Cross-Layer Architectural Framework for Highly-Mobile Multihop Airborne Telemetry Networks

Justin P. Rohrer, Abdul Jabbar, Erik Perrins, James P.G. Sterbenz
Department of Electrical Engineering & Computer Science
The University of Kansas
Lawrence, KS 66045
E-mail: {rohrej|jabbar|esp|jpgs}@ittc.ku.edu

***Abstract*—Highly dynamic mobile wireless networks present unique challenges to end-to-end communication, particularly caused by the time varying connectivity of high-velocity nodes combined with the unreliability of the wireless communication channel. Addressing these challenges requires the design of new protocols and mechanisms specific to this environment. Our research explores the tradeoffs in the location of functionality such as error control and location management for high-velocity multihop airborne sensor networks and presents cross-layer optimizations between the MAC, link, network, and transport layers to enable a domain specific network architecture, which provides high reliability for telemetry applications. We have designed new transport, network, and routing protocols for this environment: TCP-friendly AeroTP, IP-compatible AeroNP, and AeroRP, and show significant performance improvement over the traditional TCP/IP/MANET protocol stack.**

I. 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 with multiple sources and 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 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]. In particular, the current Internet protocols are unsuitable for the specific constraints and requirements of the aeronautical environment 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, any 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. 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 [4].

This paper presents the design and evaluation of a set of protocols for this environment: AeroTP – a TCP-friendly transport protocol with multiple reliability and QoS modes, AeroIP – an IP-compatible network protocol (addressing and forwarding), and AeroRP – a routing protocol that exploits location information in the test environment to mitigate the short contact times of high-velocity nodes. These protocols are assumed to run over the evolving iNET TDM MAC.

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 the airborne node location and trajectory. Previous research has developed several intelligent network protocols in the context of mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs) that attempt to exploit additional information available [5], [6]. However, in order to achieve this, we need to facilitate cross-layering across the multiple layers. For example, location and trajectory information can be used to find better paths if there exists a mechanism, either an implicit or explicit, for information exchange between the network and physical layer. As discussed in the literature, 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.

The rest of the paper is organized as follows: section I-A presents specific challenges to reliable network communication

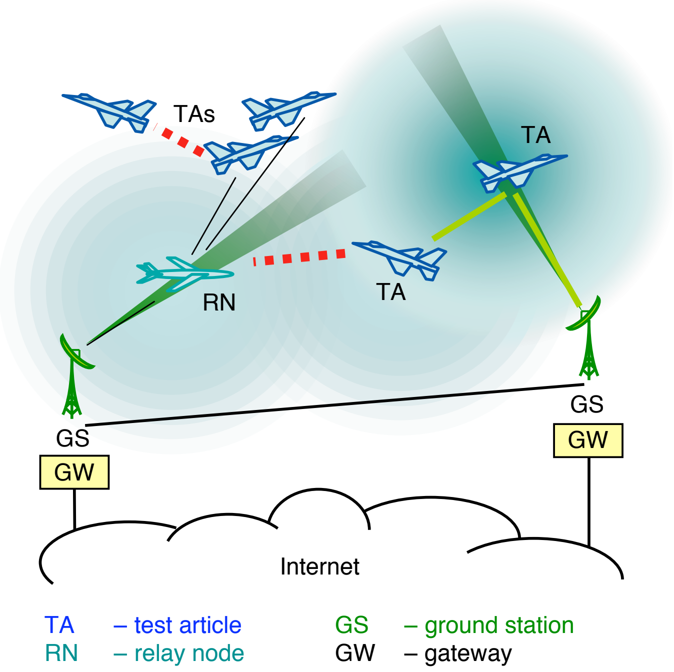


Fig. 1. iNET test and evaluation environment

in the iNET scenario. This is followed by a discussion of the current Internet architecture in section I-B and its inability to meet the demands of telemetry networks. In section II we present the architecture of AeroTP and AeroNP: cross-layer aware transport and network protocols for aeronautical telemetry. We also present AeroRP: a lightweight routing protocol leveraging node location information. Finally in section III, we present simulation results comparing AeroRP to traditional MANET protocols in a highly dynamic network.

A. *Networking Challenges in Airborne Telemetry Networks*

A typical T&E (test and evaluation) telemetry network as depicted in Figure 1 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). They house omnidirectional antennas with relatively short transmission range. The GSs are located on the ground and typically have a higher transmission range than that of a TA due to the large steerable antennas. In point-to-point communication mode, the GS tracks a given TA across some geographical space. However, due to the narrow beam width of the antenna, they can only track one TA at any given time. The GS also houses a gateway (GW) that connects the telemetry network to 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 while tracking a TA. The relay nodes are dedicated airborne nodes to improve the connectivity of the network. These nodes have additional energy needed to forward data from multiple TAs and can be arbitrarily placed in the network. The flow of 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:

1) *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.
2) *Constrained bandwidth*: Due to the limited spectrum available and the high volume of data that is sent from TA to GS, the network in general is severely bandwidth constrained.
3) *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.
4) *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.

