Performance Evaluation of the AeroTP Protocol in Comparison to TCP NewReno, TCP Westwood, and SCPS-TP

Truc Anh N. Nguyen, Siddharth Gangadhar, Greeshma Umapathi, and James P.G. Sterbenz
Department of Electrical Engineering & Computer Science
Information and Telecommunication Technology Center
The University of Kansas
Lawrence, KS 66045
{annguyen,siddharth,greeshma,jpgs}@ittc.ku.edu

ABSTRACT

Due to the unique characteristics of highly dynamic airborne telemetry environments, TCP when deployed in such networks suffers significant performance degradation. Given the limitations of TCP, the AeroTP opportunistic transport protocol with multiple reliability modes has been developed to specifically address the issues posed by telemetry networks. In our previous work, the different modes of AeroTP have been simulated and tested using the open source ns-3 network simulator. In this paper, we use ns-3 to evaluate the overall performance of AeroTP by comparing it with well-studied TCP variants: the widely-deployed TCP NewReno and TCP Westwood designed for wireless environments. Since space networks share many similar characteristics with telemetry environments, we also compare AeroTP with SCPS-TP.

I. INTRODUCTION AND MOTIVATION

Despite wide deployment of TCP (Transmission Control Protocol), it is not a suitable transport protocol for the highly dynamic airborne telemetry network environment due to the environment's unique characteristics. Error-prone asymmetric links, limited bandwidth, and highly dynamic topologies are the challenges that TCP does not handle well [1]. Given the network's constraints and the drawbacks of TCP in the context of iNET (the Integrated Network Enhanced Telemetry) program, we developed a TCP-friendly transport layer protocol AeroTP with multiple reliability modes to address these issues and to meet specific application requirements [2, 3]. In our previous work [4], the different modes of AeroTP have been simulated and tested using the open source ns-3 network simulator [5]. In this paper, we use ns-3 to evaluate the overall performance of AeroTP's fully reliable mode called AeroTP-ARQ and quasi reliable mode AeroTP-FEC by comparing them with other well-studied TCP variants: TCP NewReno, TCP Westwood, and SCPS-TP (Space Communications Protocol Specification-Transport Protocol) for varying levels of link corruption.

The rest of the paper is organized as follows: Section II explains TCP limitations corresponding to each challenge posed by the telemetry environment. Section III reviews the key mechanisms in each protocol that we choose for our comparison, followed by our simulation model and results presented in Section IV. Finally, Section V concludes with directions for future work.

II. BACKGROUND

The unique characteristics of the highly dynamic airborne telemetry environment pose significant challenges to reliable, end-to-end data transfer in general and to TCP in particular. Originally engineered for wired networks, traditional TCP has been proven to be unsuitable for wireless environments. In telemetry networks, TCP suffers severe performance degradation. In this section, we give a brief discussion on why TCP suffers to maintain its functionality in telemetry networks.

Wireless channels invariably possess high BERs (bit error rates) due to inherent noise in the environment and interference from other communication. Given this, corruption is a major source of packet loss in addition to other factors such as network congestion and link outage. Unfortunately, TCP does not have any mechanism to differentiate these sources of loss [6]. For TCP, all losses are interpreted to be due to congestion, and hence its congestion control algorithm is invoked upon the receipt of a certain number (usually three) of DUPACKs (duplicate acknowledgements) or on an RTO (retransmission timeout) event. The congestion control mechanism conservatively reduces the congestion window causing under-utilization of the bandwidth and reduced throughput.

In addition to having a high BER, the telemetry environment is highly bandwidth-constrained due to limited RF spectrum and large data transfers. Using this scarce resource efficiently is an issue that a transport protocol designed for such networks needs to address. With its 20-byte header, TCP can consume a reasonable amount of bandwidth. The effects of this issue is more evident when TCP uses multiple small packets for transmission in order to decrease the probability of a packet drop.

