End-to-End Disruption-Tolerant Transport Protocol Issues and Design for Airborne Telemetry Networks

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

ABSTRACT

Networks of airborne nodes provide unique challenges to end-to-end communication, in particular due to the highly dynamic topology and time-varying connectivity of high velocity nodes, and unreliability of the wireless communication channel. This paper explores the issues and presents a design for a domain-specific transport protocol targeted to multihop network that interconnects high-velocity airborne nodes with the telemetry application of returning sensor data with high reliability.

INTRODUCTION AND MOTIVATION

Transport protocols provide end-to-end communication across the network to applications. The service they offer is to provide a unified interface for information transfer from one end system to the other. In an ideal world this service would be characterized by negligible delay, zero errors, and an unlimited bit rate. Unfortunately the characteristics of the underlying network and lower layers place limitations on the performance of the transport layer. The transport layer then has to adapt to the lower level limitations (delay, limited bandwidth, errors), while meeting the service requirement parameters of applications. In this work we are concerned with the transmission of telemetry data from airborne test articles to ground stations.

The iNET (Integrated Networked Enhanced Telemetry) program has identified a set of needs [1] for the T&E (test and evaluation) community that require a substantially enhanced networking capability for Major Range and Test Facility Bases. There is currently a significant effort underway in the iNET community to design the physical layer communications and MAC (medium access control) [2]. The current effort targets only the lower layers of the networking stack (PHY and MAC), however a number of issues remain to be solved at the network and transport layers [3]. This paper presents the design of *AeroTP*, a TCP-friendly transport protocol with multiple QoS modes for the TmNS (telemetry network system).

The current Internet protocols are unsuitable for the specific constraints and requirements of the aeronautical telemetry network environments in a number of respects [3]. At the same time, there is a need to be compatible with both TCP/IP-based devices located on test articles (TAs) as well as with the controlling applications at the ground station (GS). Therefore we are designing a new protocol suite that is both specific to the telemetry network environment, while fully interoperable with TCP/UDP/IP via gateways at the telemetry network edge to the GS and TA. It is important to note that while the telemetry network constrains some aspects of network operations, there are also aspects that can be *exploited* by domain specific protocols, such as the knowledge of TA location and trajectory by the GS.

A. *AeroTP: TCP-Friendly Transport Protocol for Aeronautical T&E*

TCP provides a connection-oriented reliable data-transfer service, with congestion control but no explicit support for precedence or QoS. Many of these mechanisms are unsuitable for wireless networks in general and telemetry networks in particular. TCP congestion control assumes that all losses are due to congestion, and therefore makes the wrong decision when bit errors corrupt packets [4]. Furthermore, TCP requires a reliable ACK stream for self-clocking that is unsuitable for highly dynamic and asymmetric networks. A number of these problems have been researched, and a few alternative protocols exist, such as SCPS-TP (space communications protocol standards – transport protocol) [5], from which we can draw some mechanisms.

We present a new domain-specific transport protocol *AeroTP*, which is designed for the aeronautical telemetry network environment while being *TCP-friendly*[1] to allow seamless splicing with conventional TCP at the telemetry network network edge in the GS and on the TA. Thus we transport TCP and UDP through the telemetry network, but in an efficient manner that meets the goals of this environment. AeroTP has several operational modes that support different service classes: reliable, nearly-reliable, quasi-reliable, best-effort connections, and best-effort datagrams. The first of these is fully TCP compatible, the last fully UDP compatible, and the others TCP-friendly with reliability semantics matching the needs of the mission and capabilities of the telemetry network. All but he last mode are connection oriented, but do not use a three-way handshake for connection establishment.

In designing this protocol, we are specifically concerned with DoD test ranges. The goal is to move from the current point-to-point unidirectional SST (serial-streaming telemetry) to a networked environment of bidirectional links to enable scalability, provide multihop TA–TA communication beyond TA–GS range, and to permit uplink control of TAs.

