Performance and Disruption Tolerance of Transport Protocols for Airborne Telemetry Networks

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
{rohrej|jpgs}@ittc.ku.edu

ABSTRACT

The airborne telemetry environment presents unique challenges to end-to-end communications due to the highly dynamic topology and time-varying connectivity of high-velocity wireless nodes. The AeroTP transport protocol uses multiple reliability modes to trade off end-to-end reliability and efficiency as appropriate for different categories of telemetry data. Based on the architecture previously presented, we have further developed the design of this protocol, as well as performing preliminary simulations of AeroTP using the ns-3 simulation platform. In this paper we present the operational modes of AeroTP in greater detail, as well as comparing the performance of TCP with the AeroTP domain-specific transport protocol.

INTRODUCTION AND MOTIVATION

Telemetry for airborne test and evaluation is an application that poses unique challenges. Traditionally, telemetry communication has consisted primarily of point-to-point links from multiple sources to a single sink. More recently, with the increasing number of sources in the typical telemetry test scenario, there is a need to move to networked systems in order to meet the demands of bandwidth and connectivity. This need has been recognized by various groups, including the Integrated Network Enhanced Telemetry (iNET) program for Major Range and Test Facility Bases (MRTFB) across United States [1].

The current TCP/IP-based Internet architecture is not designed to address the needs of telemetry applications [2] and there remain a number of issues to be solved at the network and transport layers [3]. Given the constraints and requirements of the aeronautical environment, the current Internet protocols are not suitable in a number of respects. These constraints include the physical network characteristics such as topology and mobility that present severe challenges to reliable end-to-end communication. In order to build a resilient network infrastructure, we need cross-layer enabled protocols at the transport, network, and MAC layers that are particularly suited for the airborne telemetry networks. At the same time, there is a need to be compatible with both TCP/IP-based devices located on the airborne nodes as well as with the control applications. Therefore, a new protocol suite, while being specific to the aeronautical telemetry environment, must also be fully interoperable with TCP/UDP/IP via gateways at the telemetry network edges [4]. Due to the limited bandwidth in telemetry networks and a priori knowledge of communication

needs of a given test, the iNET community is developing a TDM (time division multiplex)-based MAC for this particular environment [5]. This paper extends the design and evaluation of a transport protocol for this environment: *AeroTP* – a TCP-friendly transport protocol introduced in [6], and further developed in [7] with multiple reliability and QoS modes. When finalized, the protocol is intended to operate cooperatively with AeroNP network and AeroRP routing protocols [8, 9]. Previous research has demonstrated that domain-specific information can dramatically improve protocol performance [10]. However, in order to achieve this, we need to facilitate cross-layering across the multiple layers. Strict layering in the network stack is not particularly suitable for wireless networks due to mobility, limited bandwidth, low energy, and QoS requirements. Therefore, it is commonly agreed upon that a tighter, more explicit, yet careful integration amongst the layers will improve the overall wireless network performance in general; and in the case of highly-dynamic, bandwidth-constrained networks may provide the only feasible solution that meets the requirements of telemetry applications.

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 primarily 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.

EXISTING TRANSPORT PROTOCOLS

Given that the T&E community will be relying in large part on existing IP-based telemetry sensors and communicating the data to existing IP-based telemetry applications, it is important to understand the implications of using the traditional Internet protocols (UDP, RTP, TCP, and IP) in the TmNS environment. There has also been substantial research in transport protocols specific to the satellite networks and routing protocols for MANETs, both of which share some characteristics with the airborne telemetry environment. When we originally presented AeroTP [6], the shortcomings of TCP and UDP were explained in detail. In this section we summarize those protocols, as well as examining the SCPS protocol in detail.

