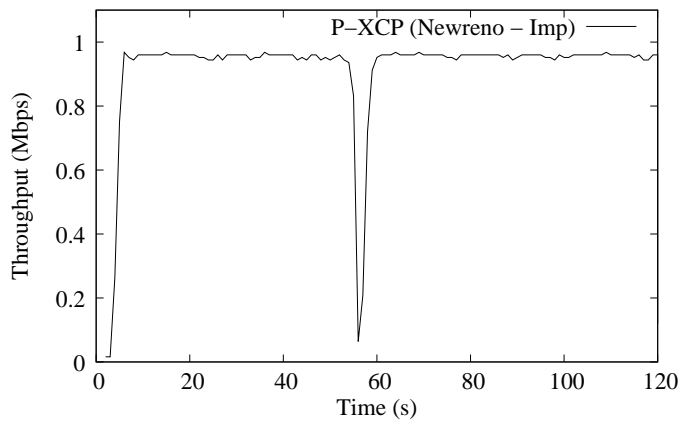


Fig. 4. Throughput evolution of P-XCP using Impatient TCP.

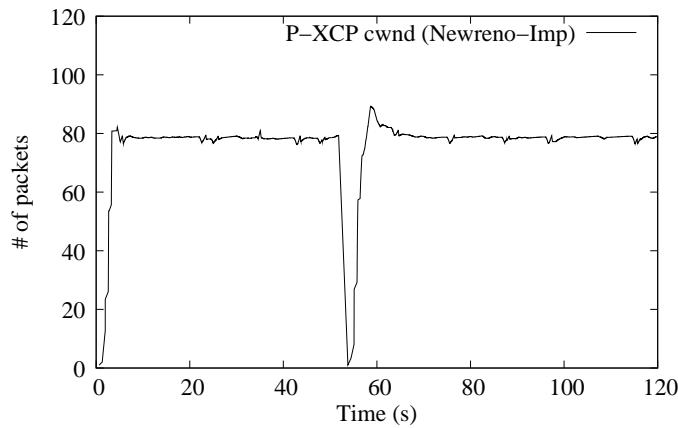


Fig. 5. Congestion window evolution of P-XCP using Impatient TCP.

### B. XCP/P-XCP over Slow-but-Steady TCP

Using the Slow-but-Steady variant of TCP New Reno would allow P-XCP to avoid timeouts in presence of multiple packet losses. Therefore, we made the same simulation previously presented except we use P-XCP over Slow-but-Steady TCP. The results of this simulation, given in Figures 6 and 7, show respectively the throughput and the congestion window evolution.

First, in Figure 6, we can observe that Slow-but-Steady TCP allows a faster recovery of the P-XCP rate in presence of multiple packet losses. Also, Slow-but-Steady allows P-XCP to keep a high link utilization during the Fast Retransmit phase of TCP even though only one lost packet per RTT is recovered. This phenomenon is a result of the inflating congestion window executed by TCP when additional duplicated ACKs (DUPACKs) are received.

However, when a packet is retransmitted and a partial ACK is received, Fast Retransmit deflates the congestion window and then sends the required data packet. Since the sliding congestion window will be limited by the partial ACK, P-XCP will be unable to send as much as needed packets to fully utilize the available bandwidth. This phenomenon leads

to a decrease of the throughput that can be observed in Figure 6.

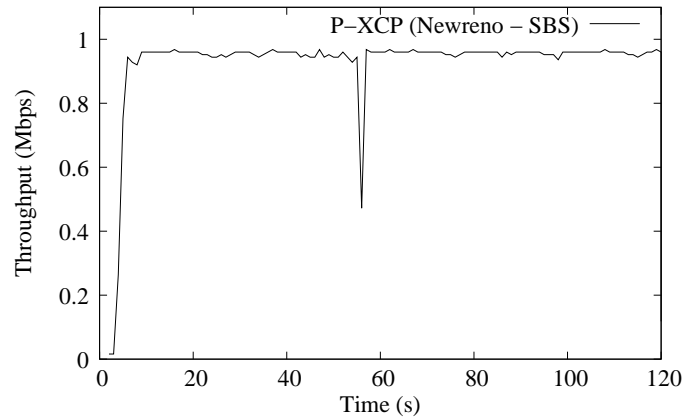


Fig. 6. Throughput evolution of P-XCP using Slow-but-Steady TCP.

