

Performance Analysis of AeroRP with Ground Station Updates in Highly-Dynamic Airborne Telemetry Networks

Hemanth Narra, Egemen K. Çetinkaya, and James P.G. Sterbenz

Department of Electrical Engineering & Computer Science

The University of Kansas

Lawrence, KS 66045

{hemanth, ekc, jpgs}@ittc.ku.edu

ABSTRACT

Highly dynamic airborne telemetry networks pose unique challenges for data transmission. Domain-specific multi-hop routing protocols are necessary to cope with these challenges and AeroRP is one such protocol. In this paper, we discuss the operation of various AeroRP modes and analyse their performance using the ns-3 network simulator. We compare the performance of beacon, beaconless, and ground station (GS) modes of AeroRP. The simulation results show the advantages of having a domain-specific routing protocol and also highlight the importance of ground station updates in discovering routes.

I. INTRODUCTION AND MOTIVATION

The present day telemetry architecture uses legacy point-to-point links connecting multiple sources (test articles) to a ground station. The increased usage of wireless devices to meet the emerging needs of Major Range and Test Facility Bases (MRTFB) led to increased requirements for bandwidth and connectivity. These legacy point-to-point links will not be able to cope with the limited spectrum. This need was recognised by various groups including the Integrated Network Enhanced Telemetry (iNET) group [1], [2]. Multihop routing protocols are necessary for the airborne test articles (TAs) to operate as an integrated system. TAs in these environments move at velocities of up to Mach 3.5 and move towards each other with relative velocities of about Mach 7.0. These high velocities lead to frequent link breaks and inconsistent routing of packets among nodes.

The Aeronautical Routing Protocol (AeroRP) is one such domain specific routing protocol that was designed for highly dynamic airborne networks [3]. AeroRP was first introduced in [4] and later modelled and analysed using the ns-3 network simulator [5, 6]. The preliminary results showed that AeroRP outperforms the traditional MANET routing protocols in terms of throughput and packet delivery ratio (PDR) [5, 6]. AeroRP uses the node's current or predicted geolocation¹ information for discovering routes. Neighbour discovery in AeroRP is multi-modal, in which various modes can be used to discover neighbours depending upon the mission needs including stealth requirements. One of those neighbour discovery modes is GS (ground station) update mode. In this paper we present the design and implementation of GS update mode of AeroRP in ns-3 network simulator [7]. We compare the results of different

¹The words location and position can be used interchangeably and mean the same in the context of this paper.

modes of AeroRP neighbor discovery modes and show that priori knowledge of the network topology in this highly-dynamic environment significantly improves the performance of the routing.

The rest of the paper is organised as follows. In Section II we present an overview of MANET routing protocols, AeroRP, and AeroNP. The AeroRP design with GS updates is explained in Section III followed by implementation discussion of AeroRP in Section IV. Gateway functionality implementation in ns-3 is presented in Section V. The performance of AeroRP is analysed in Section VI. Finally, we present the conclusions and future work in Section VII.

II. BACKGROUND AND RELATED WORK

AeroRP is a geographic routing protocol designed for highly dynamic airborne telemetry networks [4]. Contrary to the other MANET routing protocols that discover end-to-end paths, AeroRP makes only per-hop routing decisions. This is reasonable as the nodes in the telemetry network move at very high velocities often leading to breakage of links after an end-to-end path has been determined. AeroRP can operate in various modes based on the mission requirements [3].

- **TA update mechanism:** AeroRP can operate in **beacon** and **beaconless** modes. In **beacon** mode, the test article advertises its location information by broadcasting periodic **hello** messages, whereas in **beaconless** mode no messages are sent out.
- **Location information:** AeroNP may or may not include the geolocation information of TAs in their headers. This can be due to mission requirements. For example in a given mission, TAs can be programmed to be in stealth mode so that they are not making their location available to other TAs. We will explain more about location information in Section II-B.

