AFIT/GCE/ENG/99M-04

# A Performance Analysis of Dynamic Routing Algorithms in an IRIDIUM-like Low Earth Orbit Satellite System

THESIS

Stephen R. Pratt, Captain, USAF

AFIT/GCE/ENG/99M-04

Approved for public release, distribution unlimited

| | |
|---|---|
| **REPORT DOCUMENTATION PAGE** | *Form Approved*<br>*OMB No. 0704-0188* |

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| | | |
|---|---|---|
| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE<br>March 1999 | 3. REPORT TYPE AND DATES COVERED<br>Master's Thesis |

**4. TITLE AND SUBTITLE**

A PERFORMANCE ANALYSIS OF DYNAMIC ROUTING ALGORITHMS IN AN IRIDIUM-LIKE LOW EARTH ORBIT SATELLITE SYSTEM

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**

Stephen R. Pratt, Captain, USAF

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Air Force Institute of Technology
2950 P. Street
Wright-Patterson AFB, OH 45433

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFIT/GCE/ENG/99M-04

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

AFRL/IFG
Mr. Warren H. Debany Jr., PH.D., PE
525 Brooks Road, Building 3
Rome, NY 13441-4505
DSN: 587-4114 Comm: 1-315-330-4114

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

Richard A. Raines, Major, USAF
DSN: 785-3636 ext. 4715
richard.raines@afit.af.mil

**12a. DISTRIBUTION AVAILABILITY STATEMENT**

Approved for public release; distribution unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

This research presents a first of its kind comparative analysis of the Extended Bellman-Ford and Darting algorithms, using the Iridium low earth orbit (LEO) satellite system configuration for the simulation environment. The algorithms are compared to one another via discrete-event computer simulation and evaluated based on their ability to route real-time voice communications under low, medium, and high network loading conditions. The algorithms' ability to meet real-time voice constraints is evaluated with a full and degraded satellite constellation using an algorithmic satellite removal method. The investigation results indicate that both algorithms are suitable for use in a LEO environment and are capable of meeting the real-time voice communications requirements as long as a load-balancing mechanism is in place to route traffic around heavily loaded satellites. The results also indicate that the Iridium system is robust, capable of meeting the real-time voice constraints even when the constellation is degraded.

**14. SUBJECT TERMS**

Dynamic Routing Algorithms, Iridium Performance Analysis, Extended Bellman-Ford, Darting, Low Earth Orbit (LEO) Satellite Routing Algorithms, Degraded Satellite Constellation Analysis

**15. NUMBER OF PAGES**

133

**16. PRICE CODE**

| | | | |
|---|---|---|---|
| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
| Unclassified | Unclassified | Unclassified | UL |

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GCE/ENG/99M-04

# A Performance Analysis of
# Dynamic Routing Algorithms in
# An IRIDIUM-like Low Earth
# Orbit Satellite System

THESIS

Presented to the faculty of the Graduate School of Engineering

Of the Air Force Institute of Technology

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science (Computer Engineering)

Stephen R. Pratt

Captain, USAF

March 1999

AFIT/GCE/ENG/99M-04

# A Performance Analysis of Dynamic Routing Algorithms in An IRIDIUM-like Low Earth Orbit Satellite System

THESIS

Presented to the faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science (Computer Engineering)

Stephen R. Pratt

Captain, USAF

March 1999

Richard A. Raines

Richard A. Raines, Ph.D., Major, USAF
Committee Chairman

Michael A. Temple, Ph.D., Major, USAF
Committee Member

Approved for public release, distribution unlimited

# ACKNOWLEDGEMENTS

There are many people who helped me throughout the many months of work that has resulted in this thesis. Their support, guidance, and understanding has enabled me to set forth the level of effort that is reflected in this thesis.

First, I would like to thank my advisor, Major Richard A. Raines, and committee member, Major Michael A. Temple. Their guidance and insight throughout the research process, as well as their editorial comments during the preparation of this document, were invaluable. I would also like to thank Captains Ren Broyles, Chris Fogle, and Doug Lomsdalen and the assistance given while troubleshooting my simulation models. The most important people I need to thank, however, are my wife Dana and my son Tanner. They remained vigilant in their efforts to support my endeavors and were unselfish in their understanding, support, and love. Without their support I would be unable to endure the many hours it took to complete this effort.

Stephen R. Pratt

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

Today, both the military and commercial sectors are placing an increased emphasis on global communications. This has prompted the development of several Low Earth Orbit (LEO) satellite systems that promise world-wide connectivity and real-time voice communications. Although not a new concept, the tools used to implement these systems are on the cutting edge of network and satellite technology. One such tool, the routing algorithm, is one of the key components dictating the success or failure of these networks to route real-time voice communications across the globe. Very little is known about the performance of these algorithms in a LEO satellite network where the topology changes frequently. As such, this thesis focuses on the comparison of two routing protocols identified in literature as potential candidates for this type of environment.

This thesis presents a first of its kind comparative analysis of the Extended Bellman-Ford and Darting algorithms under low, medium, and high network loading conditions. Using the Iridium® LEO satellite system configuration for the simulation environment, the algorithms are compared to one another via discrete-event computer simulation and evaluated based on their ability to route real-time voice communications. The performance metrics for evaluating the algorithms are end-to-end packet delay, packet rejection rate, overhead, convergence time, and hop count. Algorithm is recorded using both uniform and non-uniform traffic distributions. The algorithms' ability to meet real-time voice constraints is evaluated with a full and degraded satellite constellation using an algorithmic satellite removal method.

Investigation results indicate that both algorithms are suitable for use in a LEO environment and are capable of meeting the real-time voice communications requirements, provided a load-balancing mechanism is in place to route traffic around heavily loaded satellite nodes. Results also indicate that the Iridium® system is capable of meeting these same constraints even when the constellation is degraded. Packet rejection rate analysis indicates that a load-balancing mechanism is needed in the network to restrict packet rejection rates to 1% or less. To that end, the algorithmic satellite removal methodology acts as a load balancing mechanism and was used to demonstrate the ability of both protocols to meet the real time voice constraints.

In 71.5% of all test scenarios, Darting routed packets up to 15.97% faster than the Extended Bellman-Ford algorithm. The Darting algorithm routed packets with 18.55% to 34.08% less overhead, while rejecting up to 3.14% less packets. In 75% of the Uniform test scenarios, Darting converged 9.47% to 40.39% faster than the Extended Bellman-Ford algorithm. Likewise, in 41.67% of *Non-Uniform* scenarios, the Darting algorithm converged 0.40% to 6.21% faster. Under both the *Uniform and Non-Uniform* test scenarios, packets travelling from Kansas City to Rio had the fewest number of hops, averaging between 3.63 and 5.29 hops each. The greatest average hop count occurred along the path from Kansas City to Capetown, averaging between 8.06 and 12.0 hops. The relationship that exists between hop count and end-to-end delay is too small to matter because end-to-end delay is mostly impacted by queuing delay.

# 1. INTRODUCTION

Current statistics indicate there will be approximately 200 million cellular and 202 million pager subscribers by the year 2000 [Rob98]. In the US alone, more than 10 million people live in rural areas where cellular communication is unavailable [Lod91]. These statistics indicate why many of today's telecommunication companies are racing to provide global communications to the masses. The number of subscribers also indicates just how dependent people are upon personal communication systems (PCS). Since these world-wide cellular and pager services are sustained by satellite systems, several companies are positioning themselves to provide improved satellite communications with increased coverage. The PCS industry hopes to build upon a growing customer base and tap into markets that have been previously unserviceable by cellular and pager communications.

## 1.1 Background

Motorola is one company that is tapping into these markets. In May of 1998, testing began on a sophisticated low earth orbit satellite (LEOS) system called Iridium®. This system is on the cutting-edge of PCS technology and promises global coverage to users of data, voice, paging, and facsimile services. Iridium® is the first low earth orbit satellite system of its kind [Bru96]. Being a pioneering system, however, means many new technological challenges must be met to make the system a reality. One such advancement is the use of inter-satellite links (ISLs)[1] to route traffic between satellites in

[1] ISLs are links established between satellites in the same plane (intra-plane) or between satellites in adjacent planes (inter-plane).

a large constellation. Another is the use of dynamic routing and link assignment algorithms in the satellite to efficiently route calls and data.

Routing algorithm selection is especially important. Protocol convergence time, average packet delay, packet rejection rate, hop count, and routing protocol overhead are key performance indicators used to assess an algorithm's ability to route traffic. A routing algorithm's performance in these areas is a key indicator of system performance. The ability of an algorithm to quickly converge to a routing solution determines how efficiently it can move data through the network. Algorithms that converge quickly typically reduce the packet traversal times by reducing the number of packet hops required to reach its destination.

Although these algorithms may converge quickly, they may also increase network overhead by introducing additional update packets. Update packets are generated and passed between nodes to keep each other abreast of network connectivity and routing changes. This can delay data and voice packets during processing and negatively impact network performance.

## 1.2 The Problem

Using specialized algorithms to manage traffic routing and link assignment is paramount. These algorithms directly impact network performance since they are responsible for efficiently routing packets through the dynamic network topology characteristic of a LEOS system. Consequently, companies deploying these systems have developed and implemented proprietary algorithms, releasing little technical information about their operational and performance properties.

Likewise, dynamic environment performance data for current routing protocol is sparse. An algorithm's ability to converge to a routing solution rapidly without introducing a great amount of overhead is a highly desirable quality needed for these dynamic topologies. Information on algorithm performance can be obtained through simulation. Simulating algorithm performance in a controlled environment provides useful information for selecting the proper routing algorithm.

## 1.3 Scope

The research presented in this thesis is a comparative analysis of two routing algorithms presented in open literature. Tradeoffs associated with the selection of a particular algorithm are identified based on specific performance criteria, specifically, protocol convergence time, average packet delay, and routing protocol overhead. To characterize the performance of these algorithms, these parameters are collected under various loading levels and in the presence of nodal failures. To see how well the algorithms adapt to nodal failure, packet rejection rates and hop counts are also collected. The model is developed is a flexible simulation, and acts as a testbed for subsequent trials involving other algorithms deemed applicable to LEOS PCS systems.

## 1.4 Approach

This thesis presents the characteristics of geostationary earth orbit satellite (GEOS), medium earth orbit satellite (MEOS), and LEOS networks and provides rational behind the shift from GEOS-based communication networks to LEOS-based communication networks. A variety of candidate routing algorithms for a LEOS network are discussed. The Iridium® LEOS system and many of its technical parameters are presented, and the methodology for testing and evaluating routing algorithms is presented.

This work expands research conducted by Fossa [Fos98], and Janoso [Jan96] by comparing two "real world" routing algorithms under higher loading levels. A simulation testbed that can obtain valuable performance criteria on a variety of routing algorithms, under various loading levels, and in the presence of nodal failures is developed. The simulation corrects the error in ISL connectivity made by Fossa and conducts simulation trials at higher loading levels on several routing algorithms, two of which were previously tested by Janoso [Jan96].

The goal of this thesis is to perform a comparative analysis of two "real world" algorithms via discrete-event simulation to gain insight into their performance. Since algorithm choice is critical to understanding system performance, this research provides insight into the performance of the Iridium® system under varying conditions. Scaling techniques developed by Fossa to achieve greater network loading levels are incorporated into the model to achieve greater loading levels and reduced simulation run times. This enhancement allows the modeled algorithms to be tested at several loading levels. Finally, routing algorithm performance is evaluated in the presence of nodal failures, assessing routing algorithm capabilities under adverse conditions. Iridium's® performance during nodal failure is of interest to the military, especially since the DoD plans to have their own access gateway [Rob98].

# 2. LITERATURE REVIEW

## 2.1 Introduction

Private companies are striving to provide truly seamless global communications to the public, making today's personal communication systems (PCS) a proving ground for new technologies. This global approach has sparked the development of several new communication satellite systems, which abandon the traditional use of geostationary earth orbit (GEO) in favor of medium earth orbit (MEO) and low earth orbit (LEO) satellite systems. LEO and MEO satellite networks increase the service regions of their designers, providing services to regions of the world where there is little or no telecommunication infrastructure, such as Asia, Africa, Eastern Europe, South America, and the Polar Regions [Gav97]. These LEO and MEO satellite networks provide global coverage to their users, which a typical GEO satellite system cannot provide.

Until recently, the technological complexity of utilizing inter-satellite links to perform network routing was beyond our reach. These technological hurdles have been overcome, and LEO and MEO satellite constellations are now the recommended configurations for providing global PCS [RuD96]. One such system, Motorola's Iridium® system, has recently been fully deployed.

This chapter introduces LEO, MEO, and GEO communication systems, and explores the complexities associated with implementing them. The LEO satellite constellation, specifically Motorola's Iridium® system, is the main focus of this chapter.

Section 2.2 provides background material for those unfamiliar with the properties of the various constellations, discusses the advantages and disadvantages associated with each, and provides the rationale for Motorola's selection of a LEO satellite system.

Section 2.3 discusses the role of routing and link assignment in telecommunication networks that have dynamic topologies, with emphasis on tasks that a routing algorithm must perform in order to achieve optimal routing and load balancing within a LEOS network. Several routing and link assignment algorithms, found in literature will be discussed in Section 2.4. Finally, Section 2.5 introduces the technical specifications, implementation, technological innovations, and overall approach to providing global communications for the Iridium® system.

## 2.2 Global Communications vs. Orbital Constraints

Both physical and performance characteristics vary with the type of satellite system deployed. To fully understand the tradeoffs associated with a GEO, MEO, and LEO system, the properties of each system and their implications are discussed in the following subsections.

### 2.2.1 Properties of a GEO Satellite System

Today, the majority of voice, video, and data services are carried by systems that utilize GEO satellites for long-haul communications [Ric95]. GEO satellites orbit at approximately 35,786 km above the equator. This altitude allows GEO satellites to rotate in unison with the earth, allowing the satellite to appear stationary with respect to a terrestrial observer. Three GEO satellites spaced 120° apart provide whole earth coverage to approximately ±70° latitude, appearing on the horizon at latitudes of ±75 and becoming undetectable beyond ±84° latitude [Ric95]. The one-way propagation delay from one point on the earth to another via a bent-pipe[2] GEO satellite is approximately 240-250 ms.

GEO satellite systems have several advantages. First, the system can provide whole-earth coverage to approximately ±70° latitude with only 3 satellites, which reduces the cost associated with the deployment of multiple satellites. Additionally, the approximate life span of a GEO satellite is 10 years, resulting in lower costs for the maintenance and replacement of a GEO communication link. A third advantage is the reduced costs for tracking the satellite. Because the satellite rotates in unison with the earth and appears fixed with an orbital period of approximately 24 hours, satellite tracking is simplified. Another significant benefit of a GEO system is that the handover[3] occurrence between satellites is very low because a user rarely travels outside the coverage area of one satellite [Re95]. This further reduces the complexity of the satellite.

On the other hand, GEO systems have several disadvantages that detract from their ability to provide truly global communications. GEO satellites do not provide coverage above ±70° latitude; therefore, no coverage is available to the Polar Regions due to elevation angle limitations (Communications to the satellite at elevation angles below 5° become unreliable or impossible) [Ric95]. Another disadvantage is the round – trip propagation delay. This delay becomes a critical factor when trying to provide voice communications, which have a 400 ms time constraint required for real-time voice over GEOS communication systems. Currently, an international call utilizing a GEOS

[2] A "bent-pipe" retransmits the data it has received from one ground station to another. It simply relays the information without processing any of the data.

[3] "Handover" is the term used to identify the transfer of a link from one satellite to another so that any communications traveling on that link may continue without interruption.

communication system takes 600 ms on average [WuM94] and results in poor quality. Likewise, these long delays inhibit the use of error correcting protocols in data communications that require error detection or selective retransmission of the erred block [WuM94]. In addition, there is a need for ground terminals to overcome losses associated with the propagation path. In order to compensate for these losses, signal power must be increased, which in turn increases the weight (and cost) of the satellite. A final constraint with GEOS systems results from its orbital altitude of 35,786 km, which places the Van Allen radiation belts in the path of the launch vehicle and the satellite. This leads to increased costs associated with launching the satellite into a higher orbit, as well as the costs and increased weight that are associated with hardening the satellite against radiation. Many of these constraints are reduced when selecting a MEO.

### 2.2.2 Properties of a MEO Satellite System

A MEO satellite system attempts to balance the benefits and limitations of geostationary and low earth orbits. A MEO satellite orbits at altitudes between 10,000 and 15,000 km. MEO satellites have a circular orbit with a period of approximately 6 hours. To achieve global coverage, 4 satellites in each of 3 planes are required. While this number of satellites is far more than the 3 satellites required for a GEO satellite system, it is far less than the number required for a LEO satellite system.

A MEOS has a one-way propagation delay of 33 to 50 ms. This delay is greater in comparison to the propagation delay associated with a LEOS, but less than the GEO delay of 250 ms. Consequently, uplink transmitters require less power to communicate with a MEO satellite than for a GEO satellite. The reduced power requirement translates to a smaller, less cumbersome transmitter.

Since the Van Allen radiation belts are located at 1500 – 5000 km and 13,000 – 20,000 km [WuM94], a MEOS is also required to have hardened electronics in order to combat the effects of radiation. As with the GEO satellite, this adds weight and expense. These shortcomings can be reduced or eliminated by utilizing LEO satellites.

### 2.2.3 Properties of a LEO Satellite System

A LEO constellation can provide global communications to the world and has significant advantages over its GEO and MEO counterparts [RiH89]. Many of these advantages are inherent in its orbital properties. The technological complexities of the implementation and design of non-GEO systems, however, temper these advantages.

A LEO satellite is located at an altitude between 500 – 2000 km. The low altitude provides benefits and induces constraints. One major benefit is reduced one-way propagation delay, which is typically between 1.67 and 6.67 ms; this is much less than GEO and MEO satellites. Furthermore, the smaller propagation delays associated with a LEO satellite have negligible effects on data communications requiring error correction and detection protocols [WuM94]. Consequently, a smaller average packet delay is achieved in LEOS networks when compared to GEOS networks of similar capacity [RaD95]. Lower orbital altitude also means it is easier to satisfy the link margin, even with low power handheld transceivers [Re95]. Additionally, a smaller degree of radiation hardening is needed since the satellites do not pass through the Van Allen Belts. This yields smaller satellites with decreased orbital masses between 50 to 700 kg [Re95]. An additional benefit includes true worldwide coverage, including polar coverage – a prime area of concern to the military [PrL93].

However, a LEOS system does have its drawbacks. First, a greater number of satellites are required to achieve global coverage. The number of satellites is set by the constellation design, so selection of an appropriate constellation is critical [Ste96]. The Iridium® constellation employs a total of 66 satellites -- 11 satellites in 6 planes. The reduced size and weight of the satellites allow multiple spacecraft to be placed into each orbital plane at the same time by a single booster. In addition, the boosters required to place satellites in a LEO constellation are smaller and cheaper than those required for a MEO or GEO. The increased number of satellites also increases survivability of the network because of node redundancy [RiH89]. Finally, the large number of orbital nodes gives a LEO satellite system the capability to achieve higher traffic densities per cell when compared to a GEO mobile satellite system (MSS) [Re95].

LEOS networks will provide a new era in global connectivity, and will present new challenges to traditional routing algorithms. The new systems promise lower propagation delays, greater survivability, global coverage, and handheld portability, but to accomplish this they must overcome the problems associated with a time-variant topology. A new breed of routing and link assignment algorithms that efficiently route traffic between satellites will overcome this hurdle.

### 2.3 Routing in a Dynamic Network Topology

One of the critical drawbacks of LEO satellite systems is the constellation's time-varying geometry and its evolving coverage caused by increased orbital speeds at lower altitudes [RaM95]. Consequently, the maximum in-view time of a satellite, with respect to a fixed point on the earth, is approximately 10 - 20 minutes, requiring frequent handovers between satellites [Re95, UzY97]. These handovers force a mobile call to be handed off multiple times across inter-satellite links to avoid forced call termination. LEO satellite crosslink hardware increases the satellite complexity since links must be dynamically established to account for changing network topology [ChK95]. The net result is that ISLs and the traffic traversing them must be managed and maintained with efficient algorithms. An algorithm's ability to rapidly converge to a routing solution without introducing a great amount of overhead are used for both algorithm and network performance indicators.

### 2.3.1 Traffic Routing

Routing algorithm performance directly impacts system [Re95, DoK95], so it is imperative that the routing algorithm converge quickly to a solution without producing a large amount of network overhead. It is therefore important to review algorithms developed specifically for use in LEOS communication networks and those potentially adaptable to these networks.

Although the literature contains many articles, studies, and papers on conventional terrestrial routing algorithms, little is available on dynamic routing algorithms, their application, and behavior in LEOS. In an effort to contribute to this specialized area, the bulk of this chapter concentrates on candidate LEOS system routing algorithms and the enhancements that can be made to them.

### 2.3.2 Selecting the Right Routing Algorithm

The primary attributes used to characterize routing protocols are complexity, loop-free[4] routing, convergence, storage overhead, and computational/transmission overhead [KrV97]. In a network where the topology is dynamic, these parameters are especially important, since faster convergence to a new route after a topology change insures quick delivery of the data.

Loops increase the time required for a data packet to reach its final destination and introduce overhead, having a negative impact on network performance. In the presence of node or link failures, loops can cause destinations to be unreachable. As a result, loop-free protocols reduce overhead and decrease convergence time. These factors are key for any LEOS routing algorithm.

Many LEO networks use dynamic link assignment to establish connections between themselves and any visible neighbors. The primary goal of link assignment algorithms is to concentrate on connectivity of the network, rather than maximization of network performance [ChK95]. A review of the latest developments in this field follows.

## 2.4 Review of Various Routing Algorithms

Using conventional routing algorithms in a dynamic network topology introduces a great deal of overhead. These algorithms use one of two methods to insure proper message routing, either synchronizing the network so that each node has the same view

[4] "Loop-free" implies that the path from one node to another does not traverse the same node twice.

of the network's connectivity, or flooding[5] the network with duplicate message packets to overcome the dynamics of the network. Both methods, however, introduce overhead into a system and ultimately have a negative impact on performance [TsM95]. In addition, the additional overhead results in extra link resource requirements in order to implement these conventional routing algorithms.

Iridium® uses a proprietary algorithm for link assignment and routing. Since direct study is impossible, it is necessary to review the literature to find routing and link assignment protocols that are suitable for a LEOS system so that performance of each can be determined via modeling and simulation.

### 2.4.1 Extended Bellman-Ford

In [ChR89], the Extended Bellman-Ford (EXBF) algorithm is presented. This algorithm is based on the conventional Bellman-Ford (BF) algorithm which solves the single-source shortest-paths problem. However, the EXBF includes several enhancements to overcome the problems that restricted BF's use in dynamic networks.

One problem is the potential for loops to exist in the connectivity matrix maintained by each node. In the presence of link or node failures, loops cause the BF algorithm to take an extended period of time before converging to a solution. In fact, under these circumstances, the BF algorithm may not converge to a solution at all [ChR89]. To have an acceptable convergence time, loops within the distance tables must be minimized or eliminated so packets do not "bounce" between nodes. The removal of

[5] "Flooding" is a methodology used by conventional algorithms to insure a given packet reaches its destination. A node will broadcast the data packet to all of its neighbors, whom in turn broadcast it to all of

loops is especially critical in networks with dynamic topologies. If loops are not removed, the algorithm may not converge to a solution. Changes in connectivity are more likely to increase their probability and may result in the changes not being propagated throughout the entire network [Pax97].

To overcome the loop problem, Cheng, et al [ChR89], maintain only the simple paths[6] to nodes, and only update the paths to selected neighbors of the current node. This approach eliminates the long convergence time experienced in the presence of loops. In addition, maintaining only simple paths to a node eliminates the failure of the BF algorithm to converge to a solution in certain cases. While not eliminating loops, the approach recommended in [ChR89] is one solution to the problems they create. In order to be totally loop-free, the algorithm utilizes inter-neighbor coordination [Gar86].

Elimination of lengthy convergence times and convergence failure are necessary for EXBF to be considered for use in a LEOS network. Janoso [Jan96] evaluated the performance of the EXBF algorithm in LEOS network simulation trials. Although the use of inter-neighbor coordination was not implemented in these simulation trials, results indicated the EXBF had a significant performance advantage over another algorithm, Darting, to be discussed next. EXBF converged to a solution faster and with less overhead when compared to Darting.

their neighbors except for the one that initially sent the packet. This continues until the packet reaches its destination, which occurs only if the destination is connected to the source of the data packet.

[6] A "simple path" is a sequence of nodes with no node being repeated more than once, i.e. a loop free path.

### 2.4.2 Darting

Darting is a protocol that has been proposed as suitable for use in LEO networks [TsM94]. This particular protocol attempts to reduce the message overhead introduced by conventional flooding algorithms. The algorithm delays the "update" messages that are sent to update the network until absolutely necessary. Darting uses two different methods for updating the network's connectivity routing tables.

First, updates are accomplished by each node encapsulating their local topology changes into the data packets. Nodes that receive the data packets incorporate these updates locally, then add their own updates and pass the data packet along; the process is repeated until the packet reaches its destination. The second method updates all nodes in a data packet's route already visited by the packet. These updates occur when a discrepancy is found between the connectivity data encapsulated in the data packet just sent and the present node's local view of connectivity. Darting creates an update packet which is sent back to the predecessor nodes; these nodes then incorporate any necessary updates. Both methods are triggered only when a data message is present, so a node's view of the network's connectivity remains unchanged in the absence of data messages,.

The authors [TsM94] performed low load simulation trials that compared Darting to conventional routing algorithms. The scope of these trials was limited and did not attempt to model and analyze performance characteristics of traffic travelling between terrestrial earth stations. The results from these preliminary simulations indicated a cost-saving potential for implementation into LEOS communication networks. Janoso [Jan96] performed additional simulations with Darting to characterize its performance further. In [Jan96], the Darting algorithm is incorporated into a simulated Iridium® system, a comparative analysis of the Extended Bellman-Ford algorithm and the Darting algorithm was performed. Although these trials modeled traffic between terrestrial earth stations, only low loading levels were attained. The low load results indicated that the Darting algorithm required as much as 72% more overhead when compared to the Extended Bellman-Ford algorithm. The additional overhead was a result of a weakness in the Darting algorithm, which manifests itself when routing packets under non-uniform traffic loads. Janoso [Jan96] found that encapsulation of updates into the data packets severely handicapped the algorithm, which diminished the overhead savings that resulted from the algorithm's selective update methodology. In summary, Janoso recommended modifications be made to Darting's link weight function and to its update frequency to improve the performance of the algorithm.

### 2.4.3 Finite State Automaton (FSA) Routing

Finite State Automaton (FSA) Routing is a static[7] link assignment algorithm proposed by [ChK95], and is based on segmentation of the visible links between satellites during specific time periods. The purpose of FSA is to create a subset of links at each node by eliminating non-optimal links and selecting the optimum link from this subset.

The algorithm uses a two-step approach. First, it solves the changing topology of the network by selecting links based upon the state[8] of the satellite constellation. FSA finds the optimal path for traffic based upon the traffic distribution over selected links.

[7] A static routing algorithm utilizes a fixed routing methodology, and is tailored to a network topology. A route is chosen deterministically from a series of candidate links, which are usually based upon shortest paths.

[8] A "state" of the FSA corresponds to an equal-length interval in the LEOS system period. Only those satellites that belong to the same state are visible to one another.

The goal is to maximize the number of carried calls. Routing is then accomplished using a routing algorithm of choice.

Each state of the constellation represents the topology during an equal-length time interval. The interval between each state corresponds to a movement of the satellite along its orbital path (in degrees). The interval is determined to be the maximum degree movement possible such that the satellite can maintain membership in the state. This allows a link assignment table, found in each satellite, to have a one-to-one correspondence with each state. A new link assignment table is loaded each time a satellite transitions between states. Link assignment is therefore based on the visibility of satellites at fixed intervals in time. The network is viewed as a fixed network at each time interval, and traffic routing is then performed during this interval using a routing algorithm.

Chang, et. al. [ChK95], assert that this technique can be applied to a LEO satellite network because LEO satellites have periodic orbital movements that result in a finite number of states. In addition, they suggest the use of a fixed routing table based upon the state of the network's topology. Each satellite loads a new routing table upon transition to a new link visibility matrix. By doing so, the authors suggest that the unstable behavior of a dynamic link assignment and routing algorithm can be avoided during transitional periods of the network.

The authors performed simulations comparing the FSA link assignment protocol using both static and dynamic routing algorithms [ChK96, ChK98]. Two methods of FSA link assignment were tested for each routing algorithm. The first method tested was a link assignment methodology that had been optimized for the traffic pattern of each state in the system. Optimization was accomplished "via simulated annealing" [KiG83, Rut89, ChK98]. The second link assignment methodology used was a standard link assignment for networks possessing a mesh-like topology. In their simulations, the optimized FSA algorithm with static routing out-performed all other candidate algorithms, having had a lower probability of blocking a newly initiated call and a higher probability of maintaining an ongoing call [ChK98]. The poor performance of the other candidate algorithms was attributed to the updates requested by the dynamic routing algorithm. After an update, time was required to synchronize all of the satellite's routing tables. This is not found in static routing, which uses pre-computed routing tables and inter-satellite link assignment tables stored in memory on-board each satellite. Additionally, testing indicated that static routing performed better than dynamic routing when the link assignment methodology remained constant.

In summary, as the network enters a new state, each satellite loads a different set of corresponding routing and link connectivity tables which are stored and maintained on the satellite itself. The satellite network is treated as a fixed network with several states, rather than a dynamic network. It is important to note that the FSA algorithm optimizes link assignment and is to be utilized in conjunction with a routing algorithm. Also, during topology changes, more links would need to be rerouted because link assignment is optimized based upon traffic patterns of the network [Uzu98]. This increased the amount of overhead introduced by the protocol.

### 2.4.4 Cluster-based Routing

Krishna, et al [KrV97], propose a Cluster-based (CB) routing approach for dynamic networks. This approach is similar to the FSA algorithm presented in [ChK98]. Based upon their connectivity, the algorithm groups nodes into clusters which overlap when nodes are shared. A topology change in the network is indicated by a change in a cluster's membership. Routing is accomplished node to node within the cluster, and then cluster to cluster. The authors suggest that this routing methodology is suitable for any dynamic network.

This protocol incurs lower overhead during a topology change and quicker convergence when compared to conventional terrestrial-based protocols [KrV97]. It is not evident whether the overhead savings will be present in LEOS networks with dynamic topologies. Membership of the clusters in LEOS networks may change rapidly, introducing additional overhead when establishing links between nodes and clusters.

### 2.4.5 Bubbles Routing

"Bubbles" is a dynamic routing algorithm proposed for use in high-speed, dynamic networks [DoK95]. The authors propose a partitioning algorithm that will divide the network up into "bubbles" in which membership changes by either expanding or contracting the bubble. The algorithm constructs bubbles based on the size bounds on connected components during a topological change. The authors hope to confine the membership changes to a particular bubble. Assigning membership to each reduces the overhead by only updating the "bubble" when membership changes, minimizing the effect of a topology change. Following this, a combination of a distributed routing database, a routing strategy, and a routing database update are used to route the traffic [DoK95]. This link-assignment methodology would not be viable in a LEOS network such as Iridium® because a change in topology could not be localized to one particular bubble. Consequently, a change in topology would have a ripple effect throughout each bubble.