Taking a look at the congestion window evolution (Figure 7), we can see that during the Fast Retransmit phase executed before second 60, the congestion window of P-XCP increases from 80 MSS to around 120 MSS. This phenomenon results from a “wrong view” of the number of active flows present in the network by the XCP routers.

Indeed, in order to correctly provide high link utilization and fairness, XCP routers estimate the number of active flows  $N$  during the control interval time as follows:

$$N = \sum_{\text{pkts in } T} \frac{1}{T \times (cwnd_{pkt} / RTT_{pkt})} \quad (1)$$

where  $RTT_{pkt}$  and  $cwnd_{pkt}$  are the current RTT and the congestion window from the XCP packet header. From this equation, it can be deduced that when the congestion window is greater than the number of packets seen by the router, such a router will underestimate the number of active flows. This underestimation will increment the estimated per packet feedback. For instance, the congestion window size needed by one sender to grab all the available bandwidth is smaller than the one needed by 0.7 active flows.

After packets retransmissions, when the congestion windows can slide and inject into the network a number of packets equivalent to the congestion window, XCP routers send negative feedbacks to avoid congestion.

Following these experiments, we can conclude that Slow-but-Steady TCP New Reno can improve the performance of P-XCP / XCP in presence of losses due to factors (e.g. error bits) other than congestion. The authors in [16] show similar results when TCP New Reno is used and less than 20 packets per congestion window are lost.

It is important to note that, even though our conclusions are derived from simulation results with only one XCP flow, we also made other simulations with several competing flows asynchronously incoming and leaving the network and have obtained similar results that have confirmed our observations.

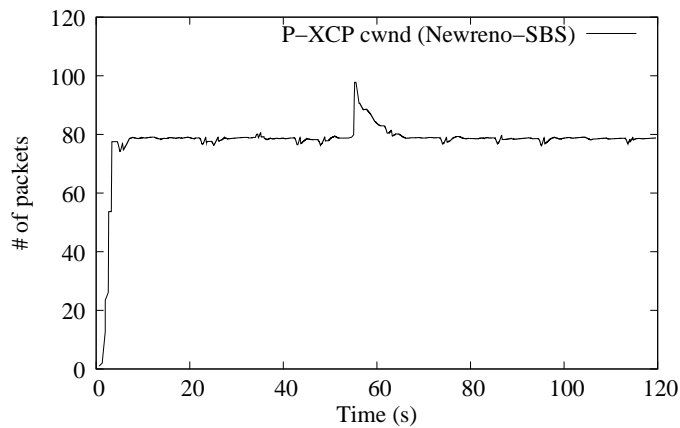


Fig. 7. Congestion window evolution of P-XCP using Slow-but-Steady TCP.

For a sake of simplicity, we then choose to present results obtained by one flow.

## VI. DISCUSSION

In this document, we presented some of our observations linked to the impact of TCP algorithms on P-XCP. So far, we have seen that only suppressing Fast Recovery to avoid throughput variation in presence of losses due to the medium is not enough when Router-Assisted protocols are implemented on top of TCP Reno. Indeed, small amount of dropped packets potentially lead to a retransmission timer expiration.

Even though TCP New Reno has been proposed to recover from multiple packet loss in a single window congestion, building P-XCP on top of TCP New Reno does not guarantee any flows' stability. In fact, when the jitter is low, which is the case of XCP in fully XCP networks, and the Impatient variant is used, a few dropped packets may lead to a similar behavior of P-XCP when used on top of TCP Reno.

However, the Slow-but-Steady variant offers better stability to P-XCP when losses occur. The only drawback we have identified is that when the senders are not able to send as many packets as reported to the routers (due to a freeze of the congestion window resulting from data losses), those routers will inflate senders congestion window. Thus, after retransmitting every needed data packets, next burstiness will potentially fully saturate the network leading to a severe congestion problems. However, in our experiments, routers were always able to correct the rate of senders before the presence of losses due to congestion. Also, we never could completely synchronize concurrent flows in order to produce burst of packets at the same time from several flows. Therefore, we encourage the use of this TCP variant to be used as support for Router-Assisted protocols, like XCP / P-XCP.

