On Improving The Efficiency and Fairness of TCP over Broadband Satellite Networks

Tarik Taleb, Nei Kato, and Yoshiaki Nemoto
Graduate School of Information Sciences
Tohoku University
Sendai, Japan 980–8579
Email: taleb,kato,nemoto@nemoto.ecei.tohoku.ac.jp

***Abstract*— Along with the vast improvements in satellite technologies and thanks to the continuous growth in the Internet, broadband satellite services are likely to form a strong market in the near future. This paper explores the behavior of one of the most dominant protocols in today's Internet, namely TCP, in LEO satellite networks.**

The paper argues that the TCP rate of each flow should be dynamically adjusted to the available bandwidth when the number of flows changes over time due to handover occurrence. The basic idea is to match the aggregate window sizes of all active flows to the network pipe. Simultaneously, the proposed scheme provides all active connections with feedbacks proportional with their RTTs so that the system converges to optimal max-min fairness. We refer to the scheme as eXplicit and Fair Window Adjustment (XFWA).

I. Introduction

Along with the ongoing improvements in satellite technologies, broadband satellite services are likely to form a strong market in the near future. There are two types of broadband satellite systems: Geostationary and Low Earth Orbit (LEO) satellite systems. The former exhibits high signal delays and power requirements, and thus affects a large number of applications based on IP protocols. Need for lower propagation delays, in conjunction with coverage of high latitude regions, has turned the spot light on LEO systems and has created the need for exploring the behavior of Internet protocols in LEO systems. Among Internet protocols, TCP is the most dominant one in today's Internet traffic. This paper purposes to examine some vexing attributes that impair TCP's performance in LEO systems and to determine the appropriate modifications to help TCP suite well in such environments.

TCP usually results in drastically unfair bandwidth allocations when multiple connections share a bottleneck link. In case of multi-hops satellite constellations, because of the high variance in the distribution of the flows RTT, the unfairness issue becomes more substantial [1]. Additionally, due to handover occurrence in LEO systems, a TCP connection may either alter its path and compete for bandwidth with a different group of connections, or just keep its path but be forced to be sharing the same link with newly coming connections. Both cases will eventually result in an abrupt change in the number of flows. If all TCP senders keep sending data without any adjustment in their sending rates, an increase in the flows count will result in overloading the link with packets causing excessive queuing delays and large number of packet drops, whereas a decrease in the flows count will lead to lower link utilization.

To tackle the above issues, this paper argues an explicit and fair control of the window sizes of all the active TCP connections sharing the same link. We dub our proposed scheme *eXplicit and Fair Window Adjustment (XFWA)*.

The rest of this paper is organized as follows. Section II discusses the state-of-art related work in the field of TCP over satellite. A detailed description of the proposed scheme is given in section III. Section IV describes the simulation set-up and presents simulation results. Concluding remarks are in section V.

II. State-of-Art Related Work

As originally specified, TCP did not perform well over satellite networks for a number of reasons related to the protocol syntax and semantics [2]. To improve throughput performance of TCP protocol in such environments, several proposals have been suggested in recent literature. To cope with slow start algorithm issues, the congestion window is initially set to a value larger than 1 packet size but lower than 4 packets size [3]. Other researchers have investigated the potential of a technique called TCP Spoofing [4] [5]. In this technique, a router near the TCP sender prematurely acknowledges TCP segments destined for the satellite host. This operation gives the source the illusion of a short delay path speeding up the sender's data transmission. One more similar concept is TCP Splitting where a TCP connection is split into multiple connections with shorter propagation delays [6]. In TCP/SPAND [7], network congestion information is cached at a gateway and shared among many co-located hosts. Using this congestion information, TCP senders can make an estimate of the optimal initial congestion window size at both connection start up and restart after an idle time. [8] discusses the usage of low-priority dummy segments to probe the availability of network resources without carrying any new information to the sender. All in all, many researchers have investigated the efficiency and throughput improvement of TCP in satellite networks. However, most proposed solutions have considered only efficiency issues, mainly problems related to slow-start phase, and TCP behavior under many competing flows in satellite networks has not been sufficiently explored.

