Performance Comparison of Routing Protocols for Transactional Traffic over Aeronautical Networks

Yufei Cheng, Egemen K. Çetinkaya, and James P.G. Sterbenz
Department of Electrical Engineering & Computer Science
The University of Kansas
Lawrence, KS 66045
{yfcheng,ekc,jpgs}@ittc.ku.edu

ABSTRACT

Emerging airborne telemetry networks present several challenges such that traditional MANET protocols do not perform well. On the other hand, application traffic characteristics are expected to be identical in this domain. Transactional application traffic performance has been studied for traditional MANETs, however this type of traffic has not been studied for domain-specific MANET scenarios. We present the simulation model for transactional traffic used in highly-mobile environments. Using the ns-3 simulator, we examine the performance of a domain-specific routing protocol, AeroRP, to route transactional traffic over the aeronautical environment and present performance comparison with two reactive (AODV and DSR) and two proactive (DSDV and OLSR) MANET routing protocols.

I. INTRODUCTION AND MOTIVATION

Modern airborne telemetry networks require the multihop routing paradigm to cope with the challenges faced in this highly-dynamic environment [1, 2, 3]. To mitigate the impacts of this highly-dynamic environment the AeroRP routing protocol exploits geolocation of test articles (TA) [4]. The performance of the AeroRP protocol is further analysed and shown that it outperforms traditional MANET routing protocols [5, 6]. On the other hand, different application traffic characteristics are expected to be identical with other network scenarios, that is CBR (constant bit rate) or VBR (variable bit rate) traffic can be sent from TAs to the GSs (ground stations). However, short contact durations among TAs challenge the performance of transactional traffic such as HTTP.

The performance of routing transactional traffic in MANETs with low mobility has been examined [7, 8], but there is no previous work that has considered transactional traffic over airborne networks with high velocity TAs. In this paper we present the interaction of transactional traffic with AeroRP in a high-mobility environment and compare the performance using legacy MANET routing protocols in the same set of scenarios. We use the ns-3 network simulator [9] to analyse the performance of transactional traffic in this highly-dynamic airborne environment. The ns-3 simulator is a recent replacement for the open-source ns-2 simulator with many advantages, including modular extensibility and mixed wired and wireless models. However, it lacks some well-known models [10]. AODV and OLSR are the MANET routing

protocols implemented in ns-3 initially, with DSDV [11] and DSR (dynamic source routing) [12, 13] developed at The University of Kansas. In this paper we show comparison among these traditional MANET protocols and show that certain modes of AeroRP with transactional traffic outperform the MANET routing protocols in terms of successful packet delivery and delay in this highly-dynamic environment.

In the remaining part of this paper, we present related work in application traffic models in Section II. Next, we present our implementation of the HTTP traffic generator for the ns-3 network simulator in Section III. We present *preliminary* performance results of the geographic-based AeroRP routing protocol for HTTP transactional traffic and compare the performance of AeroRP to legacy MANET protocols in Section IV. Finally, we conclude the paper in Section V.

II. BACKGROUND AND RELATED WORK

Routing protocol performance is commonly analysed with CBR traffic and bulk data transfer models, however, the network loads from Web-based applications are typically characterised by bursty traffic flowing over connections with variable durations ranging from a few seconds to several minutes. Web access consists of frequent, short request-response traffic based on bursts of many small requests and frequently large responses. Web users tend to surf rapidly among sites, which are verified from both client [8] and server-side traces [14].

A transactional source traffic generator is an essential part of network simulation of the Internet. It is responsible for injecting synthetic traffic into the simulation according to a model of how application users would behave in certain circumstances. As one major contributor of transactional traffic, Hypertext Transfer Protocol (HTTP) is a pervasive application protocol and consumes a significant share of application flows in the Internet. Web traffic has transformed from plain-text web pages to large size pages with embedded objects. With the sustaining influence of HTTP over the Web, we need a HTTP source traffic model to accurately represent and simulate Web traffic.