In Table I, we use the numerical values from the network characteristics of iNET [2] to determine the stability of the links. It is seen that even with optimistic transmission range, the contact duration between two neighboring nodes can be as low as 15 seconds. Note that in a multihop scenario with lower transmission power, the contact duration between a test article and ground station can be far less.

TABLE I
LINK STABILITY ANALYSIS

Scenario	**Tx range [nmi]**	**Relative velocity**	**Contact duration [sec]**
Single hop best case			
TA – GS	140	400 knots	2520
TA – TA	15	800 knots	135
Single hop worst case			
TA – GS	100	Mach 3.5	300
TA – TA	10	Mach 7.0	15

B. *Shortcomings of Current Internet Protocols & Architecture*

1) Transmission Control and User Datagram Protocols: The most widely used transport protocol in the Internet is the Transmission Control Protocol (TCP) [7], [8], which is designed for terrestrial wired networks. 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. Even after the handshake is completed, TCP's slow-start algorithm prevents full utilization of the available bandwidth for many RTTs. TCP also assumes that all loss is due to congestion, and its 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. TCP's flow control requires a reliable ACK stream, which limits its ability to handle highly-asymmetric links even when the data is flowing in the high-bandwidth direction. The practical limit to asymmetry for TCP flows is about 75:1 [9]. 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 suffering a bit-error. TCP was not designed with intermittent connectivity in mind. Short-term link outages invoke congestion control and repeated retransmission timer back-offs, which results in an inability to detect link restoration and begin utilizing the link in a timely manner [10]. A longer link outage results in TCP dropping the connection. Varying RTT can also pose a problem for TCP, because if the actual RTT becomes much larger than the current estimate, TCP will incorrectly assume a packet loss and retransmit unnecessarily as well as reduce the congestion window. Hence, many standard TCP mechanisms are unsuitable for wireless networks in general and T&E environment in particular.

The other commonly used Internet transport protocol is the User Datagram Protocol (UDP) [11]. UDP is far simpler than TCP, but does not offer any assurance or notification of correct delivery. It does not do any connection setup, congestion control, or data retransmission. Because of this it does not need to maintain consistent state at both ends of the connection, nor does it do flow control, so the need for ACKs to self clock is eliminated completely. 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.

In the T&E environment we expect to have multiple classes of traffic that have different characteristics, different tolerance of loss, and different priorities. Neither TCP or UDP have the capability to provide differentiated levels of precedence or QoS to meet these 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) [13], from which we can draw some mechanisms but are only a partial solution.

2) Internet Protocol (IP): The traditional wired Internet uses IP at the network layer, with various routing protocols such as OSPF, RIP, and BGP. TCP over IP adds a header of 40 bytes per packet. This overhead becomes significant if there are many small packets (e.g. control traffic), which is the case with the per-segment acknowledgements of TCP. 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 as described above, but also network routing. Internet protocols do not support dynamic topologies, requiring convergence of the routes, which is not suitable for airborne telemetry environment. Furthermore, the current architecture does not support explicit cross-layer information exchange to leverage unique information available in the network such as position and trajectory.

3) Ad Hoc Routing Protocols: In order to support mobile ad hoc wireless networks (MANETs), several routing protocols have been developed that adapt to changes in topology. Reactive routing protocols such as AODV [14] and DSR [15] attempt to construct source-to-destination path on demand and are not suitable because of the delay involved in finding paths and because such paths may not be valid for very long in a highly-dynamic network. On the other hand, proactive routing protocols such as DSDV [16] and OLSR[17] forward packets on a hop-by-hop basis and depend on global route convergence. This generates excessive overhead due to frequent route updates (assuming convergence is even possible) and is not suitable for a bandwidth-constrained telemetry network.

There are several other protocols that adapt to mobility by forwarding packets one hop at a time without attempting to construct the entire path. These include simplistic approaches such as flooding and other greedy algorithms that send multiple copies in the network. More complex routing schemes leverage specific information from the network. Most notable are the location-based routing protocols such as LAR, DREAM, SIFT, and GRID [18]–[22] that use GPS coordinates of the nodes to determine the next hop. However, in the previous studies none of the above protocols have been tested at speeds as high as Mach 7. In Anticipatory Routing [23] the authors present a routing scheme to track highly mobile endpoints that reach the reactive limit where the speed of the nodes is comparable to time it takes for the location tracking to converge upon the position of the node. This is an extreme case that does not apply to the current scenario as shown from the contact durations in Table I.

