

# Simulation of Two Routing Protocols Operating in a Low Earth Orbit Satellite Network Environment

Richard A. Raines, Richard F. Janoso, David M. Gallagher, and David L. Coulliette  
Air Force Institute of Technology, Wright-Patterson AFB, OH

**Abstract**—Modern warfare is placing an increasing reliance on global communications. Currently under development are several Low Earth Orbit (LEO) satellite systems that propose to deliver voice and data traffic to subscribers anywhere on the globe. However, very little is known about the performance of conventional routing protocols under orbital conditions where the topology changes in minutes rather than days.

This paper compares two routing protocols in a LEO environment. One (Extended Bellman-Ford) is a conventional terrestrial routing protocol, while the other (Darting) is a new protocol which has been proposed as suitable for use in LEO networks. These protocols are compared via computer simulation in two of the proposed LEO systems (Globalstar and Iridium), under various traffic intensities. Comparative measures of packet delay, convergence speed, and protocol overhead are made.

It is found that while both protocols have roughly equivalent end-to-end delay characteristics, Darting requires a much higher overhead. Darting also demonstrates higher instability at network update periods. Darting is handicapped by a strong correlation between data traffic and protocol overhead. Modifications to reduce this overhead would result in better performance.

**Keywords**—Low Earth Orbit, Routing, Simulation, Bellman-Ford, Darting

## I. INTRODUCTION

The lure of ubiquitous global communications has led to several large-scale commercial efforts in the US and abroad to loft multi-satellite networks in Low Earth Orbit (LEO). These systems are designed to provide mobile communications from any point on the earth. In order to reduce propagation delay, these systems attempt to minimize the number of links between the ground and space based portions of the network. Towards this end, several of the projects intend to use inter-satellite links to route the circuits completely in the orbital environment. The large velocity of the satellites with respect to the ground and one another, however, gives rise to high demands upon the circuit setup routine as it attempts to find the best path through a continuously changing network topology.

The first section of this paper discusses some of the background and supporting work that has been done in this area, elaborating on some of the challenges of designing a network protocol for orbital use. The second section describes the simulation setup that was used to compare two specific routing protocols in this environment. The final section discusses the results of those comparisons.

## II. BACKGROUND

Several studies have been done to determine constellations of LEO satellites that can ensure global coverage using an arbitrary number of satellites. Walker [10] was among the first to propose such a system, and several commercial ventures have adopted the Walker "delta" network. Walker constellations consist of several planes of inclined orbits with multiple satellites per plane. The system is

described by six parameters that are chosen by the system designer to produce the desired degree of network coverage. For example, Globalstar (one of the leading LEO contenders) proposes to construct their constellation in a 48/8/1/52°/1389 format. This means that there will be 48 satellites in 8 different orbital planes, with a 7.5° phase shift between the planes. The orbital planes themselves will have an inclination of 52 degrees and the satellites will orbit at 1389 kilometers. For Globalstar, these numbers ensure that at latitudes below 55°, there is at least one satellite visible at an elevation of 40 degrees or greater [7]. The Globalstar constellation can be seen in Figure 1.

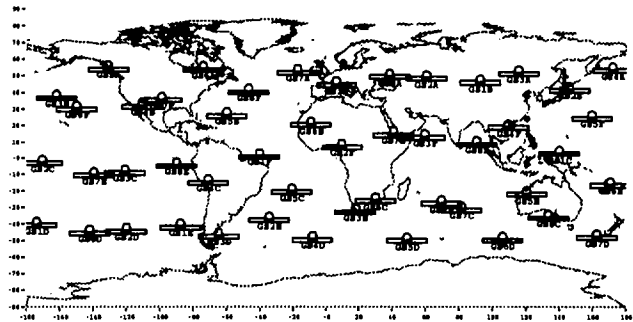


Fig. 1. SatLab Globalstar Constellation

Another popular family of constellations is based upon the work of Adams and Rider [1] who proposed the use of polar orbits to provide n-redundant global coverage. This is the orbital family used by Iridium, another leading network proposal. The constellation is based upon the idea of a "street of coverage" provided by each orbital plane. The designer can make any desired number of satellites visible from an arbitrary spot on the earth by simply "narrowing" the street (bringing the orbits closer together). Current plans for Iridium call for six 86.4° inclination orbits of 11 satellites each at an altitude of 780km [9]. Minimum viewing elevation provided by this configuration would be about 10 degrees [6]. The Iridium constellation can be seen in Figure 2.