A. HTTP

HTTP is a stateless protocol using the client-server paradigm. Its operation is transactional since the client sends a request to the server, which responds to the request with a response message. HTTP is an application layer protocol and presumes a reliable transport layer (i.e. TCP) for end-to-end data transfer. The original version HTTP/1.0 [15] uses a separate connection for each request-response transaction. HTTP/1.1 [16] *persistent* connections reduce the latency by allowing several request-response messages over a single TCP connection. Another performance enhancement in HTTP/1.1 is the *pipelining* of a series of requests on a persistent connection without waiting for a response between each request.

B. Transactional Traffic Generators

There are broadly two types of transactional traffic generators: *page-based* and *connection-based*. Page-based traffic generators only focus on the HTTP page details and fail to represent other major parameters of the HTTP traffic [17, 18]. On the other hand, connection-based models provide a better

alternative to page-based models [19] and their effectiveness has been shown [20, 21]. Connection-based transactional traffic models TCP connections in terms of connection establishment rates, request/response data sizes, and timing within the connection. We developed our HTTP traffic model based on TCP connections between web servers and clients, with each node acting as either server, client, or both. The model can run over both wired and wireless networks simulated with node mobility. To the best of our knowledge, previous studies have not considered high mobility of the nodes in the wireless medium.

C. Overview of Routing Protocols

MANET routing protocols can be categorised into two major categories: proactive and reactive protocols. Proactive routing protocols maintain routing information of all the nodes in the network and add new routes or update existing routes by periodically distributing routing tables globally; therefore, routes are ready to use instantaneously when needed. However, the cost is the large overhead involved with flooding routing tables over the network. Unlike proactive routing protocols, reactive routing protocols construct the route only when it is required; therefore, nodes in these routing protocols do not update their routing tables frequently and globally. Nodes in reactive routing protocols broadcast *route request* messages when a route is needed. Nodes with routes to the destination will send back *route reply* message with the route information, which is used to send the data packets. However, it involves delay in finding routes to new destinations and overhead to find the route.

Ad hoc On-Demand Distance Vector (AODV) [22] is a reactive distance-vector routing protocol that minimises the number of broadcast packets by finding routes on demand. Optimized Link-State Routing (OLSR) is a proactive routing protocol; therefore, routes to all destinations within the network are discovered and maintained before a packet is sent from source to destination [23]. Destination-Sequenced Distance Vector (DSDV) [24] protocol is a table-driven proactive protocol, thus it maintains a routing table with entries for all the nodes in the network and not just the neighbors of a node. The Dynamic Source Routing (DSR) [12] protocol is an on-demand routing protocol based on the source routing concept. The advantage of source routing is that it is loop-free and allows the intermediate nodes to maintain up-to-date route information.

The challenges posed by the aeronautical environment has requirements beyond the assumptions for which traditional MANET routing protocols are designed [1, 3]. AeroRP is a routing protocol specifically designed for this highly-dynamic environment [4]. It has been shown that AeroRP outperforms traditional MANET routing protocols in terms of packet delivery ratio, delay, and overhead [5, 6].

III. IMPLEMENTATION OF TRANSACTIONAL TRAFFIC GENERATOR

Web traffic shows bursty characteristics. The duration of a flow ranges from a few seconds to a few minutes while the packet size distribution consists of a high proportion of 1500 B for data packets and 40 B for ACK packets [19]; However, large responses can reach more than 100,000 B [17]. In our HTTP traffic generator implementation we consider this wide range of traffic characteristics via various parameters. We model TCP connections in terms of connection rate, the timing of request and response messages, as well as the file sizes transferred, and generate both HTTP 1.0 and HTTP 1.1 traffic. In our HTTP model the parameters are based on real-world HTTP traffic characteristics and can generate realistic traffic distributions or synthetic distributions based on user-specified values.

