# End-to-End ARQ: Transport-Layer Reliability for Airborne Telemetry Networks

Kamakshi Sirisha Pathapati, Justin P. Rohrer, and James P.G. Sterbenz
Department of Electrical Engineering and Computer Science
Information and Telecommunication Technology Center
The University of Kansas
Lawrence, KS 66045
{kamipks|rohrej|jpgs}@ittc.ku.edu

## ABSTRACT

Due to the mission-critical nature of command-and-control traffic in the telemetry environment, it is imperative that reliable transfer be supported. The AeroTP disruption-tolerant transport protocol is intended for this environment. The mechanism for reliable transfer is ARQ with end-to-end acknowledgments. This has significant performance limitations resulting from the highly-dynamic nature of airborne telemetry networks, since end-to-end paths may not persist long enough for retransmissions to be received. We use ns-3 to analyze the AeroTP ARQ mechanism, along with tunable parameters that may improve performance in reliable transfer mode.

## I. INTRODUCTION AND MOTIVATION

The are several possible mechanisms for providing reliability in network environments. One is to retransmit lost packets, another is to provide forward error correction to correct altered bits in the data, and the third is to use erasure block coding to recover lost packets. The AeroTP protocol or highly-dynamic airborne networks [1] uses all of these mechanisms, depending on the type of reliability required. In this paper we are interested in the first mechanism, ARQ, and in evaluating the effect that the challenges of highly-dynamic network have on itsperformance. In addition to the selection of mechanism, the layer at which this functionality is to be located must be chosen. These mechanisms may be applied hop-by-hop, edge-to-edge, or end-to-end, and in this paper we take a look at the performance of end-to-end ARQ.

Within the telemetry environment there are a number of variables to be taken into consideration when selecting end-to-end reliability modes. These include the requirements of the data being transferred, the stability of the paths over which the data is being transferred, the scarcity of available bandwidth along those paths, and the round-trip delay between the source and destination. ARQ is a closed-loop mechanism, and therefore requires a stable path in both the forward and reverse directions. If these are not present, significant performance degradation is usually the result. The advantage to the ARQ mechanism is that it is the only way to guarantee reliable delivery [2]. Given that the cost of this reliability can be significant, particularly in high-dynamic network environments it is important to understand the available options and tradeoffs involved. If full-reliability is not required and near-reliability is acceptable, then the

ARQ loop may be split using a custody transfer approach, which can significantly increase efficiency. If only statistical reliability is required, then open-loop FEC approaches become an attractive option.

Traditional Internet protocols offer only two options: no reliability (UDP), or full-reliability (TCP). In this paper we explore the performance of TCP in lossy environments, as well as showing an alternative ARQ algorithm that decouples the reliability mechanism from flow control.

## II. BACKGROUND AND RELATED WORK

Ideally reliable data transfer transmits data end-to-end with no delay and with no errors or losses. But, transmission in a network is often prone to delay, limited bandwidth, and multiple errors along the path towards the destination. Bit errors are the most common in wireless channels because of the channel's vulnerability to noise and interference. Packet errors are caused because of congestion, switching between multiple paths within the network, and packet-drops during the occurrence of bit errors in the packet. To avoid the errors caused by congestion, congestion control and avoidance algorithms are used. When implicit congestion notification is used they reduce the window size each time congestion is detected. Packet drops at the receiver may be caused because of corrupted packets. Error recovery schemes are often a solution to correct the errors in the received data packet. ARQ uses ACKs and retransmissions to ensure all the lost packets are successfully delivered to the destinations.

With rapid increase in wireless technologies, high bandwidth-$\times$-delay product networks are becoming increasingly common. These networks pose new challenges that worsen when the network is highly asymmetric. Asymmetry arises when there is a difference in the power used by the sender and the receiver units for transmitting information. The central transmission unit, such as a base station, has a higher transmission power compared to the individual mobile units in order to reduce power consumption. For such networks in which there is bi-directional traffic flow asymmetry makes optimal performance much more difficult to attain.

### *A. Transport Protocols*

The transport layer is responsible for delivering application data between end-system hosts through network switches or routers. It offers many services such as reliable or unreliable data delivery, connection-oriented or connectionless, flow control, and error control services.

