On Distributed, Geographic-Based Packet Routing for LEO Satellite Networks

Thomas R. Henderson
Geocast Network Systems, Inc.
190 Independence Drive
Menlo Park, CA 94025

Randy H. Katz
Electrical Engineering and Computer Science Dept.
University of California, Berkeley
Berkeley, CA 94720

Abstract— **Advances in satellite technology are enabling the deployment of large constellations of Low-Earth-Orbiting (LEO) satellites. Next-generation systems will be tailored for broadband, packet-switched services, and therefore require either distributed or centralized packet routing mechanisms. Some researchers have hypothesized that the semi-regular mesh topology of a polar-orbiting constellation admits a simple distributed routing protocol based on using geographic information embedded in the node address. In this paper, we take a closer look at this hypothesis in the context of commercially-proposed constellation designs. Using simulation, we study a distributed routing protocol that selects the next hop based on a minimization of the remaining distance to the destination. Our numerical results indicate that this routing strategy usually yields good routes, with an average latency degradation of less than 10 ms when compared with the optimal route. However, there are locations in the topology, most notably around the counter-rotating seams, the polar regions, and close to the destination of a packet, where the assumption of a regular mesh topology breaks down and it is difficult to guarantee robustness without adding significant additional complexity to the protocol.**

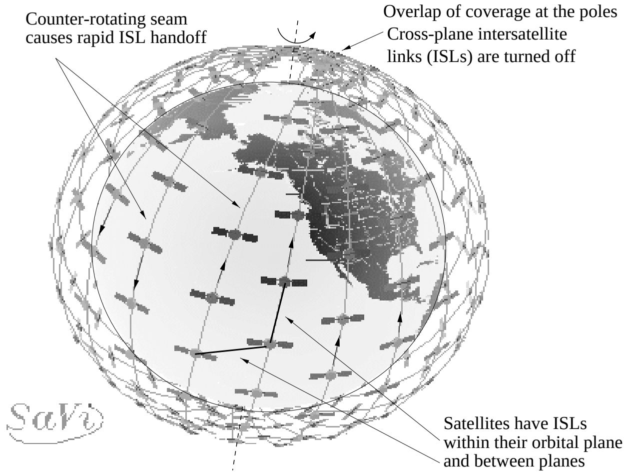

Fig. 1. Example of a polar-orbiting satellite constellation. The figure was generated using the SaVi software developed by the Geometry Center at the University of Minnesota.

I. Introduction

Low-Earth-Orbiting (LEO) satellite constellations have begun to be deployed for voice and narrowband data services. Although more complicated to design and maintain than satellites positioned at geostationary (GEO) orbit, LEO satellite networks offer the potential for smaller earth terminals (requiring less transmit power and antenna gain), lower communications latency, and frequency reuse. Future LEO networks will likely migrate to offering broadband data services based on packet-switching and offer global connectivity based on a network of intersatellite communications links (ISLs).

The LEO network can be viewed as a special type of mobile network, one in which the nodes move with respect to the fixed users. Despite its time-varying nature, the semi-regular mesh topology of polar constellation configurations has led researchers to investigate the potential for exploiting the topology to simplify packet routing. Namely, by using geographic-based addresses, the hypothesis put forward is that a simple distributed routing protocol that directs satellites to route packets in the direction that most reduces the remaining distance to the destination can yield routes that are close to optimal in terms of end-to-end latency.

In this paper, we report on a simulation study that has attempted to confirm the above hypothesis. We found that, in general, such a routing strategy does admit good performing routes, with average latency degradations of less than 10 ms (although with some instances no more than 60 ms worse) for commercially-proposed constellations. However, we found construction of a *robust* protocol based on this strategy difficult, because of the breakdown of simplifying assumptions at several locations in the topology. In particular, around the destination, around longitudinal regions where the orbital planes are counter-rotating, and near the polar regions, routing "dead-ends" are frequently encountered, and compensating for these breakdowns leads to significant additional protocol complexity. We have concluded that the complexity required to overcome the topological irregularities make this type of distributed routing strategy a much less attractive proposition than may have been previously thought.

