Aeronautical Gateways: Supporting TCP/IP-based Devices and Applications over Modern Telemetry Networks

Egemen K. Çetinkaya and James P.G. Sterbenz
Department of Electrical Engineering and Computer Science
Information and Telecommunication Technology Center
The University of Kansas
Lawrence, KS 66045
{ekc|jpgs}@ittc.ku.edu

ABSTRACT

Modern telemetry networks require the use of efficient domain-specific protocols at the transport, network, and routing layers. However, the existing end-devices and services are based on legacy protocols such as TCP/IP. This necessitates translation between the legacy and aeronautical protocol. In this paper we propose an efficient translation mechanism with the help of gateways at the telemetry network edges.

INTRODUCTION and MOTIVATION

Telemetry applications in the mobile airborne networks pose unique challenges. In the current underlying network infrastructure these applications rely on point-to-point links based on IRIG 106 standard for serial streaming telemetry (SST) [1]. Source and destination nodes should be in range of each other using the point-to-point links. In a scenario where a direct communication link is not possible between source and destination, a multi-hop network can be the only feasible solution for providing an end-to-end communication path. The need to support real-time telemetry applications in the limited radio spectrum, combined with economical benefits using a networked telemetry system is recognized by the DoD [1]. However current network protocols (e.g. TCP/IP) cannot support the proposed networked telemetry systems, because these protocols make assumptions for the environment they operate that does not hold in the highly mobile aeronautical environment. This necessitates development of new protocols for the iNET program [1] proposed by the DoD.

A preliminary communication network system for the iNET program is presented in [2, 3]. Network elements in the iNET environment is shown in Figure 1. The proposed networked system for telemetry applications include three types of network nodes: test articles (TA), ground stations (GS), and relay nodes (RN). TAs house the telemetry equipment and they are the airborne vehicles. The temeletry data can be sent to GSs directly for processing. If there is no direct communication link between the TA and GS, RNs, which are airborne vehicles, can relay the data to the destination. In such a highly mobile environment where relative speeds can reach up to 7 Mach [4], current TCP/IP networking protocols fail to provide communication paths.

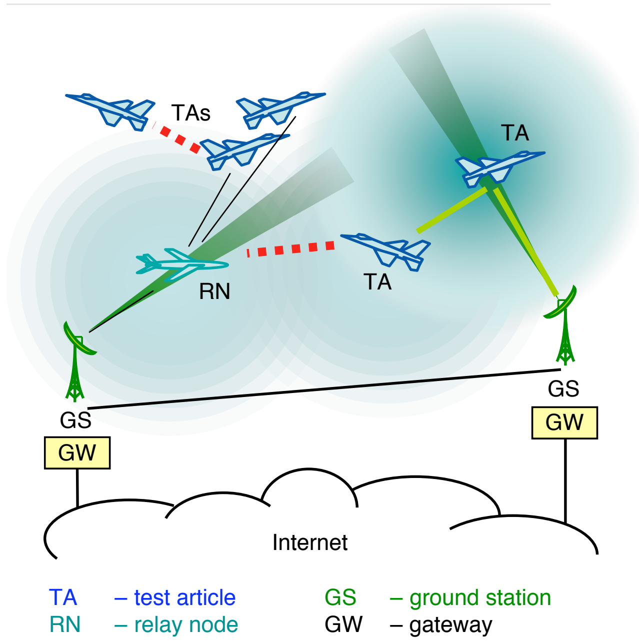


Figure 1: iNET Environment

TCP/IP based protocols assume that the underlying links are stable and reliable. This is not the case for mobile environments. The challenges and constraints are well documented in [2]. In addition to the challenges present for this aeronautical environment, the proposed network protocols should be compatible with the current TCP/IP based protocols in order to support off-the-shelf software and hardware. Therefore the proposed protocols [2, 4] must be interoperable with the TCP/IP based protocols at the proposed telemetry networked system (TmNS) at the network edges.

The rest of the paper is organized as follows: the next section presents current Internet architecture and its inability to meet the demands of telemetry networks. Challenges to reliable network communication, specifically in the iNET scenario is presented. Then, we present the functionality of the *AeroGW*, which provides an interface between the different network environments. Finally, conclusions and future work is presented.