A. Transmission Control and User Datagram Protocols

TCP provides a connection-oriented reliable data-transfer service with congestion control, and uses constant end-to-end signaling to maintain consistent state at the source and destination. This introduces overhead, prevents utilizing all available bandwidth, and prevents operation in partitioned network scenarios. Each new TCP session requires a 3-way handshake before any real data is transmitted. This wastes 1.5 RTTs (round trip times) of valuable transmission time on short-lived connections such as those in the aeronautical telemetry environment, and prevents the sending of any data before a stable end-to-end path exists. Several other aspects of TCP, including its slow-start algorithm, assumption that all loss is due to congestion, and flow control mechanism, also prevent full utilization of the available bandwidth for the duration of a connection. The other commonly used Internet transport protocol is the User Datagram Protocol (UDP) [11] which is far simpler than TCP, but does not offer any assurance or notification of correct delivery. An extension to UDP is the Real-time Transport Protocol (RTP) [12], which adds timing information to support real-time media but does not add any reliability or delivery assurance.

B. SCPS-TP

The SCPS-TP (Space Communications Protocol Standards – Transport Protocol) [13] is a set of extensions and modifications to TCP to improve operation in the space environment, particularly for satellite communications. It both adds mechanisms to deal with specific environmentally-induced problems, and modifies existing mechanisms to reduce undesirable behaviors. The use of the SCPS-TP options is negotiated at the time of connection establishment, which allows the SCPS-TP agent to emulate TCP when communicating with a non-SCPS peer.

In SCPS-TP the default loss assumption is a user-selectable parameter on a per-path basis, so it will not assume congestion on links in which congestion is unlikely. It also allows for signaling of congestion, corruption, and link outage both from the destination host and intermediate routers to explicitly determine the source of packet loss. SCPS-TP implements the TCP Vegas [14] slow start algorithm and congestion control based on RTT estimates. Additionally SCPS-TP queries the user for the path bandwidth-$\times$-delay product and enters congestion avoidance once the congestion window size reaches this value, similar to the congestion avoidance algorithm described in [15]. This is beneficial for paths with high RTT, however given the rapidly changing topology of an airborne telemetry network, it is practically impossible

to maintain consistant RTT and bandwidth-$\times$-delay estimates. To attempt to do so whould require the use of extremely conservative estimates, resulting in low utilization of the already limited bandwidth. SCPS does Explicit Congestion Notification (ECN) using source quench SCMP (SCPS specific version of ICMP) messages as in [16]. It also uses an open-loop token bucket rate control mechanism [17] for each space link to avoid congestion, with the available capacity shared in the global routing structure. The dynamic nature of TmNS environment makes it difficult to maintain globally consistent routing information, and requires flow control to be handled locally. For loss due to corruption, SCPS-TP relies on the ground-station at the receiving end of each space link to maintain a moving average of the ratio of corrupted frames received and to use explicit cross-layer messages to inform the SCPS-TP destinations when that ratio exceeds a threshold. The destinations are then responsible for continuously notifying their respective sources of the corruption, during which the sources will not reduce the congestion window or back-off the retransmission timer in response to packet loss. In the case of a link outage, SCPS-TP assumes that the outage is bi-derectional, so the endpoints of the space link are responsible for notifying the SCPS-TP source and destination nodes on their side of the link. SCPS-TP then enters a persist state in which it periodically probes for link restoration at which point it can resume transmission where it left off without multiple timeouts or retransmissions or going through slow-start again.

To deal with the problem of highly-asymmetric channels, SCPS-TP reduces the number of ACKs required by TCP [18] from every other segment to only a few per RTT. This requires other TCP mechanisms such as fast retransmit [19] to be disabled. To deal with constrained bandwidth in general, SCPS-TP employs header compression and Selective Negative Acknowledgment (SNACK) [20, 21]. The header compression is end-to-end, as opposed to the TCP/IP header compression specified in [22] that is done hop-by-hop. This is because hop-by-hop header compression requires a costly resynchronization process and looses all segments in flight every time a packet is lost or arrives out of order. The end-to-end compression achieves about 50% reduction in header size by summarizing information that does not change during the course of the transport session. It also avoids the problems incurred by changing connectivity because the compression takes place at the endpoints which remain constant. The SNACK option allows a single NAK [23] to to identify multiple holes in the the received data out-of-sequence queue. SCPS-TP also uses TCP Timestamps [24] to keep track of RTTs even with lossy channel conditions, and uses the TCP window scaling option [24] so that the channel can be kept full even while recovering from losses. Many of these techniques for handling highly-asymmentric channels are applicable to the airborne telemetry network environment and we incorporate similar mechanisms into our solution, discussed later in this paper.