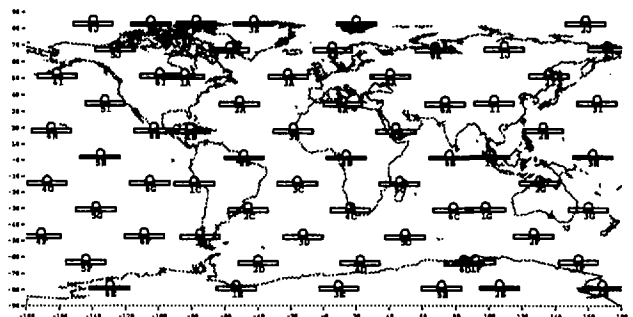


Fig. 2. SatLab Iridium Constellation

Each of the commercial efforts currently underway has plans for some type of proprietary routing protocol to address the questions of orbital routing, but published literature on these techniques is surprisingly sparse. Conventional flooding-type routing algorithms are not well suited to the orbital environment due to the large number of overhead messages they generate. This, coupled with long link propagation times, may result in transient loops forming in the network while a topology update is in progress. Gross and Ziemer [4] have investigated the performance of conventional Ford/Fulkerson and Merlin/Segall routing algorithms in satellite communication (SATCOM) networks.

Ford/Fulkerson belongs to the class of Distance Vector algorithms and operates by having each node maintain a table of costs for all destinations reachable through its outgoing links. Upon detecting a change in one of these links, a node will send a control message to each of its neighbors. These neighbors will in turn update their own internal tables with the new information and pass the update farther along the network. The advantages of this method are its simplicity and asynchronicity. Disadvantages are slow convergence time and susceptibility to looping.

In contrast, the Merlin/Segall method is designed to prevent formation of transient loops during a network update, but it does so at the expense of a slower convergence rate. In Merlin/Segall, all paths are stored as directed trees rooted at the sink node, which prevents loop formation due to the acyclic nature of trees. Updates begin at the sink node and propagate up-tree until the farthest node hears from all its neighbors. This farthest-distance estimate is then returned down-tree, with each node updating its shortest-path entry upon reception of the return packet. Multiple update cycles may be required in the presence of network node failures.

The authors compared each algorithm on a hybrid LEO/GEO network of 18 and 6 nodes respectively. They found that the extra loop-preventing overhead introduced in Merlin/Segall led to performance an order of magnitude slower than the Ford/Fulkerson method.

A drawback of the preceding methods is the relatively high overhead associated with the control traffic. Tsai and Ma [8] present a novel approach that they term "Darting" to overcome the high message overhead involved with flooding-type algorithms. The key idea behind Darting is to postpone transmission of topology update messages until it becomes necessary to actually transmit a data message.

Darting uses two update mechanisms, which are triggered with the presence of a data packet. One mechanism updates the downstream nodes (i.e. "successor" nodes that the data packet will be visiting shortly in the future) and the other mechanism updates the upstream nodes ("predecessor" nodes.) The predecessor mechanism is triggered when the local node detects a discrepancy in topology views between itself and its immediate predecessor. Successor updates are carried out by embedding all recent local topology changes in the outgoing message, with provisions to eliminate loops.

Unlike conventional flooding algorithms, which ex-

change periodic control messages to prevent the formation of "message traps" (i.e. routing loops), Darting concentrates on dynamically breaking any traps that have formed. This eliminates the need to exchange update messages on a regular basis. When a source node desires to transmit a packet to a neighbor, it consults its routing tables, and places the anticipated cost of delivering the message to the destination in a header field. Upon receiving the packet, the neighbor node can use this data to determine if the sender is using current routing information. If a discrepancy is detected, the predecessor update mechanism is invoked. Normally this mechanism only updates the immediate predecessor. Optionally, predecessor update messages can be allowed to propagate farther back upstream before being discarded. This will result in faster network convergence at the expense of additional control traffic overhead.

### III. SIMULATION MODEL

Cadence Software's BONEs Designer network simulation and SatLab satellite modeling packages were used to simulate the networks in this article. It was decided to compare the performance of Darting to the Extended Bellman-Ford protocol proposed by Cheng, et. al. [2], which is a modified version of the Distance Vector protocol originally used in the Internet. These protocols were simulated on the Globalstar and Iridium constellations mentioned earlier. Although the actual Globalstar system does not propose to use space-based crosslinks for routing, it is a convenient and well-known example of the Walker type of constellation.