II. Background and Related Work

Space limitations herein preclude a detailed overview of LEO constellation designs. The book by Pattan [1] and Wood's thesis [2] provide good overviews. More detail on the methodology and results of this paper can be found in [3].

Figure 1 illustrates the type of constellation we consider herein: the *polar orbiting* constellation, used by Iridium and proposed for Teledesic. In polar orbiting constellations, satellites are deployed in circular planes with an inclination angle close to 90°. We assume that, in general, more than one satellite may be above a given user's elevation mask. Each satellite is equipped with an antenna system capable of directed coverage of portions of the Earth's surface. To obtain higher system capacity, the antenna system incorporates frequency reuse via decomposition of the coverage area into a number of smaller spot beams. At an altitude on the order of one thousand kilometers, the satellites orbit the Earth roughly every two hours, so that continuous coverage requires link handoff between terminals and satellites. We also assume that satellites have communications links to up to four (Irid-

---

This work was supported by DARPA contract DAAB07-95-C-D154, by the State of California MICRO program, and by Hughes Aircraft, Metricom, Fuji Xerox, Daimler-Benz, and IBM.

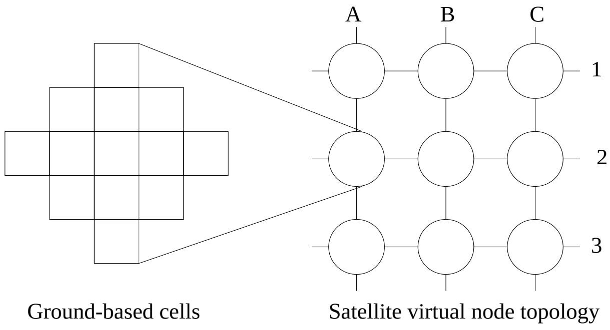

Fig. 2. A logical network topology: fixed zones on the Earth's surface are assigned a logical address, and a satellite serving a particular zone embodies the logical node serving that region (derived from [5]).

ium) or eight (Teledesic) neighboring satellites. Note also the presence of a counter-rotating "seam" that occurs in two longitudinal regions where the direction of revolution of neighboring planes is opposite. Communications links between satellites are more difficult to maintain across these seams.

Consider the implementation of a distributed packet routing protocol in such a network. Because of the many handoffs occuring continuously, routing must be dynamic and adjust quickly. However, the routing problem may be simplified conceptually by exploiting a level of indirection. If one decomposes the Earth's surface into a regular cellular structure, and assigns a logical address to the satellite node serving a particular cell, then different physical satellites can, over time, embody a fixed logical node. A natural addressing mechanism is the assignment of addresses based on geographic location; this was proposed by Shacham over ten years ago [4].

Figure 2, derived from a paper by Mauger and Rosenberg [5], illustrates this concept of a virtual node. Carried to the extreme, a static logical network can be defined and no dynamic routing need be performed. However, this extreme case implies a one-to-one mapping between terminals in a given cell and the current satellite serving the cell, which will lead to a decrease in system availability for the following reasons. First, terminals at the very edges of these fixed footprints may often find that they could receive better coverage from the satellite serving the neighboring footprint than from the satellite to which they are forced to connect. Second, there will be occasions when the satellite serving a fixed footprint will be in the same line of sight as the sun and communications are impossible (this is known as a *sun outage*) unless alternate satellites are used. Third, since user density is highly non-uniform around the globe, it will be advantageous for neighboring satellites to train additional spot beams on regions of high density (although we do not consider such system optimizations herein). Once the one-to-one correspondence between terminal and satellite virtual node is broken, some type of dynamic routing becomes necessary. Nevertheless, the satellite virtual node concept is useful if one relaxes the constraint that the footprints be fixed on the Earth's surface. If a satellite footprint can be decomposed into multiple smaller cells, then "semi-fixed" footprints (fixed for some finite amount of time before a handoff is needed) can be composed of these smaller cells such that system availability is maximized (i.e., the boundaries of the Earth-fixed footprints can dynamically change as needed).

