Workshop on ns-3 - WNS3 2015 - ISBN: 978-1-4503-3375-7
Castelldefels (Barcelona), Spain - May 13-14, 2015

Simulating Large-Scale Airborne Networks with ns-3

Ben Newton, Jay Aikat, Kevin Jeffay
University of North Carolina at Chapel Hill
Chapel Hill, North Carolina, USA
bn@cs.unc.edu, aikat@cs.unc.edu, jeffay@cs.unc.edu

ABSTRACT

Large-scale airborne networks, which connect airborne nodes with high-bandwidth communication links are being actively pursued commercially. We propose utilizing thousands of operational passenger and cargo aircraft as the principal components of an airborne network which could provide high-speed Internet to passengers on-board and on the ground. To simulate such a network we have augmented the ns-3 network simulator with a model for ingesting and processing aircraft position information, a steerable directional antenna model, a wireless point-to-point channel and associated net devices, and a distributed topology control application to manage the topology of the mesh network. We describe our implementation of these models and some tools for visualizing airborne networks. Using a simulation of a large airborne network, covering the United States, we perform experiments to evaluate the effectiveness of using the Optimized Link State Routing Protocol (OLSR) to route network traffic. Our simulations lead us to conclude that OLSR is likely not a good fit for our envisioned network.

Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: General.

General Terms

Experimentation, Design

Keywords

ns-3, Airborne Networks, Steerable Directional Antennas, Topology Control, OLSR

1. INTRODUCTION

The notion of forming high-bandwidth digital communication networks among airborne platforms was a dream previously reserved for cutting-edge military research programs. However, recent announcements by both Google and Facebook seem to have catapulted this concept into the civilian world. In June 2013 Google announced Project Loon [1] [2], and launched several balloons outfitted with special communications equipment into the stratosphere. Google plans to establish an airborne network of countless of these balloons circling the globe, with the stated purpose of connecting people in rural and remote areas to the Internet. Less than a year after Google's announcement, Facebook followed suit, purchasing a company that makes high-flying drones [3] and announcing that they've "been working on ways to beam internet to people from the sky." [4] It appears these prominent companies have entered into a race to develop a massive airborne network [5].

There are many issues and concerns which Google and Facebook will need to face, such as what waveform, protocols, and addressing methods will be used to connect the airborne nodes to the end users. However, some of the most interesting issues relate to how the airborne nodes themselves will be connected to the Internet. One option is to form a high-bandwidth mesh network among the airborne nodes, and connect that network to ground station gateways.

In order to form the long-range high-bandwidth connections required for such a network, directional communication links have been proposed [6]. Compared with traditional omni-directional links, these directional links have longer ranges, use less power, and produce less interference. Of particular interest to our research are steerable directional links, which are mounted on a gimbal, and can be "steered" to point in any direction.

The military uses steerable directional radio frequency (RF) links [7] to create high-bandwidth connections among aircraft, and between aircraft and ground stations. These military RF networks, however, have traditionally been very small-scale, generally linking one or two aircraft to a ground station [8]. More recently DARPA has experimented with free-space optics (FSO). FSO links use infrared lasers to wirelessly communicate at extremely high rates. In a way, FSO links are like wireless fiber optic connections. DARPA has demonstrated a hybrid FSO/RF link which can form air-to-air connections with ranges up to 200 km and air-to-ground connections with slant ranges up to 130 km, while maintaining data rates of 3 to 9 Gb/sec [9] [10].

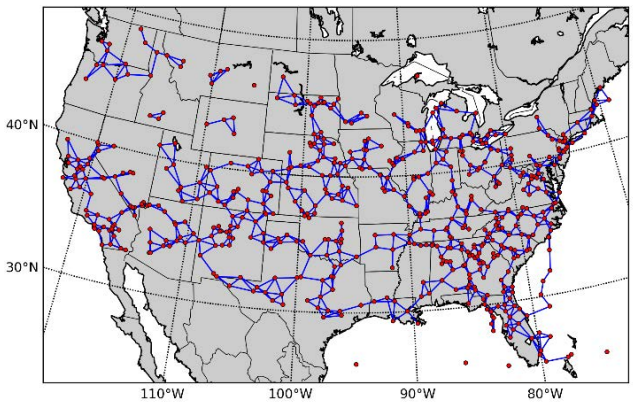

Figure 1: An example of an airborne network formed (blue lines) among 905 actual aircraft positions (red dots).

WNS3 2015, May 13 2015, Barcelona, Spain

ACM 978-1-4503-3375-7/15/05...$15.00
DOI: http://dx.doi.org/10.1145/2756509.2756514

For our research we propose using commercial aircraft as the principal components of an airborne network [11]. Figure 1 shows what the topology of this large-scale airborne network might look like.

Utilizing the already existing fleet of passenger and cargo aircraft, instead of or in addition to balloons or drones, has the added advantage of enabling high-bandwidth connections to users on-board the aircraft. High-bandwidth connections on and between commercial aircraft also opens the door for a host of other applications, such as streaming live aerial images to the ground, sharing weather and turbulence information between aircraft, or even enabling remote co-piloting.