While physical layer solutions are necessary to maximize spectral efficiency, the network and transport layers provide a key piece of the solution. The ability to multihop provides spatial reuse since the TA–TA link range is shorter than TA–GS, providing greater aggregate throughput within the same spectrum. The QoS mechanisms of AeroNP [7] permit more important traffic classes (e.g. command and control) and mission-driven higher priority traffic to be delivered when a trade-off must be made. The cross-layering mechanisms allow the routing algorithm to influence transmission power of the TA–TA links to minimize interference. AeroTP supports multiple reliability modes to permit more efficient use of the resources based on the needs of traffic. Furthermore, in reliable and nearly-reliable mode when acknowledgments are needed, they are aggregated to reduce the chattiness of the protocol and conserve bandwidth. The packet formats are designed to reduce overhead as much as practical, for example by performing address translation so that IP addresses and unnecessary TCP and IP header fields are not transported through the telemetry network. The AeroTP header is designed to permit efficient translation between TCP/UDP and AeroTP at the gateway, as described in the Gateway Functionality Section below.

[1] Note that we use the term ”TCP-friendly” in a more general sense than the established term ”TCP-friendly rate control (TFRC) [6]

CHALLENGES TO END-TO-END COMMUNICATION IN AIRBORNE NETWORKS

A typical telemetry network for airborne test and evaluation is depicted in Figure 1. It presents several specific challenges that must be addressed at the transport layer for reliable collection of telemetry data. The flow of data is primarily from the TAs (test articles) to the GSs (ground stations), however command and control data will flow in the reverse direction.

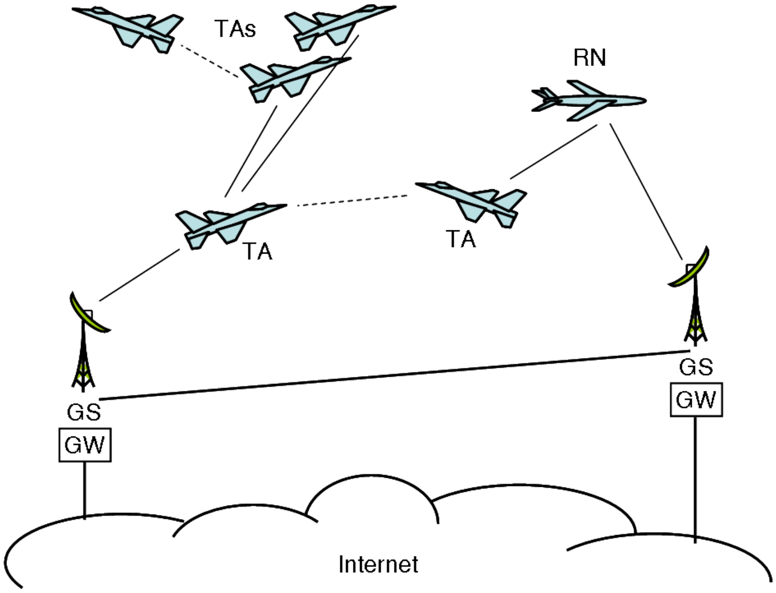

Figure 1: iNET test and evaluation environment

One challenge is that the RF spectrum available for use in these test ranges has been reduced over the years, while the quantity of data being collected has increased. The means that spatial reuse and efficiency are critical considerations. A second challenge is limited power, since telemetry modules are typically size and weight constrained. From a networking perspective this limits that range of radio transmissions and necessitates using multi hop routing, rather than broadcasting directly to ground stations in all scenarios. Thirdly and related to this is the problem of intermittent connectivity. Given that multihop transmission may be required to transmit data fro a TA to a GS, and that the test and evaluation environment may only have a few TAs in flight at any given time, network partitioning is a likely occurrence, so it is important to send data towards its destination, even if a complete path to the destination does not exist [8]. Fourthly, we have the problem of mobility; individual test articles may be traveling at speeds as high as Mach 3.5 (Mach 7 closing speed), meaning that TAs may be in contact with each other for only short periods of time. This results in a topology that is highly dynamic with frequent link connection and disconnection events. Lastly there is the problem of data corruption and loss. Wireless channels are inherently noisy, and even with reasonable levels of FEC applied at the link layer we can expect to see corruption-based loss at the transport layer due both to the channel characteristics as well as to the frequent topology changes previously described.

