AeroRP Performance in Highly-Dynamic Airborne Networks using 3D Gauss-Markov Mobility Model

Justin P. Rohrer, Egemen K. Çetinkaya, Hemanth Narra, Dan Broyles, Kevin Peters, and James P.G. Sterbenz
Department of Electrical Engineering & Computer Science
Information and Telecommunication Technology Center
The University of Kansas
Lawrence, KS 66045
{rohrej|ekc|hemanth|dbroyl01|kevjay|jpgs}@ittc.ku.edu
www.ittc.ku.edu/resilinets

Abstract—**Emerging airborne networks require domain-specific routing protocols to cope with the challenges faced by the highly-dynamic aeronautical environment. We present an ns-3 based performance comparison of the AeroRP protocol with conventional MANET routing protocols. To simulate a highly-dynamic airborne network, accurate mobility models are needed for the physical movement of nodes. The fundamental problem with many synthetic mobility models is their random, memoryless behavior. Airborne ad hoc networks require a flexible memory-based 3-dimensional mobility model. Therefore, we have implemented a 3-dimensional Gauss-Markov mobility model in ns-3 that appears to be more realistic than memoryless models such as random waypoint and random walk. Using this model, we are able to simulate the airborne networking environment with greater realism than was previously possible and show that AeroRP has several advantages over other MANET routing protocols.**

I. Introduction and Motivation

Airborne communication presents a challenging environment for mobile ad hoc networking. High mobility, limited bandwidth and transmission range, and unreliable noisy channels create a harsh environment for communications [1]–[3]. The problems of congestion, collisions, and transmission delays are only made worse in an ad hoc multi-hop environment [4]. Additionally, we cannot assume that any two nodes will be within transmission range of each other for very long. Two highly-mobile nodes moving in opposite directions at supersonic to hypersonic relative velocity might only expect to have a few seconds of opportunity to discover, setup, and transfer data, or to make a successful handoff.

New routing protocols such as AeroRP [5]–[8] are emerging to address these difficulties and there is a need to evaluate their performance, initially, through realistic simulations. The ns-3 simulator is an emerging network performance simulation environment with several mobility models [9] already built in, including random direction 2D, random walk 2D, random waypoint [10], constant velocity, constant acceleration, and constant position [11]. The fundamental problem with many synthetic mobility models is their random, memoryless behavior. Simulations using these mobility models exhibit unnatural movements with abrupt and often extreme changes in velocity and direction, uncharacteristic of highly-mobile airborne nodes. These mobility models are insufficient in simulating a highly-dynamic airborne ad hoc network. They also lack support for 3-dimensional position allocation, relative velocity between nodes in 3D space, and realistic flight behavior. To address these concerns we have implemented an ns-3 model of a 3-dimensional Gauss-Markov mobility model suitable for use in multi-tier, highly-dynamic MANET simulations [12]. We use this model to provide realistic mobility patterns for evaluating the AeroRP protocol, and compare with the effects of using memoryless mobility models.

AeroRP is a domain specific geolocation-assisted routing protocol. The main focus of AeroRP is to efficiently route data packets amongst airborne nodes (ANs) and back to a ground station (GS). It supports the incorporation of specialized relay-nodes (RNs), which have enhanced communication capabilities to facilitate the routing of data by the ANs. Each AN may use other ANs or RNs as next hops in order for the packets to reach their destination as a given AN may not be within transmission range of the GS or target AN within a reasonable amount of time. The target environment for AeroRP is one in which ANs may achieve relative speeds up to Mach 7 for aircraft, as well as Mach 20 for hypersonic vehicles. These fast moving nodes create a unique challenge for routing packets when connectivity amongst the nodes is very intermittent and episodic, which prevents traditional MANET routing protocols from providing acceptable performance.

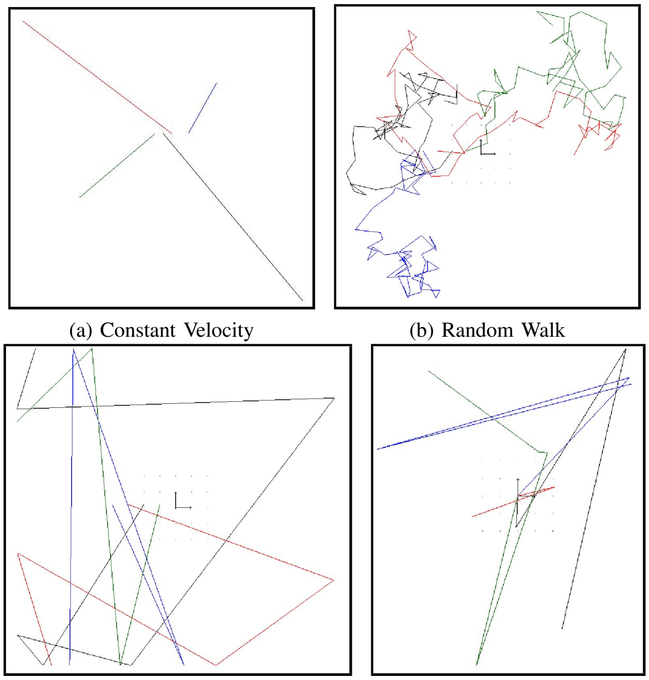
The rest of the paper is organized as follows: Section II presents the other mobility models and routing algorithms related to this research. This is followed a discussion of the requirements for a suitable airborne mobility model in Section III. We present our new 3D Gauss-Markov mobility model algorithm in Section IV. We present the algorithms used in the AeroRP model in Sections V and VI. Section VII contains simulation results comparing these two models with common alternatives. Finally we conclude in section VIII.