A. Operational Aspects of AeroRP

The operation of AeroRP can be divided into two phases. The first phase of operation is the *neighbour discovery* phase. In this phase, a TA gathers as much information as it can about the network topology in the following ways:

- **Active snooping:** Active snooping is a mechanism in which the nodes snoop packets that are being exchanged among other nodes, extract the location information from them and build or update their topology tables. To accomplish this, active-probing on the node's network interface must be enabled. Location information thus gathered is only valid for a time interval specified by `neighborHoldTime`. On expiration of this time-interval, the stored location information of a node is purged unless a new update with a higher expire time is received. This helps in keeping track of all active neighbours in this highly dynamic environment.
- **Hello beacons:** Hello beacons are transmitted by the TA when it is not transmitting any data. This ensures that its neighbouring TAs are aware of its presence. These messages are usually broadcasted periodically over `helloUpdateInterval` with *time-to-live* (TTL) set to one hop.
- **Ground station updates:** These are optional updates transmitted by the ground station during some missions having a pre-determined flight plan. These updates are broadcasted periodically and are exchanged among all the TAs in the network.

The AeroRP modes explained earlier affect the various neighbour discovery processes. In beaconless mode the *hello* messages are not sent by any of the test articles. Therefore, neighbour discovery relies on *overhearing* the packets in the medium. Depending on the mission needs, AeroRP messages can have location information of TAs. If the AeroRP routing messages include location information, then the TAs can determine network topology information. Otherwise, without the location information, TAs can only be aware of their neighbours. In GS update mode of neighbour discovery, the ground station broadcasts GSLink or GSTopology packets. GSLink packets carry only the active or predicted links among all the nodes and GSTopology packets carry the geolocation information of the nodes.

The second phase of AeroRP operation is *data forwarding*. In this phase, the sender node determines the next hop to forward a packet by using the topology table built in the *neighbour-discovery* phase. The *Time-to-intercept* (TTI) metric is used in determining this next hop neighbour [5, 6, 8]. TTI is calculated for every node from the topology table as:

$$\text{TTI} = \frac{\Delta d - R}{s_d}$$

where Δd is the euclidean distance between the current location and a predicted location of a node based on the recorded location coordinates and velocity components from the table, R is the common transmission range of all the nodes, and s_d is the recorded speed component. The assumption made here is that the nodes move at a constant speed during the interval for which we calculate the TTI value. TTI gives the estimate of time taken by the neighbours to reach within the transmission range of the destination. The neighbour with the lowest TTI value is chosen as the next hop neighbour and packets are forwarded to this neighbour.

B. AeroNP

AeroNP is an IP-compatible network protocol specifically designed for highly-dynamic telemetry networks. Telemetry systems and other devices on TAs are IP -based. In addition to replicating the services provided by IP, AeroNP provides QoS (Quality of Service), flow control, and error detection to the AeroTP transport layer [9]. QoS is provided by tagging packets based on priorities; AeroNP provides four levels of priorities. The AeroRP control packets are always given the highest priority. Mission specific command and control data will be given higher priority compared to telemetry data. The packets tagged with a particular priority are queued in specific buffer classes. The flow-control mechanism is accomplished by implementing a cross-layering mechanism with the iNET TDMA MAC layer. The congestion indicator (CI) field in AeroNP specifies the congestion level at a node. If a node identifies that the MAC buffer is full, it increases its congestion indicator value in the AeroNP header. Neighbouring nodes do not forward packets to a node with high CI value, unless it is the destination. The wireless medium is error-prone leading to packet corruption, and detecting these errors at the destination and waiting for the source to re-send the packet increases the end-to-end delay. The AeroNP corruption indicator and HEC-CRC (header error check – cyclic redundancy code) fields provide the error detection. As mentioned in Section II-A, depending on the mission requirements geolocation information can be included in the AeroNP header. We designate the AeroNP header with the geolocation information as the *extended header*, whereas the AeroNP header without the geolocation information as is referred to as the *basic header*. The AeroNP *basic* and *extended* headers are shown in Figure 1 and 2, respectively.

