REFWA Plus: Enhancement of REFWA to Combat Link Errors in LEO Satellite Networks

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

***Abstract*— As a distinguished characteristic of satellite environments, high *bit error rates* (BER) impair the performance of the Transmission Control Protocol (TCP) over satellite networks. In this paper, an error recovery mechanism is proposed to further improve the performance of the Recursive, Explicit, and Fair Window Adjustment (REFWA) scheme [1] [2] in satellite IP networks. The proposed mechanism is called REFWA Plus.**

The fundamental challenge in loss recovery over satellite networks consists in distinguishing between congestion induced packet drops from those due to link errors. In the proposed mechanism, this distinction is based on comparison of the new and old values of the explicit congestion feedback signaled by the REFWA scheme to TCP senders. Effectively, a packet drop due to network congestion is likely to be preceded by a decrease in the computed feedback, whereas packet drops that are followed by an increased feedback are likely to be due to link errors.

A set of simulations is conducted to evaluate the performance of the REFWA Plus scheme. Simulation results demonstrate that compared to TCP Newreno, TCP Westwood, and REFWA, the proposed scheme improves the system fairness and makes better utilization of the bottleneck link even in environments with high bit error rates.

I. INTRODUCTION

The need for satellite communication systems has grown rapidly during the last few years. Inter-networking with satellites began with the use of individual satellites in geostationary orbits. However, requirements for lower propagation delays and propagation loss, in conjunction with the coverage of high latitude regions for personal communication services, have sparked the development of new satellite communication systems called Low Earth Orbit (LEO) satellite systems. Due to the universality of the Transmission Control Protocol (TCP), an in-depth understanding of TCP and recognition of its merits and drawbacks in LEO satellite networks are of vital importance. This understanding underpins the research work outlined in this paper and some of our recent published works [1] [2] [3].

It has been widely demonstrated that current TCP implementations exhibit poor performance over satellite network systems [4]. The performance issues of TCP stem from the inherent characteristics of satellite links; such as long propagation delays and high bit error rates (BER).

In the first instance, a TCP sender takes a long time to increase its congestion window during the slow-start phase. Additionally, when multiple connections with high variance in their round-trip times (RTTs) distribution share a LEO link, TCP results in drastically unfair bandwidth allocations [5]. This unfairness issue becomes more substantial in case of frequent handover occurrences, a general characteristic of LEO satellite networks [6]. Such efficiency and fairness issues have been separately investigated by a large body of prior research work and a set of pioneering solutions has been proposed [7] [8] [9]. The Recursive, Explicit, and Fair Window Adjustment (REFWA) scheme is yet the sole work that has jointly addressed the two issues [1] [2].

In the second instance, a TCP sender operates on the conservative assumption that any segment losses are due to congestion. This assumption ignores the possibility of transient random errors that may be due to atmospheric factors. TCP is usually incapable of detecting the nature of the error but only its result. When an error occurs, a TCP sender backs off its transmission rate and then applies a gradual increase to its reduced window size. While this back-off strategy avoids overloading the network with packets, it comes at the cost of significantly degraded goodput in case of link errors. In the absence of a reliable algorithm that can distinguish between congestion and errors, there is a need to find ways to let TCP senders know that segment loss is due to transmission errors, not congestion and thus they should not reduce their sending rate. While a potential number of post-TCP standard improvements have been devised in recent literature to overcome the above-mentioned shortcomings of TCP in high BER links (e.g W-TCP [10] and TCP Santa Cruz [11]), TCP Westwood [12] is the most worth mentioning scheme.

TCP Westwood is a sender-side modification of the TCP congestion window algorithm. It improves upon the performance of TCP Reno in wired as well as wireless networks. The key concept of TCP Westwood is to use an estimate of the available bandwidth to set the congestion window and slow start threshold after a congestion episode. A TCP Westwood source performs end-to-end estimate of the bandwidth available along a TCP connection by measuring and averaging the rate of returning acknowledgments (ACKs). Whenever the sender perceives a packet loss, inferred by a timeout occurrence or reception of a certain number of duplicate ACKs, the sender uses the bandwidth estimate, an approximate of the effective bandwidth at the congestion time, to select the optimum values of the congestion window and the slow start threshold. By so doing, TCP Westwood ensures faster recovery and efficient utilization of network resources. Although TCP

0-7803-8924-7/05/$20.00 (C) 2005 IEEE

Westwood has been shown efficient in wireless and satellite networks, its performance remains largely dependent on the accuracy level of the network bandwidth estimation.