Aircraft, in this envisioned network, would be connected to one another and to a set of ground stations using directional links. Each aircraft participating in the network would be equipped with some number of directional links (likely 3 or 4), each of which may be used to form a connection with a neighboring aircraft or ground station. But how should the physical topology of this network be determined? Unlike traditional wireless networks, where the network topology can be implicitly controlled by adjusting transmission range [12], our network of directional links must employ explicit topology control [11]. Some protocol must manage which set of neighboring aircraft (limited in size by the number of links) each node should connect with, and how and when to make changes to these connections, as the nodes move. Once a physical topology is established, how should packets be routed through the network as the topology changes? Simulation is an obvious means to answer these important questions.

Academic research in this emerging field is extremely limited, and what defense research has been published is generally difficult to reproduce because of the need for specialized equipment or proprietary simulation systems. Currently there is little support for directional links and airborne networks in ns-3. We hope that using ns-3 to simulate our network, and contributing code to ns-3, will help make academic research in this area more accessible. Our models are still in their infancy, but we have built an initial foundation, which we hope will support further development.

Our contributions include: a model for ingesting and processing aircraft position information into ns-3, a steerable directional antenna model which enables an antenna's orientation to change during the simulation, a wireless point-to-point channel and associated net devices for simulating directional connections, a distributed topology control application to manage the topology of the mesh network, and finally an adapted version of PyViz for visualizing the airborne network.

In this paper we describe these added ns-3 models and use them to simulate a large airborne network covering the United States. We further evaluate the effectiveness of using the Optimized Link State Routing Protocol (OLSR) to route packets through this network. OLSR is a proactive routing protocol optimized for use in mobile ad-hoc networks (MANET). The remainder of the paper is organized as follows. Section 2 gives an overview of some related research projects. Section 3 describes the models and tools we have developed in ns-3 to support simulating our network. Next, Section 4 describes some initial experiments using our models in ns-3, and Section 5 discusses the results of these experiments. In section 6 we describe our ideas for future work, and we conclude in Section 7.

2. RELATED WORKS

Simple directional antennas are currently implemented in ns-3, but these are limited in that they are assumed to be stationary. Their orientation can't change as the simulation progresses. Also, the antenna orientation is specified using a single value, thus the antennas can't point at an arbitrary point in 3-dimensions.

Nakhkoob et al. developed a detailed ns-2 model for their research involving multi-transceiver free-space optical mobile ad-hoc networks [13] [14]. The design for some of the elements of our models were influenced by their implementation. Their model, however, was intended for relatively short distances, and the differences between ns-2 and ns-3 necessitated significant changes.

Much of the research related to airborne networks has been sponsored by the military, and tested using commercial network simulators such as OPNET or QualNet. Tiwari et al. observe, in [15], that a priori knowledge may be a key to better performance in both military and commercial airborne networks. They discuss the advantages of using mission plans to improve communication planning, and propose a protocol for exchanging flight path information. Their proposed network is simulated using OPNET. In [16] the same group of authors detail a mobility-aware routing protocol especially suited for Airborne Networks. The protocol pre-computes a time dependent routing table using the expected positions of all the nodes. Their protocol is analyzed in [17] using QualNet, and shown to outperform OLSR [18]. Their solution, however, doesn't appear to handle topology control, and they consider only the racetrack orbits of small military networks rather than thousands of nodes flying complex routes.

Medina, et al. have developed the Geographic Load Share Routing algorithm (GLSR) [19] specifically for airborne networks. Their protocol adds load balancing to the Greedy Perimeter Stateless Routing (GPSR) protocol [20], allowing the mesh network resources to be utilized more evenly. They simulate their protocol in the North Atlantic corridor [6], showing how the airborne nodes are able to receive packets from land-based Internet gateways. Their simulated system assumes each aircraft is equipped with a half-duplex RF uniform circular array antenna, rather than the multiple steerable directional antennas we envision.

A small-scale airborne network utilizing Free Space Optical links is simulated by Epstein and Mehta in [8]. They evaluate two standard routing protocols, Open Shortest Path First (OSPF) and Enhanced Interior Gateway Routing Protocol (EIGRP), in an Airborne Network which connects two ground stations using airborne nodes. OSPF is shown to perform poorly due to the fact that it cannot update faster than once per second. A fast response to link outages is emphasized as an important characteristic of a routing protocol for Airborne Networks. Our goal is to simulate a much larger airborne network, which will likely require a more specialized routing protocol.

3. DESIGN

In this section, we describe the various components we have added to ns-3 to support our airborne network simulations.

3.1 ASDI Model

Rather than use a model to simulate aircraft mobility, we have the luxury of being able to use previously collected flight path data to model the movements of aircraft in our network. To accommodate incorporating this data into our simulation we have created a new NS-3 model.

The Federal Aviation Administration (FAA) makes available to some industry and research partners access to aircraft position information for most of the aircraft in the United States National Airspace System (NAS). This data, called Aircraft Situation Display to Industry (ASDI) [21], includes not only the position and velocity of each aircraft, but also other information such as the flight call sign and flight plan information. We have access to ASDI data recorded over the course of several days in July 2013. We use a subset of this data for our simulations.

Our ASDI model provides facilities to open and parse ASDI files. For each ASDI file parsed, an ASDI object is created. Each ASDI object has a set of Flight objects, each containing a time-ordered list of AircraftWaypoint objects. Each AircraftWaypoint includes: the timestamp, the ground speed of the aircraft, the geodetic position of the aircraft (latitude, longitude, altitude), and the derived Cartesian coordinates of the aircraft when the Earth-centered, Earth-Fixed (ECEF) coordinate system is assumed.