### 2.4.6 Probabilistic Routing Protocol (PRP)

The PRP is a link assignment protocol developed specifically for LEOS networks [Uzu98]. The protocol reduces the number of re-routing attempts due to topology changes in the network. The protocol bypasses links that would necessitate re-routing of a newly initiated call based upon a probabilistic determination of call length. As such, a probability distribution function (PDF) for route time usage must be determined in order to utilize this protocol. Within a certain probability, the PDF indicates which route will not require a link handover during the expected length of the call [Uzu98]. A copy of the connectivity matrix is copied to the probabilistic connectivity matrix. Those ISLs that are unable to meet the desired target probability for maintaining a call are removed from the probabilistic connectivity matrix. This approach limits the potential occurrence of link handovers. Furthermore, the need to reroute calls is minimized, as is the signaling overhead that would otherwise be caused by a link handover.

The PRP defines a probabilistic connectivity matrix to minimize the number of link handovers a call may experience, after which a routing protocol of choice is used to manage traffic. During the evaluation of this protocol, it was found that the number of re-routing attempts due to link handover decreased for large target probability values.

This, however, resulted in a greater probability of call blocking. The probability of blocking was also found to be smaller for handover calls, but greater for new calls. Since it is more desirable to block a new call than to terminate or interrupt an on-going call, it is suggested that this protocol be used for new calls only.

The overhead associated with maintaining both a connectivity matrix and a probabilistic connectivity matrix was not discussed. Likewise, the computational overhead required to compute the PDF and process the traffic distribution was not addressed either. In a dynamic environment where the connectivity matrix is constantly changing, these items could adversely impact network performance. Algorithm convergence time, system overhead, and average packet delay are several criteria that need to be addressed before any conclusions can be reached about this algorithm.

### 2.4.7 Footprint Handover Re-Route Protocol (FHRP)

The FHRP is a protocol proposed by Uzunalioglu, et al [UzY97], and is suggested for use in any type of connection-oriented network. FHRP is not a stand-alone protocol; rather, it is a protocol that maintains an optimal route for calls in progress. During a call in progress, the protocol attempts to select an optimal route from a static routing table, which reduces the overhead characteristic of a dynamic routing algorithm. However, if the static routing table doesn't find a route with the necessary capacity, a dynamic routing algorithm is invoked to determine a new route.

FHRP uses the footprint (service area) of the satellite that currently is servicing a call as a reference for a two step optimal routing process – augmentation and re-routing. The footprint of the augmented satellite is checked to insure that it can support the call with up/downlinks. This is accomplished by checking for available uplink and downlink capacity in the augmented satellite. If enough bandwidth does not exist, the connection is blocked and augmentation does not take place. If enough bandwidth is available, augmentation proceeds.

During augmentation, a link is established from the satellite servicing the call to a new satellite with an optimal route. The connection is then rerouted to the augmented satellite while the first satellite is removed from the path. If the static routing table does not have a direct link to the new satellite, the routing algorithm re-routes the call dynamically to a new route with the required capacity. Regardless of which routing method is used, the resulting change in route information is sent to the ground terminals.

Updating of the static link tables occurs at predetermined intervals in order to maintain optimal routing selection. Updates are required in order to maintain optimal routing in the case when several satellite augmentations were required for a call. Thus optimal routing is maintained at the expense of network overhead.

In an event-driven simulation, the authors compared this protocol to both a static network and to pure augmentation. Their study showed that the FHRP protocol performed as well as the static network, and much better than the pure augmentation methodology. Criteria for this evaluation was based upon the probability of call blocking. No other performance metrics were collected.

## 2.5 The Iridium® System

Since the framework for testing these algorithms is based on the Iridium® system, a brief introduction to the system and its technical parameters is presented here.

Iridium® was conceived in 1987 and is the first private, global wireless communication system to provide voice, data, fax, and paging services to the world [Bru96]. The original configuration called for 77 satellites, and was named after the Iridium atom , which has 77 orbiting electrons. However, in an effort to reduce costs, the constellation was reduced to 66 satellites. The Iridium® system orbit is based on a constellation proposed by Adams and Rider [AdR87].

At an altitude of 780 km above the earth, 66 satellites are arranged in 6 planes with each plane containing 11 satellites. Planes have a near-circular orbit, with co-rotating planes spaced 31.6 degrees apart and counter-rotating planes (1 and 6) spaced 22 degrees apart [Hub97]. The minimum elevation angle for an earth station is 8.2 degrees. Average satellite connection time is approximately 10 minutes [Com93]. The orbital mass of each satellite is 1,516 lbs., and each satellite is capable of establishing four inter-satellite links. Onboard processing and ISL utilization that Iridium® are the most challenging aspects of the Iridium® system [Com93]. Having already presented the complexities associated with link assignment and traffic routing, a more detailed look at the Iridium® link mechanism is warranted.

### 2.5.1 Inter-Satellite Links

ISLs are links established between satellites in the same plane (intra-plane) and between satellites in adjacent planes (inter-plane). Intra-plane links are maintained permanently, with each satellite having forward and aft connectivity to satellites directly in front of and behind it. Inter-plane links are dynamically established and terminated as the satellite transcends its orbital path. Except for satellites in counter-rotating planes one and six, each satellite maintains four ISLs. The satellites located in planes one and six only maintain three ISLs each, 2 of which are intra-plane. Satellites in planes one and six are not allowed to establish ISLs between each other due to the rapid angular change occurring between satellites in counter-rotating planes [Gav97]. A depiction of these ISLs is shown in Figure 1 below where each arrow represents the position of an active satellite.

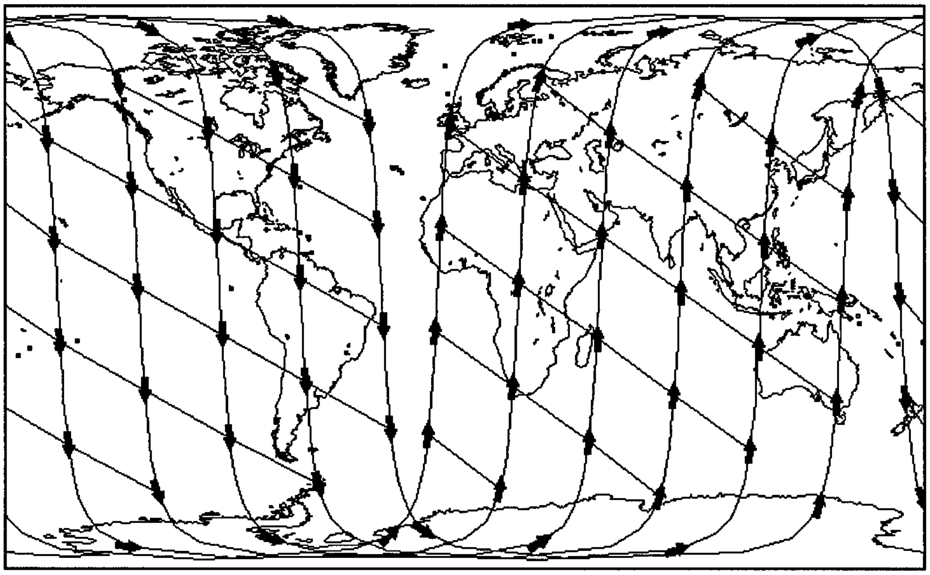

*Figure 1: Depiction of Iridium® inter-satellite links*

ISLs provide the LEOS network a greater level of autonomy when compared to GEOS networks. Fewer terrestrial gateways are needed because call routing takes place via these ISLs. As such, Iridium® does not depend on the services provided by other organizations such as regional telephone companies [Gav97], which translates into greater corporate profits since terrestrial connectivity fees are reduced [WeM95].

The complexity of the Iridium® satellites is due to on-board processing capabilities required to manage and support the ISLs and connectivity of the network [Hub97]. Using ISLs necessitates the need for on-board satellite processing and efficient link assignment and routing algorithms that can optimize network delay with little overhead. The algorithm's ability to quickly converge to a routing solution while introducing minimal overhead directly impacts network performance and the PCS. The algorithm's importance cannot be trivialized and is the basis for current and future research endeavors.

## 2.6 Summary

This chapter presented characteristics for GEO, MEO, and LEO satellite systems. Focusing on LEOS systems, the chapter detailed the many benefits telecommunication companies hope to obtain when these systems become operational and outlined the many challenges that need to be overcome in the development and implementation of these systems. One such challenge is the use of dynamic routing algorithms and link assignment protocols in LEOS networks. Several of these algorithms and protocols were reviewed. The Iridium system, the first LEOS communication network to use dynamic algorithms and protocols, was discussed. The technical parameters of the Iridium® system and its use of inter-satellite links was explained.

In the future, PCS users will become more dependent on LEOS systems, as evidenced by the recent advent and use of these systems in both the commercial and military sectors. The success of these systems is largely dependent on their ability to efficiently route traffic throughout the network. It is extremely important, therefore, that link assignment protocols and routing methodologies be explored and tested.

# 3. METHODOLOGY

## 3.1 Introduction

The object of this chapter is to define the methodology used to develop and analyze the simulation model. It rationalizes the method used for obtaining the performance criteria and addresses the testing, verification, and validation of the model. All assumptions and input parameters are also identified and discussed.

Section 3.2 restates the research problem, defines the scope of the problem, justifies the use of discrete-event simulation for this research, and reviews the protocols selected for experimentation. The operational assumptions used during simulation trials are defined in Section 3.3. Section 3.4 details the design of the simulation model and presents an operational overview. Section 3.5 discusses the simulation input parameters. Performance metrics are discussed in Section 3.6, and verification and validation procedures are outlined in Section 3.7. Finally, Section 3.8 presents a summary of the main points of this chapter and an overview of the performance metrics analyzed in Chapter 4.

## 3.2 Problem Overview

As previously stated, there is very little published research on the performance of routing algorithms in a LEO environment. Likewise, the suitability of algorithms proposed in the open literature lack any significant comparative testing and analysis in an operational environment. This section restates the main thrust of this thesis area, and begins by detailing the problem and the methodologies used to tackle this problem.

### 3.2.1 Problem Definition

The literature review indicates that little information is available on the performance of routing algorithms in a LEO network environment under moderate to high loading levels and in the presence of nodal failures. Likewise, little information is available in published literature comparing the performance of routing algorithms in a LEO network.

### 3.2.2 Problem Statement

The focus of this research is to perform a comparative analysis of two routing algorithms that are considered suitable for implementation in a LEO network. The performance of these algorithms is evaluated in a variety of network conditions: low loading levels, medium loading levels, high loading levels, and in the presence of nodal failures.

### 3.2.3 Scope of Problem

To achieve accurate results in a timely manner with available computing resources, it is necessary to limit the scope of this model. Consequently, several aspects of the Iridium® system are limited in scope. These areas, however, have minimal impact on the performance criteria used to evaluate routing algorithms. The areas limited in scope include satellite equipment failure, the number and type of users, handoff procedures, call setup procedures, and traffic distribution.

Equipment failures can cause a satellite to have limited functionality or no functionality at all. The types of equipment failures that can occur on-board a satellite are numerous. It is time prohibitive to model and simulate the many combinations and probability of these failures. As a result, this research characterizes a satellite failure as a complete loss of satellite functionality, effectively removing the satellite and its communication links from the network. These conclusions represent a worst case result when a satellite is lost.

Iridium® supports both stationary and mobile users. While it is possible for a mobile user to leave the service area of one satellite and enter that of another, it is more probable that movement of the satellite causes a handover. For the purposes of this simulation, all users are modeled as stationary earth stations. No attempt has been made to capture the traffic that occurs between the Iridium® system and the public switched telephone network (PSTN).

Seven locations, identified in [Fos98], are used for ground stations. Locations are selected to evenly distribute the traffic sources and destinations geographically throughout the world, and are summarized in Table 1.

*Table 1: Earth Station Data*

| City | Longitude | Latitude | Altitude |
|---|---|---|---|
| Rio de Janeiro | -43.22 | -22.90 | 0.01 |
| Melbourne | 144.97 | -37.80 | 0.00 |
| Kansas City | -94.59 | 39.13 | 0.23 |
| Dhahran | 50.00 | 27.00 | 0.76 |
| Beijing | 116.47 | 39.90 | 0.18 |
| Berlin | 13.42 | 52.53 | 0.03 |
| Capetown | 18.37 | -33.93 | 0.00 |

As previously stated, handoffs between satellites take place in order to maintain a call in progress. In addition, beam-to-beam handoffs take place within a satellite. During a satellite-to-satellite handoff, propagation delays between the earth station and the satellite change, as does the shortest path propagation delays. On the other hand, the delays introduced by beam-to-beam handoffs are negligible compared to the satellite-to-satellite handoff delays. Modeling the Iridium® satellite beam-to-beam handoff capabilities would likewise increase the simulation run-time, requiring an additional 3168 queues [Fos98]. Consequently, this research doesn't model the beam-to-beam handoff functionality and the associated delay that occurs with it.

One criteria for determining algorithm performance is average packet delay, defined as the delay a packet experiences as it traverses the network from source to its destination. Average packet delay does not encompass the delay associated with the call setup procedure, limiting itself to the delay experienced after the call has been established. This metric is critical since is must be kept within specific limits (i.e. 400 ms for real-time voice communications). Modeling the call setup procedure would only serve to increase simulation complexity and runtime, providing no additional insight into the average packet delay metric [Fos98].

Each ground station generates traffic based upon both a uniform and a non-uniform traffic distribution pattern. The uniform traffic pattern is used to baseline the performance of the routing algorithms. Both source and destination have uniform traffic patterns. Uniformity allows a greater number of nodes to be exercised and provides a baseline for comparison against the non-uniform cases.

Typical real-time communication system traffic patterns are inherently non-uniform. To analyze this conclusion, a communication link between two earth stations is created to carry a greater amount of the traffic. This simulation provides the basis for non-uniform analysis.

Three loading levels are used to stress the routing algorithm performance, these are low, medium, and high. The amount of traffic generated and the traffic source transmit probabilities are discussed in Section 3.3.2, 3.3.6, and 3.3.7 respectively.

### 3.2.4 Method of Evaluation

There are three acceptable methods used in the evaluation of system performance - analytical modeling, simulation, and measurement [Jai91]. As previously stated, the performance of these algorithms would optimally be evaluated on an actual LEO system. The Iridium® system is the only deployed system, and was chosen as the basis for this evaluation. Iridium® has only recently been fielded and is currently undergoing performance testing by Motorola. The system is unavailable for direct measurement and would require instrumentation on-board the satellites. Additionally, the algorithm used by Iridium® to perform its routing is proprietary information. It would not be feasible to upload a new routing protocol to the satellites so that a trade-off evaluation could be conducted on several routing algorithms. In addition, unless performed over an extended period, measurement does not provide conclusive evidence that an improvement was a result of a parameter setting rather than a random change in the environment [Jai91]. Consequently, measurement was ruled out as a technique for evaluation.

Analytical modeling was also ruled out as a possible evaluation technique. Analytical modeling only provides a low level of accuracy due to the many simplifications and assumptions that must be made in order to obtain a result [Jai91]. Results obtained using this method are usually subject to skepticism unless validated by simulation or measurement.

Simulation is the technique selected to evaluate the performance of these algorithms. Simulation allows the tester to vary the amount of detail depending on system requirements. In addition, a model can be readily validated, insuring that the assumptions used in developing the model are reasonable, and if correctly implemented, produce results that accurately depict the performance of the real would system [Jai91]. Likewise, simulation is easier to debug, which eases verification of the simulation's implementation.

## 3.3 Operational Assumptions

Throughout the development of this simulation many simplifying assumptions were made when technical parameters of the Iridium® system were unavailable. In other cases, assumptions were made in order to simplify the model to reduce both processor utilization and simulation run-time. These assumptions and their motive are outlined in the subsections below.

### 3.3.1 Packet Structure and Size

The exact size and structure of the satellite data structure used by the Iridium® system has not been published. For the purposes of this research, the satellite packet size was set to 432 bits, as derived in [Fos98]. The structure of the satellite packet is unpublished as well. The necessary fields used to properly route the data packet from its source to its destination are derived, as are fields used to collect heuristic data and parameters. The composition of the frame is not critical, however, it is important to identify its composition to assist in the description of how the model works. The structure of this model's packet is outlined in Table 2.

*Table 2: Data Structure Fields*

| Field Name | Type | Description |
|---|---|---|
| Source | Integer | Packet origination node |
| Destination | Integer | Packet destination node |
| Packet Length | Integer | Size of packet generated |
| Sequence Number | Integer | Unique identifier for packet |
| Current Node | Integer | Current location of packet |
| Next Node | Integer | Next node in path to destination node |
| TNOW | Real | Simulation runtime |
| Delay | Real | Cumulative delay of during path traversal |
| Hop Count | Integer | Cumulative number of nodes traversed |
| Time Packet Sent | Real | Time packet sent from source |
| Time Packet Received | Real | Time packet received at destination |
| Time Packet Created | Real | Time of packet creation |
| Type | Integer | Type of packet (Satellite or Update) |

### 3.3.2 Packet Arrival Rate

Each satellite is equipped with up to 48 spot beams [Hub97]. The maximum number of users per cell is 80 [Fos98]. The maximum number of users supported by each satellite is determined by multiplying the number of spot beams by the number of users per cell. This yields a theoretical total of 3840 users that can be supported by each satellite. Although several satellites will always be over regions of the earth without that many users (i.e., the Polar Regions and bodies of water), this figure is utilized to calculate the maximum packet arrival rate for each satellite. The Time Division Multiple Access (TDMA) frame of the Iridium® system is 90 ms long [Hub97]. This time is taken and divided into 3840 packets to yield 42,667 packets per second (pps), with a minimum time between packets of 23.44 microseconds [Fos98].

The calculations for the loading levels are based upon the 42,667 pps arrival rate. Each earth station generates a percentage of this packet arrival rate (Table 3) based on the desired loading level. It is important to note that this number is used because each ground station typically has only one satellite for its respective service area. Although it is possible to have more than one during certain instances, the probability of one satellite servicing only one earth station is more likely.

*Table 3: Loading Levels*

| **Loading Level** | **Earth Station Traffic Generation Rate (pps)** |
|---|---|
| 100% (High) | 42,667 |
| 83% (Medium) | 35,413 |
| 50% (Low) | 21,334 |

### 3.3.3 Packet Generation Method

Each satellite node is assumed to have only one processor. This processor performs various functions, including the routing of traffic. An appropriate method of modeling a single-processor system is to model it as a M/M/1 queue [Jai91]. Likewise, the traffic source is modeled using a Poisson traffic source, which is a typical process to use when modeling a voice communications system that utilizes M/M/1 queues to model the processors [Jai91].

### 3.3.4 Satellite Processing Delay

The processing speed of the Iridium® satellites is not published, hence, an processing speed of 14 μsec was used. This is the same processing speed used in [Fos97], and equates to 77,428 pps that can be processed by each satellite processor if utilized at 100%. The processor will not utilize 100% of its capability for traffic routing. Processor utilization rate is discussed in subsequent sections.

### 3.3.5 Loading Levels

The main thrust of this thesis is to obtain metrics on the performance of various algorithms under various loading levels. Janoso [Jan96] was only able to obtain results at relatively low loading levels (20%) due to simulation overhead. This research varies the earth station's packet generation rate to obtain a low, medium, and high loading environment.

As seen in Section 3.3.2, the maximum earth station packet generation is 42,667 pps. This number is used to determine the low, medium, and high loading levels used during simulation. Low, medium, and high loading level are arbitrarily chosen to be 50%, 83%, and 100% of 42,667 pps. In Section 3.3.5, the maximum amount of traffic that can be "theoretically" processed by each satellite, given 100% processor utilization, is 72,428 pps. Satellite processor utilization is then computed by dividing the Earth Station Arrival Rate by 72,428 pps. Loading levels, traffic generation rate, and computed processor utilization are defined in Table 4.

*Table 4: Uniform Traffic Distribution Load Levels*

| **Uplink Utilization (Loading Level)** | **Earth Station Traffic Generation Rate (pps)** | **Network Traffic Generation Rate (pps)** | **Satellite Processor Utilization** |
|---|---|---|---|
| 100% (High) | 42,667 | 298,669 | 58.91% |
| 83% (Medium) | 35,413 | 247,891 | 48.89% |
| 50% (Low) | 21,334 | 149,338 | 29.46% |

### 3.3.6 Uniform Traffic Distribution

The loading levels defined in Section 3.3.5, Table 4, are used in the uniform test scenario. Both the traffic generated by the source and traffic received at the destination are uniform in nature. This forms the baseline performance metrics that are the basis of comparison. Table 5 outlines the transmit and destination probability parameters used in this test case.

*Table 5: Uniform Traffic Distribution Breakdown*

| | | Destination Probabilities | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Location** | **Transmit Probability** | **Rio de Janeiro** | **Melbourne** | **Kansas City** | **Dhahran** | **Beijing** | **Berlin** | **Capetown** |
| Rio de Janeiro | 0.143 | 0 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 |
| Melbourne | 0.143 | 0.167 | 0 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 |
| Kansas City | 0.143 | 0.167 | 0.167 | 0 | 0.167 | 0.167 | 0.167 | 0.167 |
| Dhahran | 0.143 | 0.167 | 0.167 | 0.167 | 0 | 0.167 | 0.167 | 0.167 |
| Beijing | 0.143 | 0.167 | 0.167 | 0.167 | 0.167 | 0 | 0.167 | 0.167 |
| Berlin | 0.143 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0 | 0.167 |
| Capetown | 0.143 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0 |

### 3.3.7 Non-Uniform Traffic Distribution

This simulation utilizes a non-uniform traffic distribution to simulate the traffic patterns predicted to exist within the Iridium® network. The probability of an earth station transmitting and the destination are chosen such that there is a greater amount of traffic occurring between Kansas City and Dhahran. These two locations represent both a high-traffic link and the greatest geographic separation in the model. This link is used to analyze the impact of node removal on the performance of the routing algorithms and the network.

In order to compare the results to those found under the uniform traffic distribution case, the transmit probabilities must be adjusted such that the satellite processor utilization of those nodes in the high traffic link experience similar utilization rates. Only then can a comparison be made between the results found in the non-uniform and uniform cases. The transmit probabilities for each earth station under the various loading levels are found by using the fixed arrival rate of 149,338 pps which is used for the low-load, uniform simulation trials. These transmit probabilities and their associated arrival rates for each loading level are summarized in Table 6, Table 7, and Table 8.

*Table 6: Non-Uniform Traffic Distribution, Low Load Levels*

| **Transmit Probability (TR)** | **Uplink Utilization**<br>**HTL=High Traffic Link**<br>**OT=Other Traffic Links** | **Earth Station Traffic Arrival Rate (pps) (149,338 x TR)** | **Satellite Processor Utilization** |
|---|---|---|---|
| 0.1429 | 50% (HTL) | 21,340 | 29.46% |
| 0.1428 | 49% (OT) | 21,325 | 29.44% |

*Table 7: Non-Uniform Traffic Distribution, Medium Load Levels*

| **Transmit Probability (TR)** | **Uplink Utilization**<br>**HTL=High Traffic Link**<br>**OT=Other Traffic Links** | **Earth Station Traffic Arrival Rate (pps) (149,338 x TR)** | **Satellite Processor Utilization** |
|---|---|---|---|
| 0.2370 | 83% (HTL) | 35,408 | 48.89% |
| 0.1052 | 37% (OT) | 15,710 | 21.69% |

*Table 8: Non-Uniform Traffic Distribution, High Load Levels*

| **Transmit Probability (TR)** | **Uplink Utilization**<br>**HTL=High Traffic Link**<br>**OT=Other Traffic Links** | **Earth Station Traffic Arrival Rate (pps) (149,338 x TR)** | **Satellite Processor Utilization** |
|---|---|---|---|
| 0.2857 | 100% (HTL) | 42,665 | 58.91% |
| 0.0857 | 30% (OT) | 12,798 | 17.67% |

All other source and destination combinations are given equal probabilities so as not to bias the high traffic link. Table 9,Table 10, and Table 11 outline the test scenarios used for the low, medium, and high loading levels.

*Table 9: Non-Uniform Traffic Distribution (Low Load)*

| Location | Transmit Probability | Destination Probabilities | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **Rio de Janeiro** | **Melbourne** | **Kansas City** | **Dhahran** | **Beijing** | **Berlin** | **Capetown** |
| Rio de Janeiro | 0.1428 | 0 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 |
| Melbourne | 0.1428 | 0.167 | 0 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 |
| Kansas City | 0.1429 | 0.100 | 0.100 | 0 | 0.500 | 0.100 | 0.100 | 0.100 |
| Dhahran | 0.1429 | 0.100 | 0.100 | 0.500 | 0 | 0.100 | 0.100 | 0.100 |
| Beijing | 0.1428 | 0.167 | 0.167 | 0.167 | 0.167 | 0 | 0.167 | 0.167 |
| Berlin | 0.1428 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0 | 0.167 |
| Capetown | 0.1428 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0 |

*Table 10: Non-Uniform Traffic Distribution (Medium Load)*

| Location | Transmit Probability | Destination Probabilities | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **Rio de Janeiro** | **Melbourne** | **Kansas City** | **Dhahran** | **Beijing** | **Berlin** | **Capetown** |
| Rio de Janeiro | 0.1052 | 0 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 |
| Melbourne | 0.1052 | 0.167 | 0 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 |
| Kansas City | 0.2370 | 0.100 | 0.100 | 0 | 0.500 | 0.100 | 0.100 | 0.100 |
| Dhahran | 0.2370 | 0.100 | 0.100 | 0.500 | 0 | 0.100 | 0.100 | 0.100 |
| Beijing | 0.1052 | 0.167 | 0.167 | 0.167 | 0.167 | 0 | 0.167 | 0.167 |
| Berlin | 0.1052 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0 | 0.167 |
| Capetown | 0.1052 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0 |

*Table 11: Non-Uniform Traffic Distribution (High Load)*

|  |  | **Destination Probabilities** |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| **Location** | **Transmit Probability** | **Rio de Janeiro** | **Melbourne** | **Kansas City** | **Dhahran** | **Beijing** | **Berlin** | **Capetown** |
| Rio de Janeiro | 0.0857 | 0 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 |
| Melbourne | 0.0857 | 0.167 | 0 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 |
| Kansas City | 0.2857 | 0.100 | 0.100 | 0 | 0.500 | 0.100 | 0.100 | 0.100 |
| Dhahran | 0.2857 | 0.100 | 0.100 | 0.500 | 0 | 0.100 | 0.100 | 0.100 |
| Beijing | 0.0857 | 0.167 | 0.167 | 0.167 | 0.167 | 0 | 0.167 | 0.167 |
| Berlin | 0.0857 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0 | 0.167 |
| Capetown | 0.0857 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0 |

### 3.3.8 Routing Algorithm Selection

Four routing algorithms are used in this research – self-healing Dijkstra, self-healing Bellman-ford, Darting and Extended Bellman-Ford. Both Darting and Extended Bellman-Ford represent real-world algorithms suitable for LEO networks and are analyzed in detail.

Both of the self-healing algorithms recompute the shortest paths upon a change in node connectivity. The change in connectivity is based upon updates that originate from SATLAB, which models the movement of the Iridium® satellite constellation. Both algorithms assume that the entire constellation of satellites has instantaneous knowledge the connectivity matrix. No overhead is introduced into the simulation by these type of algorithms. These self-healing algorithms provide a baseline for comparison against the Darting and Extended Bellman-Ford algorithms.

Darting introduces update packets into the network. The current node (the present location of the packet) compares the connectivity information encapsulated in the packet to its connectivity information. When the current node receives a packet from its predecessor node that contains outdated connectivity information, an update packet is generated and sent to the predecessor so that the predecessor's connectivity information can be updated. Otherwise, if the current node's connectivity information is outdated, the algorithm updates its connectivity information based upon the information encapsulated in the packet. Darting uses the Dijkstra routing algorithm to perform updates to the routing matrix based upon the new connectivity matrix information. Unlike the implementation by Janoso, this implementation of Darting does converge to an optimal solution by having complete topological information available to all nodes via the "self-healing" Dijkstra routing algorithm.

Extended Bellman-Ford has each node maintain the shortest path to all destinations via all ISL neighboring nodes. Each node in the constellation periodically sends routing matrix updates to these neighboring nodes, which in turn update their own constellation routing matrix. The algorithm, as implemented here, uses the "self-healing" Bellman-Ford to compute shortest paths based upon the updated vectoring information passed to each node. The information is passed to each node via an update packet similar to the one used in the Darting model.

### 3.3.9 Inter-Satellite Link (ISL) Connectivity

As discussed in Section 2.5.1, up to four ISLs are established by each Iridium® satellite and is neighboring satellites. As in [Fos98], each ISL antenna is assumed to have a mean pointing angle of 50 degrees and a steering range of 45 degrees, forming ISLs between approximately ± 60 degrees latitude.

This configuration, however, incorrectly allows ISLs to be established between satellites in the counter-rotating planes of the Iridium® constellation (planes 1 and 6). This flaw allows packets to take paths that are not representative of the real system. This impacts the number of hops and average packet delay encountered by certain source-destination node pairs.

To correct this problem, the BONeS DESIGNER primitive that computes the cost matrix is modified so that no ISLs are established between satellites in these counter-rotating orbital planes. The primitive is modified so that the "cost" links are set to infinity. Once the cost matrix is passed to the routing algorithm, the routing matrix calculates the proper shortest paths matrix, interpreting all infinity values as links that can not be traversed.

### 3.3.10 Network Access

As previously stated, each earth station in this model represents numerous mobile users that are present in the footprint of a satellite. Mobile users are susceptible to blocking effects from the local environment (i.e., buildings and terrain). The effects of this blocking are not modeled in this simulation because the primary metrics for performance assume that a user has gained access to the network.

### 3.3.11 Queue Size

The queue size for each satellite node is set to 4000, which allows the model to reject packets with a packet delay greater than 400 ms [Fos98]. Selecting a queue size greater than 4000 only increases the average delay experienced by packets traversing critical nodes.

### 3.3.12 Model Scaling

A primary goal of this research is to gain data on the performance of routing algorithms under high loading conditions. Janoso [Jan96] was unable to achieve loading levels greater than 20% because of overhead produced by both the model and the algorithm implementations. The model developed in [Fos98] is utilized as the basis for this simulation model. Simulation scaling techniques utilized in [Fos98] allow greater loading levels to be achieved, reducing the number of packets generated by the simulation and still maintaining an accurate average packet delay. The scaling factor used for this model is 10,000. The code for the algorithms was optimized in order to decrease simulation overhead and run-time.