The model consists of three major parts: the distribution function generation model, the client application, and the server application. The distribution function generation model is responsible for generating HTTP parameters for simulation as shown in Table 1. The parameters are: number of web sessions, number of web pages, number of inline objects in each page, inter-request time, and object size. The number of web sessions is the total HTTP sessions in the simulation. The total number of objects consist of both the number of web pages and number of inline objects in each page, fitted by the *Gamma* distribution [19, 18]. The inter-request time is the inactive interval between HTTP requests, and is fitted well by a heavy-tailed *Weibull* distribution [18]. The object size parameter consists of request and response sizes and are both fitted by a *Lognormal* distribution [18, 19]. The mean response size is larger than request size.

Table 1: Transactional Traffic Model Parameters

Parameters	**Distributions**	**Model class**
Web sessions	User-defined	User-defined
Request size	Lognormal	HTTPFileSize
Response size	Lognormal	HTTPFileSize
Total object number	Gamma	HTTPNumPage
	Gamma	HTTPObjsPerPage
Inter-request time	Weibull	HTTPXmitRate

The client and server applications are responsible for the major functionality and generate the transactional traffic over the entire simulation process. The client application controls the parameters used in the simulation process, and is started when a new TCP connection is initiated. On the other hand, the server application starts from the beginning of the simulation. The HTTP simulation workflow is illustrated in Figure 1. When the HTTP model started, the client and server applications will be installed in client and server nodes respectively. The user can define the number of clients and servers, not restricted to one client corresponding to one server. In other words, each client can communicate with multiple servers, and one server can respond to multiple clients.

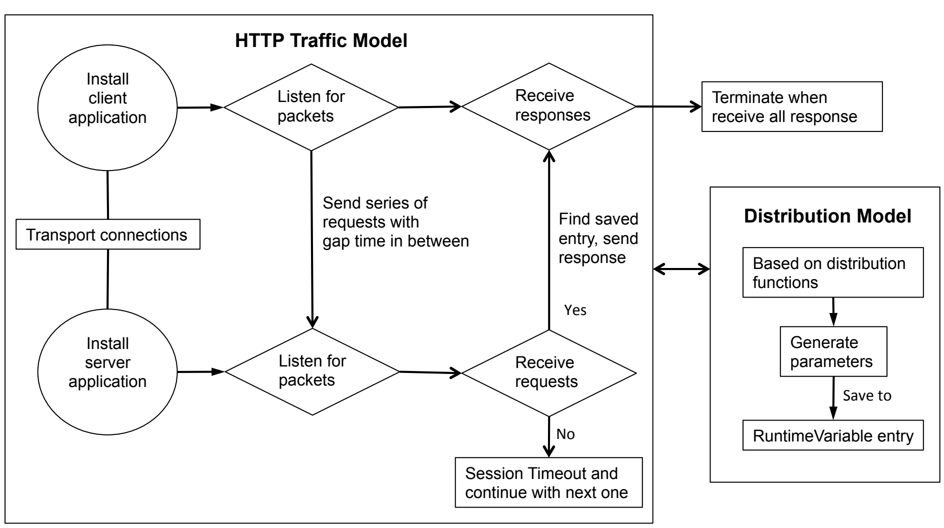

Figure 1: HTTP work flow

Each HTTP client will start by running HTTPRandomVariables, which we developed for generating HTTP parameters. This implementation of HTTP provides several ns-3 RandomVariable objects for specifying distributions of HTTP connection variables. The implementations were taken from source code provided by PackMimeHTTP in ns-2 [25] and modified to fit into the ns-3 RandomVariable framework. The model allows HTTP connection variables to be specified by any type of ns-3 RandomVariable, which now includes HTTP-specific random variables. HTTPRandomVariable includes several subclasses with each class responsible for sampling a single parameter. For each web session, the client first samples the number of objects for a specific TCP connection from the distribution of HTTPNumPage and HTTPObjsPerPage and sums up all the objects for all the number of pages. The client then checks the HTTP version defined by the user. For HTTP 1.0, it will send the next request only after receiving the response for previous request; for an HTTP 1.1 connection, the client will also sample the inter-request times based on HTTPTXmitRate and send requests without waiting for previous responses. For each of the requests, the client node will also sample the HTTP request and response sizes based on HTTPFileSize. The model will save the parameters generated to the RuntimeVariable table indexed with request size, globally accessible by all clients and servers, so they can synchronise with the parameters they use. The client sends the first HTTP request to the server and goes into the Listen state to wait for data transferred from the server. Once the client receives data from the server, it will set a timer to schedule the next request transaction and repeat the process until all the transactions are done or have a timeout without receiving any packets.