SHORTCOMINGS OF INTERNET TRANSPORT PROTOCOLS

The most widely used transport protocol in the Internet is the Transmission Control Protocol (TCP) [9, 10], which was designed for terrestrial wired networks. TCP provides a connection-oriented reliable data-transfer service with congestion control, and uses closed-loop feedback control to maintain consistent state at the source and destination. This introduces overhead and prevents utilizing all available bandwidth in networks which experience corruption-based loss and have large bandwidth-delay products. Operation in partitioned network scenarios is also prevented. Each new TCP session requires a 3-way handshake before data is transmitted. This uses 1.5 round-trip times (RTTs) even for short-lived flows such as those in the aeronautical telemetry environment, and also prevents the sending of any data before a stable end-to-end path exists. Even after the handshake is completed, TCP's slow-start algorithm prevents full utilization of the available bandwidth for many RTTs. TCP assumes that all loss is due to congestion, and it's congestion control algorithm operates by halving the transmission rate every time there is a packet loss. This is the wrong approach for wireless networks in which noisy channel conditions are expected to be the dominant cause of packet loss [4]. TCP's flow control requires a reliable ACK stream, which limits its ability to handle highly-asymmetric links even when the data is only flowing in the high-bandwidth direction. The practical limit to asymmetry for TCP flows is about 75:1 [11]. There is also substantial overhead with the 20 byte TCP header per packet, especially when using small segments for ACKs or to decrease the probability of having a bit-error. TCP was not designed with intermittent connectivity in mind; short-term link outages invoke congestion control and repeated retransmission back-offs, which results in an inability to detect link restoration. A longer link outage results in TCP dropping the connection. Varying RTT can also pose a problem for TCP, because it will incorrectly assume a packet loss and retransmit unnecessarily as well as reducing the congestion window. Many TCP mechanisms are unsuitable for wireless networks in general and the telemetry test and evaluation environment in particular.

The other commonly used Internet transport protocol is the User Datagram Protocol (UDP) [12]. UDP is far simpler than TCP, but does not offer any assurance or notification of correct delivery. It does not do connection setup, congestion control, flow control, or data retransmission. Because of this UDP does not need to maintain consistent state at both ends of the connection. An extension to UDP is the Real-time Transport Protocol (RTP) [13, 14] which adds time synchronization for real-time applications but does not add any reliability or delivery assurance.

In the test and evaluation environment we expect to have multiple classes of traffic that have different characteristics, loss tolerance, and priorities. Neither TCP or UDP have the capability to provide differentiated levels of precedence or QoS to meet these requirements. In an IP network IntServ or DiffServ could be used to achieve this however these to not address the need for different end-to-end semantics and reliability requirements. A number of these shortcomings have been researched, and a few alternative protocols exist, such as SCPS-TP (space communications protocol standards – transport protocol) [5], from which we can draw some mechanisms but are only a partial solution.

MECHANISMS TO MITIGATE EFFECTS OF CHALLENGES

In light of the challenges described previously, there are a number of specific mechanisms we can use to improve performance at the transport layer, with respect to the performance of the protocols currently employed in the Internet. Some of these mechanisms are aimed at improving TCP performance over satellite networks [11], but can also be applied to the airborne telemetry environment. One specific goal is bandwidth efficiency, because of the limited spectrum available. To mitigate this we use a handshake-free connection setup [15], and transmit at peak rate immediately instead of beginning at a low rate and gradually increasing as in TCP slow start [16]. We do not necessarily need to wait for acknowledgments before continuing to transmit, not only to improve performance but because we may not have stable E2E connectivity during partitioned network operation. Because of the concern about bandwidth efficiency we want to use as few ACKs as possible, and some classes of traffic may not require ACKs at all. Header compression [17] is also a useful mechanism for reducing overhead, especially for small packets such as ACKs. In reliable modes we want to be careful to retransmit only what is needed; the SNACK (selective negative ack) [5] mechanism allows this by identifying specific missing segments for retransmission. Because our environment may not support a stable E2E path we assume that intermediate relay nodes have enough storage capacity to buffer data until a link to the destination becomes available. A side effect of this is that buffering with priority queuing can be used if congestion is encountered, allowing the network layer (AeroNP) to handle congestion instead of the transport layer.

Because path persistence (the time over which a given sequence of links remain connected) may be very short, interactively probing for path characteristics can waste valuable transmission time. To mitigate the impact of this we can remember connection state information for subsequent similar connections, which will reduce the amount of state that the transport layer needs to re-establish for each new connection [18, 19, 20].