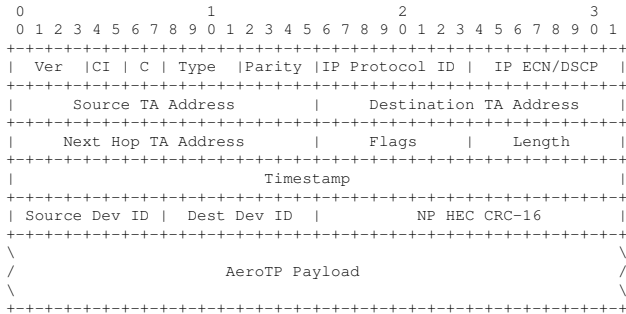


Figure 1: AeroNP basic header

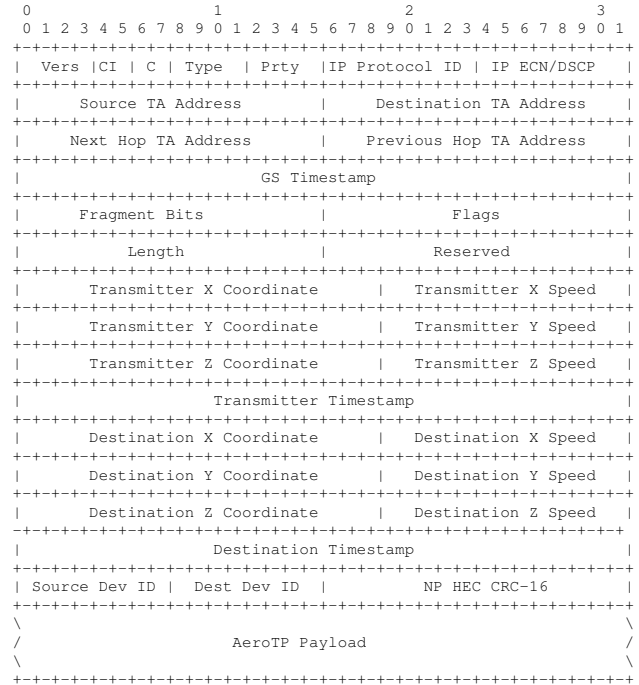


Figure 2: AeroNP extended header

III. DESIGN OF AeroRP WITH GROUND STATION UPDATES

Introducing ground station updates to the AeroRP routing protocol adds an additional mechanism for neighbour discovery in AeroRP. The ground station tracks movements of all the TAs and broadcasts either their TopologyInformation or LinkInformation to other TAs in the network.

C. GS Update Mechanism

A ground station broadcasts updates for all the TAs periodically over a time-interval called the `periodicUpdateInterval`. Depending on velocities of the TAs and the frequency at which they change direction, the `periodicUpdateInterval` is set. The frequency at which the GS sends these updates will affect the protocol’s performance significantly. If the `periodicUpdateInterval` is low, the GS broadcasts updates frequently resulting in increased control overhead. On the contrary, if the `periodicUpdateInterval` is high, the GS will broadcast updates rarely resulting in the TAs not having up-to-date information about the other TAs. Velocities of all the TAs are unlikely to be uniform and some of the TAs may change their direction more frequently than others. Therefore, sending updates of all the TAs for every `periodicUpdateInterval` alone is not sufficient. There is a need for a mechanism in which the ground station can broadcast updates in-between the `periodicUpdateIntervals` as well, called the trigger update mechanism. Whenever the GS identifies a change in direction of any TA, it will immediately broadcast the changed location and velocity information of the TA. Highly dynamic changes result in the GS sending trigger updates more frequently which leads to more control overhead and increased packet collisions in the network. To avoid this, trigger updates over a `triggerUpdateInterval` are aggregated into a single

update and sent together. On the other hand, the GS verifies if a trigger update can be cancelled and the change be sent in the next periodic update. If the time duration to the next periodic update is less than the triggerUpdateInterval, then the trigger update is not sent.

To make sure the GS advertisements do not violate the maximum transmission unit (MTU) of 1500 B set by the MAC layer, they are fragmented to multiple packets. Each packet is uniquely identified by the time at which this packet was originated and by a 16-bit fragment number.

D. Types of GS Updates

The ground station sends two types of advertisements: GStopology and GSLink. The GStopology and GSLink advertisements are explained in the following sub sections.

