BANTAN, NOUMAN, Ph.D., May 2007 COMPUTER SCIENCE

A ROUTING PROTOCOL AND ROUTING ALGORITHM FOR SPACE COMMUNICATION (165 pp.)

We have developed a routing protocol for space that creates an infrastructure which enables routers on board spacecrafts to calculate near optimum routing tables ahead of time and on-demand when network changes occur. Our routing protocol for space communication, Space OSPF (SOSPF), divides the routing domain (e.g., our solar system) into areas within areas which provides an orderly fashion of transmitting routing information throughout the routing domain. The concept of areas within SOSPF allows routing information of one area to be hidden within that area. In addition, since the trajectory of space crafts are either predictable (e.g., satellite constellation around Earth), preset (e.g., the International Space Station), or set on demand (e.g., a space shuttle), a router on board those spacecrafts calculates the time intervals where spacecrafts are in direct view with the calculating router and the propagation delays to those spacecrafts using the location of those spacecrafts and the local transmission capabilities. Then, those calculated values are dispersed throughout the routing domain. Also, we will present a new routing algorithm which allows routers on board spacecrafts to use the received routing information (i.e., the time intervals and the propagation delay) to compute the routing

table. This routing algorithm can compute shortest delay paths over conventional concurrent-link as well as intermittent-links using a store-and-forward communication scheme. Furthermore, we will present the routing performance of this new protocol in real space scenarios and show how the SOSPF routing domain stays stable after link failures as the routing domain diameter grows to the end of our solar system.

A ROUTING PROTOCOL AND ROUTING ALGORITHM FOR SPACE COMMUNICATION

A dissertation submitted
to Kent State University in partial
fulfillment of the requirements for the
degree of Doctor of Philosophy

by

Nouman Bantan

May 2007

Dissertation written by
Nouman Bantan
B.S., Indiana State University, 1991
M.S., Sam Houston State University, 1994
Ph.D., Kent State University, 2007

Approved by

Farid Fay , Chair, Doctoral Dissertation Committee
K. Anader , Members, Doctoral Dissertation Committee
Kenneth Batcher
Ruoming Jin
M. Sadegh

Accepted by

, Chair of Department of Computer Science
John R.D. Stalvey , Dean, College of Arts and Sciences

Dissertation written by

Nouman Bantan

B.S., Indiana State University, 1991

M.S., Sam Houston State University, 1994

Ph.D., Kent State University, 2007

Approved by

Javed Khan	, Chair, Doctoral Dissertation Committee
Feodor Dragan	, Members, Doctoral Dissertation Committee
Kenneth Batcher	
Kazim Khan	
Khandker Quader	

Accepted by

Robert A. Walker	,Chair of Department of Computer Science
Jerry Feezel	, Dean, College of Arts and Sciences

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENT

First I thank Allah for giving me the ability and patience to accomplish this degree. Then I thank my advisor Dr. Javed Khan for his brilliant thinking, sharp ideas, vigorous examinations, and high expectations. He helped me sharpen my analytical skills and refine my research. He made sure I was on the right track.

Most importantly, I would like to thank my wife Abir As-Sanie. She has sacrificed her life for me. Without her love, patience and generosity I would not have been able to pursue this research.

I thank my parents for giving me support and believing in me. My parents had always encouraged and looked after me from childhood until now.

I am also very grateful for having an exceptional doctoral committee and wish to thank Dr. Kazim Khan, Dr. Feodor F. Dragan, Dr. Kenneth Batcher, and Dr. Khandker Quader, for their support, patience and encouragement. I would like to thank Dr. Robert Walker for his support. I really appreciate his care for me at difficult times.

I have a special thank for our exuberant, energetic, and companionate Sue Petti. When life gets tough, she has the words and smiles that can turn your life around. Of course, any job can not be done with perfection without Marcy Curtiss and Valorie Adkins. It was an honor to be working with the most talented and caring people in our department.

Last but not least, I extend many thanks to my colleagues and friends, especially Mazin

Assanie, Yasir Drabu, Manas Hardas, Oleg Komogortsev, Sajid Shaikh, Asrar Haque, and the entire Medianet laboratory at Kent State University for their support and compassion.

I thank God for having to know and interact with a few of the most caring people that I have seen in my life.

I left my family in another country for two and a half years to finish this Ph.D. During the last Summer break before graduation, I told my daughter, Ayah,”What do you think if I get so famous and I was asked to go in a space mission to the Moon.” She said “Are you going to leave us alone again.” I had no answer for her.

CHAPTER 1

INTRODUCTION

In the future, there are plans to launch long-distance exploratory probes, build space colonies on distant planets, and place satellites around planets other than Earth. Like the Mars Lander [39], these space technologies will gather data and beam it back to Earth by suitable means [61]. To facilitate these goals, there is a need to construct a routing infrastructure explicitly for a space routing domain that provides efficient routing decisions with low delay and high throughput.

Comparing with Earth-like network, one of the different aspects of routing in space is the mobility of the routers, which causes intermittent connectivity between them [41]. On earth, mobile-end-points are moving, while, in space, both end-points and routers are moving. Though there are several successful earth protocols for mobile transport (such as Mobile Internet Protocol (IP) [55]), there is yet a routing protocol that meets the full requirements of the space routing domain.

The primary attributes used to characterize any routing protocols and their functionalities are complexity, loop-free routing, convergence, storage overhead, computational overhead, and transmission overhead [57], [42]. In a topology network, where routers are continuously moving (e.g., routers on board spacecrafts), those parameters are especially important since fast convergence to a new route after a topology change insures quick delivery of the data [57].

1.1 Routing Protocol Functionalities

Space routing protocols have different functionalities which are posed by the following six questions:

1. Will the routing protocol use network state information?
2. Where is the routing table created?
3. How is the routing table created?
4. Will the routes have intermittent links?
5. Will the routing protocol support long delays?
6. Will the routing protocol use the predictable aspects of space objects?

The answers of these questions are in the sections below.

1.1.1 The Use of Network State Information

The use of network state information can be either static or adaptive. Static routing does not integrate the dynamic nature of the network while adaptive routing takes into account the changes in the network state [9]. The network state is made of:

- Information about the topology: Which router is connected to which router?
- Information about the traffic: How is the traffic load distributed among the network elements? What is the length of the interface queues in the router? etc [29]

Adaptive routing performs better than static because they react to the occurrence of network changes but at the price of an increased complexity of the algorithm.

Moreover, adaptive routing can be either isolated or non-isolated. Isolated routing are adaptive algorithms that base their routing decisions on local information such as the length of the queues in the router's output interfaces (e.g., Hot Potato routing [68]). Non-isolated algorithms take into account the status of the network at an extent ranging from the neighboring routers to the whole network. Most of today's routing algorithms in terrestrial networks are non-isolated [27].

Non-isolated algorithms are likely to be more efficient but the information about the network state must be distributed and gathered. While using such information enables the computation of accurate routes, it consumes network resources. Therefore, one must find a proper balance between the routing accuracy and the cost of accessing the network state information [27].

1.1.2 Where Is the Routing Table Created

The creation of the routing table is centralized, *decentralized* or *distributed*. In a centralized approach, the routing tables for all routers are created statically at a central location, (e.g., Earth control center computes routing table for each router on board a spacecraft and send the routing table to the spacecraft). In a decentralized approach, each router calculates its routing table statically with no cooperation from others. In a distributed approach, a set of routers cooperate together in order to compute the routing table.

Centralized algorithms suffer from the "single point of failure" weakness. Decentralized algorithms solve the bottleneck problem but still feature a point of failure at the source. In a decentralized algorithm if a router fails, other routers do not become aware of this failure until a number of failed transmissions occur. Distributed algorithms avoid these two drawbacks by splitting the route computation among different equipments. But, distributed algorithms introduce two new problems: *routing loops* and *convergence period*.

Routing loops: *routing loops occur when a route includes a given router more than once.*

Messages going through loops uselessly consume network resources and create transmission overhead and computational overhead [27]. Loops are addressed in two ways. The first approach consists in designing algorithms that are loop free. Synchronization of the different routers prevents loop formation at the price of generating additional signaling traffic and increasing the time required by the algorithm to compute the route [12]. A second approach, referred to as optimistic, is to detect loops upon formation and then take the appropriate action. Detection is achieved by recording complete description of the routes. Loop detection impacts both the routing algorithm response time and the signaling load [27].

The convergence period: *The convergence period is the amount of time for a router to restore normal traffic handling after recovery from a network change.*

Because of the distributed computation between different routers and because of the transmission delay, the algorithm takes time to converge to the solution. If the convergence period is long, changes to the network state might occur before the routing table computation resulting from the last network change is completed.

1.1.3 How Is the Routing Table Created

Routing is to find a path between a source and a destination according to the users' requirements, namely: *Legacy*, *Type of Service* and *Quality of Service* [27].

Legacy type algorithms make use of a shortest path computation. Different shortest path algorithm computation use different types of optimization metrics, like number of hops or a measurement of the end-to-end delay. Bellman-Ford, Floyd-War shall, and Dijkstra algorithms are examples of shortest path algorithms [26], [66].

In a legacy type algorithm, an optimization metric (for example end-to-end delay) is not adequate for all types of services. Real-time applications are sensitive to delay while file transfers are only sensitive to communication reliability. Different service types require different optimization metrics [27].

These findings lead to *Type of Service* (ToS) routing. ToS in the Internet Protocol (IP) is defined as: *minimize delay*, *maximize throughput*, *maximize reliability*, *and minimize monetary cost* and *normal service* [36]. Each IP datagram is expected to be routed according to its destination address and ToS field. The ToS field value corresponds to a given optimization metric used by a shortest path computation. Legacy and ToS routing are described as *best effort* routing algorithms.

Moreover, best effort is not always what is best for the user. Considering a video stream in a ToS environment, each packet is tagged with a 'minimize delay' ToS. Because of variations of the network state and since the routing algorithm minimizes the delay, the packets will be delivered at a varying rate. Extra buffering is then required in order to deliver video frames at a constant rate [27]. *Quality of Service* (QoS) routing addresses this problem by using a set of parameters describing the route requirements where each parameter is specified with an interval of values. Considering the number of possibilities resulting from the combination of parameters, QoS routing is best used with on-demand routing [27].

1.1.4 Intermittent Paths

TCP/IP networks have been designed with a specific assumption that all routes in the routing table contain no intermittent paths [41].

Let the link between two routers x and y, be represented by (x, y). Let b_{xy} be the clock time that indicates the beginning of the time period where (x, y) is active. Finally, Let e_{xy} be the clock time that indicates the end of the time period where (x, y) is active. Then a path is considered intermittent (see Figure 1) if it contains three consecutive routers k, l, and m, such that:

$$e_{kl} < b_{lm} \qquad (1)$$

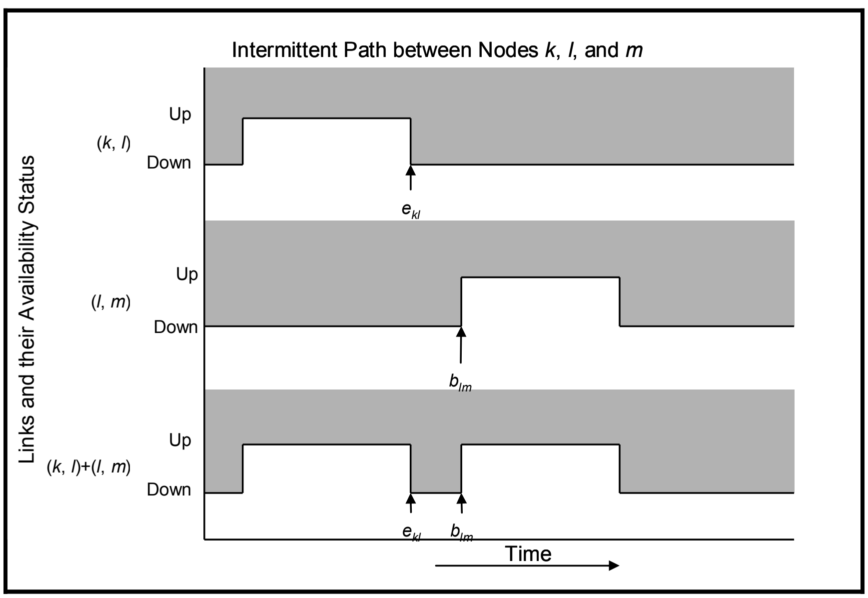

Figure 1: An intermittent path example

If there are a sufficient number of routers in space, there is no need for routes in the routing table to contain intermittent links. But until then, a space routing protocol must provide support for intermittent links. However, a routing protocol with intermittent links support has storage overhead because data traveling along an intermittent path has to be stored for long periods of time along its path route.

1.1.5 Long Delays

In space, propagation delay between a pair of routers may take hours which will cause many problems, like low throughput, loss of security synchronization, and longer convergence period [67],[70]. There are many ways to deal with a satellite link's long delay. One of them is to avoid using it for feedback or acknowledgment [71]. Another method is to minimize the occurrence of long delay transmissions. In essence, a space routing protocol has to provide a network topology structure or method to deal with long delays.

1.1.6 Predictable Mobility

All space objects are moving according to mathematical equations and their positions in space are calculated with extreme precision [25], [41]. It is advantageous to include predictability to a routing protocol because routes can be calculated ahead of time but at the cost of incurring computation overhead. In centralized routing, predicting the location of all routers is overwhelming worrying because of the single point of failure phenomena. Decentralize routing is a better solution but it reiterates the computational overhead at the router where energy limitation on board spacecrafts is an issue [30]. Distributive routing may be better than centralized and decentralized routing because the predictability computation is divided between routers.

1.2 Existing Routing Protocols

In this research, the existing routing protocols are surveyed in order to find how the functionalities of each routing protocol are capable of handling space traffic if placed in routers onboard spacecrafts. A number of routing protocols were developed over the years, although mainly for routers that have static locations. These routing protocols solves different routing domain problem ranging from internal gateway protocols (routing within an autonomous system) to external gateway protocols (routing between different autonomous system) [72].

1.2.1 Interior Gateway Protocol

Interior gateway protocol, referred to as IGP hereafter, are protocols which manage routing within an autonomous system, referred to as AS hereafter. An AS is a collection of networks that are connected to each other and can communicate with each other without resorting to routing outside the AS [72].

At the time of this research, there is no clear IGP leader, though Open Shortest Path First (OSPF) is seen as the recommend protocol. Routing Information Protocol (RIP) also has certain advantages. Intermediate System-to-Intermediate System (IS-IS) and Enhanced Interior Gateway Routing Protocol (EIGRP) are IGPs but both IS-IS and EIGRP are mainly a routing protocol for Open System Interconnection (OSI) model and a Cisco[1] proprietary routing protocol respectively. For those reason, any modification or enhancing to those protocol is limited and will not be discussed here. Other older IGPs such as *hello* and *gated* have been abandoned due to being technically obsolete [10].

[1] Cisco is a supplier of networking equipment & network management for the Internet

1.2.1.1 Routing Information Protocol (RIP)

The Routing Information Protocol (RIP) is an adaptive routing protocol that uses collected information from all other RIP routers to make a routing decision which makes RIP a non-isolated routing protocol [46],[45]. RIP operates by having each RIP router periodically broadcast the distance (the hop count) from itself to other RIP routers. RIP routers use the information broadcasted by other RIP routers to build their routing tables [72]. RIP is a decentralized routing protocol which uses Bellman-Ford shortest path routing algorithm to compute routing tables. At the time of this research, neither "intermittent links", "long delays", nor "predictable mobility" are supported in RIP standard. Moreover, RIP is not fit for space routing primarily because of two reasons:

1. RIP can lead to a long *convergence period* (known as "counting to infinity" problem in many RIP documents) in which the network detects a broken link by successively incrementing the distance metric until a value of "infinite" is reached [10]. Setting the "infinite" to a smaller value may improve this problem but will cause slow network convergence and limits the use of the distance metric hop where, in space, satellite energy and long delays have to be taken in consideration.
2. RIP uses a fixed metric (hop count) which it is not appropriate for situations where routes need to be chosen based on real-time parameters such as measured delay [45] which is a major concern in space routing since routers are always moving.

1.2.1.2 Open Shortest Path First (OSPF)

The Open Shortest Path First (OSPF) is also an adaptive non-isolated routing protocol although OSPF uses the state of links (not hop count as in RIP) between routers for routing decisions. Although the process of maintaining up to date link state information is complex, OSPF is similar to RIP in which both routing protocols use broadcast to distribute the information of the network.

OSPF has a messaging technique which enables the routing domain to converge after a network topology changes without any loops or any limitation on the metrics. OSPF is a distributed routing protocol where each OSPF router exchanges messages describing the links to other routers with its neighboring routers[2]. Those messages make the Link State Database (LSD) of the AS. Within one AS, all OSPF routers maintain identical LSDs.

Moreover, those messages provide costs of all links between a pair of routers. This OSPF cost is mainly a delay which is bandwidth dependant. With the collected link costs, OSPF uses Dijkstra routing algorithm to compute paths in the routing table.

OSPF divides the routing domain into areas where information in one area is hidden to the outside of its border. An OSPF routers sitting at the edge of the area (called area border routers) summarizes its area's routing information (IP addresses subnets in the area) and propagates it outside its area. All area border routers belong to the backbone which means that the routing information propagated by an area border router is disseminated to all other area border routers. In turn, each area border router propagates routing information of other areas into its own area.

Mainly, the OSPF protocol can be divided into three different protocols 1) The "Hello" protocol maintains communication with neighbors, 2) The "Exchange" protocol is used to synchronize database between two neighboring routers, and 3) The "Flooding" protocol is

[2] Neighboring routers are routers that have interfaces to a common network

used to propagate changes in link state to other routers in the network. [72].

Although OSPF may be robust enough for space routing, OSPF in its current form does not resolves a number of issues in space routing which are:

1. The OSPF area division mechanism is not fit for space routing. It may be feasible to divide the space routing domain into areas by proximity of planets. This type of area division mechanism will not scale well because of the following:

    - All networks that exist on one planet make one area. Since OSPF is a chatty protocol [2], as the number of routers belonging to one area (for example, satellites orbiting Earth), routers will handle more OSPF proprietary control packet than any other traffic.

    In space, a new way of dividing the space routing domain is required which can scale well and provides the same level of information hiding.

2. OSPF routers change position which, in turn, changes their propagation delay between each other. As a result, link cost can vary by time which will cause OSPF to produce inaccurate shortest path routes.

3. When a pair of routers is occluded by a celestial object, the connection is terminated. When the same pair is visible to each other, the connection establishing is restarted from the beginning where both routers have their databases exchanged. This is a waste of time since their databases may be still unchanged.

1.2.2 Exterior Gateway Protocols

Exterior Gateway Protocols, referred to EGP hereafter, are protocols for routing packet between autonomous systems. They view autonomous systems as “black boxes” with well-defined entry and exit-points, and attempt to route traffic between these points as efficiently as possible [72].

Older protocols such as Gateway-Gateway Protocol (GGP) have largely been succeeded by a newer protocol known as Border Gateway Protocol (BGP).

1.2.2.1 Border Gateway Protocol (BGP)

At the time of this research, Border Gateway Protocol, referred to BGP hereafter, is the Internet EGP standard [10], [58]. BGP is an adaptive non-isolated routing protocol (like OSPF and RIP). Also, BGP is distributive routing protocol where BGP routers announce full paths between two sites, allowing the implementation arbitrary routing policies with full loop detection [72].

Initially, a pair of peer routers (wishing to communicate with BGP) establishes a handshake (bidirectional communication), in which identification and protocol information are exchanged. If the authentication succeeds, the routers will start exchanging “update” packets to bring each other up to date. Changes are propagated to other neighboring connected routers. After these exchanges, traffic is limited to “keep alive” messages and further “update” messages if the network topology changes.

The simple form of BGP routing algorithm is that the router would choose the route with the minimum path length, with some arbitrary way to break ties between routes with the same path length, although, several additional attributes maybe altered manually to change the outcomes of the routing table [13].

BGP is not fit for space routing because of the following:

1. BGP doesn't support long delays since it routing algorithm is based on the cost provided by router who may not support long delays.
2. Connections between peer routers are not maintained periodically which makes routing decision uncertain.

1.3 Related Space Routing Research

There are a vast number of research papers for routing in space. Space routing research tackles a number of heuristics which require constellation satellites orbiting Earth and in deep space. In this research, a number of routing algorithm heuristics are surveyed in adaptive routing [15],[29],[40],[49] , dynamic routing [14],[27], distributed routing [35], static routing [14],[25],[27],[29],[38], and QoS Routing is [23],[28],[33],[38],[43].

To our knowledge, only three routing protocols encompass a total solution for routing in space: ASCoT architecture, and space time routing frame, and interrogation-based relay routing protocol,

1.3.1 ASCoT Architecture

In [34], they are presenting a complete structure for routing in space. The National Aeronautics and Space Administration (NASA) Autonomous Space Communications Technology (ASCoT) project is a routing and scheduling structure for flexible tasking and coordination among space assets. ASCoT depends on underlying systems which provide a variety of information and services to the ASCoT middleware which includes:

- Navigation information – characteristics of the links, positions of routers (current and expected), bandwidth, reliability, latency, etc.
- Local status – power, health, load of transmission queues, etc.
- Ability to send and receive messages

ASCoT uses a protocol called Positional Link-trajectory State (PLS [54]) routing to compute paths to the destination routers. PLS is an adaptation of Link State Routing (LSR [63]) to the context of space networks. The basic idea behind ASCoT is that link trajectories, together with link attributes (such as latency, data rate and error characteristics), are disseminated throughout the network. Using this information, each router can then independently compute a path to a given location by computing a shortest path tree (using a modified Dijkstra's algorithm) spanning the network. The next hop for a message is then determined by looking up the next edge leading to the destination in this spanning tree. As long as changes in neighborhood information is infrequent, and the total number of routers (and links) participating in link state routing is small, the protocol is known to work efficiently [34]. The path computation can also take into account link characteristics in order to satisfy any Quality of Service requirements that a particular message might have.

Although, the ASCoT is the closest in comparison with the proposed research, in the proposed research all the information and services are built within the protocol. Furthermore, the proposed research provides support for intermittent path but ASCoT does not.

1.3.2 Space Time Routing Framework

In [47], they are proposing a new space-time routing framework for networks leveraging the predictability in router motion. They construct space-time routing tables where the next hop is selected from the current as well as the possible future neighbors. In the space-time routing tables, they use both the destination and the arrival time of message to determine the next hop. They devise an algorithm to compute these space-time routing tables to minimize the end-to-end message delivery delay. Their routing algorithm is based on a space-time graph model derived from the mobility of routers. In [47] , the number of hops plays a pivotal role since they assume that a packet of size of *m* bytes requires equal propagation times on all links. But, this assumption may only be valid when all routers are close to each other. When the routers are orbiting multiple planets, the routing decisions produced by this algorithm are no longer guaranteed to be the path with the least end-to-end delay.

The predictably model for intermittent connections, which is used in [47], is similar to the proposed research. Although this protocol is predictable but the research is based on hop count more the propagation delay. In [47], all hops are assumed to be equal for the same message size which is troublesome if the compared hops are toward different planets.

1.3.3 Interrogation-Based Relay Routing Protocol

In [62], they introduce an Interrogation-Based Relay Routing protocol (IBRR). IBRR is a relay-based routing protocol where the source of data does not assume a complete path from source to destination. In other words, data packets are passed from one hop to another, each hop getting the message closer to the destination until the message reaches the destination. In IBRR, a next hop may not be available immediately for the current router to forward the data. The router will need to buffer the data until the router gets an opportunity to forward the data. IBRR is essentially an on-the-fly store and forward technique. The routers must be capable of buffering the data temporarily for possibly long periods to cater to the situation when there is no connectivity. IBRR uses an optimistic technique to forward messages from one router to another. The use of optimistic forwarding ensures that a message is conveyed eventually, and not dropped at any intermediate router.

In IBBR, the message may be stored for a very long time, before it can reach its destination. This limitation causes router to be resource constrained. Also, the messages with time limitation (for example a rocket is heading to Earth is detected) have to rely on luck to be delivered before their deadlines.

1.4 Why OSPF

From the start, OSPF's semantics showed an appealing outlook for the proposed routing protocol. The reasons why OSPF is chosen to be the building ground for the proposed space routing protocol is as follow:

- *OSPF's ability to divide a routing domain into areas within one AS*

  In space, the concept of areas and their information hiding of one area to another were very appealing.

  The idea of area border routers summarizing information of its area and flooded into the backbone showed that flooding can be controlled.

One AS for all routers gives us the ability to contain all structure under one set of rules.

- *Each OSPF router dynamically maintains identical view of the AS topology.*

  All routers know about each other's location.

- *All neighboring routers exchange hello message to maintain connectivity.*

  Maintaining a connection-oriented environment ensures the accuracy of the routing table.

- *When a change occurs in the network, it is detected in finite time.*

  The ability to detect changes in a volatile environment like space and propagated the changes to the affected areas is a tool to stabilize the network in finite time.

- *Master/slave semantic ensures a smooth transition to end database exchanges.*

  Traffic in space is very expensive and the master/slave relationship in the exchange protocol ensures that there is no extra transmission overhead

- *Flooding scope semantic ensure loop/free environment.*

  A loop/free is a required feature because of resources costly overhead.