Code optimization for both the Extended Bellman-Ford and Darting algorithms decreases the excessive simulation run-times encountered by Janoso [Jan96], allowing the simulation under higher loading levels. As shown in Table 12, the maximum run-time for any simulation barely exceeded 3 hours. Those run-times exceeding the mean were impacted by other user's programs operating on the same machine and utilizing up to 65% of its CPU time.

*Table 12: Simulation run-times*

| Traffic Pattern | Mean | Std. Deviation | Minimum | Maximum |
|---|---|---|---|---|
| Uniform Traffic | 0:55:48 | 0:02:10 | 0:14:16 | 3:01:03 |
| Non-Uniform Traffic | 0:38:14 | 0:08:38 | 0:15:15 | 1:57:39 |

### 3.3.13 Nodal Failure Method

The main thrust of removing satellites from the constellation is to analyze how the IRIDIUM® system responds when heavily loaded nodes are removed and how it impacts the performance of the routing algorithm. This methodology is used during the degraded performance measurement of the research. The method used to select which satellites to fail is described below and is conducted independently for both the uniform and non-uniform cases.

1. Generate packets and determine the paths from Kansas City to all others.
2. Count the number of packets that traverse each node.
3. Remove the satellite that has the most number of packets traversing it and that does not "disconnect the network"[9]. In the case of a tie, remove the satellite that has the greatest number of packets destined for Dhahran.
4. Remove the satellite that has the second most number of packets traversing it and that does not disconnect the network. In the case of a tie, remove the satellite that has the greatest number of packets destined for Dhahran.
5. Remove the satellite that has the third most number of packets traversing it and that does not disconnect the network. In the case of a tie, remove the satellite that has the greatest number of packets destined for Dhahran.
6. Repeat steps 1 through 4 to select the fourth and fifth satellites.
7. Repeat steps 1 through 4 to select the sixth and seventh satellites.

[9] A satellite that provides the only uplink for the earth station cannot be removed. The removal of this critical node would disconnect the earth station from the entire LEO satellite network and would prevent the earth station from transmitting and receiving packets.

### 3.3.14 Error Free Communication Links

Communication links are assumed to be error free since the primary focus of this research is the performance of the routing algorithms to route traffic through the network. Modeling errors and the recovery routines only serve to add another level of complexity to the model and increase simulation overhead, without gaining any additional insight into algorithm performance.

## 3.4 Model Design and Operation

Two tools were used to design the simulation, SATLAB and DESIGNER [Cad95, Cad94]. The SATLAB tool models the orbital movements of satellite constellations. Information associated with each satellite and earth station is placed into SATLAB, namely orbital parameters of the satellites, earth station coordinates, visibility information, and the overall simulation epoch used. DESIGNER is used to model and simulate communication networks. It accepts a variety of input parameters that characterize the operation of a network model. DESIGNER interfaces to SATLAB via a BONeS SATLAB Interface Module (BSIM), which passes current orbital information to the DESIGNER simulation. Thus, the Iridium® system can be accurately modeled and tested using a variety of routing algorithms. These tools allow metrics on each algorithm's performance in the Iridium® system to be analyzed and presented in a realistic environment.

Figure 2 is a depiction of the earth stations and satellites used in SATLAB for the initial configuration of the Iridium® constellation.

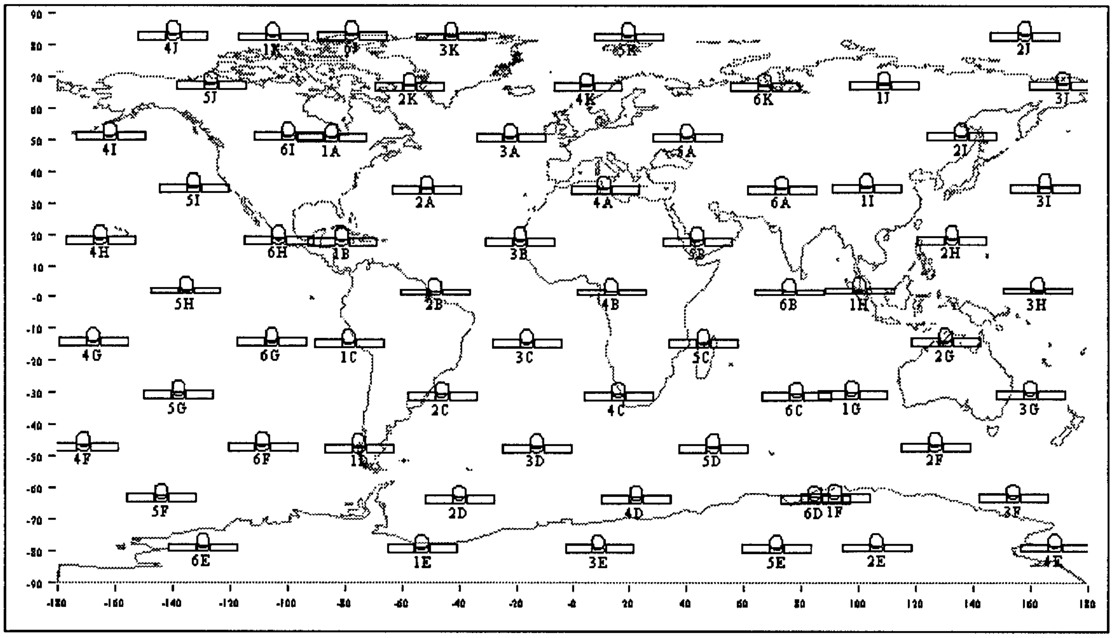

*Figure 2: SATLAB Map*

The DESIGNER model (Figure 3) shows the network simulation that is supported by the SATLAB simulation.

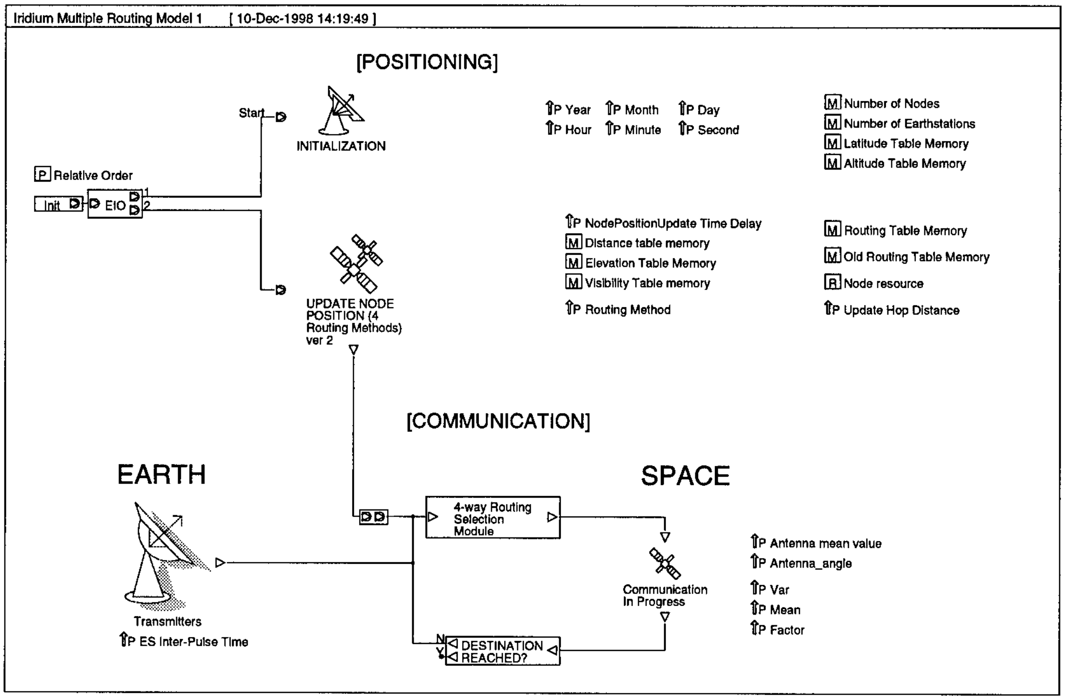

*Figure 3: DESIGNER Simulation Model*

There are two main sections of the top level DESIGNER system, the *Positioning* module and the *Communication* module. *The Positioning* module is responsible for receiving the updated constellation information from SATLAB. The *Communication* module models the network, including traffic generation, packet routing, and data collection.

Within the *Positioning* module, the *Update Node Position (4 Routing Methods ver 2)* sub-module triggers an update request from SATLAB at intervals based upon a user-defined input parameter (set to 30 seconds). The visibility information received by the *Positioning* module is stored in memory and is available to the *Communication* module.

The *Communication* module computes the routing matrix utilizing the selected routing algorithm from within *4-way Routing Selection* sub-module. The *Transmitters* sub-module is responsible for generating the traffic found within the network. The traffic is routed within the network by passing the algorithm type to the simulation model. The *4-way Routing Selection* sub-module uses the Current Node field, the Destination field, and the Algorithm Type to compute the Next Node field for the packet. All four algorithms utilized for generating the routing matrix are implemented in this module. The *4-way Routing Selection* sub-module also generates any update packets for the Darting and Extended Bellman-Ford algorithms. All packets are then passed to the *Communication in Progress* sub-module. This module determines whether the packet is passed to a satellite node or an earth station based upon the Current Node and Next Node fields. Packets are then passed to the *Destination Reached?* sub-module, which determines whether or not the packet has reached its destination or needs to be forwarded to the *4-way Routing Selection* sub-module for further routing. The packet iterates this cycle until it reaches its destination.

## 3.5 Model Input Parameters

This section outlines the parameters used in the DESIGNER and SATLAB tools used to model the Iridium® system and the routing algorithms. Both the values selected and the rational behind their selection are discussed.

### 3.5.1 DESIGNER Input Parameters

The simulation requires several parameters. The Epoch parameters (Month, Day, Year, Hour, Minute, Second) are selected to correspond to the same Epoch used in the SATLAB model for the satellite constellation. The starting date has no bearing on the simulation and was randomly set to June 1, 1998 at 0730. A Node Update parameter is used to define the frequency of updates retrieved from SATLAB via BSIM, and is set to 30 seconds based upon observations in [Fos98]. The scaling factor is set to 10,000.

The earth station inter-arrival time is used to set the traffic loading levels. These values are computed for the uniform traffic distribution case by dividing the network traffic generation rate defined in Table 4 into one. The results are then multiplied by the scaling factor (10,000). The corresponding inter-arrival times for these loads are found in Table 13.

*Table 13: Uniform Load Levels vs. inter-arrival*

| Loading Level | Network Traffic Generation Rate (pps) | Inter-Arrival Times |
|---|---|---|
| 100% (High) | 298,669 | 0.0335 |
| 83% (Medium) | 247,891 | 0.0403 |
| 50% (Low) | 149,338 | 0.0670 |

The inter-arrival time for the Non-Uniform case is fixed to 0.0670 while the transmit and destination probabilities are varied, as discussed in Section 3.3.7, to achieve the needed loading levels for the Kansas City to Dhahran link.

### 3.5.2 SATLAB Input Parameters

Each of the 66 satellites modeled in SATLAB had the following information entered: ID, Health, Semi-Major Axis, Eccentricity, Right Ascension, Inclination, Mean Anomaly, Argument of Perigee, and Epoch Group; these parameters characterize any satellite constellation. The ID parameter identifies a satellite using a two-field identifier. The first field indicates the orbital plane (1-6) and the second field identifies one of the eleven satellites in the orbital plane (A-K). The Health parameter is set to 1 when the satellite if fully functional. During the simulation trials where satellites are removed from the constellation, the Health parameter for these satellites is set to 0. The Semi-Major Axis field is set to the orbital altitude of the satellite constellation, which is 7,158 km. Eccentricity for each satellite is set to 0, as is the Argument of Perigee parameter. Inclination is set to 86. The Right Ascension and Mean Anomaly parameters vary based upon the position of the satellite in its orbital plane. Each satellite belonged to the same Epoch Group, which corresponds to June 1, 1998 at 0730.

The parameters used to define each earth station are latitude, longitude, altitude, and elevation. The minimum elevation angle for each earth station is set to 8.2 degrees.

## 3.6 Performance Metrics

The performance metrics measured during simulation trials are protocol convergence time, average packet delay, rejected packet counts, average hop count, and protocol overhead. The metrics are used to analyze the performance of each algorithm under a variety of conditions, to include the following: uniform and non-uniform traffic distributions; low loading, medium loading, and high loading conditions; and in the presence of nodal failures. The following subsections further define these metrics.

### 3.6.1 Protocol Convergence Time

This metric is used to determine how fast the routing algorithm can converge to an optimal solution for a given cost matrix that must be processed. The quicker the convergence rate, the more capable the routing algorithm is in efficiently routing traffic through the network. Algorithms that converge faster allow frequent updates to the routing matrix to occur without significantly impacting network performance.

This metric is calculated by determining how long it takes for update packets to reach their destination during each 30 second interval. In the case of the Extended Bellman-Ford algorithm, the update packets are generated at the beginning of a topology change and are generated for every node in the constellation. With the Darting algorithm, updates are generated only when data packets encounter nodes with new or outdated routing information. Those nodes that are not in a path do not receive or generate update packets.

### 3.6.2 Average Packet Delay

This metric is the total amount of time it takes all packets to traverse the network for a particular source/destination combination. The threshold for this metric is 400 ms, the time constraint that supports real-time voice communications. Delays that exceed 400 ms are considered unacceptable, however, the average performance of all packets for a specific source/destination combination is the primary metric. An average packet delay of 400 ms or less is considered acceptable performance, even though some individual measurements may exceed this value.

### 3.6.3 Rejected Packets

This metric is the total number of packets that are rejected because they exceeded the 400 ms average packet delay criteria. Loading level, traffic distribution, and number of failed nodes are expected to impact this metric. Any ratio of rejected packets to total packets for a particular source/destination link that exceeds 1% is considered unacceptable performance.

### 3.6.4 Average Hop Count

This metric is the total number of hops that a packet must make from source to destination while traversing the network. This is a descriptive metric. The average hop count is compared against the average packet delay in an effort to rate the algorithm's performance.

### 3.6.5 Routing Protocol Overhead

Both Darting and Extended Bellman-Ford introduce overhead into the network. This overhead is the total number of update packets introduced into the network to facilitate connectivity updates to individual nodes. The more update packets generated the greater the possibility of congestion in the network. In general, lower overhead indicates better performance.

## 3.7 Model Verification and Validation

The simulation presented in this research is verified and validated to insure that the model is representative of the Iridium® system and that each algorithm is correctly implemented. The process of verifying and validating the model includes several trial runs of the model. During each trial, data is collected to insure that each module within the model operates correctly and accurately depicts the Iridium® system and routing algorithms in use.

### 3.7.1 Model Verification

Each module is checked for inconsistencies, construction errors, and dependencies by the DESIGNER tool itself. This is a built-in function of DESIGNER that takes place each time the module is saved and verified. This was the first level of model verification used.

The next level used is the independent testing of each module. This involves the testing of individual modules and primitives that are created and implemented in the baseline model.

During the initial testing of the baseline simulation model, it was verified that ISLs are established between orbital planes one and six. This activity does not take place in the Iridium® constellation. This activity was further verified by running a simulation and tracing packets from their source to destination. It was found that packets routinely passed between these orbital planes. Consequently, the Cost Matrix Primitive was corrected to reflect the proper behavior between these planes (i.e., no crosslinks were established between orbital planes one and six). Post-testing indicated that the correction was functional, properly modeling the true behavior of ISLs between these two planes. Once corrected, the update packet generation module of Darting was verified for functionality.

The Extended Bellman-Ford algorithm is implemented as a two step process, which is mirrored during verification. First, the C code is developed for a "self-healing" Bellman-Ford algorithm. It is then tested in a C development environment. Once the operation is verified, the C code is moved to the DESIGNER environment, implemented, and verified. The routing table generated from the EXBF algorithm is compared against the one created by the Dijkstra algorithm. Both tables are identical, verifying proper operation of the EXBF primitive. The update packet mechanism is finally implemented and verified to be correct during similar trials.

The final level of verification involves the complete testing of the entire model. During this phase, the entire model is simulated using a low loading level. The performance of each routing protocol is verified such that each converged to a solution. The modular design and top-down implementation of the simulation facilitated the verification process.

### 3.7.2 Model Validation

Although the Iridium® system is fully deployed, no operational statistics or performance information was available during this research. Consequently, it is impossible to validate the key aspects of the model against real system measurements.

Whenever possible, the proposed Iridium® specifications are used in the model. Any assumptions and input parameters used to design the model are chosen based upon the expert intuition of the thesis advisor or are verified against theoretical values presented in [Fos98].

## 3.8 Summary

The methodology used to develop the simulation which tests a variety of routing algorithms in the Iridium® system was presented in this chapter. In addition, many simplifying assumptions and limits placed on the model were detailed in this chapter. Methods used to verify and validate the model were explained. The results of these simulations and their analysis are presented in Chapter 4.

# 4. ANALYSIS

## 4.1 Introduction

This chapter provides an analysis of the simulation results. Section 4.2 discusses the statistical accuracy of the data along with an explanation into the expected variances within each metric. Sections 4.3, 4.4, and 4.5 present the various simulation scenarios used to analyze the performance of each algorithm. Section 4.6 presents the performance metrics and the analysis of the data, followed by a comparative analysis in Section 4.7. This chapter concludes with an overall assessment of each algorithm's performance in the Iridium® system.

## 4.2 Statistical Accuracy

Each algorithm is tested under a variety of scenarios. Three different random number generator seeds are used for each simulation in order to guarantee the end-to-end delay results are not causally effected by the Poisson traffic generator. To accomplish this, three independent sets of data for each test scenario are collected for analysis.

The sample end-to-end delay and hop count means are calculated for metrics from Kansas City to all other earth stations. Sample algorithm convergence time, overhead, and packet rejection rate means are also calculated. Each data set is used to calculate an average mean and standard deviation for the metric.

A 95% confidence interval is then calculated for the end-to-end delay metric using the three different means and standard deviations found for each test scenario. This is accomplished using the *student's t-distribution* (Equation 1), where the *100(1-α)* is the confidence interval, $\bar{x}$ is the average of the three sample means, *s* is the average of the three standard deviations corresponding to the three sample means, *n* is the number of sample means, and *t* is the *student's t-distribution* [Jai91].

$$100(1-\alpha)\%\, CI \quad = \quad \overline{x} \pm t\,[1-\alpha;n-1]\,s/\sqrt{n} \qquad (1)$$

Utilizing the above equation yields 95% confidence intervals where the mean end-to-end delay range is less than 24.905-ms (±12.452-ms) for each earth station in Table 14. This interval increases significantly during the removal of heavily loaded nodes, and is greatest in the *Uniform* test scenario with five satellites removed. In this scenario, the mean end-to-end delay range increases to 251.418-ms (±125.709-ms), as shown in Table 15. The larger interval results when the heavily loaded satellites are removed from the constellation. This forces packets travelling from Kansas City to Rio to traverse a longer path. In this scenario, the path from Kansas City to Rio averages 171.371-ms. The removal of key satellites also causes packets to travel longer paths, resulting in a larger range. Further analysis of the 95% confidence ranges for each case (Appendix A) reveals that the 95% confidence range exceeded 50-ms (±25-ms) only 6.7% of the time. Therefore, running three simulations with three independent seeds provides a sufficient data set. As such, any end-to-end delay values referenced in this chapter will represent the average mean end-to-end delay values depicted in the tables found in Appendix A.

*Table 14: Results for Non-Uniform Low Load with a Full Constellation*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 131.996 | 0.553 | 130.621 | 133.371 | 2.750 |
| | Melbourne | 149.372 | 1.891 | 144.674 | 154.071 | 9.397 |
| | Beijing | 129.332 | 0.973 | 126.915 | 131.748 | 4.833 |
| | Capetown | 160.240 | 3.826 | 150.735 | 169.746 | 19.011 |
| | Rio | 101.338 | 0.881 | 99.150 | 103.526 | 4.377 |
| | Berlin | 131.996 | 0.553 | 130.621 | 133.371 | 2.750 |
| | | | | | | |
| Darting | Dhahran | 130.211 | 0.830 | 128.148 | 132.274 | 4.126 |
| | Melbourne | 147.274 | 1.425 | 143.735 | 150.813 | 7.078 |
| | Beijing | 127.498 | 0.759 | 125.612 | 129.383 | 3.770 |
| | Capetown | 156.507 | 5.012 | 144.056 | 168.959 | 24.904 |
| | Rio | 99.717 | 0.556 | 98.335 | 101.098 | 2.764 |
| | Berlin | 130.211 | 0.830 | 128.148 | 132.274 | 4.126 |

*Table 15: Results for Uniform High Load with 5 Satellites Out*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 195.086 | 9.115 | 172.442 | 217.729 | 45.287 |
| | Melbourne | 198.213 | 7.973 | 178.407 | 218.020 | 39.613 |
| | Beijing | 208.003 | 49.412 | 85.246 | 330.760 | 245.514 |
| | Capetown | 248.175 | 13.270 | 215.207 | 281.143 | 65.936 |
| | Rio | 171.371 | 50.601 | 45.662 | 297.080 | 251.418 |
| | Berlin | 195.086 | 9.115 | 172.442 | 217.729 | 45.287 |
| | | | | | | |
| Darting | Dhahran | 210.062 | 5.716 | 195.862 | 224.262 | 28.400 |
| | Melbourne | 207.341 | 4.473 | 196.229 | 218.452 | 22.223 |
| | Beijing | 236.172 | 0.566 | 234.765 | 237.580 | 2.815 |
| | Capetown | 259.576 | 6.354 | 243.790 | 275.362 | 31.573 |
| | Rio | 138.501 | 1.761 | 134.127 | 142.874 | 8.748 |
| | Berlin | 210.062 | 5.716 | 195.862 | 224.262 | 28.400 |

## 4.3 Uniform Traffic Simulation Scenarios

This section outlines scenarios that test the performance of the various algorithms. As stated previously, each test scenario was run using three unique seeds. Table 5 provides the uniform traffic distribution used for these cases.

Each scenario is run with a full constellation of satellites, as well as, with three, five, and seven non-operational satellites. Non-operational satellites are algorithmically selected for removal based upon the method outlined in Section 3.3.13. These satellites are non-operational at the beginning of the simulation and remain non-operational throughout.

### 4.3.1 Uniform Traffic Distribution, Low Load

This test case provides a baseline for comparison against the other test scenarios defined later in this section. This scenario simulates a network operating under minimal load. Both traffic distribution and loading level are low.

As outlined in Table 4, this test scenario represents the *Low Load* case (50%). Each of the seven earth stations generates 21,334 packets-per-second, which corresponds to a network arrival rate of 149,338 packets-per-second. As such, the satellite processor utilization rate translates to 29.46%. With the exception of convergence time and overhead, little difference is expected between the performance metrics of each algorithm.

This scenario also provides the basis for removing the "self-healing" algorithms from the comparative analysis. Both the mean end-to-end delays for these algorithms are expected to be less than the Extended Bellman-Ford and Darting algorithms.

### 4.3.2 Uniform Traffic Distribution, Medium Load

This test case simulates a network operating under nominal usage where both traffic distribution and loading level are moderate. It represents the *Medium Load* case (83%). Each of the seven earth stations generates 35,413 packets-per-second, which corresponds to a network arrival rate of 247,891 packets-per-second (Table 4). This translates to a satellite processor utilization rate of 48.89%. With the exception of convergence time and overhead, subtle differences between the performance metrics of each algorithm should emerge due to the effects of queuing within each node. Overhead is expected to remain at the levels seen in the low loading test case for the Extended Bellman-Ford algorithm since the number of update packets is independent of traffic load. A slight increase in overhead is expected for the Darting algorithm since the amount of overhead that occurs with Darting is associated with the number of nodes traversed by packets.

### 4.3.3 Uniform Traffic Distribution, High Load

This test simulates the *High Load* case (100%) where both traffic distribution and loading level are high. Each of the seven earth stations generates 42,667 packets-per-second, corresponding to a network arrival rate of 298,669 packets-per-second (Table 4). This translates to a satellite processor utilization rate of 58.91%.

Under this scenario, queuing delay is expected to significantly impact the performance metrics of each algorithm. Likewise, a greater variance in the metrics is expected.

## 4.4 Non-Uniform Traffic Simulation Scenarios

The following simulation scenarios represent "real-world" test cases for the algorithms. The traffic generated under these scenarios is a more realistic non-uniform distribution.

In order to model traffic arriving in a non-uniform fashion, two earth stations were selected to receive most of the traffic that occurs in the network. These earth stations were Kansas City and Dhahran. The corresponding transmit probabilities and traffic arrival rates for each case are found in Table 6, Table 7, and Table 8. The uplink utilization rates in this table are shown for all links.

### 4.4.1 Non-Uniform Traffic Distribution, Low Load

Utilizing the traffic distribution depicted in Table 9, this test scenario models two earth stations utilizing 50% of their uplink capacity. All other earth stations utilize 49% of their capacity. Network traffic arrival rate is 149,338 packets-per-second. This *Non-Uniform* test scenario represents minimal network traffic occurring between the two earth stations. Traffic is increased between these earth stations in subsequent scenarios. As shown in Table 6, the satellites servicing these earth stations experience a 29% loading level.

### 4.4.2 Non-Uniform Traffic Distribution, Medium Load

This test scenario models the two earth stations utilizing 83% of their uplink capacity, as shown in Table 7. The corresponding traffic distribution for this test case corresponds to moderate traffic load occurring between Kansas City and Dhahran (Table

10). Both of these earth stations transmit 35,408 packets-per-second, while the remaining five transmit 15,710 packets-per-second. The majority of all traffic in the network is destined for either Kansas City or Dhahran.

### 4.4.3 Non-Uniform Traffic Distribution, High Load

This test scenario models two earth stations utilizing 100% of their uplink capacity to simulate a heavily used link (Table 8). The traffic distribution for this test case corresponds to Table 11. The link between Kansas City and Dhahran is heavily utilized, while all other links are only nominally utilized.

## 4.5 Simulation Scenarios with a Degraded Constellation

Degraded constellation scenarios permit algorithm evaluation under adverse conditions. Each *Uniform and Non-Uniform* test case are tested in the presence of satellite failures. Satellites are declared non-operational according to the methodology outlined in Section 3.3.13. The satellites are non-operational during the entire simulation. Three, five, and seven satellites are removed from each case to examine how the algorithms perform with a degraded constellation.

## 4.6 Analysis of Performance Metrics

### 4.6.1 Analysis of mean end-to-end delay

The primary metric for analyzing performance is the mean end-to-end delay of packets traversing the network. The ability of the algorithm to route packets through the network while maintaining a mean end-to-end delay of 400-ms is necessary to support real-time voice communications. As stated previously, the simulation was designed so packets exceeding the 400-ms end-to-end delay criteria would be rejected from the system. Consequently, the mean end-to-end delay measured from Kansas City to all other earth stations was less than 400-ms for all test scenarios described in Sections 4.3, 4.4, and 4.5. Both end-to-end delay and packet rejection rates are impacted by the capacity of each on-board satellite queue. Increasing the queue size would reduce the packet rejection rate, but increase the end-to-end packet delay. This inverse relationship has the opposite effect when the queue size is reduced. Reducing the queue size would increase the packet rejection rate and decrease the end-to-end delay of the packets. As demonstrated by this simulation, queue size is especially important when the network doesn't perform any type of load balancing at the node level.

#### 4.6.1.1 Bellman-Ford and Dijkstra

The Bellman-Ford and Dijkstra algorithms are considered self-healing algorithms. Self-healing algorithms represent ideal routing protocols; they generate no overhead and maintain the shortest-paths connectivity matrix. This fact alone impacts the end-to-end delay of packets traversing the network and results in skewed data. At 50% uplink utilization, the end-to-end packet delays were within 10-μs of each other for each algorithm. However, the difference grows as uplink utilization increases, as shown in Figure 4. When uplink utilization is increased to 83%, packets travelling from Kansas City to Capetown experience an end-to-end delay that is 10.401-ms to 26.285-ms longer than the "self-healing" algorithms. The Extended Bellman-Ford packets had a 12.90% greater end-to-end delay than the "self-healing" algorithms, while the Darting packets had an end-to-end delay that was 5.10% greater. These results justify removal of the "self-healing" algorithms from further comparative analysis. All subsequent analysis will focus on the Extended Bellman-Ford and Darting algorithms discussed earlier.

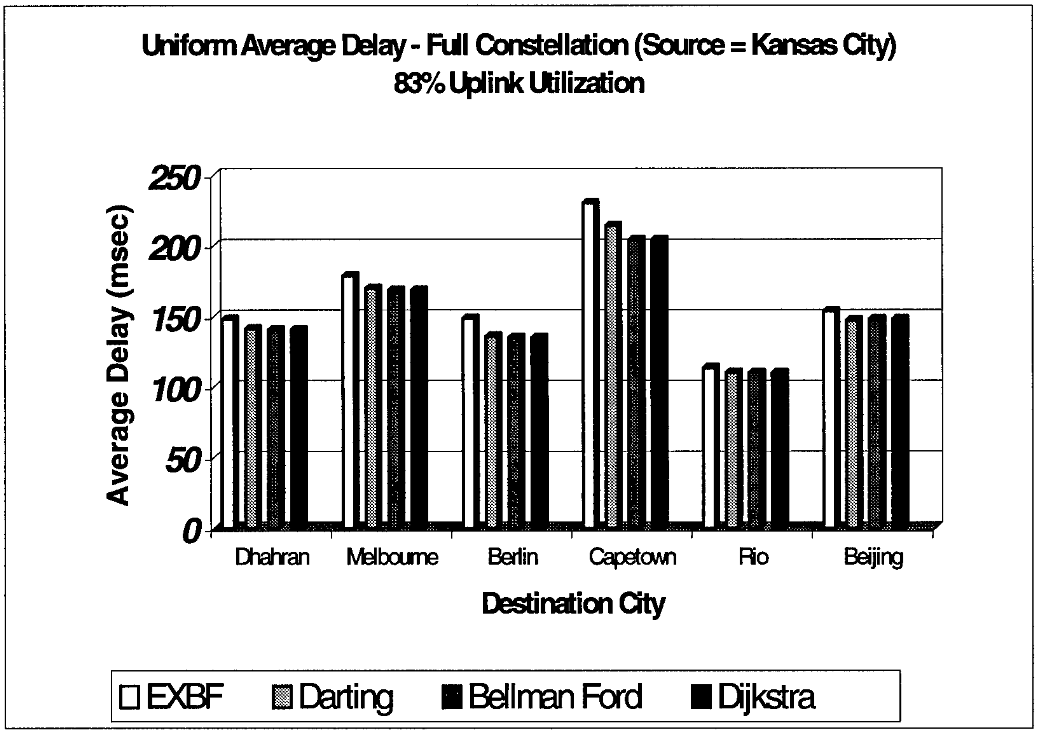

*Figure 4: Uniform Average Delay, Full Constellation, 83% Uplink Utilization*

### 4.6.1.2 Extended Bellman-Ford Algorithm

The lowest mean end-to-end packet delays occurred between Kansas City and Rio de Janeiro in both *Uniform* and *Non-Uniform Low Load* scenarios with a full satellite constellation. They measured 98.43-ms and 101.34-ms, respectively.

