12-1996

Survivability Analysis of the Iridium Low Earth Orbit Satellite Network

Douglas K. Stenger

Recommended Citation

Survivability Analysis of
The Iridium Low Earth
Orbit Satellite Network

THESIS

Douglas K. Stenger
Captain, USAF

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Survivability Analysis of
The Iridium Low Earth
Orbit Satellite Network

THESIS

Douglas K. Stenger
Captain, USAF

AFIT/GCS/ENG/96D

DTIC QUALITY INSPECTED 3

19970224 065

AFIT/GCS/ENG/96D-26

Survivability Analysis of

The Iridium Low Earth

Orbit Satellite Network

THESIS

Presented to the faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science (Computer Systems)

Douglas K. Stenger, B. S.

Captain, USAF

December, 1996

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

AFIT/GCS/ENG/96D-26

ABSTRACT

This thesis evaluates the survivability of the proposed Iridium Low Earth Orbit (LEO) Satellite Network. In addition to the complete Iridium constellation, three degraded Iridium constellations are analyzed. This analysis occurs via the use of simulation models, which are developed to use three dynamic routing algorithms over three loading levels. The Iridium network models use a common set of operating assumptions and system environments. The constellation survivability was determined by comparing packet rejection rates, hop counts, and average end-to-end delay performance between the various network scenarios. It was concluded that, based on the established scenarios, the proposed Iridium constellation was highly survivable. Even with only 45 percent of its satellites functioning (modeled with 36 failed Iridium satellites), the average packet delays were never greater than 178 milliseconds (msec), well within the real-time packet delivery constraint of 400 msec. As a result, while additional research is necessary, Iridium has demonstrated the network robustness that is required within the military communications environment.

Survivability Analysis of

The Iridium Low Earth

Orbit Satellite Network

CHAPTER 1

INTRODUCTION

1.1 Background

Reliable communications have always been a crucial part of any military operation. As the pace of warfare and the technological complexity of weaponry have increased, so has our need for rapid information to assess battlefield conditions. The increased speed and complexity of modern warfare was typified by the Gulf War, where the pace of the ground war moved quickly.

The Gulf War demonstrated the key role satellite communications will play in future conflicts. In the past several decades, the military has relied on a combination of land line phone systems and line of sight communications. Neither was available in theater during the buildup for Desert Shield, forcing dependence on satellite-based communications. At the peak of operations, 700,000 phone calls and 152,000 traffic messages were processed per day, with 75% of the communications being sent over the Defense Satellite Communications System, 5% over NATO assets, and 20% over commercial satellites [TuA93].

Until recently, GEO satellites have been the primary focus for military communications. The first MARISAT GEO satellite was launched over the Pacific Ocean in 1976 to provide communications between ships and shore stations. Because of the combination of high cost and unacceptably large equipment associated with GEO satellites, the primary application of military satellite communication, at least until recently, was for ship-to-shore communication. In addition, GEO satellite systems experience much larger propagation delays than do Low Earth Orbit systems, making voice communication difficult, at best. GEO satellites are thus infeasible for the next generation of satellite communications, given the state of technology at this time.

Until recently, the focus of mobile communications has been on land-based networks. However, with progress made in digital voice processing, satellite technology, and component miniaturization, satellite-based communications systems have become viable [Com93]. LEO satellites networks offer a number of distinct advantages and a significant number of LEO communications networks have been proposed. In addition to offering much smaller propagation delays, the satellites and corresponding ground equipment are smaller and less costly than corresponding GEO equipment. Most importantly, particularly to the military, LEOs offer the capability for two geographically separated users to connect via a global cellular telephone system, regardless of the terrain, using nothing more than a handheld telephone.

1.2 Research Goals

Currently, no research results have been found in published literature, evaluating the robustness of the proposed LEO networks in a faulting environment.

This research will concentrate on determining the minimum acceptable constellation, with respect to service degradation, for the proposed Iridium LEO satellite constellation through the use of computer simulation. The Iridium constellation was selected for two primary reasons. First, this constellation utilizes intersatellite links and

complex onboard computer processors to handle user traffic. This approach (described in more detail in Chapter 2) is much more technically challenging for constellation designers than the current method (bent-pipe) which is being implemented by most of the other proposed LEO satellite constellations. Second, Iridium appears to be commercially viable. Current plans are for Iridium to be operational by mid-1998.

In addition to the constellation configuration used, a crucial factor impacting constellation performance is the message routing algorithms used. In this study, three routing algorithms (described in Chapters 2 and 3) were implemented. The results obtained from implementing these algorithms were compared in terms of the performance metrics described below.

The survivability aspects of the proposed Iridium constellation were evaluated using three performance metrics. First, the delay a data packet experiences in traversing the satellite network from user to receiver was monitored. The number of hops a packet is required to navigate in order to reach its destination was the second metric used in this evaluation. The last metric, packet rejection rate, recorded the number of dropped (lost) packets during the evaluation period. Together, the results of these performance metrics were used to derive the conclusions reached in Chapter 5.

1.3 Summary

This chapter has presented a brief history of satellite communications. Recent technological advances have made LEO constellations an attractive alternative to existing mobile communications systems. One key concern with these proposed LEO constellations is its survivability characteristics. This study evaluates the survivability of the proposed Iridium LEO constellation.

Chapter 2 presents a discussion of these proposed satellite communications systems, focusing primarily on LEO satellite networks. The rational for favoring LEO satellite networks over other competing satellite networks based on either Geostationary

Earth Orbit (GEO) satellites or Highly Elliptical Orbit (HEO) satellites is analyzed, along with an overview of the various facets of a typical LEO satellite network. Following this, several proposed LEO constellations are introduced and compared. In closing, two key elements of the LEO environment are discussed: survivability and message routing techniques.

In Chapter 3, the research methodology for this effort is examined. After presenting the operating assumptions, design parameters and factors which are required, the simulation models used to carry out the research, are described. The validation and verification procedures used for these models are also detailed, before closing with a discussion of testing methodology and various factors, which constrained the research.

Chapter 4 analyzes the survivability performance of the LEO satellite networks by presenting the computer simulation results obtained. Corresponding recommendations and conclusions are drawn in Chapter 5 from this data.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Only once existing in the imagination of a science fiction writer, satellite-based communications systems have now become indispensable to our mobile society. This is particularly true of the military. While wars were once fought by vast armies of infantrymen, today highly mechanized units quickly sweep across large regions of territory, leaving current communications capabilities spread thin. This is true especially in regions where little or no communications infrastructure exists. This problem was typified during Gulf War, when the pace of the war moved so rapidly, military planners had extreme difficulty in evaluating the progress of the military operation.

Long infeasible due to technological constraints, numerous satellite communications systems have been recently proposed which take advantage of the newest available technology. These systems, while varying considerably in their approaches, claim they will be able to facilitate point-to-point communication anywhere on Earth, regardless of whether or not any communications infrastructure is present. This chapter presents a discussion of these proposed systems, focusing primarily on Low Earth Orbit (LEO) satellite networks.

Section 2.2 presents the rationale for favoring Low Earth Orbit satellites in the mobile communications realm over either Geostationary Earth Orbit (GEO) satellites or Highly Elliptical Orbit (HEO) satellites. A discussion of the various facets of the LEO environment comprises Section 2.3. The proposed Iridium, Teledesic and Globalstar satellite constellations are introduced in Section 2.4. This section focuses on the respective implementation, types of technologies used, and a comparison of the two

approaches. Section 2.5 describes techniques and dynamic routing algorithms utilized for traffic optimization. Survivability, the primary focus of this effort, is covered in Section 2.6, with a discussion of the survivability framework and a presentation of network failure recovery issues. In closing, Section 2.7 summarizes the information covered in this chapter.

2.2 Why Low Earth Orbit (LEO) satellites

A significant number of satellite communications systems have been proposed to meet the growing demand for mobile communications. These proposed constellations differ considerably in number of satellites, relative complexity of the satellites and associated tracking equipment, and operational performance tradeoffs. These differences are primarily a function of their altitude and orbit characteristics. This section compares and contrasts Geostationary, Highly Elliptical, and Low Earth Orbit satellite systems, pointing out why Low Earth Orbit satellites are best suited for the mobile communications environment.

2.2.1 The Geostationary (GEO) Satellite System A GEO satellite is located 35,797 kilometers (km) above the Earth's surface. The satellite orbit lies in the equatorial plane and appears from Earth to be fixed. Because of this, three GEO satellites, equally spaced 120 degrees apart, provide global coverage for all latitudes below 70 degrees.

GEO systems provide a number of benefits. First, this type of system provides world-wide coverage with a small number of satellites. Thus, fewer launches are required relative to other systems, such as LEO satellite systems, which require many satellites to provide similar coverage. In addition, the tracking of GEO satellites is greatly simplified or eliminated since the satellite locations are fixed with respect to the Earth (ignoring gradual drifts in the orbital location).

Inherent in GEO systems are a number of disadvantages which make LEO-based systems more attractive. Because of the large distance between GEO satellites and Earth,

the one way propagation delays are significant, 120 milliseconds versus three to four milliseconds for LEO systems. In addition, because of the large distance, higher transmitter power is required to compensate for the expected transmission energy losses. Thirdly, GEO satellites need to have their electronic equipment radiation hardened to avoid being damaged during launch. Because they pass through the Van Allen radiation belts enroute to their orbit, both the cost of launch and weight of the orbital vehicle are increased. The final problem involves the lack of coverage in both the far northern and southern hemispheres due to the elevation angles. Because of propagation anomalies near the horizon, the practical working limit for GEO systems is around 75 degrees. Even at this latitude, a great deal of theory and evidence suggests such service won't be consistent [WuM94].

2.2.2 Highly Elliptical Orbit (HEO) Satellite System Several HEO satellite systems have been proposed to meet the growing demand for global mobile satellite communications, among them Ellipsat [WuM94]. Satellites in this type of orbit vary in altitudes as close as 1,000 km and as far away as 40,000 km. These orbits can be inclined, relative to the equator, to provide coverage for a given region. HEO systems are built to communicate best when the satellite is farthest from the Earth.

HEO systems have a number of inherent problems as well. The satellite electronic equipment must be hardened to prevent damage from radiation. In addition, since the satellite is moving relative to the earth's surface, a Doppler shift between the transmitted and received signal must be corrected to maintain proper ranging and communication data. Steerable antennas are also required to maintain coverage over the desired region. Finally, more HEO satellites are required to maintain the same continuous coverage region as a GEO system.

2.2.3 Low Earth Orbit (LEO) Satellite System A myriad of LEO satellite systems have been proposed to meet the growing demand for global mobile satellite communications, among them Iridium, Teledesic and Globalstar. Satellites in a LEO

system maintain orbits in the range of 500 to 1500 km and have circular orbits. The LEO systems generally fall into two categories, "Big" or "Little" [WuM94]. Big Leo systems were designated as such, because their satellites would have the necessary power and bandwidth to provide near-toll-quality voice service to hand-held transceivers; they also provide other services such as paging, facsimile, and data transmission. Little Leo systems were characterized as such, because of the expected small satellite size and mass required to provide low bit rate services (1kb/s); such as two-way messaging, and positioning information.

Numerous LEO systems have been proposed because of the advantages they offer over both GEO systems and HEO systems. Propagation times are many times smaller for LEO systems, versus those for GEO systems. In addition, the constellation fault tolerance, due the large number of satellites available for a given coverage area, implies that a single satellite failure will not result in the loss of communication coverage. Also, because of the lower satellite altitudes, the transmitter power levels can be lowered (compared to GEO satellites), as well as the satellite weight (compared to both GEO and HEO satellites) since LEO satellites orbit below the Van Allen radiation belts. Because of a smaller, lighter satellite, multiple LEO satellites can be launched using a multiple launch vehicle, such as the Space Shuttle or Pegasus, reducing the cost per launch.

LEO satellite systems are not without their disadvantages. First, similarly to HEO systems, many more satellites are required to cover the same area as a GEO system. As discussed earlier, three GEO satellites, placed 120 degrees apart can achieve global coverage, while the proposed LEO system, Iridium, requires 66 satellites for global coverage. Secondly, as in HEO systems, LEO system receivers must also compensate for Doppler shifts in frequencies due to the shifting satellite position relative to the Earth. Thirdly, intersatellite (or crosslinks) are required to provide communications between geographically separated end users. This requires an efficient, robust method for consistent message delivery, involving the use of dynamic routing algorithms.

While LEO systems have drawbacks which cannot be ignored, they provide the necessary flexibility that land-based mobile communications systems require. A survivable, reliable, low cost service is attractive to both the civilian and military community.

2.3 The Low Earth Orbit Environment

Unlike the GEO environment, which has existed for decades and relied on stable technology, the LEO systems are based on emerging technology and relatively new concepts, thus simultaneously creating not only great promise, but also great risks. This fact can be seen in the often divergent approaches different companies are taking in their proposals. Globalstar, Teledesic and Iridium, discussed later in this chapter, provide a clear example of this. This section discusses the relevant aspects of LEO systems as well as pros and cons of various approaches.

2.3.1 The LEO Network Configuration One of the most important factors of a LEO system is the configuration of the network. The first aspect of this, are the communications nodes, of which there are three general categories [WeB93]. One key element is the LEO satellite, which has transmission, and possibly, a switching function. The second element is the terrestrial gateway station, which provides access to existing Public Switched Telephone Network/Public Data Networks (PSTN/PDNs). In addition to the interface, the gateways also provide switching and network management, again dependent on the network implementation. The final set of network nodes are the terminals of the fixed and mobile users who generate and receive the traffic. The other key network component is the communication links. Types of links are the Mobile User Links (MULs), which link mobile users with an overhead satellite and the Gateway Link (GWL), which links satellites and gateways in their coverage area. Thirdly, links are necessary between the gateways and the PSTN/PDNs. This allows full utilization of existing telephone and data networks. The final type of link is the Intersatellite Link

(ISL). ISLs provide interconnectivity between satellites, either within the same orbit (intra-plane) or in adjacent orbits (inter-plane).

Two primary approaches have been used for the design of the proposed LEO network. By far, the most popular approach involves extensive use of gateways in conjunction with the LEO satellites [CoM93]. Under this design, each gateway can always see at least one satellite of the constellation. Transmissions to a distant user are uplinked to a nearby satellite, which downlinks to a new gateway closer to the distant user. This "bent-pipe" process continues until the transmission reaches the destination user. The other design approach relies heavily on ISLs. In such a system, a user who wishes to communicate to a distant user, uplinks their information to a nearby satellite, which routes the information to an adjacent satellite closer to the end user. The process continues until the information reaches a satellite which has the distant user in its coverage area, at which time the satellite downlinks the information to the distant user.

The bent-pipe approach has some important advantages and disadvantages. Perhaps the largest advantage results from its use of more stable, proven technology [CoM93]. This substantially reduces the risks, always faced, when implementing a new technology. In addition, since all of the system's processing and switching operations are ground based, gateway maintenance problems can be easily corrected and newly developed technology can be implemented without having to launch a new satellite. There are several disadvantages for this method. The first is the vulnerability [TuA93] of the gateways to sabotage, which is particularly critical during wartime, when the system is likely to be needed the most. In addition, significant regulatory and network management issues face the potentially hundreds of gateways needed to cover all land areas [Ana95]. These issues include: frequency licensing and authorization requests, and authorization to build and run the facilities. Integrating these issues in the international environment could prove to be quite a challenge.