EXISTING COMMUNICATION NETWORK PROTOCOLS

The proposed TmNS architecture will primarily reuse existing IP-based telemetry peripherals. Thus, it is important to understand implications of the existing networking protocols in the TmNS environment. This section presents a brief overview of the existing protocols and examines their suitability for a highly mobile airborne evironment.

The Internet uses IP at the network layer, with various routing protocols such as OSPF, RIP, and BGP.

TCP is the widely used transport layer protocol that operates on top of the IP in the protocol stack. The current Internet architecture is based on the fundamental assumption of long lasting, stable links that does not hold true for a Mach-speed airborne network, which not only challenges TCP, but also network routing. TAs can be traveling at relative speeds up to Mach 7 which will make the network topology constantly changing. Internet protocols do not support dynamic topologies implicitly, requiring convergence of the routes, which is not suitable for airborne telemetry environment. TCP provides connection oriented end-to-end reliable service. Constant mobility of the TAs may cause the network to be partitioned, which will eliminate the use of end-to-end TCP protocol. TCP uses signalling at the end systems to handle the congestion and the flow control. Congestion control mechanisms prevent the full utilization of the bandwidth. The existing TCP/IP-based communication networking protocols will fail to operate in a highly mobile airborne environment.

It is expected that the source and destination in telemetry network will be operating on TCP/IP protocols. On the other hand, existing TCP/IP protocols are not feasible to operate in the TmNS environment. Thus, an interface that will enable the interoperability of domain specific protocols with the current TCP/IP protocol suite is needed. Gateways are designed to handle such tasks, and we will describe the AeroGW functionality in the next section.

AeroGW: INTERFACE BETWEEN AeroN|TP and TCP/IP PROTOCOL SUITE

This section describes the AeroGW functionality. The source and destination of telemetry data are expected to be TCP/IP-based systems; however TmNS will employ new domain-specific protocols suitable for the dynamic airborne environment. To overcome this challenge without requiring a total redesign of all telemetry sensors, peripherals, applications, and workstations, we introduce the Aero Gateway (AeroGW). The protocol architecture is shown in Figure 2, showing the gateway functionality at the TA and GS.

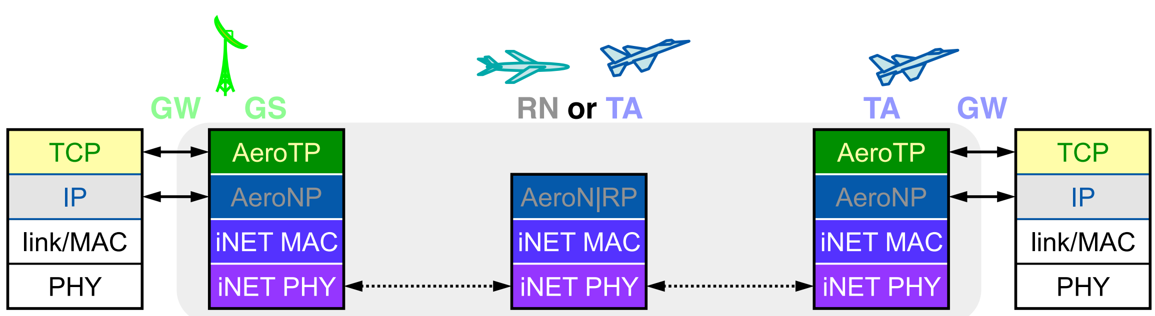
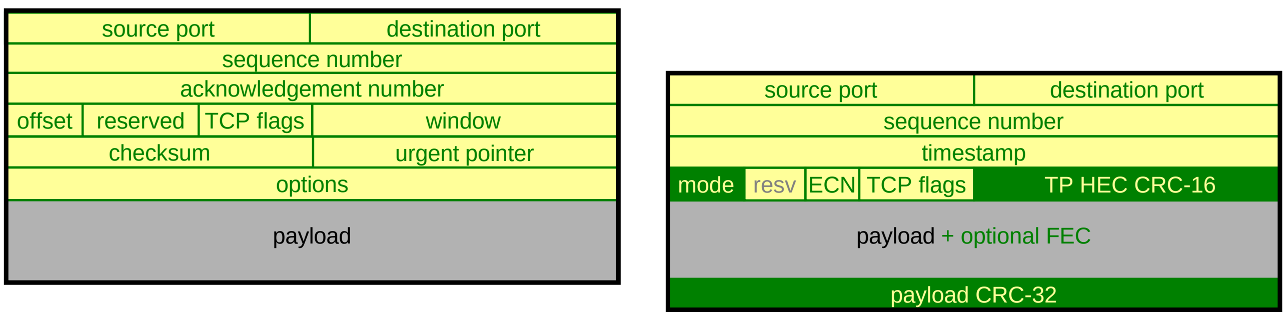