II. Background and Related Work

Previous to this work, the ns-2 and ns-3 simulator built-in mobility models included constant velocity, random walk 2D, random direction 2D, and random waypoint. We now present a brief summary of these models.

978-1-4673-0081-0/11/$26.00 ©2011 IEEE

In constant velocity, nodes proceed along their initial velocity vector for the duration of the simulation as shown in Figure 1a. There are no geographic boundaries in this model. In random walk 2D, each node is given a random trajectory (speed and direction) and travels on that trajectory for a fixed period of time or a fixed distance as shown in Figure 1b. When nodes reach the limits of the 2-dimensional boundary, they bounce off in a new direction mirroring the previous direction and velocity. Figure 1c shows an example of the random direction model, in which nodes travel on a random trajectory until they reach the 2D boundary, at which time they pause for a random period of time and head off in a new random direction and speed. In the random waypoint model, as illustrated in Figure 1d, each node travels to a random waypoint (x,y coordinate), pauses for a period of time, and then heads off to another waypoint. The waypoints, node speeds, and pause times are modeled as uniformly distributed random variables.



(a) Constant Velocity

(b) Random Walk

(c) Random Direction

(d) Random Waypoint

Fig. 1: ns-3 memoryless mobility models

Intuitively we can see that these simple synthetic mobility models do not mimic the motion of airborne nodes very well. The nodes undergo sudden changes in speed and direction at random, which is uncharacteristic of real aircraft. Later in this paper we will show that these characteristics have a noticeable effect on routing performance.

There are a number of existing multihop, geographic routing strategies with similarities to AeroRP, so we will briefly summarize them next. Several geographic routing survey papers [13]–[15] break down different geographic forwarding decisions into MFR (most forward with radius r), NFP (nearest with forward progress), and compass. MFR is the most intuitive and forwards the packet to the node that makes the most forward progress between the source and destination. NFP forwards the packet that is closest to the current node and is closer to the destination, reducing packet collisions compared to MFR by making shorter hop routing decisions. Compass forwarding chooses a node that is closest to an imaginary line drawn between itself and the destination based on the trajectory. There are many geographic routing protocols, including DREAM [16], LAR [17], GPSR [18], and SiFT [19].

Unpiloted aerial vehicle geographic routing has been simulated at 25 m/s [20], much slower than Mach 3.5 (1200 m/s). A top speed of 20-50 m/s is typical amongst geographic routing protocols [21]–[27]. There are a few routing protocols specifically for aeronautical environments. ARPAM [28] is a hybrid AODV [29] protocol for commercial aviation networks that utilizes the geographic locations to discover the shortest but complete end-to-end path between source and destination. Multipath Doplar routing (MUDOR) [30] takes relative velocity into consideration as well as the Doppler shift to measure the quality of a link. Anticipatory routing [31] tracks highly mobile endpoints that reach the reactive limit in which the speed of the nodes is comparable to the time it takes for the location tracking to converge upon the position of the node. Spray routing [32] involves unicasting a packet a specific *depth* away from the destination in which the packet is then *sprayed* or multicasted to a controlled *width* or number of levels of neighbors, for highly mobile endpoints up to 250 m/s. However, none of these approaches mention such speeds within an order of magnitude of Mach 3.5, in which rapidly varying connectivity becomes a much greater consideration.

III. Airborne Mobility Model Requirements

Some of the challenges faced by airborne networks include high mobility, limited bandwidth, limited transmission range, and intermittent connectivity [33]. Airborne nodes are highly dynamic and require 3-dimensional models. The current ns-3 mobility models are designed for 2-dimensional movement. The random waypoint model, the basic design of which is not limited to 2 dimensions, can only select from waypoints generated from the ns-3 position allocation class. And at this time, there are only three position allocation models identified in the position allocation class: grid position, random rectangle, and random disc, all of which are 2-dimensional allocation schemes.

Simulated airborne nodes represent the motion of physical objects flying through the air, for which natural laws must be obeyed. This implies that the path of an airborne node will not be completely random, but its position at any point in time will be dictated largely by its previous position and velocity vector. Therefore, the mobility model must have *memory*. The mobility models mentioned previously are all memoryless. One characteristic of a memoryless mobility model is the existence of very sharp and sudden changes in direction and speed. One measure by which a good airborne mobility model should be judged is the fluidity of movement between consecutive positions.

IV. Gauss-Markov Mobility Algorithm

In this section, we first present the basic Gauss-Markov algorithm for modeling two-dimensional mobility. We then extend the model to three dimensions as well as introduce new parameters to accurately model the mobility of airborne nodes.

A. The Basic 2D Gauss-Markov Algorithm

The Gauss-Markov mobility model introduced in [34] is a relatively simple memory-based model with a single tuning parameter, alpha α, which determines the amount of memory and variability in node movement. In this paper, we describe other tuning parameters that have a significant impact on the dynamics and characteristics of the Gauss-Markov mobility model as well as the selection of alpha.