The architecture of a telemetry communication network suggests another characteristic: asymmetry. In telemetry environments, the link from the TAs (test articles) to the GSs (ground stations) generally carries data through the TmNS (telemetry network system) while the link from the opposite direction generally carries control messages. Thus, the link that carries data normally operates at a higher bandwidth than the reverse link. Since TCP requires regular acknowledgments of data received, the ACK channel capacity is proportional to the data channel capacity, posing a considerable threat to TCP performance when being deployed in telemetry networks.

An additional problem is intermittent connectivity, which arises as a result of the high velocity of TAs and their limited transmission range due to their power and weight constraints. Given these constraints, link outages are common in telemetry networks. However, as mentioned earlier, TCP requires a steady flow of acknowledgments, that is not possible in the presence of multiple and frequent link outages. In such cases, TCP assumes data loss and invokes its congestion algorithm to retransmit data. In extreme cases, the connection is terminated because TCP reaches its retransmission limit before the link is restored. Link outages may also lead to frequent connection timeouts that require connection re-establishment. This results in increasing delay and overhead from repeated three way handshakes between the communicating entities. The effect is magnified in the presence of high delay links. In addition, when a network topology is highly dynamic, it is difficult to estimate the RTT (round trip time) in TCP. An underestimated RTT will result in premature retransmissions while an overestimated one produces unnecessary delay.

III. OVERVIEW OF TRANSPORT PROTOCOLS

Given the limitations of TCP, many new reliable transport protocols have been proposed, ranging from TCP variants that consist of a small set of enhancements such as TCP Westwood and TCP Vegas to domain-specific protocols such as SCPS-TP and AeroTP. In this section, we discuss AeroTP along with the other protocols selected for our comparison, namely TCP NewReno, TCP Westwood, and SCPS-TP. SCPS-TP is a space communications protocol and thus serves as the nearest fit in terms of the operating environment to AeroTP. TCP Westwood is specially designed for wireless networks and could be an interesting comparison to AeroTP given the similar nature of channel on which both protocols are intended to operate. Finally, we also include TCP NewReno as it is the current de facto standard of the Internet and serves as a baseline to compare all of the above protocols.

A. AeroTP

AeroTP is a connection-oriented, TCP-friendly, and domain-specific transport protocol designed to address the challenges presented in the airborne telemetry network environment [2, 3]. Similar to other transport protocols, AeroTP performs transport-layer functions such as connection setup and management, transmission control, and error control. The protocol can operate with multiple transfer modes to meet specific application requirements. The different modes of operation are completely reliable, nearly reliable, quasi-reliable, unreliable connection, and unreliable datagram. In this paper, we consider two main modes that have been implemented: AeroTP-ARQ, the fully reliable mode and AeroTP-FEC, the quasi reliable mode.

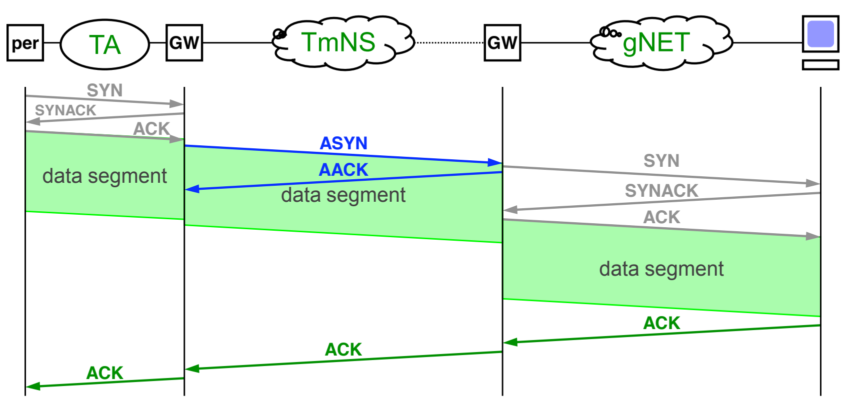

Figure 1: Reliable connection transfer mode

- **AeroTP-ARQ**: The fully-reliable AeroTP-ARQ uses end-to-end acknowledgment semantics between the two endpoints to ensure correct data delivery. To begin a connection, as shown in Figure 1, AeroTP-ARQ uses a modified version of TCP's three-way handshake in an opportunistic connection establishment that overlaps data and control messages to save an RTT delay. The error control in AeroTP-ARQ is achieved through the use of a selective-repeat ARQ mechanism [7]. To reduce the overhead, instead of acknowledging every single packet individually, AeroTP-ARQ performs ACK aggregation at the receiver side. This ACK mechanism is especially useful in bandwidth-constrained