The highest mean end-to-end packet delay, 277.24-ms, occurred between Kansas City and Capetown under the *Uniform High Load* test scenario with a full constellation. In contrast, the highest mean end-to-end packet delay that occurred under the *Non-*

*Uniform* test scenarios occurred between Kansas City and Beijing and measured 235.38-ms under the *High Load* scenario with three satellites removed.

### 4.6.1.3 Darting Algorithm

As was the case with the Extended Bellman-Ford algorithm, the lowest mean end-to-end delays occurred between Kansas City and Rio de Janeiro in both *Uniform* and *Non-Uniform Low Load* scenarios with a full satellite constellation. They measured 98.44-ms and 99.72-ms, respectively.

Another similar finding is that the highest mean end-to-end delay occurred between Kansas City and Capetown under the *Uniform High Load* test scenario with a full constellation and measured 271.88-ms. Likewise, the highest mean end-to-end delay that occurred under the *Non-Uniform* test scenarios was between Kansas City and Beijing, measuring 237.19-ms under the *High Load* scenario with three satellites removed.

### 4.6.1.4 Average Delay Trends

Several trends emerged based upon the input parameters used to define various test scenarios. These trends were found in both the Extended Bellman-Ford and Darting algorithm results.

First, non-operational satellites had a significant impact on both queuing delay and end-to-end delay. Figure 5 shows the mean end-to-end packet delays of packets travelling from Kansas City to all other earth stations with zero, three, five, and seven non-operational satellites. In 72% of the cases, the mean end-to-end packet delay decreased as non-operational satellites were added. In the remaining 28% of the cases, there was an increase in the mean end-to-end packet delay. This trend was observed across in all scenarios where satellites were removed. Figures similar to the one below can be found in Appendix B.

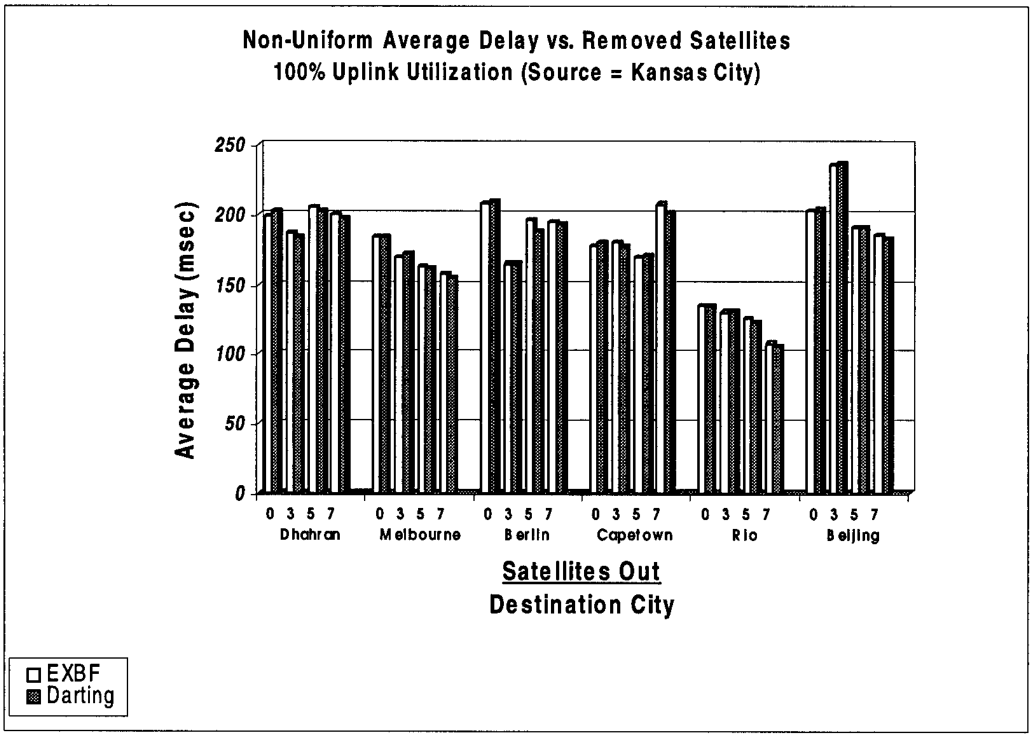

*Figure 5: Non-Uniform Average Delay vs. Removed Satellites, 100% Uplink Utilization*

This trend results from the heaviest loaded satellites being removed from the network, resulting in the removal of satellites that introduce the greatest amount of queuing delay. Upon their removal, shorter delay paths were found for the majority of source/destination pairs (clearly indicated in the Kansas City to Melbourne path in Figure 5). The removal of satellites reduced queuing delay by acting as a load-balancing mechanism, redistributing packets to other satellites.

An increase in mean end-to-end packet delay can occur when longer paths result from the removal of satellites. This trend is most noticeable in the path from Kansas City to Beijing, Berlin, and Dhahran in Figure 6. For these paths, the removal of the first five satellites resulted in longer delays due to an increase in path lengths when nodes are removed.

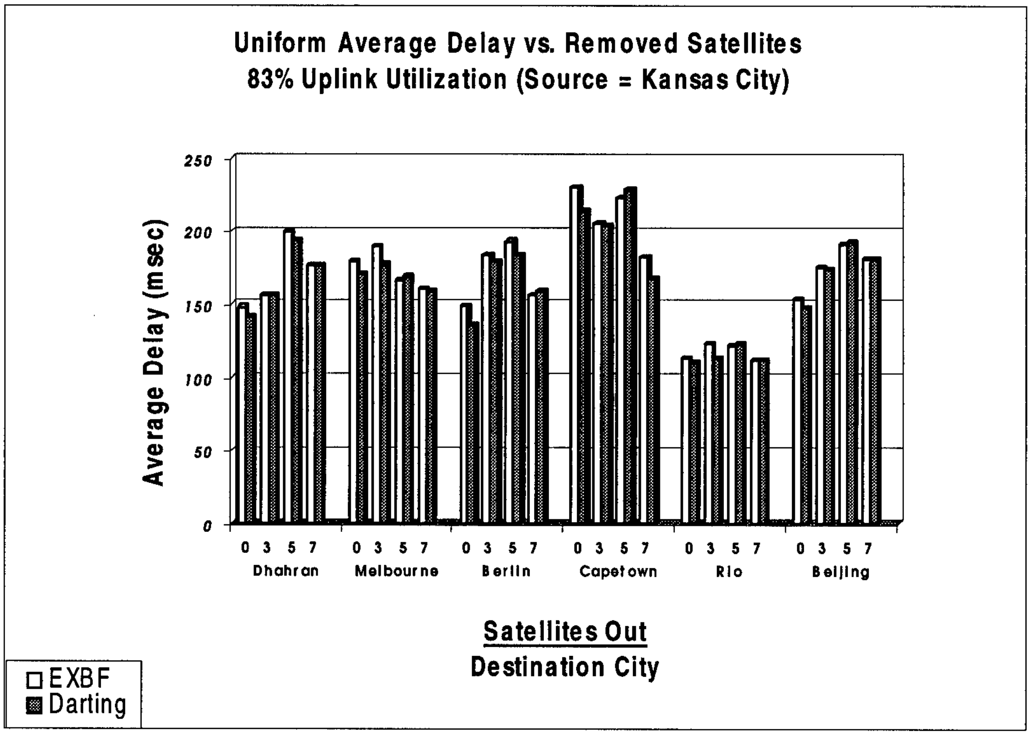

*Figure 6: Uniform Average Delay vs. Removed Satellites, 83% Uplink Utilization*

The second trend that was observed was that increasing the loading level in each simulation scenario increased the mean end-to-end packet delay for most cities (Figure 7). This was the case for both algorithms, and the greatest increase occurred when uplink utilization increased from 50% to 83%.

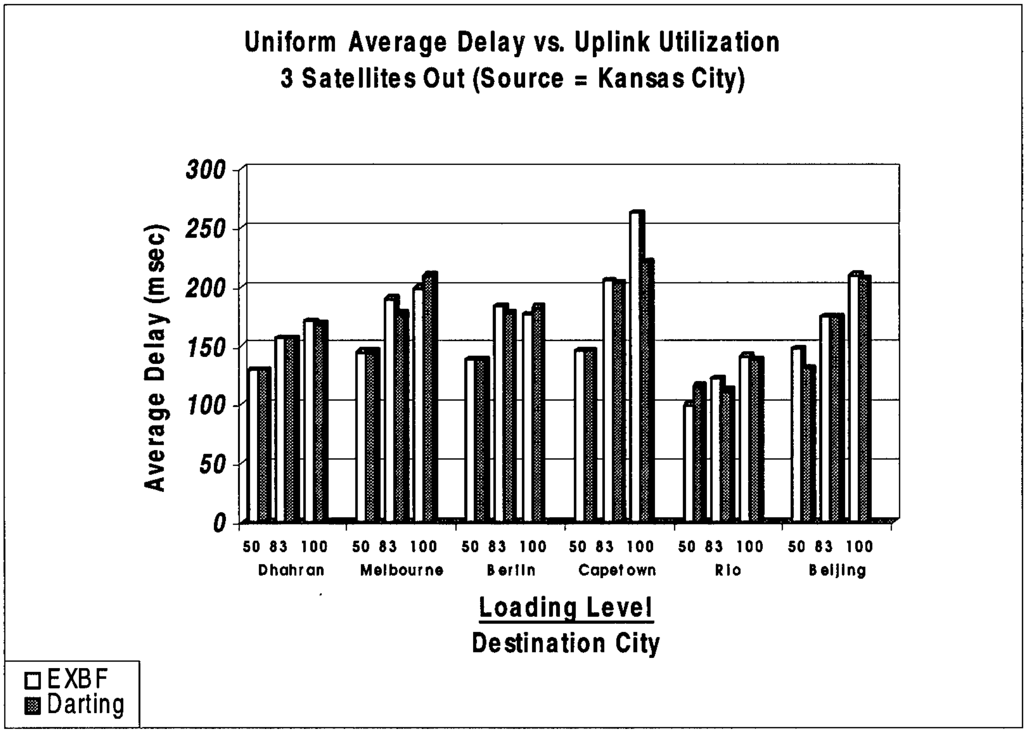

*Figure 7: Uniform Average Delay vs. Uplink Utilization, 3 Satellites Removed*

In most cases, delay continues to increase when the uplink utilization transitions from 83% to 100%. However, some delays also decrease during this transition, as was the case with packets traveling from Kansas City to Berlin utilizing the Extended Bellman-Ford algorithm. This is due to an increase in the number of packets rejected by satellites along the path when their end-to-end packet delay exceeds 400-ms.

The final trend that was observed for the end-to-end packet delay metric is drawn when a comparison is made between the algorithms. The average difference in mean end-to-end delays between these algorithms was 6.62-ms in the *Uniform* case and 2.58-ms in the *Non-Uniform* case. The maximum differences were 41.98-ms and 12.99-ms, respectively. Although the average difference in algorithm delays is small, a closer inspection of the data reveals that the Darting mean delays were lower than those of the

Extended Bellman-Ford algorithm for the majority of all *Uniform* test cases. However, this was not the case when both algorithms performed equally well in the 100% uplink utilization test case (Figure 8). As shown in Figure 9, the majority of mean delays for Darting were lower for all *Non-Uniform* test cases.

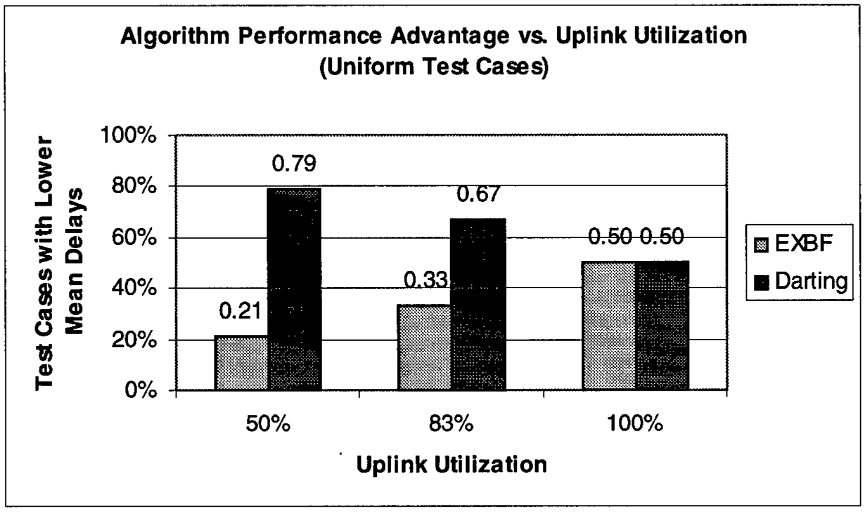

*Figure 8: Comparison of Algorithm Delay Performance, Uniform Test Cases*

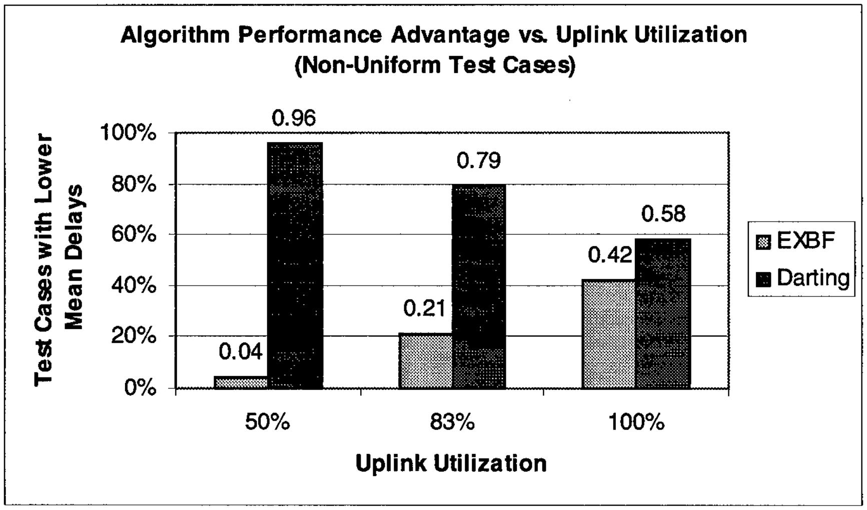

*Figure 9: Comparison of Algorithm Delay Performance, Non-Uniform Test Cases*

### 4.6.2 Analysis of Rejection Rate

Rejected packets occur when end-to-end packet delay exceeds 400-ms. As such, rejection rate becomes the primary metric for determining overall algorithm performance. A rejection rate of 1% or less is considered acceptable performance (Section 3.6.3). In each case where rejected packets were present, the rejection rate exceeded 1%.

Packet rejections did not occur in any *Low Load* test scenario. They only occurred when the network was exposed to *Medium* and *High Load* traffic. This emphasizes the importance of evaluating network performance at higher loading levels.

Packet rejection rate varied from 0% to 8.12% in *Uniform* test scenarios, with rejections occurring only in the *High Load* test cases (Table 16). The worst rejection rates were 8.12% for the Darting algorithm and 7.18% for the Extended Bellman-Ford algorithm. Both occurred in the *High Load* test case with five satellites removed.

*Table 16: Uniform Rejection Rates*

| **Satellites removed** | **Rejection Rate (Low Load)** | | **Rejection Rate (Medium Load)** | | **Rejection Rate (High Load)** | |
|---|---|---|---|---|---|---|
| | **EXBF** | **Darting** | **EXBF** | **Darting** | **EXBF** | **Darting** |
| 0 | 0 | 0 | 0 | 0 | 2.46% | 2.57% |
| 3 | 0 | 0 | 0 | 0 | 3.10% | 1.38% |
| 5 | 0 | 0 | 0 | 0 | 7.18% | 8.12% |
| 7 | 0 | 0 | 0 | 0 | 0 | 2.43% |

In the *Non-Uniform* test scenarios, packet rejection rate varied from 0% to 28.81% with rejections only occurring during *Medium* and *High Load* tests using a full constellation (Table 17). The greatest rejection rates occurred during the *High Load* test case with no satellites removed, reaching 28.81% under the Extended Bellman-Ford algorithm and 27.16% under the Darting algorithm.

*Table 17: Non-Uniform Rejection Rates*

| Satellites removed | Rejection Rate (Low Load) | | Rejection Rate (Medium Load) | | Rejection Rate (High Load) | |
|---|---|---|---|---|---|---|
| | **EXBF** | **Darting** | **EXBF** | **Darting** | **EXBF** | **Darting** |
| 0 | 0 | 0 | 6.17% | 3.03% | 28.81 | 27.16% |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 |

In the *High Load Uniform* scenario, the difference between the algorithm's rejection rate is a function of how the update packets are generated and which nodes receive the updates. The updates generated by the Darting algorithm increase packet queuing delay beyond the 400-ms criteria in cases where Darting's rejection rate was greater and vice versa.

### 4.6.2.1 Rejection Rate trends

The first apparent trend from the data presented in Table 16 and Table 17 is that traffic distribution impacts rejection rate. This is illustrated when the full constellation rejection rates of the *Uniform* test cases are compared to the *Non-Uniform* test cases. No packets are rejected under the *Medium Load Uniform* case while 3.03% to 6.17% of packets are rejected in the *Medium Non-Uniform* test case. This trend continues into the High Load test case, with 2.46% to 2.57% of packets rejected under the *Uniform* test case and 27.16% to 28.81% rejected under the *Non-Uniform* test case. A higher number of packets are rejected under *the Non-Uniform* case than are rejected in the *Uniform* case, regardless of the algorithm used.

Another trend is that as satellites are removed from the constellation, packet rejection rates drop to 0% in the *Medium and High Non-Uniform* scenarios, but continue to fluctuate between 0% and 8.12% under the same *Uniform* scenarios. This is a consequence of the methodology used to remove satellites from the constellation (See Section 3.3.13). Removal of the highest loaded satellites from the constellation has a greater impact on the packets in the *Non-Uniform* case because a majority of traffic occurs between Kansas City and Dhahran. Therefore, when those satellites are removed, all nodes whose queues are causing packets to exceed the 400-ms end-to-end delay criteria are also removed. In comparison, not all nodes of this type are removed in *Uniform* test cases, and packets continue to exceed the 400-ms end-to-end delay criteria. After removal of these nodes, the new alternate paths include more heavily traveled nodes, causing the rejection rate to increase when satellites are removed. This is best seen in Table 16, where the packet rejection rate initially increases from 2.46% to 7.18% but then drops to 0%.

### 4.6.3 Overhead

Overhead is the total number of update packets introduced into the network to facilitate connectivity updates to individual nodes. The more update packets generated the greater the possibility of congestion in the network. Overhead is calculated by dividing the total network traffic into the total number of update packets generated by the algorithm. In general, lower overhead indicates better performance.

As expected, Darting generated a significantly lower amount of overhead traffic than Extended Bellman-Ford. Overhead averaged 1.57% to 5.36% and 20.12% to 37.17% for Darting and Extended Bellman-Ford under the *Uniform* scenarios, respectively (Figure 10).

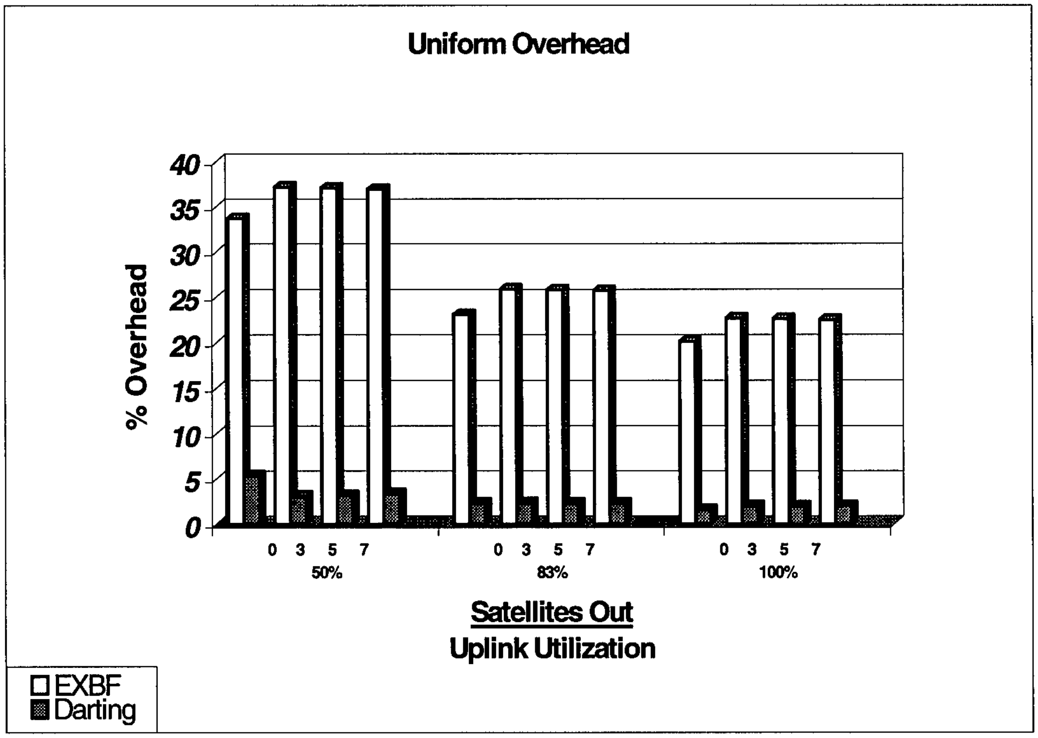

*Figure 10: Uniform Overhead*

This trend continued in the *Non-Uniform* case where Darting overhead varied between 3.29% to 3.8% and Extended Bellman-Ford overhead varied between 31.67% and 37.14% (Figure 11). Data for Figures 10 and 11 is found in Appendix C.

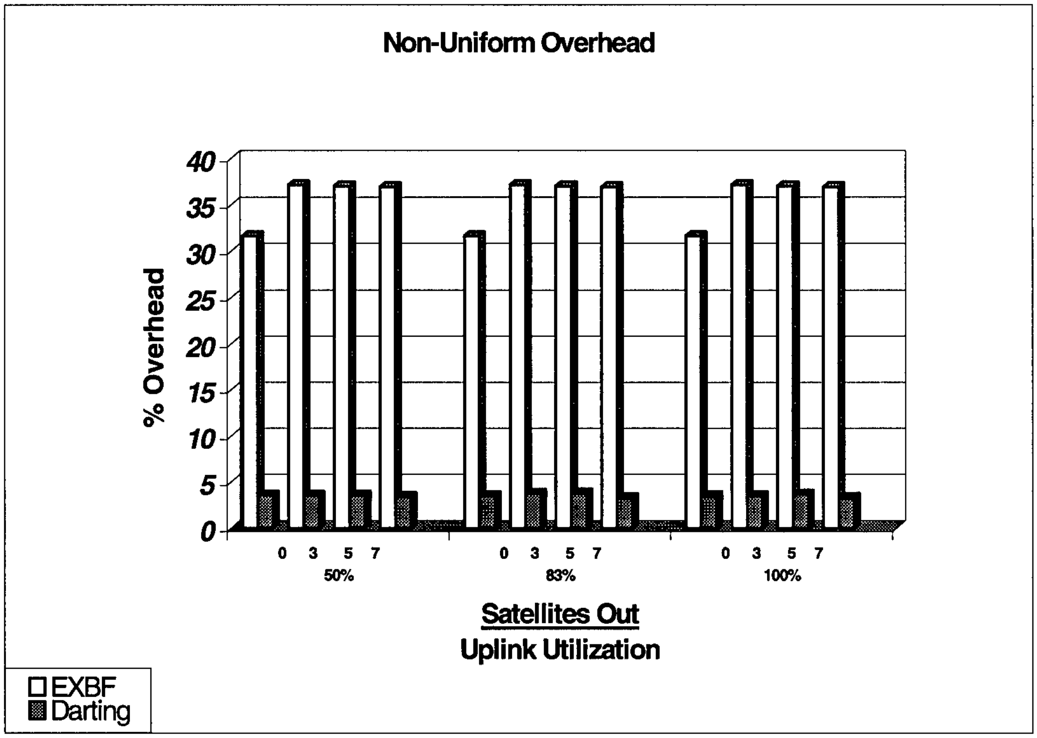

*Figure 11: Non-Uniform Overhead*

### 4.6.3.1 Overhead Trends

The most significant trend illustrated by Figure 10 and Figure 11 is that Darting produces 18.55% to 34.06% less overhead in the *Uniform* scenario and 28.08% to 33.65% less overhead in the *Non-Uniform* scenario. The increased overhead of the Extended Bellman-Ford algorithm occurs because all nodes in the network are updated during each 30-second simulation interval. In contrast, the Darting algorithm only updates those nodes being traversed during the same interval. Since the amount of overhead contributes to overall rejection rate and mean end-to-end delay metrics of each algorithm, the Darting algorithm gains an advantage by producing less overhead. However, the advantage gained by Darting is perhaps misleading given the fact that only seven earth stations were modeled in the simulation scenarios. As more earth stations are modeled, an increased number of satellites are needed to establish paths from each location, thereby increasing the amount of overhead produced by the Darting algorithm. This is not the case with the EXBF algorithm since all nodes are updated simultaneously.

### 4.6.4 Convergence

The convergence metric yielded mixed results. In 75% of all *Uniform* test scenarios, the Darting algorithm converged 9.47% to 40.39% faster than the Extended Bellman-Ford algorithm. In the remaining 25% of the cases, the Extended Bellman-Ford algorithm converged 10.27% to 149.02% faster (Figure 12, Table 44). This trend is repeated when the data is restricted to *the Uniform Medium and High Load* cases, with Darting converging faster than the Extended Bellman-Ford algorithm in 75% of the cases by the same margin.

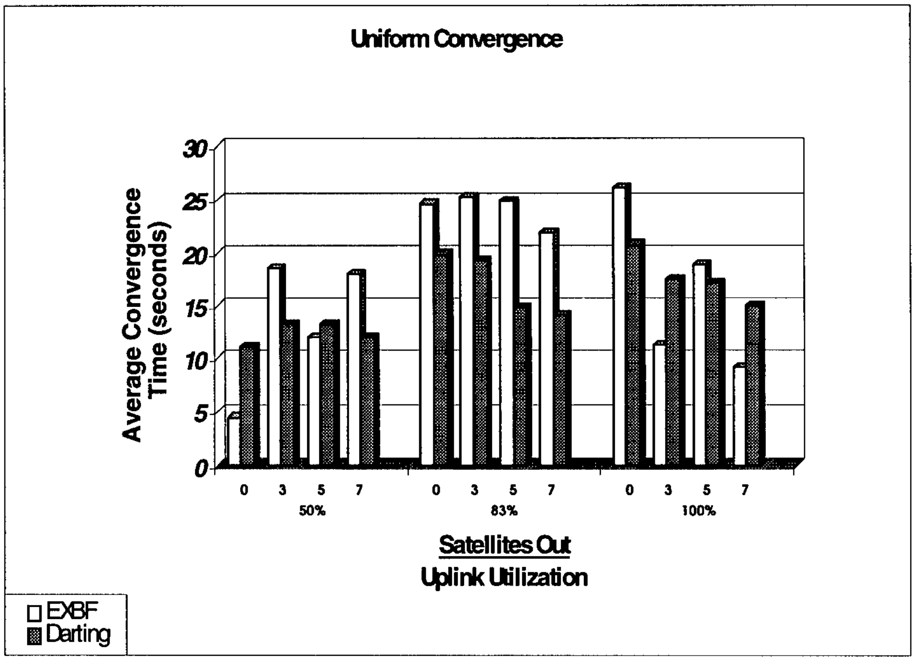

*Figure 12: Uniform Convergence*

This trend was reversed in the *Non-Uniform* test scenarios. Under these scenarios, the Extended Bellman-Ford algorithm converged 2.96% to 15.27% faster than the Darting algorithm in 58.33% of all test cases. In the remaining 41.67% of the cases, the Darting algorithm converged 0.40% to 6.21% faster (Figure 13, Table 45). This trend also increases when the data is restricted to *the Non-Uniform Medium and High Load* cases, with the Extended Bellman-Ford algorithm converging faster then the Darting algorithm in 62.5% of the cases.

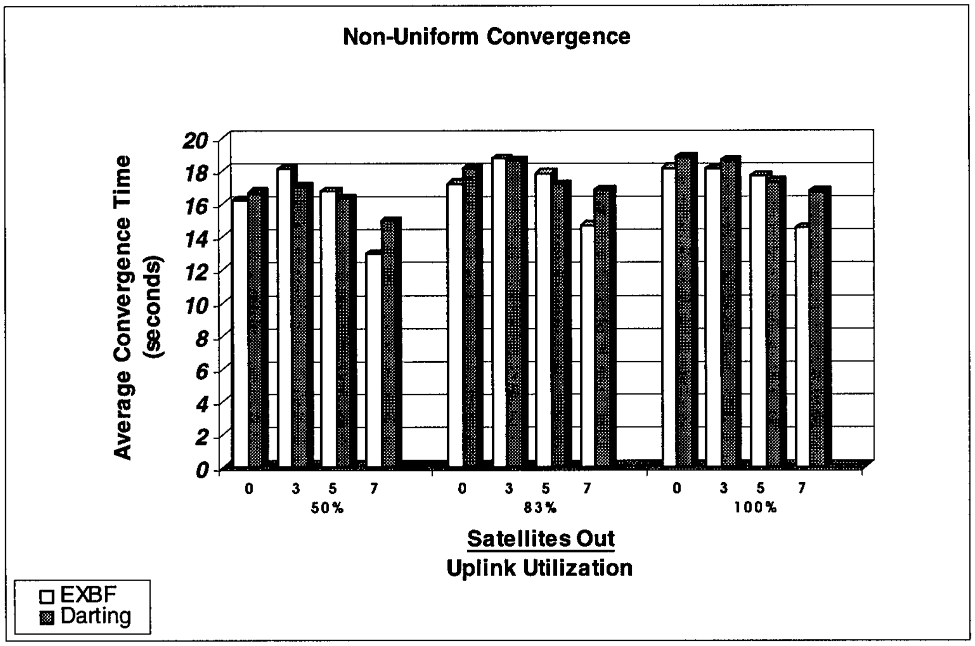

*Figure 13: Non-Uniform Convergence*

The *Non-Uniform* scenario yields an average increase of 1.28 seconds in convergence time when the load increases from 50% to 83%, while the *Uniform* scenario yields a more dramatic average increase of 7.76 seconds. This trend is a result of the traffic distribution used in the test cases. A uniform traffic pattern increases the loading on a greater number of satellites, thereby increasing the delay encountered by the converging update packets.

### 4.6.5 Hop Count

Hop count is the final metric analyzed. Under the *Uniform* test scenarios, packets travelling from Kansas City to Rio had the fewest number of hops, averaging from 3.63 to 5.04 hops each (Figure 14). The lowest hop count for the *Non-Uniform* case occurred between Kansas City and Rio, averaging 4.19 to 5.29 hops (Figure 40, Appendix E).