As a remedy to the throughput degradation due to link errors, this paper introduces an error recovery mechanism as an enhancement for the REFWA scheme. The basic idea behind the proposed mechanism consists in comparing the old and new feedback values signaled by the REFWA scheme to a TCP sender. If a packet drop is followed by no change or an increase in the feedback, the sender judges the loss as a link error and is, accordingly, exempted from shrinking its sending rate. If a packet drop is preceded by a decrease in the computed feedback, the sender can consider it as a congestion indication and should accordingly enter the congestion avoidance phase. In addition to the good performance of the REFWA scheme in terms of improving both the system efficiency and fairness, the proposed scheme has the potential of differentiating between congestion induced packet losses and link errors. This feature helps the scheme to further improve the throughput in environment with high BER rates. The proposed mechanism is dubbed REFWA Plus.

The remainder of this paper is structured as follows. Section II describes briefly the REFWA scheme and highlights the distinct features that are incorporated in the proposed scheme, REFWA Plus. Section III portrays the simulation philosophy and discusses the simulation results. The paper concludes in Section IV with a summary recapping the main advantages and achievements of the proposed scheme.

II. Operational Overview of the REFWA Plus Scheme

Before delving into details of the proposed error recovery mechanism, there is firstly a brief description of the REFWA scheme. In a LEO constellation, the REFWA scheme is implemented at each satellite. The REFWA scheme provides each active TCP flow with a feedback proportional to its RTT. The feedback value is the optimum window size a TCP sender should be sending data at so as not to overload the network with packets. Feedbacks are computed periodically in such a way that improves the system efficiency and fairness. The sum of the feedbacks of all active TCP connections sharing the same bottleneck is equal to the effective network bandwidth delay product of the bottleneck. A detailed description of the feedback computation method can be found in [1]. Feedbacks are signaled to TCP sources via the receiver's advertised window (RWND) field in the TCP header of ACKs. Senders should accordingly regulate their sending rates. It should be emphasized that when a satellite gets congested, the REFWA scheme quickly computes a smaller window feedback, and, when extra bandwidth becomes available, it feeds back TCP sources with larger window sizes. The remainder of this section explains how the proposed REFWA Plus scheme exploits this feature to combat link errors in LEO satellite networks.

TCP usually misinterprets packet losses as network congestion and unnecessarily cuts its congestion window. In deed, when a link error occurs, packets arrive at the receiver out of order. The receiver sends back an ACK immediately to inform the sender that a packet with a certain sequence number is missing. Such an ACK is referred to as a duplicate ACK (DupACK). The sender retransmits the missing packet when it receives more than three DupACKs with the same sequence number. After retransmission, the sender reduces its window size to half and enters the congestion avoidance phase. Being unaware of the transmission error, the sender assumes that the datagram loss is an indication of network congestion and gratuitously throttle its transmission rate. In case of Newreno based TCP variant [13], Partial ACKs (ParACKs) are used to indicate the occurrence of multiple losses in a single window. Upon reception of a ParACK, the sender retransmits the lost packet and waits for an ACK to come back. To retransmit multiple lost packets, multiple RTTs are thus required. This, coupled with the fact that satellite links exhibit long delays, means that the TCP sender may necessitate a long time to increase its congestion window to its value before entering the fast retransmit phase. This leads to a drastic under-utilization of the network resources.

To avoid such an unnecessary shrinkage of transmission rate, the REFWA Plus scheme makes use of the feedback signaled to TCP senders by the REFWA scheme. Fig. 1 portrays the major procedures in the proposed scheme. Upon reception of a normal ACK packet, a TCP sender records the feedback value as Φ_{NorACK}. When the sender receives a ParACK, the sender compares the feedback value signaled via the ParACK, Φ_{ParACK}, to the most recent feedback value signaled via a normal ACK, Φ_{NorACK}. If the two values satisfy the following inequality:

$$\Phi_{NorACK} \leq \Phi_{ParACK} \tag{1}$$

the TCP sender interprets the incident as a link error occurrence and retransmits the missing packets, all in the current window, without entering the congestion avoidance phase. By so doing, the unnecessary decrease of the window size can be prevented. This is based on the fact that a packet drop due to network congestion is likely to be preceded by a decrease in the computed feedback, whereas packet drops that are followed by an increased feedback are likely to be due to link errors. The idea of retransmitting all lost packets in a single window is similar in spirit to the idea of Bulk Retransmit of the Go-Back-N (GBN) error recovery scheme [14] and has the potential of recovering from lost packets within the same RTT. Bulk retransmit may however raise the issue of burstiness at the satellite network. One possible solution to this issue is the transmission of lost packets in a steady stream (multiple, small bursts) over the entire course of the RTT [15]. To keep the network safe from congestion, each sender sets its congestion window (cwnd) and the slow start threshold (ssthresh) values to its optimal sending rate Φ_{ParACK}. On the other hand, in case of high BER environment, the Retransmit Timeout (RTO) backoff algorithm [16], used in most TCP variants, doubles the RTO value until it is 64 times the original value, leading to a significant waste of both bandwidth and time. To overcome long idle waiting times due to large TCP timeouts in case of

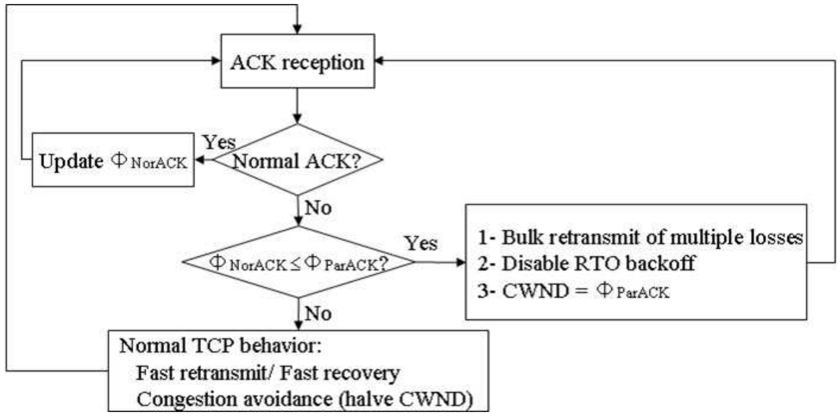


Fig. 1. Major steps in the REFWA Plus scheme

high BER environments, RTO is set to a fixed value when the sender classifies the packet loss as a link error. This concept is similar in spirit to the idea presented in [17]. If Inequality (1) does not hold, the sender retransmits the dropped packets and reduces its window size to half. In other words, it proceeds in the same way as an ordinary TCP sender. Observe that the proposed operation can be accomplished without changing the protocol and requires a merely simple modification at only the sending terminal.

III. Performance Evaluation

A. Simulation Set-up

Having described the details of the REFWA Plus scheme, we now direct our focus to evaluating its performance. The performance evaluation relies on computer simulation, using Network Simulator (ns) [18]. Particular attention is, thus, paid to the design of an accurate and realistic one. Unless otherwise noted, the parameters specified below are those used in all the experiments throughout the paper.

To illustrate the issues at hand, we model a satellite network as a one network bottleneck shared by 10 connections. Fig.2 shows the example network configuration that will be used in our study. The bottleneck link is composed of three satellites. All up-links, down-links, and inter-satellite links are given a capacity equal to $10Mbps$. Their delays are set to $20ms$ (e.g Teledesic constellation). These parameters are chosen with no specific purposes in mind and do not change any of our fundamental observations about the simulation results.

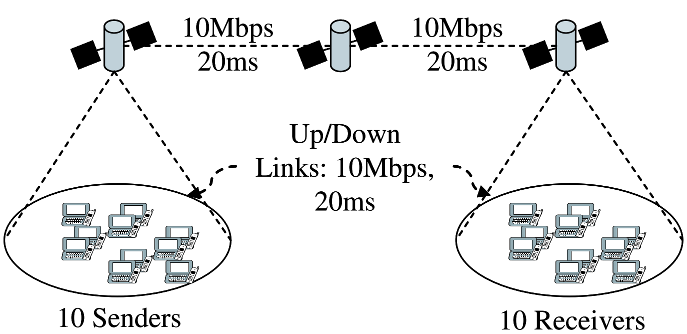


Fig. 2. Simulation Topology

In the performance evaluation, TCP Reno, TCP Newreno, TCP WestwoodNR[1], and REFWA are used as comparison terms. The reason behind the choice of the TCP WestwoodNR and TCP Newreno among other TCP implementations underlies beneath the fact that TCP Newreno achieves faster recovery from multiple losses within the same window and has the potential of significantly improving TCP's performance in the case of bursty losses.