Recently, there have been two papers that attempted to develop this concept further. Hashimoto and Sarikaya also suggest using geographic information embedded in addresses to perform distributed packet routing [6], and propose a basic routing protocol. However, they do not validate the correctness of their proposed protocol or discuss in detail how it handles a non-idealized topology. Ekici, Akyildiz, and Bender analytically developed a similar datagram routing algorithm for an idealized polar orbiting constellation [7]. Their two-phase algorithm includes a first step that identifies next hop satellites that move the packet one hop closer to the destination, and a second step that determines which of the candidate next hops most reduces the remaining distance to the destination.

In the remainder of this paper, we describe our efforts to apply this general routing strategy to two commercially-proposed constellations: the 288-satellite Teledesic constellation and a broadband version of the 66-satellite Iridium constellation.[1] As we shall illustrate, we found that such commercially-proposed constellations presented subtle difficulties that impair the ability to define a simple yet robust protocol based on these principles.

III. CONSTRUCTION OF A DISTRIBUTED ROUTING PROTOCOL

Performing packet routing by using geographic information embedded in the addresses is based on the hypothesis that, in a LEO system with a regular mesh topology, a series of locally optimal forwarding decisions (namely, routing to the neighboring satellite that most reduces the distance to the destination) will yield a route that is close to optimal when compared with the globally optimal route. Each forwarding decision is based on reducing some measure of the distance to the destination: a satellite with a packet to route first determines its distance to the destination, and then determines the distance from each of its immediate neighboring satellites to the destination. It is assumed that location information for a satellite and its immediate neighbors is readily available, and that distances can either be computed on-demand or looked up in a table. A satellite then routes a packet to the neighboring satellite that most reduces the distance to the destination.

We implemented the basic protocol in the *ns* simulator.[2] Details of the simulation methodology can be found in [3], [8]. Specifically, we assumed that each satellite knew its own location and those of its connected neighbors. When a satellite received a packet for a destination terminal that it did not serve, it computed the great-circle distance from the center of its current cell to the center of the destination cell, and likewise computed the distance from all of its neighboring satellites to the destination. If one or more neighboring satellites had a smaller distance to the destination, the satellite forwarded the packet to the satellite that most reduced the distance to the destination; otherwise, the packet was dropped.

We evaluated the routing protocol performance using the following approach: we repeatedly picked two points on the globe at random, and tried to route two packets between them. The first packet was routed using a global shortest-path algorithm based on minimization of

---

[1] The assumed Teledesic design is the 288-satellite Boeing design; subject to change. Also, the current Iridium system does not use cross-seam ISLs. In this paper we assume their use.

[2] http://www.isi.edu/nsnam/ns/

the propagation delay of the route. The second packet was routed via the distributed protocol. We were interested in two performance metrics: the *robustness*, as measured by the ability to avoid routing "dead-ends" (and hence packet drops), and the *delay degradation* of the geographically-based route as compared with the optimal route. We therefore calculated the delay experienced by both packets if the routing was successful for both packets, and noted any routing failures for packets using the distributed routing protocol (the packets routed by using globally-optimal shortest paths were never dropped). We chose to simulate a large set of random points (10,000 per trial) rather than use an exhaustive combinatorial search because the latter would have been computationally infeasible. Nevertheless, as we show below, using a large number of random trials was sufficient for evaluation purposes because it exposed a number of weaknesses in the approach.