Due to the challenge of noisy wireless channels and frequent topology changes, relying on retransmission alone for error correction at the transport layer may not be sufficient for all test scenarios. To mitigate this challenge we do erasure coding across one or more paths as they are made available by the network layer.

AeroTP: TCP-FRIENDLY TRANSPORT PROTOCOL FOR AERONAUTICAL T&E

To meet the needs of the telemetry network environment, we are developing a new domain-specific transport protocol *AeroTP*, which is designed for the aeronautical telemetry network environment while being *TCP-friendly* to allow seamless splicing with conventional TCP at the telemetry network edge in the GS and on the TA. Thus we transport TCP and UDP through the telemetry network, but in an efficient manner that meets the needs of this environment: dynamic resource sharing, QoS support for fairness and precedence, real-time data service, and bidirectional communication. AeroTP has several operational modes that support different service classes: reliable, nearly-reliable, quasi-reliable, best-effort connections, and best-effort datagrams. The first of these is fully TCP compatible, the last fully UDP compatible, and the others TCP-friendly with reliability semantics matching the needs of the mission and capabilities of the telemetry network. The AeroTP header is designed to permit efficient translation between TCP/UDP and AeroTP at the gateway.

AeroTP performs end-to-end data transfer between the edges of the telemetry network and splices to TCP connections or UDP flows at the gateways. Transport-layer functions that must be performed by AeroTP include connection setup and management, transmission control, and error control.

B. Connection Management and Rate-Based Transmission Control

AeroTP uses connection management paradigms suited to the telemetry network environment. An alternative to the overhead of the three-way handshake is an opportunistic connection establishment in which data can begin to flow with the setup message (SYN). Closed-loop window-based flow and congestion control with slow start is not appropriate to the highly dynamic wireless environment of airborne telemetry networks. Therefore we use an open-loop rate-based transmission control with instrumentation from the network layer and test plan to determine an initial rate, with backpressure to control congestion, as described in [7] for AeroNP. Error control is fully decoupled from rate control, and is service specific as described below.

C. Segment Structure and Gateway Functionality

AeroTP is *TCP-friendly*, meaning it is designed to efficiently interoperate with TCP and UDP at the telemetry network edge. To support this, gateway functionality [21, 22] does IP–AeroNP translation [7] and TCP/UDP–AeroTP splicing. A preliminary design of the AeroTP segment is shown in Table 1. Since bandwidth efficiency is critical, AeroTP does not encapsulate the entire TCP/UDP headers, but rather the gateway converts between TCP/UDP and AeroTP headers. Some fields that are not needed for AeroTP operation but are needed for proper end-to-end socket semantics are passed through, such as the *source* and *destination port* number, *ECN*, *TCP flags*, and the *timestamp*.

Table 1: AeroTP Segment Structure

source port			destination port
sequence number			
timestamp			
mode	resv.	ECN+flags	HEC
payload			
CRC-32			

The *sequence number* allows reordering of packets due to erasure coding over multiple paths or TA mobility, and is either the TCP byte sequence number or a segment number, depending on the transfer mode described below. The *HEC* (header check) field is a strong CRC (cyclic redundancy check) on the integrity of the header in the case of bit errors in the wireless channel. This allows us to correct a corrupted payload end-to-end using FEC, as long as the header is not corrupted so it can be correctly delivered to the destination application. A *CRC* checks the integrity of the data edge-to-edge across the telemetry network since there is not a separate AeroNP or link layer frame CRC, and allows measurement of the bit-error-rate for erasure code adaptation depending on the transfer mode. These last two goals mean that the AeroNP does not necessarily drop packets if they experience corruption, which is a key difference from IP forwarding policy [23]. The *mode* field indicates which of the five transfer modes is in use.