0-7803-7955-1/03/$17.00 (C) 2003 IEEE

Current TCP implementations do not communicate directly with the network elements for explicit signaling of congestion control. TCP sources infer the congestion state of the network only from implicit signals such as arrival of ACKs, timeouts, and receipt of duplicate acknowledgments (dupACKs). In the absence of such signals, the TCP congestion window grows up to the maximum socket buffer advertised by the receiver. In case of multiple flows competing for the capacity of a given link, this additive increase policy will cause severe congestion, degraded throughput, and unfairness.

One approach to control congestion is to employ scheduling mechanisms, fair queuing, and intelligent packet-discard policies such as Random Early Marking (REM) [9] and Random Early Discard (RED) [10] combined with Explicit Congestion Notification (ECN) [11]. These policies require a packet loss to signal the network congestion to TCP sources early. However, in case of large delay links (e.g. satellite links), these policies become inefficient and may still cause timeouts forcing TCP senders to invoke the slow-start phase. By the time the source starts decreasing its sending rate because of a packet loss, the network may have already been overly congested. Low et al. [12] have shown through mathematical analysis the inefficiency of these Active Queue Management schemes (AQM) in environments with high bandwidth-delay product such as satellite networks.

AQM limitations can be mitigated by adding some new mechanisms to the routers to complement the endpoint congestion avoidance policy. These mechanisms should allow network elements between a TCP source and a TCP destination to acknowledge the source with its optimal sending rate. By so doing, the whole system becomes self-adaptive to traffic demands and more active in controlling congestion and buffer overflows. To tackle TCP limitations in high bandwidth-delay product networks, several studies have been conducted providing valuable insight into TCP dynamics in such environments. TCP-Vegas [13] attempts to compute optimal setting of the window size based on an estimate of the bandwidth-delay product for each TCP connection. As knowledge of the RTT and the bandwidth-delay product of the network is not usually available at network elements, TCP-Vegas requires extensive modifications to current TCP implementations in end-systems.

Katabi et al. [14] proposed a new congestion control scheme, eXplicit Control Protocol (XCP). The scheme substantially outperforms TCP in terms of efficiency in high bandwidth-delay product environments. However, the main drawback of this protocol is that it assumes a pure XCP network and requires significant modifications at the end-system. Explicit Window Adaptation (EWA) [15] and WINdow TRAcking and Computation (WINTRAC) [16] suggest an explicit congestion control scheme of the window size of TCP connections as a function of the free buffer value similar in spirit to the idea of Choudhury et al. [17]. Since the computed feedback is a function of only buffer occupancy and does not take into account link delay or link bandwidth, the two schemes are likely to run into difficulty in face of high bandwidth or large delay links similarly to AQMs. Additionally, EWA and WINTRAC achieve fairness only in the distribution of the maximum achievable window size. The two schemes remain grossly unfair towards connections with high variance in their RTTs distribution.

III. Explicit and Fair Window Adjustment

The scheme requires routers to maintain a table of active flows ID, an estimate of their RTTs, and their last packet transmission time. Prior knowledge of the RTT estimates is usually not available at network elements in terrestrial networks. However, taking advantage of some fundamental attributes of LEO networks, the proposed scheme can compute an estimate of the RTT. By a simple monitoring of the backward and forward traffic of each flow, routers can compute the number of hops traversed by both ACK and data packets using the Time to Live (TTL) field in IP headers. Let H_b and H_f denote the number of hops traversed by an ACK and a data packet before entering the router in question, respectively. Since the value of the inter-satellite link delay, ISL_{delay}, remains constant in satellite networks, and the queuing delays have minimal contribution in the one-way propagation delay, the flow RTT can be estimated as:

$$RTT = 2 \cdot (H_b + H_f + 2) \cdot ISL_{delay}.$$

A flow is considered to be in progress if the time elapsed since its last packet transmission time is inferior than a predetermined threshold δ. This threshold is always updated to the most recent estimate of the average RTT, RTT_{avg}, of all active flows traversing the router.