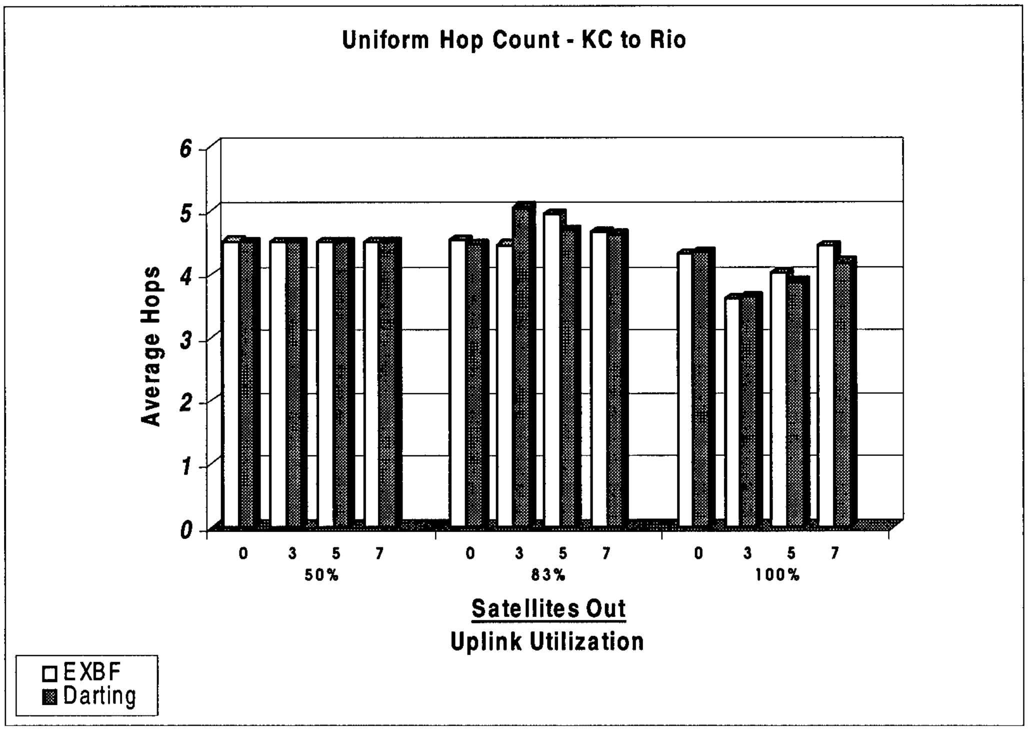

*Figure 14: Uniform Hop Count, KC to Rio*

The greatest average hop count by packets, in the *Uniform* scenarios, occurred along the path from Kansas City to Capetown. These packets averaged counts from 8.13 to 12.0 hops (Figure 15). This was also the case under the *Non-Uniform* scenarios, with the greatest hop count averaging 8.06 to 11.7 hops between Kansas City and Capetown (Figure 36, Appendix E). The relationship that exists between hop count and end-to-end delay is too small to matter because end-to-end delay is mostly impacted by queuing delay.

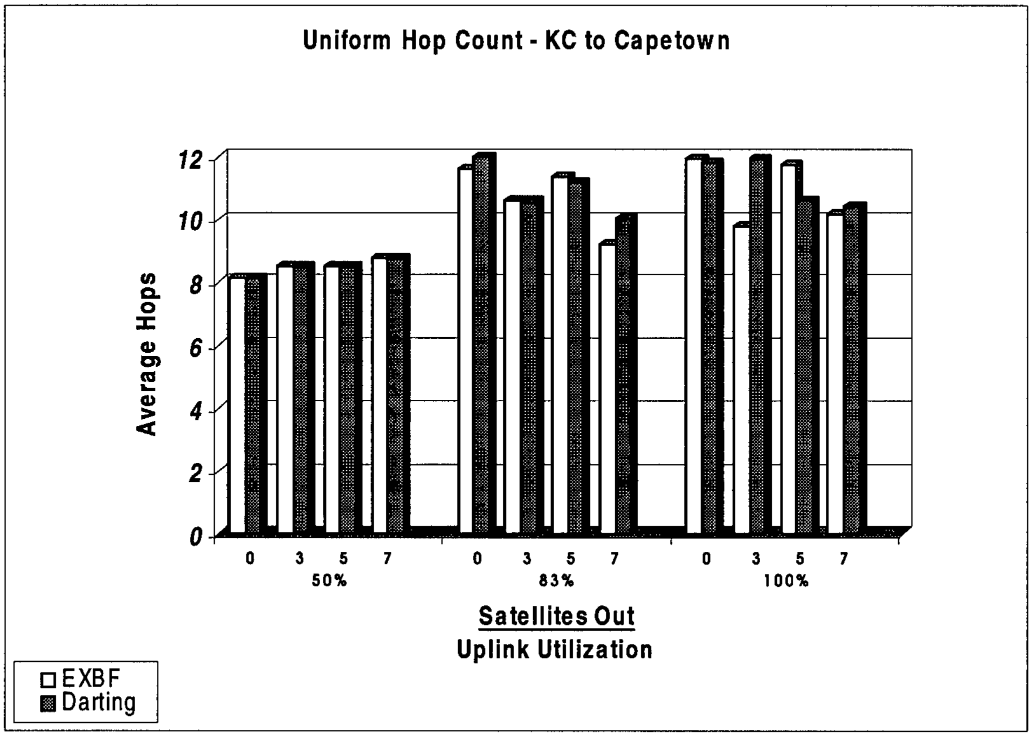

*Figure 15: Uniform Hop Count, KC to Capetown*

## 4.7 Comparative Performance Analysis

As previously mentioned, the satellite queues are modeled to reject packets whose end-to-end delay exceeded 400-ms. As such, the primary performance metric becomes the rejection rates under each algorithm. In the *Uniform* scenarios, rejection rates are 0% for all cases except the *Uniform High Load* case. Under the *Uniform High Load* scenario, rejection rates vary from 0% to 7.18% for the Extended Bellman-Ford and 1.38% to 8.12% for the Darting algorithm. The average difference in rejection rates is

1.30%. The only rejected packets occurring in the *Non-Uniform* scenarios are in the *Medium and High Load* cases with a full constellation. Under these scenarios, Darting provides the lowest rejection rates. Darting rejects 3.14% less packets in *the Non-Uniform Medium Load* case and 1.65% less in the *High Load* case. This trend is more significant since network traffic generally has a non-uniform distribution.

The next metric used in this comparative analysis is the mean end-to-end delay metric discussed in Section 4.6.1.4. Under the *Uniform* scenarios, packets routed using the Darting algorithm reached their destination faster than those using the Extended Bellman-Ford algorithm in 65.33% of the scenarios (Figure 8). This was also the case under the Non-*Uniform* scenarios, with packets routed with the Darting algorithm reaching their destination faster than those routed by the Extended Bellman-Ford algorithm in 77.67% of the scenarios (Figure 9).

Under the third most important metric, overhead traffic, Darting produced 18.55% to 34.06% less overhead packets in the *Uniform* scenario and 28.08% to 33.65% less in the *Non-Uniform* scenario. Clearly, the fact that Darting produces less overhead impacts both the mean delay and the rejection rate metrics. Modeling an increased number of earth stations would increase overhead, which in turn could have a negative impact on both rejection rate and average delay.

Convergence time is the fourth metric. In 75% of the test cases in the *Uniform* test scenario, the Darting algorithm converged 9.47% to 40.39% faster than the Extended Bellman-Ford algorithm. In the remaining 25% of the cases, the Extended Bellman-Ford algorithm converged 10.27% to 149.02% faster (Figure 12, Table 44). Under the *Non-Uniform* scenarios, the Extended Bellman-Ford algorithm converged 2.96% to 15.27%

faster than the Darting algorithm in 58.33% of the test cases. In the remaining 41.67% of the cases, the Darting algorithm converged 0.40% to 6.21% faster (Figure 13, Table 45).

The final metric is hop count. Under the *Uniform* test scenarios, packets travelling from Kansas City to Rio had the fewest number of hops, averaging between 3.63 and 5.04 hops each (Figure 14). The lowest hop count for the *Non-Uniform* case occurred between Kansas City and Rio, averaging between 4.19 and 5.29 hops (Figure 40, Appendix E). The greatest average hop count by packets was in the *Uniform* scenarios and occurred along the path from Kansas City to Capetown. These packets averaged between 8.13 and 12.0 hops (Figure 15). This was also the case under the *Non-Uniform* scenarios, with the greatest hop count averaging 8.06 to 11.7 hops between Kansas City and Capetown (Figure 36, Appendix E). As stated earlier, the relationship that exists between hop count and end-to-end delay is too small to matter because end-to-end delay is mostly impacted by queuing delay.

## 4.8 Summary

The analysis presented in Sections 4.6 and 4.7 demonstrate that both algorithms are suitable for meeting real-time communications constraints with a number of non-operational satellites. Even more important is the identification for the need for a load-balancing mechanism in the network. As was shown in the rejection rate analysis, the removal of high traffic nodes forced the algorithms to establish new shortest paths that utilized a larger subset of the network. This balanced the load on the network and forced the rejection rate to 0%. Therefore, in order to meet the constraints of a real-time communications system and meet a rejection rate of 1% or less, the network must incorporate this load-balancing feature.

When the algorithms are compared, the metrics indicate that the Darting algorithm holds a distinct advantage over the Extended Bellman-Ford algorithm. However, this conclusion is based upon the simulation of a limited number of earth stations utilizing a small subset of the entire Iridium® constellation. Further research is needed to determine the whether utilizing a greater subset of the constellation effects the performance metrics of Darting.

Overall, the analysis indicates that the Iridium® system is capable of meeting real-time communication constraints, utilizing either algorithm, as long as a load balancing mechanism is used to reroute packets around heavily congested satellites.

# 5. CONCLUSIONS AND RECOMMENDATIONS

## 5.1 Restatement of Research Goal

The goal of this research was to perform a comparative analysis on the Extended Bellman-Ford and Darting algorithms, assessing their performance in a LEO environment with high loading levels. A secondary goal was to assess the robustness of the Iridium® system with a degraded constellation.

## 5.2 Research Contributions

This work is the first compare the performance of two "real world" routing algorithms in a low earth environment and under high loading levels. A single simulation testbed that can obtain valuable performance criteria on a variety of routing algorithms, under various loading levels, and in the presence of nodal failures was developed. The simulation corrects the error in ISL connectivity made by Fossa and conducts simulation trials at higher loading levels on several routing algorithms, two of which were previously tested by Janoso at low loading levels[Jan96].

## 5.3 Algorithm Applicability in the LEO Environment

Both algorithms proved suitable for use in a LEO network, provided a load-balancing mechanism is used to distribute the network load. Network load-balancing occurred when heavily loaded satellites were removed from the network. Without a load balancing mechanism, both algorithms failed to meet the rejection rate benchmark with a full constellation.

## 5.4 Algorithm Performance

When the algorithms are compared, performance metrics indicate that the Darting algorithm holds a distinct advantage over the Extended Bellman-Ford algorithm. However, this conclusion is based upon the simulation of a limited number of earth stations. Increasing the number of earth stations modeled will likely have a negative impact on Darting's performance since a greater amount of overhead will result from an increased number of satellite nodes being utilized for routing.

## 5.5 The Iridium® System

Individual algorithm analysis indicates the Iridium® system is capable of meeting real-time communication constraints provided a load balancing mechanism is used to reroute packets around heavily congested satellites. Additionally, the Iridium® system was found to be robust enough to meet real-time communication benchmarks with a degraded constellation. Once again, the need of a load-balancing mechanism is emphasized.

## 5.6 Recommendations for Future Work

There are three areas of research that can be expanded upon. First, the implementation of an optimal load balancing mechanism to route traffic around congested satellites warrants investigation. Although there is no published information about the Iridium® routing algorithm, it is probable that a load balancing mechanism is present. The algorithmic method used to remove the satellites could act as a baseline implementation of this mechanism.

The second area to expand upon is the methodology used to process update packets. As implemented, updates are not given priority. Implementation of a priority queue in the satellites might yield better performance.

The final area requiring further research is the use of a greater number of earth stations to exercise a greater subset of the constellation. The need to analyze the network under greater loads is evident by the low overhead produced by the Darting algorithm and the 0% rejection rate that resulted when satellites were removed. When satellites are removed from the network coverage area is lost which should result in degraded performance.

## APPENDIX A – Average Delay Tabular Data

This Appendix contains the tabulated end-to-end delay data for each of the test scenarios described in Chapter 3. The 95% confidence interval is also provided with the tabulated data in order to present the range that the data fell within.

*Table 18: Results for Uniform Low Load with a Full Constellation*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Bellman Ford | Dhahran | 125.931 | 0.184 | 125.475 | 126.388 | 0.913 |
| | Melbourne | 144.245 | 0.199 | 143.750 | 144.740 | 0.990 |
| | Beijing | 124.712 | 0.342 | 123.863 | 125.561 | 1.698 |
| | Capetown | 141.615 | 1.071 | 138.955 | 144.275 | 5.320 |
| | Rio | 98.444 | 0.249 | 97.826 | 99.062 | 1.236 |
| | Berlin | 125.931 | 0.184 | 125.475 | 126.388 | 0.913 |
| | | | | | | |
| Extended Bellman Ford | Dhahran | 125.941 | 0.187 | 125.477 | 126.406 | 0.930 |
| | Melbourne | 144.237 | 0.199 | 143.743 | 144.731 | 0.987 |
| | Beijing | 124.730 | 0.327 | 123.918 | 125.542 | 1.624 |
| | Capetown | 141.625 | 1.071 | 138.965 | 144.284 | 5.319 |
| | Rio | 98.433 | 0.238 | 97.842 | 99.024 | 1.182 |
| | Berlin | 125.941 | 0.187 | 125.477 | 126.406 | 0.930 |
| | | | | | | |
| Dijkstra | Dhahran | 125.931 | 0.184 | 125.475 | 126.388 | 0.913 |
| | Melbourne | 144.245 | 0.199 | 143.750 | 144.740 | 0.990 |
| | Beijing | 124.712 | 0.342 | 123.863 | 125.561 | 1.698 |
| | Capetown | 141.615 | 1.071 | 138.955 | 144.275 | 5.320 |
| | Rio | 98.444 | 0.249 | 97.826 | 99.062 | 1.236 |
| | Berlin | 125.931 | 0.184 | 125.475 | 126.388 | 0.913 |
| | | | | | | |
| Darting | Dhahran | 125.936 | 0.186 | 125.474 | 126.398 | 0.924 |
| | Melbourne | 144.231 | 0.206 | 143.720 | 144.742 | 1.022 |
| | Beijing | 124.721 | 0.329 | 123.904 | 125.538 | 1.634 |
| | Capetown | 141.619 | 1.070 | 138.961 | 144.276 | 5.315 |
| | Rio | 98.443 | 0.251 | 97.819 | 99.068 | 1.249 |
| | Berlin | 125.936 | 0.186 | 125.474 | 126.398 | 0.924 |

*Table 19: Results for Uniform Medium Load with a Full Constellation*

| **Algorithm** | **From Kansas City to:** | **Average Sample Mean** | **Standard Deviation** | **95% Confidence Interval** | | |
|---|---|---|---|---|---|---|
| | | | | **Minimum** | **Maximum** | **Range** |
| Bellman Ford | Dhahran | 140.615 | 3.853 | 131.042 | 150.188 | 19.146 |
| | Melbourne | 168.406 | 4.752 | 156.601 | 180.212 | 23.611 |
| | Beijing | 147.901 | 0.809 | 145.890 | 149.912 | 4.022 |
| | Capetown | 203.799 | 8.732 | 182.106 | 225.492 | 43.386 |
| | Rio | 109.885 | 3.158 | 102.040 | 117.731 | 15.691 |
| | Berlin | 140.615 | 3.853 | 131.042 | 150.188 | 19.146 |
| | | | | | | |
| Extended Bellman Ford | Dhahran | 148.047 | 6.140 | 132.794 | 163.300 | 30.505 |
| | Melbourne | 178.581 | 3.292 | 170.403 | 186.759 | 16.357 |
| | Beijing | 153.509 | 3.337 | 145.218 | 161.800 | 16.582 |
| | Capetown | 230.084 | 12.927 | 197.969 | 262.199 | 64.230 |
| | Rio | 113.345 | 4.213 | 102.878 | 123.812 | 20.934 |
| | Berlin | 148.047 | 6.140 | 132.794 | 163.300 | 30.505 |
| | | | | | | |
| Dijkstra | Dhahran | 140.615 | 3.853 | 131.042 | 150.188 | 19.146 |
| | Melbourne | 168.406 | 4.752 | 156.601 | 180.212 | 23.611 |
| | Beijing | 147.901 | 0.809 | 145.890 | 149.912 | 4.022 |
| | Capetown | 203.799 | 8.732 | 182.106 | 225.492 | 43.386 |
| | Rio | 109.885 | 3.158 | 102.040 | 117.731 | 15.691 |
| | Berlin | 140.615 | 3.853 | 131.042 | 150.188 | 19.146 |
| | | | | | | |
| Darting | Dhahran | 141.150 | 3.552 | 132.327 | 149.974 | 17.647 |
| | Melbourne | 169.905 | 4.458 | 158.829 | 180.981 | 22.152 |
| | Beijing | 147.258 | 3.393 | 138.828 | 155.688 | 16.860 |
| | Capetown | 214.201 | 4.004 | 204.253 | 224.148 | 19.895 |
| | Rio | 110.215 | 2.871 | 103.083 | 117.348 | 14.265 |
| | Berlin | 141.150 | 3.552 | 132.327 | 149.974 | 17.647 |

*Table 20: Results for Uniform High Load with a Full Constellation*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | **Minimum** | **Maximum** | **Range** |
| Bellman Ford | Dhahran | 210.237 | 13.252 | 177.315 | 243.159 | 65.844 |
| | Melbourne | 175.021 | 0.703 | 173.274 | 176.768 | 3.494 |
| | Beijing | 219.038 | 8.638 | 197.577 | 240.498 | 42.921 |
| | Capetown | 244.663 | 13.143 | 212.011 | 277.315 | 65.305 |
| | Rio | 130.332 | 1.797 | 125.867 | 134.797 | 8.930 |
| | Berlin | 210.237 | 13.252 | 177.315 | 243.159 | 65.844 |
| | | | | | | |
| Extended Bellman Ford | Dhahran | 207.790 | 7.761 | 188.509 | 227.070 | 38.560 |
| | Melbourne | 175.125 | 5.679 | 161.018 | 189.233 | 28.215 |
| | Beijing | 218.591 | 4.177 | 208.214 | 228.968 | 20.754 |
| | Capetown | 277.240 | 20.934 | 225.233 | 329.247 | 104.015 |
| | Rio | 128.673 | 1.700 | 124.450 | 132.896 | 8.446 |
| | Berlin | 207.790 | 7.761 | 188.509 | 227.070 | 38.560 |
| | | | | | | |
| Dijkstra | Dhahran | 215.558 | 7.776 | 196.241 | 234.876 | 38.635 |
| | Melbourne | 174.883 | 0.584 | 173.431 | 176.334 | 2.903 |
| | Beijing | 225.430 | 8.170 | 205.133 | 245.727 | 40.594 |
| | Capetown | 254.375 | 1.276 | 251.205 | 257.545 | 6.341 |
| | Rio | 131.625 | 1.763 | 127.245 | 136.006 | 8.761 |
| | Berlin | 215.558 | 7.776 | 196.241 | 234.876 | 38.635 |
| | | | | | | |
| Darting | Dhahran | 211.974 | 9.393 | 188.637 | 235.310 | 46.673 |
| | Melbourne | 173.237 | 5.968 | 158.412 | 188.063 | 29.651 |
| | Beijing | 214.731 | 3.080 | 207.078 | 222.384 | 15.306 |
| | Capetown | 271.880 | 10.579 | 245.597 | 298.163 | 52.566 |
| | Rio | 127.533 | 0.096 | 127.294 | 127.773 | 0.479 |
| | Berlin | 211.974 | 9.393 | 188.637 | 235.310 | 46.673 |

*Table 21: Results for Uniform Low Load with 3 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 129.580 | 0.305 | 128.823 | 130.337 | 1.514 |
| | Melbourne | 144.696 | 0.430 | 143.627 | 145.764 | 2.136 |
| | Beijing | 147.419 | 0.319 | 146.625 | 148.212 | 1.586 |
| | Capetown | 145.678 | 1.307 | 142.430 | 148.925 | 6.495 |
| | Rio | 99.613 | 0.246 | 99.002 | 100.224 | 1.221 |
| | Berlin | 129.580 | 0.305 | 128.823 | 130.337 | 1.514 |
| | | | | | | |
| Darting | Dhahran | 129.576 | 0.306 | 128.816 | 130.335 | 1.518 |
| | Melbourne | 144.553 | 0.244 | 143.947 | 145.160 | 1.214 |
| | Beijing | 131.400 | 22.358 | 75.855 | 186.944 | 111.090 |
| | Capetown | 145.667 | 1.303 | 142.429 | 148.904 | 6.475 |
| | Rio | 115.617 | 22.765 | 59.061 | 172.173 | 113.112 |
| | Berlin | 129.576 | 0.306 | 128.816 | 130.335 | 1.518 |

*Table 22: Results for Uniform Medium Load with 3 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 156.064 | 5.937 | 141.315 | 170.814 | 29.499 |
| | Melbourne | 189.374 | 3.680 | 180.231 | 198.518 | 18.287 |
| | Beijing | 174.064 | 1.463 | 170.430 | 177.698 | 7.269 |
| | Capetown | 204.762 | 5.426 | 191.281 | 218.242 | 26.961 |
| | Rio | 122.398 | 11.293 | 94.344 | 150.453 | 56.110 |
| | Berlin | 156.064 | 5.937 | 141.315 | 170.814 | 29.499 |
| | | | | | | |
| Darting | Dhahran | 156.369 | 5.784 | 142.000 | 170.738 | 28.738 |
| | Melbourne | 176.880 | 4.448 | 165.830 | 187.929 | 22.099 |
| | Beijing | 173.911 | 5.466 | 160.331 | 187.492 | 27.160 |
| | Capetown | 203.681 | 3.304 | 195.474 | 211.888 | 16.414 |
| | Rio | 112.158 | 2.731 | 105.372 | 118.944 | 13.572 |
| | Berlin | 156.369 | 5.784 | 142.000 | 170.738 | 28.738 |

*Table 23: Results for Uniform High Load with 3 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 170.598 | 6.874 | 153.520 | 187.677 | 34.157 |
| | Melbourne | 198.561 | 1.794 | 194.103 | 203.019 | 8.916 |
| | Beijing | 209.544 | 2.602 | 203.079 | 216.009 | 12.929 |
| | Capetown | 262.947 | 17.384 | 219.759 | 306.135 | 86.376 |
| | Rio | 140.800 | 1.024 | 138.255 | 143.345 | 5.090 |
| | Berlin | 170.598 | 6.874 | 153.520 | 187.677 | 34.157 |
| | | | | | | |
| Darting | Dhahran | 167.791 | 4.280 | 157.157 | 178.424 | 21.267 |
| | Melbourne | 209.370 | 5.709 | 195.187 | 223.552 | 28.365 |
| | Beijing | 206.690 | 0.305 | 205.933 | 207.447 | 1.515 |
| | Capetown | 220.966 | 3.720 | 211.723 | 230.209 | 18.486 |
| | Rio | 137.818 | 2.269 | 132.181 | 143.454 | 11.273 |
| | Berlin | 167.791 | 4.280 | 157.157 | 178.424 | 21.267 |

*Table 24: Results for Uniform Low Load with 5 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 154.704 | 0.556 | 153.323 | 156.085 | 2.762 |
| | Melbourne | 144.731 | 0.402 | 143.732 | 145.731 | 1.999 |
| | Beijing | 167.773 | 0.139 | 167.428 | 168.119 | 0.691 |
| | Capetown | 146.009 | 1.326 | 142.715 | 149.303 | 6.589 |
| | Rio | 99.610 | 0.240 | 99.013 | 100.207 | 1.194 |
| | Berlin | 154.704 | 0.556 | 153.323 | 156.085 | 2.762 |
| | | | | | | |
| Darting | Dhahran | 154.696 | 0.557 | 153.311 | 156.081 | 2.770 |
| | Melbourne | 144.743 | 0.408 | 143.729 | 145.757 | 2.028 |
| | Beijing | 144.997 | 31.973 | 65.565 | 224.429 | 158.864 |
| | Capetown | 145.995 | 1.323 | 142.709 | 149.281 | 6.572 |
| | Rio | 122.326 | 32.257 | 42.189 | 202.463 | 160.274 |
| | Berlin | 154.696 | 0.557 | 153.311 | 156.081 | 2.770 |

*Table 25: Results for Uniform Medium Load with 5 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 199.596 | 4.242 | 189.058 | 210.135 | 21.077 |
| | Melbourne | 166.681 | 5.734 | 152.435 | 180.927 | 28.492 |
| | Beijing | 191.293 | 2.446 | 185.217 | 197.369 | 12.152 |
| | Capetown | 221.701 | 6.811 | 204.780 | 238.621 | 33.840 |
| | Rio | 120.880 | 4.289 | 110.224 | 131.536 | 21.313 |
| | Berlin | 199.596 | 4.242 | 189.058 | 210.135 | 21.077 |
| | | | | | | |
| Darting | Dhahran | 194.163 | 4.232 | 183.649 | 204.676 | 21.027 |
| | Melbourne | 169.092 | 5.256 | 156.033 | 182.151 | 26.118 |
| | Beijing | 191.783 | 2.613 | 185.292 | 198.273 | 12.981 |
| | Capetown | 227.411 | 6.792 | 210.537 | 244.284 | 33.747 |
| | Rio | 123.173 | 2.094 | 117.971 | 128.376 | 10.405 |
| | Berlin | 194.163 | 4.232 | 183.649 | 204.676 | 21.027 |

*Table 26: Results for Uniform High Load with 5 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 195.086 | 9.115 | 172.442 | 217.729 | 45.287 |
| | Melbourne | 198.213 | 7.973 | 178.407 | 218.020 | 39.613 |
| | Beijing | 208.003 | 49.412 | 85.246 | 330.760 | 245.514 |
| | Capetown | 248.175 | 13.270 | 215.207 | 281.143 | 65.936 |
| | Rio | 171.371 | 50.601 | 45.662 | 297.080 | 251.418 |
| | Berlin | 195.086 | 9.115 | 172.442 | 217.729 | 45.287 |
| | | | | | | |
| Darting | Dhahran | 210.062 | 5.716 | 195.862 | 224.262 | 28.400 |
| | Melbourne | 207.341 | 4.473 | 196.229 | 218.452 | 22.223 |
| | Beijing | 236.172 | 0.566 | 234.765 | 237.580 | 2.815 |
| | Capetown | 259.576 | 6.354 | 243.790 | 275.362 | 31.573 |
| | Rio | 138.501 | 1.761 | 134.127 | 142.874 | 8.748 |
| | Berlin | 210.062 | 5.716 | 195.862 | 224.262 | 28.400 |

*Table 27: Results for Uniform Low Load with 7 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 168.508 | 0.384 | 167.553 | 169.463 | 1.909 |
| | Melbourne | 144.914 | 0.467 | 143.755 | 146.073 | 2.318 |
| | Beijing | 169.217 | 0.076 | 169.027 | 169.407 | 0.380 |
| | Capetown | 149.397 | 1.571 | 145.494 | 153.299 | 7.804 |
| | Rio | 99.619 | 0.243 | 99.016 | 100.222 | 1.206 |
| | Berlin | 168.508 | 0.384 | 167.553 | 169.463 | 1.909 |
| | | | | | | |
| Darting | Dhahran | 168.345 | 0.259 | 167.700 | 168.989 | 1.289 |
| | Melbourne | 144.896 | 0.474 | 143.719 | 146.073 | 2.354 |
| | Beijing | 146.052 | 32.715 | 64.777 | 227.327 | 162.550 |
| | Capetown | 149.346 | 1.559 | 145.473 | 153.219 | 7.746 |
| | Rio | 122.781 | 32.892 | 41.067 | 204.495 | 163.428 |
| | Berlin | 168.345 | 0.259 | 167.700 | 168.989 | 1.289 |

*Table 28: Results for Uniform Medium Load with 7 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 176.788 | 0.859 | 174.655 | 178.921 | 4.266 |
| | Melbourne | 160.099 | 0.333 | 159.273 | 160.925 | 1.652 |
| | Beijing | 180.765 | 0.496 | 179.534 | 181.997 | 2.462 |
| | Capetown | 181.581 | 8.583 | 160.258 | 202.904 | 42.646 |
| | Rio | 110.907 | 0.485 | 109.702 | 112.112 | 2.410 |
| | Berlin | 176.788 | 0.859 | 174.655 | 178.921 | 4.266 |
| | | | | | | |
| Darting | Dhahran | 176.272 | 0.820 | 174.234 | 178.310 | 4.076 |
| | Melbourne | 158.237 | 0.918 | 155.957 | 160.517 | 4.560 |
| | Beijing | 180.895 | 0.193 | 180.416 | 181.374 | 0.959 |
| | Capetown | 167.365 | 3.219 | 159.367 | 175.362 | 15.995 |
| | Rio | 111.132 | 0.659 | 109.495 | 112.769 | 3.273 |
| | Berlin | 176.272 | 0.820 | 174.234 | 178.310 | 4.076 |

*Table 29: Results for Uniform High Load with 7 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 197.761 | 0.696 | 196.032 | 199.490 | 3.458 |
| | Melbourne | 206.873 | 5.087 | 194.234 | 219.512 | 25.278 |
| | Beijing | 208.261 | 6.552 | 191.984 | 224.538 | 32.554 |
| | Capetown | 223.237 | 2.583 | 216.819 | 229.655 | 12.836 |
| | Rio | 114.715 | 1.749 | 110.370 | 119.061 | 8.690 |
| | Berlin | 197.761 | 0.696 | 196.032 | 199.490 | 3.458 |
| | | | | | | |
| Darting | Dhahran | 200.959 | 0.616 | 199.429 | 202.489 | 3.060 |
| | Melbourne | 202.522 | 8.034 | 182.564 | 222.481 | 39.917 |
| | Beijing | 199.879 | 1.728 | 195.586 | 204.172 | 8.586 |
| | Capetown | 214.863 | 7.371 | 196.550 | 233.176 | 36.626 |
| | Rio | 119.700 | 2.101 | 114.480 | 124.919 | 10.439 |
| | Berlin | 200.959 | 0.616 | 199.429 | 202.489 | 3.060 |