D.1 Topology Advertisements

Topology information of all the nodes is advertised by ground station using GStopology advertisements. This advertisement carries a TypeHeader over multiple GStopologyHeaders containing the geolocation coordinates and the velocity components in x , y , and z directions. A ground station also maintains a topology table just like other TAs. The difference is that the TAs fill their table on receiving AeroRP updates, whereas the GS topology table is assumed to have updated information of the entire network at all times. The ground station uses this information to send out topology updates. It sets the EnableBroadcast flag to *true* so that this message can be rebroadcasted among all the TAs. Depending on the update mechanism used, the GS sends out updates for all the TAs or only for the TAs with changed information since the last update. The GS creates multiple topology headers, one for each node and adds them to a packet. It then creates a single TypeHeader, populates the necessary fields and adds it on top of the topology headers. This packet is then broadcasted in the network. Every TA receiving the packet will process the TypeHeader, identify the type of headers present in the packet and based on the header length it will process each header separately. Figures 3 and 4 shows the header formats for TypeHeader and GStopologyHeader, respectively.

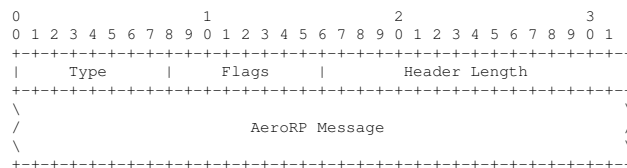


Figure 3: AeroRP TypeHeader

Topology updates have an option for adding start time and end time as well. These fields are used when a TA is flying in a pre-determined path. When the start time is populated, it is interpreted as the TA is located at the location coordinates specified by the topology header. The end time field is the time from which the TAs should stop using this location information. This field can also be set to next periodicUpdateInterval or the predicted time after which the TA might go out of the network.

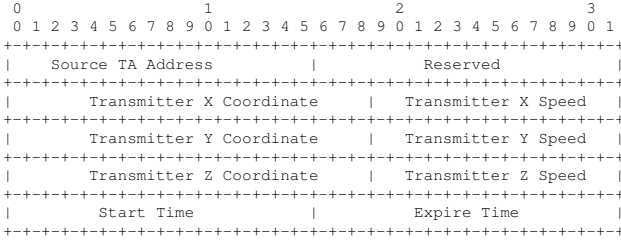


Figure 4: AeroRP GSTopologyHeader

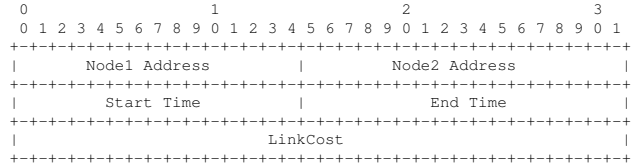


Figure 5: AeroRP GSLinkHeader

D.2 Link Advertisements

GSLink advertisements carry multiple GSLinkHeaders for all the links formed among nodes in the network. Figure 5 shows the header format for GSLinkHeader. Each GSLinkHeader will carry two 16-bit node-id addresses of the TAs forming the link, the start and the expire times for that link, and the link cost. The start and expire times are calculated based on node’s geolocation and velocity information. A link is said to be established between two nodes if the euclidean distance between the two is less than their transmission range. The assumption here is that all nodes have the same transmission range. Based on the nodes geolocation coordinates the euclidean distance is calculated. The link expire time is predicted based on the geolocation and velocity components. The expire time is increased until the euclidean distance between the new predicted locations of the two nodes is greater than their transmission ranges. Thus the GS calculates this information for all the nodes in the network. If there are n nodes in a network, considering the best case scenario in which every node is connected to every other node, the total number of links are $n \times (n - 1)/2$.

IV. PROCESSING OF AeroRP UPDATES

AeroRP uses a protocol id of 251 and works in conjunction with the AeroNP network protocol. The AeroNP protocol on receiving any packet from the MAC layer will identify an AeroRP packet by looking at the protocol id field in the AeroNPHeader, and if appropriate will deliver it to the AeroRP routing protocol along with the extracted AeroNPHeader. A GS node does not require any AeroRP control information as it always keeps track of all the nodes in the network. Each TA examines the sourceTimestamp field present in the AeroNPHeader. It compares it with the stored, last-received timestamp value for that neighbouring node in its topology table. If the received timestamp is newer, the protocol will update the node’s topology table with the geolocation information present in the AeroNPHeader. It also updates the congestion and corruption indicators for the node from which this packet was received. AeroRP then compares its own GSTimestamp value with that of the neighbouring node’s GSTimestamp value. If it identifies that the neighbour node does not have a newer update from the GS, it unicasts those updates to the neighbour node. This mechanism will make sure that every node will be as consistent as possible.