Using an assumption of uniform source distribution, traffic was injected into the network from eight equally spaced groundstations (one in each octant of the globe). However, this rough granularity of source-destination pairs led to intermittent problems in Darting's convergence algorithm, necessitating an extension to the original protocol (see Appendix A.) The extension created a new type of overhead packet used to probe areas of uncertainty in the network during convergence times, and increased the amount of overhead required for Darting to operate.

Data was collected for each protocol on packet traversal time, protocol convergence rate, and protocol overhead. Satellite positions were updated every 60 seconds and these parameters computed for each update period. Several independent simulation runs were made for each configuration to improve the confidence level in the results. Each protocol was simulated at several light to moderate network loads (1%, 10% and 20% of the Iridium maximum ground-space data rate of 12.5 Mbps [3].)

Packet traversal time was selected because the main benefit of LEO networks is the ability to conduct "real time" (less than 400ms) transactions. Thus the impact of each protocol on traversal time is of great interest.

Convergence rate was studied because it is a dominant factor in determining how efficiently a protocol moves data through the network. Protocols that determine the optimal paths faster have a significant edge in providing lower average traversal times in the dynamic LEO environment.

Overhead is directly related to power, which is a critical resource in the orbital environment. Every bit of overhead required to transmit data through the network is wasted power. Therefore the amount of overhead load inflicted on the network by each protocol is of interest. It determines how much of a penalty is imposed by each algorithm in obtaining its convergence rates and traversal times.

The simulation measures packet traversal time by accumulating all calculated delays encountered by a packet in one of the packet's fields. Each time the packet encounters a queuing, transmission, processing, or propagation delay, the duration of the delay is added to the delay field. When the packet reaches its destination, this value is recorded for later analysis.

Similarly, overhead is measured by maintaining a packet length field, and accumulating any additional overhead (if any) added by each node the packet passes through. When the packet reaches its destination, the simulation records the number of data bits and the number of overhead bits contained in each packet. Packets that contain only routing update information are counted as 100% overhead.

Convergence rate is determined by monitoring the network for the presence of routing update packets. A subsection of the simulation measures the time elapsed between delivery of update packets to their destinations. A memory variable local to the analysis section is used to accumulate these times. The convergence time for each protocol is then taken to be the total elapsed time from the last satellite position update to the most recent delivery of an update packet. In other words, each protocol is considered to have converged when no more update packets are present in the network.

The satellite communication simulation provided with Designer and SatLab was modified for use in the comparison. A representative top level diagram can be seen in Figure 3.

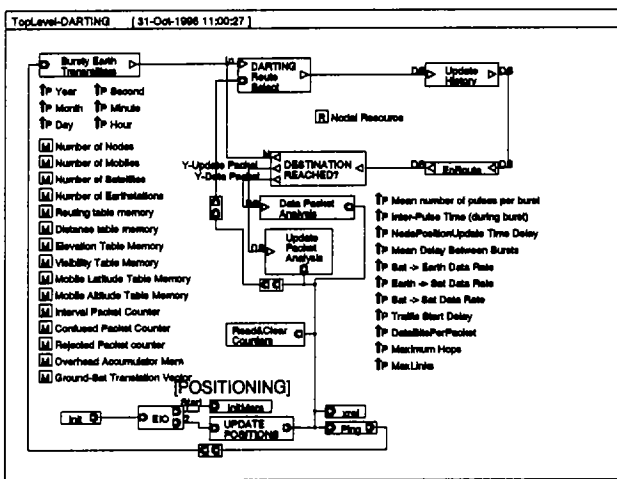


Fig. 3. Top-Level Darting Simulation Schematic

#### IV. RESULTS

The memory requirements to execute the simulations were surprisingly non-linear, severely curtailing the num-

ber of repetitions completed at the higher data loads. The original goal was to complete enough independent repetitions to ensure 99% confidence of accuracy to within 1% of the true population mean. Darting exacerbated this problem by displaying a highly variant response on convergence speed.

The simulations yielded surprisingly disappointing results for Darting, which may in some part be attributable to the modification mentioned previously. While steady-state packet traversal times of the two protocols ranged from 110 to 115 milliseconds, during satellite update periods Darting displayed significantly degraded performance (450-500 millisecond delays).