In all simulations, the sources adopt the same protocol. We model the connections as greedy long-lived FTP flows. The data packet size is fixed to $1024B$. In order to remove limitations due to small buffer size on the network congestion, we use buffers equal to the bandwidth-delay product of the bottleneck link. Due mostly to its simplicity and its wide usage in today's switches and routers, all satellites use routers with Drop-Tail as their packet-discarding policy. Simulations were all run for $120s$, a duration long enough to ensure that the system has reached a consistent behavior. The loss probability for link errors is varied within the range $[10^{-5}; 0.5]$. All presented results are an average of multiple simulation runs. Table I shows a complete list of the simulation parameters and the range of values studied.

TABLE I
SIMULATION PARAMETERS

Factor	Simulation Parameters and Range of Values
ISL capacity	10 Mbps
ISL delay	20 ms
Up/Down links capacity	10 Mbps
Up/Down links delay	20 ms
Flows count	10
Simulation duration	120 s
BER range	10^{-5} – 0.5
Packet size	1024 Bytes

Two quantifying parameters are used to evaluate the performance of the proposed scheme: bottleneck link utilization and fairness index. The link utilization is the ratio of the aggregate throughputs of the 10 connections to the bottleneck link capacity. This measure involves the aggregate traffic's behavior and indicates the efficiency of the transmission protocol. The

[1]The Newreno based version of TCP Westwood

fairness index indicates the relative throughput of flows sharing a link [19] and is defined as:

$$F(x) = \frac{(\sum_{i=1}^{N} \frac{x_i}{b_i})^2}{N \cdot \sum_{i=1}^{N} (\frac{x_i}{b_i})^2} \tag{2}$$

where x_i is the actual throughput of the i^{th} flow and b_i is the equal share of the bottleneck link capacity. The fairness index of a system ranges from zero to one. Low values of the fairness index represent poor fairness among the competing flows. Depending on the application and the number of TCP senders, gaining higher fairness values is sometimes worthwhile even at the cost of reduced efficiency.

B. Simulation Results

Fig. 3(a) shows the bottleneck link utilization in case of using the four schemes for different bit error rates. In the figure, the link utilization obtained using REFWA Plus is always higher than that obtained in case of REFWA, standard TCP, Newreno, or TCP WestwoodNR. When the error probability is low, the link utilization of REFWA Plus is the same as that of REFWA. This is due to the fact that packet losses due to link errors are rare, and most of packet drops are due to network congestion. However for higher bit error rates, the REFWA Plus achieves significantly better link utilization than the other schemes. In deed, the link utilization improvement of REFWA Plus over the other four schemes for BER values higher than 10^{-2} is higher than 50%. Furthermore, we observe that in significantly higher BER environments (e.g $BER = 0.3$), the link utilization experienced in case of REFWA, TCP Reno, TCP Newreno, and TCP WestwoodNR is almost null, whereas the REFWA Plus scheme succeeds in making use of more than 30% of the network resources.

This good performance is mainly due to the ability of the proposed scheme to distinguish between packet losses due to link errors and those due to network congestion. Whereas, the low link utilization of REFWA, TCP Reno, TCP Newreno, and TCP WestwoodNR is due to the fact that senders misinterpret packet losses as network congestion and halve their window sizes sometimes multiple times due to multiple losses within one window of data. In deed, in case of heavy errors, ssthresh tends to take values, say one to four packets, significantly smaller than the optimal values. Reducing the congestion window (cwnd) to the ssthresh value, as in normal TCP, drastically throttles the TCP throughput. The throughput degradation becomes further significant as the idle waiting time becomes longer due to the RTO backoff algorithm. On the other hand, in case of REFWA Plus, TCP senders keep on transmitting data more aggressively at congestion windows equal to Φ_{ParACK}. Although many packets will be lost on the way because of the link errors, some will manage to reach the destination. This explains the 30% utilization experienced in case of REFWA Plus.