1.5 Dissertation Organizations

The central focus of this dissertation is on the design and evaluation of a routing protocol for space. Furthermore, this dissertation is structured as follow:

CHAPTER 2: OSPFv3

In this dissertation, Open Shortest Path First version three (OSPFv3) routing protocol is explored in moderate details. Some of the semantics of OSPFv3 are explained in order for the reader to follow the stream of thought of the space routing protocol which will be presented in CHAPTER 3. The weak aspects of OSPF are identified in a space environment which will be modified in the space routing protocol.

CHAPTER 3: SPACE OSPF

The proposed routing protocol for space, Space OSPF (SOSPF) is presented here [4], [6] . This chapter will explain the critical features of SOSPF which are area hierarchal structure, neighbor data structure, predictability, link advertisement, Hello protocol and Database Exchange protocol. A detailed example is presented at the end of this chapter which explores most of the functionalities of the SOSPF routing protocol.

CHAPTER 4: SDIP ROUTING ALGORITHM

The proposed routing algorithm which can compute routes with intermittent links is presented here [5], [7]. An example is presented at the end of the chapter.

CHAPTER 5: - PERFORMANCE ANALYSIS

In this research, the effects of changes in the network state which cause flooding are evaluated. In this chapter, research shows how the SOSPF area structure maintains the level of flooding. In addition, stability and scalability are presented with the effect of changes in the network.

CHAPTER 6: SIMULATION PERFORMANCE

The performance of a number of simulations with respect to the stability and scalability are presented here. Real satellite configurations are simulated and the end-to-end delay is shown with comparison with other related routing algorithms.

CHAPTER 2

OSPFv3

Open Shortest Path First version three (OSPFv3) is a routing protocol which bases its routing decision on the states of the links between routers. OSPF runs internal to a single autonomous system (AS) where all OSPF routers maintain identical databases describing the AS's topology, called Link State Database (LSD). The consistency of having identical LSDs between routers is achieved by having routers form neighboring relationship with each other. A pair of neighboring (either directly connected to each other or via a network like Ethernet) routers 1) exchanges their LSDs when they first become bi-directional connected (or adjacent), and 2) thereafter, updates each other when their LSDs get updated.

From this LSD, a Shortest Path First (SPF) tree is constructed, and then the routing table is calculated from the SPF tree. OSPF recalculates routes quality when a topological change occurs [18], [51]. Moreover, OSPF provides area routing capabilities where the AS is divided into areas to reduce routing traffic and add increased routing protection. Also, OSPF includes security measures to avoid intruding from malicious intruders [37].

In brief, aspects of OSPF which correlate with the proceeding chapter are discussed here. This chapter draws attention to how OSPF and the space routing protocol correlate and differ.

For further details on OSPF, [18] and [51] provide more technical details.

2.1 Network Types

There are four network types in OSPF which are

- *Point to Point*: A network that joins a pair of routers
- *Broadcast*: A network supporting more than two routers with broadcast capabilities
- *Non-Broadcast-Multi-Access* (*NBMA*): A network supporting more than two routers with no broadcast capabilities, rather simulate operation of a broadcast network
- *Point to Multi Point*: A network supporting more than two routers with no broadcast capability which treat all connected routers as point to point networks

At the time of this research, the only network type for satellite communication is Point-to-Point [56] which is the only network type supported in the space routing protocol.

2.2 Neighbor Data Structure

An OSPF router converses with a neighboring OSPF router using the policies defined in a "Neighbor Data Structure" which both routers agree to. Each conversation is identified by the neighbor's identity.

The neighbor data structure contains all the information pertinent to the forming of adjacency between a pair of neighboring routers. The neighbor data structure comprises:

- State: the functional level of the neighbor conversation.
- Inactivity Timer: A single shot timer whose firing indicates that no Hello Packet (see

section 2.7) has been received from the neighbor recently.

- Master/Slave bit: When two neighboring routers are exchanging LSDs, they form a Master/Slave relationship.
- Database Description Sequence Number: This is the number of the LSD's description packet that is currently being sent to the neighboring router.
- Last Received Database Description packet: Information about the last LSD's description packet.
- Neighbor ID: the identity of the neighboring router.
- Neighbor Priority: the priority of the neighboring router.
- Database Summary List: The complete list of link state descriptions that make up the router's LSD. Each entry in the LSD is summary of a link state description, called Link State Advertisements (LSAs).
- Link State Request List: The list of LSAs that need to be received from this neighbor in order to synchronize the neighbors' LSD with the router's LSD.

2.2.1 The Neighbor State

The OSPF neighbor state diagram for point-to-point networks, shown in Figure 2, shows the state progression to establish adjacency between two neighbors. The neighbor states and their associated events are:

- *Down*: No recent information has been received from the neighbor.
- *Init*: When router *S* is started, *S* sends a hello packet (see section 2.7) to a neighbor

router *R*. When *R* receives the hello packet and sees that its router's identity is not in *S*'s hello packet, *R* issues a *Hello Received* event and move into *Init* state.

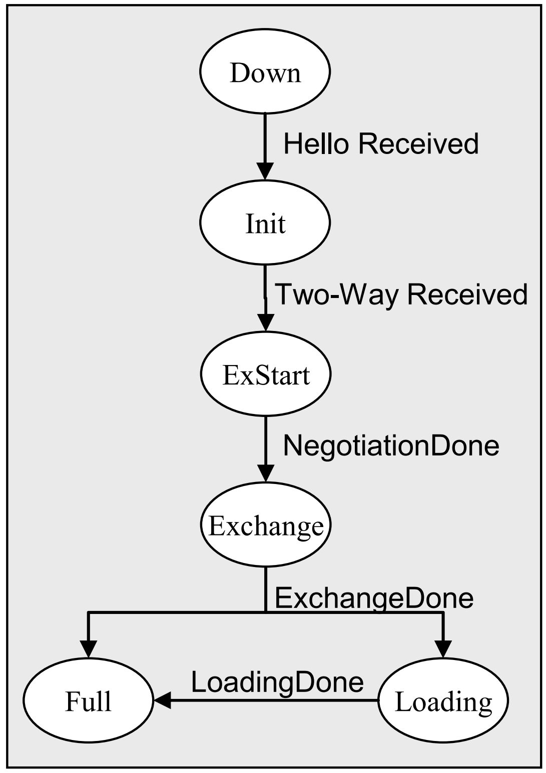

Figure 2: The neighbor state diagram

- *ExStart*: When a router receives a hello packet and sees its router's identity in the hello packet, it issues a *Two-Way Received* event and moves into *ExStart*.

- *Exchange*: When both routers decide on who will be the master and who will be the slave, both routers issue *NegotiationDone* events and move into *Exchange* state where they start exchanging packets describing their LSDs. As description packets arrive, LSAs that are not in the receiving router's LSD are inserted in the Link State Request list.

- *Loading*: When both routers finish describing their LSDs, both routers issue

ExchangeDone events. The router that has an empty Link State Request list moves to *Full* whereas the router that has one or more entries in its Link State Request list moves to *Loading* state. Any router that is in a *Loading* state sends request packets of LSAs located in its Link State Request list. In turn, when a request packet is received, the LSAs in the request packet are sent to the neighbor.

- *Full*: The router enters the *Full* state if its LSD is synchronized with its neighbor either as a result of a *Loading Done* event or an *Exchange Done* event.

Table 1: Events of retreating to a lower state

Current State	Cause Event	Future State
Loading	A request is received for an LSA that is not in the router's LSD	*ExStart*
Exchange or higher	A description packet is received which contains wrong information (e.g., unexpected Database Description Sequence Number)	*ExStart*
ExStart or higher	a Hello packet is received which does not contains the receiving router's identity	*Down*
Any	No hello packets have been received recently	*Down*
Any	Lower Level Protocol or a network manager downed the neighbor	*Down*

There are a number of events which cause a router to move back to lower state. Those cause events are described in Table 1.

The proposed space routing protocol uses the same terminology of OSPF's neighbor data structure where the state progression remains the same. Nevertheless, the proposed routing protocol adds one more state, called *Sleeping state*, which reflects on how spacecrafts lose sight of each other due to occlusion by celestial objects or simply by configuration. In addition, there are a few more events added to the space routing protocol which correlates with the *Sleeping state*.

2.3 Area

OSPF allow a collection of contiguous router, networks, and hosts to be grouped together forming an area where each area is responsible for creating its own OSPF environment. All routers within one area have identical link state database. With area in mind, routing takes two forms, depending on whether the source and destination of the packet resides in the same area or not.

- Intra-Area routing: if the source and destination reside in the same area
- Inter-Area routing: if the source and destination reside in different areas

Each area has one or more area border router that act as forwarding agent for all routers within its area. . In OSPF, all area border routers are connected together in one area, called the backbone area. Area border routers forward their areas' routing information into the backbone and into the other areas through their area border routers.

The backbone must be contiguous; however, it needs not to be physically contiguous. If a router, which is not physically connected to the backbone, needs to be part of the backbone, it has to be done through a virtual link. A virtual link can be configured between any two backbone routers that have an interface to a common area.

The concept of area in the space routing protocol is completely redefined but the information hiding strategies will remain.

2.4 Router Types

There are six router types in OSPF which are

- *Internal router* is a router with all directly connected networks belonging to the same area.

- *Area border router* is a router that is attached more than one area
- *Backbone router* is a router that has an interface to the backbone area
- *AS boundary router* (*ASBR*) is a router that exchanges routing information with routers belonging to other AS
- *Designated router* is a router that is responsible for advertising network connections to all the routers attached to a common network. The Designated router is elected.
- *Backup Designated router* becomes the designated router when the current designated router fails. The Backup Designated router is also elected

In the proposed space routing protocol, only area border routers and ASBRs are inherited from OSPFv3. The semantics of ASBRs remain the same but the area border routers are different.

2.5 Advertisement

All OSPF routers within an area must have identical view of the network. This is achieved exchanging messages which contain the costs of using the routers' interfaces. Those messages are referred to as Link State Advertisements (LSAs). Depending on the router's type, various types of LSAs are sent out the router's interfaces. There are seven LSA types in OSPF in which each LSA has a certain role in the protocol.

None of these LSAs will be used in the proposed routing protocol for space rather the proposed routing protocol introduces two new LSAs which provide the variation of delay costs, the times of direct view between routers in space, and the router's area membership (see section 3.4).

2.6 Flooding Scope

Each advertisement must be propagated to other routers. The rules for those propagations are called flooding scope. There are three types of flooding scopes which are:

- Link-Local-Scope: LSAs are flooded only on local link and no further
- Area-Scope: LSAs are flooded throughout the area where the LSA was originated
- AS-Scope: LSAs are flooded throughout the entire routing domain

Although the concept of flooding in exists in the proposed space routing protocol, the new flooding scopes' roles are completely different.

2.7 Hello Protocol

The Hello protocol is responsible for establishing and maintaining neighbor relationships between a pair of routers by exchanging hello packets at regular intervals. The hello packet contains:

- The packet header which contains the router's identity
- The router's priority
- A list of routers' identities who have been seen recently via the interface which the hello packet was transmitted from

Every OSPF router has a list of routers' identities that have been seen through interfaces which is called the *Neighboring Routers list*. There is a *Neighboring Routers list* for each router interface and is sent with every hello packet propagated through this interface.

When router *s* is started, hello packets are sent to all neighboring routers. The initial hello packet, that is sent to router *r* via interface *i*, will not contain *r*'s identity in the *Neighboring Routers list*. When *r* receives (via interface *j*) the hello packet and sees that it is not in the *Neighboring Routers list*, r does the following

1. Transition *s*'s neighbor state to *Init* (see section 2.2.1),
2. Adds *s*'s identity to the *r*'s *Neighboring Routers list*, and
3. Send a hello packet to *s*.

When *s* receives the hello packets and sees that it is in the *Neighboring Routers list*. S does the following:

1. Transition *r*'s neighbor state to *ExStart* (see section 2.2.1),
2. Adds *r*'s identity to the *s*'s *Neighboring Routers list*, and
3. Send a hello packet to *r*.

Finally, *r* receives the hello packet and *s*'s neighbor state is transitioned to *ExStart* state. Thereafter, hello packets are exchanged at regular intervals to maintain connectivity.

In the proposed space routing protocol, the mechanism of the Hello protocol remain the same; but the followings have been implemented:

- The content of the hello packets is changed
- The content of the neighboring router list is changed
- A new Space Object list is added

2.8 Exchange Protocol

In OSPF, when a pair of routers becomes bidirectional connected, the database describing the AS is synchronized. Synchronization is maintained via flooding where each router describes its entire database into a number of packets, called Database Description packets. Each Database Description packet contains a number of LSAs' descriptions and not the detail of the LSAs. After receiving Database Description packet, all LSAs (found in the Database Description packet) that do not exist at one router's database are requested in a packet called Link State Request packet, Once a Link State Request packet is received, the requested LSAs are sent in multiple packets called Link State Update packets where most of these packets are acknowledged.

In the proposed space routing protocol, the exchange protocol is enhanced to facilitate the need to exchange space parameters of spacecrafts and other celestial objects.

CHAPTER 3

SPACE OSPF

The solar system encompasses the Sun and the set of celestial objects gravitationally bound to it. Those celestial objects are planets and their moons and thousands of small bodies which consist of asteroids, meteoroids, comets, and interplanetary dust.

In the future, there are plans to network the whole solar system and beyond. Consequently, there are a number of studies and suggestions that will make the solar system network-ready which are:

- Establishing colonies on Mars, moons, comets, and asteroids [1],[17],[59],[64],
- Positioning satellite around planets other than Earth(like Mars and Mercury) [32],
- Positioning space stations in deep space [52], [53],
- Positioning satellites at a number of LaGrange points [24],[31],
- Have space shuttles travel safely into deep space [3],[53], and
- Positioning satellites in the Asteroid Belt [20].

Figure 3 illustrates a sample view of the solar system in the future.

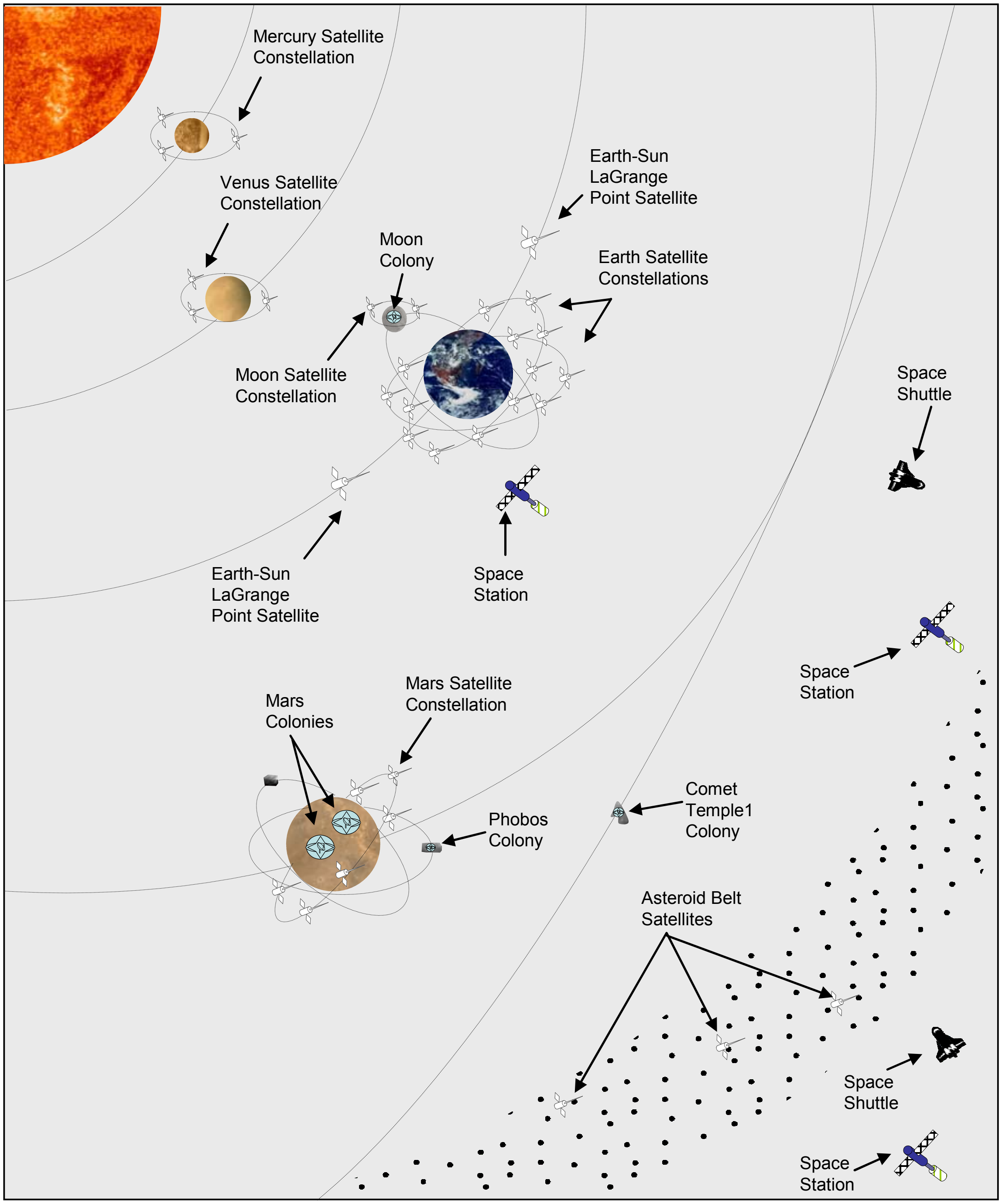

Figure 3: Future view of the solar system

3.1 SOSPF Hierarchical Area Structure

In order to network the solar system, there must be a network infrastructure which enables such task. Consequently, the solar system is divided into a hierarchical structure by nature. At the time of this research and to our knowledge, there is no space routing protocol that takes advantage of the hierarchical structure of the solar system.

The proposed space routing protocol, referred to as SOSPF hereafter, is taking advantage of this hierarchical structure by placing all celestial objects in a hierarchical area structure. Moreover, SOSPF categorizes objects in space (natural and artificial) into three area classes which are:

1. *Satellite constellation area* is a set of one or more satellites that share a common orbit. This class includes satellites formations, apace stations, and space shuttles.

2. *Colony area* is a network that is placed on the surface of a celestial object. At the time of this research, colony area is not part of SOSPF.

3. *Celestial object area* is a set of representatives of other areas which is formed for a particular celestial object (e.g., Earth, Earth-Moon, or Mars). For a celestial object area *A* that is formed for a celestial object *C*, the members of *A* are defined as follows:

    a) A satellite constellation area whose members are orbiting *C*, must have one or more of its members to belong to *A*. (see Area Border Routers in section 3.1.1)

    b) Members of a satellite constellation area *S* may be configured to belong *A* (e.g., a space shuttle) even if the members of S area not gravitationally bound to *C*.

c) One or more members of a colony area, which is formed on a colony that is located on the surface of *C*, are configured to be part of *A*.

d) Finally, other celestial objects, which have areas of their own, which are gravitationally bound to *C* must configure one or more members of their areas to belong to *A*. In this research, those celestial object areas are referred to as children of *A* and *A* is the parent of those areas as well.

Figure 4 illustrates the SOSPF hierarchical area structure which can be seen as a recursive structure. In other words, a celestial object area may contain other celestial object areas.

In this research, there is one celestial object area that is placed at the top the area hierarchy (i.e., it has no parent area). This area is referred to as the *backbone area*. An example of a backbone area is the Sun celestial object area in the solar system.

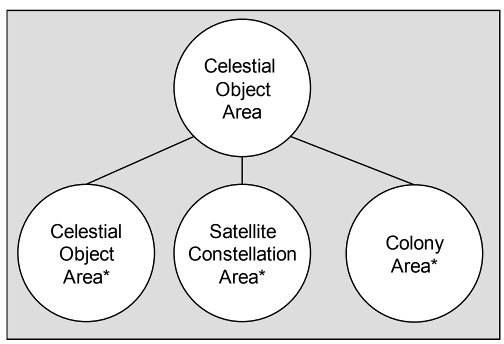

Figure 4: SOSPF hierarchical area structure

3.1.1 Area Border Routers

The purpose of areas in SOSPF is to maintain a level of information hiding and minimize protocol proprietary traffic, as defined in OSPFv3 [18], [51]. In turn, SOSPF traffic within one area does not affect the performance of other areas. Nevertheless, a summary of routing information (e.g., propagation delays between area members and network Internet

Protocol version six (IPv6) prefixes) must be propagated outside the area.

The inherited concept of “area border router” from OSPFv3 is still maintained. In OSPFv3, router *x*, which belongs to area *a*, is configured to be an area border router that summarizes the routing information of *a* and propagates it into the backbone. Moreover, when area border routers receive this routing information, they propagate it into the areas which they belong to. At the end within the AS, all routers have identical view of the AS’s routing domain.

In SOSPF, an area border router is a router on board a spacecraft (referred to as SOSPF router hereafter) which belongs to area *A*, summarizes the routing information of *A*, and propagates it to its parent area. Although area border routers for a satellite constellation areas and celestial object areas has the same functionalities, but they differ in their role in the area hierarchy.

3.1.1.1 Satellite Constellation Border Routers

Since SOSPF areas are hierarchically structured, every satellite constellation area has at least one SOSPF router which summarizes the routing information of the area it belongs to and propagates it into it its parent area (a celestial object area). An SOSPF router that is configured for this task is called a satellite constellation border router (SCBR).

Furthermore, the SOSPF routing protocol provides the ability for an SOSPF router to be hidden from the routing domain outside its area. Thus, an SCBR only propagates routing information of SOSPF routers who have their *DoNotAdvertise* bit set off (see section 3.2).

An example of an Earth-Moon celestial object area is shown in Figure 5 where there are two Earth-Moon satellite constellation areas, namely area 1:3:1:1 (description of the area numbering terminology is in B.1), area 1:3:1:2 and Space Shuttle 1 which is configured to be part of the Earth-Moon area. Moreover in Figure 6, the area hierarchy of the given example is shown where the Earth-Moon celestial object area is referred to as area 1:3:1.

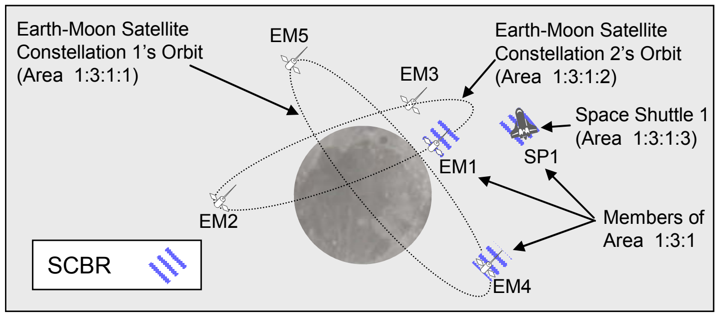

Figure 5: An example of Earth-Moon area

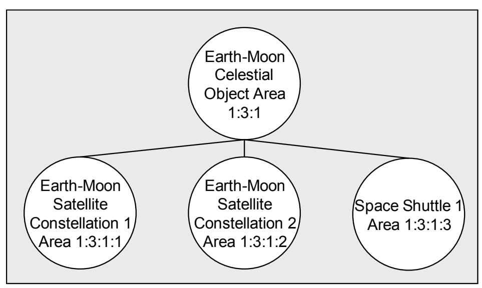

Figure 6: The Earth-Moon area example's area hierarchy

In Table 2, area 1:3:1 (the Earth-Moon area) has three members, EM1, EM4, and SP1. EM1 and EM4 summarize the routing information of area 1:3:1:1 and 1:3:1:2, respectively, and propagate them to each other and to SP1. Consequently, SP1 summarizes its routing information and propagates to EM1 and EM4. In turn, EM1 propagate the received routing information to the members of its area (area 1:3:1:1), which are EM2 and EM3. Consequently EM4 does the same for its area (area 1:3:1:2).

Table 2: The Earth-Moon area example's area memberships

Area	Members
1:3:1	EM1, EM4, and SP1
1:3:1:1	EM1, EM2, and EM3
1:3:1:2	EM4 and EM5
1:3:1:3	SP1

3.1.1.2 Celestial Object Border Routers

If celestial object area *A* is a child of another celestial object area *P*, at least one member (an SOSPF router) of *A* must be configured to summarize the routing information of *A* and propagates it into *P*. An SOSPF router that is configured for this task is called a celestial object border router (COBR)[3].

An example of an Earth celestial object area is shown in Figure 7 which is a follow up the example given in Figure 5. In Figure 7, two Earth satellite constellation areas (area 1:3:2 and area 1:3:3) and a new area for Space Shuttle 1 (area 1:3:4) are added. Moreover, Figure 8 shows the area hierarchy of the given example where the Earth celestial object area is referred to as area 1:3. Moreover, Table 3 shows the area membership the example.

Other than area border routers, SOSPF allows SOSPF routers to belong to multiple areas as seen in Figure 7 where Space Shuttle belongs to areas 1:3:4 and 1:3:1:3. Moreover, an SOSPF router that belongs to multiple areas must create the necessary data structure for each area it belongs to.