As we describe in the following three subsections, we encountered a number of difficulties in achieving a robust protocol. First, in a polar-orbiting constellation, geographic routing frequently breaks down very near a destination. Second, in the polar regions, the regular mesh topology is disrupted, again leading to dead-ends. Finally, at the counter-rotating planes, the geometry of the orbits can cause a tear in the mesh topology. The next three subsections describe our efforts to engineer around these problems.

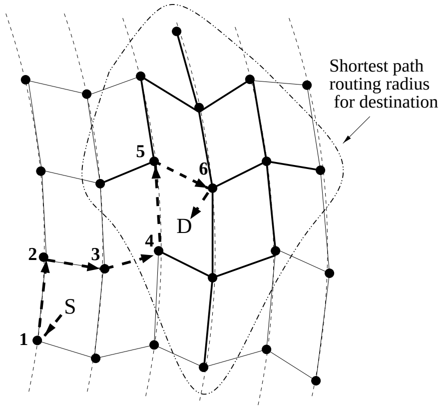

Fig. 3. Hybrid routing strategy based on geographic packet forwarding for distant destinations and locally-scoped shortest path routing for local destinations. The figure denotes a subgraph of the satellite mesh and a hypothetical packet trace. A packet sourced at S is forwarded based on geographic information to the satellite numbered 4. Satellites use shortest-path routing information to complete the routing to destination D, which is served by satellite 6.

A. Locally Scoped Shortest Path

In a perfectly regular mesh topology in which destination terminals were always connected to the closest satellite, geographic-based packet forwarding would never result in a dead-end. However, since LEO satellites typically have overlapping footprints, the geographic forwarding may break down, as can be seen by the example shown in Figure 3. In the figure, a packet routed from S (connected to satellite 1) to D (served by satellite 6) proceeds via geographic routing to satellite 4. At this point, however, satellite 4 cannot route the packet to any of its neighboring satellites without increasing the distance to the destination. By forwarding to a satellite that increases the distance to the destination, we open the possibility for a routing loop to be formed, and although techniques can be used to prevent packets from being forwarded back to a previously visited node (such as encoding the history of the traversed route in the packet header), we still cannot guarantee that a packet so forwarded will eventually find the right egress node.

Our solution was to use a locally-scoped shortest path algorithm to complete the packet forwarding process close to the destination. We implemented a basic link-state routing algorithm such as is described in [9]. Instead of flooding each link state packet (LSP) to every node, however, we flooded an LSP only as far as the routing radius for a given satellite. The routing radius was determined such that it covered every possible satellite that could potentially serve the destination. The flooding protocol makes use of packet numbers to suppress transmission of duplicates. Each satellite therefore had a map of a subgraph centered on itself. As an example, Figure 3 illustrates the case for which the routing radius is two hops, and the dashed boundary around satellite 6 denotes those links and nodes that are used in satellite 6's routing computations. The protocol therefore requires a hybrid approach that uses geographic-based packet forwarding to get a packet in the vicinity of a destination, and shortest path routing to finish the final few hops to the destination. Such a solution is also recognized by Mauger and Rosenberg [5], in that the authors propose to resolve the inherent last-hop ambiguity around a destination by flooding this connectivity information with neighboring satellites.

Let us discuss the robustness and complexity of this approach. In general, routing loops can form whenever nodes make routing decisions based on inconsistent information. In this case, since all routing information is locally-scoped, each node has a slightly different view of the network topology, which can lead to the following problems. First, if different nodes have different routing radii, it may be possible for stale routing information to persist. Second, we must prevent the occurrence of routing loops that could form if a packet enters a locally-scoped routing radius of a destination and is somehow subsequently forwarded to a satellite outside the routing radius. Third, it is well known that if different nodes use different routing metrics (such as dynamically adapting to congestion based on local information), loops are possible. This last problem is a general dynamic routing problem and can be avoided by making sure that all nodes use the same routing metric and have up-to-date link costs.