In the traditional 2-dimensional implementation of the Gauss-Markov model, each mobile node is assigned an initial speed and direction, as well as an average speed and direction. At set intervals of time, a new speed and direction are calculated for each node, which follow the new course until the next time step. This cycle repeats through the duration of the simulation. The new speed and direction parameters are calculated as follows [9]:

$$\begin{aligned} s_n &= \alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)}s_{x_{n-1}} \\ \theta_n &= \alpha \theta_{n-1} + (1-\alpha)\bar{\theta} + \sqrt{(1-\alpha^2)}\theta_{x_{n-1}} \end{aligned} \tag{1}$$

where α is the tuning parameter, $\bar{s}$ and $\bar{\theta}$ are the mean speed and direction parameters, respectively, and $s_{x_{n-1}}$ and $\theta_{x_{n-1}}$ are random variables from a Gaussian (normal) distribution that give some randomness to the new speed and direction parameters.

Special Case: $\alpha = 0$
When α is zero, the model becomes memoryless; the new speed and direction are based completely upon the average speed and direction variables and the Gaussian random variables.

$$\begin{aligned} s_n &= \bar{s} + s_{x_{n-1}} \\ \theta_n &= \bar{\theta} + \theta_{x_{n-1}} \end{aligned} \tag{2}$$

Special Case: $\alpha = 1$
When α is 1, movement becomes predictable, losing all randomness. The new direction and speed values are identical to the previous direction and speed values. In short, the node continues in a straight line.

$$\begin{aligned} s_n &= s_{n-1} \\ \theta_n &= \theta_{n-1} \end{aligned} \tag{3}$$

Setting α between zero and one allows for varying degrees of randomness and memory. In addition to α, the dynamics of the Gauss-Markov mobility model are greatly influenced by other variables like the time step, the selection of the average speed and direction, and the mean and standard deviation chosen for the Gaussian random variables. For example, choosing a standard deviation on the Gaussian distribution governing the direction that is much larger than the average direction generates a very different movement pattern than if the standard deviation and average direction values are similar.

B. Extending the Model to Three Dimensions

In this section, we discuss several methods to extend the basic Gauss-Markov 2-dimensional model to three dimensions. The first approach is to apply the Markov process to the x, y, and z axis of a 3-dimensional velocity vector. The velocity vector is computed as:

$$\begin{aligned} x_n &= \alpha x_{n-1} + (1-\alpha)\bar{x} + \sqrt{(1-\alpha^2)}x_{x_{n-1}} \\ y_n &= \alpha y_{n-1} + (1-\alpha)\bar{y} + \sqrt{(1-\alpha^2)}y_{x_{n-1}} \\ z_n &= \alpha z_{n-1} + (1-\alpha)\bar{z} + \sqrt{(1-\alpha^2)}z_{x_{n-1}} \end{aligned} \tag{4}$$

The advantage of this approach is that at each time step, the new velocity vector values can be applied directly to the constant velocity helper class in ns-3 that will calculate the position of the mobile nodes automatically. This eliminates the need for direction variables entirely, along with the trigonometric calculations that would be required to determine the node velocity vector.

The problem with this method is that it is uncharacteristic of aircraft in flight; it is not easy to model airplane flight based upon the plane's velocity in the x-direction, its velocity in the y-direction, and its velocity in the z-direction. Aircraft flight can be more accurately modeled using a velocity variable combined with variables for both direction and pitch. In the second approach, we start with the speed and direction variables found in the 2-dimensional Gauss-Markov model, and add a third variable to track the vertical pitch p of the mobile node with respect to the horizon as follows:

$$\begin{aligned} s_n &= \alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)}s_{x_{n-1}} \\ \theta_n &= \alpha \theta_{n-1} + (1-\alpha)\bar{\theta} + \sqrt{(1-\alpha^2)}\theta_{x_{n-1}} \\ p_n &= \alpha p_{n-1} + (1-\alpha)\bar{p} + \sqrt{(1-\alpha^2)}p_{x_{n-1}} \end{aligned} \tag{5}$$

Note that these formulæ represent basic 3-dimensional node movement. The objective of this model is to accurately represent the 3-dimensional node movement while limiting the complexity. It is not necessary to model the various flight controls, like the rudder, flaps, ailerons, angle of bank, etc. It is sufficient to model the aircraft movement itself using the Gauss-Markov algorithm, for which we assume the direction and pitch variables represent the actual angles at which the aircraft is moving.

After calculating these variables, the algorithm must determine a new velocity vector and send that information to the ns-3 constant velocity helper, in which the new node location is calculated. Assuming the direction and pitch variables are given in radians, the velocity vector $\bar{v}$ is calculated as:

$$\begin{aligned} v_x &= s_n \cos(\theta_n)\cos(p_n) \\ v_y &= s_n \sin(\theta_n)\cos(p_n) \\ v_z &= s_n \sin(p_n) \end{aligned} \tag{6}$$

V. AERORP DECISION METRICS

The basic operation of AeroRP consists of two phases [6]. In the first phase, each node learns and makes a list of available neighbors at any given point in time. It utilizes a number of different mechanisms to facilitate neighbor discovery, including periodic beacons and storing trajectory data in actual data packets. The second phase of the algorithm is to find the appropriate next hop to destination to forward the data packets. The location of the destination is known by all nodes, and a neighbor table is maintained that is updated based on the mechanism used in the first phase. The node uses this information to choose the next hop for each data packet. This is done by choosing the neighbor that has the smallest *time to intercept* (TTI) that indicates the time it will take for a node to be within transmission range of the destination if it continues on its current trajectory. Assume that node n_0 wants to send a data packet to the ground station D and the transmission range of all nodes is R. The TTI for each node is calculated as:

$$\mathrm{TTI} = \frac{\Delta d - R}{s_\mathrm{d}} \tag{7}$$

in which Δd gives the euclidean distance between the current location of a potential node and the destination node D and s_d is the relative velocity a potential neighbor has with respect to the destination. A high and positive s_d infers the neighbor is moving towards the destination at a high speed; high and negative s_d infers the neighbor is moving away.