[3] Note that an COBR is also a SCBR

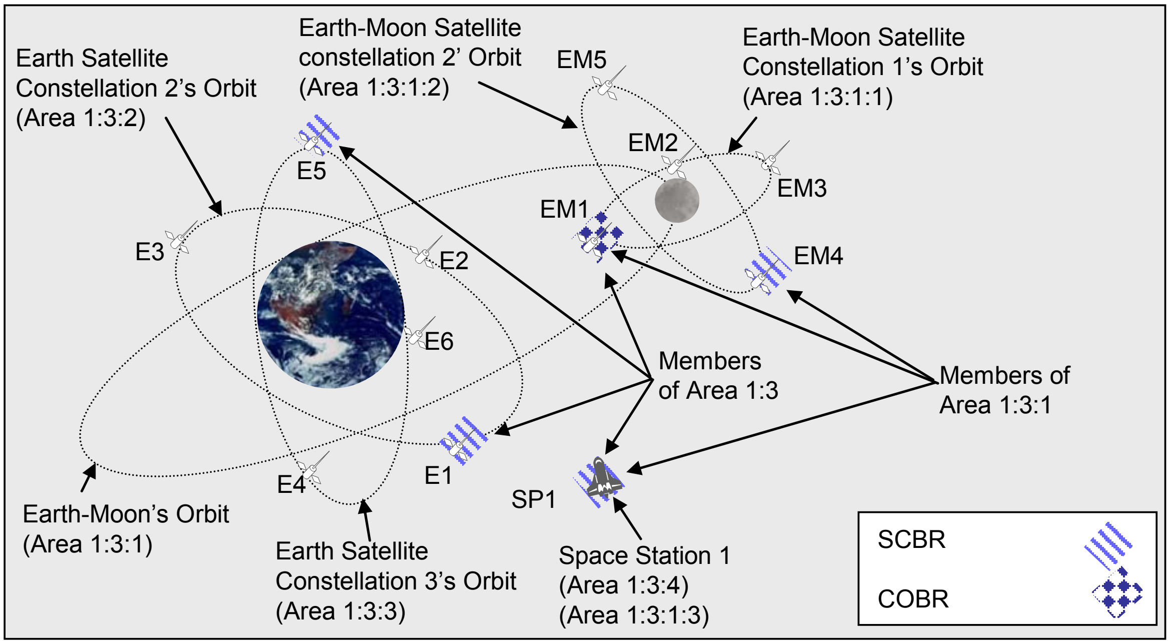

Figure 7: An example of an Earth celestial object area

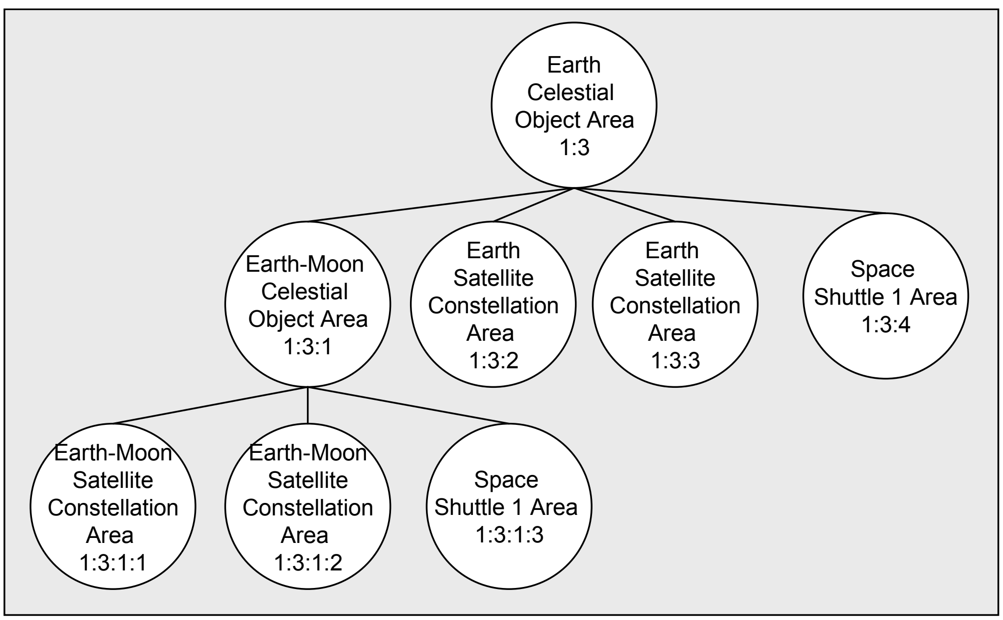

Figure 8: The Earth area example's area hierarchy

Looking at Table 3, EM1 is the COBR for area 1:3:1. Furthermore, EM1 belongs to three areas, 1:3:1:1, 1:3:1 and 1:3 which means that EM1 is configured to represent all SOSPF routers in the Earth-Moon celestial object area in the Earth celestial object area.

Table 3: The Earth area example's area memberships

Area	Members
1:3	E1, E5, SP1, and EM1
1:3:1	SP1 ,EM1 and EM4
1:3:2	E1, E2, and E3
1:3:3	E4, E5, and E6
1:3:4	SP1
1:3:1:1	EM1, EM2, and EM3
1:3:1:2	EM4 and EM5
1:3:1:3	SP1

When an SOSPF router belongs to multiple areas, it causes more SOSPF traffic cost which might deteriorate the overall performance of the network. Careful planning must be employed in area membership configuration. Otherwise, the network will suffer greatly.

The performance of the SOSPF routing protocol is further evaluated in CHAPTER 6.

3.2 SOSPF Neighbor Data Structure

For each pair of SOSPF routers that belong to a common area, a neighboring relationship is formed where a "Neighbor Data Structure" is created at each end. The neighbor data structure consists of all the required information to form an adjacency between a pair of SOSPF routers. Each neighbor data structure has the following parameters:

- *Neighboring Router's ID*: This the IPv6 address of the neighboring router.
- *Area's ID*: This is identity of the area which encompasses the router and the neighboring router.
- *HelloInterval*: This is the length of time, in seconds, between hello packets that the router sends to the neighbor.

- *Maximum Stability Period:* This is the period of time where the neighbor is checked for changes in visibility and distance.
- *Propagation List*: This is a list of propagation delays at different times to the neighboring router. The list consist of multiple entries where each entry contains the following:
  - *Birth Time*: The clock time when the neighboring router becomes in direct view.
  - *Propagation Period*: The time duration (in seconds) of direct view with the neighboring router without interruption
  - *Propagation Delay*: This is the propagation delay (in seconds) during the *Propagation Period*. This period reflects all delays incurred from delivering the maximum size packet to the neighbor (e.g., packet processing delay, packet flight time, transmission delay, and QoS delay)

  The *Birth Time* and *Propagation Period* are derived from the *Neighboring Routers list* (see section 3.2.2) and the *Propagation Delay* is derived from the Space Router-LSA (SR-LSA) whose *Destination Address* is the *Neighboring Router's ID* (see section 3.4)
- *New LSAs List*: This list contains LSA headers of LSAs that are received when the neighboring router is in *Sleeping* state (see section 3.2.1).
- *LSAs Threshold*: This is the maximum number of LSAs that can be held in the New LSAs List
- *DoNotAdvertise bit*: If set, the neighboring router wishes not to be advertised outside

this area which means that the LSAs associated with this neighboring router are not propagated outside this area

3.2.1 SOSPF Neighbor States

In addition to the OSPFv3 neighbor states defined in section 2.2.1, there is one extra state called *Sleeping* (see Figure 9). Both neighboring routers are transitioned to the *Sleeping state* once the direct view between them is occluded. In reality, this transition is based on the configuration of the *Neighboring Routers list* (see section 3.2.2). Then, the next state depends on one question which is:

1) How many changes have had happened in the network since the neighboring router has transitioned to *Sleeping* state?

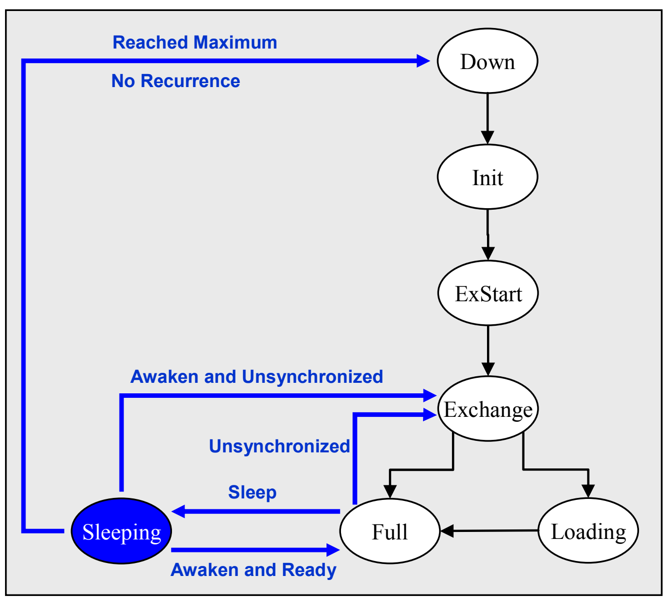

Figure 9: SOSPF State Diagram

The states are listed in the order of progressing functionality. For example, the *Down* state is listed first followed by a list of intermediate states before the final *Full* state. The *Full* state is achieved where the router and the neighboring router are in direct view and theirs link state databases (LSDs) are synchronized. In this dissertation, the reference of a state being lower or greater than another state depends on the progressing functionality between them. Aside from the *Sleeping* state, the progression of states is the same as in OSPFv3 (see section 2.2.1).

After a neighboring router is transitioned from *Full* to *Sleeping*, the router may receive LSAs from other SOSPF routers indicating changes in their LSDs (see section 3.4). These LSAs may have to be added to the *New LSA list* that is created for the neighboring router. Furthermore, the neighboring router remains in *Sleeping* state until one of the following events happen:

- The number of new/modified LSAs in the *New LSA list* has been reached *LSAs Threshold*
- The clock time of the next direct view to the neighboring router is reached
- The clock time of the next direct view to the neighboring router is unknown

The next neighboring router's state depends on the occurrence of one or more events which are shown in Table 4. The last entry in the table is about Area Membership-LSA (see section 3.4) which is explained in section 3.4.2.1.

At the time of this research, the only LSAs recorded in the *New LSA List* during *Sleeping* state are SOSPF LSAs which are Space Router-LSAs (SR-LSAs) and Area Membership-LSA (AM-LSA) (see section 3.4). Furthermore, only LSAs that are generated by SOSPF routers, and have their *DoNotAdvertise* bit set on, are inserted in the *New LSA List*.

Table 4: SOSPF new events

Event Name	Current State	Event Description	Future State
Sleep	*Full*	• The *Propagation Period* has expired indicating the neighboring SOSPF router is now out of sight.	*Sleeping*
No Recurrence	*Sleeping*	• The time clock for next direct view is unknown	*Down*
Awaken and Ready	*Sleeping*	• *Birth time* is indicating that the neighboring router is now in direct view • The *New LSA list* is empty.	*Full*
Awaken and Unsynchronized	*Sleeping*	• *Birth time* is indicating that the neighboring router is now in sight • The *New LSA list* is not empty.	*Exchange*
Reached Maximum	*Sleeping*	• Number of LSAs in the *New LSAs List* has reached the *LSAs Threshold*.	*Down*
Unsynchronized	*Full*	• After the transition from *Sleep* to *Full* state where the *New LSA list* is empty, but the neighboring router's *New LSA list* is not empty.	*Exchange*
Bad AM-LSA	*Any*	• An AM-LSA is received which contains incorrect area information	*Init*

An example scenario of SOSPF neighbor states, where SOSPF router *x*, *y*, and *z* are members of area *A* and *y*'s DoNotAdvertise bit set on, is as follow:

1. *Sleep* event causes *x* to be transitioned to *Sleeping* state and leave area *A* for *s* seconds.

2. *y* generate a new SR-LSA that changes its propagation delay to *z* and forwards it to *z*. *y* adds this LSA to the *New LSA List* in *x*'s neighbor state.

3. *z* receives the new SR-LSA but since *y*'s *DoNotAdvertise* bit set on, *z* does not forward the new LSA outside *A*. *z* adds this LSA to the *New LSA List* in *x*'s neighbor state.

4. When *s* seconds concludes, *Awaken and Ready* event is triggered in neighbor state data structure for *y* and *z* in the router *x*. This event causes *y* and *z* neighbor state located in *x*

to transition to *Full*.

5. *y* and *z* transition *x*'s neighbor state to *Exchange* after the triggering of *Awaken and Unsynchronized* event (because the *New LSA list* is not empty).

6. Thus, *y* and *z* issue a Database Description packet which contains the new SR-LSA and send it to *x*.

7. Once *x* receives the Database Description packet, it issues a *Unsynchronized* event and transition the neighbor state of *y* and *z* to *Exchange*.

8. *x* checks if the new LSA is in its LSD

9. *x* issues a Link State Request packet, sends it to *y* and *z*, and transitions the neighbor state of *y* and *z* to *Loading*.

10. Once *y* and *z* receive the Link State Request packet, both of them do the following:
    - They transition *x*'s neighbor state to *Loading*,
    - Flush *New-LSA list* , and
    - Send the new LSA in a Link State Update packet to *x*.

11. Once *x* receives the Link State Update packet, it 1) sends a Link State Acknowledgment packet to *y* and *z*, 2) transition the neighbor state of *y* and *z* to *Full* and 3) add the LSA *x*'s LSD. (the routing table may be recalculated)

12. When y and z receive the Link State Acknowledgment packet, they transition *x*'s neighbor state to *Full*.

3.2.2 Neighboring Routers in Areas

Not all members of a satellite constellation area have to have neighboring relationship with each other, but rather each SOSPF router forms neighboring relationships with at least two (if any) other SOSPF routers. The reason for not having all members of a satellite constellation area as neighboring routers is because it may occur that a few SOSPF routers are occluded by the planet they are orbiting.

In the other hand, all members of a celestial object area form neighboring relationship with each other which means that they have to have direct view of each other as well.

Thus, each area has a data structure, called *Neighboring Routers list*. This list contains a number of entries that define the neighboring routers of the area at different times. Each entry consists of the followings:

- *Neighboring Routers*: The SOSPF routers identity which have neighboring relation ship with the router
- *Neighboring Start Time*: The clock time where the SOSPF router (defined in the *Neighboring Routers list*) become neighbors
- *Neighboring Period*: The time duration of this neighboring relationship

For example, Figure 10 and Figure 11 depict areas 1:4:1, 1:4:2, and 1:4 where area 1:4:1 contains SOSPF router M1 through M6. Also, Figure 10 shows that M1 is occluded from M4 by Mars. Moreover, M2 and M6 are configured to be the neighboring router of M1 indefinitely. Thus, an example of the *Neighboring Routers list* for area 1:4 in M1 is shown in Table 5 where the *Neighboring Period* is set to *MaxMembershipPeriod* (see B.4). When the *Neighboring Period* expires, the *Neighboring Routers list* is recalculated.

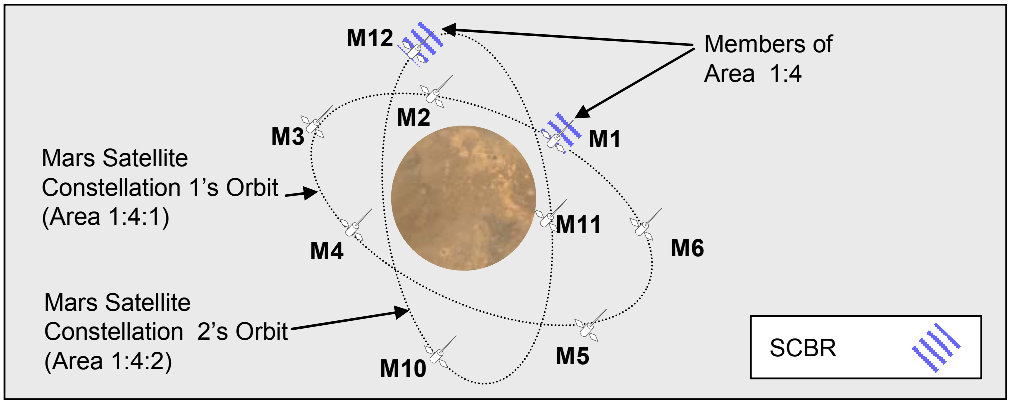

Figure 10: An Mars area example's

In addition in Figure 10, area 1:4 has two SOSPF routers M1 and m12. After some time, M1 and M12 may be occluded by Mars which indicates that one of the two areas (1:4:1 and 1:4:2) has to configure another SOSPF router to be the area border router (a SCBR) for it. Thus, an example *Neighboring Routers list* for area 1:4 in M1 is shown in Table 6. Similar to the previous example, when the *Neighboring Period* expires, the *Neighboring Routers list* is recalculated.

At the time of this research, the calculation of the *Neighboring Routers list* is exterior to the SOSPF routing protocol.

Table 5: The Neighboring Routers list for area 1:4:1 in M1

Number of entries		1
#1	Neighboring Routers	M2 and M6
	Neighboring Start Time	2010:12:00:00:00:00
	Neighboring Period	65535

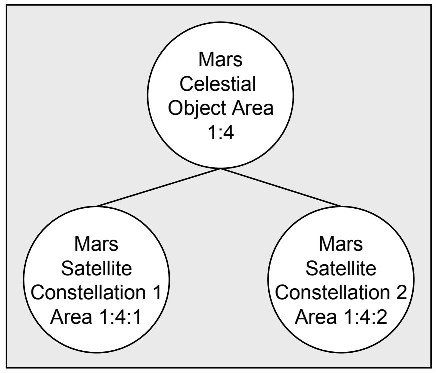

Figure 11: The Mars area example's area hierarchy

Table 6: The Neighboring Routers list for area 1:4 in M1

Number of entries		3
#1	Neighboring Routers	M12
	Neighboring Start Time	2010:12:00:00:00:00
	Neighboring Period	21845
#2	Neighboring Routers	M11
	Neighboring Start Time	2010:12:00:06:04:05
	Neighboring Period	21845
#3	Neighboring Routers	M10
	Neighboring Start Time	2010:12:00:12:08:10
	Neighboring Period	21845

3.3 Space Predictable Mobility

Techniques for the mobility management proposed for earth networks assumes that end devices are moving while the routing infrastructure is static. In space, the infrastructure itself is undergoing movement resulting in a continuous change of the communication topology.

Although celestial objects and spacecraft are continuously moving, most of their movement is predictable. All planets, moons, comets, asteroids, and fixed orbit satellites are bound to the six orbital parameters (see APPENDIX A) which means that their location can be calculated using those parameters.

Other objects, like space shuttles, space stations, and space probes have trajectories whose locations are determined by a mathematical model or a set of fixed points at different times. In either case, the location of a space object is readily available at any time.

3.3.1 SOSPF Predictable Model

SOSPF is a predictable routing protocol which means that the location of an SOSPF router must be easily calculated by all SOSPF routers. When an SOSPF router introduces itself to another SOSPF router, it uses a hello packet (see section 3.6) which contains a *calculating method tag* which corresponds to a calculating model that is used to retrieve the location of this SOSPF router.

The calculation of the location of an SOSPF router is not part of the SOSPF routing protocol, but the calculating model must be available when needed. In this research, the six orbital parameters are used as the bases of the proposed routing protocol yet those six orbital parameters can be substituted with any other parameters that have the same functionality.

3.3.2 SOSPF Unpredictable Events

The unpredictability of the nature of the solar system still exists where events (e.g., galactic cosmic rays and solar wind) can parallelize a satellite indefinitely [44],[12]. In SOSPF, bidirectional communication between two neighboring SOSPF routers is maintained at all times by exchanging hello packets (see section 2.7). Any disruption at an SOSPF router, which has bidirectional relationship with another SOSPF router, is detected when hello packet is not received within a configured waiting period. Once detected, the router will flood the failure to other SOSPF routers (see section 3.4.1.1). Furthermore, routing tables may have to be recalculated (see section 4.6.2).

3.4 SOSPF Advertisements

The SOSPF routing protocol supports all of the Link State Advertisements (LSAs) of the original OSPFv3 which can be found in [18] and [51]. In addition, SOSPF introduce two new types of LSAs, Space-Router LSA (SR-LSA) and Area-Membership LSA (AM-LSA). The SR-LSA contains the calculation of an SOSPF router's links to other SOSPF routers. And, the AM-LSA describes the areas which an SOSPF router belongs to.

3.4.1 Space Router SR-LSA

An SR-LSA describes the connection between a pair of SOSPF routers where one SOSPF router calculates the clock times of the direct view with another SOSPF router during a *Maximum Stability Period* (see section 3.2). It calculates the estimated propagation delay between them at those clock times.

Each SR-LSA contains the IPv6 address of the source router and destination router and a number of tuples which describes the link between them at different times. Each tuple consists of three fields which are:

- *Begin Time* is the clock time when the destination router becomes in direct view with the source router.
- *Connection Period* is the time period (in seconds) of the connection described in the tuple where the source router and the destination router have direct view of each other.

- *Propagation Delay* is the delay incurred from the connection described in this tuple between the source router and the destination router. This delay reflects all delays incurred from delivering the maximum size packet to the destination router which includes packet processing delay, packet flight time, transmission delay, QoS delay, and any other delay that contributes to it.

The number of tuples in an SR-LSA does not exceed *MaxTuple* (see B.4) and the *Connection Period* does not exceed the *Maximum Stability Period* (see B.4). An example of an SR-LSA is shown in Figure 12.

Source Satellite Address		5F00:0000:C001:0400::/56
Destination Satellite Address		5F00:0000:c001:2C00::/56
Number of tuples		3
Tuple # 1	Begin time	2006:08:28:20:14:50
	Connection Period	14400
	Propagation Delay	15
Tuple # 2	Begin time	2006:08:29:40:30:05
	Connection Period	2000
	Propagation Delay	20
Tuple # 3	Begin time	2006:08:30:23:14:50
	Connection Period	3000
	Propagation Delay	20

Figure 12: An example of an SR-LSA

3.4.1.1 Generating SR-LSA

An SR-LSA is generated every time an SOSPF router becomes aware of a change of any of its direct links to other SOSPF routers. There are six scenarios for a generation of an SR-LSA:

1. *An SOSPF router becomes operational*

When an SOSPF router is turned on, it calculates the time-intervals and the propagation delays for each interval between itself and all SOSPF routers listed in the *Space Routers List* (see B.2). Each time interval indicates a change in the link from the previous time interval. There are only two change which are:

- Both routers were occluded since the last time interval. And now, they are in direct view of each other
- The propagation delay is changed from the previous time interval

2. *An SOSPF router changes its Propagation Delay*

   If an SOSPF router changes its *Propagation Delay* to another SOSPF router, the affected SR-LSA is modified.

3. *An SOSPF router changes its six orbital space parameters*

   If the *calculating method tag* equals zero (see section 3.6), it indicates that the location of the SOSPF router is learned through the six orbital parameters (see APPENDIX A) which are stored in the *Space Routers List*. If an SOSPF router is to move from its orbit position to another location (e.g., same orbit or different orbit), it acts as if it was just turned on at the new location and perform the same procedure described in scenario 1 above.

4. *An SOSPF router changes its calculating method tag*

   If the SOSPF router's calculating method tag is changed (e.g., a space station change its trajectory), it acts as if it was just turned on at the new location and performs the same procedure described in scenario 1 above.

5. *A new SOSPF router is found*

   When a new SOSPF router *y* is inserted into the SOSPF routing domain, the new router sends hello packets to a preconfigured set of SOSPF routers (see section 3.6). When SOSPF router *x* receives the hello packet, it checks if the *y*'s ID is in *Space Routers List*. If the ID is found, it continues with the forming of neighboring relationship without issuing any SR-LSAs.

Otherwise, the router will do the following:

- *y*'s ID is inserted in the *Space Routers List*.
- If *y*'s *calculating method tag* (found in hello packet) equals zero, the six space parameters are also inserted in the *Space Routers List*.
- A new SR-LSA is generated whose destination is *y*'s IPv6 address. Then, *x* calculates the time intervals and the propagation delays for each interval between itself and *y*.

6. *A Link Failed*

After a pair of SOSPF routers, *x* and *y*, form neighboring relationship with each other, they maintain their relationship by exchanging hello packets at regular intervals. When *x* does not receive a hello packet from *y* within a configured waiting period, called *RouterDeadInterval* (see B.3), *x* assumes that the link that connects itself with *y* is down and does the following:

- A new SR-LSA is generated where current clock time is inserted in the *Begin Time* field
- A value of zero is inserted in the *Connection Period* field
- A configured value called *InfDelay* (see B.3) is inserted in the *Propagation Delay* field

An example of an SR-LSA of a failed link is shown in Figure 13.

After SOSPF router *x* generates an SR-LSA, x does the following:

- Stores the generated SR-LSAs in its link state database (LSD),
- Re/calculates the routing table if needed, and
- Forwards all SR-LSA to all SOSPF routers that are defined in the *Neighboring Routers list*'s entries whose time interval coincide with the current clock time except the LSAs whose source router or destination router has its *DoNotAdvertise* bit set on. These LSAs are only forwarded to SOSPF routers who are members of the same area as source router and destination router. Once a neighboring router receives the SR-LSA, it is flooded into the AS's routing domain according to the flooding procedure (see section 3.5)

Source Satellite Address		5F00:0000:C001:0400::/56
Destination Satellite Address		5F00:0000:c001:2C00::/56
Number of tuples		1
Tuple # 1	Begin time	2006:08:29:40:30:05
	Connection Period	0
	Propagation Delay	4294967295

Figure 13: An example of an SR-LSA of a failed link

3.4.2 Area-Membership LSA (AM-LSA)

Each SOSPF router must advertise the areas that it belongs to and for how long. This is done through the Area-Membership LSAs (AM-LSA)[4] where each SOSPF router advertise the areas that it will part of during the *Max Stability Period* (see section 3.2). The AM-LSA consist of source router's IPv6 address and a number of tuples where each tuple contains four fields:

- *Area ID* is the area ID of the this tuple

[4] Not to be confused with Group Membership-LSA [50]

- *Area Members* contains the list of SOSPF routers' IDs that belong to this area
- *Area Start Time* is the starting point in time for this area's definition
- *Area Period* is the time period of the validity of this tuple

AM-LSAs represent the entries of the *Neighboring Routers list* of an area or areas. An example of an AM-LSA for SOSPF router M1 is shown in Figure 14 which coincides with the first entry of the *Neighboring Routers list* shown in Table 6.