The key to avoiding such routing loops is for each node, when constructing a path, to consider the routing radii of all of the nodes along the path, and to ensure that stale routing information is successfully purged from each node. The first goal can be realized by requiring satellites to advertise their own routing radius in their LSPs. Furthermore, we modified the shortest path algorithm to construct complete paths to the destination and to check whether the satellite constructing such a path is within the routing radius of all nodes in the path. With this approach, we still must make sure stale information is purged from the system. LSP updates will naturally purge stale information, except if a node dynamically decreases its routing radius. In this case, the node needs to make sure that its old LSPs are expunged from all nodes at the periphery of its routing radius.

As for complexity, although this approach requires implementation of a shortest-path protocol, the processing and memory overhead is significantly reduced by scoping

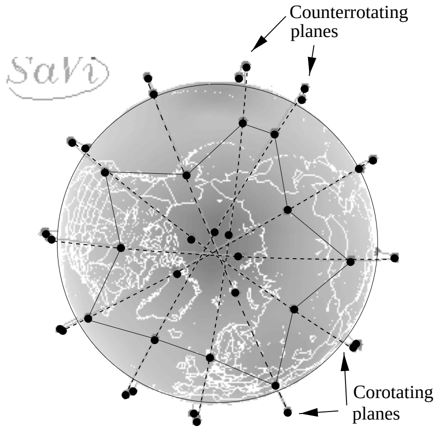

Fig. 4. View of the Iridium topology above the North pole. Satellites closest to the pole have interplane ISLs turned off. The "polar region" is bounded by the set of satellites closest to the pole that have all of their interplane ISLs active.

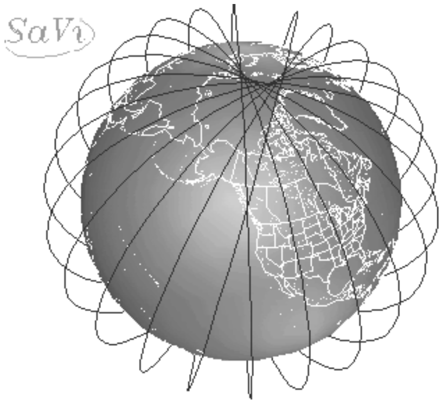

Fig. 5. Illustration of the intersection of counter-rotating planes.

the LSP propagation to a small region around each satellite. Our modifications to the shortest path algorithm discussed above do not significantly increase its complexity.

B. Routing in Polar Regions

As stated above, the routing radius is defined as including all those satellites that can be observed above the elevation mask of a terminal. In addition, the radius must be extended whenever there are breaks in the topology. In the high latitudes, the interplane ISLs must be deactivated, and for a packet to reach a satellite that has its interplane ISLs deactivated, the packet must first be routed to a satellite in the same plane but at a lower latitude. As a result, geographic-based packet forwarding can break down several hops away from the eventual destination. This implies that we should increase the routing radius such that all satellites in the polar region can obtain LSPs for all other satellites in the polar region. However, such a radius is sufficiently large (five or six hops in our simulations) that it would spill over significantly into the lower latitudes, increasing the amount of routing state required on each satellite (the amount of routing state required grows roughly quadratically with each hop). To compensate for this, we developed a special routing zone for the polar regions that specifically limited the scope of polar-area routing information to the polar region.

The key is to properly define and dynamically identify the polar region, and to make sure that all nodes in the polar region have routing state information about each other. Figure 4 illustrates a view of the polar region from directly above the rotation axis of the Earth, in which satellites near the poles do not have their interplane ISLs turned on, while satellites at lower latitudes do have interplane ISLs. The Iridium topology, with an orbital inclination of 86.4 °, is plotted. We define the polar region as including all satellites that have one or more interplane ISLs turned off (the POLAR satellites), as well as all satellites that border the POLAR satellites (the POLAR_BORDER satellites). If we define a third state (LOW_LATITUDE) that includes all other satellites, it is easy for each satellite to determine which state it is in by simply examining the state information of its neighboring intraplane satellites. Satellites can propagate state information to their polar neighbors using the same protocol as for propagating LSPs. The key, then, is to extend the scope of LSP propagation of a satellite to the entire polar region in addition to the normal routing radius. Any packets directed toward a destination in the polar region will eventually find a satellite in this polar region, and then shortest path routing can take over. Satellites expunge this extra state information when they leave the polar region, and announce their departure to the remainder of the polar region so their LSPs can be expunged from the rest of the polar satellites.