A. Speed Component

Here, we work out exactly how s_d is calculated. Given a neighbor n_i that has geographical coordinates of x_i, y_i and a velocity of v_{xi}, v_{yi}, the velocity vector for n_i is calculated as:

$$v_i = \sqrt{{v_{xi}}^2 + {v_{yi}}^2} \tag{8}$$

The angle in degrees[1] between the positive x-axis of n_i plane and n_i velocity vector is:

$$\Theta = \text{atan2}(v_{yi}, v_{xi}) \times \frac{180}{\pi} \tag{9}$$

Destination D has geographical coordinates $x_\mathrm{d}, y_\mathrm{d}$. The angle between the positive x-axis of n_i plane and the imaginary line drawn between n_i and D is:

$$\bar{\Theta} = \text{atan2}(y_\mathrm{d} - y_i, x_\mathrm{d} - x_i) \times \frac{180}{\pi} \tag{10}$$

The difference between the angles ($\Theta - \bar{\Theta}$) gives the angle between n_i velocity and the imaginary line drawn between n_i and D. This gives us s_d:

$$s_\mathrm{d} = v_i \times \cos(\Theta - \bar{\Theta}) \tag{11}$$

B. Refining the Time to Intercept

The *time to intercept* (TTI) is the primary metric used for routing decisions in AeroRP. A source node calculates the TTI of its neighbors to understand when its neighbors will potentially be within transmission range of the destination and make the decision to route to the neighbor that will potentially be within transmission range of the destination the soonest and thus has the lowest TTI. Given a potential neighbor n_i with coordinates x_i, y_i, z_i and a destination D with coordinates $x_\mathrm{d}, y_\mathrm{d}, z_\mathrm{d}$, the Euclidean distance between the two is given as:

$$\Delta d = \sqrt{(x_\mathrm{d} - x_i)^2 + (y_\mathrm{d} - y_i)^2 + (z_\mathrm{d} - z_i)^2} \tag{12}$$

The TTI is calculated as follows in which R is the transmission range of the mobile devices:

$$\mathrm{TTI} = \begin{cases} 0 & \text{for } s_\mathrm{d} < 0 \text{ and } \Delta d > R \\ \frac{\Delta d - R}{s_\mathrm{d}} & \text{otherwise} \end{cases} \tag{13}$$

TTI=0 is a special case that indicates to never choose this neighbor as a next hop because we do not choose nodes that are moving away from the destination and not within transmission range. We use a negative TTI because this is an indication of a node being within transmission range of the destination that should be chosen as a next hop (lowest TTI). We always choose nodes within transmission range of the destination over nodes that are not. We also choose nodes that are within transmission range of the destination but moving away from the destination if there are no nodes within transmission range of the destination that are moving towards the destination. The nodes within transmission range moving towards the destination will be favored over those nodes within transmission range but moving away from the destination because these nodes will have a positive TTI due to a negative s_d and a negative $\Delta d - R$, and the nodes moving towards the destination will have a negative TTI.

C. Predicting Neighbors Out of Range

In a highly dynamic mobile environment in which links are constantly being broken due to high speeds, it may not be enough to just purge entries that have not been heard from based on a configurable hold time. Hence, we try to predict nodes that will be out of transmission range and remove them from next hop consideration. The predicted distance $\hat{d}$ between n_0 and n_1 is:

$$\begin{aligned} \bar{x}_i &= x_i + v_{xi}(t_1 - t_0) \\ \bar{y}_i &= y_i + v_{yi}(t_1 - t_0) \\ \bar{z}_i &= z_i + v_{zi}(t_1 - t_0) \\ \hat{d} &= \sqrt{(x_0 - \bar{x}_i)^2 + (y_0 - \bar{y}_i)^2 + (z_0 - \bar{z}_i)^2} \end{aligned} \tag{14}$$

Finally, the logic used to predict whether or not n_i is going to be out of n_0's range is given as:

$$\text{OutOfRange} = \begin{cases} \text{true} & \text{for } \hat{d} \geq R \\ \text{false} & \text{for } \hat{d} < R \end{cases} \tag{15}$$

[1] `atan2(x,y)` is a two-argument convenience function available in most programming languages that computes the angle in radians between the positive x-axis of a plane and the x, y coordinates provided in the arguments.

VI. AeroRP Operational Flow

The AeroRP routing protocol has both a neighbor discovery and a data forwarding phase as previously discussed. In order to discover neighbors in *beacon* mode, nodes receive AeroRP hello beacons from their neighbors. The node either creates a new entry in its neighbor table or updates its current data regarding the node from which it received the hello beacon. This neighbor table is used to calculate the TTI of its neighbors in order to make routing decisions.

Given the wireless nature of node communication in MANETs, it is possible for a node to be promiscuous and overhear all packets, even those packets that are not intended for a given node. In *beaconless promiscuous* mode, AeroRP takes advantage of this behavior and adds location information to each data packet per-hop as opposed to sending periodic hello beacons with this information. All nodes within transmission range, including those nodes that are not the intended receiver, can listen to the data packet and extract the location information from the header and store this location information for making routing decisions.