Surprisingly, Darting turned in worse overhead performance than Bellman-Ford in all but one of the comparisons. Because Darting embeds link status information into passing data packets, Darting's overhead is closely correlated to traffic intensity. Only at the lowest traffic load did it provide equivalent or better performance than Bellman-Ford. Figure 4 and Figure 5 show the overhead performance of each protocol. Because of this correlation, it turns out to be extremely important to tune the link status parameters in Darting for optimal size. This simulation used 32 bit integers to encode the cost for traversing each link in the network. Reducing that to 16 bits might almost halve Darting's overhead.

The single point anomaly seen in the Iridium Bellman-Ford results at 420 seconds is most likely due to a short-term routing loop. According to Cheng's work [2], short-term routing loops can occur in the basic Extended Bellman-Ford protocol when several links fail in the network simultaneously. This is indeed the case at the 420 second point in the Iridium constellation. Cheng proposes a solution to the problem in his article, but this was not implemented here.

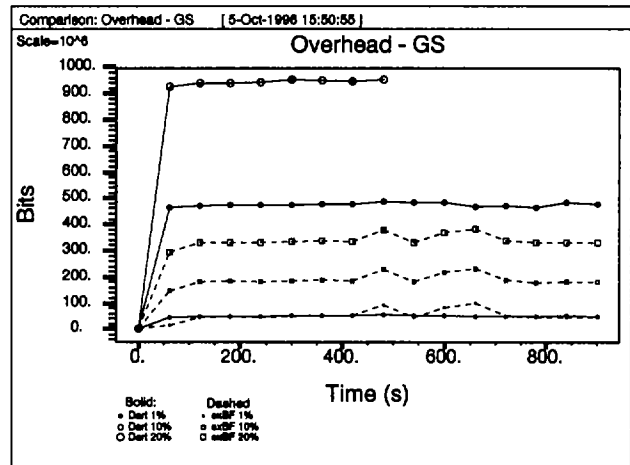


Fig. 4. Protocol Overhead Comparison in Globalstar

The fundamental concept of Darting is to trade convergence speed for overhead, so Bellman-Ford's advantage in this area is not surprising. Figure 6 and Figure 7 show the convergence rates of each protocol. An interesting feature of the Darting method is the improvement of convergence

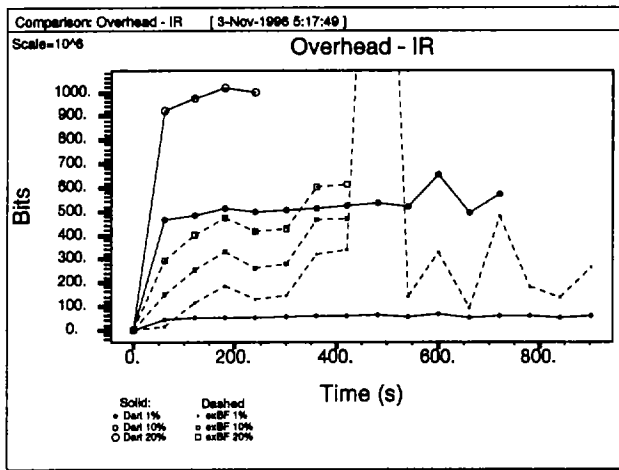


Fig. 5. Protocol Overhead Comparison in Iridium

rate from the 1% to the 10% traffic load. Because Darting relies on passing data traffic in which to embed convergence information, convergence rate suffers if insufficient traffic passes through a node.