The REFWA Plus not only outperforms the four other schemes in terms of efficiency, but has also the potential to maintain system fairness. Fig. 3(b) graphs the fairness index values in case of the five schemes for different bit error rates. The figure demonstrates that for lower bit error rates, the REFWA and REFWA Plus scheme performs similarly and achieves the highest fairness (highest values of fairness index). This performance is consistent with the simulation results presented in [1] [2]. It is observed that although TCP WestwoodNR and TCP Newreno achieve relatively good link utilization for low bit error rates (Fig.3(a)), they fail to sustain a fair service. This degraded fairness happens despite the fact that all simulated connections have the same RTT. The underlying reason beneath this behavior consists in the fact that when a TCP Newreno or TCP WestwoodNR flow experiences an error and mistakenly halves its transmission rate, the other flows rushes for using the newly available bandwidth. The first flow ends up then with little bandwidth to recover from the error. If the remaining bandwidth is not sufficient for the error recovery, a timeout may occur and the flow would consequently be forced to enter the slow-start phase. This would further decrease the throughput of the first flow and would cause a significant disproportion among the flows throughputs. On the other hand, in case of REFWA or REFWA Plus, flows are not allowed to obtain portions of the available bandwidth larger than their fair share values. Therefore, even in case of an error occurrence, flows will be always guaranteed a fair portion of the network bandwidth to recover from the loss. It is observed also that for high bit errors, the five schemes experienced an abrupt decrease in their fairness index values. The REFWA Plus exhibits however the best fairness performance. The low values of the fairness index of the REFWA Plus scheme are attributable to the error detection accuracy of the scheme. Indeed, it was observed from the simulation results that some REFWA Plus senders successfully detected all transmission errors, whereas other senders missed the detection of some errors. Successful flows experienced thus higher throughputs. This resulted in a disproportion among the flows throughput; a fact that is ultimately manifested in the form of lower values of fairness index. A possible solution to this issue is to add a detection tolerance, α, to Inequality 1 as follows:

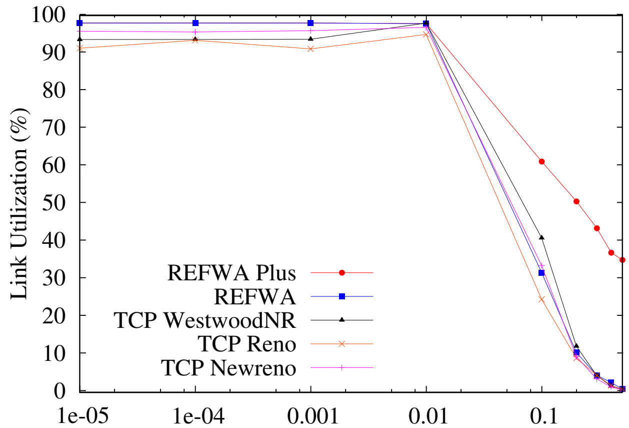
$$\Phi_{NorACK} \leq (1 + \alpha) \cdot \Phi_{DupACK} \tag{3}$$

The choice of α should be a compromise between false positive [2] and false negative [3] detections. In deed, small values of α may fail the detection of some transmission errors, whereas large values of α would take congestion induced packet drops for link errors, allow the senders to transmit data at high rates, and ultimately cause further congestion and packet drops. The setting of α deserves further study and the authors are currently investigating the performance of the REFWA Plus for different values of α.

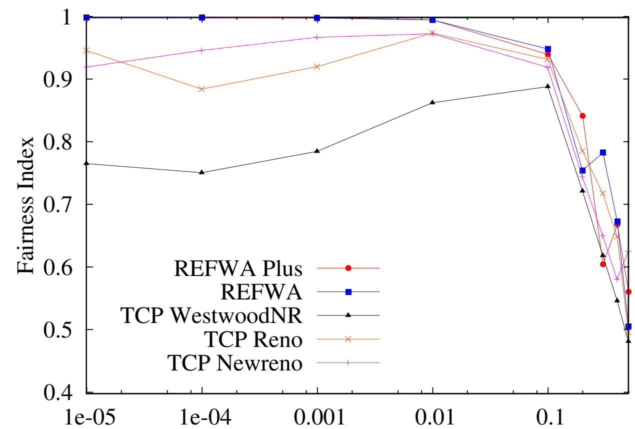
Finally, it should be noted that the presented results are of simulations of flows with same RTTs in a steady-state environment. Similar experiments were conducted considering flows

[2]Link errors that go undetected by the scheme

[3]Packet drops that are taken for link errors



(a) Link utilization



(b) Fairness

Fig. 3. Performance evaluation of REFWA Plus, REFWA, TCP WestwoodNR, TCP Newreno, and TCP Reno for different bit error rates.

with high variance in their RTTs distribution and network topologies with dynamic changes in their traffic demands. And interestingly, identical results were obtained.