When multiple TCP connections share a bottleneck link, XFWA matches the aggregate window size of all active TCP flows to the network pipe while at the same time providing all the connections with feedbacks proportional to their RTT values. The feedback of the i^{th} TCP connection is:

$$feedback_i = \frac{RTT_i}{\sum_{j=1}^{N} RTT_j}(Bw \cdot RTT_{avg} + Q_{size}), \qquad (1)$$

where N, Q_{size}, and Bw are the total number of TCP flows that traverse the router, the router's queue size, and the link bandwidth, respectively.

One of the most attributes of this feedback is that it allows the scheme to automatically adapt to the number of active TCP flows, the buffer size, and the bandwidth-delay product of the network. This attribute eventually helps to adjust the protocol's aggressiveness and to prevent persistent queues from forming. To achieve min-max fairness, the first term of equation (1) reallocates bandwidth between individual flows in proportion with their RTTs.

The window feedback is computed every RTT_{avg} time and is written in the receiver's advertised window (RWND) field carried by the TCP header of ACKs similar in spirit to the EWA approach [15]. This operation neither requires modifications to the protocol implementations in the end system, nor does it need modifying the protocol itself. RWND value can only be downgraded: If the original value of the receiver's advertised window, which is set by the TCP receiver,

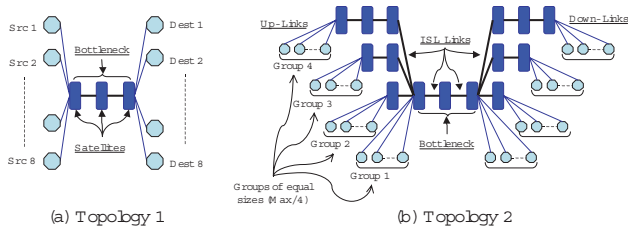

Fig. 1. Simulation Topology

exceeds the feedback value computed by a downstream node, RWND is reduced then to the computed feedback value. A more congested router later in the path can further mark down the feedback by overwriting the RWND field. Ultimately, the RWND field will contain the optimal feedback from the bottleneck along the path.

IV. Performance Evaluation

The performance evaluation relies on computer simulation, using Network Simulator (ns) [18]. To illustrate the issues at hand, a satellite network is modelled as a one network bottleneck shared by various connections (Fig.1). The bottleneck link is composed of three hops. The ISL_{delay} value is set to $20ms$. All up-links and down-links are given a capacity equal to $10Mbps$ and their delays are set to $20ms$. In all simulations, TCP sources implement the TCP NewReno version. The data packet size is fixed to $1000B$ and buffers are equal to the bandwidth-delay product of the bottleneck link. All routers use Drop-Tail as their packet-discarding policy. Simulations were all run for $20s$, a duration long enough to ensure that the system has reached a consistent behavior.

A. Robustness to Dynamic Changes in Traffic Demands

In this experiment, all flows traverse the same number of hops (3 satellites) resulting in each source having an RTT of $160ms$ (Fig.1-(a)). The ISL link bandwidth is set to $1.5Mbps$ (e.g. T1).

In order to examine how XFWA adapts to sudden change in traffic demands, the following scenario is considered. At the beginning of the simulation, 4 flows are activated. After $5s$, another 4 flows are launched. At time $t = 10s$, two flows of the first 4 flows [1] close. After $5s$, two flows of the second 4 flows [2] stop transmitting data. The remaining connections are left active until the end of the simulation.

The growths of TCP segment numbers of the simulated flows are plotted in Fig.2. The figure demonstrates the robustness of XFWA to sudden change in traffic demands. With XFWA, the slope of the segment number lines decreases at time $t = 5s$ as 4 new flows enter the system, and increases at time $t = 10s$ and $t = 15s$ as a result of extra bandwidth becoming available for the remaining active connections. Observe that XFWA helps all flows to progress fairly and to behave in an identical way, whereas, in case of only standard TCP, all TCP flows exhibit great deviations.