ISL-based systems, while considered by some to be quite risky [CoM93], also hold the promise of great rewards. These systems are attractive for a number of reasons [WeJ95]. Connectivity is a key advantage. This allows the possibility of routing long-distance traffic through multiple satellites, which increases the autonomy of the system, reduces the often uncontrollable PSTN link costs, and may even reduce communications delay. According to Motorola designers [TuA93], these systems are also much less vulnerable to gateway sabotage since ISL-based networks operate independent of gateways. Finally, the use of ISLs is well suited to carry signaling and network management traffic. ISL-based systems have significant disadvantages as well [WeJ95]. First, the presence of ISLs on a satellite results in additional weight, complexity, and the cost of the satellite payload, which includes the ISL antennas, transmitters, and receivers along with the routing capability. The estimated weight for a Globalstar satellite, a bent-pipe system, is 510 lbs, while the estimated weight for an Iridium satellite, an ISL-based system, is nearly three times as heavy at 1,516 lbs [CoM93]. Secondly, satellite complexity is increased by required ISL pointing, acquisition, and tracking (PAT) [WeJ95]. PAT requires steerable ISL antennas, as well as steerable gateway antennas, on board the satellite. Finally, local telephone operators may consider ISLs a rival to their terrestrial networks, causing system developers difficulty in negotiation of national landing rights.

2.3.2 The LEO Satellite Constellation Selection of the appropriate satellite constellation configuration is also crucial to system success. Both early and recent studies [Wan93, Wal70, Wal77] examined two promising configurations, the star network and the delta network. Walker concluded when whole-Earth coverage zones require coverage by more than one satellite; the delta network is preferable to the star network [Wal77].

Many of the proposed LEO systems are based on the delta network. This network contains a total of T satellites, with m satellites evenly spaced in each of n orbital planes,

so T = m × n [Wan93]. Generally, all the n orbital planes have the same inclination to a reference plane, usually coinciding with the equatorial plane. The ascending nodes of the orbital planes are evenly spaced, at intervals of $360^\circ/n$, in the reference plane. The m satellites are also evenly spaced, at intervals of $360^\circ/m$, within each orbital plane. The relative positions of the satellites in the constellation change over time, but the highly uniform nature of this network ensures identical configurations recur often during one orbital period. Figure 2.1 presents a delta network with m orbital planes, each with n satellites.
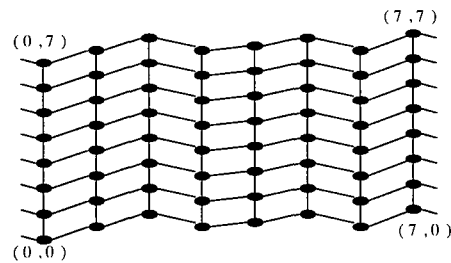
Figure 2.1. The Walker Delta Network [Wan93].

2.3.3 Multiple Access Techniques With potentially large numbers of users competing for communications resources, the multiple access scheme implemented by the LEO system designer could prove crucial. In LEO systems, four access schemes are predominantly used: Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), and Space Division Multiple Access (SDMA).

FDMA is one of the oldest multiple access techniques used in satellite communications. Under this scheme, each earth station transmits one or several carriers, at different center frequencies, to the center transponder. A frequency band and associated guard band, used between carrier bands to avoid frequency overlap between adjacent carriers, are assigned to each carrier. Earth-based receivers are tuned to listen

for their particular carrier band, which enables them to selectively receive messages intended for them.

A TDMA system divides a single carrier, among earth stations, for transmission to a satellite's transponder on a time division basis. A time burst is allocated to each earth station, during which time it has access to the complete bandwidth of the transponder. Since all transmission bursts are multiplexed in time, the earth stations must be correctly synchronized so the individual earth station bursts do not overlap. Going the other way, earth stations receive the entire burst from the satellite transponder and extract portions allocated during their time burst.

Because of the required synchronization of earth station transmissions, TDMA message overhead tends to be higher than in other multiple access techniques. Each TDMA frame contains reference bursts, traffic bursts, and guard times between the types of bursts. The reference bursts are used by a TDMA system, to provide timing synchronization for numerous earth stations using the same satellite transponder. Traffic bursts contain the information broadcast by the earth station, while the guard times are used to ensure the earth station bursts do not overlap at the satellite transponder.

CDMA systems, known also as spread spectrum multiple access (SSMA), employ a digital spread spectrum technique, which allows transmission from several users to overlap synchronously in time and frequency, using complex binary codes. These codes must correlate both at the transmitter and the receiver, which keeps the two ends in synchronization. This technique allows for a relaxation of both the accuracy of frequency and time intervals between users.

SDMA, known also as multiple beam frequency reuse, allows for reuse of the same frequency bands. Spot beam antennas separate the various radio signals by aiming them in different directions. This implementation allows simultaneous access of a satellite from two different regions of the earth, even though the frequency of both uplink signals was identical.

2.4 The Globalstar, Iridium and Teledesic systems

As previously discussed, many mobile satellite communications systems have been proposed. Globalstar, Iridium, and Teledesic were selected from these many options for examination, for several reasons. Economic and technological viability rank high on this list. Without a sound technological approach and the required capital to bring the design off the drawing board, this study would be an academic exercise at best. Regardless of how good a system sounds, it does our military no good if it is non-functional. In addition, while the Globalstar, Iridium and Teledesic systems share some commonality, their widely divergent design approach provides an interesting contrast for analysis. In the following discussion, all three systems are described and analyzed, comparing and contrasting when applicable.

2.4.1 The Globalstar System The Globalstar system is being funded and developed by a large group of companies, known as the Globalstar Partnerships [URL1]. This group includes Loral, QUALCOMM, SS/L, AirTouch, Alcatel, and numerous others. Its origins began in the 1980s, when QUALCOMM began work on a worldwide satellite-based telecommunications network and SS/L was investigating the technical feasibility of a LEO constellation designed to deliver voice, position location, and messaging to luxury automobiles.

The Globalstar system is designed to be a LEO satellite-based digital telecommunications system which will offer high quality telephony, data transmission, paging, facsimile, and position location services [URL1]. The designers expect user handsets to cost approximately $750, telephone booths to cost up to $2500, and service rates of up to $0.53 per minute.

A true bent-pipe system, Globalstar relies heavily on gateways. Designers estimate for full global land-based coverage, approximately 210 gateways will be required [URL1]. These gateways are expected to cost from $2 to $5 million per unit and would be built by local service providers in the region being served. The brains of the system,

the call processing and switching information, is located on the ground. Each gateway will contain up to four tracking antennas and radio frequency front ends, designed to track satellites in their view. The gateway equipment is designed to be easily integrated with existing service provider equipment [Wie93].

The satellites in the constellation serve as repeaters, relaying signals received directly back to Earth [CoM93]. As previously mentioned, this greatly reduces the satellite weight with respect to other systems, such as Iridium. Each satellite is expected to have an average mission life of 7.5 years.

Similar to most proposed LEO systems, Globalstar uses a CDMA multi-access implementation, with some elements of FDMA as well [CoM93]. The uplink and downlink bandwidth of 16.5 MHz for mobile users is partitioned into thirteen 1.25 MHz CDMA channels, each of which is centered orthogonally on different frequencies.

2.4.2 The Iridium System The Iridium system, has been proposed by Motorola and a group of partners. Motorola, with an extensive wireless communication heritage, designed Iridium "from the handset up" [Sco95], and selected its principle partners after it determined they shared Motorola's vision.

Originally, Iridium was designed to have a constellation of 77 satellites, hence its name. With further refinement however, the constellation number was reduced to 66. The constellation will consist of 6 orbital planes, with 11 satellites and one spare in each plane. Each satellite will orbit 420 nautical miles above the earth.

Iridium's designers envision a 'global, digitally switched network in space' [Sco95], with users anywhere on Earth, including over water, being able to communicate regardless of the existing infrastructure. Although prices appear to be dropping, Motorola originally estimated the price of their handsets at $2500 to $3000. The service cost per minute has been estimated at $3 per minute [WuM94].

The Iridium satellite is the key player in this network because it is the first commercial satellite system to implement ISL capability [CoM93]. Each satellite will

have four crosslinks; one forward within a plane, one backward within a plane, and two cross-plane links. In addition, each satellite will have on-board digital processing systems, which receive and forward calls from one satellite to another, before returning the traffic to the end user on earth. The extra equipment on-board the satellite increases each satellite's weight to three times that of Globalstar's satellites. Motorola estimates the average satellite life span will average between 5 and 7 years [Sco95].

Motorola plans to utilize a combination of TDMA and FDMA [CoM93] for multi-access in the Iridium system. Iridium utilizes TDMA in that a 12 frequency re-use scheme has been proposed over the 10.5 MHz bandwidth which crosses cell boundaries within a satellite's coverage area, as well as crossing boundaries of coverage areas of neighboring satellites. FDMA is utilized in that a 90 ms TDMA frame accommodates four 50 Kbps user access per frame.

2.4.3 The Teledesic System The Teledesic system, previously known as the Calling Network, is an extremely ambitious project, aiming to provide basic and enhanced communication services to rural areas of high-income countries and basic communication services to the general populations of developing nations. The scope of this project can be seen in the high data rates Teledesic will be designed to support, varying from a minimum of 16 Kbps to 2 Mbps [TuP93].

Once completely operational, Teledesic planners envision a constellation of 924 satellites, with 840 active satellites and 84 spares. The satellites will be arranged in 21 orbital planes, each containing 40 active satellites and 4 spares. Each satellite will orbit 700 kilometers above the earth. While the satellites are designed to last 10 years, Teledesic estimates 30% will fail prior to the 10 year point [TuP93]

ISLs are a key element of Teledesic. Each satellite will have eight crosslinks [TuP93] : two crosslinks with the satellites in front and back in the same orbital plane, and with one satellite in both of the two adjacent planes on each side. In conjunction with the ISLs, each satellite will have on-board digital processing systems, which receive and

forward calls from one satellite to another, before returning the traffic to the end user on earth.

Teledesic designers plan to use a variety of multi-access schemes. In the Teledesic concept of operations, supercells (terrestrial coverage areas) are subdivided into nine cells, all of which are assigned one of nine equal time slices. All cells are scanned in a cyclical fashion. When a particular cell is scanned, it has full access to the frequency allocation. This, obviously, consists of TDMA between cells in a supercell and SDMA between simultaneously scanned cells in adjacent supercells. During each cell's time slice, FDMA is used for the uplink and asynchronous TDMA is used for the downlink [URL2].

2.4.4 Comparison of the Globalstar, Teledesic and Iridium systems As mentioned previously, while the systems have similarities, the design approach is widely divergent. Two key areas define the respective systems. The first is the reliance on bent-pipe satellite repeaters versus the usage of ISLs. The second is the multi-access technique that is used.

The usage of ISLs, versus ground-based traffic management, unquestionably complicates the system design. While technically feasible, a large margin for error is present. Among other things, the on-board software must be able to keep track of the ever changing nearby satellite orientation; including the satellites' state of health information, be able to efficiently forward user traffic onward to its eventual destination, and correctly manipulate the on-board antennas to communicate with both nearby satellites and users on the ground. Motorola estimates 15 million lines of code could be required to implement Iridium's space and ground segment [Sco95]. Although no published information was found for Teledesic regarding complexity, intuitively it will be even more difficult to implement than Iridium. The bent-pipe solution is not as complex. The planned satellite repeater technology for Globalstar satellites has been in production for over 20 years [Wie93].

The multi-access technique, while rather straightforward for Globalstar, which uses CDMA and FDMA, may prove challenging for Iridium, which uses TDMA and FDMA and Teledesic, which uses TDMA, FDMA, SDMA, and asynchronous TDMA. The multi-access design planned for Globalstar exactly models the QUALCOMM design currently used in their terrestrial applications [CoM93]. Iridium's design will, on the other hand, be quite complicated. To implement their multi-access technique, the Iridium system will need to do three things [CoM93]. First, it must coordinate cell utilization, to account for cell overlap, as the satellites move to higher latitudes in their orbits. Secondly, cell frequency management must be dynamically coordinated both within the satellite's antenna beams and across satellite boundaries of neighboring satellites. Finally, accurate time synchronization must be provided to support the TDMA framing structure. Teledesic faces similar challenges.

The distinctions between the two systems are quite clear. Globalstar, while less ambitious a project, relies primarily on proven technology. Iridium, by contrast, pushes the state of the art, and faces considerable risks in reaching operational status. Teledesic, called "Pie in the Sky" by skeptics [URL3], is similar to Iridium in many ways, and faces similar challenges. Time will tell which strategy is most successful.

2.5 Traffic Routing

The message routing scheme, used in a LEO network, is crucial. Users of these networks will expect quick, reliable service, as well a system which is survivable, in both a civilian and military environment. Because of the relatively large numbers of nodes in a LEO network, data transmission to the next intermediate node must be handled efficiently. Generally speaking, procedures for routing message traffic fall into two categories, static and dynamic (or adaptive). As can be expected, static routing procedures do not work well for the LEO environment because of its dynamic nature.

Therefore, the routing procedures discussed here are dynamic. This section presents some of the routing techniques which have been researched.

A distributed adaptive routing algorithm for the Strategic Defense Initiative (SDI) was proposed by Cain et al. [CaA87]. This algorithm was designed to not only minimize delay and maximize throughput, but also be capable of load balancing and be adaptable to changes in connectivity. After determining the shortest paths from a source node and all of the source node's neighbors, the algorithm calculates a set of feasible paths. Briefly, feasible paths are only those paths which travel through neighboring nodes, which lie closer the destination node than the source node. A load balancing heuristic was also used. The developed algorithm was evaluated by simulating a SDI network containing 36 LEO and 12 GEO satellites and comparing the results between the newly developed algorithm and a standard adaptive shortest path algorithm. The new algorithm was found to perform much better than the standard algorithm, with the new algorithm providing the same path delay with link capacities about one-half the values required by the standard algorithm.

Another approach has been advanced by Chakraborty [Cha89], who proposed using dynamic routing by the utilization of distributed processing. His approach has every node in the network calculating the cost of each route, using perceived congestion and internodal distances as cost criteria, with the least-cost route being selected. This method, in effect, decentralizes the decision-making to each satellite. From his research, Chakraborty found that while at either low or high node capacity utilization, dynamic routing was not a good cost saving approach, it did deserve consideration if the utilization was between 40 and 50 percent.

In a system study of Clare et al. [ClW87], the authors assume every satellite knows the connectivity of the network. This assumption forces each node's routing tables to be complex to be able to keep up with the dynamic network changes. In this study, the authors also investigate the effects of random versus deterministic routing given by each

nodes knowledge of network connectivity.   Deterministic routing was found to be superior to random routing.

A study conducted by Wang [Wan95] takes a different approach for message transmission in a network. He proposed utilization of virtual cut-through switching, in place of the traditional packet switching technique, usually practiced in data networks. The virtual cut-through switching avoids the main drawback of packet switching, which involves the packet not being transmitted out to the next node until it is completely received by the current node. It forwards packets to the next node in the route without buffering, if the satellite has established a path to the next node. Important to note, cut-through routing is available only in situations where one or more intermediate satellite nodes exist between the receiving and ending nodes, and the queue of the receiving node must be empty. Implementation of cut-through routing would require a routing chip to be installed on each satellite.

Wang [Wan95] found, via numerical studies, that for low and medium traffic densities, virtual cut-through routing can significantly reduce delays. As can be expected, six ISLs reduced transmission delays more than four ISLs did. In addition, the higher the probability of cut-through, the shorter the delay as well. The author claims that his results will enable LEO system designers to provide guarantees that messages traveling a specified number of hops will be deliverable under a delay bound. Obviously, application of this technique is currently restricted to the Iridium system, since it is the only proposed system utilizing ISLs.