Source Satellite Address		5F00:0000:C001:0401::/56
Number of tuples		1
# 1	Area ID	1:4
	Area Members	M1. M12
	Area Start Time	2010:12:00:00:00:00
	Area Period	21845

Figure 14: An example of SOSPF router M1's AM-LSA for area 1:4:1

3.4.2.1 Generating AM-LSA

AM-LSAs can contain definitions of more than one area; it all depends on the setting of the router's *Multi-Areas LSA bit*. If the router's *Multi-Areas LSA bit* is set on, only one AM-LSA is configured for all areas defined in the *Neighboring Routers list*. Otherwise, Each area in the *Neighboring Routers list* has a separate AM-LSA with its own data structure. Moreover, an AM-LSA can only hold *MaxTuple* tuples. Thus, multiple AM-LSAs are generated when required.

There are eight scenarios for a generation of an AM-LSA:

1. *An SOSPF router becomes operational*

   When an SOSPF router is turned on, it forms a neighboring relationship with another SOSPF router and generates one or more AM-LSAs. Each SOSPF router has a

Neighboring Routers list for each area it belongs to. Thus, every area is represented with one AM-LSA. Although, if the SOSPF router's *Multi-Areas LSA bit* is set on, all areas definitions are inserted in one or more AM-LSA, depending on the number of produced tuples. The setting of *Area Start Time* and *Area Periods* of each tuple in the AM-LSA is defined in scenario 2.

The SOSPF routing protocol dose not check the validity of the *Neighboring Routers list*, yet the list must be kept accurate and up-to-date by a network administrator or any other automated procedure.

2. *The AM-Timer expires*

During the initial generation of AM-LSAs (in step 1 above), the SOSPF router sets a timer, called area membership timer (*AM-Timer*), for each area. This timer indicates the waiting time period prior the next generation of the area's AM-LSA. Let *i* be the area of the AM-LSA,

Let A_i be the data structure of area *i*, N_i be the first entry in *Neighboring Routers list* for area *i*, N_{i+1} be the second entry in *Neighboring Routers list* for area *i* (if available), and *ct* be the current time. Thereafter, the setting of the *AM-Timer* is as follow:

a) If N_i's *Area* S*tart Time* < *ct*

b) AM-LSA's *Area Start Time* = *ct*

c) If N_i's *Area Start Time* + N_i's *Area Period* – ct < *MaxMembershipPeriod*

d) AM-LSA's *Area Period* = N_i's *Area Start Time* +

N_i's *Area Period* – *ct*

e) A_i's *AM-Timer* = N_{i+1}'s *Area Start Time* (if available)

f) Else

g) AM-LSA's *Area Period* = *MaxMembershipPeriod*

h) A_i's *AM-Timer* = AM-LSA's *Area Start Time* + *MaxMembershipPeriod*

i) Else

j) AM-LSA's *Area Start Time* = N_i's *Area Start Time*

k) If N_i's *Area Period* < *MaxMembershipPeriod*

l) AM-LSA's *Area Period* = N_i's *Area Period*

m) A_i's *AM-Timer* = N_i's *Area Time*

n) Else

o) AM-LSA's *Area Period* = *MaxMembershipPeriod*

p) A_i's *AM-Timer* = AM-LSA's *Area Start Time* + *MaxMembershipPeriod*

The propagation delay between neighboring routers must be considered. Thus, the maximum propagation delay from the LSA's generating router in area *i* to all router of area *i*, are subtracted from area *i*'s *AM-Timer*. Also, if *Multi-Areas LSA bit* is set, the router searches for the smallest *AM-Timer* from all areas' data structure. Then nullify all the rest.

3. *Local change in a Neighboring Routers list*

   If an SOSPF router changes its current area membership time interval definition in the *Neighboring Routers list*, a new AM-LSA is generated which reflects the change. The setting of *Area Start Time* and *Area Periods* of each tuple in the AM-LSA is defined in scenario 2.

4. *Invitations to a new area*

If SOSPF router *x* receives an AM-LSA which includes *x*'s ID in the *Area Members* field, *x* will create a *Neighboring Routers list* for the new area and generates an AM-LSA for it. At the time of this research, all AM-LSA invitations are accepted. The setting of *Area Start Time* and *Area Periods* of each tuple in the AM-LSA is defined in scenario 2.

5. *Joining an area*

   If SOSPF router *x* adds a new *Neighboring Routers list* for an area it wants to join, call it *i*, a new AM-LSA is generated for *i* and forwarded to *i*'s members. Once the *x*'s AM-LSA is received by another router, call it *y*, it checks if the *Area Members* definition in *x*'s AM-LSA matches *i*'s *Neighboring Routers list'*s *Neighboring routers*. If they match, *y* generates a new AM-LSA with the new member and forwards it to the members of *i*. If the *Area Members* don't match, *y* sends its *i*'s AM-LSA to *x* to synchronize the *i*'s area membership. Also, *y* record *x*'s AM-LSA in the *Area Membership list* (for error detection, check scenario 8). Currently, *x* is waiting for an acknowledgment from y. Thus, when *x* receive *y*'s AM-LSA, it changes *i*'s *Neighboring Routers list* and *Area Membership list* accordingly. Then, *x* generates a new AM-LSA with the new area definitions and sends it to y. Once y receives the new correct AM-LSA, *y* generates a new AM-LSA with the new member and forwards it to the members of *i*.

6. *Remote change in a Neighboring Routers list*

   If SOSPF router *x* receives an AM-LSA which changes an area membership time interval definition in the *Neighboring Routers list*, a new AM-LSA is generated which reflects the change. This is done to synchronize with other members of the affected area. The setting of *Area Start Time* and *Area Periods* of each tuple in the AM-LSA is defined in scenario 2.

7. *Failed neighboring router*

   Similar to SR-LSAs (see 3.4.1.1), when SOSPF router *x* learns that a neighboring router *y* is down, *x* modifies all the *Neighboring Routers lists* for areas that *y* belongs to. Then, *x* generates an AM-LSA for each changed list. The setting of *Area Start Time* and *Area Periods* of each tuple in the AM-LSA is defined in scenario 2.

8. *Bad AM-LSA*

   There area two cases for receiving a bad AM-LSA

   a) If the following events happen:

      i) *x* receive an AM-LSA, which is generated by SOSPF router *y*, which indicate that SOSPF router *z* is no longer in the area *i*.

      ii) Within the next *MinLSArrival* (see section 3.5), *z* either sent or replied to hello packet to *x*.

      iii) And, *z* have not issued a new AM-LSA for *i* that indicates that *z* is no longer a member of *i*,

   Then, *x* issues a neighbor event *Bad AM-LSA* (see section 3.2.1) in *y*'s neighbor state data structure which causes the neighboring relationship between *x* and *y* to restart. *x* issue a new AM-LSA about area *i* without *y* in it. Also, *x* updates *i*'s *Neighboring Routers list* accordingly. The setting of *Area Start Time* and *Area*

Periods of each tuple in the AM-LSA is defined in scenario 2.

Looking at scenario 5, if SOSPF router *y* receive a new AM-LSA from SOSPF router *x* who wants to join area *i* and the *x*'s AM-LSA has incorrect information about the *i*'s *Area Members*. *y* sends its *i*'s AM-LSA to *x* to synchronize the i's area membership. Also, *y* record *x*'s AM-LSA in the *Area Membership list*. If y's receive another *x*'s AM-LSA that is the same as the one in *Area Membership list*, *x* issues a neighbor event *Bad AM-LSA* (see section 3.2.1) in *y*'s neighbor state data structure which causes the neighboring relationship between *x* and *y* to restart. *x* issue a new AM-LSA about area *i* without *y* in it. Also, *x* updates *i*'s *Neighboring Routers list* accordingly. The setting of *Area Start Time* and *Area Periods* of each tuple in the AM-LSA is defined in scenario 2.

Once an AM-LSA is generated, it is stored in the router's LSD and forwarded according the AM-LSA forwarding scheme described in the next section.

Moreover, after an AM-LSA is generated and flooded throughout the routing domain, the *Area Membership list* is created. Once it is created, each member of the area in the *Area Membership list* proceeds with establishing bidirectional relationship with all other members in the same area.

3.4.2.2 Forwarding an AM-LSA

After an AM-LSA is generated, it is encapsulated in a Link State Update packet and forwarded to all SOSPF routers in the areas that are defined in the AM-LSA and a few other SOSPF routers if needed.

An AM-LSA may be generated at any level of the area hierarchy and then forwarded to the areas which are defined in the AM-LSA. For example, a space shuttle is currently in 1:3 and will join area 1:4 in a few hours. The space shuttle can generate and forward an AM-LSA for area 1:4 while it is in area 1:3. This feature is made available in the SOSPF routing protocol for two reasons:

1. *Reduce processing overhead*.

   With this feature, an SOSPF router (e.g., space shuttle) can calculate its area membership ahead of time for all the areas that it will belong to during a specified period. Thus, an SOSPF router doesn't have to generate an AM-LSA every time it joins a new area, saving processing power.

2. *Reduce control traffic at the targeted area during their area membership period*.

   For example for security reasons, no AM-LSA transmissions are allowed in area *A* at a certain time which coincides with the time when SOSPF router *x* joins *A*. Thus, *x* sends its AM-LSA which contains its area membership details with *A* ahead of time into the area that *x* is currently part of. Thereafter, the AM-LSA finds its way into *A* as explained next.

Each received or generated AM-LSA has a list called the *AM-Forwarding List* which contains the area IDs of the areas that must receive the AM-LSAs. Whenever an AM-LSA is generated or received, it must be processed by the flooding procedure prior forwarding any further (see section 3.5). Then, the following steps are performed for each area defined in the AM-LSA:

1. If the followings occur:

    a) The router's area is equal to the AM-LSA's area.

    b) The router is the generator of the AM-LSA.

    c) And, the router belongs to areas that are in lower than the current levels in the area hierarchy.

    Then, the router adds all those lower level areas to the *AM-Forwarding List*. For example, if SOSPF router *x* belongs to areas 1, 1:3, 1:4, 1:3:1, and 1:3:2 and *x* generates an AM-LSA for area 1:3 in area 1:3, then 1:3:1 and 1:3:2 are added to the *AM-Forwarding List*.

2. If the router's area is equal to the AM-LSA's area, add the area's ID to the *AM-Forwarding List* and proceed to next area in the AM-LSA.

3. Else find the most common area field between the router's area and the AM-LSA's area. For example, if the area in the AM-LSA's is 1:4:5:2 and the router's area is 1:4:1:1, then the most common are is 1:4. Then:

    a) If the most common area is in a higher level (in the area hierarchy) than the router's area and the SOSPF router is a member of its parent area toward the common area, then add this parent area's ID to the *AM-Forwarding List*. Continuing from the previous example above, 1:4:1 is added to the *AM-Forwarding List*.

    b) Else, the router's area is most common area. Thus, add the area's ID of area in the area hierarchy downward toward the AM-LSA's area to the *AM-Forwarding List*.

4. Once, all the areas in the AM-LSA are exhausted, the AM-LSA is forwarded to all

members of each one the areas define in *AM-Forwarding List*.

3.4.2.3 Area Membership List

Each SOSPF router has a list, called the *Area Membership list*. This list is the collection of all the received AM-LSAs. The *Area Membership list* consists of multiple entries where each entry's parameters are identical to the tuple in an AM-LSA which are:

- *Area ID* is the area ID of the this tuple
- *Area Members* contains the list SOSPF routers' IDs that belong to the area defined in this tuple
- *Area Start Time* is the starting point in time for area's definition
- *Area Period* is the time period of the validity of this tuple

The *Area Membership list* is very important in the SOSPF routing protocol because of redundancy. If an area border router fails, other members of the failed area can replace the failed area border router within a finite time. The details of an area border router failure are discussed in section 3.4.2.4.

In Figure 15 shows an example *Area Membership list* for SOSPF router M1 which reflects the AM-LSAs that are issued and received by M2 according to *Neighboring Routers list* example shown in Table 5 and Table 6.

Furthermore, each SOSPF router has a list of the area border routers for each of the areas it is residing in, called area border routers (ABR) list (*ABR List*). This list contains multiple entries where each entry represents an area border router. Furthermore, each entry consists of one or more entries for each area that the area border router belongs to. Each area entry consist of the following:

- *Area ID* of the area border router
- *Membership Start Time* is the starting point in time for area's definition
- *Expiration Time* is the end time point in time for area's definition

An example of an *ABR List* for members of area 1:4:1 using the Mars example in Figure 10 is shown in Figure 16.

Number of entries		2
# 1	Area ID	1:4
	Area Members	M1. M12
	Area Start Time	2010:12:00:00:00:00
	Area Period	21845
#2	Area ID	1:4:1
	Area Members	M1. M2, and M6
	Area Start Time	2010:12:00:00:00:00
	Area Period	65535

Figure 15: An example of an Area Membership list for M1

Area Border Router's ID	Area ID	Membership Start Time	Expiration Time
M1	1:4	2010:12:00:00:00:00	2010:12:00:06:04:05
	1:4:1	2010:12:00:00:00:00	2010:12:00:18:12:15

Figure 16: An example of an ABR List for members of area 1:4:1

When SOSPF router *x* receives an AM-LSA from SOSPF router *y*, the following steps are taken for each area *i* in the AM-LSA:

1. If there is an entry for *i* in the *Area Membership list*

2. If *i*'s parameters values in the AM-LSA are the same as the *i*'s parameters values in

the *Area Membership list*

3. Replace the *i*'s entry in the *Area Membership list* with the *i*'s tuple in the AM-LSA

4. Check the next area in the AM-LSA

5. Else

6. Replace the i's entry in the *Area Membership list* with the i's tuple form the am-LSA

7. Else

8. Insert the *i*'s tuple from the AM-LSA in the *Area Membership list*

9. For each area *j* that *x* belongs to, do the following

10. If *y* belongs to *j* and *y* belongs to the parent area of *j* (this means that *y* is an area border router for area *j*)

11. Find *y*'s entry in the *ABR List*

12. If *i* is in *y*'s *Area IDs* in the *ABR List*

13. Replace *i*'s time interval in *y*'s entry located in the *ABR List* with time interval defined in the AM-LSA

14. Check the next area that *x* belongs to

15. Else

16. Add *i* to *y*'s *Area ID* in the *ABR List*

17. Add i's time interval defined in the AM-LSA in the *ABR List*'s *y*'s entry for

i's Area

18. Add *j* to *y*'s *Area ID* in the *ABR List*

19. Add *j*'s time interval defined in the AM-LSA in *y*'s entry located in the *ABR List*

20. *Else*

21. Check the next area that *x* belongs to

There is one area that doesn't have an *ABR List* which is the backbone area because there is no higher area than the backbone area.

3.4.2.4 Area Border Router Failure

All SOSPF routers belonging to one area know which SOSPF routers are the selected area border routers for this area and for how long. The *ABR List* provides those details and is updated whenever a new AM-LSA is received.

If area border router *x* for area *i* fails, a subset of SOSPF router that belong to area *i* become aware of the link failure through the Hello protocol (see section 3.6.3). Each SOSPF that belongs to area *i* will do the following:

1. For every area *i* that *x* belongs to in the *ABR List*, do the following

2. Find another valid area border router for area *i* in the *ABR List* (A valid area border router has an *Expiration Time* that is more than the current time).

3. If found,

4. Do nothing

5. Else

6. Elect a new area border router for area *i*. At the time of this research, the election process is based on the highest router ID. Since all members of an area are aware of each other through the Hello protocol (see section 3.6), no messages are exchanged and the router with the highest router ID does the following:

7. Assumes area border role

8. Generate a new *Neighboring Routers list* (see section 3.2.2).

9. Generates and forwards the necessary AM-LSAs (see section 3.4.2.1).

10. Generates and floods the SR-LSA for the failed router (see section 3.4.1.1)

3.5 The Flooding Procedure

As defined in OSPFv3 [51], The receiving process of a Link State Update packets provide the mechanism for flooding LSAs. A Link State Update packet may contain several distinct LSAs, and may have to flood each LSA one hop further from its point of origination. To make the flooding procedure reliable, each LSA must be acknowledged separately using Link State Acknowledgment packets. Many separate acknowledgments can also be grouped together into a single packet. In this dissertation, only the flooding procedure for the SOSPF LSAs is discussed. The flooding procedure LSAs of OSPFv3 are flooded according to the OSPFv3 guidelines [51].

For each LSA contained in a received Link State Update packet, the following steps are taken:

1. Find the instance of this LSA that is currently contained in the router's link state database (LSD). If there is no database copy or the received LSA is more recent than the database copy the following steps must be performed:
   a) If there is already a database copy, and if the database copy was received via flooding and installed less than *MinLSArrival*[5] (see B.3) seconds ago, discard the new LSA (without acknowledging it) and examine the next LSA (if any) listed in the Link State Update packet.
   b) Else install the new LSA in the LSD (replacing the current database copy). This may cause the routing table calculation to be scheduled (see section 4.6.2).
   c) Acknowledge the receipt of the LSA by sending a Link State Acknowledgment packet back out the receiving interface.
   d) If the LSA is an AM-LSA it is now forwarded according the AM-LSA forwarding procedure defined in section 3.4.2.2. Then, examine the next LSA (if any) listed in the Link State Update packet.
   e) If the LSA is an SR-LSA and it is generated by an SOSPF router which has its *DoNotAdvertise* bit set on (this is done by checking the Neighbor Data Structure of the originating router) examine the next LSA (if any) listed in the Link State Update packet.
   f) Otherwise immediately flood the new LSA out according the router's flooding scope (see section 3.5.1).

5 MinLSArrival is the minimum time that must elapse between reception of new LSA instances during flooding.

2. Else, if the received LSA is the same instance as the database copy, acknowledge the receipt of the LSA by sending a Link State Acknowledgment packet back out the receiving interface.
3. Else, the database copy is more recent. In this case, simply discard the received LSA without acknowledging it.

3.5.1 Flooding Scopes

When an LSA is decided to be flooded, it is flooded according to the area that the router belongs to. There are two types of flooding scopes in SOSPF, internal flooding scope and celestial flooding scope. All SOSPF routers that belong to a satellite constellation area are flooded according to the internal flooding scope. And, All SOSPF routers that belong to a celestial object area are flooded according to the celestial flooding scope.

Internal Flooding Scope: The LSA is forwarded to the SOSPF routers that have bidirectional relationship with (see section 3.6) except to the SOSPF router at the receiving interface and the LSA's originator. Those routers are defined in the *Neighboring Routers list* for the specified area.

Celestial Flooding Scope: The LSA is forwarded to all SOSPF routers that belong to the router's celestial area (bidirectional relationship is assumed) except to the SOSPF router at the receiving interface and the LSA's originator.

3.6 Hello Protocol

The main task for the Hello protocol is to establish and maintain neighbor relationships. Maintaining relationships with neighbors is done by exchanging hello packets at regular intervals. It also, ensures that communications between neighbors are bidirectional.

Source Satellite Address	5F00:0000:C001:0400::/56
Destination Satellite Address	5F00:0000:c001:2C00::/56
Calculating Method	0:0
DoNotAadvertise	0
Area	1:3
Semi-major axis (AU)	0.20943951606750488280e1
Eccentricity °	0.28189981821924448010e-3
Longitude of the ascending SOSPF router °	0.41887903213500976560e1
Argument of perigee °	0.24515445000000000000e7
Inclination of orbit °	0.13993999455124139790e-2
True angle °	0.59827148739760886020e-38
True angle Data and Time	2006:08:30:23:14:50
:	:

Figure 17: An example of a hello packet

An SOSPF hello packet contains following fields which are

- The source SOSPF router's IPv6 address
- The destination SOSPF router's IPv6 address
- The calculating method tag (see section 3.3)
- The DoNotAdvertise bit (see section 3.2)
- The Area of this bidirectional relationship
- The source SOSPF router's six orbital parameters (if the calculating method tag equal 0) (see APPENDIX A)
- The date and time where the source SOSPF router is at the true angle[6] (if the calculating method tag equal 0)

[6] True angle is one of the six orbital parameters

- Reserved parameters (not yet defined)

An example of a hello packet is shown in Figure 17. When an SOSPF router is turned on, it send hello packet to all members of the *Neighboring Routers list*. In turn, the receiving router will reply with its own hello packet and bidirectional relationship is established and neighboring relationship is being formed between the pair of routers. Thereafter, hello packets are exchanged to maintain this relationship.

3.6.1 Receiving Hello Packet

When an SOSPF router receives a hello packet from another SOSPF router, the condition of the hello packet' reception and the action that follow is as follow:

- If the received hello packet is a reply of an earlier sent hello packet, then
  - The Hello timer[7] is restarted and the no further action is required.
- If the received hello packet is a new hello packet in result of the time interval expiration at the other end, then
  - Prepare and send a reply hello packet
  - The Hello timer is restarted.
- If the received hello packet is a new hello packet in result of establishing bidirectional relationship process, then
  - The part of establishing bidirectional relationship with a neighboring router is explained next (see section 3.6.2).

[7] hello packet are sent when Hello timer reaches zero

3.6.2 Establishing Bidirectional Relationship

When SOSPF router *R* receives a hello packet from SOSPF router *S*, it checks if *S*'s IPv6 address is in the *Neighboring Routers list*. If it doesn't exist, then SOSPF router *S*'s IPv6 address and the six orbital parameters (if *S*'s calculating method tag is zero) are added to the *Neighboring Routers list*. Also, if *S*'s IPv6 address is not in the *Space Routers List*, then SOSPF router *S*'s router's IPv6 address and its six orbital parameters are added to the *Space Routers List*.

Furthermore, and *R* sends a reply hello packet to *S* and restarts its Hello timer. When *S* receives *R*'s hello packets, it restarts its Hello timer. Now, both SOSPF routers *S* and *R* have bidirectional relationships with each other.

3.6.3 Bidirectional Relationship Termination

Hello packets are used to maintain bidirectional relationships with neighboring routers by exchanging hello packet at regular interval. If a hello packet is sent and is not replied to within a configured waiting period, it is removed from the *Neighboring Routers list* and a new *SR-LSA* is issued and flooded according to the router's flooding scope (see section 3.4.1.1). The *SR-LSA* contains one entry that entails the *Propagation Delay* of InfDelay (see section 3.4), the SR-LSA's *Begin Time* is set to the current time, and the *Connection Period* is set to zero.

3.7 Database Exchange Process

Once a pair of SOSPF routers has bidirectional relationships, their link state database must be synchronized. Each SOSPF router describes its LSD into a number of network packets, called Database Description packets. Each Database Description packet contains a number of LSAs which were generated by SOSPF routers that wish to be advertised into the *Area* defined in the hello packet (see section 3.2).

Once all the Database Description packets are transmitted, each router transition the neighbor state of the other router from *Exchange* state to *Loading* state (see section 3.2.1). Then each router checks which LSAs need to be updated and request them in Link State Request packets. The recipient of a Link State Request packet finds the requested LSAs in its LSD and transmits them to the neighbor in Link State Update packets. All Link State Update packets must be acknowledged (see section 3.5).

During the database exchange, if SOSPF router *x* 1) receives an AM-LSA for area *i* that was generated by SOSPF router *y* and 2) area *i*'s entry in x's *Area Membership list* does not contain *y* in the *Area Members* field, then *x* recalculates its *Area Membership list* and proceed to form bidirectional communication and neighboring relationship with *y* (see section 3.6.2).

In SOSPF, if an LSA, other than AM-LSAs, which is generated by an SOSPF router that is not in the *Space Routers List*, the SOSPF router's ID is recorded in *New SOSPF Routers List*. The coordinates of the entries in this list will be requested at the conclusion of the database exchange process. A Coordinate Request packet is sent to the neighbor for every new SOSPF router. Once the Coordinate Request packet is received, a Coordinate Definition packet is sent which contains the calculating method tag and the six orbital parameters of the new SOSPF router (if the calculating method tag is zero). If the calculating method is not zero, the calculating method must be available at the requesting router; otherwise the LSA that caused this packet exchange will be removed from the LSD. Finally at the conclusion of the Database Exchange Process, each router transitions the neighbor state of the other router to *Full* state (see section 3.2.1).

Moreover, if an SR-LSA is received whilst two neighboring routers are in *Full* state and the SR-LSA is generated by an SOSPF router that is not in the *Space Routers List*, then the receiving router request the coordinates of the new router through Coordinate Request packet from its neighboring router which cause each routers to transition the neighbor state of the other router to *Loading* state (see section 3.2.1). The remaining steps are done as described above.

