P-XCP: A Transport Layer Protocol for Satellite IP Networks*

Kaiyu Zhou, Kwan L. Yeung and Victor O.K. Li
Department of Electrical and Electronic Engineering
The University of Hong Kong
Hong Kong, PRC.
E-mail: {kyzhou, kyeung, vli}@eee.hku.hk

Abstract- **Explicit Control Protocol (XCP) is a promising transport layer protocol for satellite IP networks. Nevertheless, two problems of XCP are identified in this paper, namely, low throughput under high link error rate conditions, and output link underutilization in the presence of rate-limited connections. To address the first problem, we propose to maintain the transmission rate of an XCP sender when triple duplicate ACK is detected. To solve the second problem, we propose to adjust the aggregated feedback based on the ratio of the number of rate-limited connections to the total number of connections sharing the link. We then combine our proposed solutions to form a new protocol, called P-XCP. Simulation results show that P-XCP overcomes the two problems of XCP. When packet error rate is over 0.1, P-XCP is shown to enjoy a throughput almost double that of XCP.**

I. INTRODUCTION

The Internet is a network of networks, running on a multitude of physical media, including fiber, terrestrial radio, and satellite links. Unfortunately, the characteristics of satellite links, such as large bandwidth delay product (BDP), relatively high bit error rate (BER) and link capacity asymmetry, greatly limit the performance of current transport layer protocols, especially TCP. Although much effort has been made to improve TCP performance in satellite networks with end-to-end mechanisms [4, 5, 10], there is an increasing interest in the splitting approach [3, 6]. The splitting approach treats the satellite portion of the network as an autonomous network with border gateways/routers interfacing a satellite-specific transport layer protocol to and from the existing TCP. It succeeds in simplifying the complexity of satellite-specific transport layer protocol design, but puts more processing burden onto the border gateways.

Following the splitting approach, TCP-Peach [1, 2] and Explicit Control Protocol (XCP) [7] are two promising transport layer protocol candidates for future satellite networks. TCP-Peach [1] and its variant TCP-Peach+ [2] are designed to deal with the large BDP and high BER problems of satellite links. By using low priority dummy packets, TCP-Peach can quickly probe the network for optimal congestion window (*cwnd*) size using its sudden start algorithm, and keep *cwnd* at its optimized value even in a high BER environment with its rapid recovery algorithm. In TCP-Peach+ [2], the original dummy packets are replaced by the so-called NIL packets for better performance. Dummy packets are lower priority packets that carry the same payload as the last sent data packet, whereas NIL packets are lower priority packets that carry the payload randomly chosen from the set of outstanding (sent-but-not-yet-acknowledged) data packets. Thanks to the extra information carried by the NIL packets, TCP-Peach+ gives a better performance [2]. However, both Peach and Peach+ require the routers to support priority drops, a feature not available in present routers. In the rest of this paper, we use TCP-Peach to denote TCP-Peach+ for convenience.

XCP [7] is originally designed to solve the congestion control problem in the Internet, especially for networks with large BDP [7]. The main contribution of XCP is the use of explicit feedback instead of the sender probing for available bandwidth. By decoupling congestion control from bandwidth allocation, XCP outperforms many existing congestion control mechanisms in terms of packet dropping rate, link utilization, queuing delay, and fair resource allocation. Although a satellite network is a typical network with large BDP, it is also characterized with high BER and link bandwidth asymmetry. The latter two characteristics may deteriorate XCP performance. Please refer to Section III for details.

Besides TCP-Peach and XCP, STP [6] is also proposed as a satellite-specific transport layer protocol. Since STP is based on SSCOP [9], it differs significantly from TCP. The protocol conversion to and from STP at border gateways tends to be more complex than TCP-Peach and XCP. So STP is not further considered in this paper.

Although XCP is shown to be a better candidate than TCP-Peach, two problems of XCP are identified in this paper. First, XCP performs poorly under high BER conditions. Second, when rate-limited connections and non-rate-limited connections share an XCP router, the fairness controller of XCP causes output link underutilization. To solve these two problems, a new protocol called P-XCP is proposed in this paper.

The rest of the paper is organized as follows. Section II reviews the major mechanisms of XCP. Section III investigates the performance of XCP under high BER environments. To solve the low throughput problem, we suggest maintaining the transmission rate of the XCP sender when triple duplicate ACK

* This research is supported in part by the Research Grant Council Earmarked Grant 7048/02E, Hong Kong.

is detected. Section IV examines the link underutilization problem of XCP. A refined fairness controller is designed to rectify this problem. Combining the proposed solutions to the two identified problems, a new explicit congestion control protocol called P-XCP is proposed and evaluated in Section V. Section IV summarizes our findings.