Two of the earliest and the most commonly used protocols are the transport control protocol (TCP) [3] and the user datagram protocol (UDP) [4]. TCP and UDP with modified or added functionalities have become the basis for many protocols to offer better services and perform efficiently in challenged environments, and they continue to be used as the basis for many of the currently proposed transport protocols.

### *B. Transmission Control Protocol and User Datagram Protocol*

TCP is a full-duplex, connection-oriented, end-to-end, reliable protocol that provides a byte-stream service. TCP was designed to operate for a wide range of communication systems based on packet-switched networks [3]. It offers its services to the application protocols that belong to the upper layer and it demands addressing, forwarding, and routing services from the lower layer Internet protocol (IP).

To provide a standard communication service between two processes in a network, TCP was designed with multiple features. TCP's operation offers a reliable data service that allows the transmitted data to recover from damaged, lost, duplicated, or out-of-order TCP segments during the segment's transmission through the network. It achieves reliability by using acknowledgements (ACKs) from the receiver. It retransmits each segment if an ACK is not received in a set period of time (timeout). It is a connection-oriented protocol that uses a 3-way handshake mechanism to explicitly establish a connection between two hosts and terminate it when the transmission is completed. It takes 1.5 RTTs to set up the connection between the sender and the receiver after which actual data transmission takes place. It implements flow control as an end-to-end mechanism that limits the amount of data transmitted by the sender at a given time to avoid choking the receiver. TCP's congestion control mechanism prevents the sender from injecting too much data into the network causing congestion. The congestion overloads the switches or routers in the network and causes the performance to degrade drastically. Another important feature of TCP is that it guarantees ordered delivery of data by having the receiver maintain buffer to store the packets in case any arrive out of order.

UDP is a protocol with minimal end-to-end message delivery mechanism for the application programs [4]. UDP is unreliable, which means there is no guarantee that the data sent will be delivered to the destination. It does not implement flow control, congestion control, or ordered delivery.

### *C. Drawbacks of Traditional Protocols*

Although TCP and UDP are the most commonly used transport protocols they fail to perform efficiently in a challenged wireless environment. In wireless networks packet losses are inevitable; link outages, lossy channel characteristics, unstable connectivity, delay, and congestion are a few examples of challenges that cause packet loss. A wireless channel is often subjected to interference and channel fading, resulting in packet loss and packet corruption. TCP assumes every packet loss is caused by congestion in the network and invokes its congestion control algorithm. This decreases the congestion window by a fraction (usually half) each time reducing the congestion window size, and thus causing inefficient use of bandwidth. Schemes such as split-TCP connections and local retransmissions were developed to circumvent the problem caused by TCP's assumption of congestion being the only cause for packet loss [5].

TCP uses ACKs to provide reliable data transmission and retransmissions. The source retransmits a TCP segment to the destination when a timeout occurs while waiting for an ACK. A connection setup is performed through a three-way handshake between the source and the destination pair of nodes. This takes up 1 round-trip time (RTT) before data may be sent, which causes significant performance degradation in a telemetry network because of short contact duration between nodes. By using a slow start algorithm, TCP takes many RTTs to ramp up the sending rate before it can fully utilize the available bandwidth. This results in a significant amount of waisted capacity in an environment which often has episodic connectivity.

TCP does not efficiently perform flow control in a network with asymmetric links since it requires a highly reliable ACK stream. Because of dynamic topology, link outages are common. The congestion control algorithm is invoked during short link outages, causing an increase in the number of retransmissions. The connection is terminated in case of longer link outages. This causes difficulty in restoring links and finding alternate paths to the destination [6]. TCP also does not provide any QoS differentiation for prioritizing the type of data being transmitted.

SCPS-TP (Space Communications Protocol Standards Transport Protocol) [7] is an extension to TCP, used particularly for satellite communications, developed to address problems posed by asymmetric links. SCPS-TP addresses some similar problems to those of telemetry networks although it is not fully suitable for telemetry applications. This is in part because it relies on channel condition information which is either pre-configured or discovered over time from the network [8].