The server will start from the beginning of the simulation and listen for requests from the associated clients. When a request arrives, the server locates the RuntimeVariable entry as shown in Figure 1, indexed with the request file size, and retrieves the response size and the server-delay time in the entry. After this, the model schedules HTTP response traffic with the response size. The above process will repeat until the requests are exhausted, and then the TCP connection will be closed.

IV. SIMULATION RESULTS

In this section, we present the results of simulations conducted with the ns-3 network simulator [9] to compare the performance of AeroRP (without ground station updates and with the ferrying enabled [26]) to the traditional MANET routing protocols such as AODV, DSDV, DSR, and OLSR. The AeroNP extended header was used to include the geolocation information in the routing messages [26]. We analysed network performance under the transactional HTTP application traffic and compared the network performance with non-transactional CBR (constant bit rate) application traffic. We used fixed object sizes of 512 B and 10240 B for HTTP traffic instead of a distribution function to directly compare results to the CBR traffic. The simulation time is 2000 s with 100 s warmup time. The timeout value for TCP connections is set as 10 seconds. The topology setup consists of 5 to 60 mobile TA (test article) nodes that are randomly distributed in a 150 km^2 simulation area with a single stationary GS (ground station) at the center of the simulation area. The GS is installed with the HTTP client application and all the TAs are installed with the HTTP server application. The TAs follow the Gauss-Markov [27] mobility model with velocities ranging from 200 m/s to 1200 m/s. Each node has a transmission power configured to have 15 nmi (27.8 km) transmission range. All data are plotted with 95% confidence interval bars.

D. Baseline Simulations with CBR Traffic

We analysed the performance of different routing protocols in the highly-dynamic aeronautical environment with CBR application traffic, using the ns-3 OnOffApplication traffic class. The number of traffic flows is set to be equal to the number of TAs in a given scenario. All the TAs are configured to send 1 packet/s with a size of 1024 B. For the CBR traffic, we consider the network performance metrics of PDR (packet delivery ratio) and delay. PDR is the number of packets received divided by the number of packets sent at the application layer. Delay is the time it takes to deliver a packet from source to destination.