II. AEROTP, AERONP, AND PROPOSED ARCHITECTURE

This section describes a new set of protocols designed for the aeronautical T&E environment: AeroTP TCP-friendly transport protocol, AeroNP IP-compatible network protocol, and AeroRP routing protocol for highly-dynamic airborne nodes.

A. AeroTP: TCP-Friendly Transport Protocol

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 seamless splicing with conventional TCP at the telemetry network edge in the GS and on the TA. Thus it transports TCP and UDP through the telemetry network, but in an efficient manner that meets

[1] Note that we use the term ”TCP-friendly” in a more general sense than the established term ”TCP-friendly rate control (TFRC) [24]

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 as described in section II-A2.

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.

1) 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 iNET. 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 section II-C for AeroNP. Error control is fully decoupled from rate control, and is service specific as described below.

2) Segment Structure and Gateway Functionality: AeroTP is *TCP-friendly*, meaning it is designed to efficiently interoperate with TCP and UDP at the border of the Ground Network (gNET) and Telemetry Network (TmNS), and at the border of the TmNS and Test Article (TA) networks. To support this, gateway functionality [25], [26] provides IP–AeroNP translation [27] and TCP/UDP–AeroTP splicing. A preliminary design of the AeroTP segment is shown in Figure 2. Since bandwidth efficiency is critical, AeroTP does not encapsulate the entire TCP/UDP and IP 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 semantics are passed through, such as the source and destination port number, TCP flags, and the timestamp.

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 error check) field is a strong CRC (cyclic redundancy check) 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 where a corrupted payload can be corrected on an end-to-end basis using FEC. A CRC 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. This method of error detection and correction implies that AeroNP does not necessarily drop corrupted packets at intermediate hops, which is a key difference from IP forwarding policy [28].

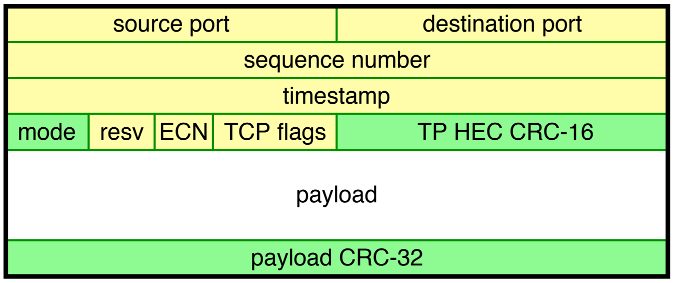


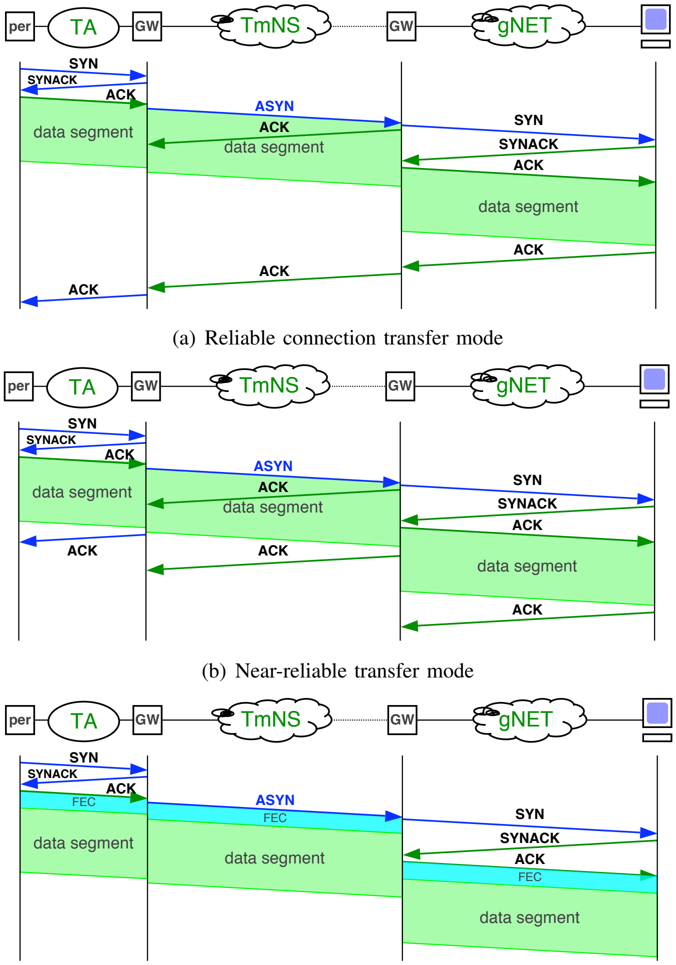
Fig. 2. AeroTP segment structure

3) 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 propose multiple transfer modes that are mapped to different traffic classes:

1) Reliable connection: end-to-end acknowledgements using ACK passthrough or custody transfer
2) Near-reliable connection: split ARQ; gateway immediately returns TCP ACKs with no buffering for TCP side
3) Quasi-reliable connection: transport-layer erasure coding, either sequential or multipath
4) Unreliable connection: best effort over FECed links
5) Unreliable datagram: stateless best effort for UDP compatibility

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 [29] 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 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



(a) Reliable connection transfer mode

(b) Near-reliable transfer mode

(c) Quasi-reliable transfer mode

Fig. 3. AeroTP transfer modes

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 erasure coding, across multiple paths if available [30]. 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 error-correcting code.
- ***Unreliable datagram*** mode is intended to transparently pass UDP traffic, and no AeroTP connection state is established at all.