[1] in Fig.2 the 3^{rd} and 4^{th} connections
[2] in Fig.2 the 5^{th} and 6^{th} connections

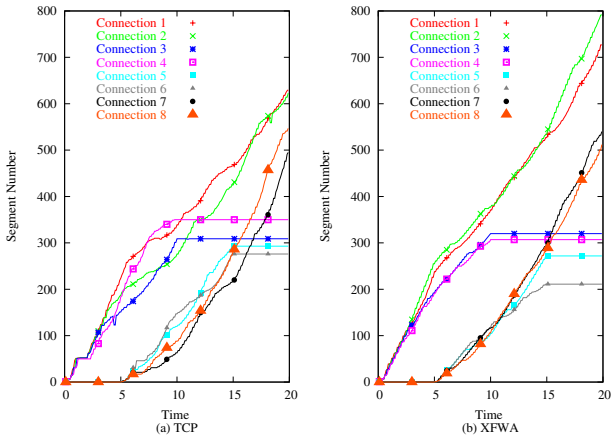

Fig. 2. Segment Number Growth

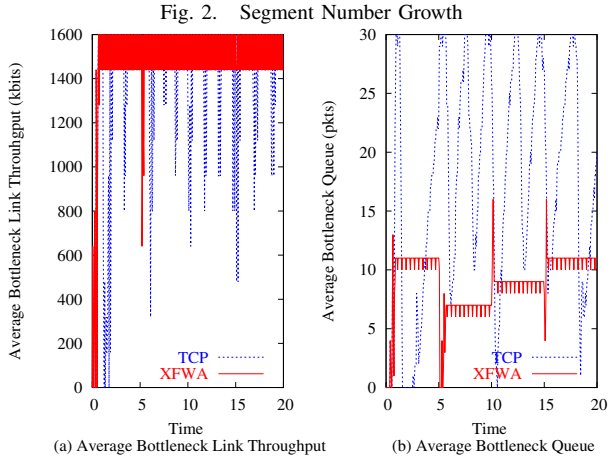

Fig. 3. Average Bottleneck Queue and Link Throughput

The queue occupancy of the bottleneck link and the average throughput, measured in intervals of $100ms$, are presented in Fig.3.

Fig.3-(a) shows that without XFWA, the throughput fluctuates irregularly. The throughput loss is mainly due to the synchronization of packet drops and their simultaneous recovery. In contrast, XFWA's utilization is always near the total capacity of the link and exhibits very limited oscillations.

Fig.3-(b) indicates that without XFWA, TCP senders increase their window size until they cause buffer overflows. This cycle occurs repeatedly and causes the queue size to oscillate more frequently. Changes in traffic demands result also in transient overshoots in the queue size. These transient overshoots and the large oscillations in the queue size can cause timeouts and frequent underflows, thereby resulting in substantially idling the bottleneck link. On the other hand, with XFWA, the buffer underflows, mostly due to the change in traffic demands, are brief and do not significantly affect the bottleneck link utilization. While XFWA aims to reduce queue sizes to minimum by preventing persistent queues from forming, it is observed that the obtained bottleneck queue size remains constant and is larger than a certain number of packets. This is mainly due to the simultaneous release of entire windows of packets, in a single burst, at the beginning of each RTT. One possible solution to this issue is the

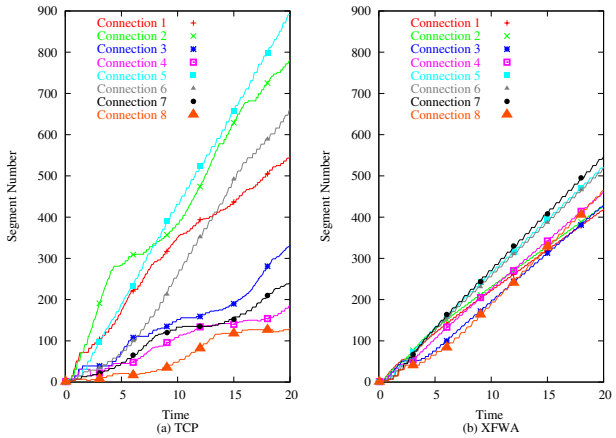

Fig. 4. Segment Number Growth (Access Link Type T1)