3.8 SOSPF Example

This example shows a brief summary of some of the functionalities of SOSPF. This example is divided into seven parts where each part illustrates a few of SOSPF functions. The example scene includes:

- One celestial object area, called area 1, for the Sun.
- One celestial object area, called area 1:3, for Earth.
- Two satellite constellation areas around Earth,
    - One satellite constellation area, called area 1:3:1 has four SOSPF routers, namely E1, E2, E3, and E4. E2 and E4 take turn in becoming the area border router for 1:3 and 1.
    - One satellite constellation area, called area 1:3:2, has one SOSPF routers, namely

E5 which is the area border router for 1:3:2

- One celestial object area, called area 1:4, for Mars.
- Two satellite constellation areas around Mars,
    - One satellite constellation area, called area 1:4:1 has three SOSPF routers, namely M1, M2, and M3. M1 is the only area border router for area 1:4:1 and 1.
    - One satellite constellation area, called area 1:4:2 has three SOSPF routers, namely M4. M4 is the only area border router for area 1:4:2
- One space shuttle, SP, which belongs to different areas at various times.

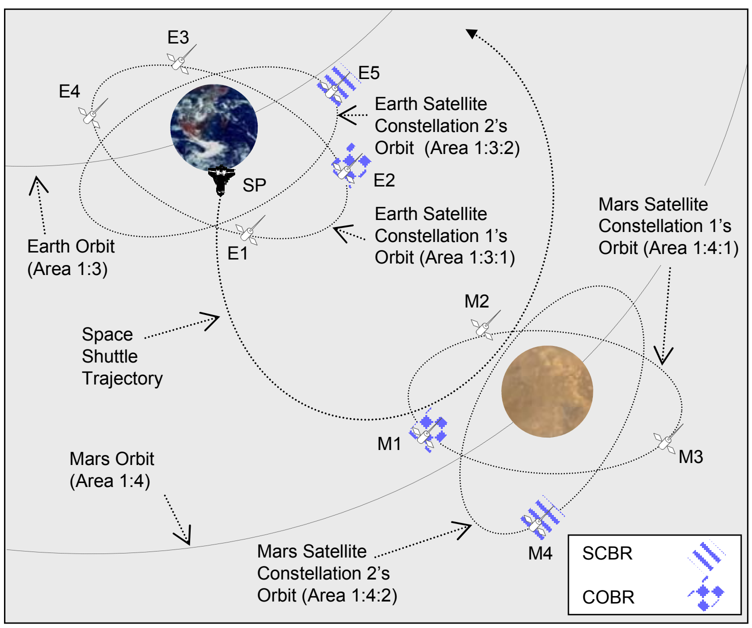

Figure 18: The example initial scene

Figure 18, Figure 19 and Table 7, illustrate the example, its initial area setting and hierarchy. Six events and the construction of the data structures resulted from each event are presented. Initially at midnight, SP is launched and the events that follow are:

1. At 1:00AM, SP joins area 1:3.

2. At 5:00AM, E2 is transitioned to *Sleeping* state in area 1:3 and 1 and E4 takes its place

3. At 6:00AM, SP changes the following:

   a) *DoNotAdvertise* bit to on

   b) Change its membership with 1:3 to exit the area at an earlier time

4. At 10:00AM, E4 is transitioned to *Sleeping* state in area 1:3 and 1 and E2 takes its place

5. At 12:00PM, SP joins 1:4

6. At 1:00PM, E2 fails

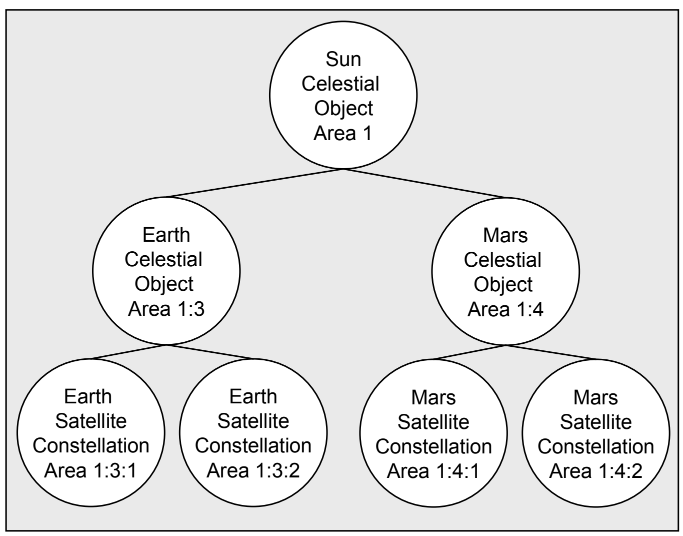

Figure 19: The area hierarchy of the failed link example

The initial settings for SP are as follow:

- SP's *DoNotAdvertise* bit is set off

- SP's *Multi-Areas LSA* bit is set on

- SP will join area 1:3 from 1:00AM till 3:00PM and join area 1:4 from 12:00PM till 6:00PM as presented in their *Neighboring Routers list* in Table 8 and Table 9 respectively.

The *ABR List* for the members of area 1:3:1 is shown in Table 10. The *ABR List* for the other areas are not shown here since they will not be affected in this example.

Table 7: The area membership of the failed link example

Area	Members
1	E2 and M1
1:3	E2 and E5
1:4	M1 and M4
1:3:1	E1, E2, E3, and E4
1:3:2	E5
1:4:1	M1, M2, and M3
1:4:2	M4

The current settings of E2's areas are shown in Table 11. The other SOSPF routers' data structure can be deduced Table 11 as well.

Table 8: SP's Neighboring Routers list for area 1:3

Number of entries		1
#1	Neighboring Routers	E2,E5
	Neighboring Start Time	1:00:00
	Neighboring Period (seconds)	50400

Table 9: SP's Neighboring Routers list for area 1:4

Number of Entries		1
#1	Neighboring Routers	M1, M4
	Neighboring Start Time	12:00:00
	Neighboring Period (seconds)	21600

Table 10: The ABR List for members of area 1:3:1

Area Border Router's ID	Area ID	Membership Start Time	Expiration Time
E2	1	2010:12:00:00:00:00	2010:12:00:05:00:00
	1:3	2010:12:00:00:00:00	2010:12:00:05:00:00
	1:3:1	2010:12:00:00:00:00	2010:12:00:05:00:00

Table 11: E2's current Area Membership *list*

Number of entries		3
Entry # 1	Area ID	1
	Area Members	E2,M1
	Area Start Time	00:00:00
	Area Period (seconds)	18000
Entry # 2	Area ID	1:3
	Area Members	E2,E5
	Area Start Time	00:00:00
	Area Period (seconds)	18000
Entry # 3	Area ID	1:3:1
	Area Members	E1,E2,E3
	Area Start Time	00:00:00
	Area Period (seconds)	65535

3.8.1 SP Joins Area 1:3 at 1:00AM

Hereafter, the number to the left of event coincides with number on the arrow in the referenced figure. Figure 20 shows the events that happen when time reaches 1:00AM where, according to the SP's *Neighboring Routers list* (see Table 8), SP must join 1:3 are as follow:

1.a. At 1:00AM, SP Sends hello packet to E2 and E5 (see Table 12)

In response, E2 and E5 reply with their own hello packets (see Table 13). Then SP calculate the SR-LSAs to all routers and stores them in its LSD. Since SP's *Multi-Area LSA* bit is set on, SP calculates one AM-LSA (shown in Table 14) for all areas in the *Neighboring Routers list* shown in Table 8. Then SP stores it in its LSD.

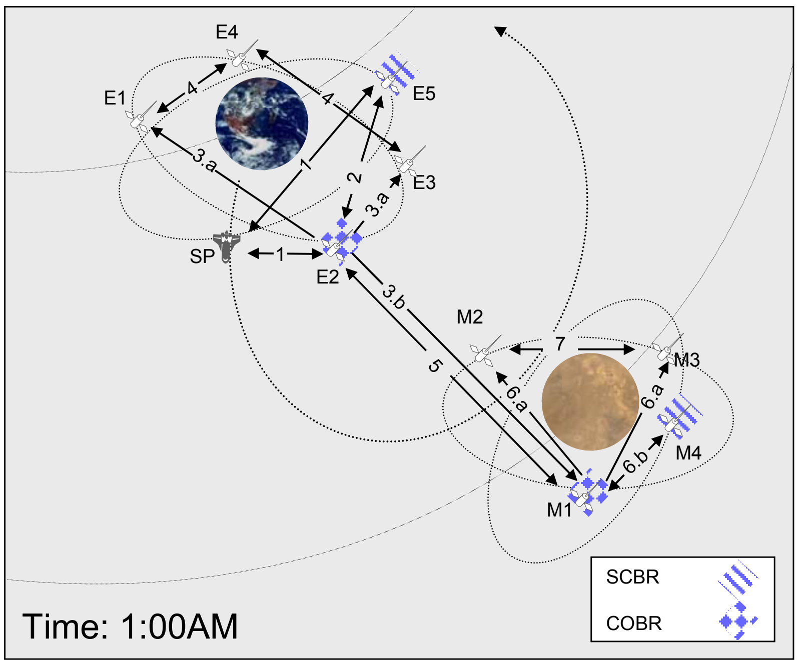

Figure 20: The example scene when SP joins area 1:3

2.a. SP Exchanges LSD with E2 and E5.

SP forms neighboring relationship with E2 and E5 where they exchange the LSA entries of their LSDs. When they are done exchanging their LSDs, E2 and E5 transition SP's neighbor state to *Full*; and SP transition neighbor state for E2 and E5 to Full, as well.

Table 12: SP's Hello packet to E2

Source Satellite Address	5F00:0000:C001:0601::/56
Destination Satellite Address	5F00:0000:C001:0402::/56
Calculating Method	0:1
DoNotAadvertise	0
Parameter 1	…
…	…

Table 13: E2's Hello packet to SP

Source Satellite Address	E2
Destination Satellite Address	SP
Calculating Method	0:0
DoNotAadvertise	0
Area	1:3
Semi-major axis (AU)	…
Eccentricity °	…
Longitude of the ascending router °	…
Argument of perigee °	…
Inclination of orbit °	…
True angle °	…
True angle Data and Time	…
:	:

Now, E2 and E5 calculate an SR-LSA to SP which provides the time interval and the propagation delay of the link between them and SP (see Table 15). Table 15 shows that E2 is in direct view with SP until 6:00AM and will be in direct view again at 8:00AM. Concurrently, when SP's AM-LSA is received at E2 (E5 will do the same action), E2 will do the following:

- Since SP's *DoNotAdvertise* bit is off, E2 insert area 1:3 and 1 to its *AM-Forwarding List* according to the AM-LSA forwarding guideline in section 3.4.2.2.

- E2 forwards SP's AM-LSA to members of area 1:3, namely E5
- E2 generates its own AM-LSA and send it to SP and E5 after storing it in its LSD
- E2 recalculates entries for 1:3 in *Area Membership list*

Concurrently, since SP's *DoNotAdvertise* bit is off, E2 floods SP's SR-LSAs and E2's SR-LSA according to the flooding procedure (see section 3.5).

3.a. E2 forwards the new LSAs to the members of area 1:3:1.

E2 forwards the E2's AM-LSA to E1 and E3 according to the AM-LSA forwarding guideline (see section 3.4.2.2). And, E2 floods the SR-LSAs from itself, SP, and E5 to E1 and E3, as well

3.b. E2 forwards the new LSAs to the member of area 1

E2 forwards the SP's AM-LSA to M1 since area 1 is in SP's AM-LSA's *AM-Forwarding List*. E2 floods the SR-LSAs from itself, SP, and E5 to M1, as well

Table 14: SP's AM-LSA for all areas

Source Satellite Address		5F00:0000:C001:0601::/56
Number of tuples		2
# 1	Area ID	1:3
	Area Members	SP,E2,E5
	Area Start time	01:00:00
	Area Period	50400
# 2	Area ID	1:4
	Area Members	SP,M1,M4
	Area Start time	12:00:00
	Area Period	21600

Table 15: E2's SR-LSA to SP

<table>
<tr><td colspan="2">Source Satellite Address</td><td>5F00:0000:C001:0402::/56</td></tr>
<tr><td colspan="2">Destination Satellite Address</td><td>5F00:0000:C001:0601::/56</td></tr>
<tr><td colspan="2">Number of tuples</td><td>1</td></tr>
<tr><td rowspan="3"># 1</td><td>Begin time</td><td>01:00:00</td></tr>
<tr><td>End time</td><td>06:00:00</td></tr>
<tr><td>Propagation Delay</td><td>…</td></tr>
<tr><td rowspan="3"># 2</td><td>Begin time</td><td>08:00:00</td></tr>
<tr><td>End time</td><td>15:00:00</td></tr>
<tr><td>Propagation Delay</td><td>…</td></tr>
</table>

4. Members of 1:3:1 flood the received LSAs to each other.

   This example assumes that SP's trajectory is known to members of 1:3:1 which means that they will not request SP's parameters when they receive an SR-LSA about SP.

5. The portion of the example assumes that M1 does not SP's trajectory; thus, when M1 receive SP's SR-LSA, M1 exchanges the required information with E2. The process is as follow

   a) M1 transition E2's neighbor state to *Exchange*

   b) M1 requests the coordinates of SP in a Coordinate Request packet

   c) M1 transition E2's neighbor state to *Loading*

   d) Once received, E2 transition M1's neighbor state to *Loading*

   e) E2 send SP's coordinates in a Coordinate Definition packet

   f) E2 transition M1's neighbor state to *Full*

   g) Once received, M1 send acknowledgment

   h) M1 transition E2's neighbor state to *Full*

6. M1 floods SP to 1:4 and 1:4:1 according to the flooding procedures (same set of events

which described in (5) occur between members of 1:4 and 1:4:1)

7. Members of 1:4 flood the SP's SR-LSA according to the flooding procedure.
8. (not shown) For every router that received new information about SP, they calculate their SR-LSA and flood their areas as mentioned above.

3.8.2 E2 Is Transitioned To Sleeping State at 5:00AM

Members of areas 1 and 1:3 trigger Sleep event with E2, and vice versa. This event transition E2's neighbor states in all those members to Sleeping state.

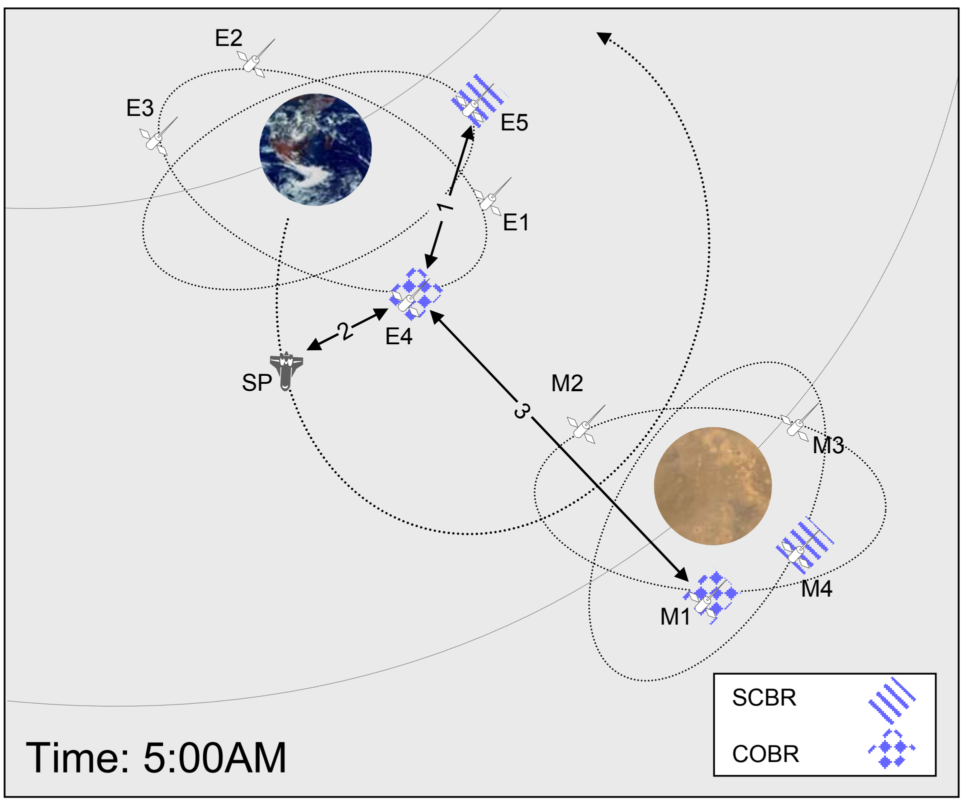

Figure 21: The example scene when E4 joins 1:3 and 1

Also, this event causes a *New LSA list* to be created at each router. The events that follow are:

1. E4 joins area 1:3

   E4 is new to area 1:3; this portion of the example assumes that E4 know only of E5 existence in 1:3. Thus, E4 joins area 1:3 by sending Hello packet to E5 and form neighboring relationship and exchange their LSDs. Also, E4 send its own AM-LSA which contains only E4 and E5 (see Table 16). But when E5 receive E4's AM-LSA for area 1:3, it finds that SP is missing from E4's AM-LSA. Then E5 sends its own AM-LSA for area 1:3 which causes E4 and E5 to synchronies their area 1's area membership. (see section 3.4.2.1 scenario 5)

   Once the correct E4's AM-LSA is accepted, E4's AM-LSA is forwarded to all members of 1:3:1 which causes the replacement of the *Area Border Router's ID* in the *ABR List* of all the members of 1:3:1, namely E1, E2, and E3.

2. E4 form neighboring relationship with SP

   This process is similar to the actions taken by E2 in the first scenario where both router, E4 and SP, exchange hello packets and exchange their LSDs descriptions. At the end of the database exchange process, each router transitions the neighbor state of the other router to *Full* state.

3. E4 form neighboring relationship with M1

   This process is similar to the previous process. At the conclusion of the E4 establishing relationship with M1, E4 is a member of area 1.

Table 16: E4's AM-LSA for area 1:3

Source Satellite Address		5F00:0000:C001:0404::/56
Number of tuples		1
# 1	Area ID	1:3
	Area Members	E4,E5
	Area Start time	05:00:00
	Area Period	21800

3.8.3 SP Is Changing Its Data Structure at 6:00AM

The network administer changes SP's *DoNotAdvertise* bit to on and change its *Neighboring Routers list's* entry for 1:3 to exit the area at 10:00AM instead of 3:00PM as define in SP's *Neighboring Routers list* shown in Table 8.

This change causes SP to Recalculate its *Area Membership list and* generate a new AM-LSA. SP only forwards the new AM-LSA to all members of 1:3, namely E5 and E4. Since SP's *DoNotAdvertise* bit is on, the new AM-LSA does not travel outside 1:3.

Furthermore, E5, E4, and SP Add this LSA to the *New LSA list* in E2's neighbor data structure.

3.8.4 E4 Is Transitioned To Sleeping at 10:00AM

At 10:00AM, it is E2's turn to become the area border router for areas 1:3 and 1. E5 and SP trigger *Sleep* event with E4 which transition E4's neighbor states to *Sleeping*, and vice versa. In addition, *New LSA list* is created at each router.

The events that occur between E2 and E5 only (the events that occur between SP and E2 are similar) are explained further. E2 has not seen any changes because the *New LSA list* in the neighbor data structure for E5 is empty. Thus E2 triggers *Awaken and Ready* event which moves the neighbor state of E5 to *Full*. But, E5 has one new LSA (SP's AM-LSA). Thus, E5 triggers *Awaken and Unsynchronized* event which moves E2's neighbor state to

Exchange. Then, the following events occur :

1. E5 sends a Database Description packet to E2

   E5 sends Database Description packet which contains the new AM-LSA to E2. E2 receive the Database Description packet from E5. It triggers an *Unsynchronized* event which moves E5's neighbor state to Exchange.

2. E2 request new LSA

   The received Database Description packet is marked with The More bit set off (as defined in OSPFv3 [59]) which indicate that this packet is the last Database Description packet. This cause each router to transition the neighbor state of the other router to *Loading* state where E2 will request the new AM-LSA via Link State Request packet

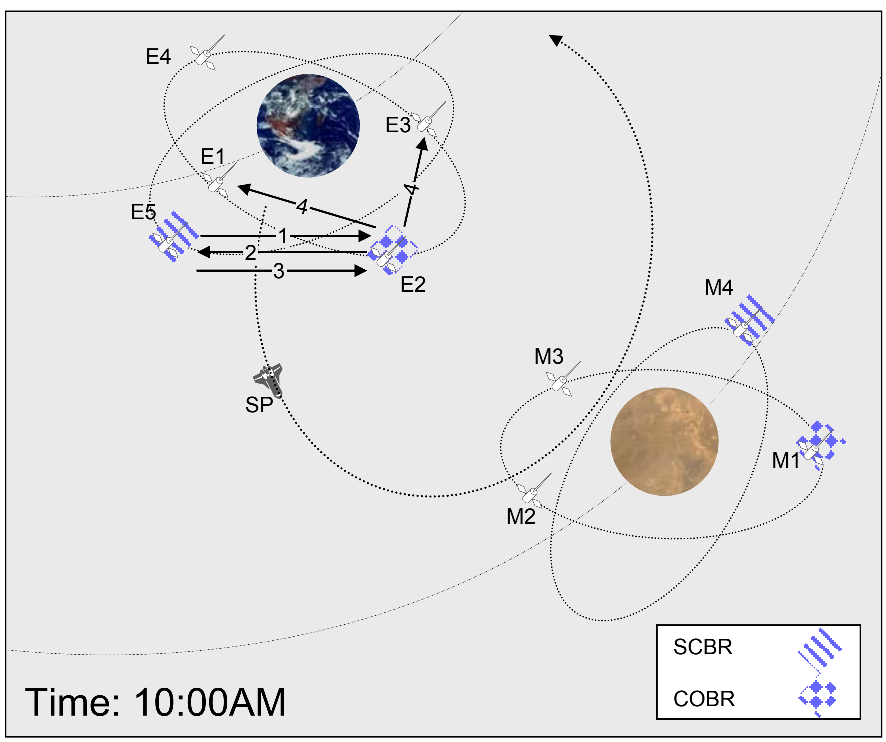

Figure 22: The example scene when E2 joins 1:3 and 1 again

3. E5 sends new LSA

   Once E5 receive the Link State Request packet, it sends the new AM-LSA in a Link State Update packet. Once E2 receives the AM-LSA, it does the following:

   - Sends acknowledgment to E5
   - Recalculate its *Area Membership list*
   - Transition E5's neighbor state to *Full*

   Once E5 receives the acknowledgment, it transition E2's neighbor state to *Full*

4. E2 announce to the members of 1:3 that it is now the area border router for 1:3

   According the E2's *Neighboring Routers list* for area 1:3, E2 will be area border router for areas 1:3:1 and 1:3 for next five hours. This causes E2 to send its AM-LSA for those areas which causes the replacement of the Area Border Router's ID in the ABR List of all the members of 1:3:1 and 1:3. The ABR List which located in the data structure of the members of area 1:3:1 is shown in Table 17

Table 17: The ABR List for member of area 1:3:1

Area Border Router's ID	Area ID	Membership Start Time	Expiration Time
E2	1	2010:12:00:10:00:00	2010:12:00:15:00:00
	1:3	2010:12:00:10:00:00	2010:12:00:15:00:00
	1:3:1	2010:12:00:10:00:00	2010:12:00:15:00:00

3.8.5 SP Joins Area 1:4 at 12:00PM

Members of area 1:4 know that SP will be part of their area at 12:00PM from SP's AM-

LSA for area 1:4 (see Table 9) that was received earlier from scenario (1).

M1 and M4 exchange Hello packet with SP and form neighboring relationship and exchange LSA.

3.8.6 E2 Fails at 1:00PM

At 1:00PM, E2 stopped functioning. Members of the areas, that E2 belongs to, find out that E2 is down after *RouterDeadInterval* seconds elapsed without any communication from it. Also in area 1, M1 finds out about the failure of E2 but M1 can't do anything because there is no *ABR List* for backbone area. In the other hand, members of areas 1:3 and 1:3:1can elect a new area border router for their respected area.

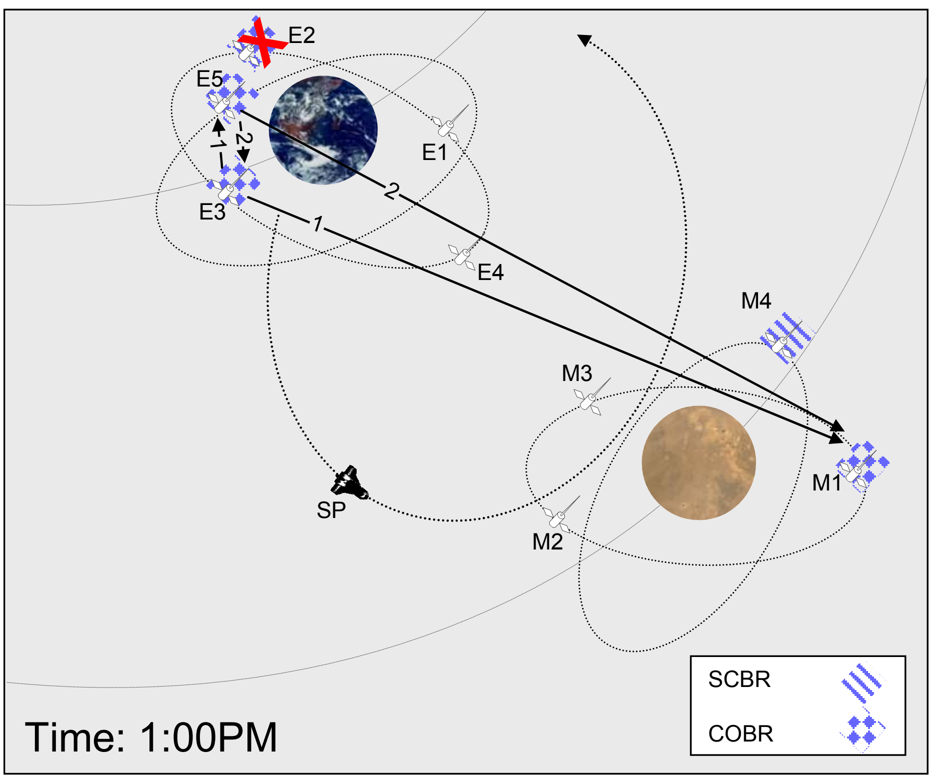

Figure 23: The example scene when E2 fails at 1:00PM

First in area 1:3:1, E3 (E1 does the same thing) does not receive a hello packet from E2 within *RouterDeadInterval* seconds. E3 looks for another area border router in the ABR List (see Table 17). Since there is none, E3 elects a new area border router. The election is base the highest router ID which is E3. E3 assumes the role of the area border router for area 1 and 1:3 and does the following:

1. E3 informs the members of areas 1:3 and 1 with the new change

   E3 dose the following:

   - Generates new *Neighboring Routers lists* for areas 1:3 and 1.
   - Generates an AM-LSA for area 1:3 and an AM-LSA for area 1, forwards to members of area 1:3, namely E5 and SP, an area 1, namely M1, respectively
   - Generates SR-LSA for E2 with infinity, floods it into 1:3:1, 1:3 and 1

   Moreover, E3 informs the members of area 1:3:1 of the new change, as well

2. Also, E5 becomes an area border router for area 1:3, similar to the events described above. E5 informs the members of areas 1 and 1:3 with the new change.