Figure 3 shows the difference between reliable, near-reliable, and quasi-reliable transfer modes. The near-reliable mode is able to make more efficient use of the wireless channel, at the cost of acknowledging TCP packets before they reach their final destination.

B. *Cross-Layer Mechanisms*

Despite the fact that link-load aware routing was developed as a part of the first ARPANET routing protocol [31], cross-layered routing utilizing link and physical layer information in route selection is not widely used. The reason for this tends to be two fold: firstly, intelligent cross-layer aware network protocols tend to be inherently complex and secondly, in wired networks, physical links are highly reliable and are frequently over-provisioned. This has led to shortest path being most widely deployed routing algorithm. It has been noted that this is clearly not enough for effective routing in wireless networks [32]. Hence the need to exploit available information through cross-layering to make better forwarding decisions at each node.

In Table II, we consider the potential knobs at each layer that enable higher layers to influence certain mechanisms at lower layers, based on the information made available through dials. For example, the transport layer influences path selection through the *forwarding mode* knob, thus requesting a certain level of reliability for given data flow.

TABLE II
KNOBS AND DIALS FOR A TELEMETRY NETWORK STACK

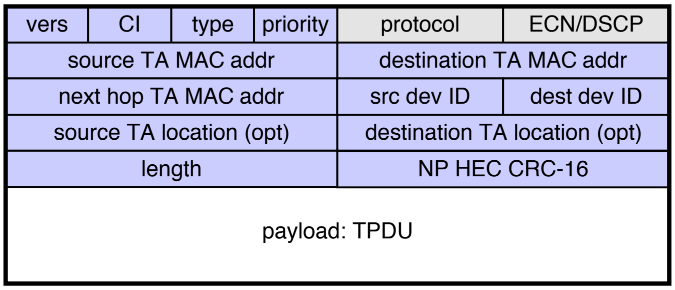
Layer	**Knobs**	**Dials**	**Influencing Layer**
transport	reliability mode	service requirements	application
network	forwarding mode	path characteristics	transport
link & MAC	ARQ & FEC settings	link characteristics	network
physical	coding	channel conditions, coding schemes	link

In the proposed architecture, we employ cross-layer optimizations not only among the transport (AeroTP) [33] and network (AeroNP) protocols, but also with the MAC and PHY layer. This involves investigating the tradeoffs in type and strength of FEC (forward error correction) at the PHY layer with respect to channel conditions and BER (bit error rate), as well as optimizing TDM parameters and slot assignment based on the transfer mode of AeroTP and QoS parameters (precedence and service type) of AeroNP. Furthermore, if the transport protocol were to erasure code across multiple TA-GS paths [33] when available, it requires coordination of GS and MAC slot assignment with AeroNP routing. Finally, the support for multicast and broadcast requires coordination of AeroNP routing with the broadcast capabilities of the MAC.

C. *AeroNP and AeroRP Network Protocols*

AeroNP is designed for the telemetry network environment and includes the packet format and a dynamic location-aware multihop routing protocol AeroRP. The small contact duration between two TAs indicates the need for an intelligent multihop routing protocol for reliable communication over a highly dynamic physical topology.

1) Header Format, Addressing, and IP Transparency: AeroNP is an *IP-compatible* network layer with the additional functionality needed for aeronautical telemetry. A preliminary format of the AeroNP packet, shown in Figure 4, is 32 bits wide.



<table>
<tr><td>vers</td><td>CI</td><td>type</td><td>priority</td><td colspan="2">protocol</td><td colspan="2">ECN/DSCP</td></tr>
<tr><td colspan="4">source TA MAC addr</td><td colspan="4">destination TA MAC addr</td></tr>
<tr><td colspan="4">next hop TA MAC addr</td><td colspan="2">src dev ID</td><td colspan="2">dest dev ID</td></tr>
<tr><td colspan="4">source TA location (opt)</td><td colspan="4">destination TA location (opt)</td></tr>
<tr><td colspan="4">length</td><td colspan="4">NP HEC CRC-16</td></tr>
<tr><td colspan="8">payload: TPDU</td></tr>
</table>