Although UDP is a simpler protocol than TCP, it does not offer any guarantee for guaranteed data delivery, so it is unreliable. Unlike TCP, UDP does not have connection set-up mechanism and does not provide congestion control or flow control or data retransmissions. UDP also does not provide differentiated levels of precedence or QoS for the classes of data available in the telemetry environment.

### *D. Optimizations for Mobile Wireless Networks*

TCP assumes that congestion is the only reason for the loss of data segments in the network. To prevent congestion collapse, aggressive congestion control and avoidance algorithms were developed, which made TCP adapt to congestion by decreasing the size of its congestion window substantially. This mechanism has proven very effective in wired networks. As networks have evolved to included various wireless link types as well, the possibility of losses due to causes other than congestion has increased. Factors such as high link error rates, channel fading, interference, long propagation delays, and noisy channel conditions increasingly become the reason for packet losses. Hence, decreasing the congestion window size for each loss detected in a wireless network is the wrong approach and results in a major drop in the overall utilization of the network, as shown later in our simulations.

Active research to find out other ways to deal with losses in a wireless network has spurred development of newer algorithms. TCP Peach [9] and TCP Westwood [10] are two such algorithms developed for wireless networks. TCP Peach was developed as a congestion control scheme for satellite IP networks. Satellite networks are often characterized by long propagation delays and high error rate channels. It was necessary that the algorithm could differentiate when the loss occurred due to congestion or corruption. It introduced two new algorithms, sudden start and rapid recovery, along with traditional congestion avoidance and fast retransmit algorithms. TCP Peach performed better in terms of throughput and also provided an overall fair share of network resources compared to traditional TCP algorithms [11].

TCP Westwood was developed to improve the performance TCP in both wired and wireless environments. TCP Westwood makes use of end-to-end bandwidth estimation to discriminate the cause of packet loss in the network. It calculates the rate of connection continuously at the TCP sender side and computes the congestion window threshold and slow start threshold. It monitors the rate by tracking the rate of returning ACKs [12]. The main advantage of using TCP Westwood is that the only modifications to the TCP algorithm are at the sender side. Improvements in throughput and fair usage of link capacity are other advantages.

Other techniques have been designed to improve the performance of TCP in wireless environments [5]. Three different techniques were employed to improve the performance of TCP. The first technique involves a direct end-to-end protocol implementation in which the sender is responsible for error recovery. The error recovery is performed using TCP SACK and explicit loss notification (ELN) mechanisms. The second technique provides link-layer reliability and the third technique implements a split-connection protocol, where the end-to-end connection breaks at the base station. The results show providing local reliability at the link-layer that is TCP-aware improves TCP performance in wireless networks.

A detailed study of the effects of asymmetry on performance of TCP in a network has been performed by [13]. Techniques such as ACK congestion control (ACC), ACK filtering (AF) used to control the frequency of ACKs, TCP sender adaptation (SA), ACK reconstruction (AR) and scheduling data and ACKs were developed to minimize the number of ACKs to counter the problem of asymmetry.

The performance of the TCP protocol also was evaluated in a network with high bandwidth-delay product and random loss [14]. TCP showed deterioration in the throughput performance when random losses occurred and also was unfair towards connections with larger round-trip times (RTTs) when multiple connections share a bottleneck link.

### *E. ARQ Algorithms*

ARQ algorithms improved over time, achieving better performance in terms of bandwidth utilization, delay, and reliability. The simplest approach in achieving reliability is through the stop-and-wait algorithm. The algorithm employs a feedback mechanism in which the sender is notified of the delivery of the packet. In this approach, the sender transmits a packet to the receiver and waits for an ACK. In case the sender does not receive an ACK for a packet, which might be because of lossy link characteristics or packet-drops, the sender waits for a timer to expire, after which it retransmits the un-ACKed packet. This causes the link to be idle for the entire time the sender is waiting for an ACK causing inefficient utilization of available bandwidth and a delay of one RTT per packet.