The ECEF coordinate system has its origin (0,0,0) at the center of the earth's mass, and is fixed to the earth (rotating with the earth as it spins). Use of ECEF positions has also been proposed by other NS-3 models, such as the TV Spectrum Transmitter model, which added ECEF methods in the GeographicPositions class of the Mobility Module. Calculating and storing the ECEF positions allows us to quickly determine the distance between two aircraft, using normal Cartesian methods.

The other main function of the ASDI model is to create the appropriate nodes, representing the aircraft in the ASDI file, and to schedule their mobility. Our model utilizes the constant velocity mobility model to schedule the mobility. It first sets appropriate initial positions for each aircraft. Then, it need only schedule velocity changes at appropriate times and the constant velocity mobility model handles all the hard work of efficiently determining the interpolated node positions as the simulation runs. To accurately determine what velocity to set when the aircraft arrives at waypoint *n*, our model utilizes waypoint *n+1*. It determines what velocity (both speed and direction) is required to reach waypoint *n+1*'s position at the time associated with waypoint *n+1*. It then schedules an event to update the velocity to this value at the time associated with waypoint *n*. The velocities are computed scheduled in this way for each waypoint of each flight. The ASDI model also handles the extra processing required to accommodate nodes which enter and/or leave the simulation.

3.2 Steerable Directional Antenna Model

We provide two modes for forming connections in our system. The first, is more realistic, and involves simulation of the actual pointing of the antennas. In this mode a Topology Control Algorithm commands antennas to point toward other nodes. This causes the pointing direction to be updated. A timer allows a separate function to periodically determine which antennas are aligned. If two are aligned, the alignment table in the channel is updated, enabling packets to flow from one node to another.

Steerable directional antennas often employ a tracking system to automatically adjust the pointing of the antennas once connected. The second mode assumes the existence of such a tracking system. Upon creation of a connection between two nodes, where the connection did not exist in the previous iteration, each node explicitly informs the channel of its intention to connect to the remote node. Once both nodes have connected, the alignment table in the channel is updated. This mode is potentially less realistic, but saves a lot of computation time. This mode doesn't require the steerable direction antenna model, discussed next.

When simulating using the first mode, each directional connection is formed by two directional antennas, one at each end of the connection. The term antenna, however, can be misleading as many directional RF antennas look and behave much more like satellite dishes. We similarly borrow the term for FSO links, though these "antennas" may look and act more like telescopes with lasers.

Our SteerableDirectionalAntennaModel inherits from the standard ns-3 AntennaModel, but adds the attributes and functionality described below. Each directional antenna has a beam width, a transmitter diameter, and a pointing direction consisting of azimuth and elevation components. The beam width of a directional antenna is a measure of how quickly a signal spreads when leaving the antenna. More technically, beam width generally measures the angle between lines, on opposing sides of the beam center, which represent where the irradiance is half that of the beam's peak irradiance. For optical links this is also often called the divergence angle. RF directional antennas may have a beam width of 7 degrees, while infrared lasers may have divergence angles as small as 10 microradians. The transmitter diameter is just that, the diameter of the transmitting antenna. These two values are used to determine the radius of the beam at any distance from the transmitting antenna, a functionality which will prove useful for the implementation of the wireless channel (described in 3.3).

The azimuth and elevation values, together, are able to specify any pointing direction relative to the antenna's local reference frame. The azimuth value (-180 to 180) specifies where in the horizontal plane the antenna should point, while the elevation value (-90 to 90) specifies where in the vertical plane the intended target lies. For example, if an antenna were mounted on the belly of an aircraft, and its reference frame was such that 0 azimuth, and 0 elevation were straight ahead (parallel to the line through the center of the aircraft and its nose), commanding the antenna to point at 90 degrees azimuth, 0 degrees elevation would cause the antenna to point directly to the right, assuming azimuth increases in a clockwise direction. Further, assuming elevations increase downwards, commanding an antenna to point at -90 degrees azimuth, 45 degrees elevation, would cause the antenna to point directly to the left and downwards 45 degrees. Currently in our model all azimuths and elevations are given in reference to a global frame, but accounting for the aircraft's frame of reference is something we hope to pursue in the future.

The directional antenna contains a pointer to a mobility model instance. This mobility model represents the current position of the directional antenna, and allows it to point at a specific point in three-dimensional space (determining the exact azimuth and elevation value internally). This mobility model could be a static position, a copy of an aircraft's mobility model, or even a fixed offset from an aircraft's mobility model.

3.3 Wireless Point-to-Point Channel

The real key to the simulation of our network is a new channel for what we call wireless point-to-point links. We added this channel to ns-3 to simulate the unique characteristics of directional

wireless connections. When connected, these links behave much like a wired point-to-point connection, but the existing point-to-point model doesn't suffice. Unlike omnidirectional channels, which send packets to all the Net Devices attached to the channel, our channel model need only send a packet to the Net Devices whose antennas are aligned with the antenna of the sending Net Device. To accomplish this, the channel stores an alignment map, which indicates which other Net Devices are aligned with a given Net Device. Currently our implementation ignores interference and allows only pairings of Net Devices. For the FSO links we are currently simulating, this seems to be a reasonable assumption, given the small beam widths, and large distances. However, we would like to implement interference and more sophisticated mappings in the future.