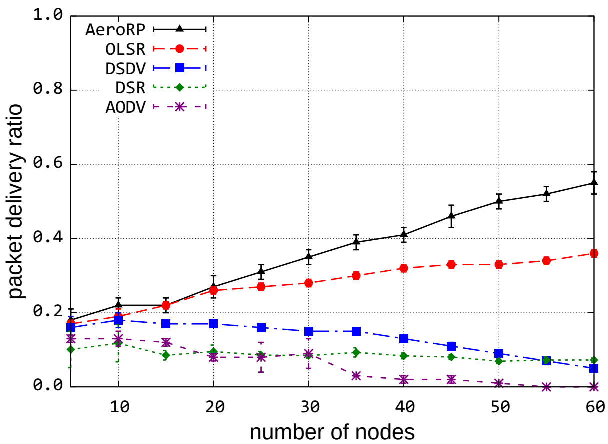

Figure 2: PDR of different number of nodes with CBR

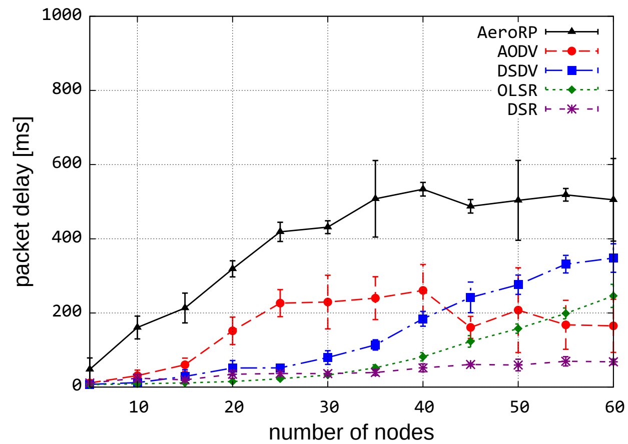

Figure 3: Delay of different number of nodes with CBR

Figure 2 shows the average PDR as the number of nodes increase when traveling at varying speeds of 200 m/s to 1200 m/s for CBR traffic using the 3-D Gauss-Markov mobility model. The node density of the network affects all of the routing protocols with AeroRP performing the best. The PDR for both AODV and DSDV decrease as the number of nodes increases; OLSR starts to decrease from 55 nodes. The PDR of DSR remains almost constant around 10% with increasing node densities. However, the PDR of AeroRP increases when the node number increases reaching 60% delivery at node number 60. This is most likely because with density increasing, the network is more connected, thus AeroRP can make accurate routing decisions based on a larger set of geolocation information from other nodes.

End-to-end delay of the routing protocol is shown in Figure 3. Delay of AeroRP is higher than the other routing protocols since AeroRP utilises ferry mode, in which the packet cache will store all unsent packets, which makes the delay higher when the packets with no route stack up in the queue. Note that this is the cost for higher PDR in which these packets do eventually get delivered. DSR has the lowest delay around 50 ms. DSDV, OLSR, and DSR stays below 200 ms for all the scenarios mainly because they use drop-tail queue with limited queue sizes.

E. Transactional Traffic Performance

For transactional traffic, we use the same set of scenarios from CBR traffic. For the object size, we chose 512 B as the starting point to fit in a single packet since the TCP model in ns-3 uses 536 B as fragmentation threshold. We test the object sizes of 1 and 20 packets which are 512 B and 10240 B in size

respectively. We also set the maximum number of web transactions at 500. We use object delivery ratio (ODR) and user perceived delay (UPD) as the performance metrics while evaluating HTTP traffic over different routing protocols:

- **Object Delivery Ratio (ODR)** is the number of objects received divided by the number of objects sent from server to client
- **User Perceived Delay (UPD)** is the time from the object request issue to the time when the whole object has been received by the client, one measure of response time

The reason we choose ODR and UPD instead of PDR and packet delay is that for HTTP traffic, we do not take account of the delivery of a singe packet, instead, the percentage of objects delivered is much more important for the Web experience. The same also applies for UPD since users usually do not care about the delay for a single packet, instead, they will just notice the loading delay of the objects.

Figure 4 shows the variation of ODR with different routing protocols for an object size of 512 B. We can see that AeroRP performs better than all the other routing protocols with an average object delivery ratio of around 60% throughout the simulations. The ODR for the distance vector protocols DSDV and AODV drops as the number of nodes increase. DSDV and AODV exchange neighbor topology changes in the network and as the network density increases in this highly dynamic network, the routing information will not be up to date. DSR and OLSR have mediocre object delivery ratio with 50% and 30%.

Figure 5 shows the variation of UPD with a fixed object size of 512 B. The largest delay will be 10 s since we set the timeout value of the TCP connection as 10 seconds. We chose 10 s value after some test trials: with a timeout value of 1 s a majority of the objects were not delivered in this highly dynamic environment, whereas with a relatively high value of 30 s timeout all of the objects were delivered. We see that AeroRP performs better than all the other routing protocols with a delay of around 4 seconds. Using the non-transactional CBR traffic (Figure 3) the delay of AeroRP was seen to be 500 ms. Inclusion of transport layer protocol such as TCP in HTTP affects the increase in this delay. The delay values vary between 5 s and 10 s among the traditional MANET routing protocols. We also notice in our simulations that only AeroRP can finish the maximum number of transactions (set to 500) within the simulation time.