An alternate is the pipelined go-back-$N$ algorithm. Multiple packets are sent simultaneously to the destination and the sender waits for all the ACKs. Once the sender misses an ACK for a single packet a retransmission of all the packets since the lost packet occurs. Although this eliminates the round-trip delay caused by waiting for an ACK for each packet, it introduces the delay caused by retransmitting packets since the loss. Fast retransmit is an optimized go-back-$N$ algorithm in which the sender retransmits packets even before the timer expires. Retransmission occurs when the sender receives more than a certain number of duplicate ACKs. This algorithm recovers quickly from lost ACKs but it still faces similar problems as go-back-$N$.

An alternative to the fast retransmit algorithm is the selective repeat ARQ. In this algorithm, retransmission of only those packets for which the sender did not receive an ACK takes place. When the packets are retransmitted they arrive out of sequence. Hence, the receiver maintains a buffer to store the packets to rearrange them at the end of the entire session. The algorithm's complexity increases since both the sender and the receiver have to maintain a consistent state throughout the session and the receiver must have an increased buffer size. It also fails in the case too many packet losses occur during transmission if there is limited space for ACK blocks, as in the case of TCP SACK [15]. Another version of selective repeat ARQ sends ACKs for a group of packets instead of a single packet each time. This reduces the overall complexity at the receiver buffer space and corrects the behavior of the protocol during multiple packet losses. Selective negative acknowledgement (SNACK) is an alternative to SACK in which the receiver sends ACKs requesting a damaged or lost packet. The receiver explicitly notifies the sender which packets were lost or corrupted and thus may need to be retransmitted. TCP SNACK was originally implemented in satellite communications in which the end-to-end delay was long. SNACK provides the sender with a complete view of the receiver buffer when the sender receives an ACK specifying damaged or lost packets. The sender aggressively sends packets that are lost without waiting for a timeout. In this case TCP congestion control is not invoked and utilization of bandwidth is improved.

## III. NETWORKING CHALLENGES IN AIRBORNE TELEMETRY NETWORKS

A typical T&E (test and evaluation) telemetry network consists of three types of nodes: test articles (TA), ground stations (GS), and relay nodes (RN). The TAs are the airborne nodes involved in the test and contain several data collection devices that are IP devices (e.g. cameras) called *peripherals*. TAs house omnidirectional antennas with relatively short transmission range. The GSs are located on the ground and typically have a much higher transmission range than that of a TA through the use of large steerable antennas. In point-to-point communication mode, the GS tracks a given TA across some geographical space in a test range. However, due to the narrow beam width of the antenna, a GS can only track one TA at any given time. The GS also houses a gateway (GW) that connects the telemetry network to the Internet and several terminals that may run control applications for the various devices on the TA. Furthermore, the GSs can be interconnected to do soft-handoffs from one to another while tracking a TA. The RNs are dedicated airborne nodes to improve the connectivity of the network. These nodes have enhanced communication resources needed to forward data from multiple TAs and can be arbitrarily placed in the network. The flow of T&E information is primarily from the TAs to the ground stations GSs, however command and control data flows in the reverse direction. There are a number of challenges to communication protocols in this environment:

- *Mobility*: The test articles can travel at speeds as high as Mach 3.5; the extreme is then two TAs closing with a relative velocity of Mach 7. Because of high speeds, the network is highly dynamic with constantly changing topology.
- *Constrained bandwidth*: Due to the limited spectrum allocated to T&E and the high volume of data that is sent from TA to GS, the network in general is severely bandwidth constrained.
- *Limited transmission range*: The energy available for telemetry on a TA is limited due to power and weight constraints of TA telemetry modules, requiring multi-hop transmission from TA to GS.
- *Intermittent connectivity*: Given the transmission range of the TA and high mobility, the contact duration between any two nodes may be extremely short leading to network partitioning. Furthermore, the wireless channels are subject to interference and jamming.

The result of these challenges is that end-to-end paths may be available only for a few seconds, or not at all.

### *F. AeroTP Reliability Mechanisms*

In this section we review the reliability modes of AeroTP, which are described in greater detail in [8]. Based on the application requirements, there will be a number a classes of data being transmitted over the telemetry network. For this reason we define multiple *transfer modes* that are mapped to different traffic classes. All modes except unreliable datagram are connection-oriented for TCP-friendliness and will use byte sequence numbers for easy translation to TCP at the AeroGW, so that packets may follow varying or multiple paths and be reordered at the receiver-side gateway.