II.EXPLICIT CONTROL PROTOCOL

In Explicit Control Protocol (XCP), a *congestion header*, as shown in Fig. 1, is attached to each segment sent. *H_throughput* and *H_rtt* are two header fields that record the sender's estimated throughput and round trip time (RTT). *H_feedback* carries the feedback from the intermediate routers and the receiver. The XCP protocol involves three parties, sender, receiver, and router. The sender adjusts its congestion window size based on the feedback calculated at the routers. To compute the feedback, an XCP router uses an efficiency controller (EC) and a fairness controller (FC). In each control interval (denoted by d and set to the average RTT of all connections carried by the router), with the help of the information carried in the congestion header, the EC computes a desired increase or decrease (denoted by λ) in the *aggregated* traffic load carried on the output link, i.e.

$$\lambda = \alpha \bullet S - \beta \bullet Q / d \,, \qquad (1)$$

where α and β are pre-determined weighting factors with a recommended value of 0.4 and 0.226, respectively. The spare capacity S is defined as the difference between the input traffic rate and the output link capacity. S becomes negative if the input rate is larger than the output capacity. Q is the minimum queue length as measured in a control interval.

The fairness controller (FC) is responsible for maintaining fair resource sharing among all carried connections. It computes a *per-connection* feedback (carried in *H_feedback*) based on λ:

- *If $\lambda \geq 0$, allocate the spare bandwidth to all connections equally.*
- *If $\lambda < 0$ (i.e. senders should slow down), allocate the negative "spare" bandwidth to connections proportional to their current throughputs.*

This equally-increase-and-proportionally-decrease mechanism assures the fairness of XCP.

To prevent convergence stalling when λ is around 0, XCP introduces the concept of *bandwidth shuffling*. Bandwidth shuffling simultaneously allocates and de-allocates part of the bandwidth such that the total traffic load carried on the output link is the same, but the throughputs of individual connections gradually converge to their fair shares. During each control interval, at most γ of the aggregated throughput will be reallocated, where γ is a predefined constant with a recommended value of 0.1.

An XCP sender adjusts its sending rate based on the *H_feedback* value received. Like TCP, XCP also adopts the triple duplicate ACK as a sign of congestion. In other words, an XCP sender halves its sending rate upon receiving three duplicate ACKs in a row.

We have done extensive simulations to compare the performance of TCP-Peach and XCP in satellite networks. (Due to space limitations, please refer to [14] for details.) We find that XCP outperforms TCP-Peach in most situations. Nevertheless, we have identified two potential problems with XCP. First, XCP performs poorly in high BER environment. Second, when rate-limited connections and non-rate-limited connections share an XCP router, the fairness controller of XCP may cause the problem of output link underutilization. These two problems together with our proposed solutions are detailed in the next two sections.

III. PROBLEM 1: POOR PERFORMANCE UNDER HIGH BER

The performance of XCP in a network under high BER is studied by simulations using ns-2 [13]. A dumbbell shaped network as shown in Fig. 2 is adopted, with 20 sender-receiver pairs traversing a satellite link. To model the link bandwidth asymmetry, we set the forward and reverse satellite link bandwidths to 1300 and 65 packets/s respectively. (Different link bandwidth asymmetric levels and different number of sender-receiver pairs have also been studied and they show the same trend as presented below.) The buffer size at the two routers in Fig. 2 is set to 200 data packets. Each simulation runs for a period of time equal to 1000 times of the RTT. Each data packet is 1000 bytes and each receiver's advertised window size is 64 data packets. The size of ACK packet is set to 40 bytes. The bit error rate of the satellite link is converted into packet error rate (PER) for ease of presentation. Due to the different packet sizes, data packets experience higher PER than ACK packets.