To maintain the optimal routing for message traffic, shortest path calculations must be frequently executed. These calculations take into account a link cost factor, which is dependent on the network in question. One typical shortest-path algorithm, attributed to Dijkstra, determines the shortest paths from a source node to all of the other nodes in the network. Another popular shortest-path algorithm, known as Bellman-Ford, determines the shortest paths from all sources to a single destination. Many variations of these

routing algorithms exist, which can be tailored to specific applications. Descriptions of these can be found in a standard computer science text.

2.6 Network Survivability

Survivability, the focus of this effort, will be a large factor in determining the success or failure of the proposed LEO systems. This issue is particularly critical in the military environment, where satellites could be disabled by physical threats such as anti-satellite weapons and high-power lasers. Survivability presents system design challenges, because on one hand low cost is desired, implying highly efficient use of resources. However, survivability implies the utilization of excess capacity and resources to mitigate any possible threats. A considerable amount of research has focused primarily on design issues. This section discusses the survivability framework developed by the American National Standards Institute (ANSI) and presents some of the research approaches taken to respond to network failures.

The ANSI standard for survivability was designed to provide a consistent foundation enabling comparison of network survivability techniques. Loosely based on the layered approach, used in the Open Systems Interconnection reference model, the survivability standard model is partitioned into four layers; 1) Service, 2) Logical, 3) System, and 4) Facility [SeF93]. The service layer provides information transfer and network management services. This layer supports survivability by controlling network access, detecting and adjusting to changes in configuration, and monitoring and managing the network. The logical layer manages the network reconfiguration and rerouting of data, as well as managing the link capacity utilization. Survivability is implemented here via dynamic routing and capacity allocation algorithms. The system layer supplies or accepts signals over a link. Reliable end-to-end connectivity is the responsibility of this layer. The facility layer is responsible for providing a secure operating environment.

Survivability is enhanced at this layer by protection of network assets through use of secure facilities, redundancy, and fault detection systems.

In a study conducted by Gross and Ziemer [GrZ89], two common routing algorithms were compared, with the intent of demonstrating their adaptive efficiency. Both were applied to a simple network and a satellite network with the purpose of analyzing network recovery after a node or link failure in terms of number of algorithm iterations performed and number of control messages generated. The first algorithm used was a distributed version of the Ford and Fulkerson algorithm, which guarantees convergence to a shortest path, but is susceptible to looping. The second algorithm, designed by Merlin and Segall [MeS79], also guarantees convergence to a shortest path, but eliminates looping. When a link failure was simulated on the simple four node network, the Merlin and Segall algorithm required the same number of iterations as the Ford and Fulkerson algorithm, but required less than 1/3 the number of control messages. However, when a link failure was simulated on a typical satellite network, consisting of 18 LEO and 6 GEO satellites, the Merlin and Segall algorithm required nearly 7 times as many iterations and 15 times as many control messages as the Ford and Fulkerson algorithm.

Also noting the poor convergence speed of the Merlin and Segall algorithm, Garcia-Luna-Aceves et al [GaC89] proposed an extension of the distributed Bellman-Ford algorithm to better address network restoration. This extension was designed to overcome the primary disadvantages of the standard Bellman-Ford algorithm: the susceptibility to looping and failure of the algorithm to converge when the network becomes disconnected. The authors expand this protocol by requiring the algorithm to maintain only loop-free paths and correspondingly conduct the shortest path search only from the restricted set of loop-free paths. By doing this, they claim to eliminate the disadvantages of the standard Bellman-Ford algorithm, while only slightly increasing network computational overhead.

More recently, two competing network restoration algorithms have been proposed. The first, known as Max Flow, was first proposed by Goldberg and Tarjan [GoT88], and refined into a distributed algorithm by Baker [Bak91]. The Max Flow algorithm is designed to obtain the maximum rerouting capacity based on a maximum flow criterion. The second algorithm, known as k-shortest paths (KSP) and proposed by Grover et al. [GrV91], finds the set of k-successively shortest link disjoint paths in a network.

In a study conducted by Dunn et al. [DuG94], the authors claim the distributed KSP algorithm is preferable to the distributed Max Flow algorithm. According to the study, the KSP algorithm has several inherent advantages and one primary disadvantage. First, KSP is less computationally complex than Max Flow, running in $O(n \log n)$ time versus $O(n^3)$. In addition, KSP is easier to implement since the state of technology is currently more advanced (for the distributed KSP than for the distributed Max Flow). The primary problem with KSP is that it does not always find all the paths found by the Max Flow algorithm. This study found that KSP has a restorable capacity of greater than 99.9% of that of Max Flow, when analyzed over 15 network models. In summary, the authors recommend the KSP algorithm since the slightly smaller restorable capacity is greatly outweighed by the speed and complexity deficiencies of the Max Flow algorithm.

Research conducted by Tipper et al. [TiH94] investigated the topic of traffic congestion after a network failure. After noting that traditional congestion control schemes, either end-to-end windowing schemes or rate based policing mechanisms, may not prevent congestion, the researchers investigated the congestion which occurs in a network. Focusing on the congestion resulting at the primary node, or the source node for the traffic over the failed link, an analytical model was developed to predict queue lengths and packet loss probabilities. The output of the model matched up closely, when compared against a discrete event simulation ran against a generic packet switched, virtual circuit, wide area network.

Although the Tipper et al. [TiH94] study provided some interesting results, additional work is required to draw lasting conclusions. While the authors recommend using their results to implement congestion control after a failure, thereby preventing or reducing congestion at the primary nodes, they recommend further research be conducted to investigate the impacts of congestion on neighboring nodes. In addition, the authors only consider the loss of a link, not a node. Obviously, in a highly connected network, the impact of losing a node far outweighs the impact of losing a link. As concluded by the Tipper et al. study, the subject of congestion needs further investigation.

Another promising routing algorithm, known as DARTING [TsM95], has been recently proposed by Tsai and Ma. Instead of relying on flooding the network with routing updates, or periodically exchanging messages between nodes, DARTING relates its topology updates to the data traffic rates that are being experienced. DARTING uses two basic mechanisms, successor update and predecessor update. The successor update mechanism occurs by requiring every predecessor node to embed its local network topology changes within the data message passing through the node. The predecessor update occurs by forcing every successor node to create a control message when it detects a difference in topology views between itself and its immediate predecessor. The control message is then sent along the same path just traversed by the data traffic. In short, the DARTING algorithm focuses primarily on defeating message loops, rather than preventing them. Simulation data obtained from running the DARTING algorithm versus a conventional flooding algorithm on a 64 node dynamic-topology network clearly indicated the superiority of the DARTING algorithm, at least in terms of end-to-end message delay, minimum buffer sizes, and link utilization.

2.7 Summary

This chapter first discussed why LEO satellites are the best platform for a mobile communications system, then presented key aspects of a general LEO satellite system,

some specific information pertaining to the two satellite constellations to be studied, and various approaches for routing message traffic. The discussion, within this chapter, was bounded to specifically address the problem at hand.

Section 2.2 compared and contrasted GEO, HEO, and LEO systems, emphasizing that LEO systems were best suited for the mobile communications environment. In Section 2.3, several key characteristics of LEO systems were presented, which shape system design, and will impact this study greatly. Section 2.4 described the Globalstar, Teledesic and Iridium systems, in light of the material covered in Section 2.3, with some comparisons and contrasts drawn between the two design approaches. Various message routing techniques were discussed in Section 2.5 while network survivability was covered in section 2.6.

A large amount of literature has been written on LEO systems, some of which was discussed in this chapter. However, due to the relative newness of this area, the literature is spread over the many problem regions which need to be addressed, to design and field LEO systems, most of which is out of the scope of this effort. While some of the available literature discusses LEO network survivability, the key issue of this endeavor, the majority of the work focuses on the survivability framework, rather than the wide-ranging problems associated with failure recovery. This fact further points out the need to conduct this study.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The complexity of the LEO environment is easily seen from the literature review, that was conducted in Chapter 2. This chapter will focus on the modeling of the Iridium satellite network, emphasizing the constellation survivability.

Section 3.2 describes the research problem, scoping details, and the expected research results. In Section 3.3, a discussion of performance metrics is conducted. The rational for the use of simulation, versus other modeling techniques, is presented in Section 3.4. Sections 3.5, 3.6, and 3.7, discuss the operating assumptions, design parameters, and design factors, which underlie this research. The actual simulation models for the proposed constellations are presented in Section 3.8. This section focuses on the operations and functions of the primary components of the simulation models. In addition, implementation of survivability constraints are also discussed. In Section 3.9 and 3.10, the constellation model's validation and verification are detailed. The testing rational and methodology are defined in Section 3.11. Section 3.12 concludes this chapter, by summarizing the previously discussed information.

3.2 Problem Overview

After restating the research problem, this section discusses the scoping issues faced in this effort, presenting justification for assumptions made. In addition, a geographic frame of reference for this work is given. The expected results of this effort conclude this section.

3.2.1 Problem Definition Currently, no research results have been found in published literature, evaluating the robustness of the proposed LEO networks in a faulting environment.

3.2.2 Problem Statement This research is focused on determining the minimum acceptable constellations, with respect to service degradation, for the proposed Iridium LEO satellite network.

3.2.3 Scope The research effort must be properly scoped to allow sufficient time to thoroughly investigate the problem. To that end, this research is constrained by three factors: 1) satellite constellation selection, 2) the evaluation of network performance via the use of computer simulation, and 3) simplifying assumptions made.

Initially, the intent of this research was to evaluate the survivability aspects of both the Iridium and Teledesic networks. During initial tests using the Satcom_router, it quickly became apparent the hardware resources available were not able to handle either the memory requirements or processing speed needed to generate the necessary data. Therefore, the Teledesic network evaluation was dropped from this effort.

The proposed Iridium satellite network was selected for study for two reasons. First, both networks rely on Intersatellite Links (ISLs) to handle user traffic. During the review of the literature for this research effort, both the traditional method of utilizing satellites for communications, via bent-pipe transmission and the newer method of ISL utilization were investigated. Further investigation of Globalstar, a bent-pipe system, from this research's survivability focus, revealed the need for greater understanding of the terrestrial network. Globalstar will need to rely on the terrestrial network in a degraded operating environment. Since the focus of this effort is survivability of space-based communications networks, not terrestrial networks, Globalstar was eliminated from the group of systems being investigated. Second, both networks appear to be commercially viable. The Iridium project is currently projected to be fully operational by mid-1998.

Due to the nature of this problem, and the time allotted to conduct the research, computer simulation will be used to evaluate the performance of the various satellite networks. This decision, obviously, drives the methodology used to research the problem. Further explanation and justification of this decision is discussed in Section 3.4

Two broad, simplifying assumptions were made to clarify network performance and allow more extensive investigation of the problem. First, all transmissions are assumed to be error-free. While errors do occur, their frequency is such that they can be ignored for the purposes of this investigation. Second, mobile users are ignored in all simulation experiments. This assumption was valid since mobile users can only travel 10 to 15 miles in the evaluation period. Hence, for this effort, mobile users can be successfully modeled as fixed gateways.

3.2.4 Geographic Area of Interest The area under investigation for this performance study, is bounded by the geographic locations with the latitude coordinates of 15° - 40° N and longitude coordinates of 80°W - 50°E. These coordinates are bounded on the western side by the east coast of the United States, and on the eastern side by the Middle Eastern country of Iraq. In addition, these coordinates encompass most of the Atlantic Ocean and the Mediterranean Sea. This area of interest was selected to model real-world traffic during a wartime scenario such as the Gulf War, as well as provide a wide area to analyze the survivability aspects of the subject constellations.

3.2.5 Communications Systems Architecture The communications network consists of two parts: the satellite constellation under evaluation, and the earth station gateways. The satellite constellation is the Iridium satellite network. The network contains two earth station gateways: the first is located on the east coast of the United States, at Washington DC (approximately 77°W and 39°N) and the second in Jerusalem, Israel (approximately 32°N and 47°E). The number of gateways was restricted, to two, due to the large amount of data analysis required.

3.2.6 Data Traffic All traffic is transmitted in a packet-switched, real-time environment. Real-time implies all transmitted information is received no later than 400 milliseconds after it was transmitted.

3.2.7 Expected Results Two outcomes are expected upon the conclusion of this research. First, constellation degradation is anticipated to be rather gradual, not abrupt as would be expected in a GEO satellite. Second, the various routing algorithms are expected to significantly impact the various network's survivability.

3.3 Performance metrics

Many different performance metrics can be chosen for analysis in a performance study. In this research, three specific metrics were chosen to derive the appropriate conclusions: 1) network delay, 2) hop count, and 3) number of dropped data packets. Network delay is the time required by a packet to traverse the network from its source to its destination node. Delays longer than the real-time threshold of 400 milliseconds will be considered unacceptable performance. The hop count measures the number of hops the packet must traverse enroute to its destination node. The number of dropped packets reflects the level of network congestion. This metric will depend upon both the state of the network (level of degradation), and the loading level being applied. Any ratio of dropped data packets versus the total number of data packet which is greater than a 1 percent Grade of Service criteria will be considered unacceptable performance.

3.4 Approach

Of the three basic methods used for performance evaluation [Jai91], 1) analytical modeling, 2) simulation, and 3) measurement, simulation will be used to analyze this research problem. Simulation models for the Iridium constellation was constructed using the integrated commercial simulation packages Bones Designer and SatLab [Cad95]. The Designer package models the communications portion of the system, while SatLab models the satellite and earth station positioning.

Simulation was chosen for this research for two reasons. First, the other available performance measurement techniques, measurement and analytical modeling, were either impossible or infeasible. Measurement analysis of LEO networks is currently impossible, since no operational LEO networks exist at present. Analytical modeling, based on queuing theory, is infeasible. While analytical models may be possible for a given network node (if certain assumptions are made), no analytical solution is known for multiple node networks without making significant Markovian assumptions, such as a Poisson arrivals. This problem forces the analyst to make many simplifying assumptions to develop approximate analytical models, thus negatively impacting the accuracy of the results. Second, properly constructed simulations allow detailed, systematic analysis of computer networks. Since simulations do not require details to be abstracted out, the results are, generally, more often closer to reality than those obtained from analytical models [Jai91].

3.5 Operating Assumptions

To accurately model the Iridium network, many system operating assumptions have to be made. These assumptions are similar to those published decisions made by Motorola in designing its proposed systems [Mot90, LeM93].

3.5.1 Satellite Coverage The satellite networks examined are designed to provide whole-earth coverage. For the purposes of this research, the geographic area of interest lies in the Northern Hemisphere, as specified in Sections 3.2.4 and 3.2.5.

3.5.2 Period of Evaluation of the Network The evaluation period of the network is bounded from the period of time the network configuration repeats. For this effort, the evaluation period was restricted to 15 minutes for two primary reasons. First, sensitivity analysis showed that 3 minute positional updates provided the optimal tradeoff between more frequent updates and their impacts on the network, in terms of network overhead. Second, the most degraded constellation evaluated in this effort, Iridium with 36 failed

satellites, maintained gateway visibility (with respect to a nearby Iridium satellite) for 16 minutes. Therefore, based on both considerations just mentioned, a 15 minute evaluation period was chosen.

3.5.3 Simulation Epoch One simulation epoch is used for all the simulations in this research. This epoch, used for the initial satellite location determinations, is randomly chosen to be 10:30 am on July 30, 1992.

3.5.4 Traffic Distribution Specific data traffic generation distributions are used for gateway transmissions. Since LEO systems currently do not exist, there is a lack of relevant data regarding data traffic characteristics. Thus, the traffic distribution, used in this model, is based on discussion with faculty and engineering intuition. The data traffic is modeled, using a satellite-based, packet-switched data communications system environment.