- ***Reliable connection*** mode must preserve end-to-end acknowledgement semantics from source to destination as the only way to *guarantee* delivery. We do this using TCP ACK passthrough, which has the disadvantage of imposing TCP window and ACK timing onto the AeroTP realm, but will never falsely inform the source of successful delivery.
- ***Near-reliable connection*** mode uses a custody transfer mechanism similar to that used in DTNs [16] to provide high reliability, but can not *guarantee* delivery since the gateway immediately returns

TCP ACKs to the source on the assumption that AeroTPs reliable ARQ-based delivery will succeed using SNACKs (selective negative acknowledgements) [7] supplemented by a limited number of (positive) ACKs. This still requires that the gateway buffer segments until acknowledged across the telemetry network by AeroTP, but is more bandwidth-efficient than full source–destination reliability. However, the possibility exists of confirming delivery of data that the gateway cannot actually deliver to its final destination.

- ***Quasi-reliable connection*** mode eliminates ACKs and ARQ entirely, using only open-loop error recovery mechanisms such as FEC (Forward Error Correction) or erasure coding, across multiple paths if available [17]. In this mode the strength of the coding can be tuned using cross-layer optimizations based on the quality of the wireless channel being traversed, available bandwidth, and the sensitivity of the data to loss. This mode provides an arbitrary level of statistical reliability but without absolute delivery guarantees.
- ***Unreliable connection*** mode relies exclusively on the FEC of the link layer to preserve data integrity and does not use any error correction mechanism at the transport layer. Cross-layering may be used in future work to vary the strength of the link-layer error-correcting code.
- ***Unreliable datagram*** mode is intended to transparently pass UDP traffic, and no AeroTP connection state is established at all.

In the simulations presented in this paper are focused on the performance of the reliable connection mode.

## IV. SIMULATIONS AND RESULTS

We compare the performance of AeroTP in the reliable connection mode with TCP and UDP protocols using the ns-3 open-source simulator [18]. The selective-repeat ARQ algorithm is used to provide reliable edge-to-edge connection between nodes for the AeroTP protocol. The network in this simulation setup consists of two nodes communicating via a wired link that is prone to losses. One node is configured as a traffic generator, and the other as a traffic sink. The traffic generator sends data at a constant data rate of 4.416 Mb/s (3000 packets/s with an MTU of 1500 B). The path consists of a 10 Gb/s link representing the LAN on the TA, a 5 Mb/s link with a latency of 10s representing the mobile airborne network, and a second 10 Gb/s link representing the LAN at the ground station. Bit-errors are introduced in the middle link with a fixed probability for each run, and the performance for each probability of bit-errors is shown in the plots described in the next section. A total of 1 MB of data is transmitted during one single simulation between the two nodes. The link is made unreliable by introducing losses using an error model with varying bit-error probabilities ranging for 0 to 0.0001 for each of the protocols. Since none of the protocols in this mode use FEC, any packet experiencing bit-errors are dropped. Each simulation is run 10 times to obtain the results needed for comparison.

Figure 1 shows that AeroTP is able to achieve significantly better performance than TCP, which backs off substantially as the BER (bit-error rate) increases. At the same time TCP's end-to-end delay doubles with a BER of $1 \times 10^{-5}$, as shown in Figure 2. Over the course of the simulation, both TCP and AeroTP are able to deliver the full amount of data (1 MB) transmitted, however UDP looses a percentage of the