telemetry networks. The number of ACKs being aggregated is a tunable parameter determined based on the sending data rate and the packet loss probability.

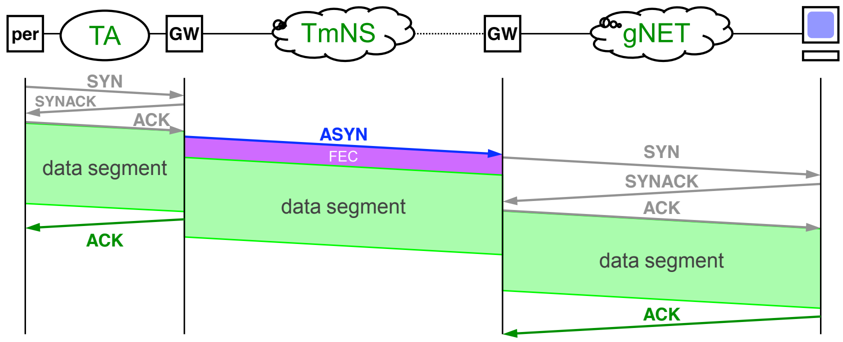

Figure 2: Quasi-reliable connection transfer mode

- **AeroTP-FEC**: The quasi-reliable AeroTP-FEC mode shown in Figure 2 provides different levels of statistical reliability by using the open-loop error recovery mechanism FEC (forward error correction) [8]. The level of statistical reliability is entirely based on the FEC strength attached to each packet. The higher the FEC strength, a higher amount of bits are added thus aiding in increased error detection and correction. On the other hand, increasing FEC strength contributes to increasing overhead and bigger packet sizes contributing to higher delays in the network. Based on the channel quality, bandwidth availability, and packet loss probability, the FEC strength can be tuned to meet network requirements.

B. TCP NewReno

As the current de facto standard of the Internet, TCP NewReno [9, 10] improves performance by modifying its predecessor (TCP Reno [11]) *Fast Recovery* mechanism. Instead of leaving the fast recovery state upon the receipt of a partial ACK as in TCP Reno, TCP NewReno remains in the state, performing retransmissions until all the outstanding data before the fast recovery begun is acknowledged. With this modification, TCP NewReno is capable of recovering from multiple packet losses per window with a single fast recovery invocation. SACK (selective acknowledgment) [12, 13] is an alternative to TCP NewReno's fast recovery.

TCP NewReno has its own disadvantages in addition to the classic non-discrimination of corruption and congestion. Because of the improved fast recovery phase, the algorithm is not inherently designed to take delayed or unordered data delivery even though it is common in wireless networks, and in congested conditions. When segments arrive out of order, specifically more than 3, the algorithm enters the fast recovery phase causing unnecessary packets to be retransmitted while trying to make up for the multiple packet losses. With the increasing use of wireless networks, TCP NewReno is increasingly regarded as a suboptimal solution as a transport layer protocol because of these disadvantages. The main reason for the inclusion of NewReno in our comparative study is the significance and the present day wide deployment of the protocol that serves as the ideal baseline for our analysis.