Figure 2: System Architecture

Gateways can be thought of as an interface that provides translation between two entities, can support different access technologies or different protocols, and generally have sufficient processing resources. The gateway concept is well established [5] as a mechanism for bridging between disparate network environments. In this case its operation is similar to TCP-Splice [6], however instead of splicing TCP with TCP, it will splice TCP (and UDP/RTP) to AeroTP and IP to AeroNP. This translation functionality resides in the AeroGW, which is incorporated into the ground network (gNET) interface element on the ground station and the vehicle network (vNET) element on each TA.

A. *TCP – AeroTP Splicing*

AeroTP is the transport layer proposed [3, 7] to overcome the challenges of the highly dynamic wireless, mobile aeronautical environment. It provides opportunistic connection setup and management, transmission control, and flexible error control functions. AeroTP supports different modes of operation: reliable, nearly-reliable, quasi-reliable, best-effort connections, and best-effort datagrams. Each of the operational modes is designed for the needs of specific missions. AeroTP is TCP friendly in that it allows efficient splicing with TCP at the gateways. TCP is well documented in numeruous RFCs [8, 9, 10, 11]. Excluding the 4 byte long optional fields, TCP adds a 20 byte header per packet as shown in Figure 3(a). This is a substantial overhead in the case of control traffic such as ACKs. For a side by side comparison, the proposed AeroTP header format is shown in Figure 3(b). It should be noted that the light colored fields are from the TCP header, while the dark colored fields are new in the AeroTP header and trailer.



(a) TCP Segment

(b) Aero TPDU

Figure 3: Transport Layer Protocol Data Units

The AeroTP header consists of a 16 byte header. Source and destination port fields identify the sending and receiving ports. 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 an AeroTP TPDU number, depending on the transfer mode. The timestamp field is 32 bits wide for end-to-end TCP transparency. There are five modes of operation to match the reliability needs of a telemetry application. The Header Error Check (HEC) field is a strong Cyclic Redundancy Check (CRC) on the integrity of the header to detect bit errors in the wireless channel. This allows the packet to be correctly delivered to AeroTP at the destination when a corrupted payload can be corrected on an end-to-end basis using FEC. A CRC trailer protects the integrity of the data edge-to-edge across the telemetry network in the absence of a separate AeroNP or link layer frame CRC and enables measurement of the bit-error-rate for erasure code adaptation depending on the transfer mode.

TCP requires a three-way handshake process to establish a connection before data is transmitted. This connection setup consumes 1.5 round trip times (RTTs) and prevents the sending of any data before a stable end-to-end path exists. 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 AeroSYN (ASYN) setup message, which will not waste the 1.5 RTT. The opportunistic connection signalling paradigm for networked telemetry system is shown in Figure 4.