The object delivery ratio of AeroRP and traditional MANET routing protocols when using the 10240 B object size is shown in Figure 6. ODR decreases as the object size is increased from 512 B. This is because as the object size is increased, TCP requires more segments to be generated for a single object. The ODR for AeroRP reduces from 60% to 40% as the object size is increased 20-fold. The object delivery performance of the other MANET routing protocols also reduced. It should be noted that we also analysed object sizes of 1024, 5120, and 10240 B. The ODR relative performance characteristics between the routing protocols were similar, all degrading as the object size is increased.

User perceived delay performance of transactional traffic is shown in Figure 7. The delay for all the protocols follow the same trend from Figure 5 when the object size is 512 B, however only the absolute value of delay increases almost 2 seconds as the object size increases from 512 B to 10240 B. For example, the delay of AeroRP for 5 nodes increases from 4 to 6 seconds.

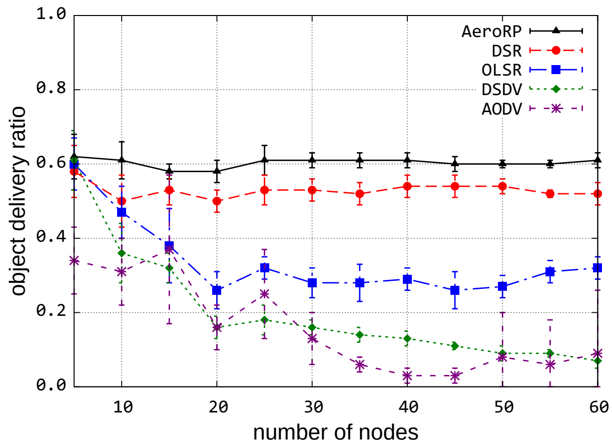

Figure 4: ODR with HTTP object size 512 B

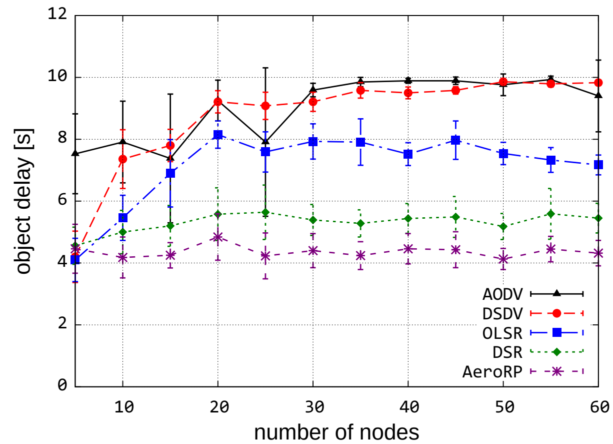

Figure 5: UPD with HTTP object size 512 B

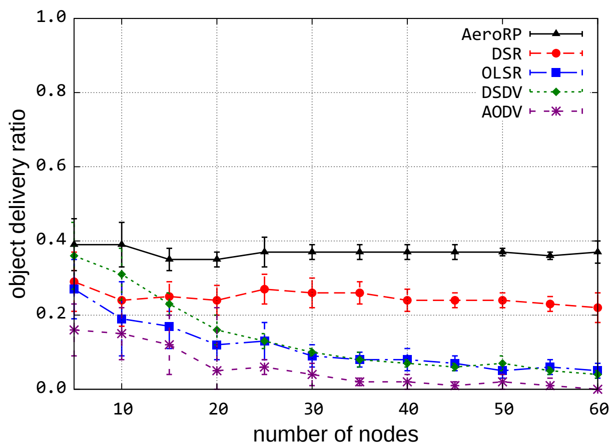

Figure 6: ODR with HTTP object size 10240 B

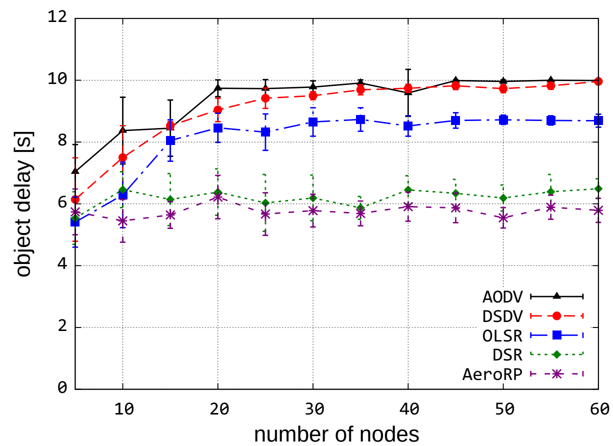

Figure 7: UPD with HTTP object size 10240 B

V. CONCLUSIONS AND FUTURE WORK

In this paper we present our implementation of the HTTP traffic generator for the ns-3 network simulator. We analyse HTTP performance over AeroRP in a highly-dynamic environment and compare its performance to traditional MANET routing protocols. Our *preliminary results* show that AeroRP outperforms other routing protocols in terms of both ODR and UPD. The involvement of transport layer protocol TCP in HTTP traffic also impacts the network performance since it requires three-way handshake, slow start, and connection timeout mechanisms.