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 from different traffic classes:

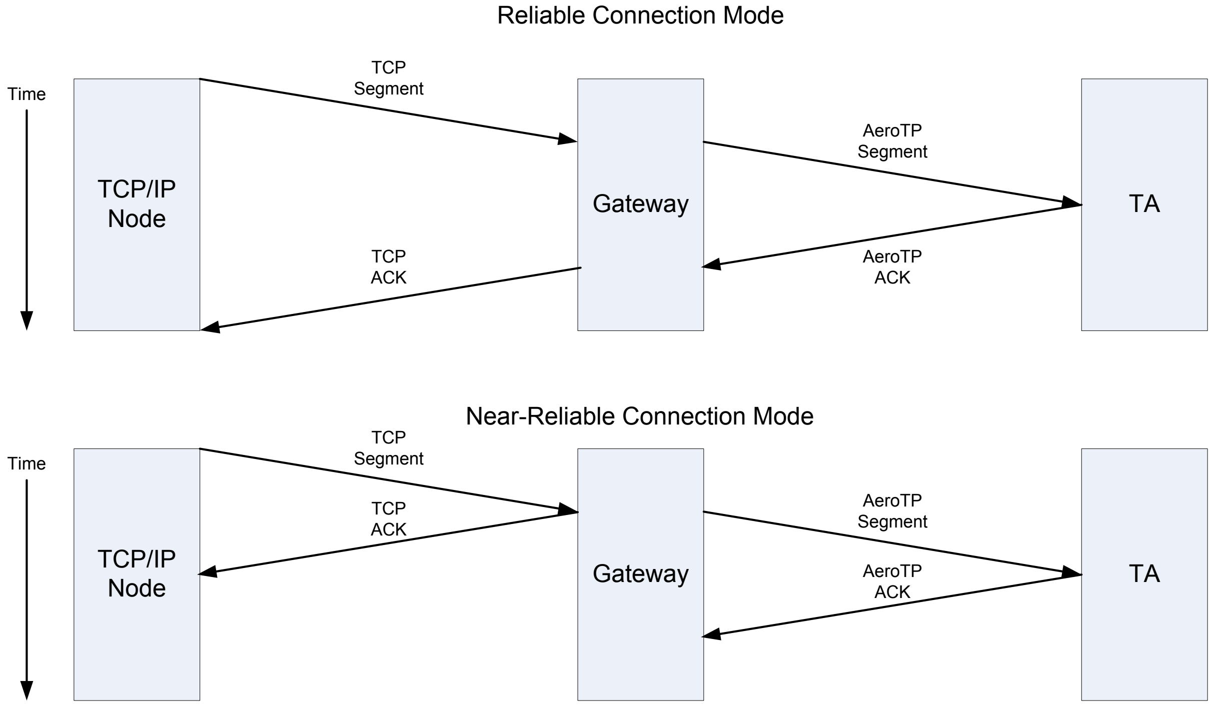

Figure 2: Reliable and near-reliable transfer modes

All modes except unreliable datagram are connection-oriented for TCP-friendliness and will use byte sequence numbers for easy translation to TCP at the gateway, so that packets may follow varying or multiple paths and be reordered at the receiver.

- ***Reliable connection*** mode must preserve end-to-end acknowledgement semantics from source to destination as the only way to *guarantee* delivery. Two possible mechanisms are ACK passthrough, which has the disadvantage of imposing TCP window and ACK timing onto the AeroTP realm, or custody transfer [24] (Figure 2a) that splits the TCP ACK loop at the gateway, at the cost of buffering AeroTP segments in the gateway until fully acknowledged.

- ***Near-reliable connection*** mode is highly reliable, but does not *guarantee* delivery since the gateway uses split ARQ and immediately returns TCP ACKs to the source (Figure 2b) on the assumption that AeroTPs reliable ARQ-based delivery will succeed using SNACKs (selective negative acknowledgements) [5] 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 and erasure coding across multiple paths if available [25]. 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 (if available) to preserve data integrity and does not use any error correction mechanism at the transport layer. While we may eventually use cross-layering to vary the strength link FEC, we cannot assume this capability.
- ***Unreliable datagram*** mode is intended to transparently pass UDP traffic, and no AeroTP connection state is established at all.

E. Cross-Layer Optimization Between AeroTP/AeroNP and the iNET MAC/PHY

Cross-layering, in the form of clearly defined knobs and dials, plays a critical role in the operation of AeroTP. At the application–transport interface this allows the transport layer to indicate the service level of the available path to the application through the *service level* dial, and allows the application to indicate the priority and characteristics of the data being transferred through the *reliability mode* knob. At the transport–network interface the network layer is able to give the transport layer the *path characteristics* such as available bandwidth and multipath availability, and based on that information the transport layer can set the *forwarding mode* knob appropriately. Table 2 shows these knobs and dials along with others influencing the operations of the lower layers of the network stack that are discussed further in the companion paper [7].