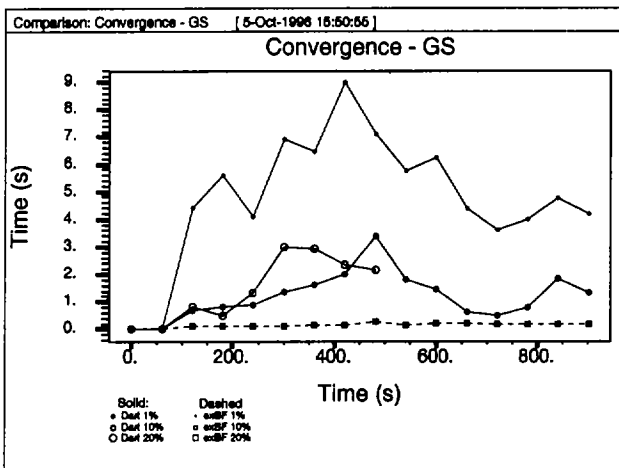


Fig. 6. Protocol Convergence Comparison in Globalstar

## V. CONCLUSION

Contrary to expectations, for the characteristics measured, there is a clear advantage to employing a distance vector routing protocol such as Extended Bellman-Ford over Darting as traffic intensities increase. Darting is severely handicapped by the requirement to place link data in each passing data packet. This results in overhead several times larger than Bellman-Ford on the same constellation. Massaging the frequency of update insertions and tweaking the resolution of the link weight function would seem critical to obtaining good performance from Darting. Better overhead results could have been obtained from Darting (at the expense of convergence rate) by re-adjusting these parameters. Mean packet traversal delay for each protocol was within a few percentage points difference at all loading levels.

The overriding delay component at the loading levels

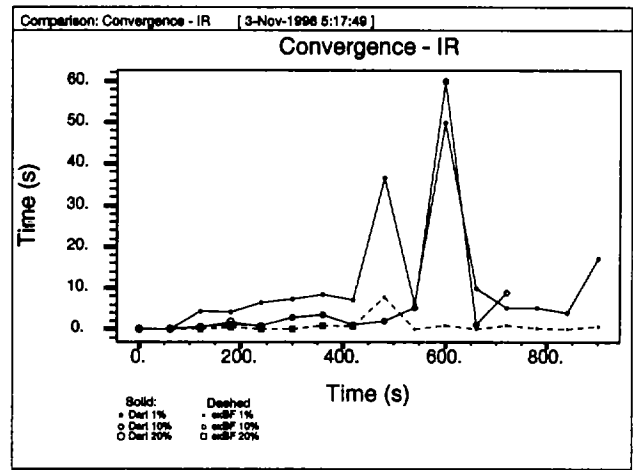


Fig. 7. Protocol Convergence Comparison in Iridium

investigated is the propagation delay between nodes. Because there is usually only one optimal path to any destination, protocol merit is determined by how quickly and efficiently an algorithm can determine the components of that optimal path. Also, in a satellite system, power is an especially critical resource and therefore protocol overhead is an important parameter to manage. Each unnecessary bit of overhead is wasted transmission power. For the environments tested, Extended Bellman-Ford has a significant advantage over Darting in this area. It obtains equal performance with a smaller overhead. More work needs to be done to optimize Darting before it should be considered for use in LEO networks.

## APPENDIX

### MODIFICATIONS TO THE DARTING ALGORITHM

Correct operation of the Darting algorithm assumes that traffic flows through all branches of the network; thus eventually disseminating complete topological information to all nodes. If this is not the case, it is possible for situations to arise where Darting will not converge to an optimal configuration, and predecessor update packets will be generated indefinitely. To alleviate this problem, a new type of "ping" packet was added to the algorithm to enable nodes that detect this type of discrepancy to exercise the portion of the network that is in question.

Let a subsection of the network be in the state shown in Figure 8, and let the local state at node  $a$  be as shown in Table I. Let there be a steady stream of traffic from  $a$  to  $g$ . Further, let this traffic be traversing the optimal path  $a-c-d-h-g$  at a cost of 4.

Now, let the link from  $d$  to  $h$  fail, as shown in Figure 9. When  $a$  next attempts to transmit to  $g$ , it will choose to go through  $c$  with an estimated cost from  $c$  to  $g$  of 3. For this first packet,  $c$  will agree with the estimate of 3 and forward the packet on to node  $d$ . Node  $d$ , having detected the failure of the  $d-h$  link, will return the packet to node  $c$  along with the information about the failed link. Node  $c$  will then realize that its best path to  $g$  lies through  $e$

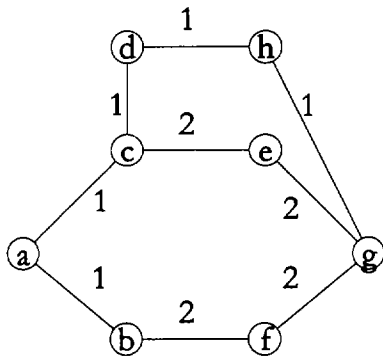


Fig. 8. Degenerate Topology 1

TABLE I  
INITIAL STATE OF NODE A

Known links at a:	
a->b=	1
a->c=	1
b->f=	2
c->d=	1
c->e=	2
d->h=	1
e->g=	3
f->g=	2
g->h=	1

and will forward the packet accordingly. No further update packets are sent because the predecessor update mechanism only updates one node upstream from the point at which the discrepancy is detected.