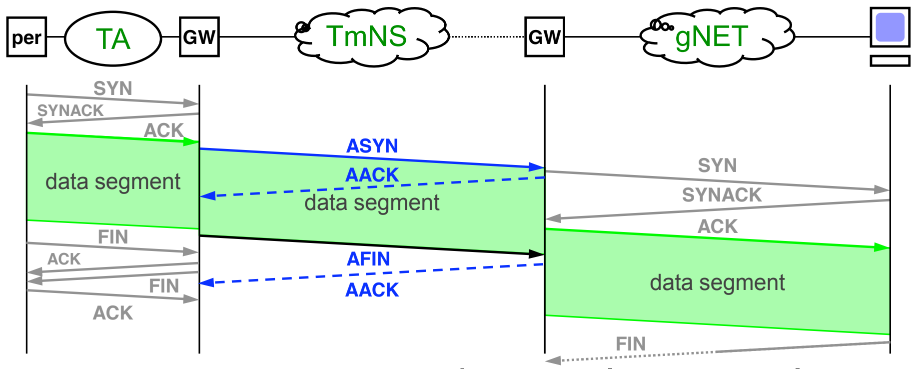


Figure 4: Transport Layer Opportunistic Connection Signalling

A packet must pass through two gateways on its path from source to destination. The ingress gateway will convert the TCP messages to AeroTP messages, while egress gateway will convert AeroTP messages to TCP format. It should be noted that ingress and egress gateways are not additional network elements in the TmNS environment, but rather the gateway functionality will be built in TAs and GSs. The flow diagram for the TCP to AeroTP translation that occurs at the ingress gateway is presented in Figure 5. Essentially, ingress gateway will splice the end-to-end TCP protocol. Once the TCP SYN message is received, gateway will send back a SYN ACK message. Upon receiving the SYN ACK message the source will send the TCP ACK message. The gateway will transmit the ASYN message along with the data to the TmNS after receiving the TCP ACK message. The data can piggyback the ASYN message. The ingress gateway will check the succesfull transmission of the data to the egress gateway via incoming AeroACK (AACK) messages. If the ASYN message is delivered to the egress gateway, data can continue to flow from source to destination. In the case of a failed delivery of the ASYN message, it should be sent again to preserve the end-to-end TCP semantics. Once the destination receives the application data, it will send a TCP FIN message to the gateway signalling termination of the connection. The egress gateway will send the corresponding AeroFIN (AFIN) message to the ingress gateway to end the connection.

The flowchart for the AeroTP to TCP translation that occurs at the egress gateway is shown in Figure 6. The egress gateway complements the splicing function by reconstructing the TCP messages. Upon receiving the ASYN message, the egress gateway will send the TCP SYN message to the destination. Delivery of the TCP SYN message is checked with the SYN ACK message. If SYN ACK is not received, the egress gateway will retransmit the TCP SYN message. Upon receiving the SYN ACK, the egress gateway can start transmitting the data, which includes the ASYN and application or control data it received from the ingress gateway. Once the TCP FIN message is received from the destination, the egress gateway will transmit the AFIN message to the TmNS for connection termination.

B. IP – AeroNP Translation

The network layer provides services to the transport layer. To adapt the needs of AeroTP, a new network protocol, AeroNP [2], is proposed for the telemetry environment. AeroNP is *IP-compatible*, meaning