Fig. 4. AeroNP packet structure

The *version* is the AeroNP protocol version, the congestion indicator (*CI*) is set by each node to notify the neighboring nodes of its congestion level as discussed later. The *type* and *priority* fields specify the QoS level of a given packet. The number of QoS classes can be customized for a given scenario. *Protocol* is the demux protocol (id) to which AeroNP hands off the packets. In order to be IP compatible, the *ECN/DSCP* (explicit congestion notification/diffserv code point) nibble is carried over from the IP header. Since the MAC is based on TDM, an AeroNP packet is inserted directly into a TDM slot, and thus contains the *MAC addresses*: *source*, *destination*, and *next hop*. Significant efficiency can be gained if the AeroNP header does not carry the 32-bit source and destination IP addresses (or the even worse 128 bit addresses for IPv6). By performing an ARP-like address translation process, the IP address can be mapped to MAC addresses in the gateway. However, each TA can have multiple peripherals, each of which has an IP address. Therefore, we include a device id field in the header, and the ⟨*MAC-address*, *device-id*⟩ tuple is mapped to IP address at the gateway. While dynamic mapping procedures are possible, it is more efficient to preload the translation table at the beginning of each test. Optionally, *source* and *destination location* is included, which can be the GPS coordinates that are used in location aware routing. The *length* indicates the actual length of the header in bytes. A strong check on the integrity of the header, *HEC*, is included to protect against bit errors. Unlike Internet protocols [28], the default behavior of the AeroNP is to forward the errored packets to the transport layer instead of dropping them at the network layer. This permits FEC at the transport layer to correct errors [33].

D. Routing Algorithm

As discussed previously, existing routing mechanisms generate significant overhead and do not converge quickly for a highly dynamic topology and hence are not suitable for telemetry networks. We propose developing a proactive routing protocol that leverages location information combined with limited updates to build a next-hop routing table. In addition to the bandwidth constraints, telemetry networks may also impose security limitations on the extent of location and trajectory information made available and its advertisement in the network header. We propose several alternatives for cases where no information regarding the location is available.

The basic operation of the proposed routing protocol is to maintain a table of available neighbors at any given point in time. The primary mechanism used by the node to determine its neighbors is snooping. In a wireless TDMA network, a node that is not transmitting listens to all transmissions on the wireless channel. In the proposed scheme, when a node hears a data packet over the air interface, it adds the source MAC address of the decoded packet to the neighbor table. This implies that if a node can hear transmissions from a node, it can also communicate with that node. Stale entries are removed from the neighbor table if no transmissions from a node are heard for a predetermined interval of time (related to the anticipated contact duration).

The second part of the protocol is to find the appropriate next hop to forward the data packets. In order to to forward the packets towards a specific destination, additional information such as location data or route updates is required. Below is a list of mechanisms, in the order of stealthiness, through which such information can be obtained.

1) nodes include state vector explicitly as a field in the header of AeroNP protocol
2) nodes include only their GPS location as a field in the network packet header
3) The GS periodically broadcasts (optionally on a encrypted channel) the state vector of all the nodes so that each node can predict its connectivity ahead of time
4) No location information is made available; instead nodes exchange their neighbor table upon contact

In the first two cases, nodes discover both the neighbors and their locations by snooping network packets. In lightly loaded network conditions, a periodic hello message is sent by a node to inform other nodes of its presence. The data packets are forwarded to the node that is nearest to the destination as calculated from GPS coordinates. We assume that the all nodes are preprogrammed with the location of GSs. However, there is a time lag during which the node will snoop out its neighbors. In the third case, the GS broadcasts the topology information ahead of time so that the each node can predict its neighbors' trajectory and forward the data packets to the appropriate nodes. In the last scenario, we assume that no location or trajectory information is made available due to security policy. Instead, when two nodes are in transmission range, they exchange their neighbor table. Thus each node can build a partial routing table. In case of a connected network, each node will have the complete routing table after some initial learning phase. However, this approach could generate significant overhead due to dynamic nature of the telemetry network and may have low efficiency due to the delay involved in learning the routes.

Ground Stations are special nodes in this network. They listen to all the transmissions and forward packets that are destined to other GSs. In other words, GSs are universal sinks and may have the same MAC address. For uplink data, a GS forwards data to the node that is closest to the destination node. The GS is aware of the location of all node either from mission planning or learns it during the test while tracking various TAs.

Relay nodes, if present, are always the default next-hop. They accept data from all the TAs and forward them directly to the ground station or another TA. Since the GS has narrow beam width and can only track one TA, it is more efficient for the GS to track relay nodes and have individual TAs forward the data to relay nodes. Given the varied service requirements of telemetry, we expect that the routing protocol should support multiple modes for both open and secure scenarios.