For the case when the node receives a packet for which the node itself has the lowest TTI but is not within transmission range of the destination, the packet can be queued in a configurable sized queue for a configurable amount of time. The queue is checked at a configurable frequency to see if there is a neighbor with a lower TTI than the local node. When a neighbor with a lower TTI is encountered, the packets from the queue are sent at a configurable data rate. There are currently three different AeroRP modes for when the local node has the best TTI: 1) **Ferry**: queue the packets indefinitely until a node with a lower TTI is found, 2) **Buffer**: queue the packets in a finite sized queue with a finite timeout until a node with a lower TTI is found, and 3) **Drop**: drop the packet. The flow of receiving and routing a data packet in AeroRP is illustrated in Figure 2.

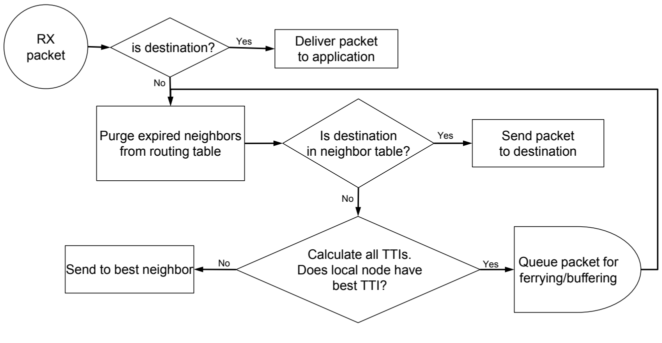

RX packet
is destination?
Yes
Deliver packet to application
No
Purge expired neighbors from routing table
Is destination in neighbor table?
Yes
Send packet to destination
No
Send to best neighbor
No
Calculate all TTIs. Does local node have best TTI?
Yes
Queue packet for ferrying/buffering

Fig. 2: AeroRP flow

When receiving a data packet, a node uses its neighbor table to decide how to route the packet. If the node is not the packet's destination, the node will clean its neighbor table of stale entries and those nodes that are predicted to be out of range as discussed in Section III.C. If one of the neighbors is the destination of the packet, the packet will be transmitted to the destination. Otherwise, the packet will be transmitted to a neighbor that has a better TTI. If the local node has the best TTI, it will ferry, buffer, or drop the packet depending on the AeroRP mode.

VII. Performance Analysis

We present the results of simulations conducted with the ns-3 simulator [11] to compare the performance of AeroRP in beaconless mode against traditional MANET routing protocols such as OLSR (optimized link state routing) [35], [36] and DSDV (destination-sequenced distance vector) [37], [38]. We choose these MANET protocols because they are common baseline link-state and distance-vector MANET routing protocols[2]. We compare the simulation performance under the different mobility models: Gauss-Markov, random waypoint, and constant position.

A. Gauss-Markov Variables

In order to examine the results of the mobility model in 3D, an MS-Windows application was developed that parses the ns-3 trace file data and displays the mobile node paths on the computer screen. This viewer allows one to zoom in and out and rotate the traces in three dimensions about the origin to observe changes in node mobility as various parameters of the Gauss-Markov model are adjusted.

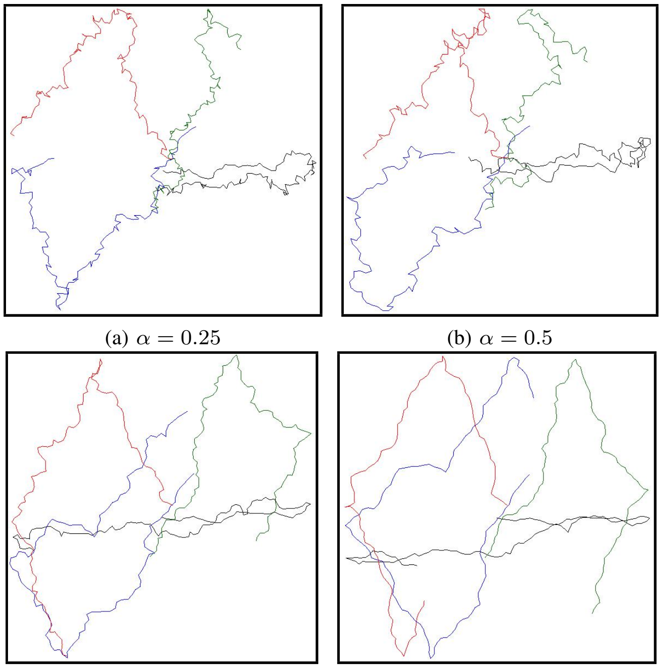
The Gauss-Markov model has several variables that can be modified [12]. Setting α between zero and one allows us to tune the model with degrees of memory and variation. In order to analyze the impact of α on the mobility, we conducted baseline simulations. Figure 3 shows variation in node movement with varying values of α. We observe that as α increases, the node paths become less random and more predictable. For the rest of the simulations we kept α value to be 0.85 to have some predictability in the mobility of the nodes, while avoiding abrupt AN direction changes.

Another significant parameter affecting node movement is the `TimeStep`, or how often a new set of values are calculated for each node. Since the nodes' velocity and direction are fixed until the next timestep, setting a large timestep will result in long periods of straight movement. A short timestep such as 0.25 s, will result in a path that is almost continuously changing. The timestep value for our simulations is set to 10 s.

B. Simulation Variables

For the random waypoint mobility model, we choose the pause time to be 0 s. since aircraft continuously move. The nodes are randomly placed in locations in the ns-3 constant position mobility model. The topology setup consists of between 5 and 60 wireless ANs that are randomly distributed over the simulation area. A single stationary sink node is located in the center of the simulation area representing the ground station. Other details of the simulation are shown in Table I. These parameters are chosen to identify routing performance and not to focus other layer issues [39]. 802.11b with 11 Mb/s is the most common and reliable wireless link layer protocol in ns-3 at the time of the AeroRP implementation.