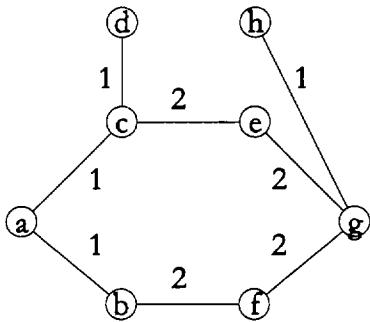


Fig. 9. Degenerate Topology 2

Now for the next packet, *a* (not having any new information) will again choose to forward through node *c*. At this point, however, *c* will realize that *a* is using outdated information because *c*'s cost to *g* is now 4, not 3.

Unfortunately, when *c* builds the update packet to send back to *a*, it enters the link information from the downstream nodes as perceived by node *c*. Thus *c* will update *a* with  $c-d=1$ ,  $c-e=2$ ,  $e-g=2$ ,  $g-h=1$ . However, none of this is new information to node *a*! Thus, *a*'s behavior will not change, and it will continue to estimate the cost from *c* to *g* as 3 (via node *d*) and *c* will continue to try to update *a* with the correct cost of 4, via node *e*. This will go on

indefinitely.

The problem occurs because in this instance, no traffic ever returns to node *a* through the path from node *d*. A mechanism is needed to force traffic to flow through that path to break the update cycle. Toward this end, a new type of "ping" packet was introduced into the protocol, and the contents of the fields in a predecessor update packet were slightly altered. The cost field was changed to reflect the cost to get from node *c* to node *g* (the cost *a* should have calculated), and the source field in a type 3 packet now has the value of "g" instead of "c".

Now, when node *a* receives the predecessor update packet from node *c*, it can check to see if the cost asserted by node *c* matches the cost it calculates from its local tables (after being updated with the new link information from *c*). If there is still a discrepancy, node *a* generates a ping packet addressed to node *d*. The ping packet is treated just like a data packet by each of the intermediate nodes, which place their local link data into the SCL field of the packet. Once the packet reaches node *d*, it is turned around by reversing the source and destination fields, and returned to node *a*. Thus, the path from *d* to *a* is exercised, and *a* will receive the information about the failed link and adjust accordingly.

## REFERENCES

- [1] Adams, W. S. and L. Rider. "Circular polar constellations providing continuous single or multiple coverage above a specified latitude," *The Journal of the Astronautical Sciences*, 35:155-192 (April-June 1987).
- [2] C. Cheng, R. Riley, S. Kumar and J. Garcia-Luna-Aceves. "A Loop Free Extended Bellman-Ford Routing Protocol Without Bouncing Effect." *SIGCOMM 1989*. 224-236. New York: ACM Press, 1989.
- [3] Commission., Federal Communications. *Comments of the Hughes Aircraft Company In the Matter of the Application of Motorola Satellite Communications Inc. for a Low Earth Orbit Based Mobile Satellite Communications System*. FCC Filing 9-DSS-P-91(87) CSS-91-010, Federal Communications Commission, June 1991.
- [4] Gross, J. and R. Ziemer. "Distributed Routing Network Performance In Hostile Environments." *Proceedings of the SPIE1059*. 22-26. 1989.
- [5] Janoso, R. *Performance Analysis of Dynamic Routing Protocols in a Low Earth Orbit Satellite Data Network*. Masters Thesis, AFIT/GE/ENG/96D-08, School of Engineering, Air Force Institute of Technology (AU), December 1996.
- [6] Leopold, R. "Low Earth Orbital Global Cellular Communications Network." *Proceedings of the ICC95*. A.2.1 - A.2.4. 1991.
- [7] Rouffet, D. "GLOBALSTAR: a Transparent System," *Electrical Communications*, 84-90 (1993).
- [8] Tsai, K. and R. Ma. "Darting: A Cost Effective Routing Alternative For Large Space-Based Dynamic Topology Networks." *1995 IEEE MILCOM Conference Record*. 682-687. New York: IEEE, 1995.
- [9] W. Wu, E. Miller, W. Pritchard and R. Pickholtz. "Mobile Satellite Communications." *Proceedings of the IEEE82*. 1431-1446. New York: IEEE, September 1994.
- [10] Walker, J. G. *Continuous whole-earth coverage by circular-orbit satellite patterns*. Technical Report 77044, Royal Aircraft Establishment, September 1977.