For comparison, both TCP-Sack (Sack) [11, 12] and TCP-Peach are implemented. For TCP-Peach, DropTail queue management is assumed at the two routers in Fig. 2. For Sack, both DropTail and RED [8] are implemented. RED parameters

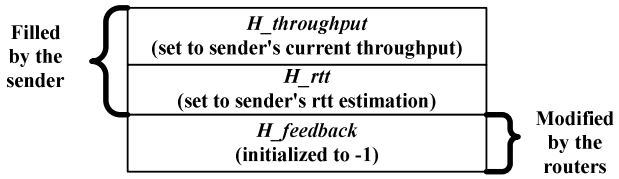

Fig. 1. Congestion Header

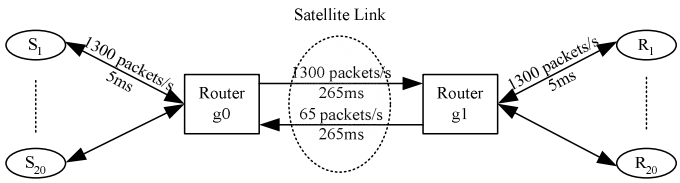

Fig. 2. Simulated network

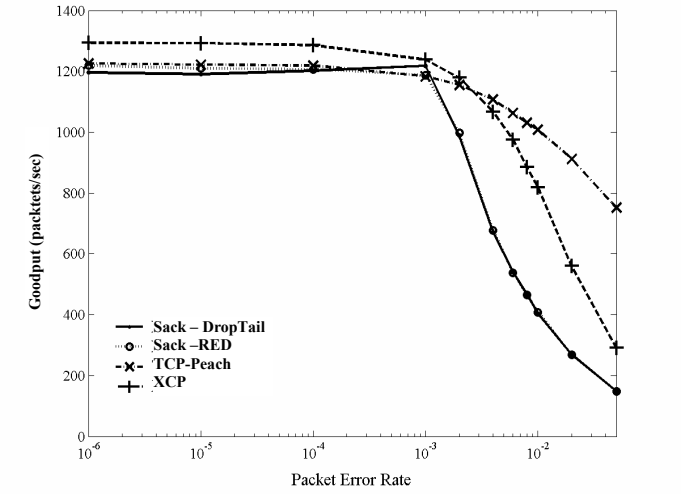

Fig. 3. Goodput vs. packet error rate

are set as follows: the maximum dropping probability, 0.1, the weighting factor, 0.002, the minimum and maximum queue thresholds, 50 and 100 packets, respectively.

Fig. 3 shows the goodput performance of the four transport layer protocols against PER. We see that the goodput of XCP is always higher than that of the two Sack protocols. But it loses to TCP-Peach when the PER is higher than 10^{-3}. The goodput performance gap between XCP and TCP-Peach grows wider as PER increases further.

We find that the reason for the poor performance of XCP under high BER is due to the adoption of the triple duplicate ACK as a sign of congestion. For an explicit rate control protocol like XCP, the packet loss due to congestion is very rare. It is shown in [7] that XCP is stable in various network conditions and its congestion loss rate is always lower than 10^{-6}. This can be explained as follows. Suppose congestion builds up at an XCP router, the router will detect a negative spare bandwidth *S* (before the buffer overflows). Then all the *H_feedback* values in this control interval will be set to negative. Upon receiving negative feedbacks, the associated senders slow down to ease the congestion. Thus congestion loss due to buffer overflow is rare.

As the result, if a sender receives three duplicate ACKs in a row, it can simply conclude that the packet loss is due to transmission errors. As such, halving its sending rate after retransmitting the lost packet is not justified. Therefore, we recommend maintaining the sender's transmission rate when packet loss is detected. With this modification, an XCP sender depends only on the congestion header to adjust its sending rate.

IV. PROBLEM 2: LINK UNDERUTILIZATION

If a connection is bottlenecked at some upstream router (or at the source), we call it a rate-limited connection. Otherwise, we call it a non-rate-limited connection. In this section we show that if XCP is used, upstream congestion can cause permanent link underutilization in the downstream links.

This problem can be seen from the simulations based on the network shown in Fig. 4. For convenience, we denote the connections from A to C and B to C as Connection 1 and Connection 2, respectively. Both connections share the router R and the common link RC. Simulations are conducted by adjusting the receiver's advertised window size to get a rate-limited connection. In a real network, a rate-limited connection will be the result of upstream congestion or insufficient receiver buffer. In the simulation, receiver C advertises a window size (500 packets) that is large enough to fill the pipe to sender A, and a very small window size (1 packet) to sender B. Since all data packets are of size 1000 bytes, the BDP for both Connections 1 and 2 are 100 packets. Due to the small advertised receiver window, Connection 2 cannot send faster than 0.1 Mbps. This is far below its fair share of 5 Mbps on link RC that the fairness controller of XCP at router R tries to allocate.