We also used this state information to "tunnel" packets to outside of the routing radius. When sending a packet out of the polar region, the satellite should use the location information of the two POLAR_BORDER satellites in computing the forwarding direction, instead of the location of the immediately neighboring satellites. This location information can be provided to the POLAR satellites for such computations.

Although constructing a special polar routing radius increases the amount of state kept by satellites at higher latitudes, and accounts for a increased message overhead in that region, this increase is somewhat offset by the fact that the normal traffic density in polar regions is likely to be extremely light.

C. Problems at the Seams

Although handling the polar regions and the regions around the destinations required additional protocol mechanisms, we were able to eliminate routing dead-ends in our experiments. A third complication, however, presented more of a challenge. As mentioned above, the counter-rotating planes in a polar constellation form a "seam." It is possible to establish ISLs across this seam, although the link acquisition and synchronization associated with these ISLs are much more difficult than with interplane ISLs. However, the mesh is distorted in this region. First, (in the proposed Teledesic constellation) there is only one ISL per satellite across the seam, since the second ISL will be used to acquire the next satellite before handover occurs. Therefore there is a paucity of links available in this region. A more significant problem, however, is that the (non-polar) inclination angle of the orbital planes causes the two counter-rotating planes to intersect at a much lower latitude than the other planes. This effect is clearly visible in Figure 5 for Teledesic

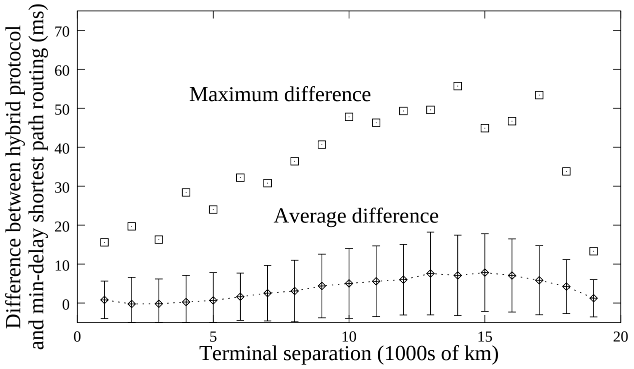

Fig. 6. Average and maximum delay difference between using geographic forwarding and minimum-delay shortest path routing as a function of terminal separation (Teledesic constellation). Error bars denote one sample standard deviation from the sample mean.

(which plans an inclination angle of 84.7 °, where the two planes intersect at a latitude of approximately 54 °. If we let i denote the inclination angle of the orbital planes, and s denote the spacing between planes, then the latitude at which the cross-seam planes intersect is given by $\arctan(\sin(s/2) * \tan(i))$. As a result, the cross-seam ISLs must be switched off at a relatively low latitude, which causes a tear in the ISL connection mesh (and causes routing dead-ends for packets attempting to cross this seam). Regardless of whether geographic forwarding is used or not, this appears to be a drawback to deploying satellites in planes that deviate from 90 ° inclination. However, launching satellites into a purely polar orbital plane is considered to be prohibitively expensive.

Although we tried various techniques (all based on distributed protocols) to tunnel around this tear in the topology, we were not successful in finding one that was reasonably simple to implement. Even when we constructed tunnels around these tears in the topology, we could always find cases for which the hybrid routing protocol faced a dead-end. These routing failures are likely to persist, at least intermittently, for as long as the seam separates the two endpoints (which could be hours). We note also that similar dead-ends are likely to occur when there are other tears in the topology due to satellite failures, which we did not investigate. In summary, we were not successful in guaranteeing the robustness of a geographic-based routing in the presence of a counter-rotating seam for the Teledesic and Iridium (with cross-seam ISLs) constellation topologies.