While SCPS-TP solves a number of the problems associated with airborne telemetry networks and our solution uses some of the same mechanisms, we have determined that SCPS-TP is not ideal for our application because it relies too heavily on channel condition information which is either pre-configured or learned gradually over multiple end-to-end connections. This process cannot adapt adequately to the rapidly changing T&E environment, or opportunistically make use of available bandwidth on a hop-by-hop basis.

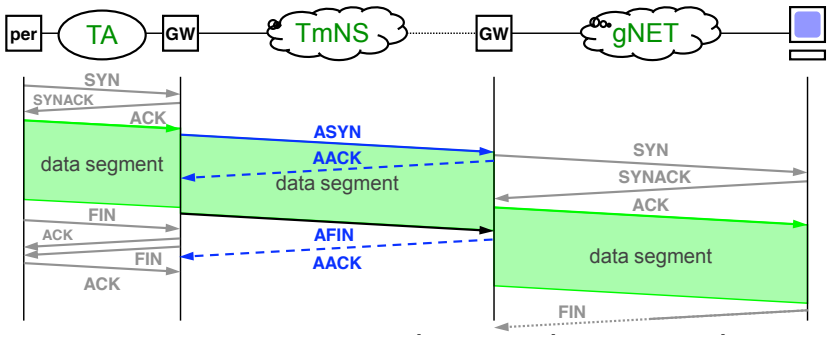

Figure 1: AeroTP connection setup

AeroTP OPERATION

AeroTP is a new domain-specific transport protocol designed to meet the needs of the telemetry network environment while being *TCP-friendly*[1] to allow efficient splicing with conventional TCP at the AeroGWs [4] in the GS and on the TA. AeroTP performs end-to-end data transfer between the edges of the telemetry network and splices to TCP connections or UDP flows at the AeroGWs. Transport-layer functions that must be performed by AeroTP include connection setup and management, transmission control, and error control. In this paper we will discuss the connection setup and error control.

C. *Connection Setup*

As an alternative to the overhead of the three-way handshake, AeroTP uses opportunistic connection establishment (Figure 1) in which data can begin to flow with the ASYN (AeroSYN) setup message. Error control is fully decoupled from rate control, and is service specific as described below.

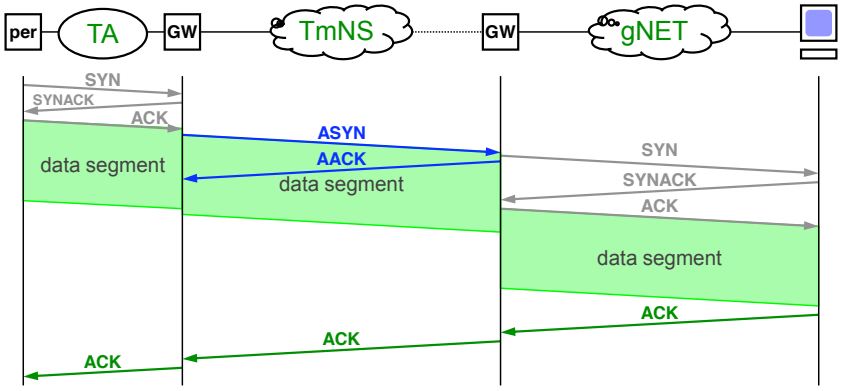

Figure 2: Reliable connection transfer mode

[1]Note that we use the term "TCP-friendly" in a more general sense than the established term "TCP-friendly rate control" (TFRC) [25]