C. TCP Westwood

TCP Westwood was proposed to overcome the above limitations for the increasing use of wireless channels for data transmission [14]. As a sender side modification, TCP Westwood uses an intelligent algorithm to estimate the bandwidth upon the receipt of every ACK and determines the amount of data to be sent accordingly in the face of a triple DUPACK or RTO event. TCP Westwood also takes into account the presence of delayed and DUPACKs for bandwidth estimation. In comparison with Reno and NewReno, Westwood achieves a significant performance improvement in wireless networks with lossy channels. This is mainly due to the fact that the algorithm continuously probes for the bandwidth of the link, and in the presence of a triple DUPACK or retransmission timeout, the amount of data that needs to be sent is a function of the bandwidth rather than just a conservative fraction as with Reno. As the bandwidth of the link is independent of the corruption of packets and most corruptions occur due to the nature of the channel, Westwood outperforms Reno and NewReno for wireless networks. An enhancement Westwood+ has been proposed to minimize the errors caused in bandwidth estimation due to ACK compression and also mitigate the aggressive behavior of Westwood in the presence of multiple connections [15].

D. SCPS-TP

SCPS-TP, specifically designed for space communications, has the ability to distinguish between different sources of packet loss and handle them accordingly [16]. In the presence of a congestion-induced loss, SCPS-TP uses the TCP Vegas congestion control algorithm [17]. In the presence of a corruption based loss, SCPS-TP enters a four-phase corruption handling process in which multiple ICMP messages are exchanged between related nodes to communicate the presence of corrupt packets. SCPS-TP handles link outage losses using a different four-phase mechanism. To cope with the limited link capacity issue, SCPS-TP uses two mechanisms: header compression and the SNACK (selective negative acknowledgement) option. Unlike the original TCP/IP header compression, SCPS-TP compresses only the TCP header end-to-end to avoid any problems caused by changing connectivity. The decompression of headers are independent from each other and thus each packet could be processed separately. In addition to the above benefits, header compression also helps reduce the overhead of ACK packets. To further improve the utilization of bandwidth, SCPS-TP uses the SNACK option that is a combination of the TCP SACK (selective acknowledgment) and a NAK (negative acknowledgment) [18] option. While dealing with asymmetric channels, SCPS-TP relaxes the ACK requirement in TCP. Instead of acknowledging at least every other packet received, SCPS-TP delays the ACKing for a certain interval of time that is a function of the estimated RTT.

V. SIMULATION RESULTS

In this section, we describe our simulation model and the results obtained for each experiment. Due to the lack of needed modules in the standard ns-3 distribution, in addition to using the already existing TCP NewReno module, we implemented TCP Westwood and SCPS-TP. The performance of AeroTP-FEC and AeroTP-ARQ modes are compared through various experiments with the other transport protocols. In each experiment, we compare performance based on network metrics such as average goodput and delay.

E. Simulation Model

To test the behavior of each protocol in a lossy network, we use a simple simulation topology consisting of a single sender and a single receiver interconnected to each other through a router at either end. A bulk transfer application is used as the traffic generating source for TCP protocols and a constant bit rate application with an application data rate equal to the link rate is used for AeroTP. The path consists of 5 Mb/s links that are prone to errors. For each simulation between the two endpoints, a data file of 1 MB is transmitted. Each simulation case has a 20000 s duration and is run 20 times to increase confidence in results.

We simulate two different scenarios for comparing the performance of the different protocols. In the first scenario, errors are introduced in the link with varying probabilities from 10^{-8} to very high error rates of 10^{-4} using the ns-3 rate error model. In the second scenario, the bottleneck link delay is varied over a range of parameters starting from 0 ms to 500 ms to signify typical delay parameters ranging from LAN to geosynchronous satellites. As part of this scenario, we also include performance with different BER values for the delay parameters considered.

F. Varying levels of channel loss

For our first experiment, we compare the performance of AeroTP-FEC and AeroTP-ARQ modes with TCP NewReno, TCP Westwood, and SCPS-TP in the presence of varying BER. We observe that the goodput of AeroTP-FEC is almost unperturbed until 10^{-5} and then gradually decreases as shown in Figure 3.