Although XCP will let Connection 1 expand its window to use part of the bandwidth wasted by Connection 2, link underutilization is unavoidable, and the duration of underutilization is the same as the life of the rate-limited Connection 2. From Fig. 5, we see that the amount of bandwidth wasted is about 10% of the bandwidth on link RC.

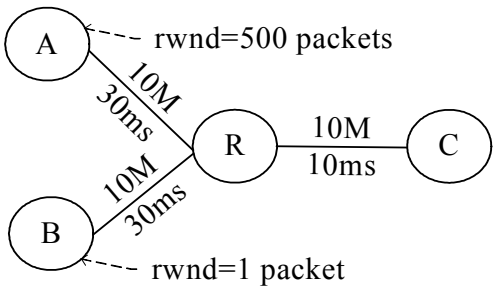

Fig. 4. Simple topology of two connections

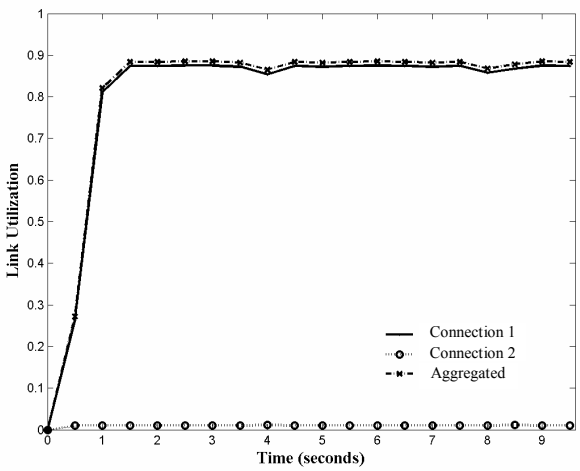

Fig. 5. Link Underutilization Problem

This problem tends to become more serious for more realistic (thus more complicated) topologies.

In the following, we try to derive an upper bound of bandwidth wastage for a more general case. Recall that *bandwidth shuffling* in FC tries to make each connection converge to its fair share, while avoiding convergence stalling. If there is a rate-limited connection, like Connection 2 in the example, we investigate the impact of bandwidth shuffling on the performance of XCP as below.

Let y be the aggregated traffic rate of all N connections sharing the same output link of an XCP router, y_i be the traffic rate of connection i. Define γ^* as

$$\gamma^* = \max(\gamma - \frac{\lambda}{y}, 0) . \quad (2)$$

γ^* can be treated as the effective de-allocation ratio and it ensures that no more than $\gamma \bullet y$ bandwidth is de-allocated. In bandwidth shuffling, XCP first de-allocates the amount of bandwidth $\gamma^* y_i$ from each connection, and then (re-)allocates $\gamma y/N$ to each, so that the total traffic rate carried on the link remains to be y.

When $\gamma y/N$ is bigger than $\gamma^* y_i$, a connection should increase its sending rate. But for a rate-limited connection, other limitations, such as upstream congestion or receiver buffer limitations, prevent it from increasing. So the amount of bandwidth wasted by each rate-limited connection is the difference between the amount of allocated and de-allocated bandwidths. The total amount of bandwidth wasted is given by

$$\phi = \sum_F \max(\frac{\gamma y}{N} - \gamma^* y_i, 0) , \quad (3)$$

where F is the set of rate-limited connections.

From (3), the upper bound of the bandwidth wasted is the amount of bandwidth shuffled, which is γy. But the aggregated feedback λ is only a portion of the spare bandwidth at the output link, so the actual value of the bandwidth wasted will be bigger than the amount of the bandwidth shuffled. From (1), parameter α determines the amount of spare bandwidth allocated to the aggregated feedback λ. So the upper bound of bandwidth wasted is given by

$$\phi = \sum_F \max(\frac{\gamma y}{\alpha N} - \gamma^* y_i, 0) . \quad (4)$$

Substituting the recommended parameter values in [7] in (4), the maximal amount of wasted bandwidth is 25% of the link bandwidth in the worst case.

To solve the link underutilization problem, we propose to adjust the aggregated feedback based on the ratio of the number of rate-limited connections to the total number of connections sharing the link. In XCP, FC tries to allocate aggregated feedback λ equally to each connection in a control interval. If among all the connections sharing the gateway, r of them are rate-limited, the amount of wasted bandwidth is $r \bullet \lambda$. It means that only $(1-r)\lambda$ bandwidth will be successfully (re-)allocated. If we adjust the aggregated feedback λ to $\lambda/(1-r)$, the amount of successfully allocated bandwidth becomes λ, and the link underutilization problem can be solved.