3.5.4.1 Source Generation Rates The generation of packets by the system gateways is controlled by a bursty process. Since actual traffic distributions are unknown, the burst mean burst size was set to 50 packets, providing one possible system workload characterization.

Traditionally, Poisson traffic models have been used to model traffic generation for communications networks. However, recent work [WiW94, PaF94] has shown the failure of the Poisson traffic characterization for most types of traffic. This failure has led to a significant under estimation of buffer sizes required as well as end to end packet delay performance. Based on these studies, a bursty traffic model was selected.

3.5.4.2 Source Address Distribution Since this research is investigating network performance between two gateways, the source addresses are equally divided between the two.

3.5.4.3 Destination Address Distribution Similarly to the previous section, the destination addresses are equally divided between the two gateways. Of course, this distribution assumes a gateway can not send messages to itself.

31

3.5.5. Satellite Link Availability For the purposes of this research, the satellite link will always be available unless a satellite failure occurs.

3.5.6. Message Routing Network message routing is accomplished via a variety of routing algorithms. Several algorithms were used during this evaluation: 1) Dijkstra, 2) Extended Bellman-Ford (exBF) and 3) Darting. Each is described in detail below.

3.5.6.1 Dijkstra The Dijkstra algorithm is a commonly used, least-cost, routing algorithm. The algorithm is forward searching, in that it finds the least cost path from the source node to all other nodes in the network. In the implementation used for this effort, the distance between the nodes is used as the least cost metric.

The Dijkstra implementation used in this research must be considered as best-case. The *Satcom_router* assumes every node in the network knows the status and visibility of every other node in the network immediately after a positioning update from SatLab (see Section 3.8.1 for SatLab positioning discussion) occurs. This assumption has a two-fold impact on network performance. First, the routing paths selected are globally optimal since they are not based on only localized information. Second, the network is not required to processing local routing update traffic as adjacent nodes inform each other of network status. This lack of overhead should be reflected in both improved delay performance and reduced packet rejection.

3.5.6.2 Extended Bellman-Ford As discussed in the literature review, this algorithm is an extension to the standard distributed Bellman-Ford. The goal of this enhancement is to eliminate the original algorithm's susceptibility to looping and failure to converge when the network becomes disconnected

Unlike the Dijkstra algorithm, all nodes in the network do not immediately know the status of all other network nodes, once a SatLab update has been received. This scenario requires additional overhead, in the form of network update packets, to be sent throughout the network to update neighboring nodes of local network status. The

32

resultant extra network traffic restricts network capacity available to standard data packets.

3.5.6.3 Darting The DARTING algorithm was proposed by Tsai and Ma [TsM95] as a favorable routing alternative to periodically flooding the network with update packets. Rather than trying to prevent loops from occurring, this algorithm focuses on breaking loops when they arise.

Similar to the exBF algorithm, DARTING will not know the global status of all network nodes immediately after a SatLab update, thereby resulting in some network overhead to properly route data packets. For DARTING, the successor update information is embedded in the standard data packets as they traverse the network, while the predecessor update function requires that additional update packets be introduced into the network to inform neighbors of the latest network status.

3.5.7. Multiple Access Technique TDMA is used for multiple access. Based on a 60 millisecond time slot, a 30 millisecond delay was assumed as the average delay the gateway would experience when transmitting data.

3.5.8. Minimum Look Angles The gateway minimum look angle is 10° relative to the nearby horizon. This value was chosen to mitigate the effects of terrain and vegetation on the propagation path.

3.5.9. Satellite Crosslink Communications Every satellite, in the subject constellations, utilizes intersatellite links (ISLs). Iridium has one ISL with its nearest neighboring satellites forward and aft in the same plane, for a total of two intraplanar ISLs. Also, Iridium satellites have one ISL, with the satellite in each of the two adjacent orbital planes. Thus, as described in Chapter 2, each Iridium satellite has four ISLs.

3.5.10. Bit Error Rate (BER) Since this research was conducted in an error-free transmission environment, the BER for the models used is 0.

3.5.11. Control of Satellite Capacity The capacity control is distributed throughout the network. The satellites in the Iridium network are designed to handle the digital processing and routing required in the distributed scheme.

3.5.12 TDMA Frame Length The TDMA frame length, 60 milliseconds, is derived from the Iridium system [Mot90, LeM93]. This value is representative of LEO class systems.

3.5.13 Packet Lengths For this research, the packet lengths are fixed at 1024 bits. This figure is representative of proposed LEO systems.

3.5.14 System Queues The network system queues are all first-in-first-out (FIFO). The satellite input queues are a fixed size, set at 500 packets in length. In addition, each individual queue's memory space is 1 Mb, which is representative of proposed systems [Mot90].

3.5.15 Packet Retransmission Packets which are blocked, because of completely filled input queues at the satellites, shall be dropped without retransmission. This approach is taken to eliminate any possible impacts a specific retransmission protocol might induce. The number of dropped packets will be used to evaluate blocking probabilities.

3.5.16 Regenerative Links Both systems under analysis have regenerative capabilities at their satellites. This regenerative capability means, after the incoming signal is demodulated, the data is detected and processed before being remodulated prior to transmission. This assumption is consistent with the digital nature of the satellite for systems utilizing TDMA.

3.5.17 Satellite Processing Time The time required to decode an incoming packet and deduce the next location, as it traverses the network, is defined as satellite processing time. For this effort, this delay is assumed to be represented by a normal distribution with a mean equal to 100 microseconds and a variance of 5 microseconds [ClJ89].

34

3.6 Presentation of Design Parameters

This section defines the system data rates that are required to successfully model a LEO communications network. This network is comprised of three areas: terrestrial, orbit, and links. The terrestrial portion of the network is comprised of earth based transmitters and receivers, while the orbit portion is made up of the orbiting satellites. The link portion of the system is made up by the uplink, downlink, and crosslink channels. The following discussion defines the data rates associated with each of the three parts.

The values were selected to insure the simulation model is an accurate model of the systems under investigation. The chosen values were derived from those proposed by Motorola [Mot90] for their system satellite in the Iridium constellation.

3.6.1 Terrestrial Data Rate The data rate for the earth gateways uplink and downlink is 12.5 Mbps.

3.6.2 Satellite Data Rate The data rate for the Iridium satellite is 25 Mbps.

3.7 Presentation of Design Factors

As alluded to previously, the proposed LEO networks, under investigation, rely on state of the art technology. This fact, obviously, implies vast areas of study are needed in this new environment. However, since the focus of this effort is the survivability of the Iridium network, it is necessary to concentrate on several of the areas directly impacting constellation survivability and hold the remainder of the factors constant. Three factors influencing constellation survivability were identified as keys to this research effort: 1) Varying degrees of satellite failure in the network, 2) Loading levels, and 3) Impacts of utilizing various dynamic routing algorithms. These factors are combined in a systematic manner to ensure meaningful results. Test procedures, in Section 3.11, illustrate the ways the three varying factors are integrated.

3.7.1 Satellite Failure The amount of satellite failure, and its resultant impact on the associated network, are crucial to this effort. Such impacts are clear in a GEO

environment, where the loss of one satellite can be catastrophic in overall network performance. In the Iridium network, however, the constellation is designed to mitigate satellite loss. The objective is gradual network degradation in response to the loss of one or more satellites. This research will systematically "kill" satellites to observe the network response in terms of the performance metrics defined in Section 3.3.

3.7.2 Network Loading The network loading is generated by the various gateways, using the network. Different loading levels on the network are generated by varying the number of user requests, from the gateways within the geographic area, specified in Section 3.2.4. It was the original intent of this investigation to examine the variations in the loading levels by using three loading scenarios, set at 10, 50, and 90 percent respectively. Unfortunately, the computer resources available were unable to support these levels due to the extensive computer time and memory required. (Refer to Section 4.2 for a detailed discussion of computer time requirements.) Therefore, three lower loading levels, 1, 6 and 11 percent, were used. The performance metrics collected from these scenarios are used in the overall survivability analysis.

At present, the workload characterizations of data traffic distributions of the gateways are not well understood. Recent studies, such as those referenced in Section 3.5.4.1, indicate that most data traffic is of a bursty nature. Traditionally of course, network performance studies utilize Poisson processes to model data traffic. For this effort, a Poisson process was compared against a bursty generator, using a burst size of 50 packets and a 11 percent loading level.

The comparison between the Poisson and bursty traffic generators bore out the results found in the literature. The delays generated by the bursty generator were typically 50 percent higher than those generated by the Poisson generator. In addition, the input queue occupancies were generally 3 to 4 times as large with the bursty generator versus the Poisson generator. Based on this data, and consultation with faculty, a bursty generator with burst size of 50 packets was used for this effort.

36

3.7.3 Routing Algorithms The ability of the satellite constellation to quickly and efficiently move data traffic through the network from source to destination, particularly through various stages of network degradation, is crucial. Dynamic routing algorithms, which are able to adapt to the quickly changing LEO environment, are a must. These algorithms must be able to handle both the routine changes a LEO constellation undergoes, as well as quickly adapt traffic to satellite failures, while not increasing network overhead to unmanageable levels. This problem is not trivial and is itself currently the focus of separate research efforts.

For this research, implementations of several dynamic routing algorithms were used to crystallize the overall survivability picture of the Iridium network. The algorithms used were C programming language coded models from a parallel research effort [Jan96] in the LEO environment. These algorithms were implemented with the underlying complete and degraded Iridium networks and loading levels. As before, the performance metrics collected provided key survivability insights.

3.8 LEO Network Simulation

The simulation of the Iridium network is detailed in this section. All the simulations are performed using the Designer and SatLab commercial simulation packages [Cad95] and supplementary C language primitive subroutines. Simulation of these constellations required that both Designer and SatLab, be used simultaneously. Designer, models the communications portion of the network, while SatLab handles the positioning functions for the gateways and satellites. The remainder of this section discusses the simulation model design hierarchy.

3.8.1 The SatLab Model The positioning information associated with the gateways and satellites is performed by SatLab, as noted previously. In general, the system modeler must provide three types of system information: the constellation's orbital parameters, the gateway's positioning information, and the simulation epoch.

The network satellite's orbital parameters consist of the following: 1) satellite identification number (externally referenced as a number-letter pair to represent the orbital plane and satellite within the plane), 2) the orbital inclination, measured with respect to the equator, 3) the satellite mean anomaly, 4) orbit eccentricity, 5) argument of perigee, and 6) the argument of right ascending node. These parameters when, combined, form one entry in the constellation definition.

SatLab's earth station data defines the terrestrial gateways. The gateways are defined by their latitude, longitude, and altitude positions relative to the Prime Meridian. Therefore, western hemisphere latitudes and southern hemisphere longitudes are assigned negative values.

The epoch is used to define the starting date and time of the simulation. This information is used to derive initial positions for all the nodes, both satellites and gateways, in the network. The communications portion of the simulation then makes use of this information. In addition, information such as the total number of nodes, line-of-sight distances between two nodes, and relative velocities are also available to the communications model.

3.8.2 The Satellite Communications Model The satellite communications model is implemented and simulated using the Designer modeling tool, supplemented with C code subroutines as required. Designer requires the top level module, or main simulation driver, be at the system level, implying the module cannot have external input or output ports for sending or retrieving data. For this research effort, the main driver is named *Top Level*. The main driver, as shown in Figure 3.1, encapsulates a node positioning segment and a communications segment. Both are described further in this section.

A key difference in the routing methodology used by the *Satcom_router* module and the *DARTING* and *exBF* primitive routing modules required significant simulation model tailor to enable the complete model to function correctly with each of the routing modules. The *Satcom_router* module routes packets from the first gateway to the second

38

gateway, while the *DARTING* and *exBF* modules routes packets from the satellite nearest the first gateway to the satellite nearest the second gateway. The basic simulation model is presented below, with the routing module-inspired differences annotated as appropriate.

Figure 3.1 Top Level Diagram
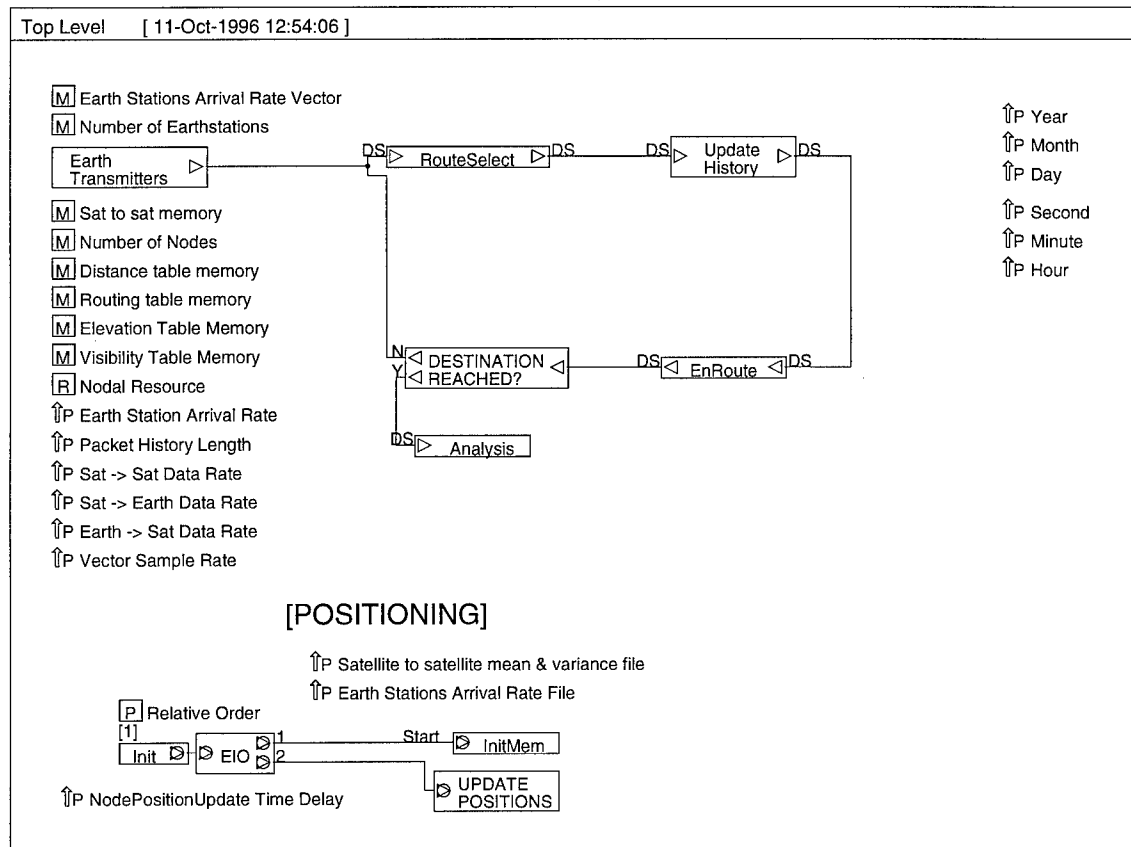
3.8.2.1 Positioning The positioning segment of the *Top Level* driver consists of two sublevel modules: *initialize* and *update_node_position*. Once the simulation begins, the two sublevel modules are given a higher execution priority, over other modules at the same level, in the model. At the simulation kickoff, the *initialize* module is executed prior to the *update_node_position* module. This execution ordering is controlled by the *Init* and *EIO* modules. A simulation flag, inside the *Init* module, tells the simulation that the blocks connected to this module have execution precedence over other system modules. The *EIO* module is an execute in order module, meaning any blocks attached to

39

the first output port execute before those connected to the second output port. The Cadence-provided *BSIM* module is called by the *initialize* module to retrieve positioning information from SatLab, related to the number of nodes (satellites and earthstations), as well as the locations of fixed earthstations. After this information is received, the simulation's global memories containing the number of nodes, number of earthstations, earthstation latitudes, and altitudes are initialized. Once the *initialize* module has completed execution, control of the simulation execution is then handed over to the *update_node_position* module.