[2] The ns-3 built in AODV model contains serious bugs, which we hope will be fixed soon; we are currently implementing DSR



(a) $\alpha = 0.25$

(b) $\alpha = 0.5$

(c) $\alpha = 0.85$

(d) $\alpha = 0.95$

Fig. 3: Gauss-Markov model traces with varying α

TABLE I: Simulation variables

Variable	Values
Simulation runs	10
Warmup time	100 s
Simulation time	1000 s
Simulation area	150 km^2
Velocity	$\leq$ Mach 3.5 (1200 m/s)
Initial position allocator	Random rectangle
Link layer	802.11b 11 Mb/s
RTS/CTS?	no
Packet fragmentation?	no
Propagation loss model	Friis
Transmission range	27.8 km
TX power	50 dbm
Packet size	1000 bytes
Sending rate	8 kb/s CBR
Transport protocol	UDP

We look at three different metrics to measure the performance of the different routing algorithms:

- **Packet delivery ratio (PDR)**: the number of packets received divided by the number of packets sent at the application layer.
- **Overhead**: bytes (typically headers) in excess of the payload data, required for the operation of the various protocols.
- **Delay**: the time between the initial transmission of a given packet and its arrival at the destination.

C. Simulation Results

Figure 4 shows the PDR of network performance of AeroRP in beaconless mode, OLSR, and DSDV routing protocols using the Gauss-Markov mobility model with α 0.85 and timestep of 10 s. The velocity of the nodes is kept at 1200 m/s for this baseline case. AeroRP outperforms the traditional link state and distance vector MANET routing protocols as the node density is increased. OLSR performs better than DSDV, which degrades significantly as the node density increases.

The network performance of AeroRP in terms of PDR for varying velocities is shown in Figure 5. We use the Gauss-Markov mobility model with an α value of 0.85 and timestep of 10 s. As the node density is increased from 20 to 60 in increments of 20, the PDR increases, due to the fact that there are fewer partitions in the network since the simulation area is covered by more nodes [39].

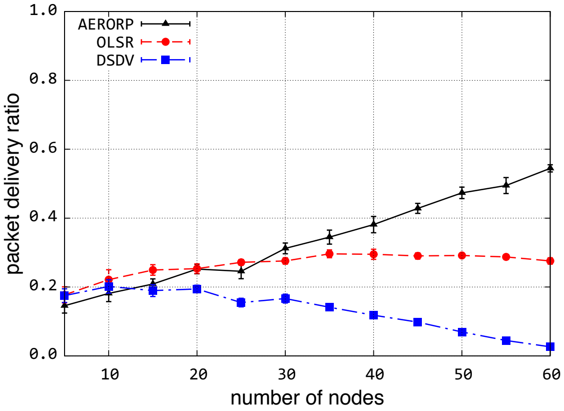


Fig. 4: Effect of node velocity on PDR

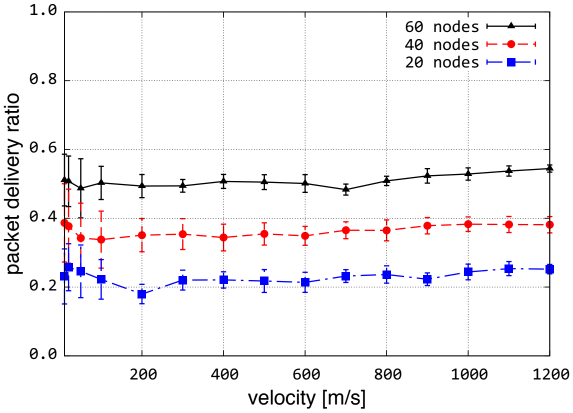


Fig. 5: Effect of node density on PDR

Figure 6 shows the average PDR of AeroRP as the number of nodes is increased. The velocity of the nodes is constant at 1200 m/s. As the number of nodes increases, the PDR increases for all three mobility models. This can be attributed to reduced partitioning as the node density increases. However, the PDR of the network is at maximum 55% using the Gauss-Markov mobility model, while the PDR reaches 80% using the random waypoint and constant mobility models. One reason is that the z-direction component of the 3-D Gauss-Markov model adds additional distance for the transmission range of the nodes, compared to the 2-D random waypoint and constant