3. (not shown) For each router that received new information about E2, it creates an SR-LSA for E2 with *Propagation Delay* of *InfDelay, Begin Time* of the current time, and the *Connection Period* of zero, and floods it into their areas.

CHAPTER 4

SDIP ROUTING ALGORITHM

The Shortest Delay Intermittent Pathway (SDIP) routing algorithm finds approximate optimum paths in an intermittent link environment. If a number of routers are placed on board spacecrafts or space colonies, the connectivity between them will be intermittent because of occlusion of celestial objects or other limitations. The SDIP routing algorithm is not specifically defined for routing in space, but rather for any routing domain where such intermittent links exist. Nevertheless, space is an attractive routing domain for the SDIP routing algorithm because of the existence of the SOSPF routing protocol.

4.1 SDIP Algorithm Graph Definition

The SDIP algorithm defines a network as a connected directed graph $G = (V, L, T)$, where V is the set of vertices (routers), L is the set of links, and T is the set of clock times which indicate the beginning and end of the time period where links in L are active.

Let n be the number of vertices in V. Let the link connecting vertex x to vertex y be represented as (x, y). Let $b(x, y)$ be the entry in T which indicates the beginning of the active time period for (x, y). Let $e(x, y)$ be the entry in T which indicates the end of the active time period for (x, y). An illustration of the beginning and the end of a link's active time period is shown in Figure 24.

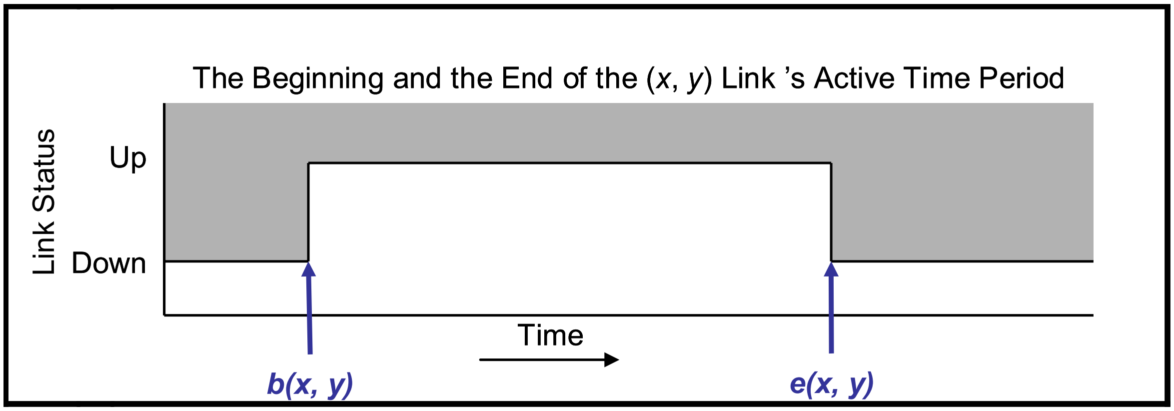

Figure 24: The beginning and end of an active time period

A cost function $c : L \rightarrow R$ is defined, where R is the set of real numbers, determines the length of (x, y), if (x, y) $\in$ L. Let the cost function associated with (x, y) be presented as c(x, y). The cost is a delay metric which is the estimated propagation delay between a pair of vertices. An illustration of the cost of a link is shown in Figure 25. Since link costs in the SDIP algorithm are delay matrices with the assumption that all link costs have positive values.

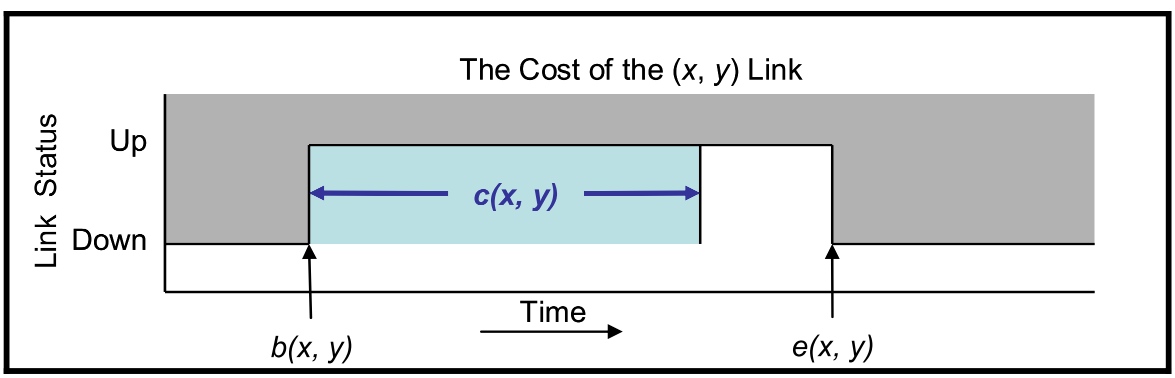

Figure 25: The Cost Metric

4.2 Input and Output Matrices

First, the graph G is represented by four matrices which are used for input and output where entries in one or more matrices may be used to modify one entry in a another matrix when a shorter path is found. Those matrices are:

1. An $n \times n$ path matrix $P = [p_{ij}]$ where p_{ij} is the shortest path from vertex i to vertex j and may contain intermediate vertices which yielded p_{ij} to be the shortest path from i to j. The initial setting of P is done as follow:

$$p_{ij} = \begin{cases} \langle i, j \rangle & \text{if } i \neq j \text{ and } (i, j) \in L \\ \text{NULL} & \text{Otherwise} \end{cases} \tag{2}$$

2. An $n \times n$ cost matrix $C = [c_{ij}]$ where c_{ij} represents the cost of the path p_{ij}. The initial setting of C is as follow:

$$c_{ij} = \begin{cases} 0 & \text{if } i = j, \\ c(i, j) & \text{if } i \neq j \text{ and } (\mathrm{i, j}) \in L, \\ \infty & \text{if } i \neq j \text{ and } (\mathrm{i, j}) \notin L. \end{cases} \tag{3}$$

3. An $n \times n$ time matrix $B = [b_{ij}]$ where b_{ij} represents the beginning of the active time period of the path p_{ij}. The initial setting of B is as follow:

$$b_{ij} = \begin{cases} 0 & \text{if } i = j, \\ b(i, j) & \text{if } i \neq j \text{ and } (i, j) \in L, \\ \infty & \text{if } i \neq j \text{ and } (i, j) \notin L. \end{cases} \tag{4}$$

4. An $n \times n$ time matrix $E = [e_{ij}]$ where e_{ij} represents the end of the active time period of the path p_{ij}. The initial setting of E is as follow:

$$e_{ij} = \begin{cases} 0 & \text{if } i = j, \\ e(i, j) & \text{if } i \neq j \text{ and } (i, j) \in L, \\ \infty & \text{if } i \neq j \text{ and } (i, j) \notin L \end{cases} \tag{5}$$

Next, the SDIP routing algorithm produces one output matrix which is an $n \times n$ delay matrix D = $[d_{ij}]$ where d_{ij} represents the shortest estimated arrival clock time of the propagated data (refer to as the shortest delay hereafter) from vertex i to vertex j using the path p_{ij}. d_{ij} is set to be the cost of path p_{ij} plus the clock time of the beginning of the path's active time period (see Figure 26). The initial setting of D is done as follow:

$$d_{ij} = c_{ij} + b_{ij} \quad \forall i, j \in V \tag{6}$$

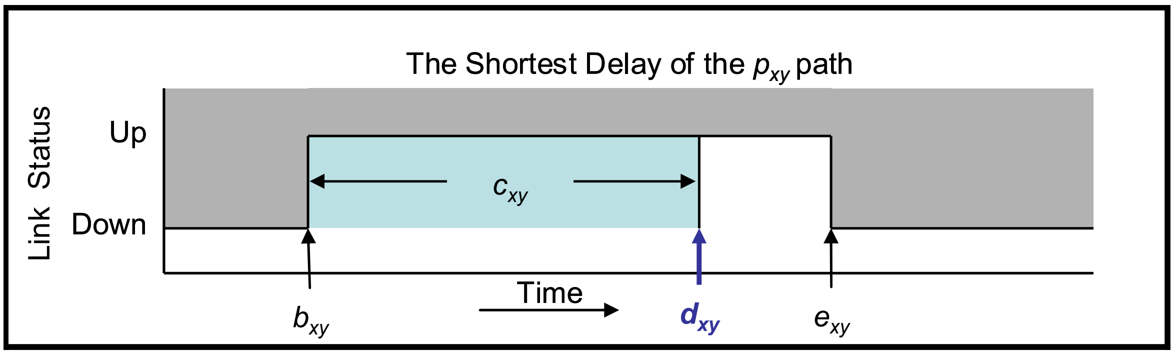

Figure 26: The delay of a path

4.3 SDIP Routing Algorithm Decomposition

Let $p_{ij}^{(k)}$ be the shortest path from vertex i to vertex j such that any intermediate vertices on the path (if any) are chosen from the set {1, 2, … , k} and $d_{ij}^{(k)}$ be the shortest delay using $p_{ij}^{(k)}$. For example, $d_{ij}^{(0)}$ is the shortest delay using $p_{ij}^{(0)}$ which is the path from vertex i to vertex j with no intermediate vertices if $(i, j) \in L$. Moreover, let the values of $c_{ij}^{(k)}$, $b_{ij}^{(k)}$ and $e_{ij}^{(k)}$ correspond to the $p_{ij}^{(k)}$. Lastly, let $D^{(k)}$ be the $n \times n$ matrix $[d_{ij}^{(k)}]$. Thus, $D^{(n)}$ is the final delay matrix which is solved by computing $D^{(k)}$ for k = 0,1,…, n.

For $p_{ij}^{(k)}$, there are two cases. The first case is the path from vertex i to vertex j which doesn't contain the vertex k. In reality, this is $p_{ij}^{(k-1)}$ which the shortest path found from the previous calculation.

In the second case, the path from vertex i to vertex j containing the vertex k is considered. This path consists of a sub-path $p_{ik}^{(k-1)}$, and a sub-path $p_{kj}^{(k-1)}$, provided that the combining of $p_{ik}^{(k-1)}$ and $p_{kj}^{(k-1)}$ sub-paths produce a valid path. The combination of $p_{ik}^{(k-1)}$ and $p_{kj}^{(k-1)}$ sub-paths is valid if one the following two conditions is achieved:

Condition 1: If the shortest delay of $p_{ik}^{(k-1)}$ is equal to or less than the beginning of $p_{kj}^{(k-1)}$'s active time period, then the combination of the two sub-paths is valid (see Figure 27) , i.e.,

$$d_{ik}^{(k-1)} \leq b_{kj}^{(k-1)} \qquad (7)$$

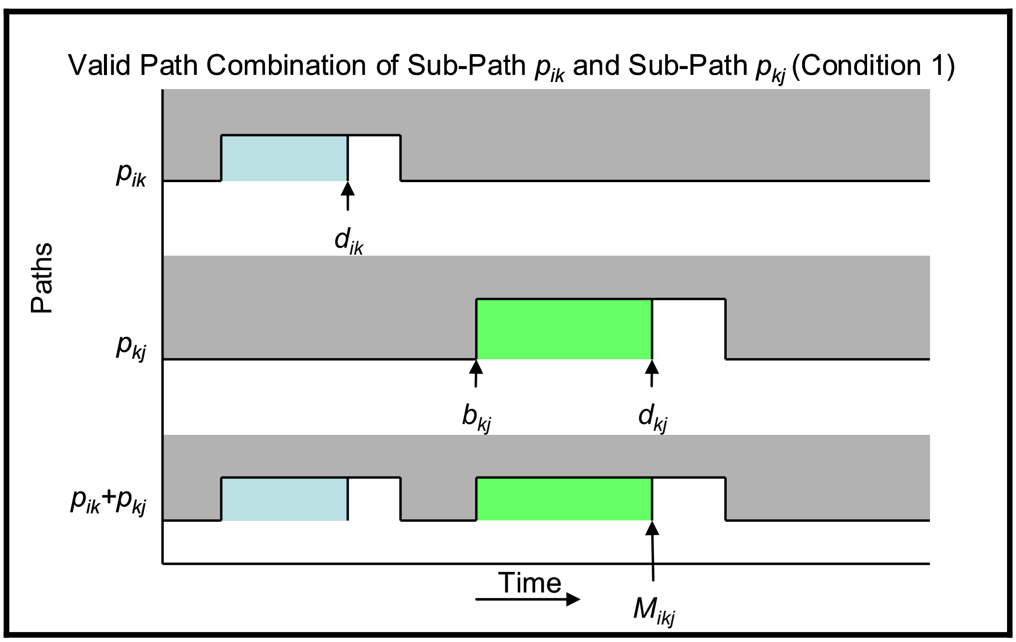

Figure 27: A valid path combination of the first condition

Condition 2: If 1) the shortest delay of $p_{ik}^{(k-1)}$ is more than the beginning of the sub-path

$p_{kj}^{(k-1)}$'s active time period and 2) the shortest delay of the sub-path $p_{ik}^{(k-1)}$ plus the cost of the sub-path $p_{kj}^{(k-1)}$ is equal to or less than the end of the sub-path $p_{kj}^{(k-1)}$'s active time period, then the combination of the two sub-paths is valid (see Figure 28), i.e.,

$$d_{ik}^{(k-1)} > b_{kj}^{(k-1)} \text{ and } d_{ik}^{(k-1)} + c_{kj}^{(k-1)} \le e_{kj}^{(k-1)} \tag{8}$$

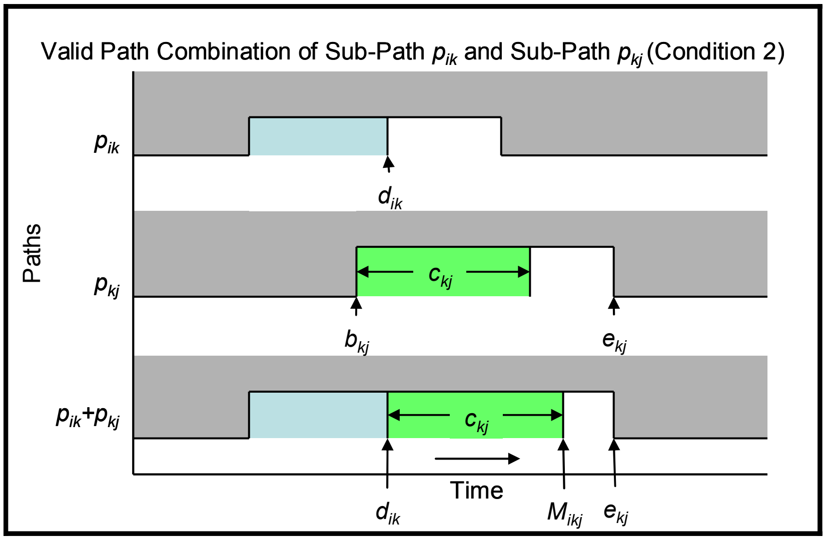

Figure 28: A valid path combination of the second condition

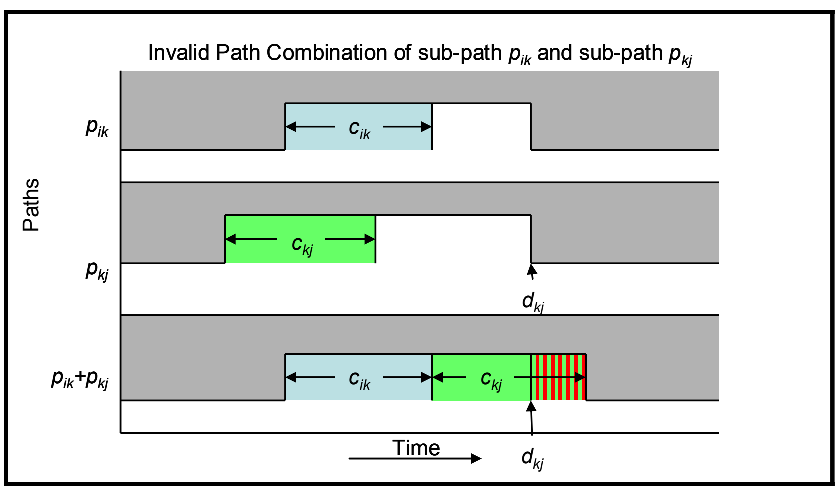

Figure 29: An invalid path combination

An invalid path combination of $p_{ik}^{(k-1)}$ and $p_{kj}^{(k-1)}$ sub-paths is illustrated in Figure 29 where the propagated data using $p_{ik}^{(k-1)}$ path will not be able to reach vertex *j* before the end of $p_{kj}^{(k-1)}$'s active time period.

If the combination of $p_{ik}^{(k-1)}$ and $p_{kj}^{(k-1)}$ sub-paths is valid, the shortest delay of the new path is calculated. For ease of terminology, let $M_{ikj}^{(k-1)}$ be the shortest delay resulting from a valid bath combination of $p_{kj}^{(k-1)}$ and $p_{kj}^{(k-1)}$ sub-paths. The value of $M_{ikj}^{(k-1)}$ depends on its validity condition.

Condition 1: $M_{ikj}^{(k-1)}$ is set to the shortest delay of $p_{kj}^{(k-1)}$ (see Figure 27) ,i.e.,

$$M_{ikj}^{(k-1)} = d_{kj}^{(k-1)} \tag{9}$$

Condition 2: $M_{ikj}^{(k-1)}$ is set to the shortest delay of $p_{ik}^{(k-1)}$ plus the delay cost of $p_{kj}^{(k-1)}$ (see Figure 28), i.e.,

$$M_{ikj}^{(k-1)} = d_{ik}^{(k-1)} + c_{kj}^{(k-1)} \tag{10}$$

Now, the two cases are compared, the path without *k* as an intermediate vertex case and path with *k* being an intermediate vertex case, i.e.,

$$d_{ij}^{(k)} = \min \{ d_{ij}^{(k-1)}, M_{ikj}^{(k-1)} \} \tag{11}$$

If $M_{ikj}^{(k-1)}$ is the shorter delay, entries in all the five defined matrices have to be modified. First, the two $p_{kj}^{(k-1)}$ and $p_{kj}^{(k-1)}$ sub-paths are combined to form the new shortest path $p_{ij}^{(k)}$. Path. Let $\oplus$ be a union function which combine the paths of two sub-paths such that it maintains the internal order composition of the sub-paths. Thus the combination of $p_{kj}^{(k-1)}$ and $p_{kj}^{(k-1)}$ which yields the $p_{ij}^{(k)}$ path is as follow:

$$p_{ij}^{(k)} = p_{ik}^{(k-1)} \oplus p_{kj}^{(k-1)} = \langle i, \ldots, k, \ldots, j \rangle \tag{12}$$

Then, the beginning and end of the active time period are set. The beginning of $p_{ij}^{(k)}$'s active time period is set to the beginning of $p_{ik}^{(k-1)}$'s active time period (see Figure 30 and Figure 31), i.e.,

$$b_{ij}^{(k)} = b_{ik}^{(k-1)} \tag{13}$$

Similarly, the end of $p_{ij}^{(k)}$'s active time period is set to the end of $p_{kj}^{(k-1)}$'s active time period (see Figure 30 and Figure 31), i.e.,

$$e_{ij}^{(k)} = e_{kj}^{(k-1)} \tag{14}$$

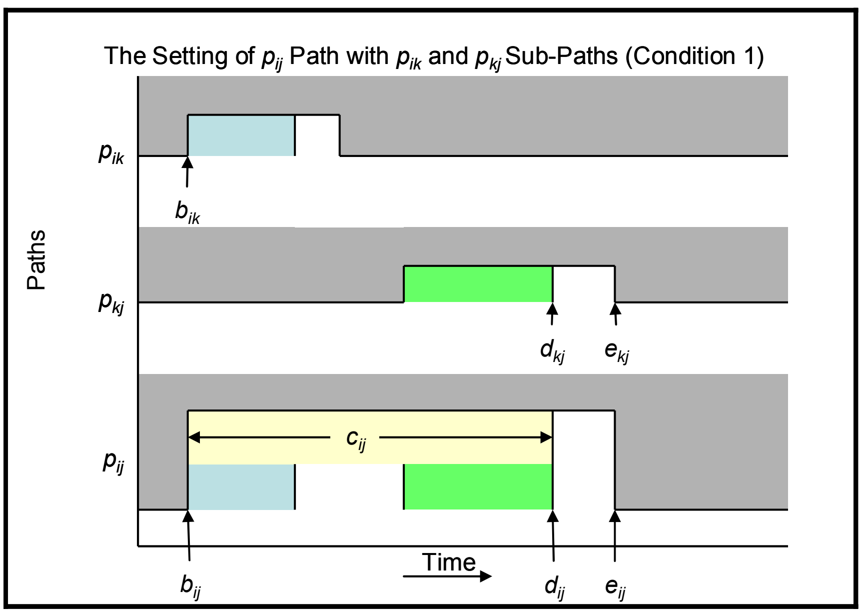

Figure 30: The setting of the new path of the first condition

The values of the shortest delay, $d_{ij}^{(k)}$, and the delay cost, $c_{ij}^{(k)}$, depend on its validity condition.

Condition 1: $d_{ij}^{(k)}$ is set to the shortest delay of $p_{kj}^{(k-1)}$ (see Figure 30),i.e.,

$$d_{ij}^{(k)} = d_{kj}^{(k-1)} \tag{15}$$

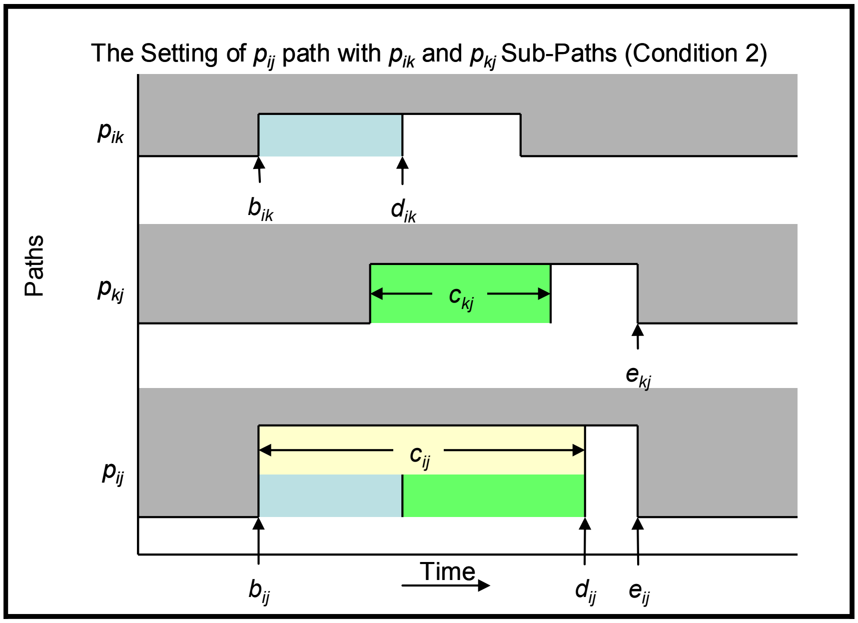

Figure 31: The setting of the new path of the second condition

And, $c_{ij}^{(k)}$ is set to the shortest delay of $p_{ij}^{(k)}$ subtracted by beginning of $p_{ij}^{(k)}$'s active time period (see Figure 30), i.e.,

$$c_{ij}^{(k)} = d_{ij}^{(k)} - b_{ij}^{(k)} \tag{16}$$

Condition 2: $d_{ij}^{(k)}$ is set to the shortest delay of $p_{ik}^{(k-1)}$ plus the delay cost of $p_{kj}^{(k-1)}$ (see Figure 31), i.e.,

$$d_{ij}^{(k)} = d_{ik}^{(k-1)} + c_{kj}^{(k-1)}. \tag{17}$$

Similar to the first condition , $c_{ij}^{(k)}$ is set to the shortest delay of $p_{ij}^{(k)}$ subtracted by beginning of $p_{ij}^{(k)}$'s active time period (see Figure 31), i.e.,

$$c_{ij}^{(k)} = d_{ij}^{(k)} - b_{ij}^{(k)} . \qquad (18)$$

4.4 SDIP Routing Algorithm

SDIP () {
For i = 1 to n
    For j = 1 to n
        $d_{ij} = b_{ij} + c_{ij}$
For k = 1 to n
    For i = 1 to n
        For j = 1 to n
            If $i \neq j$
                If $d_{ik} \leq b_{kj}$
                    $M_{ikj} = d_{kj}$
                Else If $d_{ik} + c_{kj} < e_{kj}$
                    $M_{ikj} = d_{ik} + c_{ki}$
                If $M_{ikj} < d_{ij}$
                    $p_{ij} = p_{ik} \oplus p_{kj}$

$d_{ij} = M_{ikj}$

$b_{ij} = d_{ik}$

$e_{ij} = e_{kj}$

$c_{ij} = d_{ij} - b_{ij}$

}

4.5 SDIP Routing Algorithm Example

An example network with four vertices, 1, 2, 3, and 4 is setup. In this example, there are only four links in L, namely (1, 2), (2, 3), (3, 4), and (1, 4). The active time periods and delay costs of those links are shown in Figure 32.