We also implemented well-known MANET protocol DSR in ns-3 and show the preliminary performance of DSR in this highly dynamic environment. We show that using either transactional or non-transactional application traffic over the routing protocols affects the performance in a similar manner. On the other hand, increasing the object size of a HTTP response message degrades the network performance. Our results also confirms our previous studies that our domain-specific routing protocol AeroRP

outperforms the traditional MANET routing protocols. As part of future work, we plan to compare the performance of transactional traffic with CBR background traffic to pure transactional traffic. We will also simulate the AeroTP protocol with transactional traffic to remove the degradation due to TCP mechanisms.

ACKNOWLEDGEMENTS

The authors would like to thank the Test Resource Management Center (TRMC) Test and Evaluation/Science and Technology (T&E/S&T) Program for their support. This work was funded in part by the T&E/S&T Program through the Army PEO STRI Contracting Office, contract number W900KK-09-C-0019 for AeroNP and AeroTP: Aeronautical Network and Transport Protocols for iNET (ANTP). The Executing Agent and Program Manager work out of the AFFTC. This work was also funded in part by the International Foundation for Telemetering (IFT). We would like to acknowledge the ResiliNets group members Hemanth Narra and Justin P. Rohrer for useful discussions on this work. We would also like to thank Kip Temple and the membership of the iNET working group for discussions that led to this work.

REFERENCES

[1] J. P. Rohrer, A. Jabbar, E. K. Çetinkaya, E. Perrins, and J. P. Sterbenz, "Highly-Dynamic Cross-Layered Aeronautical Network Architecture," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, Oct. 2011.

[2] J. P. Rohrer, A. Jabbar, E. Perrins, and J. P. G. Sterbenz, "Cross-Layer Architectural Framework for Highly-Mobile Multihop Airborne Telemetry Networks," in *IEEE MILCOM*, (San Diego, CA), pp. 1–9, Nov. 2008.

[3] J. P. Rohrer, A. Jabbar, E. K. Çetinkaya, and J. P. Sterbenz, "Airborne Telemetry Networks: Challenges and Solutions in the ANTP Suite," in *IEEE MILCOM*, (San Jose, CA), pp. 74–79, Nov. 2010.

[4] A. Jabbar and J. P. Sterbenz, "AeroRP: A Geolocation Assisted Aeronautical Routing Protocol for Highly Dynamic Telemetry Environments," in *International Telemetering Conference*, (Las Vegas, NV), Oct. 2009.