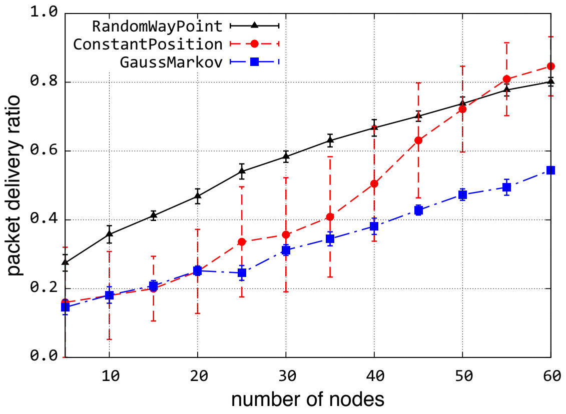


Fig. 6: Effect of node density on PDR

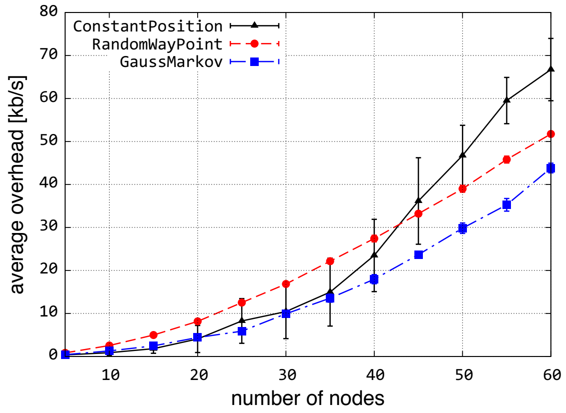


Fig. 7: Effect of node density on protocol overhead

position mobility models. Secondly, random waypoint can result in nodes covering most of the simulation area with an even distribution, whereas using the Gauss-Markov model, depending on the initial placement and direction, the nodes can be out of transmission range of each other for longer periods of time [40]. The fact that using the Gauss-Markov model results in lower performance indicates that it is more challenging for the routing protocols to adapt to realistic motion, making it imperative that new protocols be tested against this mobility model and not random waypoint alone.

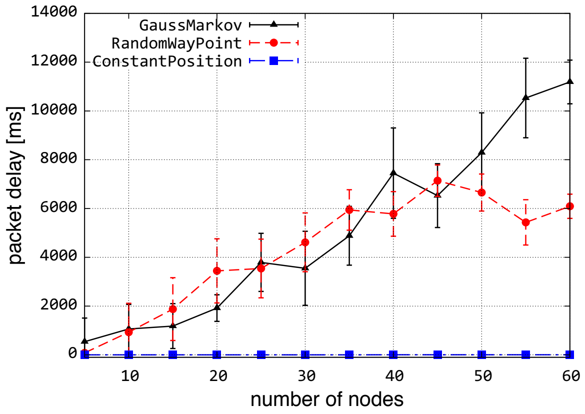


Fig. 8: Effect of node density on packet delay

As illustrated in Figure 7, the average overhead of the network increases with the node density. The overhead of AeroRP beaconless mode in Gauss-Markov mobility model is the lowest as the node density is increased. Since the Gauss-Markov mobility model results in fewer abrupt direction changes compared to the random waypoint model, route calculations are done less frequently. Also, since the PDR is less for the Gauss-Markov mobility model, intuitively one can expect to have less overhead for Gauss-Markov model. The overhead in constant mobility model is the largest compared to the other mobility models at high node-densities.

The effect that node density on the delay of data packet transmissions is shown in Figure 8. In the constant position mobility model initial placement of the nodes affects end-to-end delay at the MAC layer. Thus if the network path is connected packets are delivered to destination, or if the network is partitioned packets are buffered but never delivered and thus do not affect the delay statistics. For the random waypoint and Gauss-Markov models, buffering delays dominate the delay statistics, which increase around the same rate until 50 nodes are in the network. The delay increases more for Gauss-Markov mobility model compared to the random waypoint model. Our simulation results are in line with the conclusions made in [41] that network performance depends on the values of the mobility model chosen.

VIII. Conclusions and Future Work

The three dimensional Gauss-Markov mobility model presented in this paper can be used to model airborne MANETs. Compared to the previously available ns-3 mobility models, the Gauss-Markov model provides more realistic node movement. Different types of aircraft movements can be simulated by varying the three basic tuning parameters of α, standard deviation, and time step. When comparing simulations run with the Gauss-Markov model to results acquired using other mobility models, we see that the performance is lower, indicating that the Gauss-Markov model presents a greater challenge to the routing protocols than is seen with memoryless models. In evaluating AeroRP performance against other legacy MANET routing protocols using the ns-3 simulator in realistic high-velocity scenarios, our results indicate that AeroRP generally outperforms them regardless of mobility model used.

Our future work includes comparing AeroRP to AODV (when fixed) and DSR (under development by the KU ResiliNets group). Furthermore we will test AeroRP with the AeroTP transport protocol [42] to analyze their performance when combined, including the effects of buffering modes on transport layer operation.

Acknowledgments

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 member Abdul Jabbar for the original AeroRP algorithm development.

References

[1] J. P. Rohrer and J. P. G. Sterbenz, "Performance and disruption tolerance of transport protocols for airborne telemetry networks," in *Proceedings of the International Telemetering Conference (ITC)*, (Las Vegas, NV), October 2009.

[2] 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.

[3] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks: Special Issue on Resilient and Survivable Networks (COMNET)*, vol. 54, pp. 1245–1265, June 2010.

[4] J. Prokkola, L. Leppännen, and T. Bräysy, "On the effect of traffic models to the performance of ad hoc network," in *IEEE MILCOM*, pp. 422–427, October 2003.

[5] J. P. Rohrer, A. Jabbar, E. Perrins, and J. P. G. Sterbenz, "Cross-layer architectural framework for highly-mobile multihop airborne telemetry networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, (San Diego, CA, USA), pp. 1–9, November 2008.

[6] A. Jabbar and J. P. G. Sterbenz, "AeroRP: A geolocation assisted aeronautical routing protocol for highly dynamic telemetry environments," in *Proceedings of the International Telemetering Conference (ITC)*, (Las Vegas, NV), October 2009.

[7] J. P. Rohrer, A. Jabbar, E. K. Çetinkaya, and J. P. Sterbenz, "Airborne telemetry networks: Challenges and solutions in the ANTP suite," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, (San Jose, CA, USA), pp. 74–79, November 2010.

[8] J. P. Rohrer, A. Jabbar, E. K. Çetinkaya, E. Perrins, and J. P. Sterbenz, "Highly-dynamic cross-layered aeronautical network architecture," *IEEE Transactions on Aerospace and Electronic Systems (TAES)*, vol. 47, October 2011. (to appear).