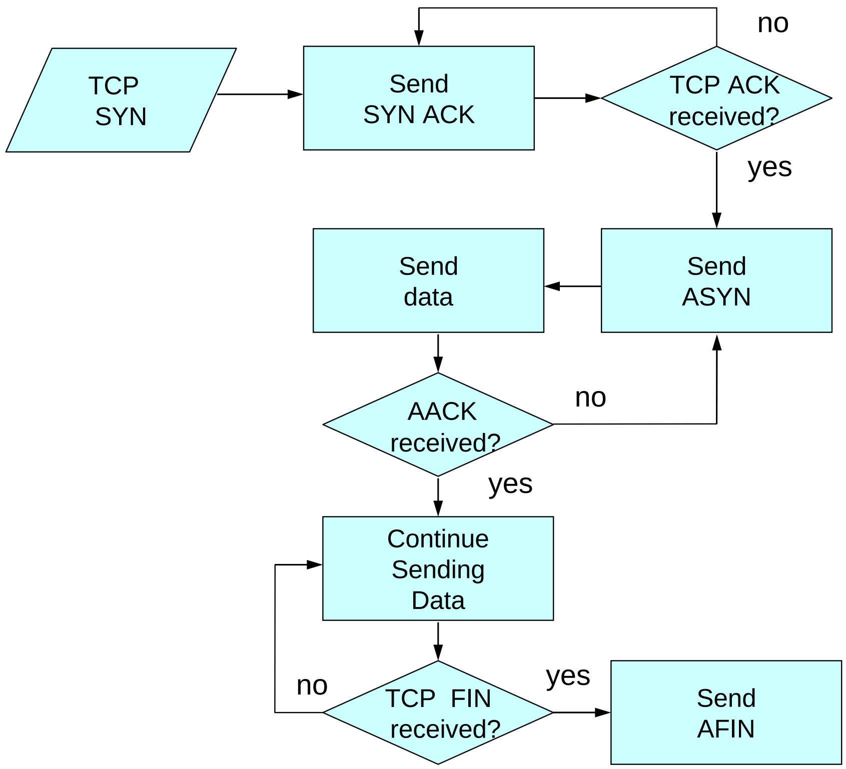


Figure 5: TCP to AeroTP Conversion

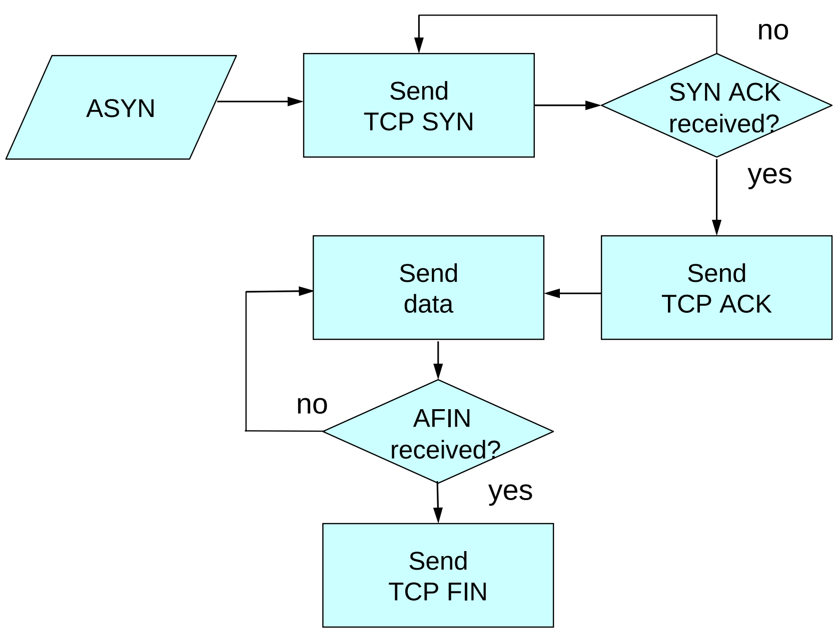
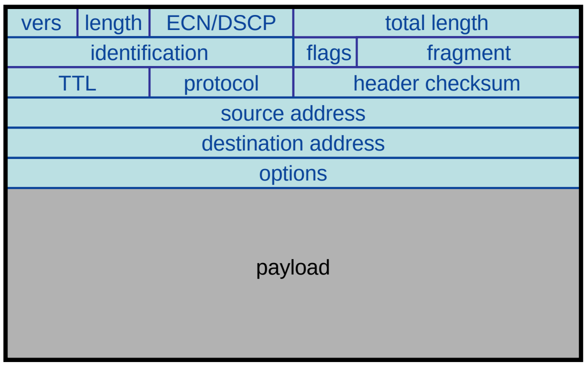
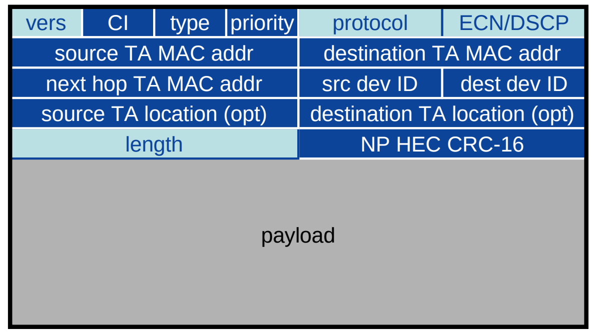


Figure 6: AeroTP to TCP Conversion

that, given the IP-based end devices such as HMI applications on the grund network and the TCP/IP peripherals on the TA, it is critical for the network protocol on the telemetry subsystem to be interoperable with IP. IP is the de facto network layer protocol for the Internet and its operation is standardised and well documented in RFCs [12, 13]. Excluding the optional fields, the IP header is 20 bytes long and its format is shown in Figure 7(a). The proposed AeroNP header is 32 bits wide, shown in Figure 7(b).