The *update_node_position* module has two purposes: 1) An update delay time (varying by constellation) controls how often SatLab positional information is retrieved, via a call to the *BSIM* module and 2) global memories used to store routing information and relative positions between communicating nodes, are either created or updated while the routing table is formed with visibility and nearest neighbor constraints. The routing table is built using either the *Satcom_router*, provided in SatLab, or the *exBF* primitive routing routine, described in Section 3.5.6. For the *exBF* algorithm, update packets are sent out into the network to update the neighboring nodes. Once these update packets complete their network traversal, they are dumped back into the routing module for further processing, if necessary. Although the *DARTING* module isn't located inside the *update_node_position* module, the *update_node_position* module contains a triggering device which alerts the *DARTING* module of a new SatLab update. This module is repeatedly executed at the user-specified intervals throughout the simulation, via the implementation of a parameter-based delay module and a corresponding feedback loop to the delay input.

3.8.2.2 Communications The communications segment of the *Top Level* driver is responsible for performing five main functions: transmitters, routing selection, transmission path, updating of packet history metrics, and determination if the data has

reached its final destination. The functions form a closed loop communications path for the data packets to traverse the network. Each function is described in more detail below.

The transmitter function starts the communications simulation by creating and transmitting data packets from the source nodes. For this effort, the gateways are the only packet generators. The packets are created and transmitted by the *Earth_transmitter* module, which generates packets based on a bursty generator. The *Earth_transmitter* module is also responsible for creating the two data structures used during the simulation: the Sat_DS and the Sat_Route_DS data structures. Both structures are defined in Tables 3.1 and 3.2. The Sat_DS data structure contains packet information relative to the source and destination types, the packet sequence number, and the packet creation time. In addition, a Sat_DS w/Payload data structure is required to implement the *exBF* and *DARTING* modules. The Sat_DS w/Payload data structure (a child of the Sat_DS data structure) includes all the fields of the Sat_DS data structure , as well as some additional fields used by the routing algorithm primitives to update nearby network nodes (see Table 3.3 for Sat_DS w/Payload data structure). The Sat_DS data structure is encapsulated inside the Sat_Route_DS data structure within its Data field. The Sat_Route_DS data structure is designed to represent the packet traversing the network. In addition to the Data field, the Sat_Route_DS data structure also contains the current location address, the next location in the path's address, the packet hop count, and the packet history, which records all nodes traversed in the network. To implement the *exBF* and *DARTING* modules, a priority field was added to the Sat_Route_DS data structure to create a child data structure, Sat_Route_DS w/Priority (see Table 3.4). In the *Earth_transmitter* module, the priority field is assigned a value of 0, ensuring the higher priority update packets created by the *exBF* and *DARTING* modules (with a priority of 1) will be treated appropriately.

Table 3.1 The Sat_DS Data Structure.

FIELD	DATA TYPE	VALUE
source	Integer	0 to number of earthstations -1
destination	Integer	0 to number of earthstations -1
packet length	Integer	1024 bits
sequence number	Integer	0 to infinity
time stamp	Real	time of creation

Table 3.2 The Sat_Route_DS Data Structure.

FIELD	DATA TYPE	VALUE
Current	Integer	Current node location
Next	Integer	Next node in path
Data	Sat_DS	Encapsulated packet
Hop Count	Integer	Number of network hops
History	Int Vector	History of network traversal

Table 3.3 The Sat_DS w/Payload Data Structure.

FIELD	DATA TYPE	VALUE
source	Integer	0 to number of earthstations -1
destination	Integer	0 to number of earthstations -1
packet length	Integer	1024 bits
sequence number	Integer	0 to infinity
time stamp	Real	time of creation
packet type	Integer	type of packet (DARTING)
cost	Integer	path cost
from_node	Integer	previous node
to_node	Integer	next node
payload	Int Vector	routing update information
scl_list	Vector	DARTING routing update

After generation, the packet flows into the *RouteSelect* module where the next hop, along the packet's traversal of the network, is determined. Because of the routing module differences described above, the *RouteSelect* module had to be specifically tailored to work with each routing module. The three *RouteSelect* modules are described below.

42

The *RouteSelect* module for the *Satcom_router*-based model is the simplest of the three. The module looks at the current location and destination, using them to calculate an index value into the routing table global memory. The value, read out of the global memory table, is the next node the packet is routed to. Before leaving this module, the Next field of the Sat_Route_DS data structure is updated. Upon completion, the packet flows into the *Update History* module.

Table 3.4 The Sat_Route_DS w/Priority Data Structure.

FIELD	DATA TYPE	VALUE
Current	Integer	Current node location
Next	Integer	Next node in path
Data	Sat_DS	Encapsulated packet
Hop Count	Integer	Number of network hops
History	Int Vector	History of network traversal
Priority	Integer	Packet Priority (0,1)

The *RouteSelect* module for the *exBF*-based model is somewhat more complex. Upon entering the module, if the packet is on its first hop (gateway to nearby satellite) or its last hop (nearby satellite to gateway), the Next field is set manually, based on a gateway to nearest satellite translation vector set in conjunction with each SatLab update, and the packet is forwarded to the module's output port. If the packet is not on its first or last hop, the packet is forwarded to a tailored *RouteSelect* module, which uses a slightly smaller routing memory table based on satellite to satellite routing. After the tailored *RouteSelect* module assigns the Next field, the packet is passed on to the *Update History* module. Update packets generated by *exBF* are assigned their routing information in the *Build/Chop* module and completely bypass this module.

The *RouteSelect* module for the *DARTING*-based model is the most complex of the three routing modules. Upon entering the module, if the packet is on its first hop (gateway to nearby satellite) or its last hop (nearby satellite to gateway), the Next field is

set manually, based on the same translation vector described above, and the packet is forwarded to the module's output port. If the packet is not on its first or last hop, the packet is forwarded to the *DARTING* module, which appends routing data to the end of the packet. Update packets can also be triggered by the incoming data packets. After being released from the *DARTING* module, the Next field is set before leaving the module, based on the routing data appended from the *DARTING* module.

The *Update History* module has two primary functions. First, it updates the History field of the Sat_Route_DS data structure with the latest value for Current. Second, the Hop Count field of the packet is incremented by one as it traverses the network. Upon leaving this module, the packet flows into the *EnRoute* module.

The *EnRoute* module is responsible for deciding which path the packet should take: uplink, crosslink, or downlink. This determination is made by analyzing the Current and Next fields. Each link is implemented by block modules (*Sat->Sat*, *Sat->Earth*, and *Earth-> Sat*). The link modules handle the delay aspects of the model (transmission, propagation, queuing, and processing).

Upon leaving the *EnRoute* module, the *Destination Reached?* module determines if the packet has reached its final destination. If the Current field of the Sat_Route_DS data structure matches the address of the Destination Field in the Sat_DS data structure, the packet is routed to the *Analysis* module for data analysis. If the *exBF* or *DARTING* algorithm is being used, and the priority of the packet is 0, and the Destination field of the packet is set to the Last Hop flag, then the packet is routed to the *Analysis* module as just described. If, instead, the packet's priority is 1, then the packet is routed to the *Build/Chop* module. Otherwise, the packet is routed back to the *RouteSelect* module for further handling.

The *Build/Chop* module, used only for the *exBF* implementation, handles the encapsulation/decapsulation of routing update packets from and to the routing algorithm primitive located in the *update_node_position* module. The routing algorithm primitive

generates packets in the Sat_DS w/Payload form. These packets are then encapsulated by the *Build/Chop* module into the Sat_Route_DS w/Priority data structure. The priority field of the packet is set to 1, the Current and Next fields are adjusted for routing purposes, and the remainder of the packet's fields are set appropriately to permit the packet to traverse the communications portion of the model. Once the routing update has returned to the *Build/Chop* module via the *Destination Reached?* module, the Sat_DS w/Payload portion of the packet is extracted from the network packet and is returned to the *exBF* module for further processing.

3.9 Model Verification

To adequately verify, or debug, the simulation model, the verification process was divided into two portions: the satellite and earthstation positioning segment, and the communications segment. Both portions are discussed below.

3.9.1 Positioning Verification To correctly verify the positioning functionality of SatLab, the correct location of the orbiting satellites needed to be determined. The constellation data file, including both the gateway and satellite positional information, was loaded in SatLab and several test epochs were specified. The positional information at each epoch was compared against positional information calculated from Keplerian orbital mechanics parameters. The calculated positional information matched the simulation-determined positions exactly. In addition, the simulation was executed to verify the orbital path, polar crossings, and satellite period performed as expected.

3.9.2 Communications Verification The verification of this portion of the model required a bottom-up testing approach. This approach involved testing the lowest level modules, or C coded primitives, before evaluating the higher level modules. In general, two types of testing were required to completely verify the model, the Designer-built hierarchical modules, and the user-designed C code primitives.

3.9.2.1 Designer Module Block Verification The verification process tested modules at the lowest possible level and then built upward from these modules. This process continued until the main driver module, *Top Level*, was fully verified for correct operation.

The Designer Interactive Simulation Manager (ISIM) was used, extensively, to perform the initial module verification. The ISIM was used to monitor packet flow through the network. Breakpoints, set at various points throughout the model, allowed detailed investigation of the data structures and delay, and routing calculations to be made. Theoretical values were compared to the actual values observed from the simulation. Internal displays, at key points in the model (such as inside the multidimensional node resource), permitted the verification of expected queuing behavior.

Short simulations were executed with data collection probes placed throughout the model. This effort verified the overall performance of the model. Particular concern was placed in the examination of the various delay components throughout the model. The values obtained, were again compared with the theoretical values. In addition, the effects of various loading levels used were monitored to evaluate the performance of the networks under evaluation.

3.9.2.2 Primitive Verification Primitive modules were used for two primary purposes in this research, communication between SatLab and Designer (the *BSIM* module), and the various routing algorithms, including the *Satcom_router* module, the *exBF* module, and the *DARTING* module.

The *BSIM* module had one significant flaw. According to the SatLab documentation, if the state of health flag for a given satellite was set to zero (indicating a satellite failure had occurred), the satellite visibility flag was to be set to zero as well. During the verification process of the *BSIM* module, when the state of health flag was set

to zero, the visibility flag passed back through *BSIM* still showed a one (indicating a visible, healthy satellite).

The *Satcom_router* module shared the same problem as the *BSIM* module, since it used the BSIM-derived visibility matrix. *Satcom_router* readily established routing paths to satellites which had a state of health set to zero. Manual calculations were made to verify the routing paths selected by the router were correct.

Because satellite survivability is the key focus of this effort, the apparent inoperability of the state of health flag was a major concern. However, further investigation revealed an easy workaround. If the altitude of a satellite was adjusted to a level at or below the earth's surface, SatLab was forced to set the satellite's visibility flag to zero. From Designer's perspective, the modified satellite constellation behaved as expected with the pseudo-failed satellite workaround.

The verification process for the *exBF* and *DARTING* algorithms was similar to that for the *Satcom_router*. Via ISIM, the distance and visibility tables produced by *BSIM* were used to verify valid routing paths were selected by the algorithm primitives. The routing paths selected were verified for optimality. In addition, the model infrastructure required to implement the *exBF* and *DARTING* algorithms was verified for correct operation via the ISIM.

3.10 Model Validation

Traditionally, the validation of a simulation model requires validating the operating assumptions, the input parameter values and distributions, and the output values and conclusions associated with the model [Jai91]. Each of these three aspects are generally subject to validity tests, using one or more of three possible sources: expert intuition, real system measurements, and theoretical results. For this research effort, expert intuition was used. Measurement of real systems was impossible, since no LEO systems are currently operational. Theoretical models are also not applicable, since classical queuing

47

models do not fit the dynamic environment of the satellite networks being modeled. Thus, the model validity involved a step-wise approach for the operating assumptions, the input parameters, and output results.

3.10.1 Validation of Operating Assumptions As discussed previously in Section 3.5, the modeled operating environment matches that of the proposed Iridium network [Mot90, LeM93]. In those cases, where required operating assumptions were not addressed in previous literature, engineering intuition and faculty consultation was utilized. The traffic distributions associated with source and destination packet addresses is one such example.

3.10.2 Validation of Input Parameters The data rates, as described in Section 3.6, were derived from the proposed Motorola (Iridium) [Mot90] satellite system.

3.10.3 Validation of Output Results The validation of output results utilized the bottom-up approach taken in the model verification. The output results of interest, as specified in Section 3.3, are the packet delay through the network, packet hop count, and the number of dropped packets.

3.10.3.1 Packet Delay Validation The packet delay refers to the difference in time between the transmittal of the first bit, of a given packet at the source transmitter, and the receipt of the last bit of the packet, at the destination by the receiver. The delay experienced has five components: 1) packet transmission, 2) multiple access, 3) propagation, 4) satellite processing, and 5) queuing.

The delay verification followed two step process. First, the ISIM was used at an extremely low loading level (the loading level was selected to ensure no queuing delay occurred. The communications model was executed in a stepwise fashion through each of the first four delay components listed above to validate the delay calculations. After all discrepancies were removed, full background simulations were run at several different loading levels to validate the queuing delays. Probes placed inside the simulations validated the traditional queuing delay versus load relationship.

48

3.10.3.2 Hop Count Validation The hop count of a packet refers to the number of hops a packet makes to traverse the network. This portion of the model was very straightforward to evaluate, since the hop count information was calculated inside the *update_history* module. Via use of the Designer ISIM tool, the Hop Count field of the Sat_Route_DS data structure was examined, before and after entering the update_history module, verifying the hop count was incremented. Furthermore, an additional check was made once the packet reached its final destination, by inspecting the History field of the Sat_Route_DS data structure, to verify the information contained therein reflected the Hop Count number.

3.10.3.3 Dropped Packet Validation Dropped packets refer to those packets which were not successfully transmitted. In this model, packets are dropped due to insufficient queue space (fixed at 500 packets) in the multidimensional server resource. Queue occupancy probes, in conjunction with the packet rejection probes placed on the dimensioned server resource, validated the presence of dropped packets by looking at the maximum values of each dimension over the length of the simulation.

For the *exBF* and *DARTING* implementations of the model, an additional category of dropped packet was possible. Categorized as confused packets, this category reflects the possibility of a packet either exceeding its allowable hop count or in attempting to proceed directly from one gateway to another without utilizing satellite links. This scenario is possible with the both custom algorithms when standard data packets enter the network before the routing update from SatLab has been completely processed by the routing algorithm primitive.

3.11 Detailed Test Procedures

One of the main disadvantages of utilizing the simulation approach for performance analysis, is the significant time investment required to completely investigate the problem. This effort is no exception. This research has three variable factors: 1)

Various levels of constellation degradation, up to and including the complete constellation, 2) various levels of traffic loading, and 3) implementation of multiple dynamic routing algorithms. The problem is further complicated by the need to run several statistically independent trials of the same scenario (identical constellation degradation level and routing algorithm), which significantly increases the numbers of trials required. Furthermore, the simulations ideally should be run for at least the period of time required for the subject constellation to repeat itself, in order to garner a complete survivability picture. From the review of previous research, the simulation time required is measured in calendar **days**, not hours.

The time-intensive nature of this research, mandates the need for a tightly focused, well-defined testing approach. The basic research was conducted in two phases: 1) preliminary trials and 2) detailed tests. Both phased are discussed in further detail below.