Each time a packet enters the channel we determine which NetDevice, if any, is aligned with the source NetDevice. The receipt of the packet is then scheduled on the destination NetDevice at some future time, where the future time accounts for the transmission time and propagation delays, as in other wireless channels. Figure 2 shows a diagram of how these components work together. Many wireless point-to-point NetDevices are connected to the wireless point-to-point channel. Each NetDevice has an associated steerable directional antenna. If the antennas are correctly aligned, then the wireless point-to-point channel "connects" the Net Devices together, such that packets sent from one reach the other, and vice versa. Packets sent by NetDevices which are not aligned to any other link are dropped.

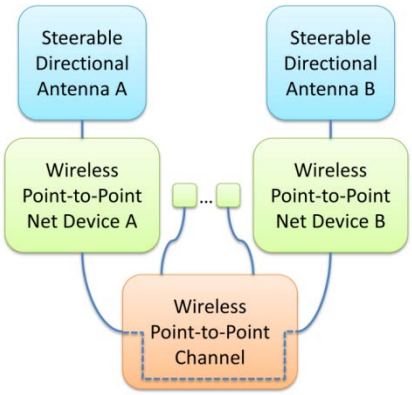

Figure 2: Diagram showing the relationship and use of the components which enable simulating directional links.

The lingering question is then, how does the alignment map get generated and updated? For the first mode, described above, where the directional antennas determine alignment, the channel employs a periodic timer which calls the private ReviseAlignment() method. This method first checks that entries in the alignment map are valid, removing them if they are not, and then searches for antennas that are newly aligned, adding them to the map. Mutual alignment of antennas is determined by another method which uses the beam radius method of the directional antenna model, to determine whether each antenna is inside the other antenna's transmission beam. For the second mode, where tracking is assumed, the alignment map is explicitly controlled by the topology control protocol described in Section 3.5.

3.4 Wireless Point-to-point Net Device

Our wireless point-to-point net device is much like the traditional point-to-point net device with two exceptions. First, our wireless version has an associated Directional Antenna. Second, we add to the wireless point-to-point protocol header two extra fields: the sending and receiving MAC addresses.

More sophisticated Link Layer protocols could, and likely should be considered here, but this simple model appears to suffice for our initial experiments.

3.5 Topology Control Application

We have explained how the alignment map is updated, based upon the current pointing direction of the antennas associated with each NetDevice, or by explicit connection creation (as in the second mode). However, we have not yet explained how the antenna pointing directions are set, or how the connections are explicitly created. This is the function of the topology control protocol, which we have chosen to simulate at the application layer. Given a set of nodes in the network, the positions of the nodes, and the number of links on each node, the job of the topology control protocol is to determine the physical topology of the network. In other words, the topology control protocol must determine for each node with N links, which N neighbors it should point its links at in order to achieve some global network topology. It is important to note, that it is the physical topology of the network, not the routing topology that is being managed by this protocol.

Our proposed topology control protocol makes use of Automatic Dependent Surveillance-Broadcast (ADS-B) transmissions [22]. ADS-B is an on-aircraft beacon system that sends altitude, position, and airspeed information about an aircraft to FAA monitoring stations, and to all nearby aircraft. By the year 2020 all large commercial aircraft will be required to be ADS-B capable. ADS-B allows each aircraft in our envisioned network to store the current positions of all other aircraft within about 280 km (the approximate air-to-air range of ADS-B transmissions).

Because ADS-B already handles exchanging position information with neighbors, our network topology can be managed utilizing "zero overhead" on its links. Our fully distributed topology control protocol builds on the results presented in our previous work [11]. Our implementation is similar to Localized Tree-based Reliable Topology (LTRT) [23], but augmented to honor degree constraints. Using only the positions of nodes within ADS-B range, each node computes a local topology, and then points its antennas accordingly. This yields a global topology with the desirable attributes for our airborne network. Each node periodically runs ComputeLocalTopology(), updating the pointing directions of its associated NetDevices accordingly, or explicitly connecting the NetDevices (as in the mode where tracking is assumed).

3.6 Propagation Loss Model

We are also developing a propagation loss model for long-range FSO links based on the link budgets for military FSO links [24]. This model takes into account the atmospheric turbulence, the altitude, and the visibility. For our initial tests, however, and the simulations described in this paper we have decided to ignore propagation loss.

3.7 Adapted Version of PyViz

To enable effective visualization and debugging of our unique network simulation we adapted the existing PyViz utility. PyViz is a Python-based network visualization tool which is integrated with ns-3. One thing we did to make PyViz work for our simulations was to redefine the axes to represent latitude and longitude, adjusting the scale accordingly. Figure 3 shows a

screenshot of several aircraft and ground stations displayed through the adapted version of PyViz.

The lighter lines represent all the potential connections to neighbors within range, whereas the dark lines represent the connections the topology control protocol has chosen to use to form a network topology.