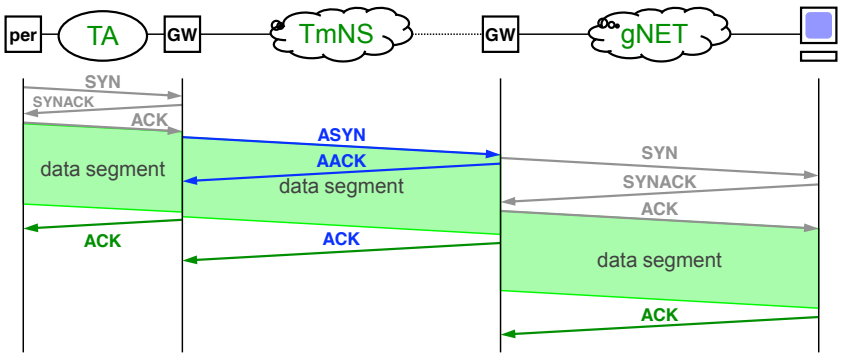

Figure 3: Near-reliable transfer mode

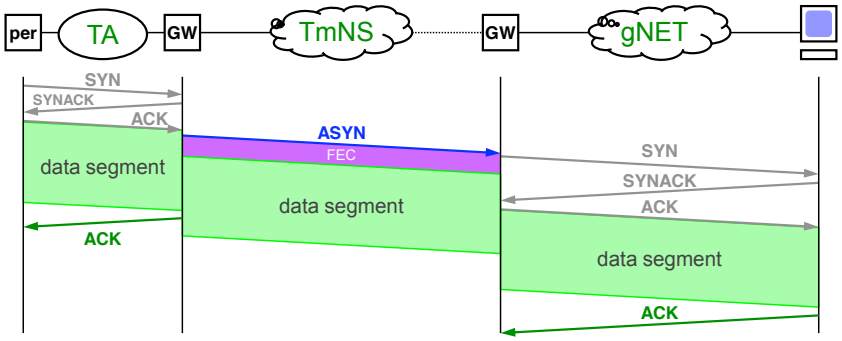

Figure 4: Quasi-reliable transfer mode

D. Error Control and QoS-Based Transfer Modes

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 (Figure 2) 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 (Figure 3) uses a custody transfer mechanism similar to that used in DTNs [26] 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) [13] 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.