1) Quality of Service: The wireless links in the telemetry network are bandwidth-constrained and are often underprovisioned for the traffic generated during a field test. Hence, it is essential to implement a quality of service mechanism in this network to ensure that high priority data such as command and control can be reliably delivered. The AeroNP protocol uses two fields in the header to specify the quality of service of data packets in the network: data *type* (e.g command and control, telemetry) and *priority* with in a given type. The application requirements determine the type and priority for a given data flow and is passed to the network layer through the transport layer (AeroTP) via out-of-band signaling. The scheduling at nodes is a weighted fair queue based on type and priority.

2) Broadcast and Multicast: The AeroNP protocol [27] will support both broadcast and multicast natively. The typical all-ones MAC address is chosen as the broadcast address. Similarly, a range of MAC addresses are assigned to sub-groups in the network. These multicast address groups are generally preprogrammed in the nodes and GS. Note, however, that given the highly dynamic nature of the network, for sparse networks multicast may not achieve any significant benefit over a simple broadcast in terms of efficiency.

3) Congestion Control: The telemetry networks are often bandwidth constrained. For a given test scenario, individual TAs are under provisioned. Therefore, in a heavily loaded network with little bandwidth to spare, multi-hop routing can induce severe congestion in the nodes involved in multi-hop forwarding. A MAC-level solution would be to assign more slots to the forwarding nodes than the non-forwarding nodes. Since this is too complex in this highly dynamic environment, we propose a simple congestion control mechanism at the network layer using *congestion indicators* and *back pressure*.

In the first mechanism, the node uses the *CI* (congestion indicator) field to indicate its own congestion level. All packet transmissions from a node carry the CI field along with the type and priority of the data. Neighboring nodes eavesdrop on the transmission and are made aware of the congestion at a given node. If a node is congested, the neighbors back off if the data that they have is of equal or lesser priority; higher priority data is nevertheless forwarded to a congested node.

The second mechanism through which congestion control is achieved in the telemetry network is back pressure. Each node eavesdrops on the neighboring nodes and knows when one of its neighbor is congested and has stopped forwarding its packets. This is possible because the source MAC address is carried in the header field. Barring malicious behavior, the source node then backs-off. Similarly, in a multi-hop scenario, if a bottleneck is encountered, each intermediate hop either stops or slows down its transmissions to the congested node successively until the source of the traffic is reached.

III. Preliminary Simulation Results

Using the ns-2 simulator [34] we have implemented AeroRP and compared its performance to the traditional MANET routing protocols AODV (Ad-hoc On-demand Distance Vector) and DSDV (Destination-Sequenced Distance Vector). AODV only finds routes as needed, while DSDV updates its routing tables as the topology changes.

A. Topology Setup

We use 60 wireless nodes randomly distributed over an area that is 150 km by 150 km, and a single stationary sink node which is located in the center of the simulation area representing a ground station. The first 60 nodes follow a modified random-waypoint movement model for a total of 2000 seconds. The pause times are zero to more accurately represent the movement patterns of aircraft. We used two different test cases: In the first case each node's speed is randomly selected to be between Mach 0.3 and Mach 3.5 (100 to 1200 m/s) for each leg of the random-waypoint movement; in the second case the nodes always move at Mach 3.5. Each node has an omnidirectional antenna with a maximum range of 15 nautical miles (27.8 km). This yields a total coverage ratio of 6.5:1. The velocities and radio transmission ranges are based on the iNET architecture [2], and we selected the node density such that the network would not be partitioned most of the time. In our simulations we found that on average a node was partitioned from the sink node 6.6% of the time.

B. Traffic Setup

Each node sends data at a constant 0.2 Mb/s using 1000-byte packets resulting in 25 packets being sent per node per second, and a combined total of 1,350,000 packets for all nodes over the course of the simulation. We set the wireless link bandwidth to 11 Mb/s so that congestion would not be a factor in the results and verified that this is the case in our simulations. Data transmission does not start until the 1050th second to allow through mixing of the nodes as well as route table population for DSDV. Data transmission stops at the 1950th second, and the simulation runs for an additional 50 seconds to allow buffered packets to be delivered.