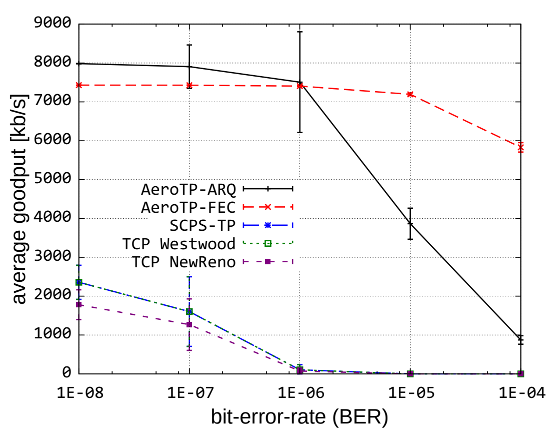

Figure 3: Average goodput comparison

AeroTP-ARQ goodput on the other hand, drops substantially with increasing BER. This is mainly attributed to the basic difference between ARQ and FEC. FEC employs an open loop control by adding extra bits with the transmitted segments. ARQ on the other hand, uses closed loop control and sends feedback to the sender through the ACK mechanism. In a high loss channel, there is an increased number of retransmissions resulting from this feedback, while FEC tries to correct errors with the help of the

added bits. Thus in such cases, the ARQ mechanism contributes to high overhead in the networks and delays caused by the ACKs and retransmissions. It should also be noted that in such lossy channels, there are chances that ACKs get lost. However, both modes of AeroTP perform much better than NewReno, Westwood, and SCPS-TP. We also observe that both TCP variants cease transmission at the 10^{-6} BER level, and SCPS-TP has a similar behavior due to the fact that it is an extension of TCP.

Similar trends are observed for the end-to-end delays over varying BER as shown in Figure 4. AeroTP-ARQ and AeroTP-FEC perform the best by a large margin with almost no delays experienced over the BER range. TCP Westwood, TCP NewReno, and SCPS-TP experience high delays starting from 10^{-6} and increase exponentially beyond a BER of 10^{-5}.

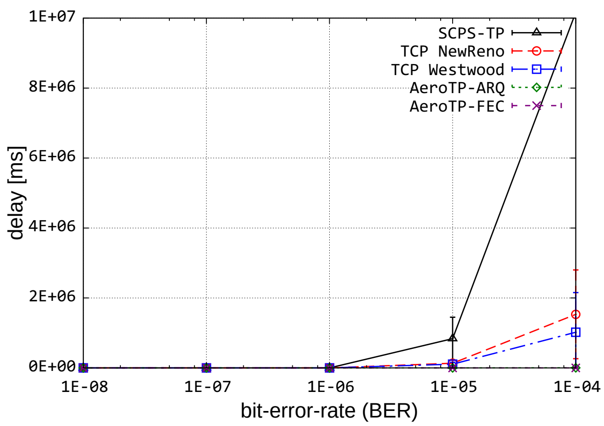

Figure 4: Average delay comparison

G. *Varying levels of link delay*

In this scenario, the intermediate link delay is varied over a range of values from 0 ms to 500 ms. The different delays are considered, based on the geographic scope of different kinds of networks ranging from LANs to geosynchronous satellites. The results are separately plotted for the various protocols over varying BER.

Figures 5 and 6 provide the goodput characteristics of FEC and ARQ modes of AeroTP, respectively. Figures 7 and 8 show the goodput characteristics of Westwood and NewReno. An overall trend of decreasing goodput is observed for increasing link delays for the various BER values. The plots also show that in all cases, the two modes of AeroTP outperforms both TCP variants. The superior performance is clearly observed in the FEC mode with a minimum achieved goodput of approximately 2900 kb/s. In the presence of low BER, due to the large packet overhead, AeroTP-FEC goodput starts to drop at 100 ms delay (Figure 5) while AeroTP-ARQ is able to maintain high goodput until the delay reaches 300 ms (Figure 6). However, when the BER is at the extreme case of 10^{-4}, due to the large number of retransmissions coupled with high delay, AeroTP-ARQ suffers more goodput degradation than AeroTP-FEC, but still performs better than Westwood and NewReno that are shown to breakdown completely in this case.