Table 2: Knobs and Dials

Layer	Knobs	Dials	Layer influencing knob
transport	reliability mode	service requirements	application
network	forwarding mode	path characteristics	transport
link & MAC	ARQ & FEC	link characteristics	network
physical	coding	channel conditions, available coding schemes	link

We expect that there will be significant benefits by employing cross-layer optimizations not only among AeroTP and AeroNP, but also with the iNET MAC and PHYs. Therefore, we are investigating the tradeoffs in type and strength of FEC at the PHY layer with respect to channel conditions and BER (bit error rate), as well as optimizing TDM (time division multiplexing) parameters and slot assignment based on the transfer mode of AeroTP and QoS parameters (precedence and service type) of AeroNP. Furthermore, the quasi-reliable mode of Aero-TP erasure codes across multiple TA–GS paths when available, requiring coordination of GS and iNET MAC slot assignment with AeroNP routing and AeroTP transport. Finally, the support for multicast and broadcast requires coordination of AeroNP routing with the broadcast capabilities of the iNET MAC.

CONCLUSIONS AND FUTURE WORK

Airborne telemetry networks present a number of unique challenges to traditional network end-to-end protocols. AeroTP provides a domain-specific transport protocol which is compatible with existing Internet protocols through the use of a gateway.

We emphasize that the design ideas presented for AeroTP and AeroNP are *preliminary*; future research is needed to simulate, prototype, and finalize their design. The next phase of research for AeroTP involves conducting simulations using the ns-2 simulation environment. The simulations will allow the comparison of mechanisms and evaluation of performance with various wireless topologies. Following the simulation phase, an implementation on physical ground-based mobile platforms is needed to verify the accuracy of the simulations and performance in a real-world environment.

REFERENCES

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

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

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

[4] 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," *Comput. Netw.*, vol. 46, no. 3, pp. 343–362, 2004.

[5] 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.

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

[7] 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 (to appear)*, (San Diego, CA), October 27–30 2008.

[8] J. P. G. Sterbenz, R. Krishnan, R. R. Hain, A. W. Jackson, D. Levin, R. Ramanathan, and J. Zao, "Survivable mobile wireless networks: issues, challenges, and research directions," in *WiSE '02: Proceedings of the 3rd ACM workshop on Wireless security*, (New York, NY, USA), pp. 31–40, ACM Press, 2002.

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

[10] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP." RFC 3168 (Proposed Standard), Sept. 2001.

[11] M. Allman, S. Dawkins, D. Glover, J. Griner, D. Tran, T. Henderson, J. Heidemann, J. Touch, H. Kruse, S. Ostermann, K. Scott, and J. Semke, "Ongoing TCP Research Related to Satellites." RFC 2760 (Informational), Feb. 2000.

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

[13] A.-V. T. W. Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications." RFC 1889 (Proposed Standard), Jan. 1996. Obsoleted by RFC 3550.

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

[15] D. Feldmeier, *An Overview of the TP++ Transport Protocol Project*, ch. 8. Kluwer Academic Publishers, 1993.

[16] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control.” RFC 2581 (Proposed Standard), Apr. 1999. Updated by RFC 3390.

[17] G. Pelletier, K. Sandlund, L.-E. Jonsson, and M. West, “RObust Header Compression (ROHC): A Profile for TCP/IP (ROHC-TCP).” RFC 4996 (Proposed Standard), July 2007.

[18] R. Braden, “T/TCP – TCP Extensions for Transactions Functional Specification.” RFC 1644 (Experimental), July 1994.

[19] R. Braden, “Extending TCP for Transactions – Concepts.” RFC 1379 (Informational), Nov. 1992. Updated by RFC 1644.

[20] J. Touch, “TCP Control Block Interdependence.” RFC 2140 (Informational), Apr. 1997.

[21] “iNET TmNS Ground Segment Architecture, version 2007.” Central Test and Evaluation Investment Program (CTEIP), July 2007.

[22] “iNET TmNS Test Article Segment Architecture, version 2007.” Central Test and Evaluation Investment Program (CTEIP), July 2007.

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

[24] K. Scott and S. Burleigh, “Bundle Protocol Specification.” RFC 5050 (Experimental), Nov. 2007.

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