*Table 30: Results for Non-Uniform Low Load with a Full Constellation*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 131.996 | 0.553 | 130.621 | 133.371 | 2.750 |
| | Melbourne | 149.372 | 1.891 | 144.674 | 154.071 | 9.397 |
| | Beijing | 129.332 | 0.973 | 126.915 | 131.748 | 4.833 |
| | Capetown | 160.240 | 3.826 | 150.735 | 169.746 | 19.011 |
| | Rio | 101.338 | 0.881 | 99.150 | 103.526 | 4.377 |
| | Berlin | 131.996 | 0.553 | 130.621 | 133.371 | 2.750 |
| | | | | | | |
| Darting | Dhahran | 130.211 | 0.830 | 128.148 | 132.274 | 4.126 |
| | Melbourne | 147.274 | 1.425 | 143.735 | 150.813 | 7.078 |
| | Beijing | 127.498 | 0.759 | 125.612 | 129.383 | 3.770 |
| | Capetown | 156.507 | 5.012 | 144.056 | 168.959 | 24.904 |
| | Rio | 99.717 | 0.556 | 98.335 | 101.098 | 2.764 |
| | Berlin | 130.211 | 0.830 | 128.148 | 132.274 | 4.126 |

*Table 31: Results for Non-Uniform Medium Load with a Full Constellation*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 196.456 | 3.506 | 187.747 | 205.165 | 17.418 |
| | Melbourne | 180.596 | 0.836 | 178.519 | 182.673 | 4.153 |
| | Beijing | 192.474 | 3.169 | 184.601 | 200.347 | 15.746 |
| | Capetown | 176.143 | 3.244 | 168.083 | 184.203 | 16.120 |
| | Rio | 131.645 | 0.303 | 130.892 | 132.397 | 1.504 |
| | Berlin | 196.456 | 3.506 | 187.747 | 205.165 | 17.418 |
| | | | | | | |
| Darting | Dhahran | 183.457 | 4.185 | 173.061 | 193.854 | 20.792 |
| | Melbourne | 175.596 | 2.486 | 169.421 | 181.772 | 12.351 |
| | Beijing | 180.092 | 3.477 | 171.454 | 188.729 | 17.275 |
| | Capetown | 183.203 | 2.221 | 177.685 | 188.721 | 11.036 |
| | Rio | 127.891 | 1.373 | 124.481 | 131.301 | 6.820 |
| | Berlin | 183.457 | 4.185 | 173.061 | 193.854 | 20.792 |

*Table 32: Results for Non-Uniform High Load with a Full Constellation*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 198.600 | 2.633 | 192.060 | 205.140 | 13.081 |
| | Melbourne | 183.397 | 0.832 | 181.329 | 185.464 | 4.135 |
| | Beijing | 202.558 | 6.334 | 186.822 | 218.295 | 31.473 |
| | Capetown | 177.533 | 3.335 | 169.248 | 185.817 | 16.569 |
| | Rio | 135.003 | 0.284 | 134.299 | 135.708 | 1.409 |
| | Berlin | 198.600 | 2.633 | 192.060 | 205.140 | 13.081 |
| | | | | | | |
| Darting | Dhahran | 201.984 | 2.322 | 196.216 | 207.753 | 11.537 |
| | Melbourne | 183.514 | 1.060 | 180.880 | 186.147 | 5.267 |
| | Beijing | 202.969 | 8.951 | 180.732 | 225.206 | 44.474 |
| | Capetown | 179.741 | 3.204 | 171.780 | 187.702 | 15.922 |
| | Rio | 134.767 | 0.670 | 133.102 | 136.431 | 3.329 |
| | Berlin | 201.984 | 2.322 | 196.216 | 207.753 | 11.537 |

*Table 33: Results for Non-Uniform Low Load with 3 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 163.773 | 0.556 | 162.392 | 165.154 | 2.762 |
| | Melbourne | 147.390 | 1.833 | 142.835 | 151.944 | 9.109 |
| | Beijing | 172.565 | 2.571 | 166.177 | 178.954 | 12.777 |
| | Capetown | 152.922 | 2.379 | 147.011 | 158.834 | 11.823 |
| | Rio | 101.837 | 1.227 | 98.788 | 104.886 | 6.098 |
| | Berlin | 163.773 | 0.556 | 162.392 | 165.154 | 2.762 |
| | | | | | | |
| Darting | Dhahran | 162.512 | 0.882 | 160.320 | 164.704 | 4.385 |
| | Melbourne | 146.863 | 1.850 | 142.268 | 151.458 | 9.190 |
| | Beijing | 171.111 | 1.511 | 167.357 | 174.866 | 7.509 |
| | Capetown | 151.334 | 1.305 | 148.093 | 154.575 | 6.482 |
| | Rio | 100.942 | 0.583 | 99.492 | 102.391 | 2.898 |
| | Berlin | 162.512 | 0.882 | 160.320 | 164.704 | 4.385 |

*Table 34: Results for Non-Uniform Medium Load with 3 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 226.708 | 4.196 | 216.284 | 237.131 | 20.847 |
| | Melbourne | 176.375 | 0.699 | 174.639 | 178.111 | 3.472 |
| | Beijing | 231.496 | 3.731 | 222.225 | 240.766 | 18.540 |
| | Capetown | 190.840 | 2.168 | 185.453 | 196.227 | 10.774 |
| | Rio | 129.458 | 0.678 | 127.773 | 131.143 | 3.370 |
| | Berlin | 226.708 | 4.196 | 216.284 | 237.131 | 20.847 |
| | | | | | | |
| Darting | Dhahran | 224.337 | 5.884 | 209.719 | 238.956 | 29.238 |
| | Melbourne | 174.981 | 0.540 | 173.639 | 176.322 | 2.684 |
| | Beijing | 227.488 | 6.964 | 210.187 | 244.789 | 34.602 |
| | Capetown | 192.022 | 5.471 | 178.431 | 205.613 | 27.182 |
| | Rio | 128.934 | 1.722 | 124.656 | 133.213 | 8.557 |
| | Berlin | 224.337 | 5.884 | 209.719 | 238.956 | 29.238 |

*Table 35: Results for Non-Uniform High Load with 3 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 186.008 | 4.689 | 174.360 | 197.656 | 23.296 |
| | Melbourne | 169.829 | 1.471 | 166.175 | 173.483 | 7.307 |
| | Beijing | 235.383 | 5.523 | 221.662 | 249.103 | 27.441 |
| | Capetown | 179.881 | 9.760 | 155.633 | 204.128 | 48.494 |
| | Rio | 130.213 | 0.453 | 129.086 | 131.340 | 2.253 |
| | Berlin | 186.008 | 4.689 | 174.360 | 197.656 | 23.296 |
| | | | | | | |
| Darting | Dhahran | 183.328 | 4.144 | 173.032 | 193.624 | 20.591 |
| | Melbourne | 171.663 | 0.840 | 169.575 | 173.751 | 4.176 |
| | Beijing | 237.186 | 0.884 | 234.991 | 239.382 | 4.391 |
| | Capetown | 176.997 | 4.806 | 165.058 | 188.936 | 23.878 |
| | Rio | 131.084 | 0.991 | 128.623 | 133.546 | 4.922 |
| | Berlin | 183.328 | 4.144 | 173.032 | 193.624 | 20.591 |

*Table 36: Results for Non-Uniform Low Load with 5 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 188.357 | 2.581 | 181.944 | 194.769 | 12.825 |
| | Melbourne | 145.744 | 1.215 | 142.724 | 148.763 | 6.038 |
| | Beijing | 171.245 | 0.349 | 170.378 | 172.112 | 1.734 |
| | Capetown | 159.893 | 4.894 | 147.734 | 172.052 | 24.318 |
| | Rio | 100.125 | 1.079 | 97.444 | 102.806 | 5.363 |
| | Berlin | 188.357 | 2.581 | 181.944 | 194.769 | 12.825 |
| | | | | | | |
| Darting | Dhahran | 186.086 | 2.811 | 179.103 | 193.070 | 13.967 |
| | Melbourne | 145.703 | 1.481 | 142.024 | 149.382 | 7.358 |
| | Beijing | 171.084 | 0.271 | 170.411 | 171.757 | 1.345 |
| | Capetown | 156.948 | 2.186 | 151.517 | 162.378 | 10.861 |
| | Rio | 100.084 | 1.034 | 97.515 | 102.652 | 5.137 |
| | Berlin | 186.086 | 2.811 | 179.103 | 193.070 | 13.967 |

*Table 37: Results for Non-Uniform Medium Load with 5 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 197.471 | 5.750 | 183.186 | 211.756 | 28.570 |
| | Melbourne | 156.323 | 2.205 | 150.846 | 161.801 | 10.955 |
| | Beijing | 186.748 | 3.993 | 176.828 | 196.669 | 19.840 |
| | Capetown | 173.350 | 6.507 | 157.186 | 189.515 | 32.329 |
| | Rio | 114.836 | 3.661 | 105.741 | 123.931 | 18.190 |
| | Berlin | 197.471 | 5.750 | 183.186 | 211.756 | 28.570 |
| | | | | | | |
| Darting | Dhahran | 191.681 | 1.709 | 187.435 | 195.927 | 8.492 |
| | Melbourne | 153.511 | 0.967 | 151.108 | 155.914 | 4.805 |
| | Beijing | 182.756 | 3.061 | 175.151 | 190.362 | 15.210 |
| | Capetown | 175.726 | 5.970 | 160.893 | 190.558 | 29.665 |
| | Rio | 110.616 | 1.377 | 107.195 | 114.037 | 6.843 |
| | Berlin | 191.681 | 1.709 | 187.435 | 195.927 | 8.492 |

*Table 38: Results for Non-Uniform High Load with 5 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 204.734 | 14.530 | 168.637 | 240.832 | 72.195 |
| | Melbourne | 162.387 | 5.458 | 148.827 | 175.947 | 27.121 |
| | Beijing | 190.675 | 6.641 | 174.176 | 207.173 | 32.997 |
| | Capetown | 168.874 | 0.740 | 167.035 | 170.713 | 3.677 |
| | Rio | 126.040 | 10.777 | 99.265 | 152.815 | 53.550 |
| | Berlin | 204.734 | 14.530 | 168.637 | 240.832 | 72.195 |
| | | | | | | |
| Darting | Dhahran | 201.929 | 13.923 | 167.339 | 236.519 | 69.179 |
| | Melbourne | 160.909 | 5.858 | 146.356 | 175.463 | 29.107 |
| | Beijing | 189.970 | 9.077 | 167.420 | 212.521 | 45.101 |
| | Capetown | 170.779 | 2.497 | 164.576 | 176.982 | 12.407 |
| | Rio | 122.936 | 11.277 | 94.921 | 150.951 | 56.030 |
| | Berlin | 201.929 | 13.923 | 167.339 | 236.519 | 69.179 |

*Table 39: Results for Non-Uniform Low Load with 7 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 194.095 | 3.194 | 186.159 | 202.031 | 15.872 |
| | Melbourne | 148.415 | 1.086 | 145.718 | 151.112 | 5.394 |
| | Beijing | 175.585 | 0.672 | 173.915 | 177.254 | 3.338 |
| | Capetown | 171.751 | 3.674 | 162.625 | 180.877 | 18.252 |
| | Rio | 100.310 | 0.777 | 98.380 | 102.239 | 3.859 |
| | Berlin | 194.095 | 3.194 | 186.159 | 202.031 | 15.872 |
| | | | | | | |
| Darting | Dhahran | 192.706 | 2.240 | 187.141 | 198.271 | 11.130 |
| | Melbourne | 147.729 | 1.075 | 145.058 | 150.400 | 5.343 |
| | Beijing | 175.359 | 0.302 | 174.608 | 176.110 | 1.503 |
| | Capetown | 165.425 | 3.804 | 155.975 | 174.875 | 18.900 |
| | Rio | 100.042 | 0.733 | 98.221 | 101.862 | 3.641 |
| | Berlin | 192.706 | 2.240 | 187.141 | 198.271 | 11.130 |

*Table 40: Results for Non-Uniform Medium Load with 7 Satellites Removed*

| Algorithm | From Kansas City to: | Average Sample Mean | Standard Deviation | 95% Confidence Interval | | |
|---|---|---|---|---|---|---|
| | | | | Minimum | Maximum | Range |
| Extended Bellman Ford | Dhahran | 202.510 | 1.082 | 199.821 | 205.199 | 5.378 |
| | Melbourne | 152.644 | 0.305 | 151.887 | 153.401 | 1.514 |
| | Beijing | 179.038 | 0.652 | 177.419 | 180.657 | 3.238 |
| | Capetown | 191.549 | 4.679 | 179.926 | 203.172 | 23.246 |
| | Rio | 103.484 | 0.412 | 102.461 | 104.508 | 2.048 |
| | Berlin | 202.510 | 1.082 | 199.821 | 205.199 | 5.378 |
| | | | | | | |
| Darting | Dhahran | 201.620 | 1.710 | 197.372 | 205.867 | 8.495 |
| | Melbourne | 151.696 | 0.339 | 150.852 | 152.539 | 1.686 |
| | Beijing | 178.475 | 0.482 | 177.279 | 179.672 | 2.394 |
| | Capetown | 191.330 | 6.849 | 174.315 | 208.345 | 34.030 |
| | Rio | 102.983 | 0.392 | 102.008 | 103.958 | 1.950 |
| | Berlin | 201.620 | 1.710 | 197.372 | 205.867 | 8.495 |

*Table 41: Results for Non-Uniform High Load with 7 Satellites Removed*

| **Algorithm** | **From Kansas City to:** | **Average Sample Mean** | **Standard Deviation** | **95% Confidence Interval** | | |
|---|---|---|---|---|---|---|
| | | | | **Minimum** | **Maximum** | **Range** |
| Extended Bellman Ford | Dhahran | 199.431 | 3.317 | 191.191 | 207.672 | 16.481 |
| | Melbourne | 157.833 | 2.340 | 152.018 | 163.647 | 11.628 |
| | Beijing | 185.669 | 1.590 | 181.719 | 189.619 | 7.900 |
| | Capetown | 207.038 | 3.020 | 199.536 | 214.541 | 15.005 |
| | Rio | 107.910 | 1.543 | 104.077 | 111.742 | 7.665 |
| | Berlin | 199.431 | 3.317 | 191.191 | 207.672 | 16.481 |
| | | | | | | |
| Darting | Dhahran | 197.473 | 3.578 | 188.584 | 206.362 | 17.778 |
| | Melbourne | 155.058 | 0.973 | 152.642 | 157.474 | 4.833 |
| | Beijing | 182.578 | 1.712 | 178.325 | 186.832 | 8.507 |
| | Capetown | 200.595 | 3.102 | 192.889 | 208.301 | 15.412 |
| | Rio | 105.756 | 1.163 | 102.867 | 108.644 | 5.777 |
| | Berlin | 197.473 | 3.578 | 188.584 | 206.362 | 17.778 |

## APPENDIX B – Average Delay Figures

This Appendix contains figures for the end-to-end delay data of each test scenarios described in Chapter 3. Tables are presented for both to show both trends for increased uplink utilization and satellite removal.

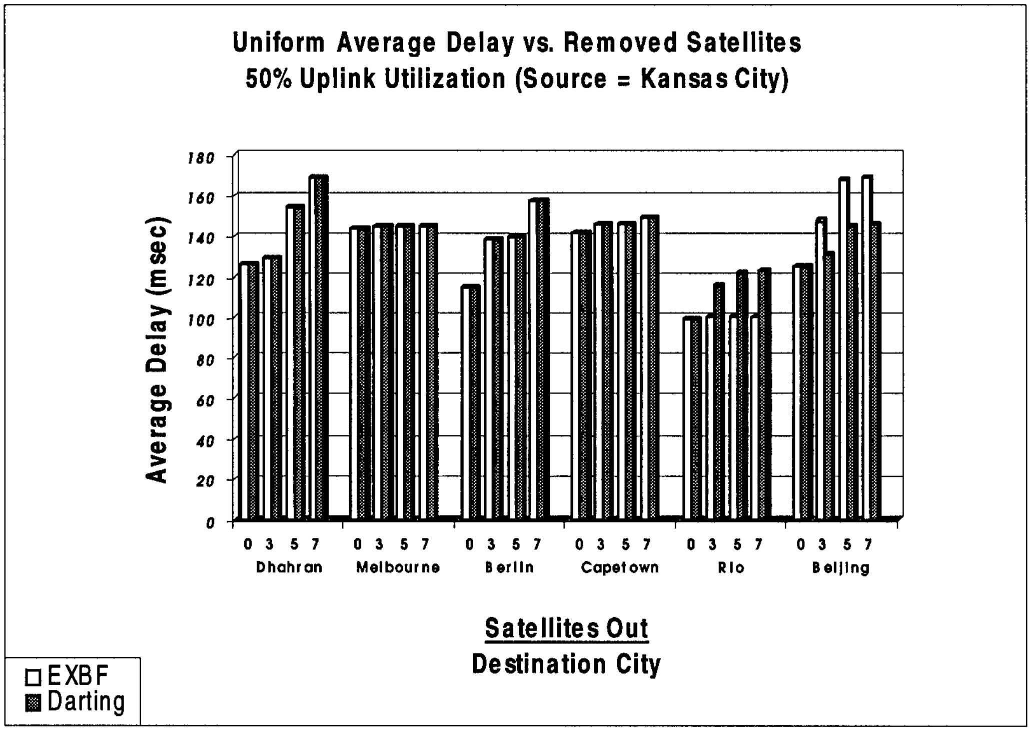

*Figure 16: Uniform Average Delay vs. Removed Satellites, 50% Uplink Utilization*

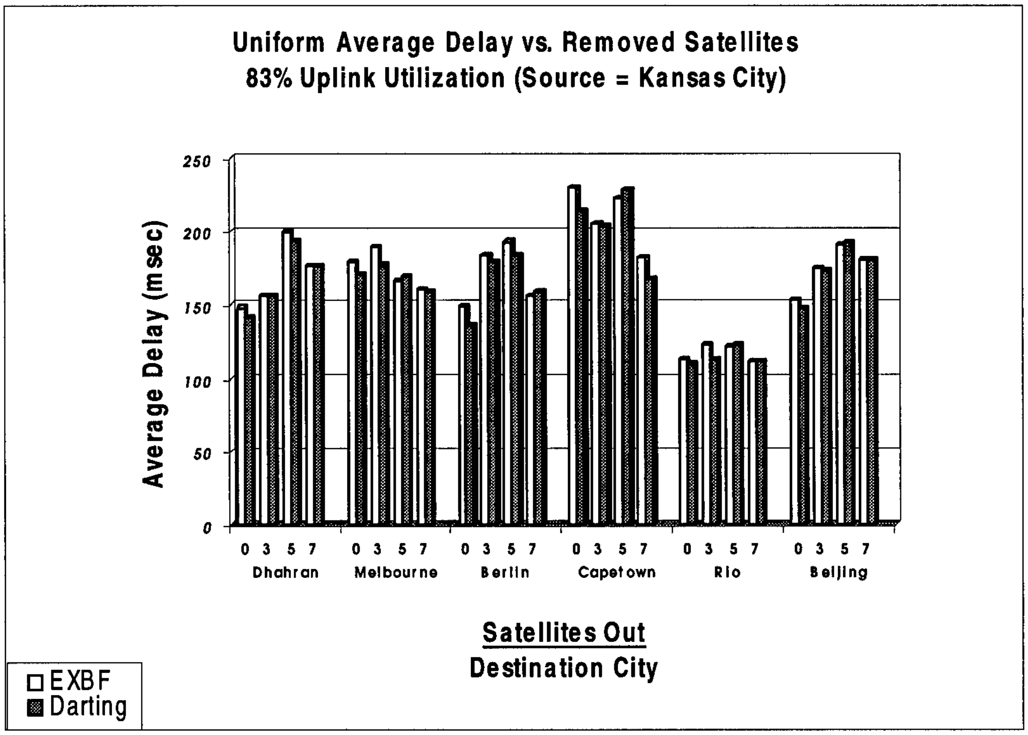

*Figure 17: Uniform Average Delay vs. Removed Satellites, 83% Uplink Utilization*

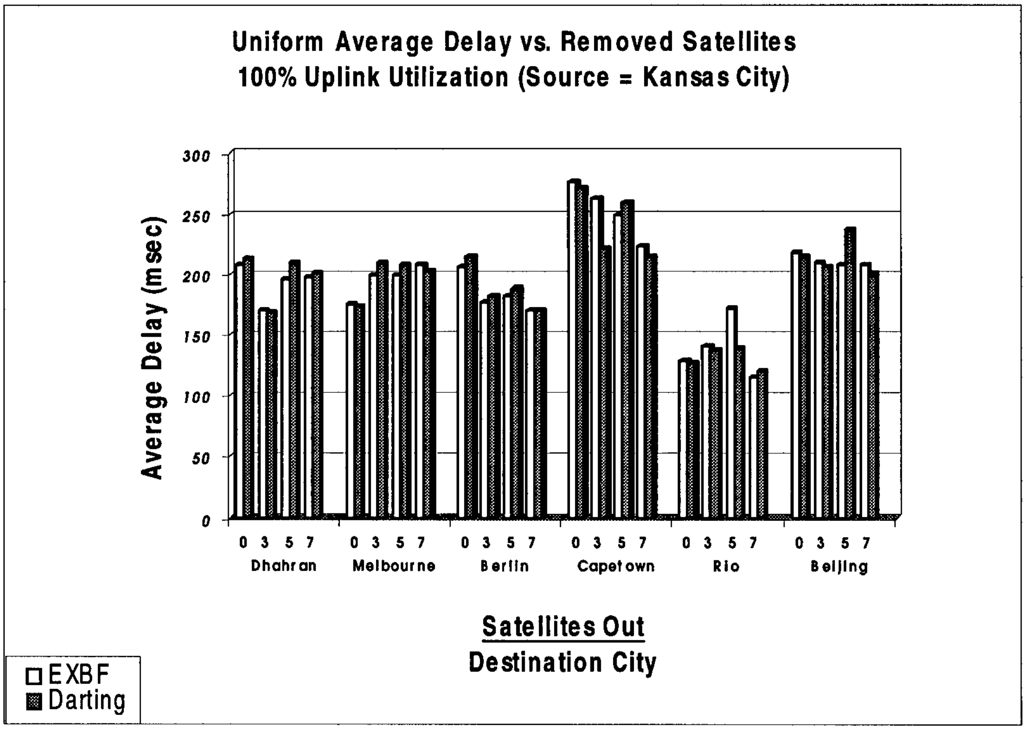

*Figure 18: Uniform Average Delay vs. Removed Satellites, 100% Uplink Utilization*

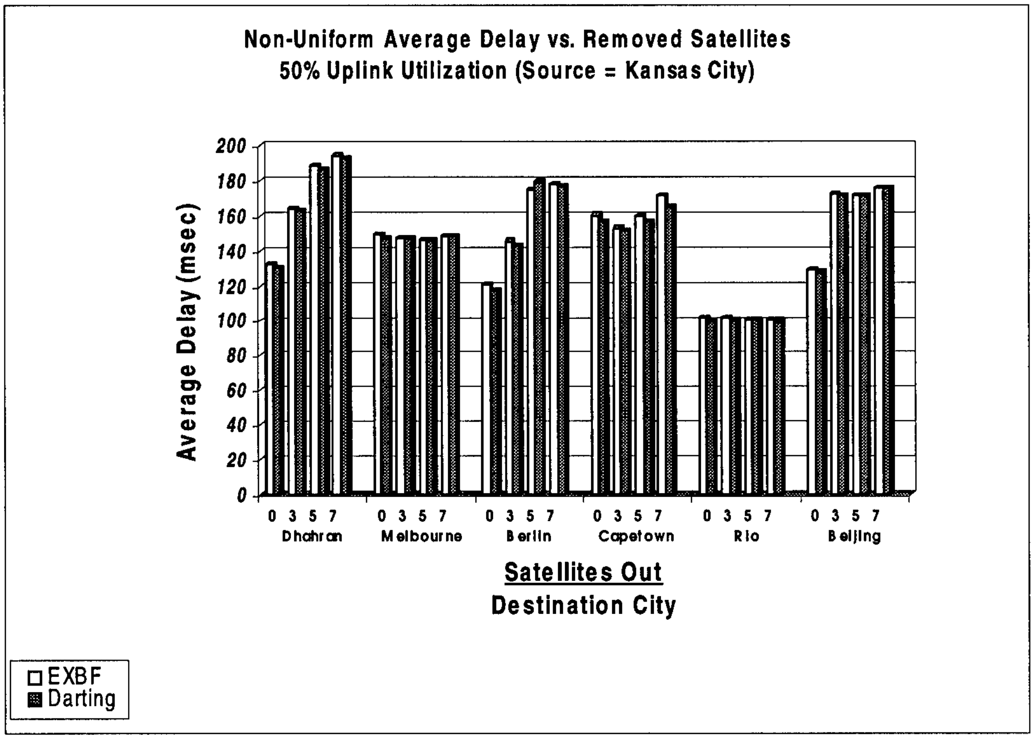

*Figure 19: Non-Uniform Average Delay vs. Removed Satellites, 50% Uplink Utilization*

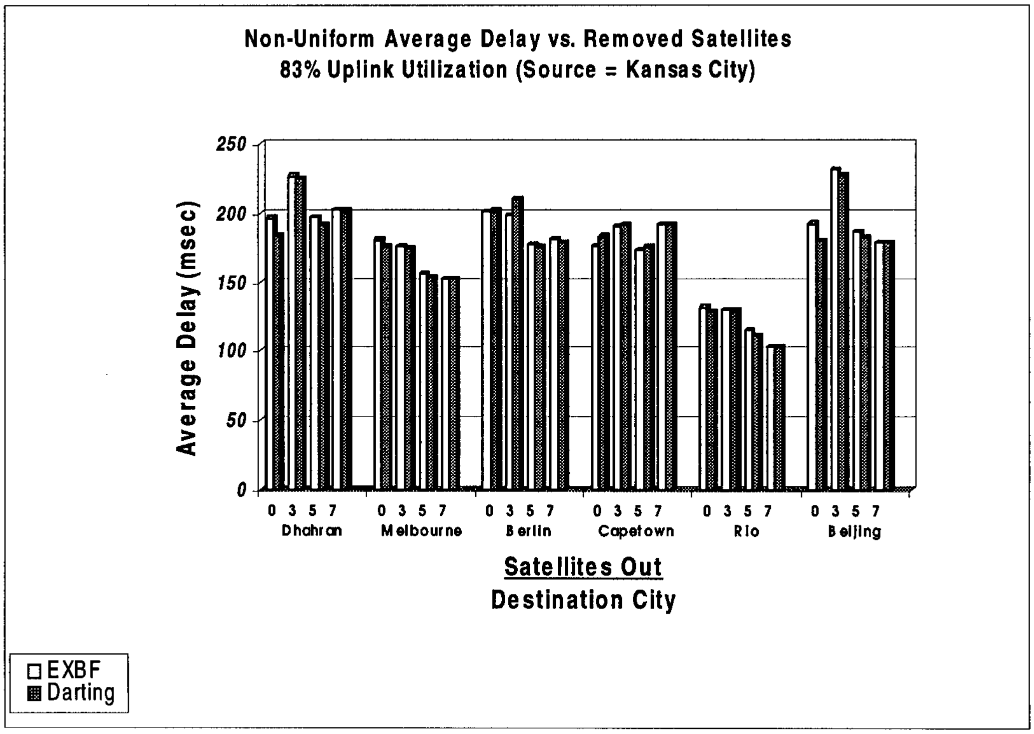

*Figure 20: Non-Uniform Average Delay vs. Removed Satellites, 83% Uplink Utilization*

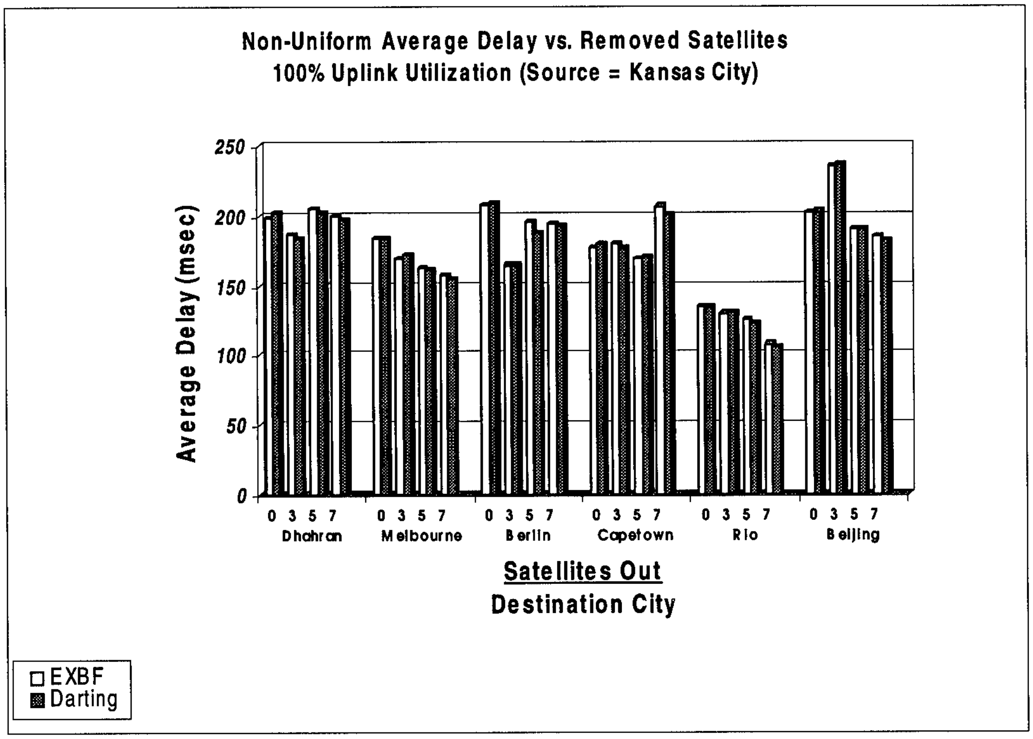

*Figure 21: Non-Uniform Average Delay vs. Removed Satellites, 100% Uplink Utilization*