As mentioned in Section III, AeroRP uses three types of control messages to broadcast its updates. They are hello messages, GSLink advertisements, and GSTopology advertisements. A hello message does not have any information to be used by the AeorRP routing protocol. It is only used to inform a node’s neighbours of its presence. Upon receiving a GSTopology advertisement, AeroRP unpacks

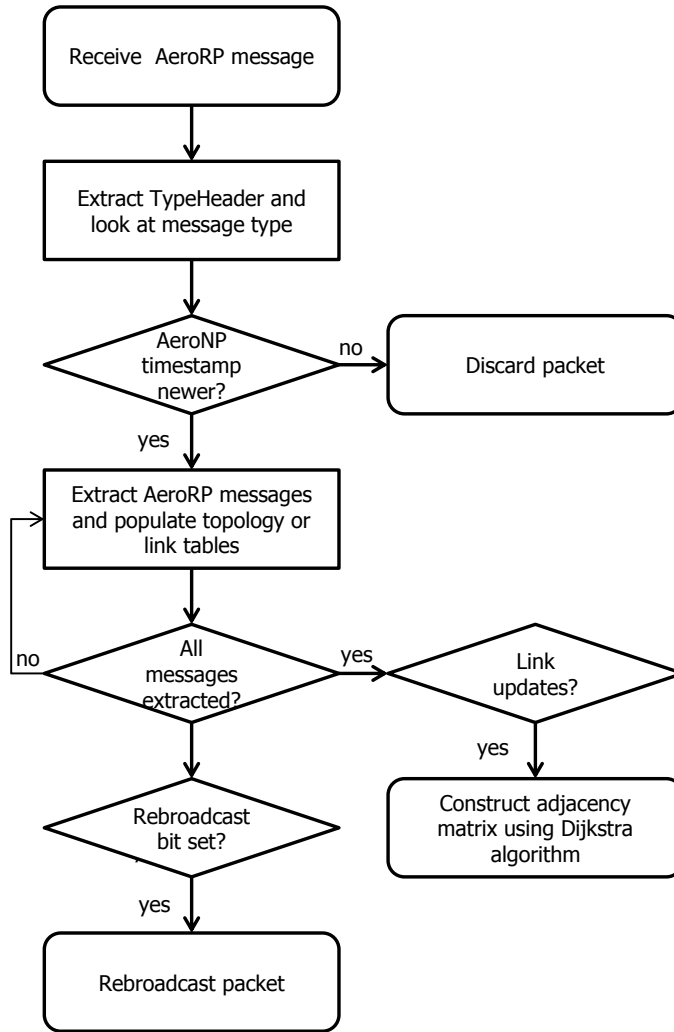


Figure 6: Flowchart for processing a GS advertisement

all the `GSTopologyHeaders` one by one and updates the topology table entry for that node. `GSLink` advertisements will also be processed the same way as a `GSTopology` advertisement by extracting the `GSLinkHeaders`. However, after all the `GSLinkHeaders` are updated in the topology table, the protocol will determine if the Dijkstra shortest-path algorithm should be run based on the changed link costs. If there is a change in the links, AeroRP runs the Dijkstra algorithm, otherwise the algorithm is scheduled to run at the time determined by the creation of a new link based on the predicted geocoordinates and velocity components. A timer schedules the Dijkstra algorithm by keeping track of the new link formations or breakages based on the information from the topology table. Figure 6 is the flowchart showing the GS update process.