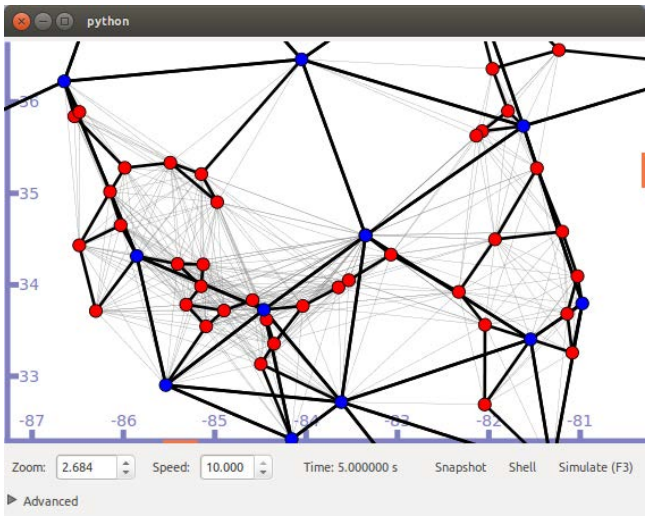

Figure 3: a screenshot of an airborne network viewed in our adapted version of PyViz. Light lines represent potential connections, thick lines represent established connections, when viewed in color red circles represent aircraft and blue circles, ground stations.

Another feature we added to PyViz is the ability to periodically export the network information to a Keyhole Markup Language (KML) file. This KML file, when loaded into Google Earth, or a similar application, allows a user to view a representation of the 3-dimensional airborne network being simulated in a more compelling and realistic context. Figure 4 shows a screenshot of a simulated airborne network visualized in Google Earth.

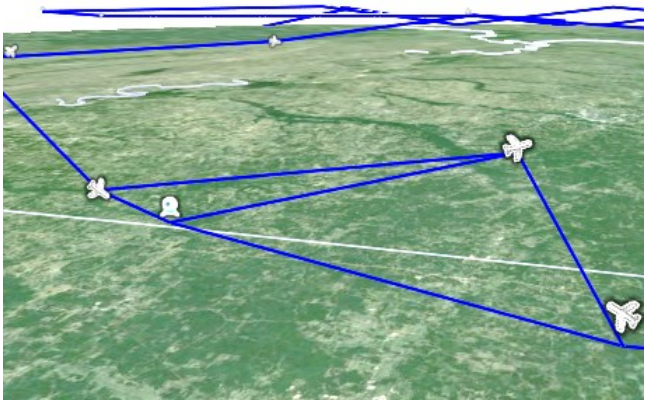

Figure 4: a screenshot of a network being simulated in ns-3, and visualized in Google Earth. Three nearby aircraft are connected to the ground station with other aircraft and connections visible on the horizon.

4. SIMULATION EXPERIMENTS

Using our models we perform a set of simulations utilizing the OLSR routing protocol and our distributed topology control protocol. Our purpose is to demonstrate the functionality of our added models, and to evaluate whether OLSR is a good candidate for routing data in the envisioned airborne network. For this experiment we consider a subset of the aircraft flying above 3,048 meters (10,000 ft.) and within the bounds of the continental United States. We assume a maximum range of 200 km for air-to-air FSO connections. The network for this experiment also includes 100 ground stations (each with three FSO links) which have been strategically placed within the continental United States. This is roughly equivalent to the number of ground stations employed by Aircell in 2009 to support Gogo Inflight Internet. The maximum range for an air-to-ground connection is estimated to be 100 km (shorter than an air-to-air connection due to the increased atmospheric attenuation when propagating through the layers of atmosphere). Our goal in this experiment is to evaluate the effectiveness of OLSR when presented with data flowing from the ground stations to the air nodes (which we anticipate will represent the majority of the traffic in our network).

4.1 Assumptions

We make several simplifying assumptions in order to constrain the problem for our initial experiments. First, we assume that all aircraft are equipped with the same number of FSO links, which we assume to be 3. This seems to be a good number, since more than 2 are required to do anything but form strings of nodes, and 4 or more may not be reasonable due to cost and weight constraints. We also ignore the issues of aircraft occlusion, and assume that the wings and fuselage do not obstruct the connections. Further, we assume there are no clouds obstructing the FSO connections. This will often be the case for short connections between high-flying aircraft, but is not realistic in general. Currently we do not simulate the link attenuation, and instead assume that all packets are transmitted without error across the wireless link. We recognize that this is unreasonable, but ignoring the attenuation and bit errors initially, allows us to focus on creating a full simulation before we transition to making it more realistic. Finally, although FSO links will require a few seconds to re-point and re-acquire a connection, we do not yet account for that time in our simulations.

4.2 OLSR

We use the Optimized Link State Routing Protocol (OLSR) for our experiments. OLSR [18] is a commonly studied mobile ad-hoc network (MANET) routing protocol. It is categorized as a proactive routing protocol because it proactively creates and manages routes between nodes in a network. Compared with on-demand protocols, which wait to establish routes until they are needed, proactive protocols generally require more overhead, but have decreased latency in establishing routes. OLSR uses hello messages to discover nodes, and uses topology control messages to disseminate link state information throughout the network. Each node uses this information to compute the next hop for every destination in the network. We use the implementation of OLSR which is integrated with ns-3.

Table 1 shows a listing of the attributes for this OLSR model, their default values, and the values used in our experiments.