(a) IP Packet



(b) Aero NPDU

Figure 7: Network Layer Protocol Data Units

In the Aero NPDU the light colored fields come from the IP header and dark colored fields are specifically designed for the AeroNP header. The congestion indicator field is set by each node to notify the neighboring nodes of its congestion level. This field will provide support of reliable communication paths in multi-hop scenarios in which the node that is congested should not be considered for path setup by the AeroRP routing algorithm. The optional source TA location and destination TA location fields designed to be used by the AeroRP. These fields contain the GPS coordinates that are used in location-aware routing [14].

Given that wireless links are bandwidth constrained and different applications have their own requirements to operate, it is essential to have a quality of service mechanism. The type and priority fields specify the QoS level of a given packet. The ECN/DSCP field is also used for QoS mechanisms, but this field is carried for end-to-end IP transparency. Version, protocol, and length are the other AeroNP fields that will passed in the gateways without any modification for IP-compatibility.

AeroNP does not carry the 32-bit source and destination IP address fields. Instead, it utilizes the iNET MAC address of existing hardware. Since iNET MAC is based on TDM, an AeroNP packet is inserted directly into a TDM slot. Some header space efficiency will be gained by eliminating the use of network address fields in the header. However to be compatible with the existing IP based services, it is clear that an IP address to AeroNP MAC address translation is necessary at the gateways. This address conversion mechanism can be similar to Address Resolution Protocol (ARP) function. In ARP, a table is constructed such that there is a mapping between an IP and MAC address. The next hop TA MAC address field will be used by the AeroRP for identifying the route to the next hop.

TAs can have multiple peripherals for different purposes. Each of the peripherals will use a seperate IP address. Since, AeroNP header does not have IP address fields, each peripheral can be mapped to a device id. This translation functionality is similar to Network Address Translation (NAT) and will reside at the gateways. While IP address to device IDs can be mapped dynamically, it will be more efficient to preload the mapping table at the beginning of each mission. Finally, HEC field is used to check the integrity of the AeroNP header fields against bit errors.

AeroNP is IP compatible to support end-to-end transparency to TCP/IP based applications. The primary translation required is for IP address to MAC address, and IP address to device ID mapping. Both of the translation and mapping tables can be a preprogrammed for efficiency and simplicity. Unlike AeroTP–TCP translation, AeroNP–IP translation is forseen to be simpler.

CONCLUSIONS and FUTURE WORK

The existing Internet protocols are not well suited for telemetry applications in highly-dynamic airborne networks, due to extreme mobility and limited bandwidth nature of this environment. This necessitates the design of domain specific protocols. However, use of TCP/IP based telemetry applications are expected. To bridge between TCP/IP based and domain specific protocols, a gateway functionality is required. In this paper, we discuss the gateway functionality that addresses proper translation between two protocol formats. While providing the necessary translation mechanism, gateway functionality will add some complexity to the system. We are also working on ns-3 simulations of AeroNP and AeroTP. The gateways will be built upon those protocols and will be simulated afterwards.

Acknowledgements

This work was funded in part by the International Foundation for Telemetering (IFT). We would like to acknowledge Justin P. Rohrer and Abdul Jabbar 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] 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.

[3] 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*, October 27–30 2008.

[4] 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.

[5] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations." RFC 3135 (Informational), June 2001.

[6] D. A. Maltz and P. Bhagwat, "TCP Splice for application layer proxy performance," *Journal of High Speed Networks*, vol. 8, no. 3, pp. 225–240, 1999.

[7] J. P. Rohrer, E. Perrins, and J. P. G. Sterbenz, "Performance Comparison of Transport Protocols for Airborne Telemetry Networks," in *Proceedings of the International Telemetering Conference*, October 27–30 2009.

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

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

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

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

[12] J. Postel, "Internet Protocol." RFC 791 (Standard), Sept. 1981. Updated by RFC 1349.

[13] R. Callon and H. Braun, "Guidelines for the use of Internet-IP addresses in the ISO Connectionless-Mode Network Protocol." RFC 1069, Feb. 1989.

[14] A. Jabbar and J. P. G. Sterbenz, "Evaluating AeroRP: A Geolocation Assisted Aeronautical Routing Protocol for Highly Dynamic Telemetry Environments," in *Proceedings of the International Telemetering Conference*, October 27–30 2009.