[9] T. Camp, J. Boleng, and V. Davies, "A survey of models for ad hoc network research," *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends, and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[10] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* (T. Imielinski and H. F. Korth, eds.), vol. 353 of *The Kluwer International Series in Engineering and Computer Science*, ch. 5, pp. 153–181, Norwood, MA: Kluwer Academic Publishers, 1996.

[11] "The ns-3 network simulator." http://www.nsnam.org, July 2009.

[12] 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 *Proceedings of the International Telemetering Conference (ITC)*, (San Diego, CA), October 2010.

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

[14] C. Lemmon, S. M. Lui, and I. Lee, "Geographic Forwarding and Routing for Ad-hoc Wireless Network: A Survey," in *IEEE NCM*, pp. 188–195, 2009.

[15] M. G. de la Fuente and H. Ladiod, "A performance comparison of position-based routing approaches for mobile ad hoc networks," *Vehicular Technology Conference (VTC)*, pp. 1–5, October 2007.

[16] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," in *ACM MobiCom*, pp. 76–84, 1998.

[17] Y.-B. Ko and N. H. Vaidya, "Location-aided Routing (LAR) in Mobile Ad Hoc Networks," in *ACM MobiCom*, pp. 66–75, 1998.

[18] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *ACM MobiCom*, pp. 243–254, 2000.

[19] A. Capone, L. Pizziniaco, I. Filippini, and M. de la Fuente, "A SiFT: an efficient method for trajectory based forwarding," in *2nd International Symposium on Wireless Communication Systems*, pp. 135–39, September 2005.

[20] R. L. Lidowski, B. E. Mullins, and R. O. Baldwin, "A Novel Communications Protocol using Geographic Routing for Swarming UAVs Performing a Search Mission," in *IEEE PERCOM*, pp. 1–7, 2009.

[21] E. Kuiper and S. Nadjm-Tehrani, "Geographical Routing in Intermittently Connected Ad Hoc Networks," in *IEEE AINI*, pp. 1690–1695, March 2008.

[22] B. Zhou, Y.-Z. Lee, and M. Gerla, "Direction assisted geographic routing for mobile ad hoc networks," in *IEEE MILCOM*, pp. 1–7, November 2008.

[23] Z. Jin, N. Yan, and L. Bing, "Reliable On-Demand Geographic Routing Protocol Resolving Network Disconnection for VANET," in *IEEE WiCOM*, pp. 1–4, Sept. 2009.

[24] S. Jung *et al.*, "A Geographic Routing Protocol Utilizing Link Lifetime and Power Control for Mobile Ad Hoc Networks," in *ACM FOWANC*, pp. 25–32, 2008.

[25] Q. J. Chen *et al.*, "Adaptive Position Update in Geographic Routing," in *IEEE ICC*, vol. 9, pp. 4046–4051, June 2006.

[26] J. Na and C. kwon Kim, "GLR: A novel geographic routing scheme for large wireless ad hoc networks," *Computer Networks*, vol. 50, no. 17, pp. 3434–3448, 2006.

[27] J. Li and S. Shatz, "Toward Using Node Mobility to Enhance Greedy-forwarding in Geographic Routing for Mobile Ad Hoc Networks," in *Proc. of MODUS*, pp. 1–8, April 2008.

[28] M. Iordanakis *et al.*, "Ad-hoc Routing Protocol for Aeronautical Mobile Ad-Hoc Networks," in *Proc. of CSNDSP*, pp. 543–547, 2006.

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

[30] N. K. E. Sakhaee, A. Jamalipour, "Aeronautical Ad Hoc Networks," in *IEEE WCNC*, vol. 1, pp. 246–251, April 2006.

[31] F. Tchakountio and R. Ramanathan, "Anticipatory Routing for Highly Mobile Endpoints," in *Proc. of IEEE WMCSA*, pp. 94–101, 2004.

[32] F. Tchakountio and R. Ramanathan, "Tracking Highly Mobile Endpoints," in *ACM WoWMoM*, pp. 83–94, 2001.

[33] J. P. Rohrer and J. P. Sterbenz, "Performance and disruption tolerance of transport protocols for airborne telemetry networks," in *International Telemetering Conference (ITC)*, (Las Vegas, NV), October 2009.

[34] B. Liang and Z. Haas, "Predictive distance-based mobility management for PCS networks," in *The Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, pp. 1377–1384, Mar. 1999.

[35] P. Jacquet *et al.*, "Optimized Link State Routing Protocol for Ad Hoc Networks," in *Proc of IEEE INMIC*, pp. 62–68, Aug. 2002.

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

[37] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *The conference on Communications architectures, protocols and applications (SIGCOMM)*, (New York, NY, USA), pp. 234–244, ACM, 1994.

[38] 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 *Proceedings of the ICST SIMUTools Workshop on ns-3 (WNS3)*, (Barcelona, Spain), March 2011.

[39] K. Peters, A. Jabbar, E. K. Çetinkaya, and J. P. Sterbenz, "A geographical routing protocol for highly-dynamic aeronautical networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, (Cancun, Mexico), pp. 492–497, March 2011.

[40] 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.

[41] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[42] K. S. Pathapati, J. P. Rohrer, and J. P. G. Sterbenz, "Edge-to-edge ARQ: Transport-layer reliability for airborne telemetry networks," in *Proceedings of the International Telemetering Conference (ITC)*, (San Diego, CA), October 2010.