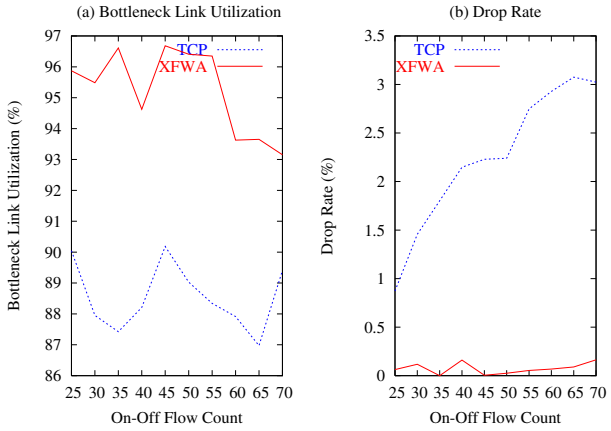

Fig. 6. Link Utilization & Loss Rate in The Presence of Web-like Traffic (Persistent Flows Count 10, Access Link T2)

transmission of packets in a steady stream (multiple, small bursts) over the entire course of the RTT [19].

B. Robustness to Variance in The Flows RTT Distribution

To explore the performance of XFWA in environments with high variance in the RTT distribution, the network topology depicted in Fig.1-(b) is considered. All flows are grouped in four equally-sized groups. Flows belonging to the i^{th} group traverse $2i+1$ satellites causing each flow to have an RTT of $(2i+2)\cdot 40ms$. A number of test scenarios was created by setting the Inter-Satellite Link to different typical link speeds: $1.5Mbps$ (e.g. T1), $10Mbps$ (e.g. T2), $45Mbps$ (e.g. T3), and $155Mpbs$ (e.g. OC3). For each link type, a maximum number of flows, MAX, is fixed so that a minimum value of link fair-share can be guaranteed among all competing flows.

The segment number growths of the TCP connections in case of the access link T1 are plotted in Fig.4. The figure demonstrates the robustness of the proposed scheme to variance in the flows RTT distribution. With the XFWA scheme, the progress of all TCP connections remains parallel during the running time of the simulation. This is because the XFWA scheme divides the available bandwidth fairly among all competing flows while taking into account each flow's RTT. In contrast, in case of standard TCP, which is grossly unfair towards connections with higher RTTs, the segment number growths of the TCP flows exhibit great deviations. Fig.5 confirms the fairness of the proposed scheme and demonstrates the greediness of standard TCP. In case of XFWA, all the flows could send nearly the same amount of packets, whereas, in case of standard TCP, short RTT connections conquer most of the link bandwidth and send significantly larger number of packets compared to the long RTT connections.

C. Performance Evaluation with WEB-Like Traffic

Since a large number of flows in today's Internet are short WEB-like flows, the remainder of this section discusses the interaction and resulting impacts of such dynamic traffic on the XFWA scheme. It has been reported in [20] that WEB-like traffic tends to be self-similar in nature. In [21], it is shown that self-similar traffic can be modelled as several ON/OFF TCP sources whose ON/OFF periods are derived from heavy-tailed distributions such as the Pareto distribution. In this experiment, a scenario where a mix of 10 long-lived FTP flows and a number of non-persistent flows compete for the bottleneck link bandwidth is considered. The considered network configuration is similar to that of Fig. 1-(a). The ISL link bandwidth is set to $10Mbps$ (e.g T2). The simulation starts with 10 persistent connections at time $t = 0s$. The persistent TCP flows remain open until the end of the simulation. At time $t = 5s$, the On-Off TCP flows are activated and remain open for a duration of $10s$. The On/Off periods of the non-persistent connections are derived from Pareto distributions with the mean On period and the mean Off period set to $160ms$, a value on the average equal to the flows RTT. This choice is made deliberately to prevent the On/Off TCP sources from entering the slow-start phase even after periods of idleness. This will thus help to illustrate the resiliency of the XFWA even in the case of significant amount of burstiness in the network.