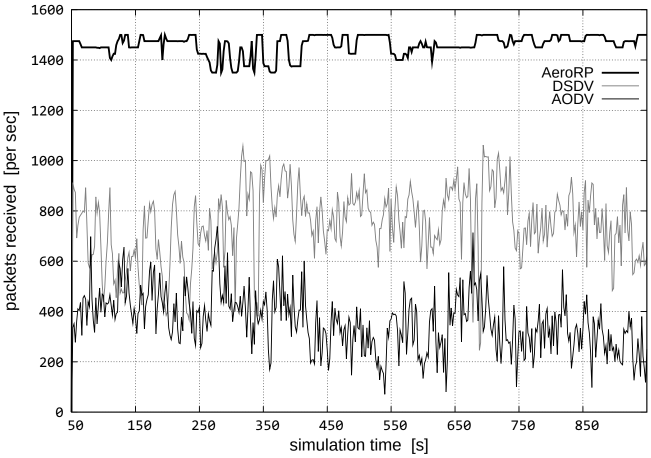


Fig. 5. Packet delivery rate for Mach 0.3 to 3.5

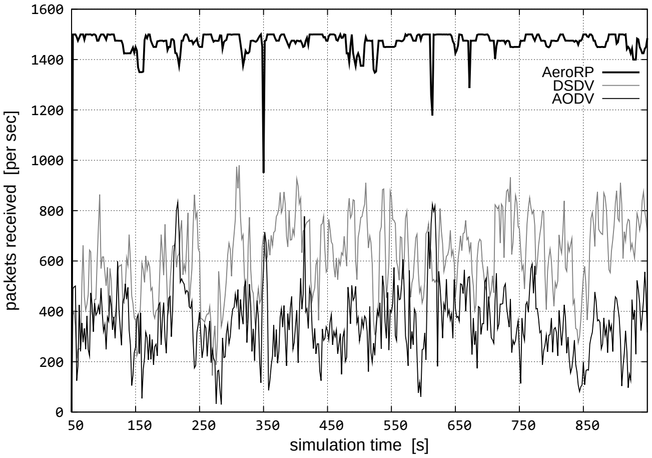


Fig. 6. Packet delivery rate for Mach 3.5

C. Results

In the first case using AODV only 3.25×10^5 out of 1.35×10^6 packets (24%) were received by the sink node. DSDV performed better with 6.74×10^5 packets (50%) being received. With geolocation assisted predictive routing (AeroRP), 1.31×10^6 packets (97%) were received at the sink node. Figure 5 shows the packet delivery rate (for an aggregate source rate of 1500 packets per second) for these three protocols when the speed is varied between Mach 0.3 and Mach 3.5. Figure 6 show the results for the second case in which nodes move at a constant speed of Mach 3.5. In this case AODV received 3.17×10^5 packets (23%), DSDV received 5.54×10^5 packets (41%), and AeroRP received 1.32×10^6 packets (97%). The overhead incurred by AODV and DSDV for both cases is plotted in Figures 7 and 8 in terms of aggregate bytes transmitted per second. AODV incurs greater overhead due to the fact that it is on demand and therefore its overhead is directly proportional to the mobility. DSDV incurs less overhead than AODV because of the periodic nature of its update messages, which are not mobility dependent. However, as the simulations progress DSDV is unable to converge due to the highly dynamic topology and generates increasing numbers of update messages. Unfortunately, we cannot plot the comparison to AeroRP at this time because the size and format of the routing messages has not been finalized and remains a part of the future work. Preliminary evaluations show that the AeroRP overhead is much lower than AODV or DSDV since it does not transmit event-based updates.

Based on our examinations of the simulation trace data, the poor performance of AODV and DSDV is caused by the timescale on which they operate. In both cases they can take 30 seconds to 5 minutes to determine that a route has failed and reroute [35]. In an environment in which paths may only be stable for a few seconds, these protocols simply cannot keep up. While it is possible to minimize their route convergence time using modification such as shorter update intervals and faster dead-link detection, this would inevitably lead to increased overhead.