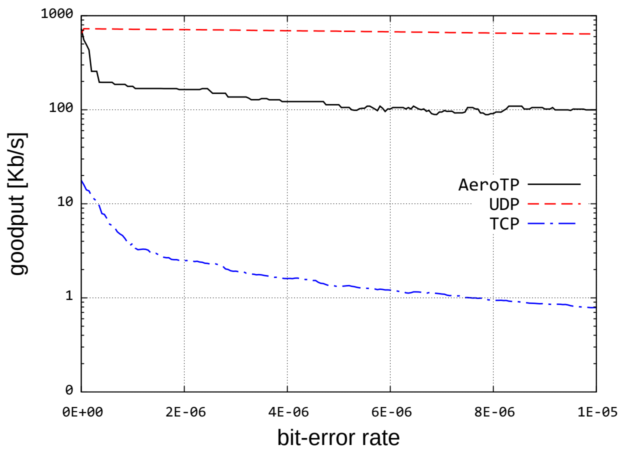


Figure 1: Average goodput

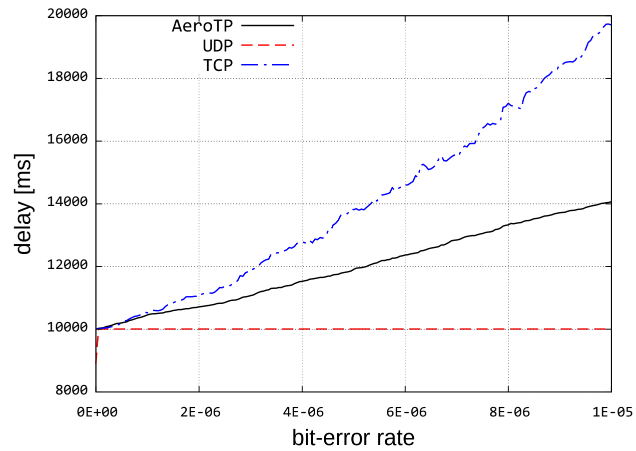


Figure 2: Average delay

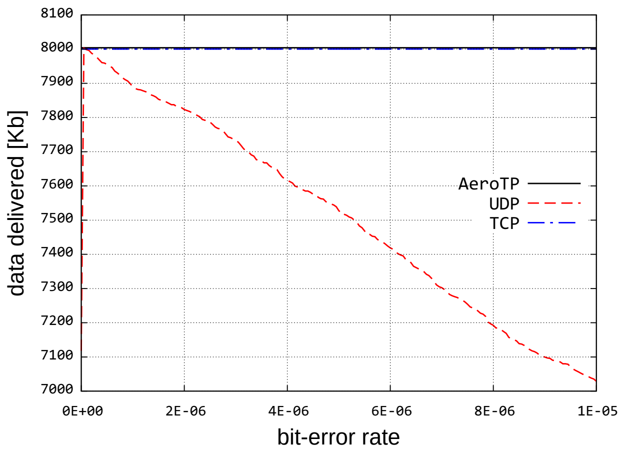


Figure 3: Cumulative goodput

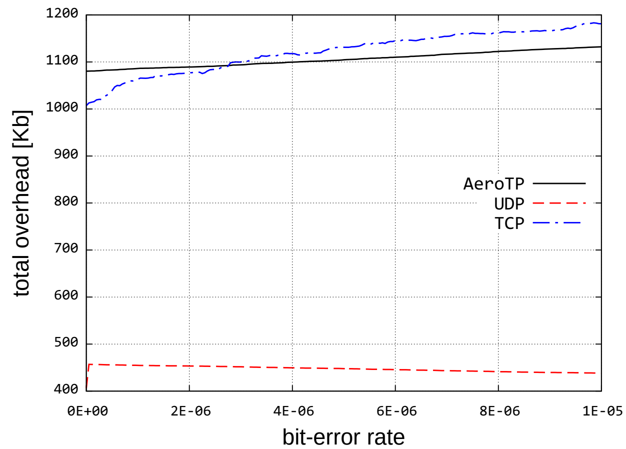


Figure 4: Cumulative overhead

data due to corruption and the BER increases as shown in Figure 3. Lastly, we see that these performance characteristics are achieved with a cost in overhead comparable to that of TCP in Figure 4.