Fig.6 shows the bottleneck utilization and drops rate for different number of On/Off flows count. The results demonstrate the resiliency of the XFWA scheme to accommodate bursty traffic. The scheme maintains higher utilization of the bottleneck link and significantly reduces the number of drops even for higher number of On/Off flows. The main reason behind this performance is that, unlike standard TCP, the XFWA algorithm attempts to bound the aggregate window sizes of all active TCP flows to the bandwidth-delay product of the network and thus avoids overloading the bottleneck link with packets.

V. Conclusion

In this paper, we proposed an explicit and fair window adjustment method to improve TCP performance over satellite networks. The proposed scheme controls the overall network utilization by matching the sum of window sizes of all active TCP connections to the bandwidth-delay product. Min-max fairness is achieved by providing each connection with a feedback value proportional to its RTT. Signaling feedbacks

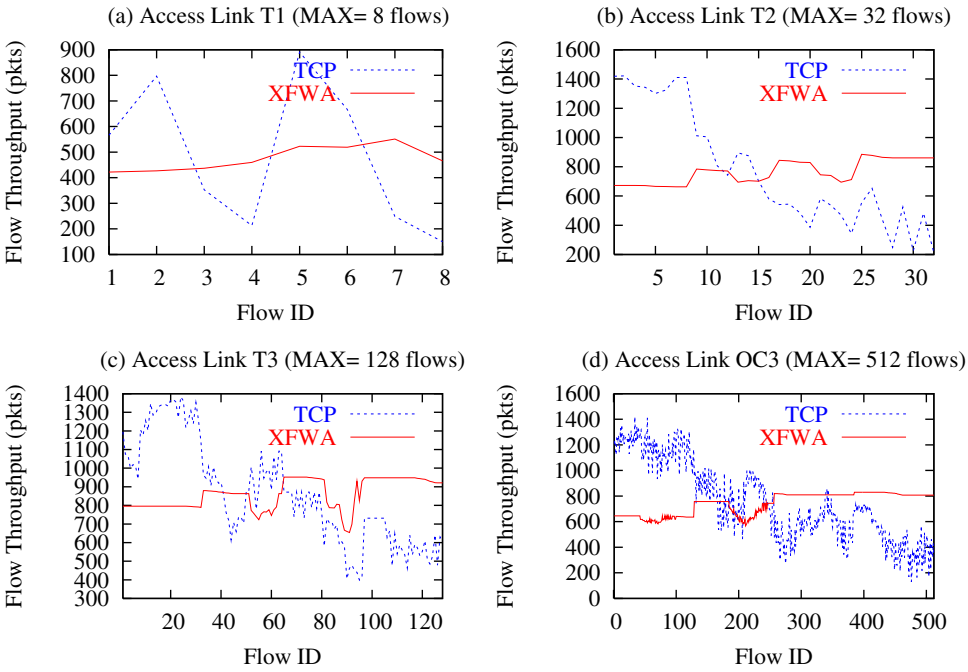

Fig. 5. Individual Flow Throughputs for Different Access Links

to TCP senders is accomplished by modifying the receiver's advertised window (RWND) field carried by TCP ACKs. The sending rate of a TCP connection can be regulated by the sender according to the intersection of the congestion window (CWND) and the RWND value. Simulation results showed that XFWA had the potential to substantially improve the system fairness and to make better utilization of the link. Experiments with dynamic changes in traffic demands showed that XFWA managed to control the buffer occupancy well and to achieve stability when a change in traffic load occurred. Resiliency of the proposed scheme to accommodate WEB-like traffic was verified by investigating the dynamics of the proposed scheme in a scenario where a mix of greedy and non-persistent flows competes for the bandwidth of the bottleneck. Due to paper length limitation, the authors are compelled to omit discussion on the necessary computation load and implementation issues. For a detailed discussion on the practicality of the scheme and more experimental results, the interested reader is directed to [22].