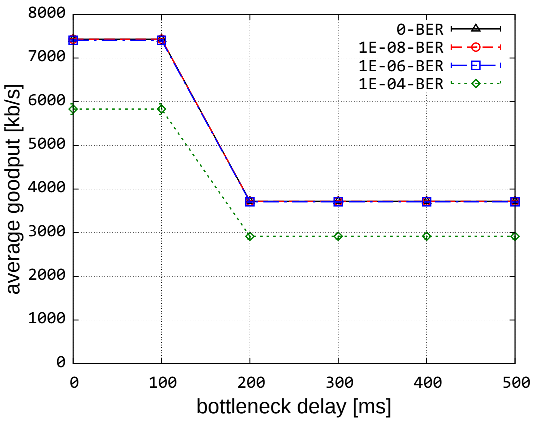

Figure 5: Average goodput of AeroTP-FEC mode

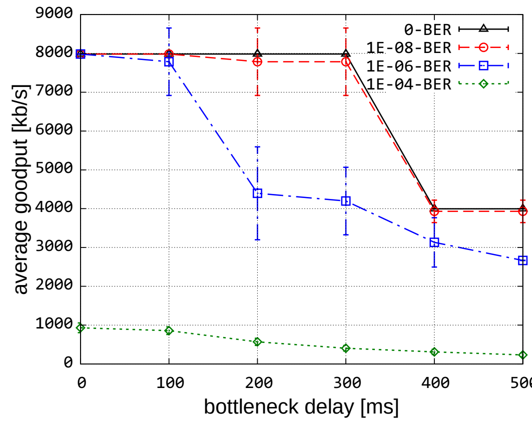

Figure 6: Average goodput of AeroTP-ARQ mode

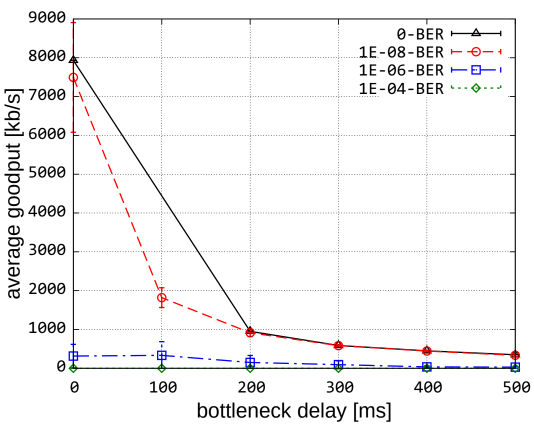

Figure 7: Average goodput of TCP Westwood

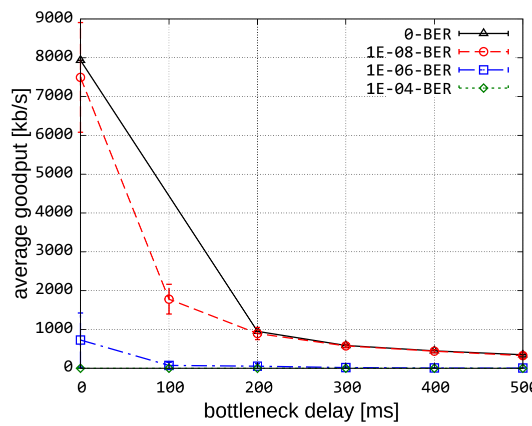

Figure 8: Average goodput of TCP NewReno

VI. CONCLUSIONS AND FUTURE WORK

In this paper the AeroTP-FEC and AeroTP-ARQ modes have been compared with popular TCP variants such as TCP NewReno, Westwood, and SCPS-TP over different BER and link delay ranges using ns-3. It is observed that both AeroTP modes either outperform or are at least equal to the considered TCP protocols. The performance is seen to be similar only in the extreme case of 10^{-4} when very little communication is possible. In the future we will conduct extended testing of the AeroTP modes with the inclusion of the recently developed hybrid ARQ mechanism [19].