V. DESIGN AND ns-3 IMPLEMENTATION OF AeroGW

Telemetry data is expected to originate from a system supporting TCP/UDP/IP. This data should be moved over the domain-specific ANTP protocol suite and handed over to the destination which is again

expected to be TCP/UDP/IP-based. All the current telemetry applications and devices are IP-based which arises compatibility issues of the new protocol suite. To overcome this, we have introduced an interface called the Aero Gateway (AeroGW) [10] that resides on every node in the telemetry network including the GS. Telemetry data originating from, or destined to these applications will be processed by the AeroGW to convert to the ANTP protocol suite. The AeroGW will translate the IP header to the AeroNP header and the TCP or UDP header to the AeroTP header. The original TCP/UDP/IP headers will be removed from the packet and replaced with the newly generated AeroTP/AeroNP headers.

The AeroGW simulation in ns-3 was implemented with only the functionality of translating IP addresses used by the GS and the TAs to 16-bit device ids used in the ANTP protocol suite. As the ANTP protocol suite is being simulated in ns-3 that creates all sockets by binding them to IP addresses, it is necessary for the protocols being simulated in ns-3 to use IP addresses. To correctly evaluate the performance of the Aero protocols, the 32-bit IP addresses are mapped to a 16-bit device ids in these AeroGWs.

VI. SIMULATION RESULTS

We performed the simulations over an area of $150 \times 150 \text{ km}^2$. All the simulations were averaged over 10 runs with each simulation running for 1000 s and plotted with 95% confidence interval bars. Simulations were performed with varying node densities ranging from 5 to 60 nodes. The communication model is peer-to-peer with as many flows as the number of nodes in the network. All the TAs are configured to send 1 packet/s. Using this lower packet rate, we can correctly evaluate the performance of the protocol by removing any possibility of congestion at MAC layer. We used the ns-3.10 to perform these simulations. The On-Off application in ns-3 is used to generate CBR (constant-bit rate) traffic. The 802.11b MAC is used over the Friis propagation loss model to limit the transmission ranges of nodes. The transmit power was set to 50 dBm to achieve a 27800 m (15 nautical mi) transmission range. The mobility model used is 3D Gauss-Markov [11] with node velocities ranging from 50 – 1200 m/s. Setting α between zero and one allows us to tune the model with degrees of memory and variation. We set the α to 0.85 to have some predictability in the mobility of the nodes, while avoiding abrupt TA direction changes. Ferrying of packets is enabled by default in AeroRP using a drop-tail queue implementation. The maximum buffer size is set to 400 and the maximum buffering time is set to 30 s. Any packet that reaches its buffer expiry time will be purged. Active snooping on all interfaces is enabled by default.

E. Performance Metrics

The performance metrics for the evaluation of AeroRP are packet delivery ratio (PDR), routing overhead, and delay.

- **PDR:** The number of packets received divided by the number of packets sent by the application
- **Routing overhead:** The fraction of bytes used by the protocol for AeroRP control messages
- **Delay:** The time taken by the packet to reach the destination node's MAC protocol from the source node's MAC protocol

F. Simulation Analysis

We compare the AeroRP performance when the AeroNP header includes geolocation information of the nodes (topology information) or no location information is present in the AeroNP header (link information). Figure 7 shows the performance of different modes of AeroRP in terms of PDR as the node density increases from 5 to 60 nodes, moving at a velocity of 1200 m/s. The GS update mode with LinkInformation performed better until the node density reached 35 nodes and later it started to degrade as the node density further increased. The increase in node density led to more link breakages. This may have resulted in breakages in the end-to-end paths determined by the node. Though the GS tries to update the changes in link information by sending out trigger updates, there is a chance that some nodes might not receive all the updates leading to inconsistent routing of packets. As the node-density increased, the AeroRP update mode without GS updates and using geolocation information in the TA update mode began to perform comparatively better as the nodes in this mode exchanged geolocation information in the AeroNP headers.