Prior running the SDIP algorithm, the initial values for the algorithm's data structure are setup. The initial path matrix (P) delay cost matrix (C) and beginning and end of the link's active time periods matrices (B and E respectively), and the shortest delay matrix (D) are shown in Table 18, Table 19, Table 20, Table 21, and Table 22 respectively.

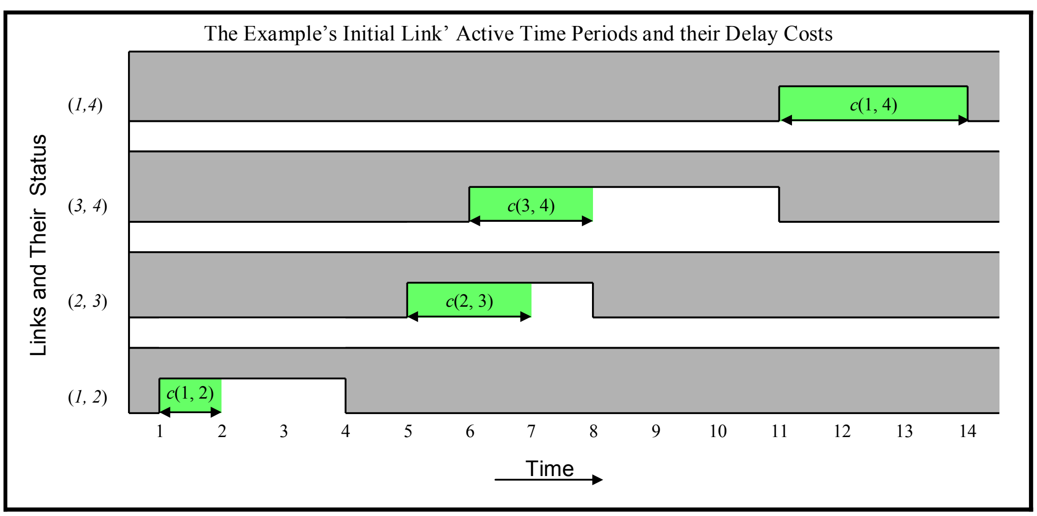

Figure 32: The example's Links and Their Parameters

Table 18: The example's initial path matrix

	1	2	3	4
1	NULL	<1,2>	NULL	<1,4>
2	NULL	NULL	<2,3>	NULL
3	NULL	NULL	NULL	<3,4>
4	NULL	NULL	NULL	NULL

Table 19: The example's initial cost matrix

	1	2	3	4
1	0	1	∞	3
2	∞	0	2	∞
3	∞	∞	0	2
4	∞	∞	∞	0

Table 20: The example's initial beginning of the active time period matrix

	1	2	3	4
1	0	1	∞	11
2	∞	0	5	∞
3	∞	∞	0	6
4	∞	∞	∞	0

Table 21: The example's initial end of the active time period matrix

	1	2	3	4
1	0	4	∞	14
2	∞	0	8	∞
3	∞	∞	0	11
4	∞	∞	∞	0

Table 22: The example's initial shortest delay matrix

	1	2	3	4
1	0	2	∞	14
2	∞	0	7	∞
3	∞	∞	0	8
4	∞	∞	∞	0

Since the SDIP routing algorithm is an iterative procedure, the only discussed runs are the ones when the reasoning portion of the algorithm is used. There is only three times where shortest delays are changed. Looking at the SDIP routing algorithm (see section 4.4), the path's shortest delay changes when the running process enters the code block from line 1418. The SDIP routing algorithm enter this code block when the values of parameters *k*, *i*, and *j*, defined at lines 5, 6, and 7 respectively, are set according to Table 23.

Table 23: The values where the example is affected

Run Number	k	i	j
1	3	2	4
2	2	1	3
3	3	1	4

In the first run, a new valid path is found in a result of the second validity condition where 1) the shortest delay of p_{23} is more than the beginning of p_{34}'s active time period and 2) the shortest delay of p_{23} path plus the delay cost p_{34} is less than the end of p_{34}'s active time period. Moreover, the combined path's shortest delay is shorter than the original p_{24}'s shortest delay. Thus, a new p_{24} is constructed and its related matrices' entries are set as follow:

$$p_{24} = p_{23} \oplus p_{34} = \langle 2,3 \rangle \oplus \langle 3,4 \rangle = \langle 2,3,4 \rangle \quad (19)$$

$$d_{24} = d_{23} + c_{34} = 7 + 2 = 9 \quad (20)$$

$$b_{24} = b_{23} = 5 \quad (21)$$

$$e_{24} = e_{34} = 11 \quad (22)$$

$$c_{24} = d_{24} - b_{24} = 9 - 5 = 4 \quad (23)$$

The illustration of the example's first run is shown in Figure 33.

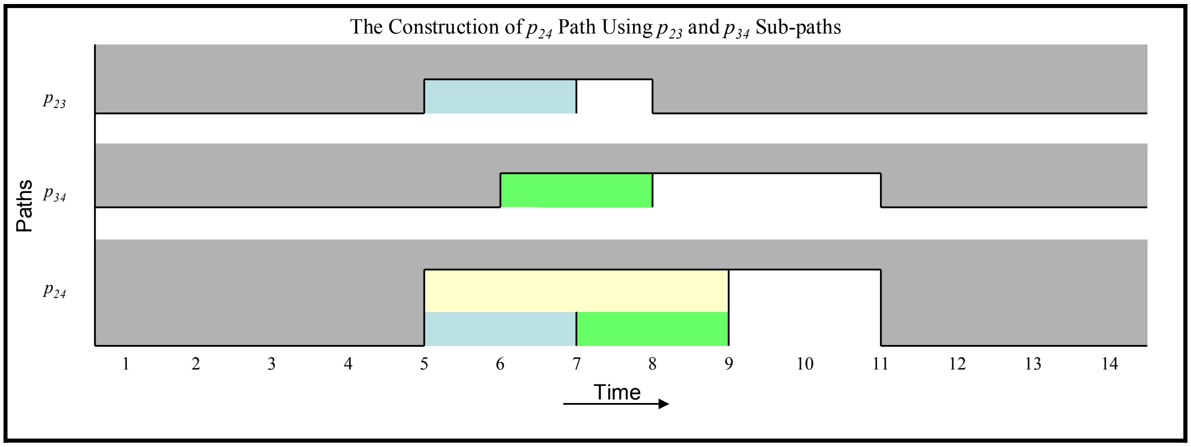

Figure 33: The example's first run process

In the second run, a new valid path is found in a result of the first validity condition where the shortest delay of p_{12} is less than the beginning of p_{23}'s active time period. Moreover, the combined path's shortest delay is shorter than the original p_{13}'s shortest delay. Thus, a new p_{13} path is constructed and its related matrices' entries are set as follow:

$$p_{13} = p_{12} \oplus p_{23} = \langle 1,2 \rangle \oplus \langle 2,3 \rangle = \langle 1,2,3 \rangle \tag{24}$$

$$d_{13} = d_{23} = 7 \tag{25}$$

$$b_{13} = b_{12} = 1 \tag{26}$$

$$e_{13} = e_{23} = 8 \tag{27}$$

$$c_{13} = d_{13} - b_{13} = 7 - 1 = 6 \tag{28}$$

The illustration of the example's second run is shown in Figure 34.

In the third run, a new valid path is found as a result of the second validity condition where 1) the shortest delay of p_{13} is more than the beginning of p_{34}'s active time period and 2) the shortest delay of p_{13} plus the p_{34}'s delay cost is less than the end of p_{34}'s active time period. Moreover, the combined path's shortest delay is shorter than the original p_{14}'s shortest delay. Thus, a new p_{14} path is constructed and its related matrices' entries are set as follow:

$$p_{14} = p_{13} \oplus p_{34} = \langle 1,2,3 \rangle \oplus \langle 3,4 \rangle = \langle 1,2,3,4 \rangle \tag{29}$$

$$d_{14} = d_{13} + c_{34} = 9 \tag{30}$$

$$b_{14} = b_{13} = 1 \tag{31}$$

$$e_{14} = e_{34} = 11 \tag{32}$$

$$c_{14} = d_{14} - b_{14} = 8 \tag{33}$$

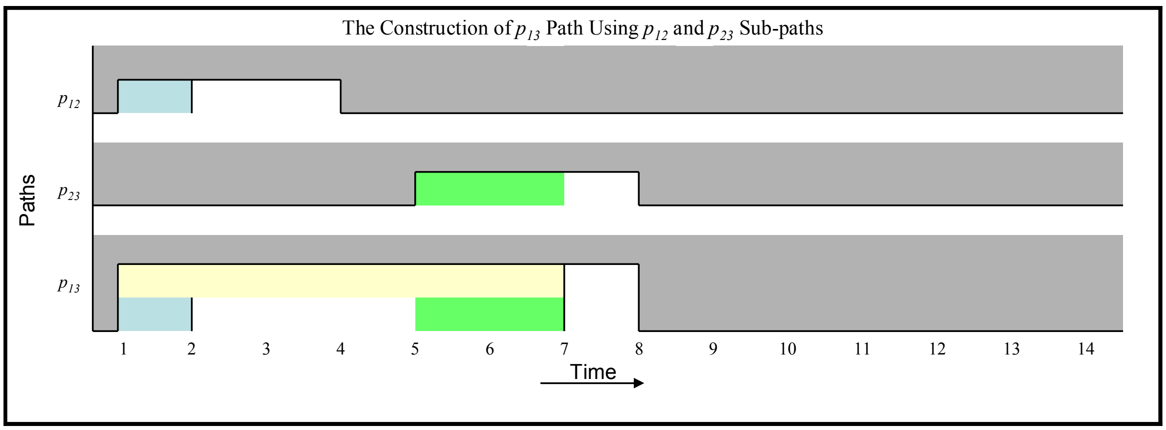

Figure 34: The example's second run process

The illustration of the example's third run is shown in Figure 35.

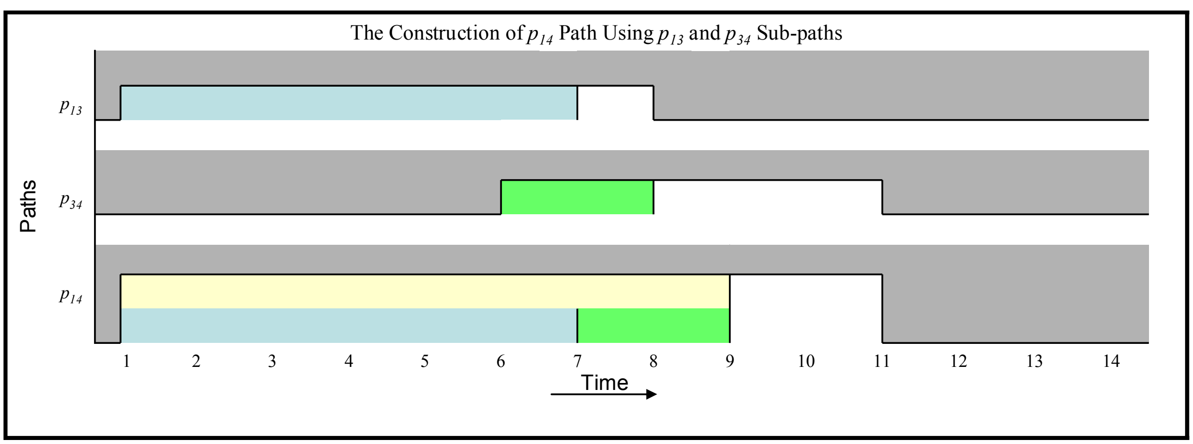

Figure 35: The third run of the example

Figure 36 shows the outcome of the SDIP routing algorithm where two new paths (p_{24} and p_{13}) are found and p_{14}'s shortest delay is reduced.

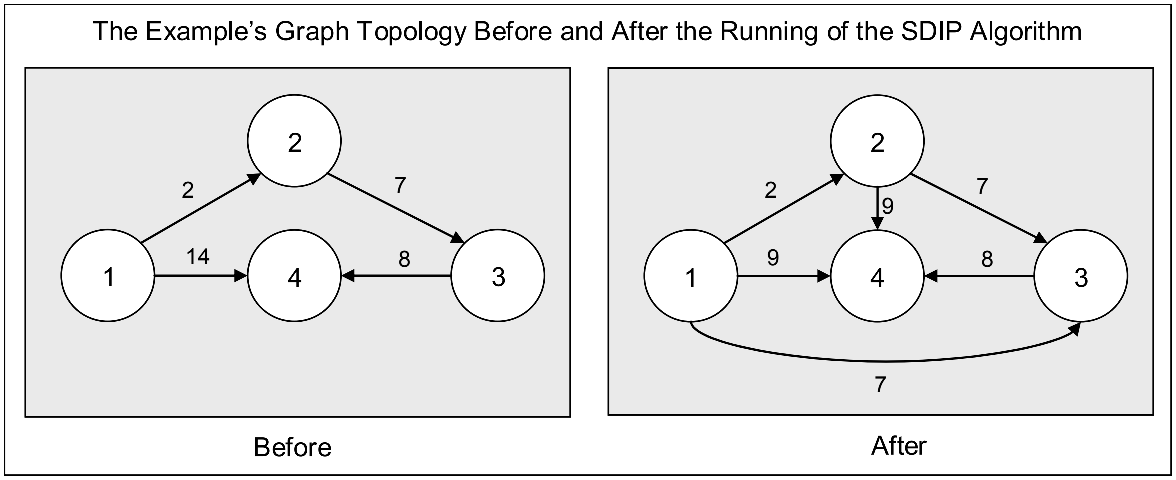

Figure 36: The SDIP algorithm's effect of example's graph topology

4.6 SDIP Routing Algorithm Analysis

The SDIP routing algorithm is a modified version of Floyd-Warshall shortest path algorithm [26]. Thus, it the has same running time complexity which is $\Theta(n^3)$ [19].

The SDIP algorithm is not like many shortest path algorithms where the routing table is $O(n)$. The SDIP algorithm's routing table is $O(n^2)$ because the whole path is stored in the routing table instead of the next hop. There are two transmission option when using the SDIP routing algorithm:

1. The packet includes the source and destination addresses only.

   It is assumed that the path matrix (P) is identical in all vertices (routers). Thus, all routers produce the same routing table. Moreover, when router z receives a packet from router x in its way to router y, router z looks up the p_{xy} entry and forwards the packet to the next router q after its own (z) in p_{xy} during its active time period with router q.

2. The packet includes the whole path.

   When router z receives a packet from router x in its way to router y, router z forwards the packet to the next router q after its own (z) in the path that is stored in the packet during its active time period with router q. At the time of this research, this option is made available using IPv6 Routing Header, the Routing Header [21].

The decision of using either option is highly dependable on the QoS requirements. If the first option is utilized, intermediate hops have to consume local processing power. In the second options, there is no processing power overhead but rather the packet is larger. The SOSPF routing algorithm is implementing the second option because of power limitations on board satellites [11].

4.6.1 SDIP Optimality

As mentioned previously, the SDIP routing algorithm produces approximate optimum paths. The reason that it is not optimum is because it may occur that the a link of two vertices, call them x and y, has an active time period that is smaller than the link's delay cost (see Figure 37), i.e.,

$$e(x, y) - b(x, y) < c(x, y) \qquad (34)$$

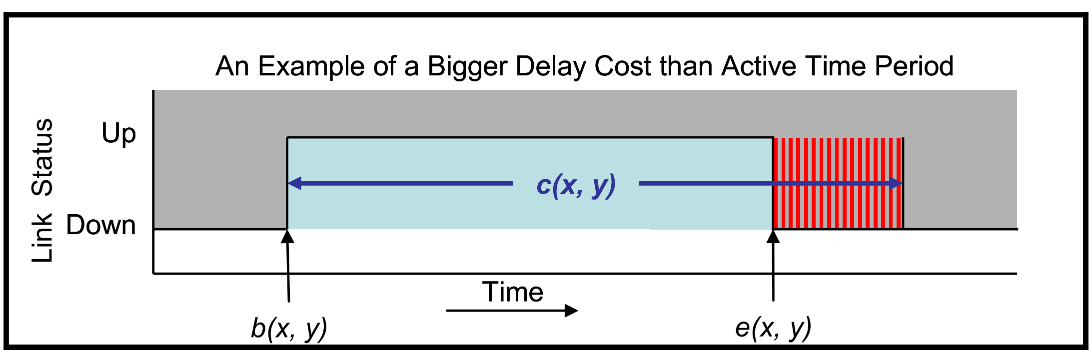

Figure 37: An Example of a bigger delay cost than the active time period

In this case, data may be propagated a head of time from vertex x before the (x, y) link is up in away that the propagated data reaches vertex y when (x, y) is up. Using this scenario, a shorter path in the topology may exist that the SDIP routing algorithm will not use. For this reason, the proposed SDIP routing algorithm is only an approximation routing algorithm.

The SDIP routing algorithm assumes that all delay costs are equal or less than their respected active time periods. Then the SDIP routing algorithm is optimum based on this assumption and the optimality is proved in the following Lemma.

Lemma:

Let p be the shortest delay path from u to v. Consider any two vertices x and y on this path. The part of the path between vertices x and y will be the shortest delay path between x and y. [22]

Proof:

If there was a sub-path from x to y that was not the shortest delay path from x to y, then we could replace this sub-path with the shortest delay path from x to y, obtaining a lesser cost for the overall path. This contradicts the statement that the path from u to v was the shortest delay path, so the lemma is true.

4.6.2 Link Change Effect

When a new link between two vertices, call them x and y, has emerged, the SDIP routing algorithm will not recalculate the routing table if the shortest delay for (x, y) is the highest of all links in L, i.e.,

$$d_{xy} \geq d_{ij} \forall i, j \in V \ni (i, j) \in L \tag{35}$$

Otherwise, a new link will cause the entire routing table to be recalculated. This is also true if the one of the existing links change its status (e.g., a new delay cost, a change in the active period time, or a dead link). If the shortest delay of the link in question used to have the highest delay of all links and no longer available or maintain it highest delay status, the routing table needs not to be recalculated.

4.7 SDIP Routing Algorithm in SOSPF

Source Satellite Address		5F00:0000:C001:0400::/56
Destination Satellite Address		5F00:0000:c001:2C00::/56
Number of tuples		3
Tuple # 1	Begin time	2006:08:28:20:14:50
	Connection Period	14400
	Propagation Delay	15
Tuple # 2	Begin time	2006:08:29:40:30:05
	Connection Period	2000
	Propagation Delay	20
Tuple # 3	Begin time	2006:08:30:23:14:50
	Connection Period	3000
	Propagation Delay	20

Figure 38: An example of an SR-LSA

The SOSPF routing protocol benefits from the SDIP routing algorithm since it provides the necessary information to run the SDIP routing algorithm. The input parameters of the graph *G* are obtained from the SR-LSAs in the SOSPF link state database (LSD). Each SR-LSA contains the three elements used to run the SDIP routing algorithm. An example of an SR-LSA is illustrated in Figure 38.

Although SR-LSAs may contain multiple tuples, it is assumed that the graph *G* will use the information of the tuple that is valid at the time of the routing table calculation.

CHAPTER 5

PERFORMANCE ANALYSIS

The SOSPF routing protocol is a chatty protocol by its inheritance nature of the original OSPFv3 [2]. Thus, SOSPF flooding is the main concern since it contributes greatly in how stable and scalable the SOSPF routing protocol is. Prior discussing the stability and scalability analysis, the SOSPF flooding mechanism and its traffic cost per link effect are explained.

5.1 Flooding

SOSPF flooding takes place whenever a new LSA or an LSA that is newer than router's link state database copy is received. Then, the LSA/s are encapsulated in a Link State Update packet and forwarded to a subset of neighboring routers.

In a satellite constellation area, an SOSPF router will forward the packet to (at most) two SOSPF routers in its satellite constellation area according to the internal flooding scope. (see 3.5) Thus, the internal flooding traffic cost in a satellite constellation area O(1)

5.1.1 Celestial Flooding Scope

All SOSPF routers in celestial object areas forward the Link State Update packet according to the celestial flooding scope. Let *n* be the number of SOSPF routers in a celestial object area in question. Initially, the SOSPF router that generates the packet will forward the packet to n-1 SOSPF routers. In the worst case, each SOSPF router will forward the packet to n-2 SOSPF routers according to the celestial flooding scope (see 3.5). The packet will not be forwarded anymore because that each router has already a copy of the LSAs in the Link State Update packet. Thus, the celestial flooding traffic cost per link per is O(n).

5.2 Stability

The term Stability can vary by definition from one research to another. For example, in [8], stability is evaluated by a mixer of 1) network convergence period, 2) routing load on processors, and 2) the number of route flaps. Furthermore, others researches have defined it differently [16], [60],[65]. In this dissertation, stability is defined as follow:

Stability: *Stability is the degree to which a system can maintain valid connectivity when disruptions occur.*

Primarily, the convergence period is measured between a pair of "network change" events. In SOSPF, when a link state changes or a link failure is detected, an LSA is generated and encapsulated in a Link State Update packet where it is flooded to the affected areas. Every link update that occurs in the network will require a *convergence period* before the network enters a *stability period*.

The convergence period: *The convergence period is the amount of time for a router to restore normal traffic handling after recovery from a network interruption.*

The stability period: *The stability period starts at the time when all SOSPF routers have valid routing table until the next network interruption.*

Figure 39 Illustrates a periodical cycle between convergence periods and stability periods.

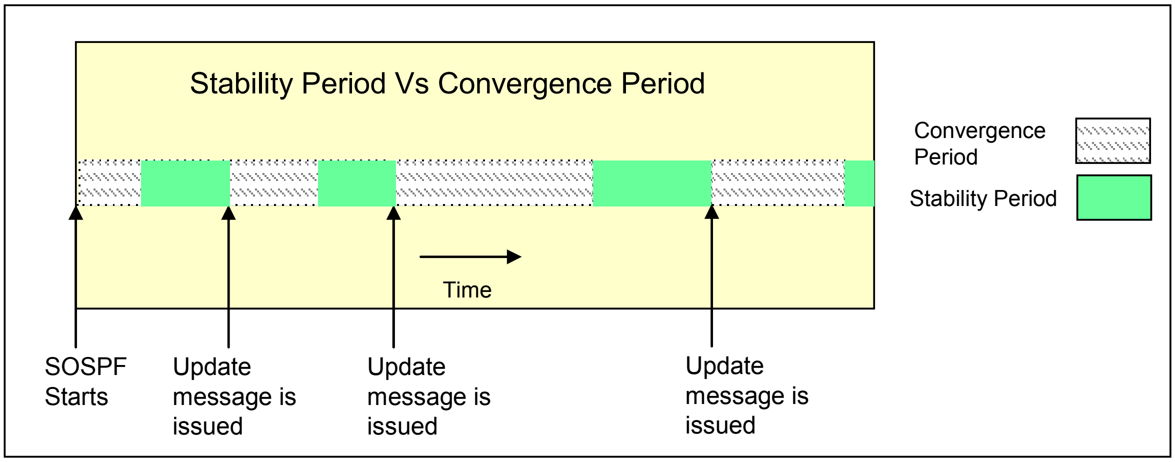

Figure 39: Sample period with links failures

- Let t_1 and t_4 be the clock times of two consecutive network interruptions.
- Let F be the time interval between the two network interruptions, i.e.,

$$F = t_4 - t_1 \tag{36}$$

- Let t_2 be the clock time where at least one SOSPF router becomes aware that a network interruption happened at t_1.
- Let D be the detection time interval for the interruption that happened at t_1, i.e.,

$$D = t_2 - t_1 \tag{37}$$

- Let t_3 be the clock time where all SOSPF routers have valid routing tables.
- Let P be the time interval required for all SOSPF router to have valid routing tables from the moment that the interruption was detected, i.e.,

$$P = t_3 - t_2 \tag{38}$$

Figure 40 shows all parameters that have been defined. Although there are many ways of measuring the stability, in this dissertation, the measurement of stability is the ratio of time where the network is in a stable period against the period between network interruptions.