ACKNOWLEDGEMENTS

The authors would like to thank the Test Resource Management Center (TRMC) Test and Evaluation/Science and Technology (T&E/S&T) Program for their support. This work was funded in part by the T&E/S&T Program through the Army PEO STRI Contracting Office, contract number W900KK-09-C-0019 for AeroNP and AeroTP: Aeronautical Network and Transport Protocols for iNET (ANTP). The

Executing Agent and Program Manager work out of the AFFTC. This work was also funded in part by the International Foundation for Telemetering (IFT). We would like to thank Kip Temple and the membership of the iNET working group for discussions that led to this work.

REFERENCES

[1] J. P. Rohrer, A. Jabbar, E. K. Çetinkaya, and J. P. Sterbenz, "Airborne telemetry networks: Challenges and solutions in the ANTP suite," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, (San Jose, CA), pp. 74–79, November 2010.

[2] J. P. Rohrer, E. Perrins, and J. P. G. Sterbenz, "End-to-end disruption-tolerant transport protocol issues and design for airborne telemetry networks," in *Proceedings of the International Telemetering Conference (ITC)*, (San Diego, CA), October 2008.

[3] J. P. Rohrer, A. Jabbar, E. K. Çetinkaya, E. Perrins, and J. P. Sterbenz, "Highly-dynamic cross-layered aeronautical network architecture," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, pp. 2742–2765, October 2011.

[4] K. S. Pathapati, T. A. N. Nguyen, J. P. Rohrer, and J. P. Sterbenz, "Performance analysis of the AeroTP transport protocol for highly-dynamic airborne telemetry networks," in *Proceedings of the International Telemetering Conference (ITC)*, (Las Vegas, NV), October 2011.

[5] "The ns-3 network simulator." `http://www.nsnam.org`, July 2009.

[6] R. Krishnan, J. P. G. Sterbenz, W. M. Eddy, C. Partridge, and M. Allman, "Explicit transport error notification (ETEN) for error-prone wireless and satellite networks," *Computer Networks*, vol. 46, no. 3, pp. 343–362, 2004.

[7] S. Lin, D. J. C. Jr., and M. J. Miller, "Automatic-Repeat-Request Error-Control Schemes," *Communications Magazine, IEEE*, vol. 22, pp. 5–17, December 1984.

[8] M. Watson, M. Luby, and L. Vicisano, "Forward Error Correction (FEC) Building Block." RFC 5052 (Proposed Standard), Aug. 2007.

[9] J. C. Hoe, "Improving the start-up behavior of a congestion control scheme for TCP," in *SIGCOMM '96: Conference proceedings on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 270–280, ACM, 1996.

[10] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm." RFC 2582 (Experimental), Apr. 1999. Obsoleted by RFC 3782.

[11] V. Jacobson, "Modified TCP congestion avoidance algorithm," April 1990.

[12] V. Jacobson and R. Braden, "TCP extensions for long-delay paths." RFC 1072, Oct. 1988. Obsoleted by RFCs 1323, 2018.

[13] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options." RFC 2018 (Proposed Standard), Oct. 1996.

[14] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: End-to-end congestion control for wired/wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 467–479, 2002.

[15] S. Mascolo, L. Grieco, R. Ferorelli, P. Camarda, and G. Piscitelli, "Performance evaluation of Westwood+ TCP congestion control," *Performance Evaluation*, vol. 55, pp. 93–111, Jan. 2004.

[16] R. C. Durst, G. J. Miller, and E. J. Travis, "TCP extensions for space communications," in *MobiCom '96: Proceedings of the 2nd annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 15–26, ACM Press, November 1996.

[17] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 24–35, 1994.

[18] R. Fox, "TCP big window and NAK options." RFC 1106, June 1989.

[19] K. S. Pathapati, J. P. Rohrer, and J. P. Sterbenz, "Comparison of adaptive transport layer error-control mechanisms for highly-dynamic airborne telemetry networks," in *Proceedings of the International Telemetering Conference (ITC)*, (San Diego, CA), October 2012.