REFERENCES

[1] G. Montenegro, S. Dawkins, M. Kojo, V. Magret, and N. Vaidya, *Long Thin Networks*. Internet RFC 2757, 2000.

[2] C. Partridge and T.J. Shepard, *TCP/IP Performance over Satellite Links*. IEEE Network. pp. 44-49. September/October 1997.

[3] M. Allman, S. Floyd, and C. Partridge, *Increasing TCP's Initial Window*. Internet RFC 2414, 1998.

[4] T. Henderson, R. Katz, *Transport Protocols for Internet-Compatible Satellite Networks*. IEEE Journal on Selected Areas in Communications, Vol. 17, No. 2, pp. 326-343, February 1999.

[5] H. Balakrishnan, V.N. Padmanabhan, and R. Katz, *A comparison of mechanisms for improving TCP performance over wireless links*. IEEE/ACM Trans. Networking, Vol. 5, December 1997.

[6] A. Bakre and B.R. Badrinath, *I-TCP: Indirect TCP for mobile hosts*. Proc. 15th Int. Conf. Distributed Computing Systems (ICDCS), pp. 136-143, May 1995.

[7] V. Padmanabhan and R. Katz, *TCP fast start: A technique for speeding up web transfers*. Proc. IEEE GLOBECOM, November 1998.

[8] I.F. Akyildiz, G. Morabito, and S. Palazzo, *TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks*. IEEE/ACM Trans. Networking, Vol. 9, No. 3, June 2001.

[9] S. Athuraliya, V.H. Li, S.H. Low, and Q. Yin, *REM: Active Queue Management*, IEEE Network, January 2001.

[10] S. Floyd and V. Jacobson, *Random Early Detection Gateways for Congestion Avoidance*. IEEE/ACM Trans. on Networking, Vol. 1, No. 4, August 1993.

[11] K.K. Ramakrishnan and S. Floyd, *Proposal to Add Explicit Congestion Notification (ecn) to IP*. Internet RFC 2481, January 1999.

[12] S.H. Low, F. Paganini, J. Wang, S. Adlakha, and J.C. Doyle, *Dynamics of TCP/AQM and a Scalable Control*. Proceedings of INFOCOM 2002.

[13] 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, October 1995.

[14] D. Katabi, M. Handley, and C. Rohrs, *Congestion Control for High Bandwidth-Delay Product Networks*. Proceedings of SIGCOMM 2002.

[15] L. Kalampoukas, A. Varma, and K.K. Ramakrishnan, *Explicit Window Adaptation: A Method to Enhance TCP Performance*. IEEE/ACM Trans. on Networking, Vol. 10, No. 3, June 2002.

[16] J. Aweya, M. Ouellette, D.Y. Montuno, and Z. Yao, *WINTRAC: A TCP Window Adjustment Scheme for Bandwidth Management*. Performance Evaluation Vol. 46, 2001.

[17] A.K. Choudhury and E.L. Hahne, *Dynamic Queue Length Thresholds in a Shared Memory ATM Switch*. Proceedings of INFOCOM, March 1996.

[18] UCB/LBNL/VINT, *Network Simulator - ns (version 2)*. http://www.isi.edu/nsnam/ns/

[19] J. Kulik, R. Coulter, D. Rockwell, and C. Partridge, *Paced TCP for High Delay-Bandwidth Networks*. Proceedings of IEEE GLOBECOM, December 1999.

[20] K. Park, G. Kim, and M. Crovella, *On the Relationship between File Sizes, Transport Protocols, and Self-Similar Network Traffic*. Proceedings of ICNP, 1996.

[21] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, *Self-Similarity through High Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level*. Proceedings of SIGCOMM, 1995.

[22] T. Taleb, *TCP Performance Evaluation over Multi-Hops Satellite Constellations*. Master thesis, Graduate School of Information Sciences, Tohoku University, February 2003