IV. PERFORMANCE

Despite the routing breakdowns due to the counter-rotating planes, we did find that, on average, the delay performance of our hybrid protocol was adequate. Figure 6 plots the average and maximum delay differences between geographic-based forwarding and min-delay shortest path routing for the Teledesic constellation. The data is drawn from an experiment of 10,000 random terminal locations. Two cases were run with the same set of terminals: the hybrid routing protocol described above and global min-delay shortest path routing. We then took the results from the hybrid protocol and computed the delay difference, point-by-point, between that protocol and the shortest path protocol. We have collated the data points into 1000 km bins before performing the averages. The main points to consider are those above 5000 km, for those are the ones for which a packet must traverse one or more geographic forwarding hops before hitting the shortest path routing radius. In addition to the averages, we tracked the maximum delay difference (penalty) due to using the geographic-based protocol. We note from the figures that, on average, the geographic routing is less than 10 ms worse than min-delay shortest path routing. Such an increase in average delay would probably not be considered significant to LEO network users. However, the maximum delay differences can be very large (up to 60 ms), and are from a small set of outliers. These points occur near the poles when the geographic routing initially brings the packet close to the destination in terms of distance, but far away from it in terms of topology, and it consequently must be routed back towards the particular orbital plane containing the satellite serving the destination.

V. SUMMARY

We have described some practical difficulties in constructing a distributed packet routing protocol based on minimization of geographic distance. Although the delay performance of the hybrid routing protocol that we designed was adequate, the robustness in terms of avoidance of routing failures was not, due to the following complications: i) the ambiguity in last-hop satellite coverage near the destination, ii) the disruption of the regular mesh structure in the polar regions, and iii) the tear in the mesh topology located around counter-rotating planes when the inclination angle is less than 90 °. Furthermore, we did not even consider the possibility of node failures, which would burden the protocol further. For these reasons, we have concluded that, for polar-orbiting constellations, basing a distributed routing protocol on geographic forwarding is prone to either failure modes or high complexity.

ACKNOWLEDGMENT

Marie-Jose Montpetit and Don Hoffman of Teledesic both were very helpful in sharing proposed constellation parameters with us.

REFERENCES

[1] B. Pattan, *Satellite-Based Cellular Communications*, McGraw Hill, 1998.
[2] L. Wood, *Network Performance of Non-Geostationary Constellations Equipped with Intersatellite Links*, Masters thesis, University of Surrey, 1995.
[3] T. Henderson, *Networking over Next-Generation Satellite Systems*, Ph.D. Dissertation, University of California, Berkeley, 1999.
[4] N. Shacham, “Protocols for Multi-Satellite Networks,” *Proceedings of IEEE MILCOM ’88*, pp. 501–505, 1988.
[5] R. Mauger, “QoS Guarantees for Multimedia Services on a TDMA-Based Satellite Network,” *IEEE Communications Magazine*, vol. 35, no. 7, pp. 56–65, July 1997.
[6] Y. Hashimoto and B. Sarikaya, “Design of IP-based Routing in a LEO Satellite Network,” *Proceedings of Third International Workshop on Satellite-Based Information Services (WOSBIS ‘98)*, pp. 81–88, 1998.
[7] E. Ekici, I. Akyildiz, and M. Bender, “Datagram Routing Algorithm for LEO Satellite Networks,” *Proceedings of INFOCOM 2000, Tel Aviv*, 2000.
[8] T. Henderson and R. Katz, “Network Simulation for LEO Satellite Networks,” *Proceedings of 18th International Communication Satellite Systems Conference, Oakland, CA*, Apr. 2000.
[9] R. Perlman, *Interconnections: Bridges and Routers*, Addison Wesley, 1992.