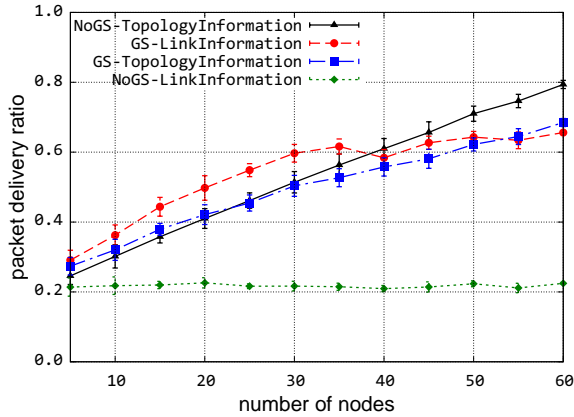


Figure 7: Effect of node density on PDR

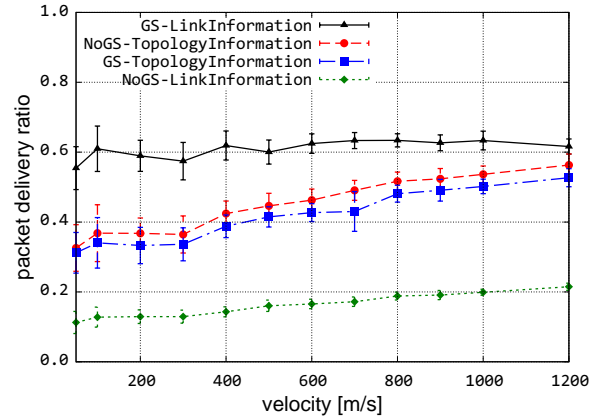


Figure 8: Effect of node velocity on PDR

Figure 8 shows the variation of PDR as the velocity increases from 50 m/s to 1200 m/s for 35 nodes. At 35 nodes the average network connectivity is 75%. The average network connectivity is determined by using the geolocation information of nodes and their transmission ranges. Average network connectivity of 75% is chosen to ensure we can analyse the PDR performance in a not fully connected network. We see that AeroRP in presence of GS and using LinkInformation performed better. This is because with the 3D Gauss-Markov mobility model TAs move without any abrupt change in their direction. Thus the links predicted by the GS remains same most of the time. However, as the velocities increase there are many link breakages leading to the GS sending an increased number of updates. As the TAs run the Dijkstra shortest-path algorithm they need to have consistent information from the GS, otherwise this may lead to inconsistent routing of packets. The other modes of AeroRP using TopologyInformation, with and without the presence of GS updates, perform better as the velocities increase. This is because at high velocities the nodes have more chance of identifying a path to the destination as they come across different neighbours which increases their chance to identify a route to the destination. Furthermore, they are aware of the entire topology of the network as they exchange geolocation information in the AeroNP headers. AeroRP mode using LinkInformation in the absence of GS updates does not perform very well as it knows only its next hop neighbours.