Network traffic varies and this scheme may temporarily over-amplify the amount of free bandwidth. This will increase the queue length variation at an XCP router. A highly dynamic queue length causes dynamic feedback and will affect the stability of XCP. To stabilize the system, we adjust the feedback from $\lambda/(1-r)$ to $k\lambda/(1-r)$, where k is a constant set to 0.9 by heuristics. With this adjustment, the upper bound of bandwidth wastage is 2% of the link bandwidth from (4), and the stability of XCP is maintained.

On the average, a router receives $(cwnd_i \bullet d) / RTT_i$ packets from connection i during a measuring interval of d. We use (5) to estimate N, the number of connections sharing the router.

$$N = \sum_{i \in P} \frac{RTT_i}{cwnd_i \bullet d} , \quad (5)$$

where P is the set of packets arriving at the router in d.

Similarly, we use (6) to estimate the number of rate-limited connections N_L. Let P_L be the set of packets that has a feedback value smaller than that calculated at the router. We have

$$N_L = \sum_{i \in P_L} \frac{RTT_i}{cwnd_i \bullet d} . \quad (6)$$

V. P-XCP WITH VALIDATION

Combining the proposed solutions in Sections III and IV, a new explicit rate control protocol can be designed. Since the new protocol adjusts the aggregated feedback proportional to the ratio of rate limited connections to total connections, we call it Proportional XCP (P-XCP). The pseudocode for P-XCP is shown in Fig. 6.

Applying P-XCP to the network in Fig. 4, we show that the utilization of link RC increases to about 100 percent with negligible increase in average queue length. (Results not included due to space limitation.) We then compare the link utilization performance of P-XCP and XCP based on a more complex network shown in Fig. 7. There are three groups of senders (SG1, SG2, and SG3) and two groups of receivers (RE1, RE2). SG1 has 30 connections connecting to RE1, and

On each packet arrival:
  $N_{total} = pkt\_rtt / (pkt\_cwnd * avg\_rtt) + N_{total}$
On each packet departure:
  $pos\_fbk = pos\_fbk * \max(k / (1 - r), 1)$
  if ($H\_feedback < feedback$) then
    $N_{limited} = pkt\_rtt/(pkt\_cwnd * avg\_rtt) + N_{limted}$
On estimation-control timeout:
  $r = N_{limited} / N_{total}$
  $N_{total} = N_{limited} = 0$
On triple duplicate ACK arrival:
  Retransmit the lost packet and continue

Fig. 6. Pseudocode of P-XCP

each of SG2 and SG3 has 10 connections connecting to RE2. Each receiver advertises a window bigger than the connection's BDP to the sender. The simulation results in Fig. 8 show that the link under-utilization problem in XCP is solved by P-XCP.

To demonstrate the performance improvement of P-XCP in satellite networks, we compare the goodput of P-XCP and XCP in Fig. 9 based on the network in Fig. 2. We see that the throughput problem of XCP under high link error rate is solved.