## V. CONCLUSIONS AND FUTURE WORK

The traditional TCP and UDP protocols, although widely used, do not meet the performance expectations of a T&E environment. These protocols do not offer any QoS or differentiated levels of precedence for various kinds of data that exist in a T&E environment. Certain classes of data in this environment require reliable data delivery service. The AeroTP protocol in the reliable-connection mode is capable of guaranteeing delivery of data without significant performance degradation. In this paper, we implemented the AeroTP reliable-connection mode in ns-3 using selective-repeat ARQ mechanism for a simple two node network with losses introduced in the link. We can see that with increasing bit-error rates TCP fails to achieve a desirable goodput although it delivers all the packets using retransmission, and UDP fails to deliver packets to the destination. AeroTP performs better than both TCP and UDP in terms of both the average and cumulative goodput. In the future, we will test and compare the performance of these protocols in a much more complex scenarios and mobility models. In the future we will combine the AeroTP simulations with the the Gauss-Markov random mobility model [19] and use the AeroRP routing protocol [20] in the network to test the performance of AeroTP protocol in the reliable-connection mode. We will also compare the performance of reliable-connection mode against the other reliability-modes to observe the performance in terms of latency, overhead, and channel utilization.

## ACKNOWLEDGMENTS

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). This work was also funded in part by the International Foundation for Telemetering (IFT). We would like to acknowledge the ResiliNets group members Abdul Jabbar and Hemanth Narra for useful discussions on this work. We would also 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, 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*, (San Diego, CA), October 27–30 2008.

[2] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Transactions Computer Systems*, vol. 2, no. 4, pp. 277–288, 1984.

[3] J. Postel, "Transmission Control Protocol." RFC 793 (Standard), Sept. 1981. Updated by RFCs 1122, 3168.

[4] J. Postel, "User Datagram Protocol." RFC 768 (Standard), Aug. 1980.

[5] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 756–769, 1997.

[6] J. P. Rohrer, A. Jabbar, E. Perrins, and J. P. G. Sterbenz, "Cross-layer architectural framework for highly-mobile multihop airborne telemetry networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, (San Diego, CA, USA), pp. 1–9, November 2008.

[7] 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.

[8] J. P. Rohrer and J. P. G. Sterbenz, "Performance and disruption tolerance of transport protocols for airborne telemetry networks," in *International Telemetering Conference (ITC) 2009*, (Las Vegas, NV), October 2009.

[9] I. Akyildiz, G. Morabito, and S. Palazzo, “TCP-Peach: a new congestion control scheme for satellite ip networks,” *Networking, IEEE/ACM Transactions on*, vol. 9, no. 3, pp. 307–321, Jun 2001.

[10] 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.

[11] I. F. Akyildiz, G. Morabito, and S. Palazzo, “Tcp-peach: a new congestion control scheme for satellite ip networks,” *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 307–321, 2001.

[12] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, “Tcp westwood: Bandwidth estimation for enhanced transport over wireless links,” in *MobiCom ’01: Proceedings of the 7th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 287–297, ACM, 2001.

[13] Balakrishnan, Hari and Katz, Randy H. and Padmanbhan, Venkata N., “The effects of asymmetry on tcp performance,” *Mob. Netw. Appl.*, vol. 4, no. 3, pp. 219–241, 1999.

[14] T. V. Lakshman and U. Madhow, “The performance of TCP/IP for networks with high bandwidth-delay products and random loss,” *IEEE/ACM Trans. Netw.*, vol. 5, no. 3, pp. 336–350, 1997.

[15] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, “An Extension to the Selective Acknowledgement (SACK) Option for TCP.” RFC 2883 (Proposed Standard), July 2000.

[16] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, “Delay-tolerant networking: an approach to interplanetary internet,” *IEEE Communications Magazine*, vol. 41, pp. 128–136, June 2003.

[17] A. J. McAuley, “Reliable Broadband Communication Using a Burst Erasure Correcting Code,” *SIGCOMM Comput. Commun. Rev.*, vol. 20, no. 4, pp. 297–306, 1990.

[18] “The ns-3 network simulator.” http://www.nsnam.org, July 2009.

[19] D. Broyles, A. Jabbar, and J. P. G. Sterbenz, “Design and analysis of a 3–D gauss-markov mobility model for highly-dynamic airborne networks,” in *Proceedings of the International Telemetering Conference (ITC)*, (San Diego, CA), October 2010. (to appear).

[20] K. Peters, E. K. Çetinkaya, and J. P. G. Sterbenz, “Analysis of a geolocation-assisted routing protocol for airborne telemetry networks,” in *Proceedings of the International Telemetering Conference (ITC)*, (San Diego, CA), October 2010. (to appear).