Table 1: OLSR Attribute Configuration

Parameter	Value(s)
Hello message emission interval	2 (default), 0.5, and 0.125 seconds
TC message emission interval	5 seconds
MID message emission interval	5 seconds
HNA message emission interval	5 seconds
Willingness (to carry/forward traffic for other nodes)	default

Table 2 shows a listing of other simulation attributes and values used for our experiment.

Table 2: Other Simulation Attribute Values

Attribute	Value(s)
Simulated Scenario Time	10 minutes
Simulation Area	Contiguous United States
Number of Nodes	Increasing from 33 to 588
Topology Update Interval	1.0 second
Data Rate Air-to-Air	10 Gbps
Data Rate Ground	10 Gbps
Propagation Delay Model	Constant Speed (speed of light in a vacuum)
UDP Flow Rate	1 Mbps
Packet Size	512 bytes
CPU Time required to simulate	6 minutes to 10 hours per sample
Number of samples	20 sets of nodes, 10 times each, 3 hello intervals = 600 total samples

4.3 Experiment

Using the OLSR model in ns-3 and given the data and assumptions described above, we analyze the effect of adjusting the hello message interval on our network. Because the network topology is so dynamic, it is expected that performance will be improved if the hello messages are sent more rapidly. For this experiment we consider a subset of the air traffic (specifically Delta Airlines and Southwest Airlines aircraft) flying directly above the continental United States. Once aircraft fly over the international boarder or the coastline they are no longer a part of the network. For this simulation we use the second mode for determining alignment (where tracking is assumed and alignment is explicitly established). Our simulation also assumes the existence of 100 strategically placed ground stations spread across the United States. Figure 5 shows a map of the United States with dots indicating the ground station locations. We simulate 10 minutes of actual flight activity beginning at noon Eastern Time (17:00 UTC) on July 11, 2013, and randomly select increasingly larger sets of aircraft to participate in the network. The capacity of the directional links is set to 10 Gbps. A constant bit-rate (CBR) UDP flow is created for each aircraft (excluding those active over the United States for less than 5.0 seconds during the experiment). All flows send CBR UDP traffic at a rate of 1 Mbps. Packets of a given flow originate at whichever ground station is geographically nearest the destination node at the beginning of the simulation, and are routed through the network to the destination node. Ground stations are connected together with a mesh network of regular (non-wireless) terrestrial point-to-point connections, with a delay of 2 milliseconds and capacity of 10 Gbps.

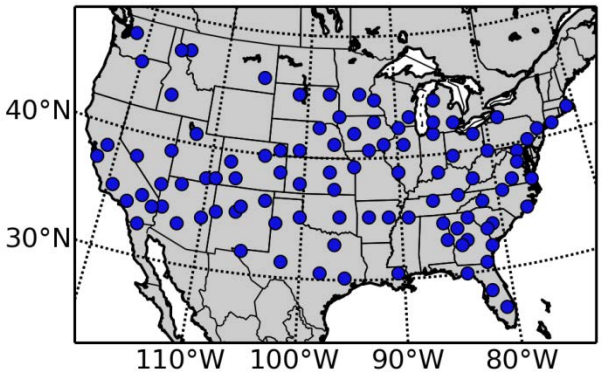

Figure 5: A map of the United States with ground station locations indicated with dots.

For each simulation, increasingly larger sets of nodes are selected, and the results of overhead, delay, and packet delivery ratio are recorded. Each experiment (for a given hello interval and number of nodes) was repeated 10 times for different random samplings of the increasingly larger sets of nodes. The results presented are the mean values from the 10 samples.

Simulations were run on a high-performance machine with 16 Intel Xeon (3.5 GHz) CPUs, featuring a total of 96 GB of RAM. Several samples were simulated in parallel, with each sample requiring anywhere from 6 minutes to 10 hours to simulate. Simulations with the largest number of nodes and a lowest Hello Interval took the longest.

5. RESULTS

Figure 6 shows the mean end-to-end delay as a function of various hello intervals, for an increasing number of nodes.

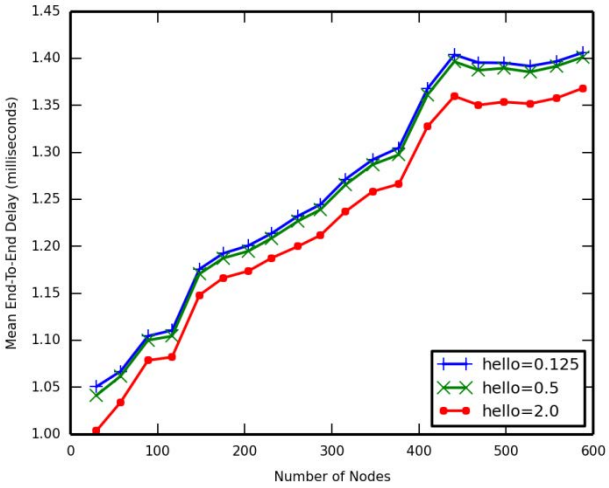

Figure 6: The mean end-to-end delay for various hello intervals.

We consider only transmission, propagation, and queueing delay in our simulation, and employ only a very simple MAC protocol. Therefore, the mean delay values are very small. The delay steadily increases from about 1.0 to about 1.35 milliseconds, and then levels off.