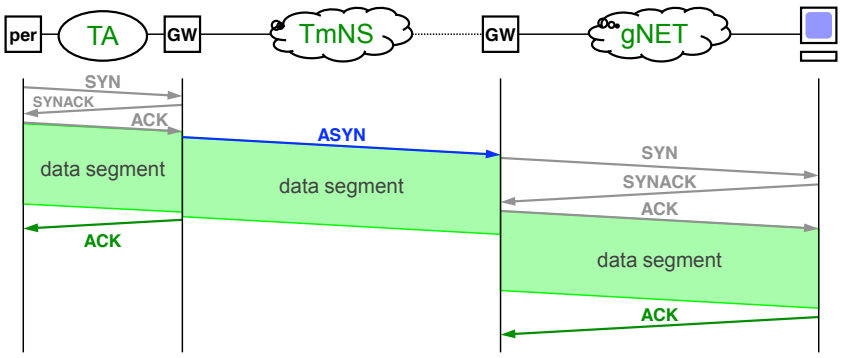

Figure 5: Unreliable connection transfer mode

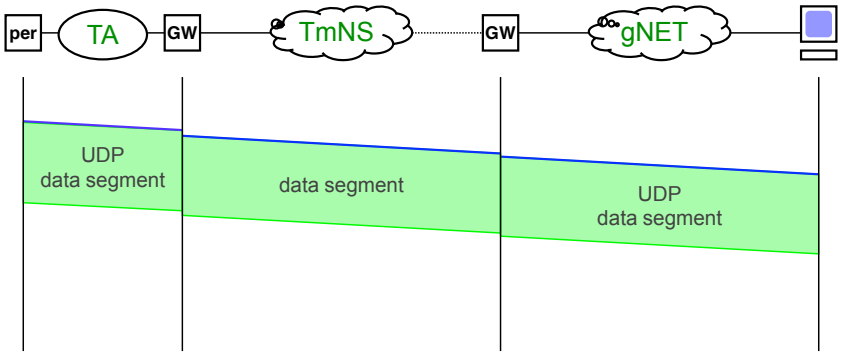

Figure 6: Unreliable datagram transfer mode

- ***Quasi-reliable connection*** mode (Figure 4) eliminates ACKs and ARQ entirely, using only open-loop error recovery mechanisms such as erasure coding, across multiple paths if available [27]. 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 (Figure 5) 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 (Figure 6) is intended to transparently pass UDP traffic, and no AeroTP connection state is established at all.

PRELIMINARY SIMULATION RESULTS

As mentioned previously, one of the drawbacks to TCP for this type of environment is the three-way-handshake used for connection establishment. For this reason AeroTP is designed to establish a connection when the first data packet in a flow is received. If the first packet is lost, the connection can

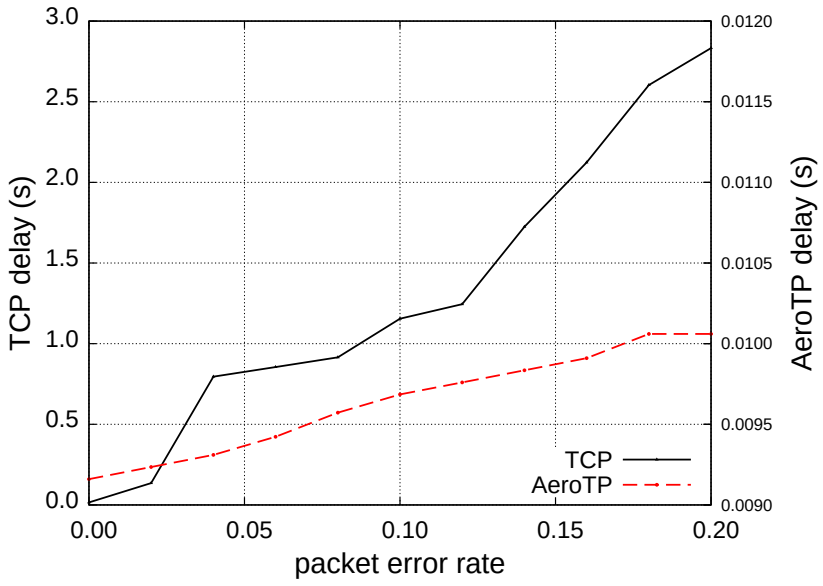

Figure 7: Reliable connection transfer mode

still be established using header information from the second or subsequent data packet, and the first packet can be retransmitted later if required by the specified reliability mode. To illustrate the difference between these two approaches, we have done a preliminary simulation, comparing the time required to establish a standard TCP connection, compared to a AeroTP connection.

The simulations are implemented in the ns-3 open-source simulator [28]. Each simulation consists of two nodes connected by a 10 MB/s link with 5 ms latency and a fixed probability of packet loss. Node 0 is configured as a traffic generator (TCP or AeroTP as appropriate) and node 1 is configured as a traffic sink. For each packet-loss probability point plotted, the simulations were run 100 times and the results averaged. Each simulation consists of a single connection attempt by either TCP or AeroTP. We record the delay starting when the `connection_setup` command is issued to the transport protocol, and stopping when the first *data* packet is received by the data sink.

Figure 7 shows the results of these simulations. To conserve space in this paper, both the TCP and AeroTP results are presented in a single plot, however, please note that they are plotted against two different y-axes: TCP on the left, and AeroTP on the right.. The TCP delay starts at about 20 ms when no losses occur, and increases linearly until it approaches 3 s when the packet-loss rate is 20%. The delay for AeroTP on the other hand has a delay of 9.2 ms when no losses occur, and increases linearly to 10.1 ms when the packet-loss rate is 20%. This shows an improvement of two orders-of-magnitude, which will play a large role in enabling AeroTP to successfully send data over paths which only exist for a few seconds, while TCP would still be trying to establish the connection.