IV. Conclusion

In this paper, we introduced an improvement over the REFWA scheme to combat link errors in LEO satellite networks. The proposed error recovery mechanism, dubbed REFWA Plus, has the potential of differentiating between congestion induced packet losses and link errors. This feature helps the scheme to further improve the network resources utilization in environment with high BER rates, as is confirmed by the conducted simulation results. Besides the system efficiency, the REFWA Plus improves also the system fairness and provides a fair share of network resources; a characteristic inherited from the REFWA scheme.

Work is still in progress in many directions. We are currently investigating the usage of Inequality 3 to further refine the detection accuracy of the scheme. In the conducted simulations, we have considered the architecture of "REFWA Plus over Newreno". Further improvements can be obtained by considering other architectures (e.g REFWA Plus over WestwoodNR). This forms the basis of our future research work.

References

[1] T. Taleb, N. Kato, and Y. Nemoto, *"An Explicit and Fair Window Adjustment Method to Enhance TCP Efficiency and Fairness over Multi-Hops Satellite Networks",* IEEE J. Select. Areas Commun., vol. 22, no. 2, pp.371-387, Feb. 2004.

[2] T. Taleb, N. Kato, and Y. Nemoto, *"A Recursive, Explicit and Fair Method to Efficiently and Fairly Adjust TCP Windows in Satellite Networks",* in Proc. of IEEE Int. Conf. Commun., Paris, France, Jun. 2004.

[3] T. Taleb, N. Kato, and Y. Nemoto, *"On Improving the Efficiency and Fairness of TCP over Broadband Satellite Networks",* in Proc. of IEEE VTC 2003-Fall, Orlando, Fl., Oct. 2003.

[4] C. Partridge and T.J. Shepard, *"TCP/IP performance over satellite links",* IEEE Network Mag. PP. 44-49. Sept/Oct. 1997.

[5] G. Montenegro, S. Dawkins, M. Kojo, V. Magret and N. Vaidya, *"Long Thin Networks",* Network Working Group, RFC 2757, Jan. 2000.

[6] L. Wood, G. Pavlou and B. Evans, *"Effects on TCP of Routing Strategies in Satellite Constellations",* IEEE Comm. Mag., vol. 39, pp. 172-181, Mar. 2001.

[7] 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, pp. 307-321, June 2001.

[8] T. Henderson and R. Katz, *"Transport Protocols for Internet-Compatible Satellite Networks",* IEEE J. Select. Areas Commun., vol. 17, no. 2, pp. 326-343, Feb. 1999.

[9] 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, pp. 338-350, June 2002.

[10] P. Sinha, T. Nandagopal, N. Venkitaraman, R. Sivakumar, V. Bharghavan, *"WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks",* in Proc. of ACM Mobicom 99, Seattle, WA, USA, Aug. 1999.

[11] C. Parsa, J.J. Garcia-Luna-Aceves, *"Improving TCP Congestion Control over Internets with Heterogeneous Transmission Media",* in Proc. of 7^{th} IEEE Int. Conf. Network Protocols, ICNP 99, Toronto, Canada, Nov. 1999.

[12] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, *"TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links",* in Proc. of ACM SIGMOBILE 2001, Rome, Italy, July 2001, pp. 287-297.

[13] S. Floyd and T. Henderson, *"The NewReno Modifications to TCP's Fast Recovery Algorithm",* RFC 2582, April 1999.

[14] H. Lima and O. Duarte, *"A Go-Back-N protocol with multicopy retransmission for high speed satellite communications",* In Proc. of IEEE ICC 1994, Piscataway, NJ, USA, May 1994.

[15] J. Kulik, R. Coulter, D. Rockwell, and C. Partridge, *"Paced TCP for High Delay-Bandwidth Networks",* in Proc. of IEEE GLOBECOM, Dec. 1999.

[16] P. Karn and G. Partridge, *"Improving round-trip time estimates in reliable transport protocols",* In Proc. of ACM SIGCOMM, Aug. 1987.

[17] G. Holland, N. Vaidya, *"Analysis of TCP performance over mobile ad hoc networks",* in Proc. of ACM/IEEE MobiCom, Washington, USA, Aug. 1999.

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

[19] D. Chiu and R. Jain, *"Analysis of the increase and decrease algorithms for congestion avoidance in computer networks"* Compute Networks and ISDN Systems, 17:1-14. 1989.