The preliminary phase of the research was designed to derive baseline model information to be used in developing the detailed tests. This work focused on gaining a better understanding of the impacts of satellite degradation and various loading levels on the networks being modeled. For the satellite degradation portion, the granularity of removing satellites from the network model was evaluated (should the satellites be removed 1 at a time, 3 at a time, or 5 at a time). The time required to execute the full length simulations was also closely monitored. These results were used to fully define the detailed tests described later.

It quickly became apparent that full length simulation would require extensive amounts of time. A complete set of three independent tests (each 15 simulation minutes in length), using the *Satcom_router*, took 1.5 days to run with a 1% loading level and 5 days to run with a 11% loading level. This time requirement was heavily dependent on the priority set by the Designer simulation tool. The numbers quoted above reflect the highest possible priority and memory utilization available. Simulations which were run with lower priority and memory utilization settings took much longer to complete.

Based on the preliminary tests conducted, the satellite degradation rate and loading levels were established. The 1%, 6%, and 11% loading levels were selected. Four specific Iridium constellations were selected, starting with the complete constellation, and ending with 36 satellites removed from the constellation. These various levels and constellations were chosen based on a tradeoff between the usefulness of the data provided and the large amount of CPU simulation time required for a finer granularity of simulation criteria.

The degraded Iridium constellations were selected in a systematic manner. The packet traversal paths, built by the *Satcom_router* over the length of the simulation, were examined. The 12 most critical satellites were removed from the constellation, and the process was repeated until a total of 36 satellites had been removed from the constellation. This method of failed satellite selection was compared against random selection of satellite failure and was found to provide better constellation continuity over the period of evaluation. In addition, a Iridium constellation with 48 satellites removed was attempted, but network connectivity throughout the evaluation period could not be maintained due to lack of line-of-sight visibility between the remaining satellites. As mentioned in Section 3.2.3, the Teledesic network was not evaluated due to the extensive computer time and memory requirements.

The constellation selection was bounded by the design decision that both gateways had to have at least one satellite in view at all times throughout the evaluation period. Obviously, failure to do so would result in the loss of transmission/receiving capability. In such cases, alternate methods of transmitting data, such as using land-line communications to another gateway with access to a satellite, would be required. Such scenarios, however, are outside the scope of this effort.

The full-length detailed tests, as defined in Table 3.5 provided the data for the analysis of the problem, presented in Chapter 4. All combinations of the three variable

factors described above were used to provide a clear picture of Iridium's survivability. Also note, each test was run 3 times with statistically independent seeds.

Table 3.5 The Iridium Test Cases.

NUMBER OF DEAD SATELLITES	LOADING LEVEL (Percentage)	DYNAMIC ROUTING ALGORITHM USED
0	1	Dijkstra
0	1	ExBF
0	1	DARTING
0	6	Dijkstra
0	6	ExBF
0	6	DARTING
0	11	Dijkstra
0	11	ExBF
0	11	DARTING
12	1	Dijkstra
12	1	ExBF
12	1	DARTING
12	6	Dijkstra
12	6	ExBF
12	6	DARTING
12	11	Dijkstra
12	11	ExBF
12	11	DARTING
24	1	Dijkstra
24	1	ExBF
24	1	DARTING
24	6	Dijkstra
24	6	ExBF
24	6	DARTING
24	11	Dijkstra
24	11	ExBF
24	11	DARTING
36	1	Dijkstra
36	1	ExBF
36	1	DARTING

NUMBER OF DEAD SATELLITES	LOADING LEVEL (Percentage)	DYNAMIC ROUTING ALGORITHM USED
36	6	Dijkstra
36	6	ExBF
36	6	DARTING
36	11	Dijkstra
36	11	ExBF
36	11	DARTING

3.12 Summary

This chapter has focused on developing a satellite communication model for the evaluation of the Iridium satellite network. Section 3.2 contained the problem definition. The research scope, systems architecture, geographic area of evaluation, data traffic, and expected results were presented in this section. The various performance metrics used for this research problem, were covered in Section 3.3. The metrics chosen for this evaluation were network delay, hop count, and the number of dropped data packets. The operating assumptions, design parameters, and design factors which define the simulation model were discussed in detail in Sections 3.5, 3.6 and 3.7. Section 3.8 presented the simulation model used in this evaluation. The verification of the model was discussed in Section 3.9. Both the positioning and communications verification were presented. The validation of the model was covered in Section 3.10, where the operating assumptions, input parameters, and output results were all evaluated. Section 3.11 described the testing approach taken for this research effort.

CHAPTER 4

ANALYSIS

4.1 Introduction

In this chapter, the data generated from the test cases, described in Chapter 3, is analyzed. Section 4.2 presents a simulation time analysis, demonstrating the extensive amount of time and resources required to conduct this effort. The test data for the Satcom_router, Extended Bellman-Ford (exBF), and DARTING algorithms are discussed in Sections 4.3, 4.4, and 4.5. Section 4.6 discusses and demonstrates the overall survivability of the Iridium constellation. A summary of the chapter is presented in Section 4.7.

4.2 Simulation Time Analysis

As alluded to in the previous chapter, the time required to conduct the simulations (for the various scenarios specified in Chapter 3) limited the number of scenarios for investigation. Two factors caused significant simulation time durations. The first factor was the loading level. For the three routing algorithms simulated, the higher the loading level, the longer the durations of the simulation. This behavior resulted from increases in the numbers of data packets in the system at any given time. The other major factor, in simulation time required, was the number of satellites in the constellation. For the Satcom_router and DARTING algorithms, the simulation time increased predictably, (shown in Table 4.1) both with respect to increasing loading levels and decreasing numbers of satellites. The loading levels increased the time due to the greater number of data packets in the network. The decreased number of satellites resulted in longer data transmission routes from source to destination, which also increased the simulation overhead (and time required), particularly for the satellite queuing and crosslink

processing. However, for the exBF scenarios (also shown in Table 4.1), while the simulation time did increase as the loading level increased, the simulation time decreased sharply as the number of satellites in the constellations decreased. This increase was due to the large numbers of update packets (covered in detail in Section 4.4.3) that were present for the IRFull and IR-12 constellations (IRFull refers to the complete Iridium constellation, while the IR-12 constellation simulates a Iridium constellation with 12 failed satellites.).

Table 4.1. Simulation Execution Time (Calendar Days/CPU time (seconds))

IRFull				
Algorithm	**Base**	**1 % Load**	**6 % Load**	**11 % Load**
Dijkstra	0.003/143	0.065/5242	0.50/28821	0.65/51416
exBF	7.1/316240	6.1/132182	7.7/278668	8.2/433165
DARTING	0.013/906	0.24/19617	1.7/108630	4.7/185489
IR-12				
Dijkstra	0.004/148	0.074/6046	0.39/31429	0.72/58286
exBF	3.3/178560	1.24/98211	5.9/235767	5.26/291028
DARTING	0.02/1129	0.39/31428	2.4/119093	3.4/264002
IR-24				
Dijkstra	0.006/240	0.12/9755	0.65/51977	1.26/100516
exBF	0.0475/2029	0.27/18206	1.2/89164	2.2/160440
DARTING	0.04/1410	0.67/33850	2.3/169643	5.7/349355
IR-36				
Dijkstra	0.006/269	0.15/12164	0.76/61763	1.37/111009
exBF	0.030/1229	0.38/16355	1.87/79874	2.33/161764
DARTING	0.02/1522	0.60/41113	2.3/192517	8.6/354651

A summary of execution times for all scenarios can be seen in Table 4.1, where the simulation time for a single iteration is presented in terms of both calendar days and CPU time (in seconds). Recall, that the loading levels reflect the number of user requests from the two gateways (set at 1, 6 and 11 percent of gateway capacity). The Base column, in Table 4.1, refers to the initial test cases used to determine the various routing paths

chosen and establish the system non-queuing delays (The data rate for each of the gateways in these cases was set to 1 data packet per second). In all cases, the simulation priority (set to zero priority) used, was the maximum allowed by Designer. Initially, lower priorities were attempted, but the resulting simulation execution times at a minimum, doubled. Individual simulation completion times varied widely, depending on the number and type of other processes on any given machine.

Because of the different processing requirements of the algorithms used in the simulation models, the simulation time utilization varied. This variance was measured by enabling the Designer simulation profiler, which generated simulation CPU time statistics, broken down by module. For the Satcom_router-based model, the largest single CPU time consumer was the overhead needed to manage and collect the data for the satellite input queues, requiring approximately 34 percent of all CPU time spent. The routing module, Satcom_router, took 1 percent of the overall CPU time. The bulk of the CPU time was taken up by low level data management routines. For the exBF-based model, the exBF module took up 36 percent of the CPU time, while 24 percent was taken up by the satellite input queue servicing. Finally, for the DARTING-based model, the DARTING module took up 37 percent of the CPU time, while 24 percent was taken up by the satellite input queue servicing. Obviously, the simulation time spent executing the Satcom_router algorithm was dwarfed by the execution time required by the exBF and DARTING modules. This disparity was expected, since the exBF and DARTING modules utilize update packets to respond to a SatLab update, while the Satcom_router module does not, assuming instead instantaneous network updates.

4.3 Analysis for Dijkstra's Algorithm

The survivability analysis conducted, using the Dijkstra-based Satcom_router algorithm, was composed of tests which included the four Iridium constellations and three separate loading levels, as described in Table 3.5. The analysis focused on the three

performance metrics specified in Section 3.3: 1) end to end packet delay, 2) hop count, and 3) number of dropped data packets.

4.3.1 Dijkstra Packet Rejection Rate The packet rejection rate, or number of dropped packets, was not a factor for any of the combinations of Iridium constellations or loading levels when using the Dijkstra algorithm. For the 1 percent loading level, no rejected packets were recorded for any of the simulations. At the 6 percent loading level, the maximum packet rejection rate recorded was 0.004 percent, while at the 11 percent loading level, the maximum recorded rate was 0.02 percent. These extremely low rejection rates were impacted by two important factors. First, the satellite input queue buffers had sufficient space to handle the bursty traffic. Second, as mentioned in Section 3.5.6, the Satcom_router assumes the entire network immediately knows the status of all other network nodes, immediately after a SatLab update has occurred. This assumption freed the network from having to pass around network update packets which take resources away from standard data packets, correspondingly packet rejections are increased. While unrealistic, this assumption provided a baseline "best-case" scenario, using the specified loading levels and constellations, against which the more realistic exBF and DARTING scenarios (which utilize update packets to propagate network status) could be compared.

4.3.2 Dijkstra Hop Count The hop count was the primary variable factor influencing packet delay for the Dijkstra analysis. The hop count measures the number of hops the packet must traverse enroute to its destination node (from Washington, DC to Jerusalem, Israel, or vice-versa). Each hop refers to a communication link between two network nodes. For this effort, there are three types of links: 1) gateway to satellite (uplink), 2) satellite to satellite (crosslink), or 3) satellite to ground (downlink). For the complete Iridium constellation (IRFull), 4 hops were required (the hop count varied slightly during the evaluation period). For the worst-case degraded Iridium constellation with 36 satellites removed (IR-36), each packet required 9 hops to reach its destination.

The IRFull and IR-36 constellations are shown in Figures 4.1 and 4.2 respectively (Similarly, IR-24 refers to the degraded Iridium constellation with 24 satellites removed, while the IR-12 constellation simulates a Iridium constellation with 12 failed satellites).

Figure 4.1 IRFull Constellation
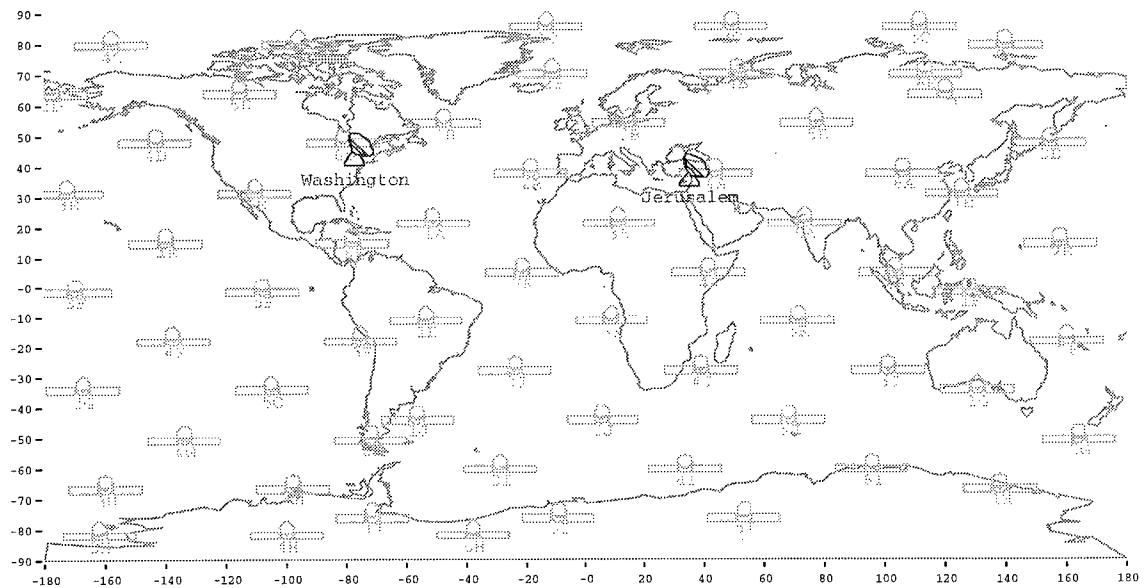
Recall from Chapter 3, a Iridium constellation with 48 satellites removed was examined, but network connectivity throughout the evaluation period could not be maintained due to lack of line-of-sight visibility between the remaining satellites. Therefore, the IR-36 constellation was the most severely degraded constellation evaluated.

By far, the largest change in hop count, three (as shown in Table 4.2), was between the IR-12 and IR-24 constellations. This variation was due to the lack of satellites in the northern polar region for the IR-24 constellation, forcing the packets to instead traverse through the equatorial region.

4.3.3 Dijkstra Packet Delay While the packet delay, the key metric for this effort, varied considerably, at no point did the delay metric exceed the real-time performance criteria of 400 milliseconds (msec) that was established in Section 3.2.6. This result was

58

expected given the low loading levels used. An explanation of the delay components and detailed results follow below.