CONCLUSIONS AND FUTURE WORK

The existing Internet protocol architecture is not well suited for telemetry applications in highly-dynamic airborne networks, which present unique challenges due to extreme mobility and limited band-

width. Typical transport protocols such as TCP are not designed for unstable paths, as are found in aeronautical telemetry environments. In this paper, we discuss AeroTP, a new transport protocol that addresses these issues with domain-specific solution developed to leverage cross-layer information in optimizing end-to-end performance. By sending data opportunistically, instead of waiting for a stable end-to-end path, AeroTP can make much more efficient use of available network capacity. We performed preliminary simulations that show the new protocol performs significantly better when establishing new connections in this environment. In the future we will perform more extensive AeroTP simulations, in conjunction with the AeroNP and AeroRP components of the protocol suite.

ACKNOWLEDGMENTS

This work was funded in part by the International Foundation for Telemetering (IFT). We would like to acknowledge the ResiliNets group members Abdul Jabbar and Egemen K. Çetinkaya 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] "iNET System Architecuture, version 2007.1." Central Test and Evaluation Investment Program (CTEIP), July 2007.

[2] "iNET Needs Discernment Report, version 1.0." Central Test and Evaluation Investment Program (CTEIP), May 2004.

[3] "iNET Technology Shortfalls Report, version 1.0." Central Test and Evaluation Investment Program (CTEIP), July 2004.

[4] E. K. Çetinkaya and J. P. G. Sterbenz, "Aeronautical gateways: Supporting TCP/IP-based devices and applications over modern telemetry networks," in *Proceedings of the International Telemetering Conference*, (Las Vegas, NV), October 26–29 2009. to appear.

[5] iNET Working Group, "http://www.inetprogram.org."

[6] 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.

[7] J. P. Rohrer, A. Jabbar, E. Perrins, and J. P. Sterbenz, "Cross-layer architectural framework for highly-mobile multihop airborne telemetry networks," in *Proc. IEEE MILCOM2008*, (San Diego, CA, USA), November 2008.

[8] A. Jabbar, E. Perrins, and J. P. G. Sterbenz, "A Cross-Layered Protocol Architecture for Highly-Dynamic Multihop Airborne Telemetry Networks," in *Proceedings of the International Telemetering Conference*, (San Diego, CA), October 27–30 2008.

[9] A. Jabbar and J. P. G. Sterbenz, "AeroRP: A geolocation assisted aeronautical routing protocol for highly dynamic telemetry environments," in *Proceedings of the International Telemetering Conference*, (Las Vegas, NV), October 26–29 2009. to appear.

[10] A. Jabbar, J. P. Rohrer, A. Oberthaler, E. K. Çetinkaya, V. S. Frost, and J. P. Sterbenz, "Performance comparison of weather disruption-tolerant cross-layer routing algorithms," in *Proc. IEEE INFOCOM 2009. The 28th Conference on Computer Communications*, April 2009.

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

[12] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications." RFC 3550 (Standard), July 2003.

[13] 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.

[14] P. Danzig, Z. Liu, and L. Yan, "An evaluation of TCP Vegas by live emulation," 1995.

[15] 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.

[16] S. Floyd, "TCP and explicit congestion notification," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 5, pp. 8–23, 1994.

[17] C. Partridge, *Gigabit Networking*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1994.

[18] R. Braden, "Requirements for Internet Hosts - Communication Layers." RFC 1122 (Standard), Oct. 1989. Updated by RFCs 1349, 4379.

[19] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms." RFC 2001 (Proposed Standard), Jan. 1997. Obsoleted by RFC 2581.

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

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

[22] V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links." RFC 1144 (Proposed Standard), Feb. 1990.

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

[24] V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High Performance." RFC 1323 (Proposed Standard), May 1992.

[25] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification." RFC 3448 (Proposed Standard), Jan. 2003.

[26] 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.

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

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