[5] K. Peters, E. K. Çetinkaya, A. Jabbar, and J. P. G. Sterbenz, "Analysis of a Geolocation-Assisted Routing Protocol for Airborne Telemetry Networks," in *International Telemetering Conference*, (San Diego), Oct. 2010.

[6] K. Peters, A. Jabbar, E. K. Çetinkaya, and J. P. Sterbenz, "A Geographical Routing Protocol for Highly-Dynamic Aeronautical Networks," in *IEEE WCNC*, (Cancun, Mexico), pp. 492–497, March 2011.

[7] T. D. Dyer and R. V. Boppana, "On Routing Web and Multimedia Traffic in Mobile Ad Hoc Networks," in *36th Annual Hawaii International Conference on System Sciences (HICSS)*, 2003.

[8] T. D. Dyer and R. V. Boppana, "Routing HTTP traffic in a mobile ad hoc network," in *IEEE MILCOM*, vol. 2, pp. 958–963, 2002.

[9] "The ns-3 Network Simulator." `http://www.nsnam.org`, July 2009.

[10] E. Weingartner, H. vom Lehn, and K. Wehrle, "A Performance Comparison of Recent Network Simulators," in *IEEE International Conference on Communications (ICC)*, pp. 1–5, June 2009.

[11] H. Narra, Y. Cheng, E. K. Çetinkaya, J. P. Rohrer, and J. P. Sterbenz, "Destination-Sequenced Distance Vector (DSDV) Routing Protocol Implementation in ns-3," in *ICST SIMUTools Workshop on ns-3 (WNS3)*, (Barcelona, Spain), March 2011.

[12] 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.

[13] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile computing*, pp. 153–181, 1996.

[14] M. F. Arlitt and C. L. Williamson, "Web Server Workload Characterization: The Search for Invariants," in *ACM SIGMETRICS*, pp. 126–137, 1996.

[15] T. Berners-Lee, R. Fielding, and H. Frystyk, "Hypertext Transfer Protocol – HTTP/1.0." RFC 1945 (Informational), May 1996.

[16] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1." RFC 2616 (Draft Standard), June 1999. Updated by RFC 2817.

[17] F. D. Smith, F. H. Campos, K. Jeffay, and D. Ott, "What TCP/IP Protocol Headers Can Tell Us About the Web," in *ACM SIGMETRICS*, pp. 245–256, 2001.

[18] H. Choi and J. O. Limb, "A Behavioral Model of Web Traffic," in *IEEE ICNP*, pp. 327–334, 1999.

[19] J. Cao, W. Cleveland, Y. Gao, K. Jeffay, F. Smith, and M. Weigle, "Stochastic Models for Generating Synthetic HTTP Source Traffic," in *IEEE INFOCOM*, vol. 3, pp. 1546–1557, Mar. 2004.

[20] M. Weigle, "Improving Confidence in Network Simulations," in *Winter Simulation Conference*, pp. 2188–2194, Dec. 2006.

[21] F. Hernandez-Campos, K. Jeffay, and F. D. Smith, "Modeling and Generating TCP Application Workloads," in *BROADNETS*, pp. 280–289, 2007.

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

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

[24] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *ACM SIGCOMM*, pp. 234–244, 1994.

[25] "The network simulator: ns-2." `http://www.isi.edu/nsnam/ns/`, December 2007.

[26] H. Narra, E. K. Çetinkaya, and J. P. G. Sterbenz, "Performance Analysis of AeroRP with Ground Station Updates in Highly-Dynamic Airborne Telemetry Networks," in *International Telemetering Conference*, (Las Vegas, NV), Oct. 2011. to appear.

[27] D. Broyles, A. Jabbar, and J. P. G. Sterbenz, "Design and analysis of a 3–D Gauss-Markov Mobility Model for Highly-Dynamic Airborne Networks," in *International Telemetering Conference*, (San Diego, CA), Oct. 2010.