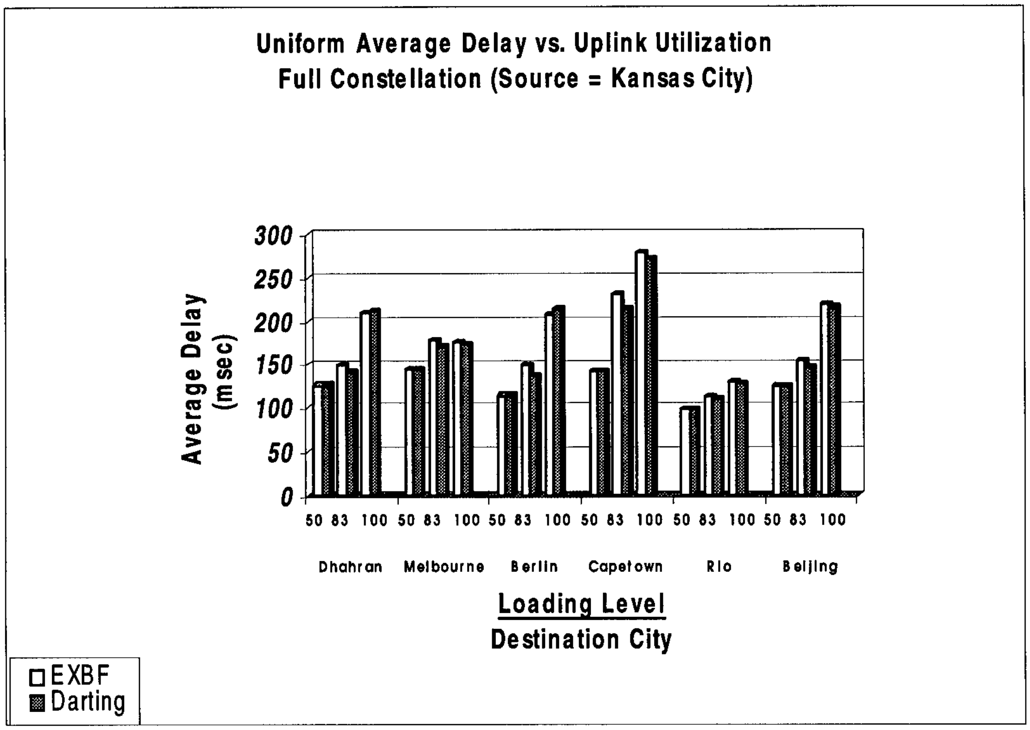

*Figure 22: Uniform Average Delay vs. Uplink Utilization, Full Constellation*

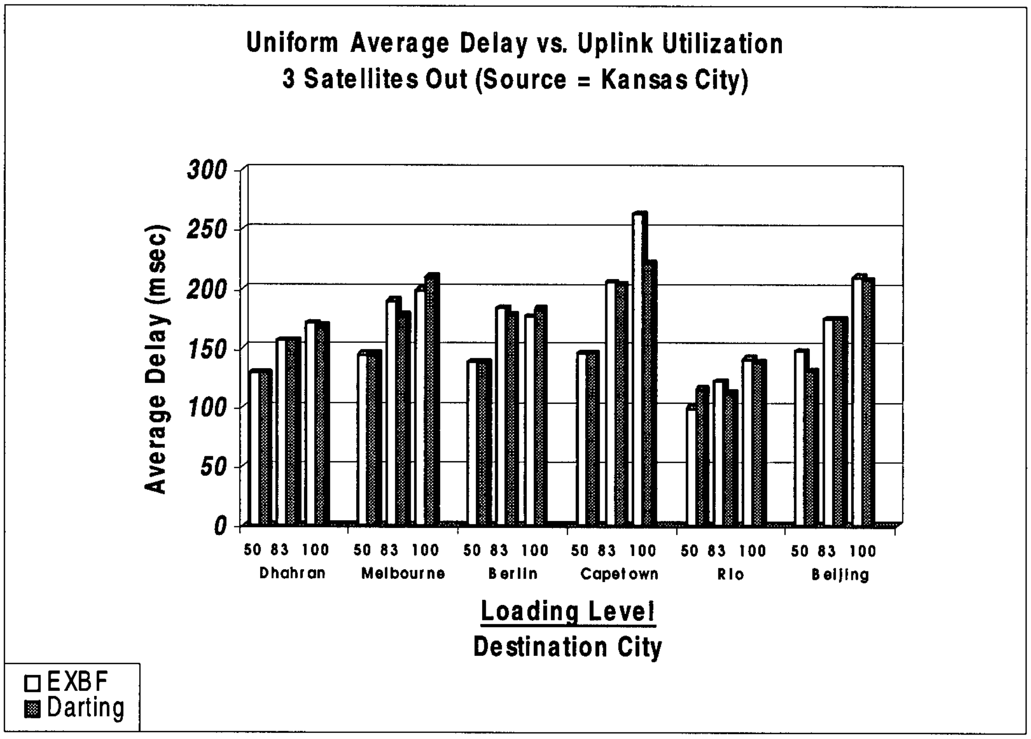

*Figure 23: Uniform Average Delay vs. Uplink Utilization, 3 Satellites Removed*

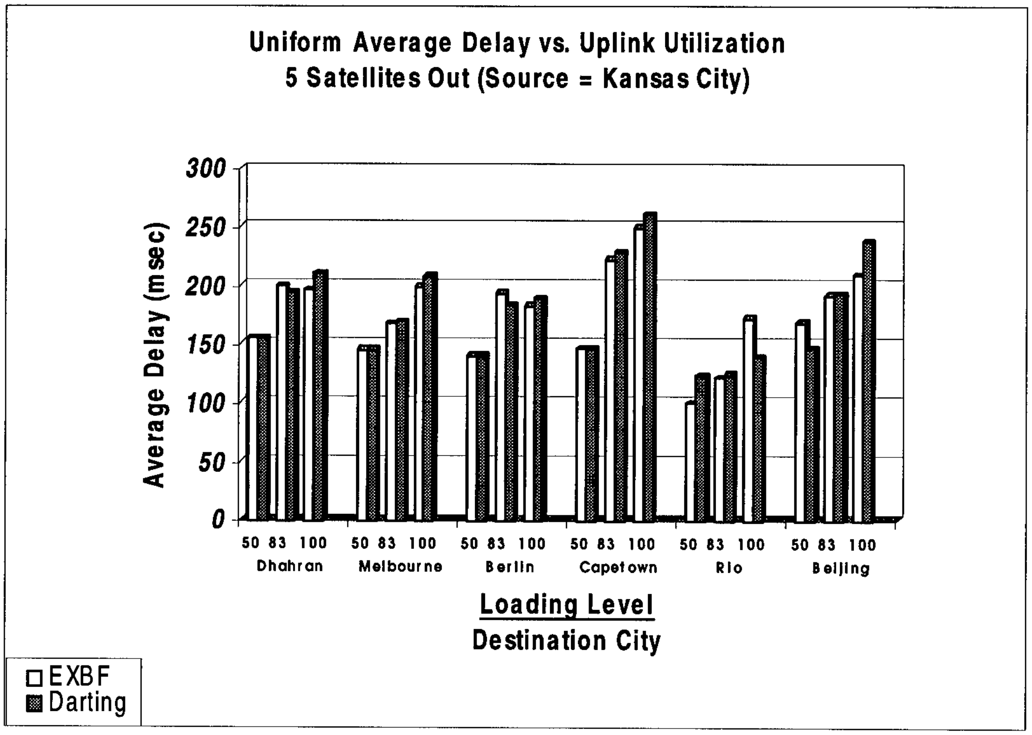

*Figure 24: Uniform Average Delay vs. Uplink Utilization, 5 Satellites Removed*

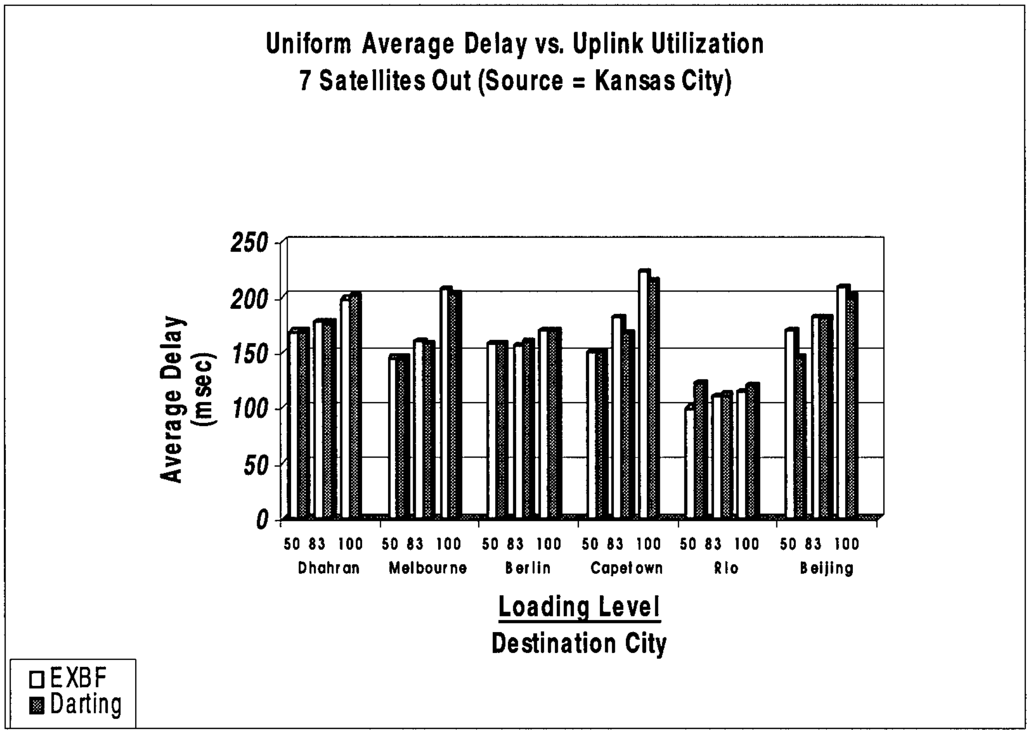

*Figure 25: Uniform Average Delay vs. Uplink Utilization, 7 Satellites Removed*

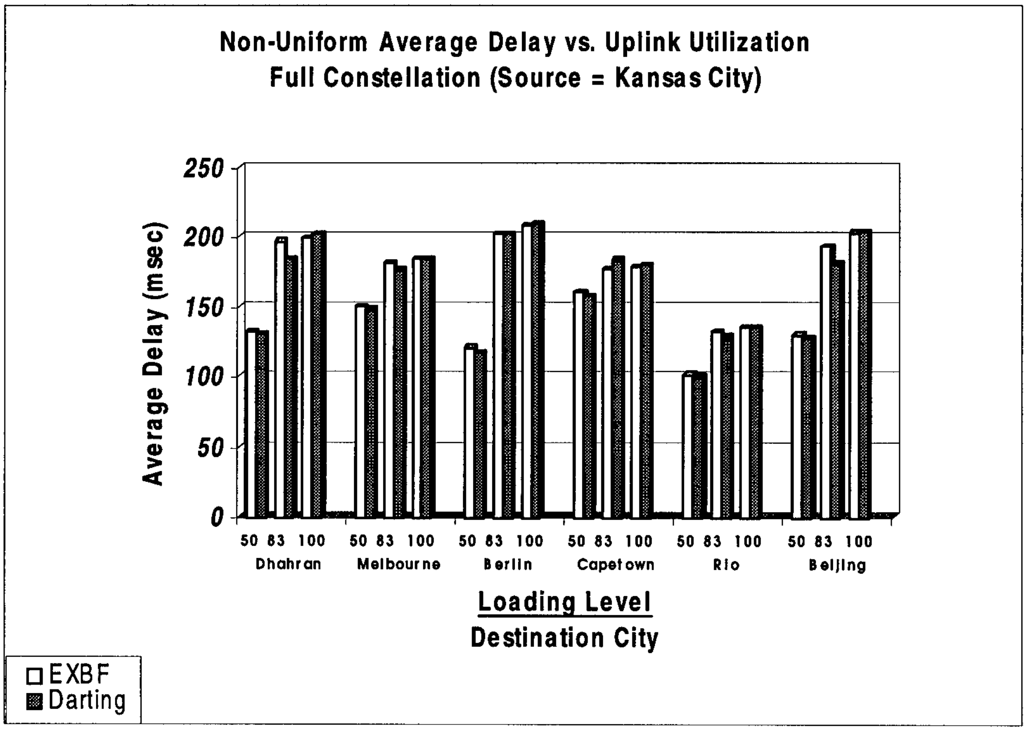

*Figure 26: Non-Uniform Average Delay vs. Uplink Utilization, Full Constellation*

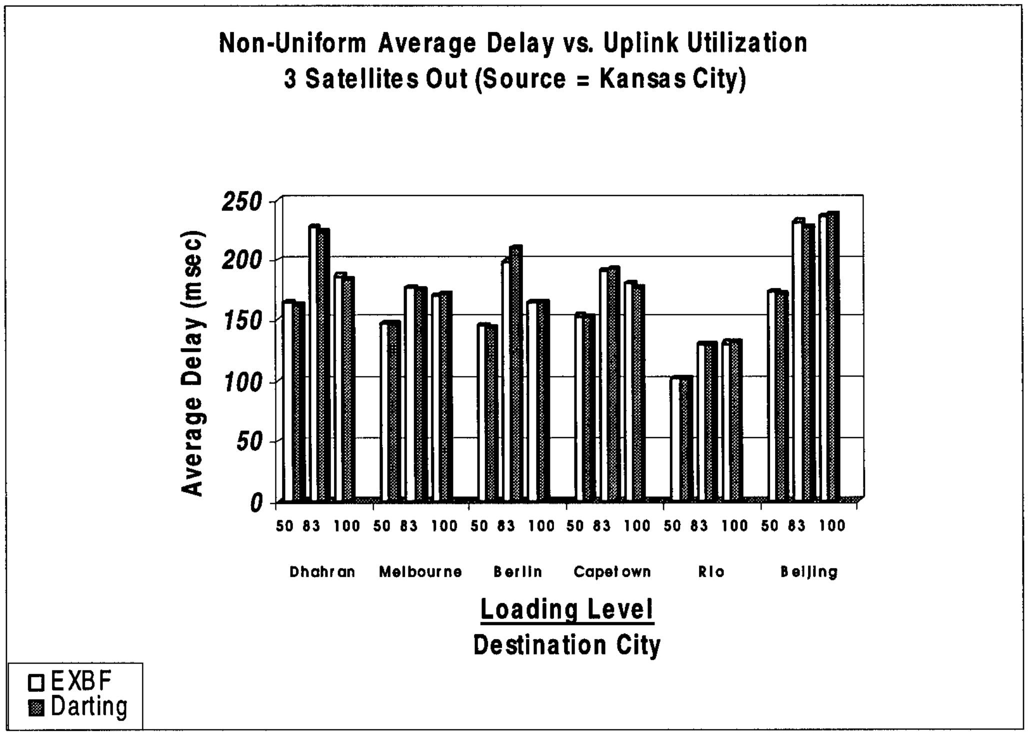

*Figure 27: Non-Uniform Average Delay vs. Uplink Utilization, 3 Satellites Removed*

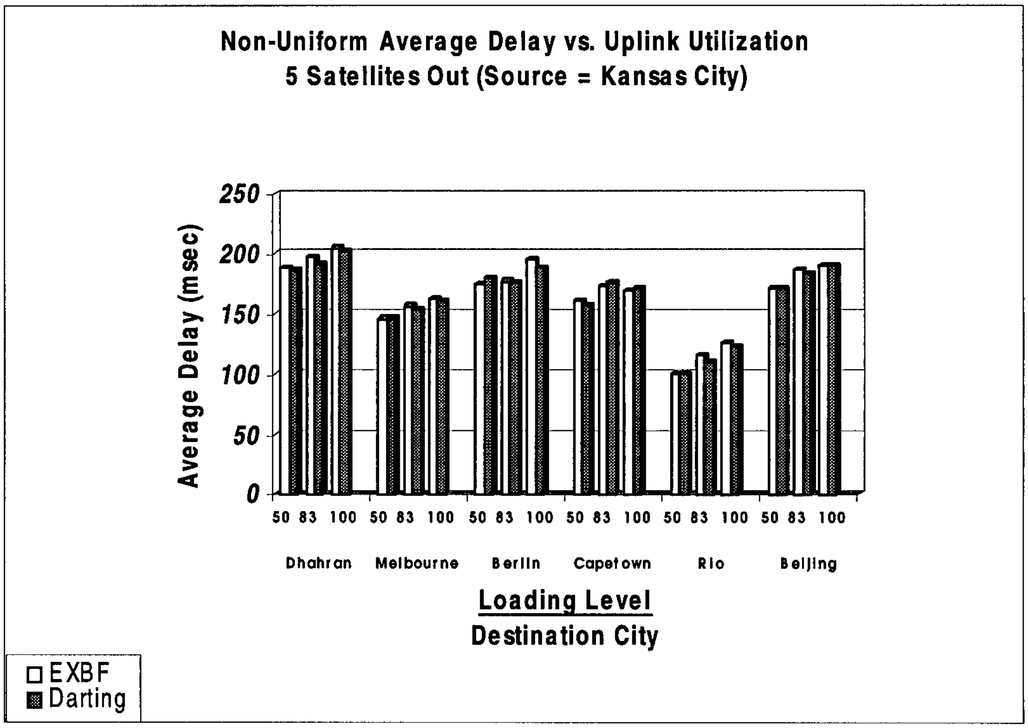

*Figure 28: Non-Uniform Average Delay vs. Uplink Utilization, 5 Satellites Removed*

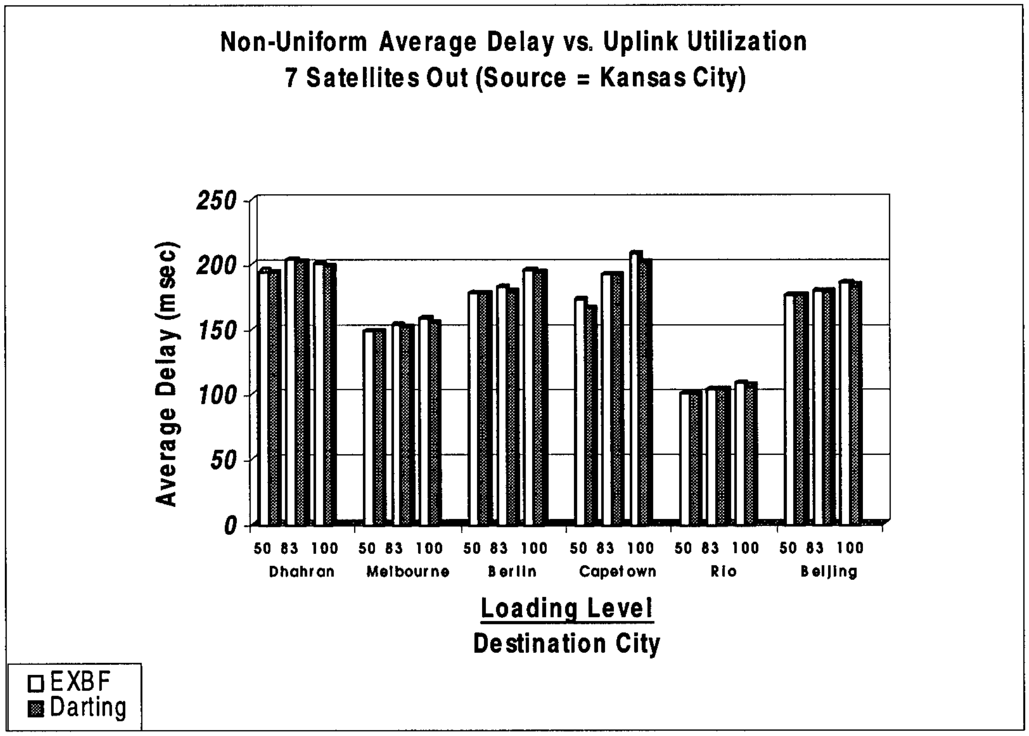

*Figure 29: Non-Uniform Average Delay vs. Uplink Utilization, 7 Satellites Removed*

## APPENDIX C – Overhead Tabular Data

This Appendix contains the tabular data for the overhead metric.

*Table 42: Uniform Overhead*

| Satellites removed | Overhead (50 % Uplink Utilization) | | Overhead (83% Uplink Utilization) | | Overhead (100% Uplink Utilization) | |
|---|---|---|---|---|---|---|
| | EXBF | Darting | EXBF | Darting | EXBF | Darting |
| 0 | 33.69% | 5.36% | 23.09% | 2.31% | 20.12% | 1.57% |
| 3 | 37.17% | 3.11% | 25.90% | 2.35% | 22.70% | 2.00% |
| 5 | 37.06% | 3.16% | 25.81% | 2.31% | 22.60% | 1.96% |
| 7 | 36.96% | 3.38% | 25.73% | 2.30% | 22.53% | 2.01% |

*Table 43: Non-Uniform Overhead*

| Satellites removed | Overhead (50 % Uplink Utilization) | | Overhead (83% Uplink Utilization) | | Overhead (100% Uplink Utilization) | |
|---|---|---|---|---|---|---|
| | EXBF | Darting | EXBF | Darting | EXBF | Darting |
| 0 | 31.67% | 3.59% | 31.67% | 3.50% | 31.67% | 3.49% |
| 3 | 37.14% | 3.54% | 37.14% | 3.77% | 37.14% | 3.49% |
| 5 | 37.04% | 3.57% | 37.04% | 3.80% | 37.04% | 3.69% |
| 7 | 36.94% | 3.40% | 36.94% | 3.29% | 36.94% | 3.33% |

## APPENDIX D – Convergence Tabular Data

This Appendix contains the tabular data for the convergence metric.

*Table 44: Uniform Convergence*

| Satellites removed | Convergence Time (50 % Uplink Utilization) | | Convergence Time (83% Uplink Utilization) | | Convergence Time (100% Uplink Utilization) | |
|---|---|---|---|---|---|---|
| | EXBF | Darting | EXBF | Darting | EXBF | Darting |
| 0 | 4.48 | 11.17 | 24.64 | 19.85 | 26.21 | 20.76 |
| 3 | 18.49 | 13.24 | 25.29 | 19.24 | 11.28 | 17.41 |
| 5 | 12.03 | 13.27 | 24.85 | 14.81 | 18.91 | 17.12 |
| 7 | 18.01 | 11.94 | 21.87 | 14.16 | 9.22 | 15.04 |

*Table 45: Non-Uniform Convergence*

| Satellites removed | Convergence Time (50 % Uplink Utilization) | | Convergence Time (83% Uplink Utilization) | | Convergence Time (100% Uplink Utilization) | |
|---|---|---|---|---|---|---|
| | EXBF | Darting | EXBF | Darting | EXBF | Darting |
| 0 | 16.91 | 16.69 | 17.23 | 18.13 | 18.18 | 18.88 |
| 3 | 18.14 | 17.02 | 18.73 | 18.66 | 18.13 | 18.64 |
| 5 | 16.76 | 16.30 | 17.87 | 17.16 | 17.71 | 17.33 |
| 7 | 12.99 | 14.95 | 14.69 | 16.84 | 14.53 | 16.75 |