## VII. CONCLUSION

This paper provides performance measurements of ERN protocols conjointly with internal TCP mechanisms in order to use a routers-assisted approach in a proxy satellite context. Following this study and the final discussion of the previous

section, we claim that P-XCP with TCP New Reno with Slow But Steady variant (meaning that Impatient variant must be disabled) is the optimal configuration for a satellite PEP. Following our conclusion, the logical next step is the port of our P-XCP version inside the XCP FreeBSD kernel and the experimentation in a real context.

## VIII. ACKNOWLEDGEMENTS

The authors would like to thank (without specific order) Fabrice Arnal, Isabelle Buret, Fabrice Hobaya and Julien Fasson concerning discussions about these mechanisms.

## REFERENCES

- [1] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 2582 (Experimental), April 1999, obsoleted by RFC 3782. [Online]. Available: <http://www.ietf.org/rfc/rfc2582.txt>
- [2] R. Droms, "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3646 (Proposed Standard), Dec. 2003. [Online]. Available: <http://www.ietf.org/rfc/rfc3646.txt>
- [3] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," in *ACM SIGCOMM*, 2002.
- [4] D. L. Yueping Zhang and D. Loguinov, "JetMax: Scalable Max-Min Congestion Control for High-Speed Heterogeneous Networks," in *INFOCOM*, April 2006.
- [5] S. Floyd, M. Allman, A. Jain, and P. Sarolahti, "Quick-Start for TCP and IP," RFC 4782 (Experimental), January 2007. [Online]. Available: <http://www.ietf.org/rfc/rfc4782.txt>
- [6] I. Dukkupati and N. Mckeown, "Rcp-ac: Congestion control to make flows complete quickly in any environment," in *Proc. High-Speed Networking Workshop: The Terabits Challenge, IEEE INFOCOM*, 2006.
- [7] Y. Zhang and T. R. Henderson, "An Implementation and Experimental Study of the eXplicit Control Protocol (XCP)," in *INFOCOM*, 2005, pp. 1037–1048.
- [8] S. H. Low, L. L. H. Andrew, and B. P. Wyrowski, "Understanding XCP: Equilibrium and Fairness," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005, pp. 1025–1036. [Online]. Available: <http://netlab.caltech.edu/publications/XCP-infocom05.pdf>
- [9] D. M. Lopez Pacheco, C.-D. Pham, and L. Lefevre, "XCP-i : eXplicit Control Protocol for Heterogeneous Inter-Networking of High-Speed Networks," in *Globecom 2006*, San Francisco, California, USA, November 2006.
- [10] D. M. Lopez Pacheco, L. Lefevre, and C.-D. Pham, "Fairness Issues When Transferring Large Volume of Data on High Speed Networks With Router-Assisted Transport Protocols," in *High Speed Networks Workshop 2007, in conjunction with IEEE INFOCOM 2007*, Anchorage, Alaska, USA, May 2007.
- [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, pp. 2707–2711 Vol.5, Nov 2004.
- [13] J. Fasson, F. Hobaya, and F. Arnal, "Xcp in a geostationary satellite context," *Satellite and Space Communications, 2008. IWSSC 2008. IEEE International Workshop on*, Oct. 2008.
- [14] D. Pradas, L. Jiang, M. A. V. Castro, P. Barsocchi, and F. Potorti, "Cost-efficient design of hybrid network for video transmission in tropical areas," in *proceedings of the IEEE Vehicular Technology Conference (VTC)- Spring*, Apr. 2009.
- [15] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581 (Proposed Standard), Apr. 1999, updated by RFC 3390. [Online]. Available: <http://www.ietf.org/rfc/rfc2581.txt>
- [16] N. Parvez, A. Mahanti, and C. Williamson, "Tcp newreno: Slow-but-steady or impatient?" *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 2, pp. 716–722, June 2006.