The increase in delay is the result of a larger number of hops being required to route packets. Because each aircraft and ground station have a finite number of links (3), adding a new node to the network can require a node which previously connected directly to the ground station, to now connect through the new node, increasing the end-to-end delay. Using smaller hello intervals causes an increase in the mean delay in the network. It is assumed this is due to increased queueing delays.

Figure 7 displays one measure of the routing overhead of OLSR on our network. In the future, we would like to be able to compare the total routing overhead of OLSR with routing protocols such as DSR, which add data to each packet. Therefore, we chose to measure and plot the aggregate routing overhead. The aggregate routing overhead is simply the sum of sizes (in bytes) of all OLSR overhead packets sent by any node.

The aggregate routing overhead exhibits quadratic growth as the number of nodes increases. This is expected, because OLSR proactively seeks to maintain routes between every pair of nodes in the network. As the number of nodes increases the amount of overhead required to maintain those routing tables increases substantially.

Also as expected, sending hello messages at a higher rate causes an increase in the overhead. The rapid growth in the overhead as the number of aircraft increases means that OLSR will likely not scale well to route packets through our envisioned network of thousands of nodes.

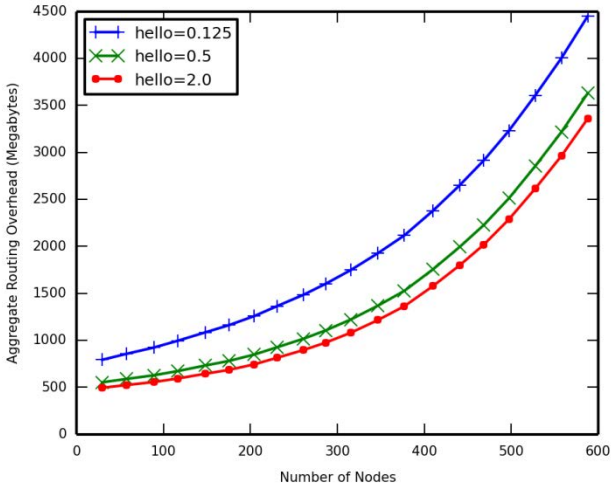

Figure 7: The aggregate routing overhead required to maintain OLSR routes for various hello intervals.

In Figure 8 the packet delivery ratio (PDR) for this experiment is plotted.

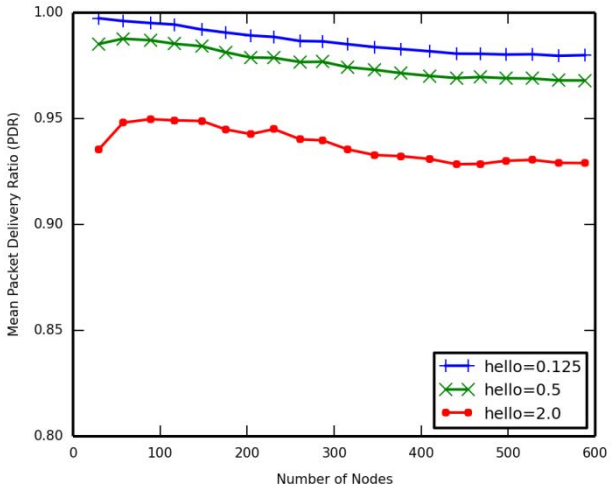

Figure 8: The packet delivery ration (PDR) as the number of nodes increases, for three different hello intervals.

The PDR is the number of data packets received divided by the number of data packets transmitted. The values differ when packets are lost while being routed through the network. This can happen, for example, when the topology of the network changes and a route to a given destination becomes invalid. Packets may be dropped until OLSR can determine a new route to the destination. Higher PDR values for the case where a hello interval of 0.125 is used, indicate that faster hello intervals do yield better results, but with an increase in the overhead cost, as described above. For all three hello intervals the PDR decreases as the number of nodes increases. The intuition here is similar to that described previously in analyzing the delay. With limited numbers of directional links, adding more nodes essentially means increasing the mean number of hops. More hops means a higher probability that an unexpected change will happen along the path between a node and its ground station, yielding a lower PDR. However, with fewer nodes, and, on average, fewer hops to a ground station, the PDR increases.

The quadratic growth of the overhead required by OLSR leads us to believe it will suffer from scalability issues, especially when thousands of nodes are considered. OLSR is likely not a good fit for our airborne network as currently envisioned. This is especially true if end-to-end communication between aircraft is uncommon. If a high percentage of traffic is between a node and a ground station, as is the case for Internet applications, then there is no need to maintain routes between every pair of nodes proactively, and an "on-demand" or geographic routing protocol may be a better fit.

6. FUTURE WORK

We have laid a good foundation for simulating large-scale high-capacity airborne networks in ns-3, but there are several ways we plan to make our simulation more realistic. First we need to fully incorporate our attenuation model, and correctly simulate the bit error rate. Similarly we should allow the alignment map to connect a Net Device to multiple other Net Devices and handle interference.

We plan to introduce a realistic delay to the movement of the steerable directional antennas. This will require the Topology Control Protocol to be proactive about making changes before links fail.

Many of the mobile ad-hoc routing protocol implementations in ns-3 don't support nodes with multiple NetDevices. We hope to add multiple interface support to some of these protocol implementations. This will allow us to make comparisons between the protocols. If, as we speculate, no existing routing protocol is able to efficiently react to the constant changes in the network, we will work to design our own protocol for this type of network.