Figure 4.2 IR-36 Constellation
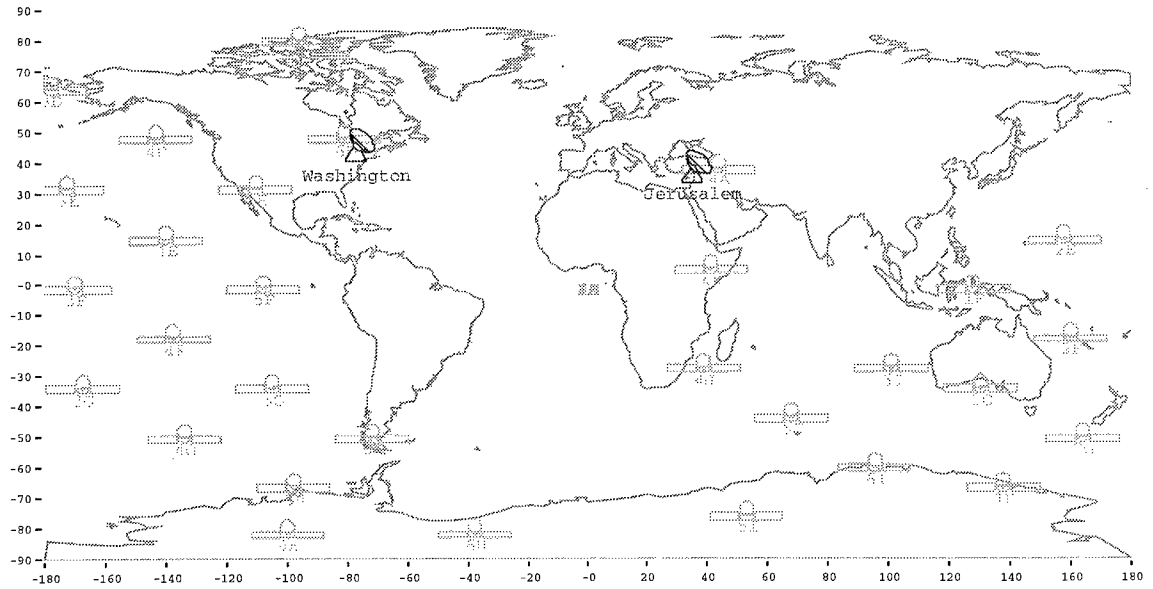
Before discussing the specific results, a brief discussion of the various delay components experienced by an individual data packet is in order. The average delay of a data packet, D_{packet}, can be represented symbolically by Equation 4.1:

$$D_{packet} = D_{TDMA} + D_{trans} + D_{uplink} + N(D_{satproc} + D_{satq}) + (N-1)D_{cross} + D_{downlink} \qquad (4.1)$$

where D_{TDMA} represented the delay incurred by using the TDMA scheme, D_{trans} is the transmission time of the packet, D_{uplink}, $D_{downlink}$, and D_{cross} are the propagation times for the various links, $D_{satproc}$ and D_{satq} are the satellite processing time (packet decoding and routing selection) and the amount of time spent by the packet in a satellite queue, respectively, and N refers to the number of satellites a packet traverses from its source to destination.

A breakdown of the delay factors, in Equation 4.1, is also necessary. The D_{TDMA} factor is set at 30 milliseconds, as described in Chapter 3. The mean satellite processing time was 1 msec. The transmission delay, D_{trans}, made up of the transmission times for the uplink, crosslinks and downlink, was less than 1 msec. Based on a maximum distance of 867 kilometers (km) [Rai94] between a gateway and the nearby satellite, the maximum value for the $D_{downlink}$ and D_{uplink} factors is 2.9 msec. In addition, the D_{cross} factor, based on a maximum distance of 4355.3 km [Rai94] between satellites, reached a maximum value of 14.5 msec.

As alluded to earlier, the hop count experienced by packets greatly influences the delay they experience as they traverse the network. This influence can be clearly seen by the impact of the factor N, the number of satellites traversed. The additive affects of these hop count related delays (for the moment ignoring satellite queuing delays) are illustrated in Table 4.2.

Table 4.2. Average Base Packet Delay (msec) by Constellation

Constellation	Ave Delay (msec)	Hop Count (low/high/most common)
IRFull	79.7	4/5/4
IR-12	82.6	4/5/5
IR-24	130.3	7/9/8
IR-36	152.0	9/10/9

The queuing delay (reflected in Table 4.3), was influenced both by the level of degradation in the Iridium constellation and to a lesser extent, the loading level. This relationship can be clearly seen by observing the larger variance in delays (from the non-queuing delays) between the IRFull constellation and the IR-36 constellation (3 and 8 msec respectively), versus the variance in delays (again from the non-queuing delays) corresponding to the 1% and 11% loading levels for the IRFull constellation. The

queuing delays averaged 6.5 msec for the 1 percent loading level, 9.5 msec for the 6 percent loading level, and 12.25 msec for the 11 percent loading level.

Table 4.3. Average Packet Delay (msec) by Constellation

Constellation	1% Load	6% Load	11% Load
IRFull	83	86	88
IR-12	91	94	96
IR-24	137	140	144
IR-36	160	163	166

In summing up the packet delay for the Satcom_router-based runs, it must be emphasized that no combination of loading level/degraded constellation resulted in a packet end-to-end delay greater than 400 msec. In fact, the worst-case packet delay, 307 msec at worst (shown in Table 4.4), was well below the 400 msec threshold when a path could be found (e.g., the number of satellite failures was less than 48).

Table 4.4. Worst-Case Packet Delay (msec) by Constellation

Constellation	1% Load	6% Load	11% Load
IRFull	153	189	225
IR-12	163	205	236
IR-24	199	249	280
IR-36	239	263	307

A total of three independent replications of each simulation trial were executed. A unique seed value was used for each replication. For all scenarios, the standard deviation from the mean delay of a particular replication was less than 1 percent.

4.4 Analysis for Extended Bellman-Ford

Similar to the Dijkstra analysis, the exBF analysis focused on the three performance metrics, specified in Chapter 3. Unlike the Satcom_router, the exBF algorithm utilized update packets to keep network status current. The presence of these update packets were reflected in the results presented below.

4.4.1 Extended Bellman-Ford Packet Rejection Rate The packet rejection statistics were determined by two types of rejected packets. The first type occurred when a standard data packet was displaced by an update packet. If this action occurred when the satellite queue's input buffer was full, the displaced data packet was lost. In six of the exBF scenarios (all IRFull and IR-12 constellation scenarios), the exBF algorithm failed to converge in numerous occasions after a SatLab update, due to the initial loss of an update packet. The term "converge" refers to the ability of the routing algorithm to generate a global routing solution (not necessarily optimal) after a SatLab update. In the context of this effort, the algorithm was considered to have successfully converged when no additional update packets were being generated by the algorithm. This loss resulted in a looping situation which generated millions of extraneous update packets. This looping condition lasted until the next SatLab update was received. However, the large number of update packets did not result in large numbers of rejected data packets, since the looping situations occurred between nodes not on the transmission path. For example, in one of the IR-12 scenarios, the route used by the data packets to go from gateway to gateway traversed through 5 satellites (45, 35, 57, 48, and 59), while a looping situation had occurred between satellites 39 and 40. The second type of rejected packet (confused packet) occurred when a packet entered the network before a suitable route had been established to its destination, or when a suitable path was not found by the routing algorithm. When this occurred (typically at the higher loading levels), the packet would either wander until its hop count was exceeded or attempt an illegal path (gateway to gateway), where it would be dropped from the network. While this was not a important factor for the IRFull, IR-12, and IR-24 constellations, the IR-36 constellation experienced significant numbers of confused data packets (shown in Table 4.5). This large number of confused data packets was due to the inability of the exBF algorithm to establish a suitable path after a SatLab update midway through the IR-36 scenarios from one of the gateways.

62

Table 4.5. Confused Data Packets

Constellation	1% Load	6% Load	11% Load
IRFull	8	8	100
IR-12	0	100	136
IR-24	0	25	75
IR-36	26558	132367	240117

The overall packet rejection rates for exBF, except for the IR-36 constellation, were within a 1% Grade of Service (GOS). While the IR-36 constellation consistently experienced packet rejection rates of nearly 10 percent, the remainder of the rejection rates were under the 1 percent GOS criteria (as shown in Table 4.6), experiencing a packet rejection rate of no greater than 0.05 percent.

Table 4.6. exBF Packet Rejection Rate (%)

Constellation	1% Load	6% Load	11% Load
IRFull	0.06	0.05	0.05
IR-12	0	0.01	0.02
IR-24	0	0.002	0.003
IR-36	9.7	9.9	10.0

4.4.2 Extended Bellman-Ford Hop Count While the hop count was an important factor in determining average packet delay, this metric was not as reliable a delay predictor for the exBF algorithm as was the Dijkstra-based Satcom_router algorithm. If the exBF algorithm converged after a SatLab update, the hop count corresponded closely to the delay value. If however, the exBF algorithm did not converge, the delays were often distorted, as will be illustrated in the next section.

4.4.3 Extended Bellman-Ford Packet Delay The packet delay metric was closely related to both the hop count and the number of update packets produced by the exBF algorithm.

As noted in Section 4.2.3, the hop count experienced by packets greatly influences the delay they experience, as they traverse the network. This influence can be seen by the impact of the factor N (in equation 4.1), the number of satellites traversed. The additive affects of these hop count related delays (ignoring satellite queuing delays) are shown in Table 4.7.

Table 4.7. Average Base Packet Delay (msec) by Constellation

Constellation	Ave Delay (msec)	Hop Count (low/high/most common)
IRFull	91	6/7/6
IR-12	101	6/9/8
IR-24	147	9/11/10
IR-36	163	11/13/12

The overall average packet delay was also primarily impacted by the hop count, although the large amount of update packets also affected the data. The data, shown in Table 4.8, reinforces the relationship between the hop count and the average packet delay (with queuing included). The magnitude of update packets can be seen in both the IRFull and IR-12 constellations by observing the update packet to data packet ratios, shown in Table 4.9, with update to data packet ratios ranging from 3.9:1 to 26.3:1. The presence of these additional update packets are reflected in the inflated queuing delays for the IRFull scenarios. Recall the IR-12 scenarios' lack of convergence did not impact the path traversal by the data packets. One striking example is the 15 msec delay increase between the base and 1 percent delays, for the IRFull constellation. This increase is 5 times larger than the 3 msec increase between the base and 1 percent IRFull delays, using the Satcom_router. For the IRFull scenarios, the average packet delays were 15 to 19 percent higher (16.7 percent average increase) than the baseline Satcom_router delays, once again reflecting the impact of the additional update packets. On the other hand, when exBF converged (or the lack of convergence did not impact the data path, as in the IR-12 scenarios), the average packet delays increased over the baseline delays from 5 to

64

16 percent (10.8 percent average increase). Overall, the queuing delays averaged 9.7 msec for the 1 percent loading level, 9 msec for the 6 percent loading level, and 12.25 msec for the 11 percent loading level.

Table 4.8. Average Packet Delay (msec) by Constellation

Constellation	1% Load	6% Load	11% Load
IRFull	102	101	105
IR-12	108	111	114
IR-24	155	158	161
IR-36	170	172	175

In Table 4.10, the worst-case packet delay continues to illustrate the impact of the update packets. A maximum delay of 5672 msec was recorded for the 1 percent IRFull constellation. Without exception, the maximum delays occurred immediately after the SatLab updates. This occurred because data packets were continually displaced by the update packets (because of the update packet's higher priority) during the initial burst of update traffic. This displacement forced the data packets to spend long periods in the satellite input queues. While the remainder of the affected simulations were not impacted as severely as the case just addressed, the maximum delays were still far higher than they would have been without the presence of the extra update packets.

Table 4.9. Ratio of Update to Data Packets

Constellation	1% Load	6% Load	11% Load
IRFull	14.1	7.3	3.9
IR-12	26.3	7.0	3.9
IR-24	0.2	0.04	0.02
IR-36	0.1	0.02	0.01

A total of 3 independent replications of each simulation trial were executed. A unique seed value was used for each replication. For all scenarios, the standard deviation from the mean delay of a particular replication was less than 1 percent.

Table 4.10. Worst-Case Packet Delay (msec) by Constellation

Constellation	1% Load	6% Load	11% Load
IRFull	5672	638	1227
IR-12	212	245	260
IR-24	226	267	280
IR-36	249	282	295

4.5 Analysis for DARTING

In the process of implementing the DARTING algorithm, as specified by Tsai and Ma [TsM95], an oversight was discovered. The original DARTING algorithm assumed that all nodes of the network are transmitting data packets. For this effort, only the two gateways were used to generate data traffic. Because of this, some nodes never received updated network information, resulting in a lack of algorithm convergence. Noting this problem, Janoso [Jan96] modified the original DARTING algorithm to generate "ping" packets from the impacted nodes whenever a lack of node usage was detected by the algorithm. This modified algorithm, henceforth referred to as Modified DARTING (MDARTING), was used for the analysis which follows.

4.5.1 Modified DARTING Packet Rejection Rate Similar to the exBF algorithm, the packet rejection rate was made up of both types of rejected packets. In the periods immediately after a SatLab update occurred, the quick burst of update traffic resulted in rejected packets. However, since the MDARTING algorithm converged in all scenarios, the packet rejection rate was never higher than 1.3 percent (which occurred once, for an IR-24 simulation), while 96.2 percent of the MDARTING simulations recorded levels at or below the 1 percent GOS.

4.5.2 Modified DARTING Hop Count The hop count was the primary variable factor influencing packet delay for the MDARTING analysis. For the complete Iridium constellation (IRFull), 7 hops were required (the hop count varied slightly during the

66

evaluation period). For the IR-36 constellation (worst-case), each packet required 12 hops to reach its destination.

4.5.3 Modified DARTING Packet Delay As noted previously, in Sections 4.3.3 and 4.4.3, the hop count experienced by packets greatly influences the delay they experience as they traverse the network. The additive affects of these hop count related delays (ignoring satellite queuing delays) are shown in Table 4.11.

The overall packet delays (reflected in Tables 4.12 and 4.13) were impacted by both the hop counts and the update packets generated by the MDARTING algorithm. The average packet delays (shown in Table 4.12), while about 7 to 17 percent higher than the Satcom_router delays (12.5 percent average increase), are well within the real-time packet delivery criteria. This result is intuitive when the additional hops the packet must make, along with the impact of the update packets, is considered. The queuing delays averaged 8.75 msec for the 1 percent loading level, 11.5 msec for the 6 percent loading level, and 14.75 msec for the 11 percent loading level. The worst-case delays (shown in Table 4.13) exceeded the real-time packet delivery criteria for 66.7 percent of the MDARTING scenarios. These delay spikes occurred without exception right after a SatLab update, because of the newly injected update traffic.

Table 4.11. Average Base Packet Delay (msec) by Constellation

Constellation	Ave Delay (msec)	Hop Count (low/high/most common)
IRFull	89	6/7/7
IR-12	101	6/9/8
IR-24	147	9/11/10
IR-36	163	11/13/12

A total of 3 independent replications of each simulation trial were executed. A unique seed value was used for each replication. For all scenarios, the standard deviation from the mean delay of a particular replication was less than 1 percent.

Table 4.12. Average Packet Delay (msec) by Constellation

Constellation	1% Load	6% Load	11% Load
IRFull	99	101	104
IR-12	108	112	115
IR-24	156	158	162
IR-36	172	175	178

4.6 Analysis

The initial focus of this effort was to evaluate the survivability of the Iridium and Teledesic satellite communications networks. As noted in Section 3.2.3, simulation of the Teledesic network proved to be infeasible, both in terms of the memory required and the processing speed needed. Thus, this effort was centered on the evaluation of the Iridium network.

Table 4.13. Worst-Case Packet Delay (msec) by Constellation

Constellation	1% Load	6% Load	11% Load
IRFull	593	461	456
IR-12	307	279	292
IR-24	575	695	1060
IR-36	383	492	643

The Iridium network proved to be highly survivable. Even with only 45 percent (IR-36) of its satellites functioning, the average packet delay performance was well within the real-time packet delivery constraint of 400 msec. This finding held for all loading levels, constellations (levels of degraded Iridium), and routing algorithms used in this effort. With more than 36 satellites removed from the constellation, network connectivity throughout the evaluation period could not be maintained due to lack of line-of-sight

visibility between the remaining satellites. Therefore, degraded Iridium constellations with fewer than 30 functional satellites were not evaluated.