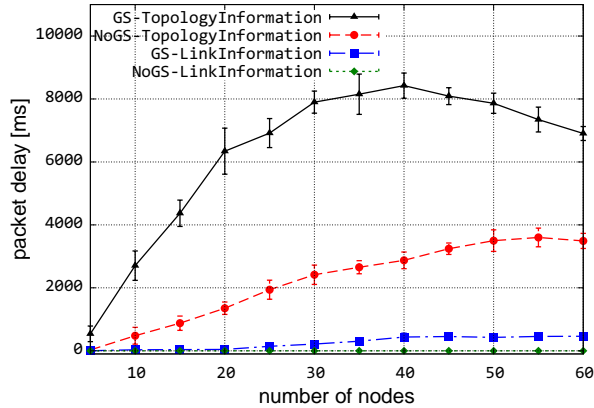


Figure 9: Effect of node density on end-to-end delay

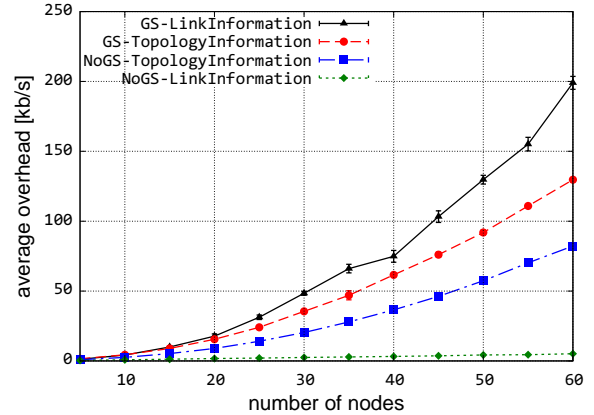


Figure 10: Effect of node density on control overhead

Figure 9 shows the end-to-end packet delay with varying node densities. We see that the end-to-end packet delay is greater while using `TopologyInformation` with and without the presence of GS updates. This is because the calculation of TTI is based on the node’s current and predicted geolocation coordinates. By using `TopologyInformation` from GS updates, we can predict the node’s trajectory information for a longer duration as the expire time provided by the GS is usually greater compared to the `neighbourholdTime`. This results in having higher TTI values while using `TopologyInformation` from the GS updates. TTI values specify how long a packet should be buffered before identifying a route to the destination. The other two modes using `LinkInformation` have very low delay as they do not use the TTI metric to perform routing decisions.

The variation of AeroRP control overhead with node-density in its various modes of operation is analysed in Figure 10. The control overhead in the absence of GS updates and using `LinkInformation` is significantly less as there are no AeroRP messages sent out except for the initial bootstrap beacons. Control overhead however, increases exponentially with the increase of node density for AeroRP mode with GS updates enabled and sending out `LinkInformation`. This is because as the node density increases, so do the number of links between those nodes leading more updates from the GS whenever a new link is formed or an existing link is broken.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we presented the various modes of AeroRP concentrating on the ground station update mode using both `TopologyInformation` and `LinkInformation`. We presented a detailed analysis of these different modes by modelling them in ns-3 and analysing their performance under various operations scenarios. We showed that location information in the AeroNP header increases packet delivery ratio as the network density increases. Our results also indicate that using ground station update mode increases the overhead.

As part of the future work, we plan to analyse AeroRP performance using TDMA MAC model and compare its performance benefits over 802.11 MAC. We are also planning to analyse AeroRP performance with the AeroTP protocol [9, 12]. We are also starting to implement these protocols on prototypes [13].

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 Telemetry (IFT). We would like to acknowledge the ResiliNets group members Abdul Jabbar, Justin P. Rohrer, and Kevin Peters 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 Working Group, “<http://www.inetprogram.org>.”
- [2] “iNET Needs Discernment Report, version 1.0.” Central Test and Evaluation Investment Program, May 2004.
- [3] 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, October 2011.
- [4] A. Jabbar and J. P. G. Sterbenz, “AeroRP: A Geolocation Assisted Aeronautical Routing Protocol for Highly Dynamic Telemetry Environments,” in *International Telemetry Conference*, (Las Vegas, NV), October 2009.
- [5] 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.
- [6] K. Peters, E. K. Çetinkaya, and J. P. G. Sterbenz, “Analysis of a Geolocation-Assisted Routing Protocol for Airborne Telemetry Networks,” in *International Telemetry Conference*, (San Diego, CA), Oct. 2010.
- [7] “The ns-3 network simulator.” <http://www.nsnam.org>, July 2009.
- [8] K. Peters, “Design and Performance Analysis of a Geographic Routing Protocol for Highly Dynamic MANETs,” Master’s thesis, The University of Kansas, Lawrence, KS, June 2010.
- [9] 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 *International Telemetry Conference*, (San Diego), Oct. 2008.
- [10] E. K. Çetinkaya and J. P. G. Sterbenz, “Aeronautical Gateways: Supporting TCP/IP-based Devices and Applications over Modern Telemetry Networks,” in *International Telemetry Conference*, (Las Vegas, NV), Oct. 2009.
- [11] 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 Telemetry Conference*, (San Diego, CA), Oct. 2010.
- [12] K. S. Pathapati, T. A. N. Nguyen, J. P. Rohrer, and J. P. G. Sterbenz, “Performance Analysis of the AeroTP Transport Protocol for Highly-Dynamic Airborne Telemetry Networks,” in *International Telemetry Conference*, (Las Vegas, NV), Oct. 2011. to appear.
- [13] M. Alenazi, S. A. Gogi, D. Zhang, E. K. Çetinkaya, J. P. Rohrer, and J. P. G. Sterbenz, “ANTP Protocol Suite Software Implementation Architecture in Python,” in *International Telemetry Conference*, (Las Vegas, NV), Oct. 2011. to appear.