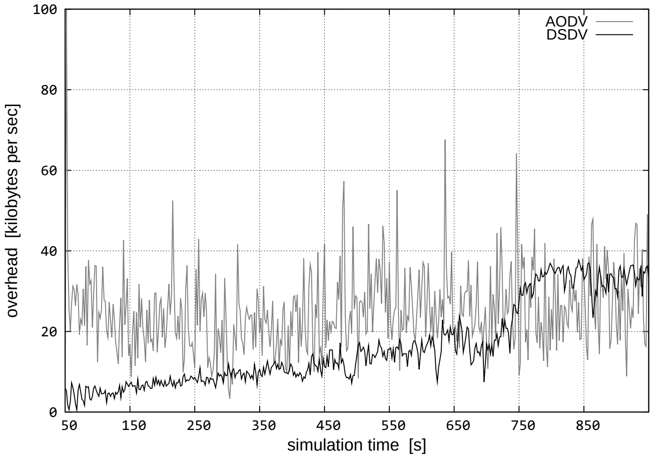


Fig. 7. Routing overhead for Mach 0.3 to 3.5

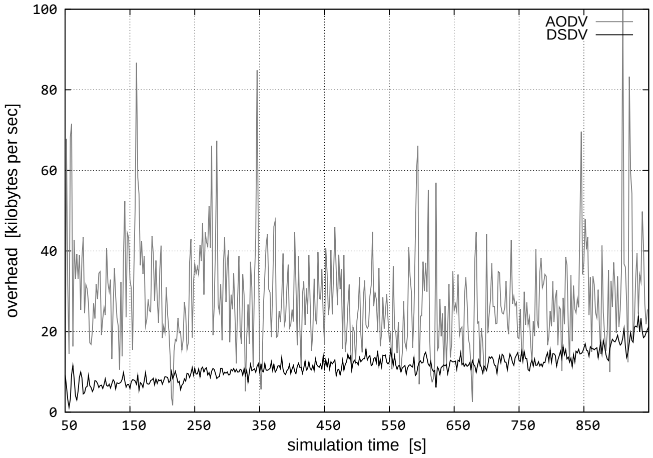


Fig. 8. Routing overhead for Mach 3.5

IV. 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 bandwidth. Typical MANET routing protocols such as AODV and DSDV are not designed for topologies that are as dynamic as the ones found in aeronautical telemetry environments. In this paper, we discuss a new protocol architecture that addresses these issues with domain-specific transport and network layers. It is observed that the exchange of information across layers provides significant benefit in the aeronautical environment. We have developed domain-specific transport (AeroTP) and network (AeroNP) protocols to leverage cross-layer information in optimizing end-to-end performance. By predicting when links will be available based on trajectory information, as well as actively listening for nearby nodes, AeroRP can send data opportunistically towards its destination and make much more efficient use of available network capacity. We performed simulations that show that the new routing protocol performs significantly better then traditional MANET protocols in this environment. In the future we will perform more extensive AeroRP simulations with varying node densities, mobility models, etc. We are also working on simulations of AeroTP and will eventually be able to simulate the entire Aero protocol suite together.

Acknowledgments

Thanks 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 19 2004.

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

[4] iNET Working Group, “http://www.inetprogram.org.”

[5] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, “A review of routing protocols for mobile ad hoc networks,” *Ad Hoc Networks*, vol. 2, pp. 1–22, January 2004.

[6] L. Iannone, R. Khalili, K. Salamatian, and S. Fdida, “Cross-layer routing in wireless mesh networks,” *Wireless Communication Systems, 2004. 1st International Symposium on*, pp. 319–323, 2004.

[7] J. Postel, “Transmission Control Protocol.” RFC 793 (Standard), Sept. 1981. Updated by RFC 3168.

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

[9] 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.

[10] 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.

[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] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing.” RFC 3561 (Experimental), July 2003.

[15] D. Johnson, Y. Hu, and D. Maltz, “The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4.” RFC 4728 (Experimental), Feb. 2007.

[16] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers,” in *SIGCOMM ’94: Proceedings of the conference on Communications architectures, protocols and applications*, (New York, NY, USA), pp. 234–244, ACM, 1994.

[17] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR).” RFC 3626 (Experimental), Oct. 2003.

[18] W.-H. Liao, J.-P. Sheu, and Y.-C. Tseng, “GRID: A fully location-aware routing protocol for mobile ad hoc networks,” *Telecommunication Systems*, vol. 18, no. 1-3, pp. 37–60, 2001.

[19] M. de la Fuente and H. Ladiod, “A performance comparison of position-based routing approaches for mobile ad hoc networks,” *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pp. 1–5, 30 2007-Oct. 3 2007.

[20] L. Galluccio, A. Leonardi, G. Morabito, and S. Palazzo, “A mac/routing cross-layer approach to geographic forwarding in wireless sensor networks,” *Ad Hoc Netw.*, vol. 5, no. 6, pp. 872–884, 2007.

[21] M. Mauve, A. Widmer, and H. Hartenstein, “A survey on position-based routing in mobile ad hoc networks,” *Network, IEEE*, vol. 15, no. 6, pp. 30–39, 2001.

[22] M. Yuksel, R. Pradhan, and S. Kalyanaraman, “An implementation framework for trajectory-based routing in ad hoc networks,” *Ad Hoc Networks*, vol. 4, no. 1, pp. 125–137, 2006.

[23] F. Tchakountio and R. Ramanathan, “Anticipatory routing for highly mobile endpoints,” in *WMCSA ’04: Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications*, (Washington, DC, USA), pp. 94–101, IEEE Computer Society, 2004.

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

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

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

[27] 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.

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

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

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

[31] J. McQuillan, I. Richer, E. Rosen, B. Beranek, and N. Inc, “The New Routing Algorithm for the ARPANET,” *IEEE Transactions on Communications*, vol. 28, no. 5, pp. 711–719, 1980.

[32] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris, “Performance of multihop wireless networks: Shortest path is not enough,” in *Proceedings of the First Workshop on Hot Topics in Networks (HotNets-I)*, (Princeton, New Jersey), ACM SIGCOMM, October 2002.

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

[34] “The network simulator: ns-2.” http://www.isi.edu/nsnam/ns/, December 2007.

[35] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” *MobiCom ’98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 85—97, Oct 1998.