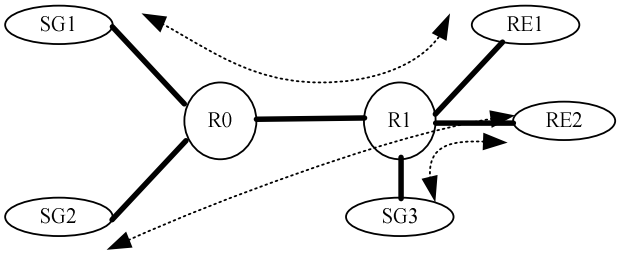

Fig. 7. More complex network

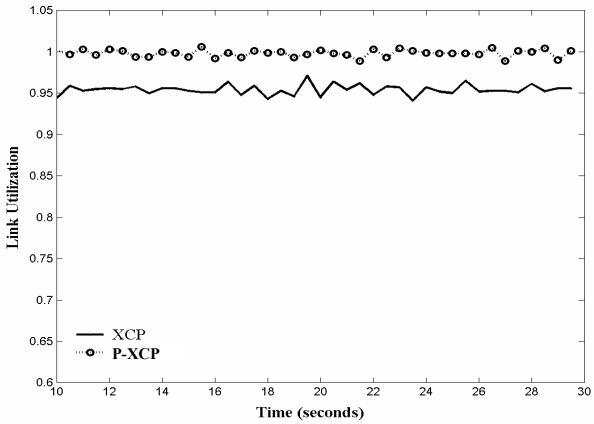

Fig. 8. Link utilization at the link from R1 to RE2

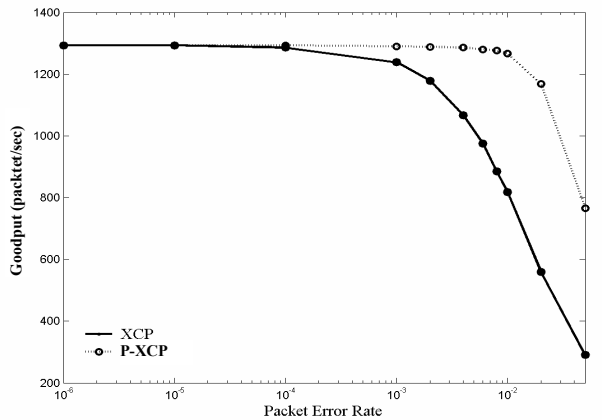

Fig. 9. Goodput of P-XCP with different PER

Comparing with Fig. 3, we also see that P-XCP outperforms TCP-Peach. Simulation results (not included due to space limitation) also show that P-XCP retains the excellent queue length stability and low congestion dropping of the original XCP.

VI. CONCLUSION

This paper focuses on improving the existing XCP protocol over satellite networks. Two problems with XCP are identified, namely, the low throughput problem experienced by XCP under high link error rate, and the link underutilization problem in the presence of rate-limited connections. To address the first problem, we propose to maintain the XCP sending rate when triple duplicate ACK is detected. To tackle the link underutilization problem, we adjust the aggregated feedback based on the ratio of the number of rate-limited connections to the total number of connections sharing the link. We then combine our proposed solutions to the two identified problems to form a new protocol, called P-XCP. Simulation results show that P-XCP overcomes the two problems of XCP.

REFERENCES

[1] I.F. Akyildiz, G. Morabito and S. Palazzo, "TCP Peach: A New Congestion Control Scheme for Satellite IP Networks," IEEE/ACM Transactions on Networking, Vol. 9, No. 3, pp. 307-321, June 2001.

[2] I.F. Akyildiz, X. Zhang and J. Fang, "TCP Peach+: Enhancement of TCP Peach for Satellite IP Networks," IEEE Communication Letters, Vol. 6, No. 7, pp. 303-306, July 2002.

[3] I.F. Akyildiz, G. Morabito and S. Palazzo, "Research Issues for Transport Protocols in Satellite IP Networks," IEEE PCS Magazine, Vol. 8, No. 3, pp. 44-48, June 2001.

[4] M. Allman, S. Floyd, and C. Partridge, "Increasing TCPs initial window," RFC 2414, 1998.

[5] M. Allman, D. Glover, and L. Sanchez, "Enhancing TCP over satellite channels using standard mechanism," RFC 2488, 1999.

[6] T. R. Henderson and R. H. Katz, "Transport protocols for Internet-compatible satellite networks," IEEE JSAC, vol. 17, pp326-344, Feb. 1999.

[7] D. Katabi, M. Handley and C. Rohrs, "Internet Congestion Control for High Bandwidth-Delay Product Networks," ACM SIGCOMM 2002, Pittsburgh, August, 2002.

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

[9] "B-ISDN signaling ATM adaptation layer—Service specific connection oriented protocol (SSCOP)," ITU-T Recommendation Q.2110, 1994.

[10] V. Jacobson, R. Braden and D. Borman, "TCP extensions for high performance," RFC 1323, 1996.

[11] G. Leerujikul and K.M. Ahmed, "TCP over satellite link with SACK enhancement," PACRIM, vol 2, pp. 26-28, Aug. 2001.

[12] S. Floyd, J. Mahdavi, M. Mathis and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP," RFC2883, 2000.

[13] UCN/LBL/VINT. Network Simulator – NS2. http://www-mash.cs.berkeley.edu/ns.

[14] K. Y. Zhou, K. L. Yeung and V. O. K. Li, "P-XCP: A-Transport Layer Protocol for Satellite IP network", available at http://www.eee.hku.hk/~kyzhou/pxcp.pdf.