## APPENDIX E – Hop Count Tabular Data

This Appendix contains the tabular data for the hop count metric

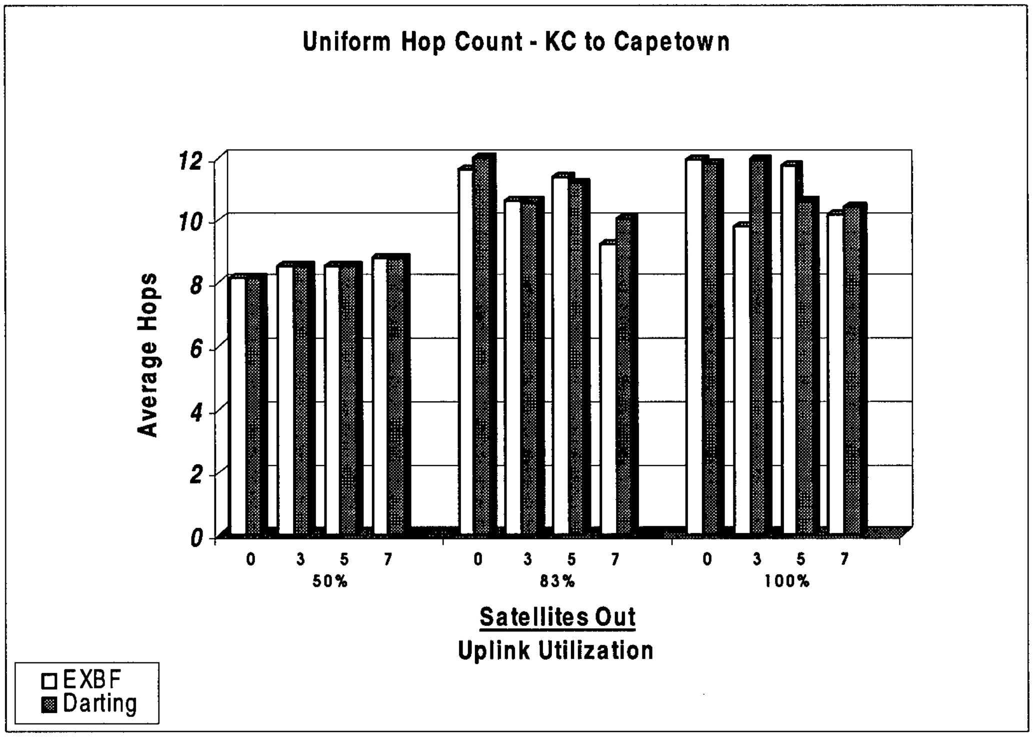

*Figure 30: Uniform Hop Count, KC to Capetown*

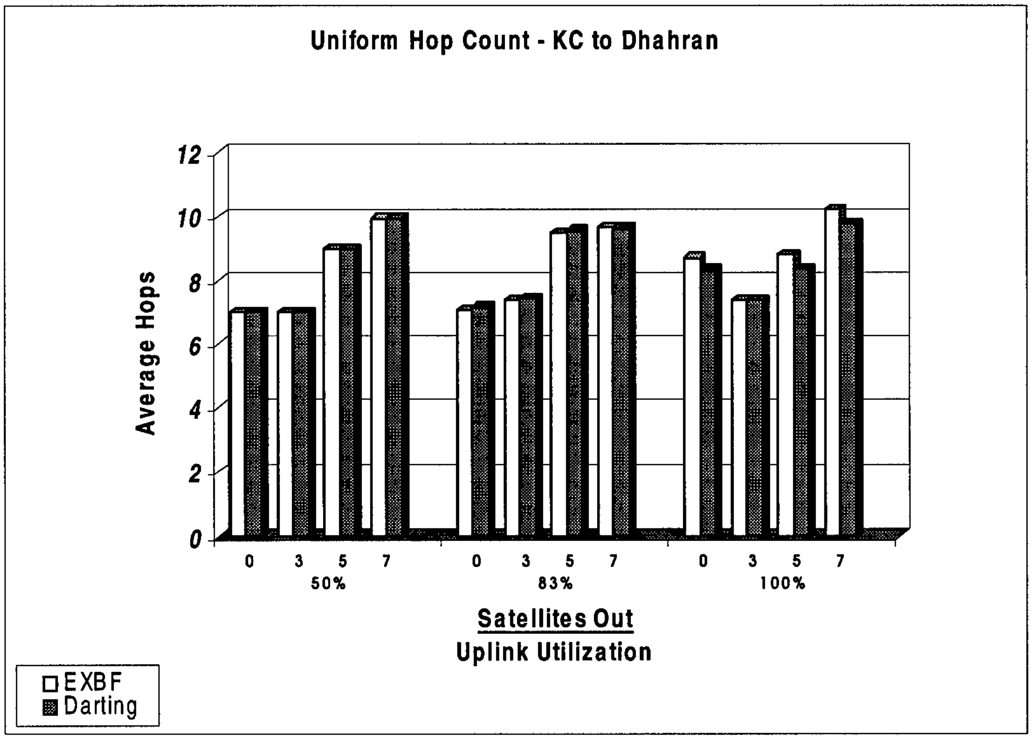

*Figure 31: Uniform Hop Count, KC to Dhahran*

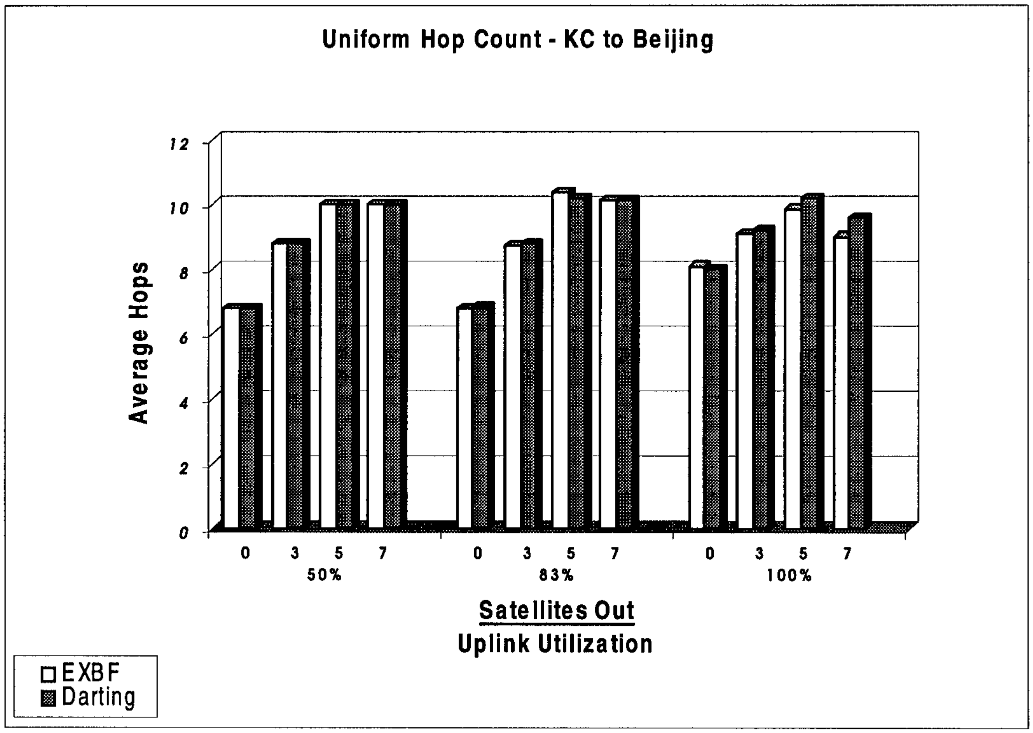

*Figure 32: Uniform Hop Count, KC to Beijing*

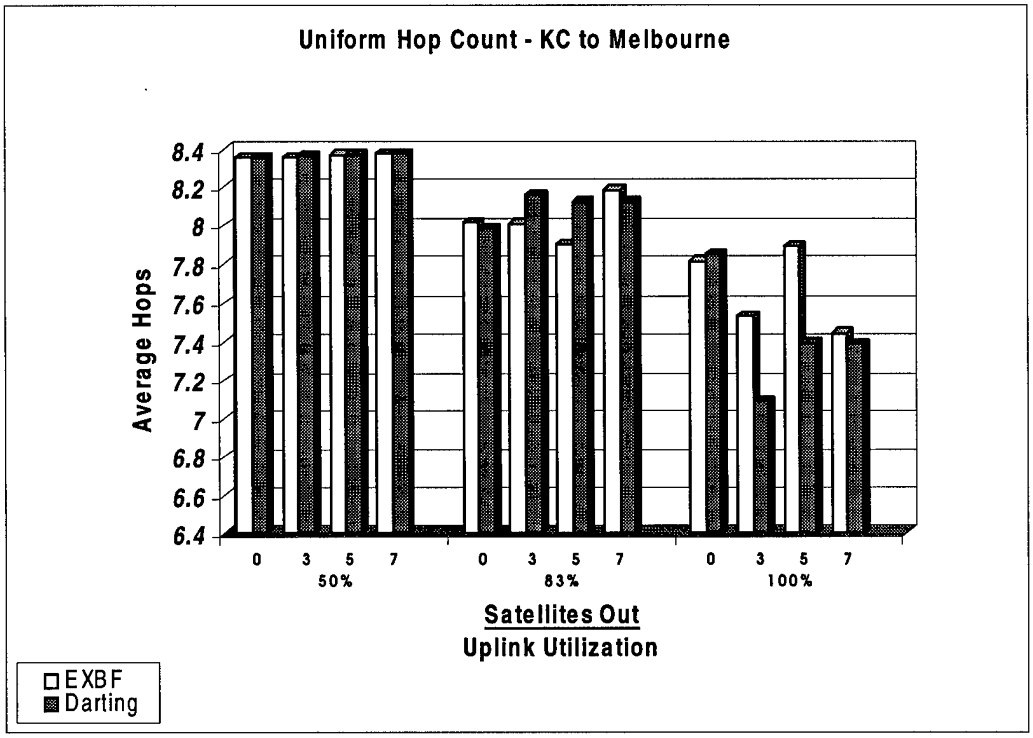

*Figure 33: Uniform Hop Count, KC to Melbourne*

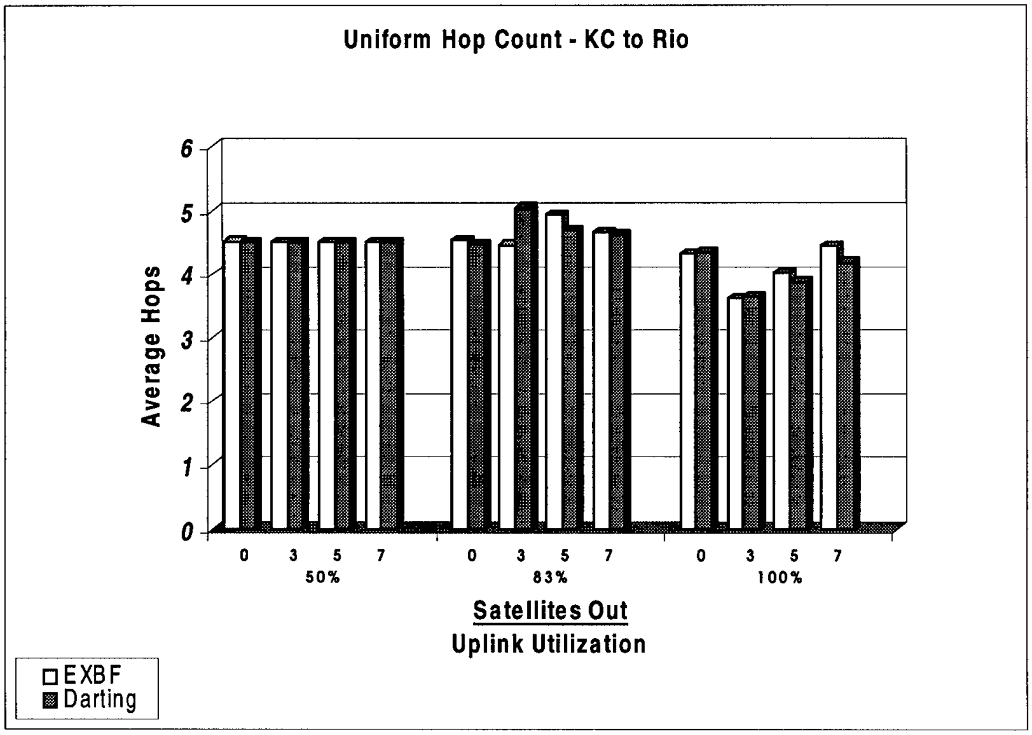

*Figure 34: Uniform Hop Count, KC to Rio*

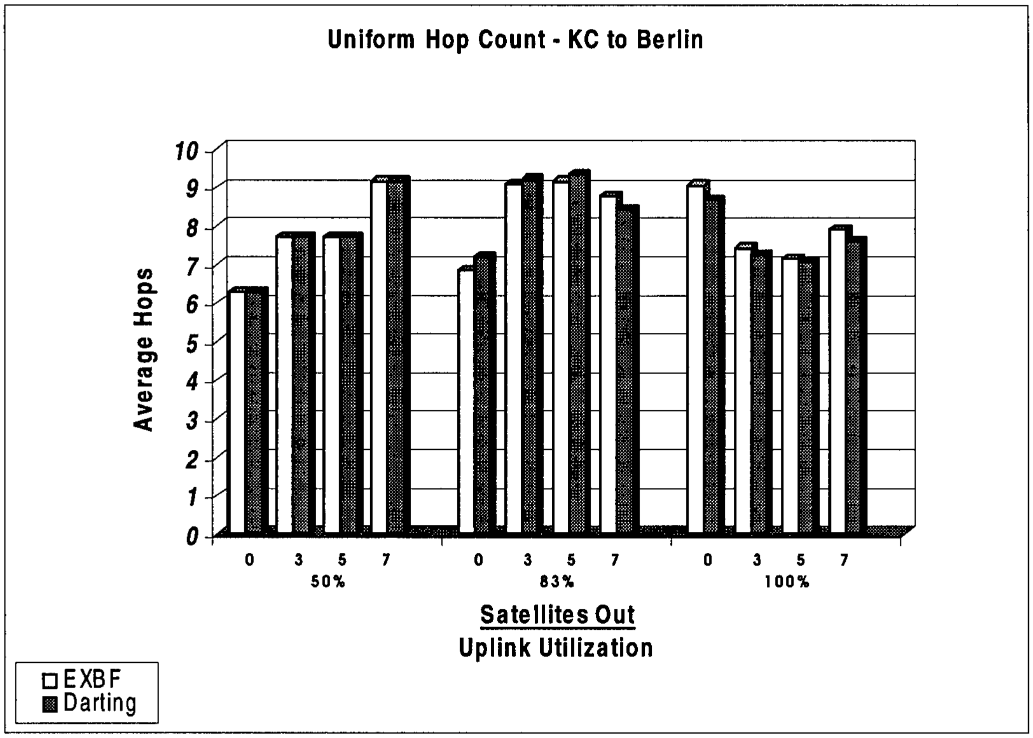

*Figure 35: Uniform Hop Count, KC to Berlin*

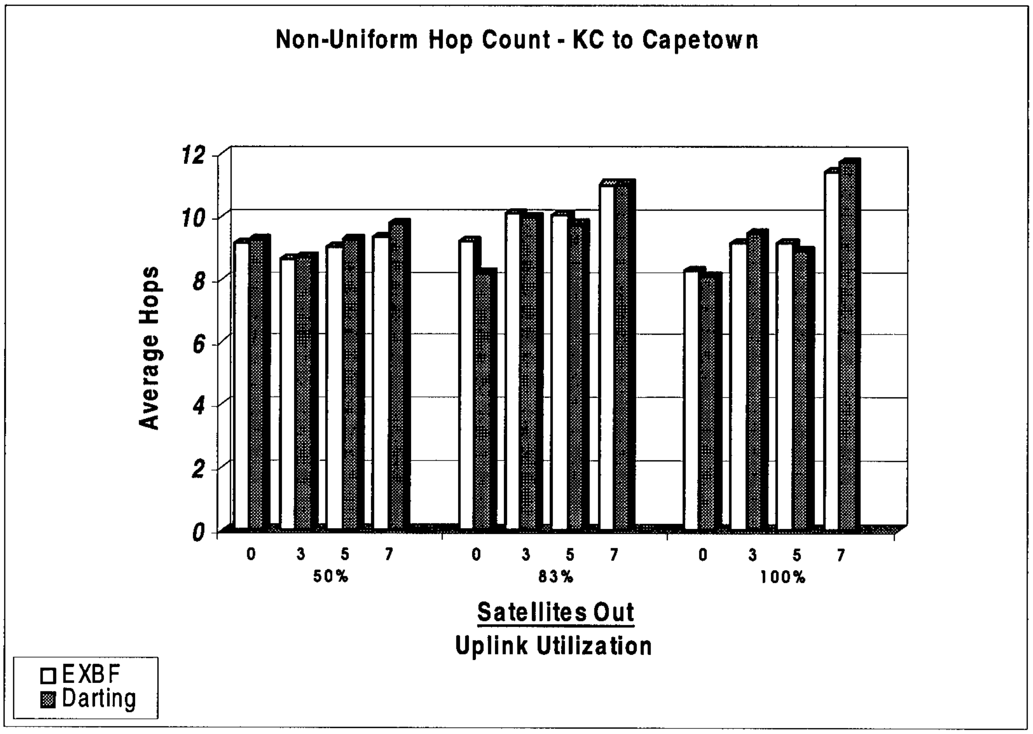

*Figure 36: Non-Uniform Hop Count, KC to Capetown*

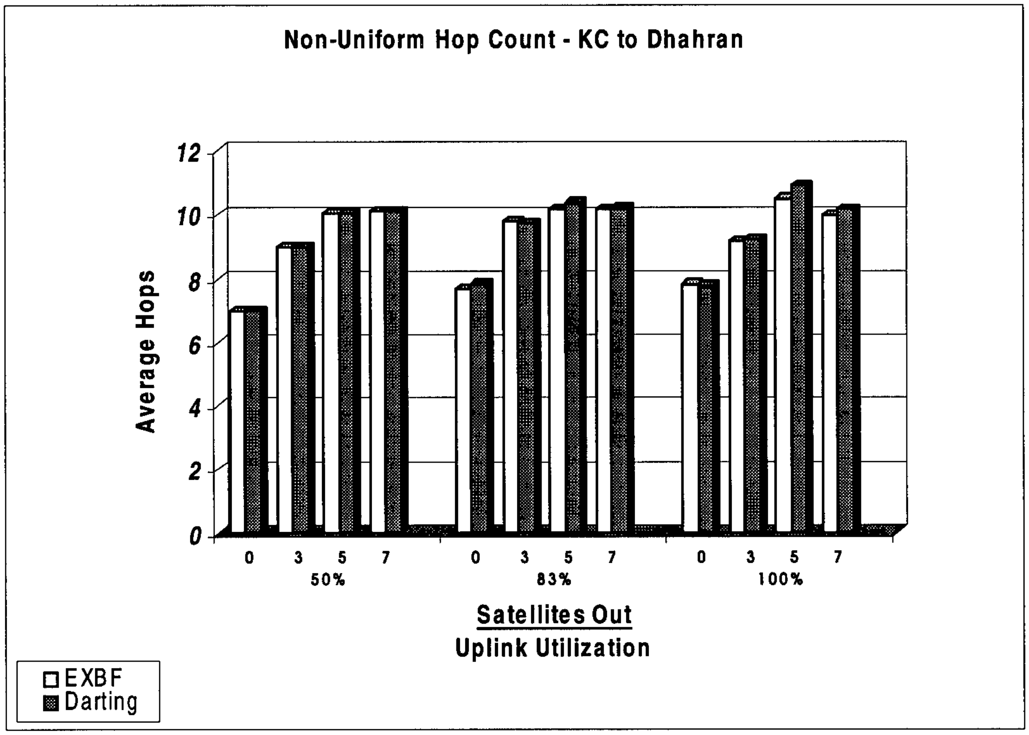

*Figure 37: Non-Uniform Hop Count, KC to Dhahran*

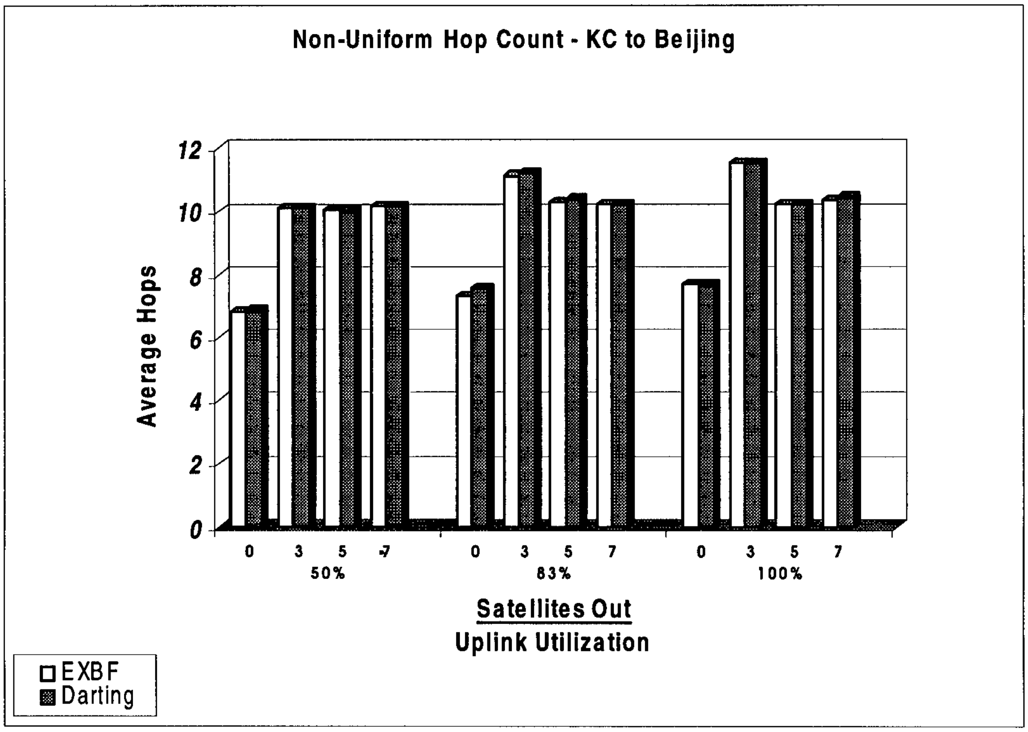

*Figure 38: Non-Uniform Hop Count, KC to Beijing*

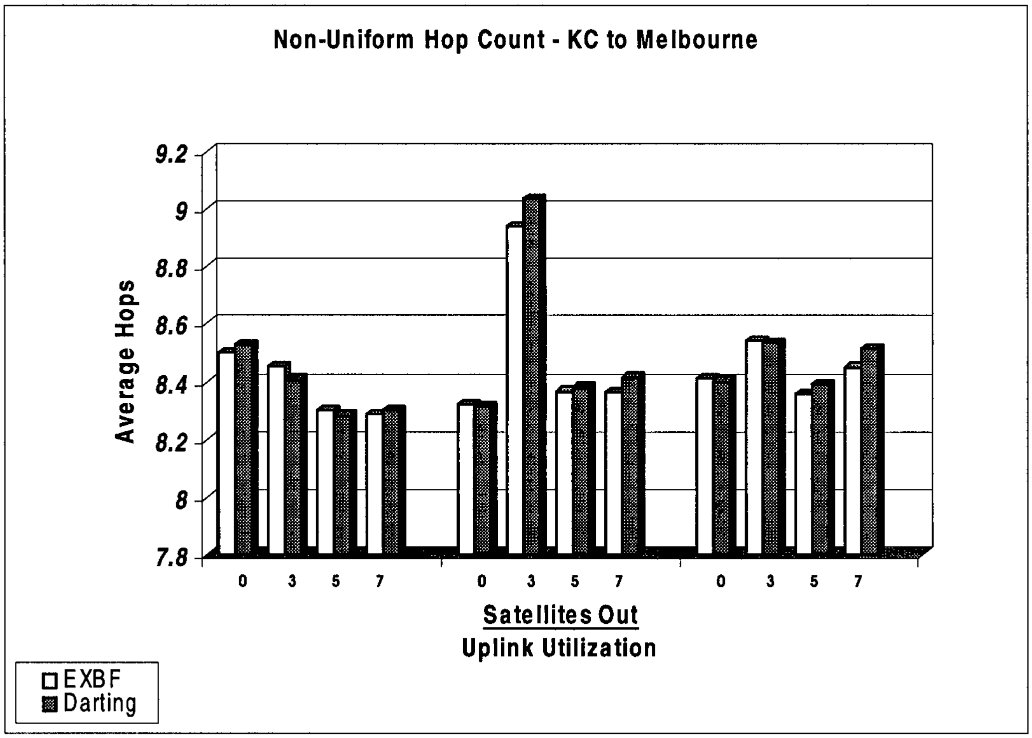

*Figure 39: Non-Uniform Hop Count, KC to Melbourne*

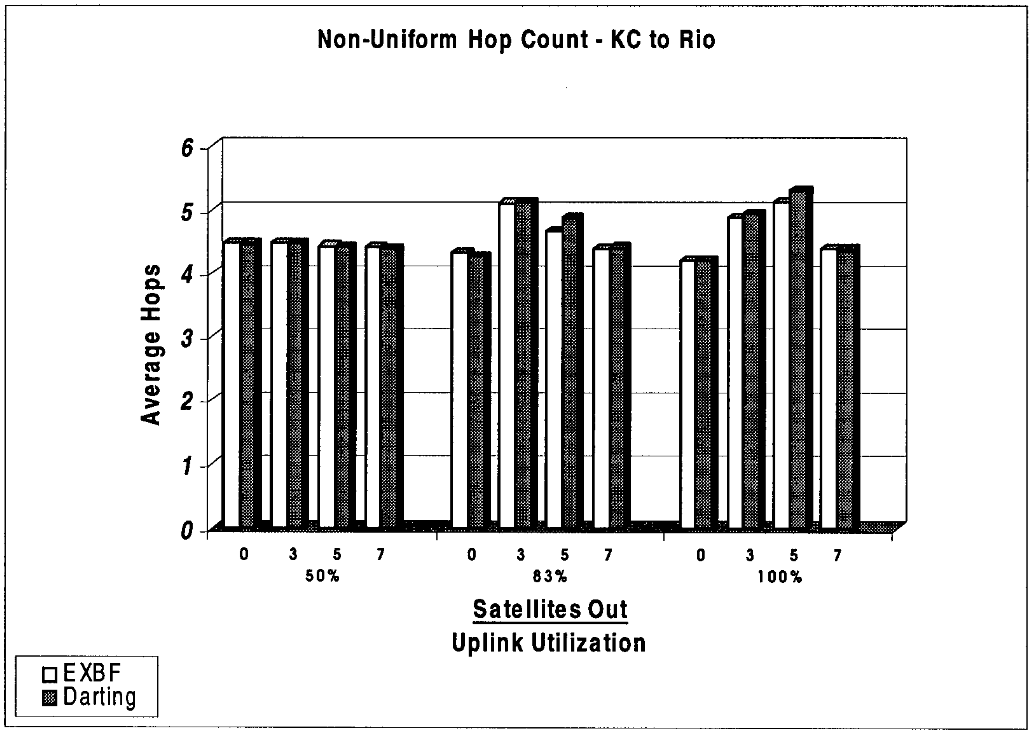

*Figure 40: Non-Uniform Hop Count, KC to Rio*

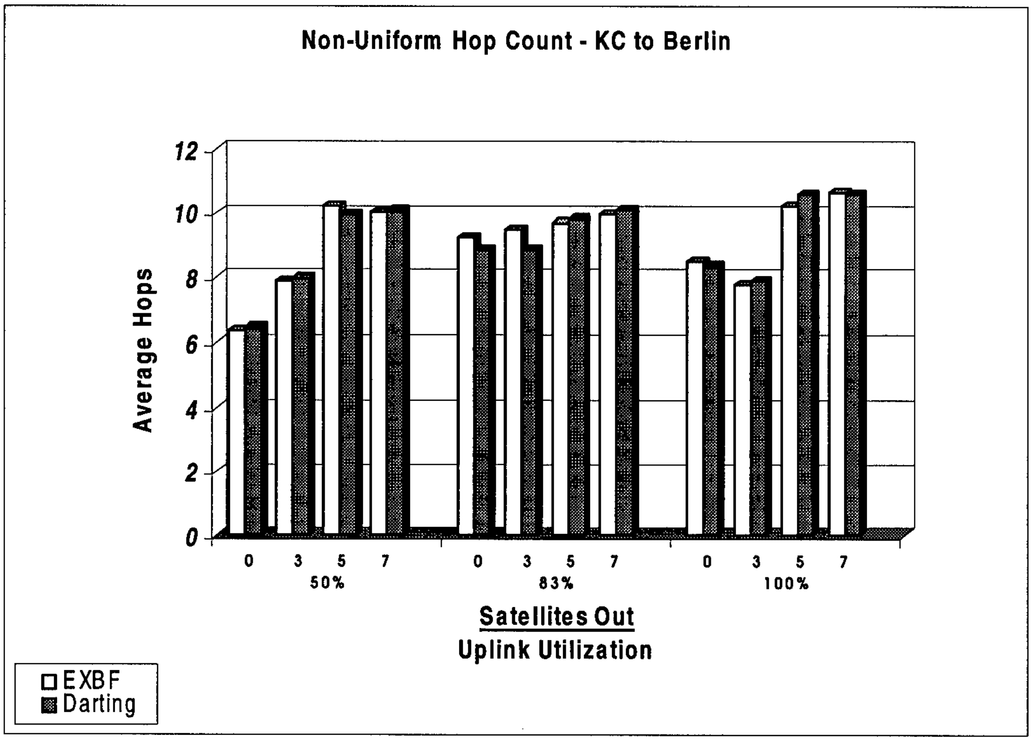

*Figure 41: Non-Uniform Hop Count, KC to Berlin*

# BIBLIOGRAPHY

[AdR87] Adams, W. S. and Rider, L., "*Circular Polar Constellations Providing Continuous Single or Multiple Coverage Above a Specified Latitude*," The Journal of Astronautical Sciences, Vol. 35, No. 2 April-June 1987, pp. 155-192.

[BoA88] Boyd, Adams, Spellman, and Lucas, "*Survivable Space Networks: The Physical Layer*," IEEE Military Communications Conference, Vol. 2, 1988, pp. 521-526.

[Bru96] Brunt, P., "*IRIDIUM® - Overview and Status*," Space Communications, Vol. 14, No. 2, 1996, pp. 61-68.

[Cad94] BoNEs DESIGNER User's Guide, Cadence Design Systems, Incorporated, December 1994.

[Cad95] BoNEs SATLAB User's Guide, Cadence Design Systems, Incorporated, June 1995.

[ChK95] Chang, Kim, Lee, Choi, Min, Yang, and Kim, "*Topological Design and Routing for Low Earth Orbit Satellite Networks*," IEEE Globecom, Vol. 1, 1995, pp. 529-535.

[ChK96] Chang, Kim, Lee, Choi, Min, Yang, and Kim, "*Performance Comparison of Static Routing and Dynamic Routing in Low Earth Orbit Satellite Networks*," 1996 IEEE 46th Vehicular Technology Conference, pp. 1240-1243.

[ChK98] Chang, Kim, Lee, Choi, Min, Yang, and Kim, "*FSA-Based Link Assignment and Routing in Low Earth Orbit Satellite Networks,*" IEEE Transactions on Vehicular Technology, Vol. 47, No. 3, August 1998, pp. 1037-1049.

[ChR89] Cheng, Riley, and Kumar, "*A Loop-Free Bellman-Ford Routing Protocol Without Bouncing Effect*," SIGCOMM 1989, pp. 224-236.

[Com93] Comparetto, Gary M., "*A Technical Comparison of Several Global Mobile Satellite Communications Systems*," Space Communications, Vol. 11, No. 2, 1993, pp. 97-104.

[CoL97] Cormen, Lieserson, and Rivest, Introduction to Algorithms, McGraw-Hill, New York, 1997, pp. 514-615.

[DoK95] Dolev, Kranakis, Krizanc, Peleg, *"Bubbles: Adaptive Routing Scheme for High-Speed Dynamic Networks,"* ACM STOC 1995, pp. 528-537.

[Fos98] Fossa, Carl E., "A Performance Analysis of the Iridium® Low Earth Orbit Satellite System," Master's Thesis, School of Electrical Engineering, Air Force Institute of Technology, 1998.

[Gar86] Garcia-Luna-Aceves, J. J., *"A Minimum-Hop Routing Algorithm Based on Distributed Information,"* Computer Networks and ISDN Systems, Vol. 16, May 1986, pp. 367-386.

[Gav97] Gavish, Bezalel, *"Low Earth Orbit Satellite Based Communication Systems – Research Opportunities,"* European Journal of Operational Research 99, 1997, pp. 166-179.

[Hub97] Hubbel, Yvette C., *"A Comparison of the Iridium® and AMPS Systems,"* IEEE Network, Vol. 11, No. 2, March/April 1997, pp. 52-59.

[KeS96] Keller, Harald and Salzwedel, Horst, *"Link Strategy for the Mobile Satellite System Iridium®,"*1996 IEEE 46th Vehicular Technology Conference, Vol. 2, pp. 1220-1224.

[KiG83] Kirkpatrick, Gelatt, Vecchi, *"Optimization by Simulated Annealing,"* Science, Vol. 220, No. 4598, May 1983, pp. 671-680.

[KrV97] Krishna, Vaidya, and Pradhan, *"A Cluster-based Approach for Routing in Dynamic Networks,"* Computer Communications Review, Vol. 27, No. 2, April 1997, http://www.acm.org/sigcomm/ccr/archive/1997/apr97/ccr-9704-krishna.html.

[Jai91] Jain, Raj, The Art of Computer Systems Performance Analysis, John Wiley and Sons, Inc., New York, 1991 pp. 30-33.

[Jan96] Janoso, Richard F., "Performance Analysis of Dynamic Routing Protocols in a Low Earth Orbit Satellite Data Network," Master's Thesis, School of Electrical Engineering, Air Force Institute of Technology, 1996.

[Lod91] Lodge, John H., *"Mobile Satellite Communications Systems: Toward Global Personal Communications,"* IEEE Communications Magazine, November 1991, pp. 24-30.

[NoC93] Noakes, Cain, Nieto, and Althouse, *"An Adaptive Link Assignment Algorithm for Dynamically Changing Topologies,"* IEEE Transactions on Communications, Vol. 41, No. 5, May 1993, pp. 694-705.

[Pax97] Paxson, Vern, “*End-to-End Routing Behavior in the Internet*,” IEEE/ACM Trans. Networking 5, No. 5, Oct 1997, pp. 601-615.

[PrL93] Price, Kent M. and Leamon, Richard G., “*Definition of Commercial Mobile Satellite Services Network to Meet DoD Communication Needs*,” IEEE MILCOM, Vol. 3, 1993, pp. 821-825.

[RaM95] Radzik, Jose and Maral, Gerard, “*A Methodology for Rapidly Evaluating the Performance of Some Low Earth Orbit Satellite Systems*,” IEEE Journal on Selected Areas in Communications, Vol. 13, No. 2, February 1995, pp. 301-309.

[RaD95] Raines, Richard and Davis, Nathaniel, “*Personal Communications Via Low Earth Orbit Satellite Communication Networks*,” IEEE Military Communications, Vol. 3, 1995, pp. 1229-1233.

[Ram96] Ramesh, R., “*Availability Calculation for Mobile Satellite Communication Systems*,” 1996 IEEE 46th Vehicular Technology Conference, Vol. 2, pp. 1033-1037.

[Re95] Re, Enrico Del, “*Efficient Dynamic Channel Allocation Techniques with Handover Queuing for Mobile Satellite Networks*,” IEEE Journal on Selected Areas in Communications, Vol. 13, No. 2, February 1995, pp. 397-404.

[RiH89] Richaria, Hansel, Bousquet, O’Donnell, “*A Feasibility Study of a Mobile Communication Network Using a Constellation of Low Earth Orbit Satellites*,” Globecomm 1989, Vol. 2, pp. 773-777.

[Ric95] Richaria, M., Satellite Communication Systems, McGraw-Hill, New York, 1995, pp. 16-48.

[Rob98] Robinson, Clarence A., “*Government’s Gateway Grants Global Communications Access*,” Signal, May 1998, p. 65-70.

[RuD96] Ruiz, Doumi, and Gardiner, “*Teletraffic Analysis and Simulation of Mobile Satellite Systems*,” 1996 IEEE 46th Vehicular Technology Conference, Vol. 1, pp. 252-256.

[Rut89] Rutenbar, R. A., “*Simulated Annealing Algorithms: An Overview*,” IEEE Circuits and Devices Magazine, Vol. 5, No. 1, January 1989, pp. 19-26.

[SaA94] Saadawi, Tarek N. and Ammar, Mostafa H., Fundamentals in Telecommunication Networks, John Wiley and Sons, New York, 1994, p. 25.

[Ste96] Stenger, Douglas K., "Survivability Analysis of the Iridium® Low Earth Orbit Satellite Network," Master's Thesis, School of Electrical Engineering, Air Force Institute of Technology, 1996.

[TsM94] Tsai, K., and Ma, R., "*Darting: A Cost Effective Routing Alternative For Large Space-Based Dynamic Topology Networks*," MILCOMM 1995, pp. 682-687.

[UzY97] Uzunalioglu, Yen, and Akyildiz, "*A Connection Handover Protocol for LEO Satellite ATM Networks*," ACM Proceedings IEEE MOBICOM 1997, www.eecom.gatech.edu/users/2875, pp. 204-214.

[Uzu98] Uzunalioglu, Huseyin, "*Probabilistic Routing Protocol for Low Earth Orbit Satellite Networks*," to appear in the IEEE Proceedings ICC 1998, www.eecom.gatech.edu/users/2875.

[WeM95] Werner, Markus, Jahn, Axel and Lutz, Erich, "*Analysis of System Parameters for LEO/ICO Satellite Communications Networks*," IEEE Journal on Selected Areas in Communications, Vol. 13, No. 2, February 1995, pp. 371-381.

[WuM94] Wu, Miller, Pritchard, and Pickholtz, "*Mobile Satellite Communications*," IEEE Proceedings, Vol. 82, No. 9, September 1994, pp. 1431-1447.

# VITA

Captain Stephen R. Pratt was born on January 11, 1967 in Mesa, Arizona. He received an Air Force Reserve Officer Training Corps (ROTC) 4-year scholarship to attend Purdue University, West Lafayette, Indiana. He graduated from on May 5, 1989, with a Bachelor of Science degree in electrical engineering. Upon graduation, he was commissioned as a 2nd Lieutenant in the United States Air Force. From 1989 to 1991 Captain Pratt served as a Test Engineer in the 46th Test Wing, Eglin AFB, FL. While there, he pursued and completed a Master's of Science in management from Troy State University, graduating in March of 1993. He was subsequently assigned in 1994 to the 83rd Fighter Weapons Squadron, Tyndall AFB, Fl as an Advanced Missile Analyst. One year later, he was serving as Telemetry Flight Commander, overseeing all aspects of the squadron's live-fire missile and data collection assets. In 1997, Captain Pratt entered the Air Force Institute of Technology to pursue a Master's of Science degree in computer engineering. Captain Pratt is a graduate of the Air Force's Squadron Officer School. He is level 3 certified in the Test and Evaluation and Systems, Programming, and level 2 certified in Program Management in the Air Force's Acquisition Profession Development Program (APDP). His awards include the Meritorious Service Medal and the Air Force's Commendation Medal.

Permanent Address: 706 Brandeis Avenue, Panama City, FL 32405