7. CONCLUSION

We have developed ns-3 models to enable large-scale simulations of airborne networks utilizing steerable directional links. Further, we have demonstrated how these models work by conducting an experiment with the OLSR routing protocol. The results lead us to conclude that OLSR is not a good fit for this Airborne Network. This foundation of simulation tools and models will allow us to make advances in the protocols needed to succeed in this unique networking environment.

8. REFERENCES

[1] R. Devaul, E. Teller, C. Biffle and J. Weaver, *Balloon network with free-space optical communication between super-node balloons and rf communication between super-node and sub-node balloons,* Google Patents, 2013.

[2] E. Teller and W. Patrick, *Balloon clumping to provide bandwidth requested in advance,* Google Patents, 2013.

[3] J. Garside, "Facebook buys UK maker of solar-powered drones to expand internet," *The Guardian,* March 2014.

[4] H. Kelly, "Facebook looks to drones, lasers and satellites for Internet access," *CNN,* March 2014.

[5] M. Wohlsen, "Facebook Drones to Battle Google Balloons in the War of Airborne Internet," *Wired,* March 2014.

[6] D. Medina, F. Hoffmann, F. Rossetto and C.-H. Rokitansky, "A Geographic Routing Strategy for North Atlantic In-Flight Internet Access Via Airborne Mesh Networking," *Networking, IEEE/ACM Transactions on,* vol. 20, no. 4, pp. 1231-1244, August 2012.

[7] Q. Balzano, J. Rzasa, S. Milner, and C. Davis, "High Capacity Tactical Networks with Reconfigerable, Steerable, Narrow-Beam Agile Point-to-Point RF Links," in *Military Communications Conference, 2007. MILCOM 2007. IEEE*, 2007.

[8] B. Epstein and V. Mehta, "Free space optical communications routing performance in highly dynamic airspace environments," in Proceedings of *Aerospace Conference, IEEE*. 2004.

[9] "Unmanned Systems Integrated Roadmap FY2013-2038," United States Department of Defense, reference number 14-S-0553, 2013.

[10] L. Stotts, N. Plasson, T. Martin, D. Young and J. Juarez, "Progress towards reliable free-space optical networks," in *MILCOM 2011*, 2011.

[11] B. Newton, J. Aikat, and K. Jeffay, "Analysis of Topology Algorithms for Commercial Airborne Networks," in *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, 2014.

[12] P. Santi, "Topology Control in Wireless Ad Hoc and Sensor Networks," *ACM Comput. Surv.,* vol. 37, no. 2, pp. 164-194, Jun 2005.

[13] B. Nakhkoob, M. Bilgi, M. Yuksel, and M. Hella, "Multi-transceiver optical wireless spherical structures for MANETs," *Selected Areas in Communications, IEEE Journal on,* vol. 27, no. 9, pp. 1612-1622, December 2009.

[14] B. Nakhkoob, M. Bilgi, M. Yuksel, and M. Hella, "Multi-transceiver Optical Wireless Spherical Structures for MANETs," *IEEE J.Sel. A. Commun.,* vol. 27, no. 9, pp. 1612-1622, December 2009.

[15] A. Tiwari, A. Ganguli, A. Kothari, S. Avadhanam, J. Yadegar, M. Compton, and K. Hopkinson, "Feasibility of communication planning in Airborne Networks using mission information," in *Military Communications Conference, MILCOM 2009. IEEE*, 2009.

[16] A. Tiwari, A. Ganguli, A. Sampath, D. Anderson, B.-H. Shen, N. Krishnamurthi, J. Yadegar, M. Gerla, and D. Krzysiak, "Mobility Aware Routing for the Airborne Network backbone," in *Military Communications Conference, MILCOM 2008. IEEE*, 2008.

[17] A. Kothari, B.-H. Shen, A. Tiwari, A. Ganguli, S. Xu, and D. Krzysiak, "Performance characterization of ad hoc routing protocols with mobility awareness," in *MILCOM 2010*, 2010.

[18] T. Clausen and P. Jacquet, Eds., *Optimized Link State Routing Protocol (OLSR),* United States: RFC Editor, 2003.

[19] D. Medina, F. Hoffmann, F. Rossetto, and C.-H. Rokitansky, "A Crosslayer Geographic Routing Algorithm for the Airborne Internet," *2010 IEEE International Conference on Communications (ICC)*, 2010.

[20] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2000.

[21] "Aircraft Situation Display To Industry: Functional Description and Interface Control Document for the XML version," U.S. Federal Aviation Administration, Egg Harbor Township, NJ, 2011.

[22] *Automatic Dependent Surveillance-Broadcast (ADS-B) Out Performance Requirements To Support Air Traffic Control (ATC) Service; Final Rule,* 2010.

[23] K. Miyao, H. Nakayama, N. Ansari, and N. Kato, "LTRT: An efficient and reliable topology control algorithm for ad-hoc networks," *Wireless Communications, IEEE Transactions on,* vol. 8, no. 12, pp. 6050-6058, 2009.

[24] L. B. Stotts, P. Kolodzy, A. Pike, B. Graves, D. Dougherty, and J. Douglass, "Free-space optical communications link budget estimation," *Appl. Opt.,* vol. 49, no. 28, pp. 5333-5343, October 2010.