One of the key simulation factors in this effort, the routing algorithm, also yielded interesting results, in terms of both packet delay and rejection rates. For the MDARTING algorithm, the average packet delays increased an average of 12.5 percent over the Satcom_router baseline delays. The worst-case delays exceeded the 400 msec packet delivery criteria for 66.7 percent of the MDARTING scenarios. The exBF algorithm also yielded mixed delay results. In 75 percent of those scenarios where the algorithm converged (all IR-24 and IR-36 scenarios) or the lack of convergence did not impact the transmission path (IR-12), the average packet delays increased an average of 10.8 percent over the baseline delays, with a worst-case packet delay of 295 msec. However, when exBF's non-convergence impacted the data path (all IRFull scenarios), the average packet delays increased on average 16.7 percent over the baseline delays. In addition, the worst-case delays exceeded the 400 msec delay criteria for 16.7 percent of the exBF scenarios, all occurring using the IRFull constellation. The packet rejection results were mixed as well. For MDARTING, the packet rejection rate was never higher than 1.3 percent (occurring once), while 96.2 percent of the MDARTING scenarios recorded levels at or below the 1 percent GOS criteria. For those exBF scenarios involving the IRFull, IR-12, and IR-24 constellations, the rejection rate was 0.05 percent or better. For the IR-36 scenarios, however, the exBF rejection rates were nearly 10 percent.

4.7 Summary

This chapter focused on the analysis of the data generated from the test cases, specified in Section 3.11. After a simulation timing analysis in Section 4.2, the performance data resulting from the three algorithms utilized in this effort was presented in Sections 4.3, 4.4, and 4.5. In Section 4.6, the complete survivability picture of the Iridium constellation was analyzed, in light of the data gathered.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary of Thesis Investigation

A brief review of the material presented in this thesis effort is necessary before addressing the conclusions and recommendations resulting from this research. In Chapter 1, a background review of recent trends in satellite communications was presented. In addition, the research goals of this investigation were defined to scope the thesis effort.

Chapter 2 provided the background information necessary to understand the Low Earth Orbit (LEO) satellite environment. After discussing the rational for favoring LEO satellites over other competing classes of satellite constellations, various facets of the LEO environment were explored. Several proposed LEO satellite constellations were discussed. Finally, several key issues shaping the design and implementation of LEO systems were presented.

The methodology used in this research effort was given in Chapter 3. In addition to further defining and scoping the problem, all operational assumptions and parameters of this investigation were presented. The focus of this chapter was to present the detailed simulations models used to attack the problem. The remainder of the chapter covered the validation and verification process used, as well as the selection of the test cases specified to evaluate the constellation survivability.

Chapter 4 presented the analysis of the data from the test cases proposed in Chapter 3. A discussion of the extensive time and CPU resources required to conduct this effort was presented. An analysis of the three routing algorithms' performance followed and examined the performance metrics specified in Chapters 1 and 3. Conclusions were then drawn from the data obtained.

5.2 Conclusions

The Iridium satellite network was shown to be highly survivable. Even with only 45 percent (IR-36) of its satellites functioning, the average packet delays were never greater than 178 msec, well within the real-time packet delivery constraint of 400 msec. This finding held for all loading levels, constellations (levels of degraded Iridium), and routing algorithms used in this effort. It shows that even if Iridium loses half its satellites, communication links can still be maintained (if both gateways have access to a Iridium satellite) with acceptable (within 400 msec) packet delivery performance.

The algorithms used, also provided some interesting results. While the MDARTING algorithm was more consistent, it was frequently outperformed by the exBF algorithm when it converged (or exBF's lack of convergence did not impact the data path, as was the case in the IR-12 scenarios). The average delays for the Modified DARTING (MDARTING) algorithm were 12.5 percent greater than the best-case Satcom_router. Similarly, the average delays for the Extended Bellman-Ford (exBF) algorithm were 10.8 percent higher than the best-case delay values, for the converging (and non-converging IR-12) exBF scenarios, while for the non-converging IRFull exBF scenarios, average delays increased 16.7 percent, as compared to the baseline values. In addition, the MDARTING algorithm experienced worst-case packet delays over 400 msec for 66.7 percent of its scenarios, while the exBF algorithm experienced worst-case packet delays over the 400 msec criteria for 16.7 percent of its scenarios. The packet rejection rate results were mixed, as well. The exBF algorithm experienced excessive packet rejection rates (greater than the 1 percent Grade of Service criteria) in 25 percent of its scenarios. In contrast, 96.2 percent of the MDARTING scenarios recorded a packet rejection rate of 1 percent or better (lower).

5.3 Recommendations for Future Research

This investigation has provided an in-depth analysis of the survivability of the Iridium satellite network. The evaluation was conducted with scenarios using three

loading levels, three routing algorithms, and four Iridium network configurations (three of which were degraded constellations). Because of the time constraints and computer resource requirements (CPU speed and memory) such studies require, two areas were unable to be evaluated. These areas form a base for future research in the area of LEO satellite network performance analysis. The proposed enhancements are as follows:

1. Conduct this investigation with much higher loading levels. While this would require computing resources not currently available, it would provide a much more comprehensive picture of the survivability of the Iridium satellite network under more realistic loading conditions.

2. Investigate the Teledesic satellite network in the same manner the Iridium network was investigated, in this effort. In addition to showing the survivability of the Teledesic network and allowing comparison with the Iridium network, the robustness of the exBF algorithm (or lack thereof) and the Modified DARTING algorithm could be analyzed further.

In closing, this effort has shown the Iridium satellite network to be highly survivable. While additional research needs to be performed to evaluate the many facets which complicate LEO networks, and further investigation of the survivability aspects utilizing more realistic scenarios, the initial results obtained here are promising. In addition, this thesis serves as the first performance analysis of a LEO system from the perspective of survivability.

BIBLIOGRAPHY

[Ana95]   F. Ananasso, "An Overview of Mobile Satellite Systems and Their Evolution Towards Satellite Personal Communication Networks," *Tenth International Conference on Digital Satellite Communications,* Vol. 2, No. 403, May 1995, pp. 435-441.

[Bak91]   J. E. Baker, "A Distributed Link Restoration Algorithm With Robust Preplanning," *1991 IEEE Globecom,* Vol 1, pp. 306-311.

[CaA87]   J. B. Cain, S. L. Adams, M. D. Noakes, T. Kryst, and E. L. Althouse, "A 'Near-Optimum' Multiple Path Routing Algorithm For Space-Based SDI Networks," ," *MILCOM 87: 1987 IEEE Military Communications Conference,* October 1987, pp. 578-584.

[Cha89]   D. Chakraborty, "Survivable communication concept via multiple low earth-orbiting satellites," *IEEE Transactions on Aerospace and Electronic Systems,* Vol. 25, No. 6, November 1989, pp. 879-889.

[ClW87]   L. P. Clare, C. Y. Wang, and M. W. Atkinson, "Multiple satellite networks: performance evaluation via simulation," *MILCOM 87: 1987 IEEE Military Communications Conference,* October 1987, pp. 404-411.

[ClJ89]   D. D. Clark, V. Jacobson, J. Romkey, and H. Salwen, "An analysis of TCP processing overhead," *IEEE Communications Magazine,* Vol. 27, No. 6, June 1989, pp. 23-29.

[Com93]   G. M. Comparetto, "A Technical Comparison of Several Global Mobile Satellite Communications Systems," *Space Communications,* Vol 11 (1993), pp. 97-103.

[Com95]   BONeS SatLab User's Guide, Cadence Design Systems, Incorporated, June 1995.

[DuG94]   D. A. Dunn, W. D. Grover, and M. H. MacGregor, "Comparison of *k*-Shortest Paths and Maximum Flow Routing for Network Facility Restoration," *IEEE Journal on Selected Areas in Communications,* Vol. 12, No. 1, January 1994, pp. 88-99.

[GaC89]   J. J. Garcia-Luna-Aceves, C. Cheng, R. Riley, and S. P. R. Kumar, "A Loop-Free Extended Bellman-Ford Routing Protocol Without Bouncing Effect," *SIGCOMM 89,* pp. 224-236.

[GoT88]   A. V. Goldberg, and R. E. Tarjan, "Finding Minimum-Cost Circulations by Successive Approximation," *Mathematics of Operations Research*, Vol. 15, August 1990, pp. 430-466.

[GrV94]   W. D. Grover, B. D. Venables, M. H. MacGregor, and J. H. Sandham, "Development and Performance Verification of a Distributed Asynchronous Protocol for Real-Time Network Restoration," *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 1, January 1991, pp. 112-125.

[GrZ89]   J. L. Gross, and R. E. Ziemer, "Distributed Routing Performance in Hostile Environments," *Proceedings of the SPIE - The International Society for Optical Engineering*, Vol 1059, 1989, pp. 22-26.

[Jai91]   R. Jain, *The Art of Computer System Performance Analysis*, John Wiley and Sons, Inc., New York, 1991.

[Jan96]   R. Janoso, *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.

[LeM93]   R. J. Leopold and A. Miller, "The Iridium™ communications system: person to person on a truly global scale," *IEEE Potentials*, April 1993, pp. 6-9.

[MeS79]   P. M. Merlin, and S. Segall, "A Failsafe Distributed Routing Protocol," *IEEE Transactions on Communications*, Vol. 27, No. 9, September 1979, pp. 1280-1287.

[Mot90]   Motorola Satellite Communications, Inc., FCC filing for application of a low earth orbit mobile satellite system (IRIDIUM), Washington, D. C. , December 1990.

[PaF94]   V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *SIGCOMM 94*, September 1994, pp. 257-268.

[Rai94]   R. A. Raines, "Performance Modeling and Analysis of Parallel Processing and Low Earth Orbit Satellite Communications Systems", Ph.D. Dissertation, Virginia Polytechnic Institute and State University, July 1994.

[Sco95]   W. B. Scott, "Iridium on Track for First Launch in 1996," *Aviation Week and Space Technology,* April 3, 1995, pp. 56-61.

[Sef93]   T. Sellers, and L. Fulton, "Using a Survivability Framework When Planning and Developing Survivable Communications Networks," *MILCOM 93: 1993 IEEE Military Communications Conference*, October 1993, pp. 178-181.

[TiH94]   D. Tipper, J. L. Hammond, S. Sharma, Khetan, B. K. Archana; and S. Menon, "An Analysis of the Congestion Effects of Link Failures in Wide Area Networks," *IEEE Journal on Selected Areas in Communications*, Vol 12, No. 1, January 1994, pp. 179-191.

[TsM93]   K. Tsai, and R. P. Ma, "Darting: A Cost-Effective Routing Alternative For Large Space-Based Dynamic-Topology Networks," *MILCOM 95: 1995 IEEE Military Communications Conference*, October 1995, pp. 682-686.

[TuA93]   R. L. Turcotte, S. Arneson, "The Role of Commercial LEOs in Global Grid," *MILCOM 93: 1993 IEEE Military Communications Conference*, October 1993, pp. 247-251.

[TuP93]   E. F. Tuck, D. P. Patterson, J. R. Stuart, and M. H. Lawrence, "The Calling Network: a global telephone utility," *Space Communications*, Vol 11, No. 2, pp. 141-161.

[URL1]   D. C.Benton, "Globalstar," http://www.wp.com/mcintosh_page_o_stuff /globals.html, June 1996.

[URL2]   T. E. Wisloff, "Tor E. Wisloff's Big LEO tables," http://www.idt.unit. no/~turwi/synopsis.html.

[URL3]   C. Levin, "Space Race," http://www.pcmag.com/issues/1505/pcm00011. html, 12 March 1996.

[Wal70]   J. G. Walker, "Circular orbit patterns providing continuous whole earth coverage," *Royal Aircraft Establishment*, Technical Report 70211, November 1970.

[Wal77]   J. G. Walker, "Continuous whole-earth coverage by circular-orbit satellite patterns," *Royal Aircraft Establishment*, Technical Report 77044, September 1977.

[Wan93]   C. J. Wang, "Structural Properties of a LEO Satellite Constellation - the Walker Delta Network," *MILCOM 93: 1993 IEEE Military Communications Conference,* Vol 3, pp. 968-972.

[Wan95]   C. J. Wang, "Delivery Time Analysis of a Low Earth Orbit Satellite Network for Seamless PCS," *IEEE Journal on Selected Areas in Communications,* Vol. 13, No. 2, February 1995, pp. 389-396.

[WeB93]   M. Werner, A. Bottcher, A. Jahn, and E. Lutz, "Analysis of Connectivity Requirements for Communication Networks Based on Low Earth Orbit Satellites," *Proceedings of 2nd International Conference on Universal Personal Communications,* Vol. 1012, October 1993, pp. 299-303.

[WeJ95]   M. Werner, A. Jahn, E. Lutz, and A. Bottcher, "Analysis of System Parameters for LEO/ICO Satellite Communication Networks," *IEEE Journal on Selected Areas in Communications,* Vol 13, No. 2, February 1995, pp. 371-381.

[Wie93]   R. A. Wiedeman, "The Role of Globalstar, a Satellite-based Personal Communications System, in the Wireless Revolution," *Pacific Telecommunications Council Fifteenth Annual Conference. Proceedings.,* January 1993, pp. 660-664.

[WiW94]   W. Willinger, D.V. Wilson, W.E. Leland, and M.S. Taqqu, "On Traffic Measurements that Defy Traffic Models (and vice versa): Self-Similiar Traffic Modeling for High-Speed Networks," *ConneXions*, Vol 8, No. 11, November 1994, pp. 14-24.

[WuM94]   W. W. Wu, E. F. Miller, Pritchard, L. Wilbur and R. L. Pickholtz, "Mobile Satellite Communications," *Proceedings of the IEEE*, Vol. 82, No. 9, September 1994, pp. 1431-1448.

VITA

Captain Douglas K. Stenger ████████████████████████████████████ Pennsylvania. After graduating from Greencastle-Antrim High School, Greencastle, Pennsylvania in 1983, he attended Hagerstown Junior College where he received an Associates Degree in Mathematics. He completed his Bachelor of Science Degree in Mathematics at Slippery Rock University of Pennsylvania in May 1987. Upon graduation, he was commissioned in May 1987 through the Air Force ROTC program and was subsequently assigned to the Basic Communications Officer Training Course in the 3401 Student Squadron in Biloxi, Mississippi in January 1988. In May 1988, he was assigned to the 1912 Computer Systems Group at Langley AFB, Virginia, as the lead Programmer for the tactical Mission Planning System. From Langley, he was transferred to Offutt AFB, Nebraska in March 1992. where he served as Systems Analyst and Test Director working on the ground system of the Defense Meteorological Satellite Program. While at Offutt AFB, he also finished a Master of Arts in Economics from the University of Oklahoma, graduating in December 1994. In May 1995, Captain Stenger entered the Computer Systems Program at the School of Engineering, Air Force Institute of Technology.

P████████████████████████████████████

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1996	Master's Thesis

4. TITLE AND SUBTITLE
Survivability Analysis of the Iridium Low Earth Orbit Satellite Network

5. FUNDING NUMBERS

6. AUTHOR(S)
Douglas K. Stenger, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
2750 P. Street
WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GCS/ENG/96D-26

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ Space Warfare Center/AE
Colonel Feldman
730 Irwin Ave Ste 83
Falcon AFB, CO 80912-7383

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This thesis evaluates the survivability of the proposed Iridium Low Earth Orbit (LEO) Satellite Network. In addition to the complete Iridium constellation, three degraded Iridium constellations are analyzed. This analysis occurs via the use of simulation models, which are developed to use three dynamic routing algorithms over three loading levels. The Iridium network models use a common set of operating assumptions and system environments. The constellation survivability was determined by comparing packet rejection rates, hop counts, and average end-to-end delay performance between the various network scenarios. It was concluded that, based on the established scenarios, the proposed Iridium constellation was highly survivable. Even with only 45 percent of its satellites functioning (modeled with 36 failed Iridium satellites), the average packet delays were never greater than 178 milliseconds (msec), well within the real-time packet delivery constraint of 400 msec. As a result, while additional research is necessary, Iridium has demonstrated the network robustness that is required within the military communications environment.

14. SUBJECT TERMS
Satellite Communications, Satellite Networks, Survivability, Performance Analysis, Network Simulation

15. NUMBER OF PAGES
89

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