- Let λ be the stability i.e.,

$$\lambda = \frac{F-(D+P)}{F} \tag{39}$$

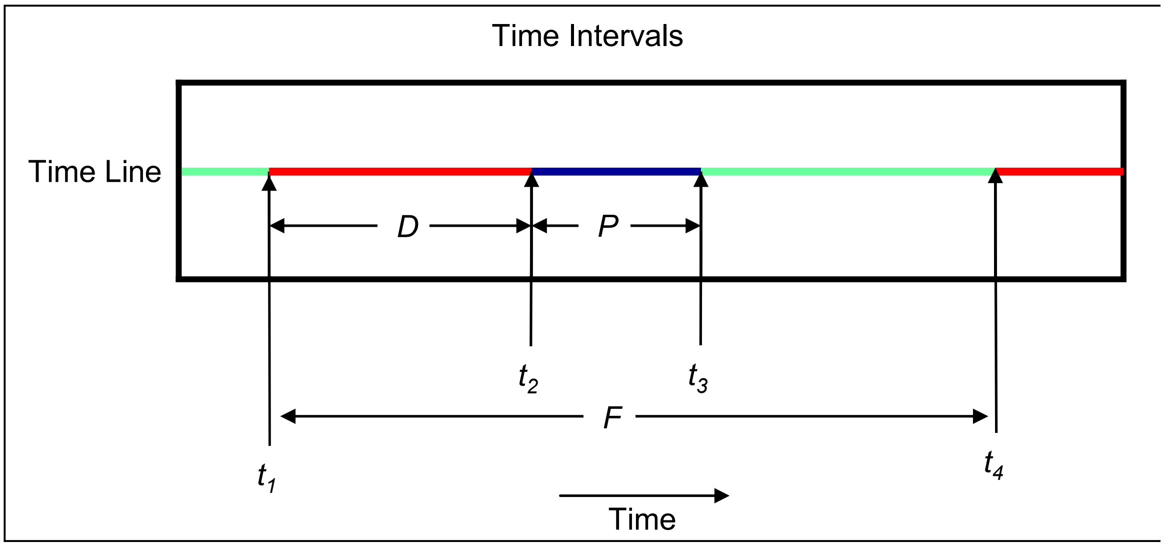

Figure 40: Time Interval layout

5.2.1 Stability Analytical Model

The model defines a satellite network as a connected directed graph $G = (V, E, A)$, where V is the set of vertices (routers on board satellites), E is the set of edges (direct links), and A is a set areas. Let n be the number of routers in V. Let the link connecting router x to router y in E be represented as (x, y). A cost function $c: E \rightarrow R$ is defined, where R is the set of real numbers, determines the cost of (x, y) if $(x, y) \in E$. Let the cost function associated with (x, y) be represented as $c(x, y)$. The cost is a delay metric which is the estimated propagation delay between a pair of routers. An area function $a: V \rightarrow A$ is defined, determine the set of areas of router x if $x \in V$. Let the area function associated with router x be presented as $a(x)$.

Let router x and router y belong to a common area, i.e.,

$$a(x) \cap a(y) \neq \theta \text{ and } (x, y) \in E \tag{40}$$

Thus, hello packets are exchanged between router x and router y at regular intervals (see section 3.6) which is equal to $c(x, y)$. If router x fails, router y will detect the failure when it doesn't receive a hello packet from router x after k multiples of $c(x, y)$. In this dissertation, it is assumed that k equals 1. Let D^{SOSPF} be the biggest link failure detection interval in the SOSPF routing protocol where router x and router y are the furthest two closest routers who belong to one common area, i.e.,

$$D^{SOSPF} = \max(\min_{y=1,x=1}^{n}(c(x, y)) \text{ where } a(x) \cap a(y) \neq \theta \tag{41}$$

Since SOSPF routers are configured in a hierarchal area structure, not every network interruptions have to be propagated throughout the entire routing domain. If the router x which has its *DoNotAdvertise* bit set on fails, only the members of its area learn about its failure.

Furthermore, let P^{SOSPF} the biggest time interval required to have valid routing tables from the moment that router y detected the failure of router x in SOSPF. Let router z and router y be the furthest two routers where router x, router y, and router z belong to one common area. P^{SOSPF} is set to be the summation of propagation delays between the two furthest router in all areas where the failed router x occurred, i.e.,

$$P^{SOSPF} = \sum \max_{y=1,z=1}^{n}(c(y, z)) \text{ where } a(y) = a(z) = a(x) \tag{42}$$

Using formula 38 above, let λ^{SOSPF} be the stability of the network for SOSPF, i.e.,

$$\lambda^{SOSPF} = \frac{F - \left(\max(\min_{y=1,x=1}^{n} (c(x,y)) + \sum \max_{y=1,z=1}^{n} (c(y,z)) \right)}{F} \text{ where } a(x)=a(y)=a(z) \quad (43)$$

In the other hand, currently, ground station on Earth send beacon signal periodically to maintain connectivity with satellites [23], hereafter referred to as referenced technology (*RT*). Therefore, the link failure detection interval for the referenced technology is the roundtrip propagation delay of a beacon signal. Let *g* be the Earth ground station and D^{RT} be the biggest link failure detection interval which is the propagation delay between the furthest router *x* to the Earth ground station *g*, i.e.,

$$D^{RT} = \max_{x=1}^{n}(c(g,x)) \quad (44)$$

In the reference technology, the time interval required to rebuild the routing table is the detection interval which the propagation delay between the furthest router *y* to the Earth ground station *g*. Let P^{RT} the biggest time interval required to rebuild the routing table for the referenced technology, i.e.,

$$P^{RT} = \max_{y=1}^{n}(c(g,y)) \quad (45)$$

Also, using formula 38 above, Let λ^{RT} be the stability of the network for referenced technology, i.e.,

$$\lambda^{RT} = \frac{F - 2 * \left(\max_{x=1}^{n}(c(g,x)) \right)}{F} \quad (46)$$

Table 24: Stability of the network

Stability Value	Description
$\lambda=1$	The network is free of any updates and there is no convergence period
$1>\lambda>0$	The network is stable for a period of time. The closer λ is to 1, the more stable the network
$\lambda \leq 0$	The network is not stable. The updates are occurring before the previous convergence period is done.

As can be seen from Table 24, the closer the stability value gets to 1 the more stable the network is. The stability plays a major role on deciding whether SOSPF is a feasible solution for space networking. Since there is no statistics about how network interruptions occur between routers on board satellites, there is no valid randomization mechanism which will create a real-life scenario in space. However, a number of simulation runs are shown with current or purposed satellite constellation configurations in CHAPTER 6.

5.3 Scalability

The most critical scalability issue is how much overhead an individual SOSPF router will incur when the number of SOSPF routers increases. The most significant factor that contributes to this overhead is the amount of protocol traffic generated per link.

The next question is how does traffic scale with the number of SOSPF routers? The answer to this question is highly dependent on the topology of the network. An SOSPF router x, which belongs to the backbone area, may belong to an area at each level of the area hierarchy (see section 3.1.1.2) where *x* floods each area it belongs to when a new LSA is received (see section 3.5). Thus, in the naive case, where every SOSPF router is a member the backbone area, the SOSPF routing protocol performs poorly.

With intelligent placement where the number of SOSPF routers belonging to areas at multiple levels is kept small, the traffic per SOSPF router will remain feasible.

Let *C* be the subset of *A* where *C* contains the celestial object areas and *c* be the number of areas in *C*. Let ac_i be the i^{th} area in *C* and m_i be the number of SOSPF routers in area ac_i. Let *S* be the subset of *A* where *S* contains the satellite constellation areas and *s* be the number of areas in *S*.

Let β be the number of propagations in the event of a network change that must be received by all SOSPF routers. The flooding traffic cost per SOSPF router in a satellite constellation area is O(1); and flooding traffic cost per SOSPF router in a celestial object area is O(*n*) (where n is the number of SOSPF routers in the area) (see 5.1.1). of Thus β is:

$$\beta = s + \sum_{i=1}^{c} m_i \text{ where } ac_i \in C \tag{47}$$

Depending on how new SOSPF routers are configured, β can get large very quickly. In the worst case, if every new SOSPF router is added to the backbone area, then β becomes O(*n*l*) where *l* it the number of levels in the area hierarchy.

In addition, scalability has a direct correlation with stability since any increase of SOSPF routers in all areas will prolong the convergence period which will decrease the stability ratio. Thus, SOSPF router area placement must be done with great care.

CHAPTER 6

SIMULATION PERFORMANCE

In this research, three types of performances are evaluated, 1) stability, 2) scalability, and 3) end-to-end delay.

6.1 Stability

As defined in section 5.2, the degree of the stability of the network depends on which SOSPF router fails. Figure 41 shows an example configuration in space where there are six satellite constellations areas, two for the Moon, two for Earth, and two for Mars. The SOSPF area hierarchal is shown in Figure 42 and the description and the members of each area is given in Table 25.

In Table 25, EM1, an SOSPF router which belongs to the Earth-Moon satellite constellation area, is a member of areas at four levels, 1:3:1:1, 1:3:1, 1:3, and 1. Thus, a change of the link state of EM1 will result to traffic propagation in all four areas. The number of area levels each SOSPF router belongs to plays a major role in the stability of the network.

The next section shows how stability is affected by a network interruption at various levels of the solar system area hierarchy

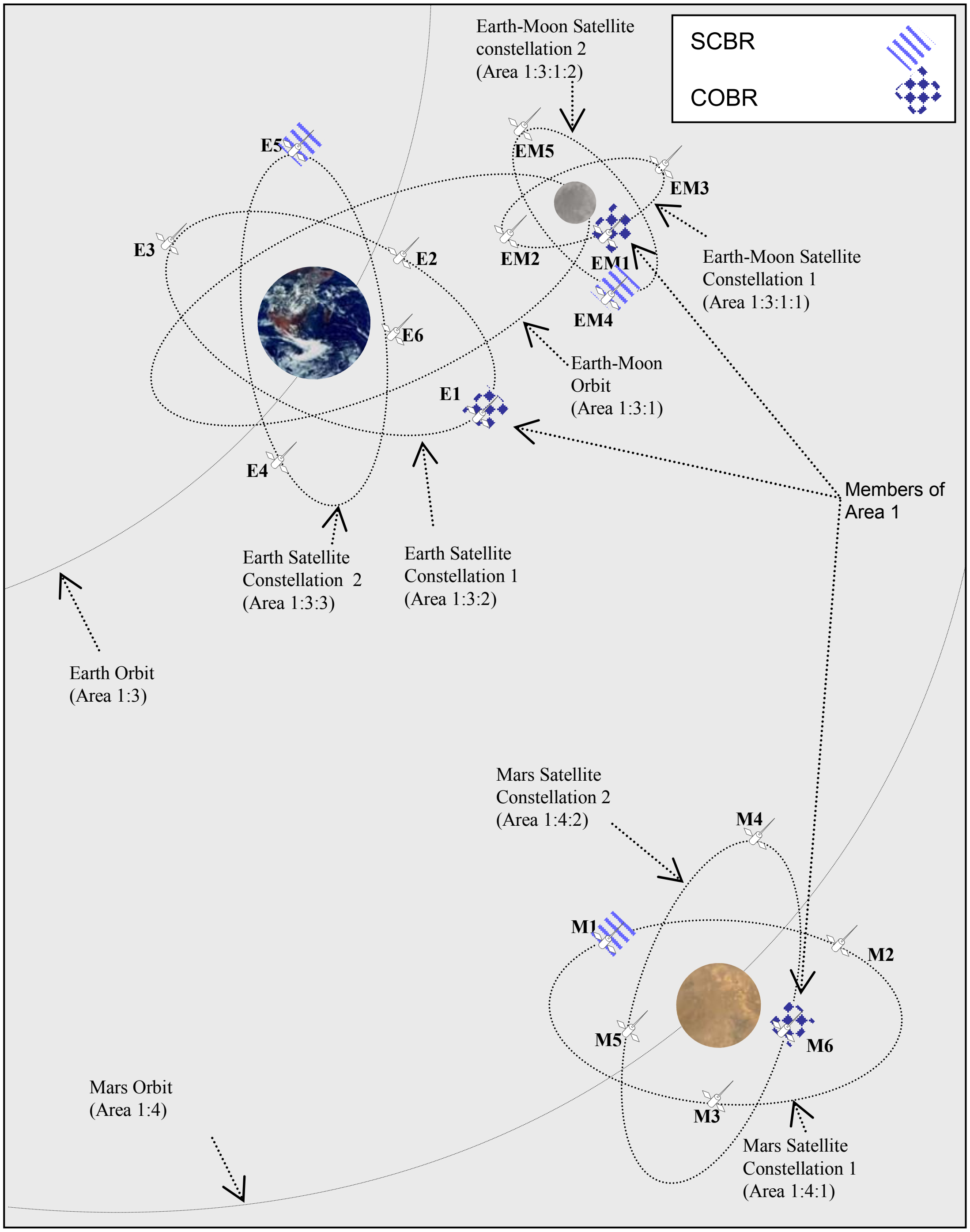

Figure 41: A sample configuration in space

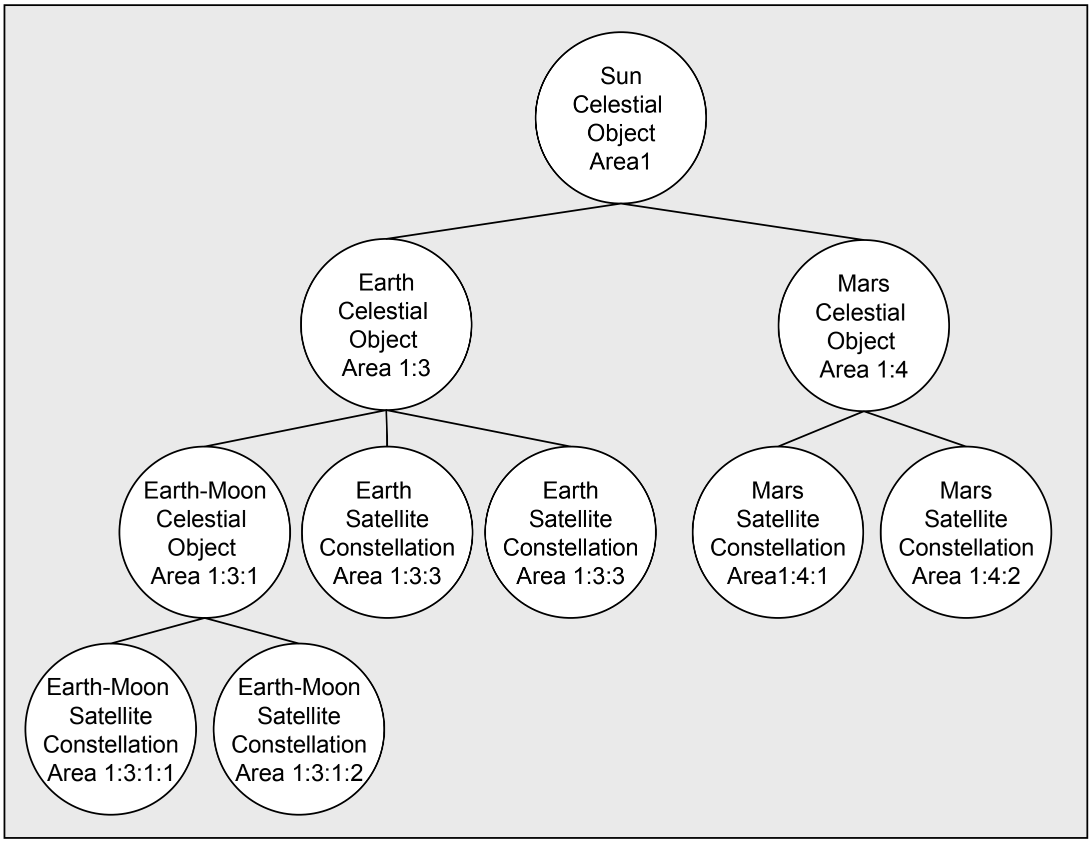

Figure 42: Area hierarchal for the sample configuration

Table 25: The sample configuration's area membership

Area	Members
1	E1, M1, and EM1
1:3	E1 and E3, and EM1
1:4	M1 and M6
1:3:1	EM1 and EM4
1:3:2	E1, E2, and E3
1:3:3	E4, E5, and E6
1:4:1	M1, M2, and M3
1:4:2	M4, M5, and M6
1:3:1:1	EM1, EM2, and EM3
1:3:1:2	EM4 and EM5

6.1.1 Stability Experiment

In an abstract view, the following setup is used:

- A satellite network with a diameter of 1013 km.
- 5000 SOSPF routers are scattered on the orbits of the eight planets and on the orbit of the Earth-Moon using a uniform random function
- The failure interval is 6666 seconds.

Figure 43 shows the stability of the network after a network interruption which occurred at an SOSPF router using stability formula 43 defined in section 5.2.1. In the worst scenario, SOSPF routers which belong to the backbone area (which means that they belong to areas at four levels) have poor performance. In the contrary, SOSPF routers that belong to area at three levels or less maintain a high stability value even at a network diameter which reaches Neptune.

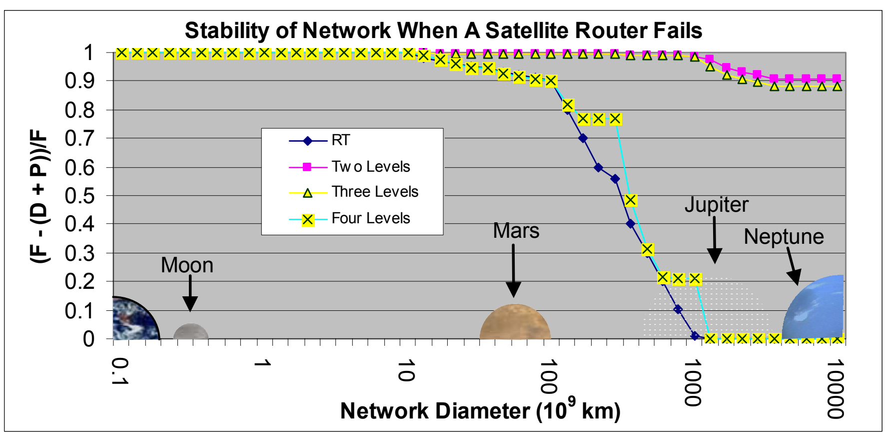

Figure 43: The stability of the network after a failure

Furthermore, if a link failure occurred and must be propagated throughout the routing domain, it has the same effect as the worst scenario. Even in the worst scenario, SOSPF's stability did not go lower than the reference technology's stability.

6.2 Scalability

To measure the scalability performance of the SOSPF routing protocol, scalability formula is applied for incremental count of SOSPF routers up to three thousand SOSPF routers. This evaluation measures how many propagated packets are issued per SOSPF router when a link update occurs using a number of scenarios defined in Table 26 and plot them in Figure 44 using formula 47 defined in section 5.3. In Table 26, each scenario reflects on how many SOSPF routers belong to areas at multiple levels. For example in scenario four, the number of SOSPF routers that belong to areas at four levels is the total number of SOSPF routers divided by forty eight.

Table 26: Scenarios of scalability analysis

Scenario #	1 Level	2 Levels	3 Levels	4 Levels (Backbone)
1	n	n	n	n
2	n	n/2	n/2	n/2
3	n	n/4	n/8	n/16
4	n	n/24	n/36	n/48

The worst scenario (scenario 1) in SOSPF is when each added SOSPF router belongs to the backbone. As can be seen from Figure 44, with just moderate modification in the SOSPF router placement, the number of propagated packets per SOSPF router drops significantly. Furthermore, with Intelligent assignment, as in scenario 4, the number of propagated packets per SOSPF router will remain at a low level. Careful planning is required when adding a new SOSPF router to conclude that SOSPF is scalable.

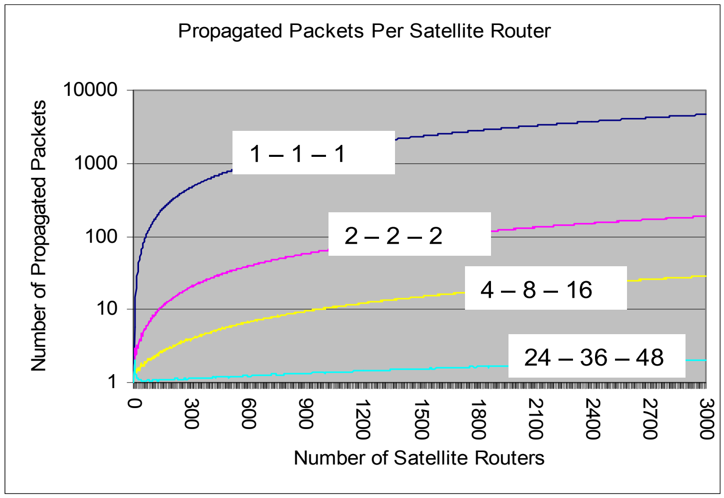

Figure 44: Traffic after a network interruption

6.3 End-To-End Delay

The end-to-end delay performance study of the SOSPF routing protocol is an example of a future satellite network scenarios for various plausible earth-mars communications. In this simulation study, SOSPF routers area positioned on various Mars and Earth orbits where the performance on an incremental infrastructure scenarios around Earth and Mars is presented.

The simulation makes use of real satellites' parameter proposed or currently used by NASA [69]. The time and date of the simulation is for one hour from 2002, August 1, 0:0:0 to 2002, August 1, 1:0:0[8]. In the simulation, there are 1000 packets to be propagated from one SOSPF router in an Earth satellite constellation area to an SOSPF router in Mars satellite constellation area. The packets sizes are from 112KB to 128KB. All satellites have a bandwidth of 128Kbps.

[8] the choice of time and date is arbitrary

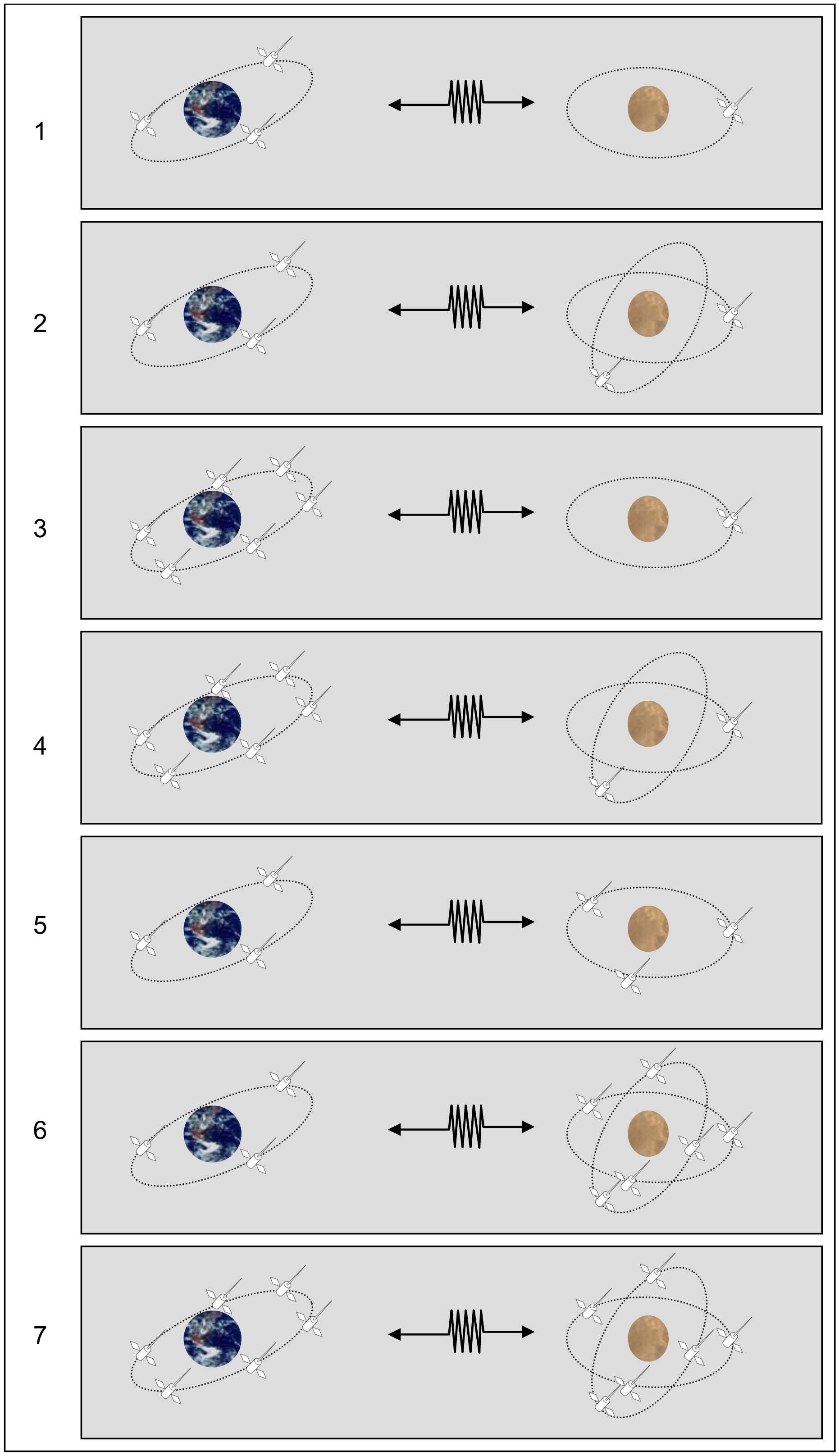

Figure 45:The seven simulation scenarios

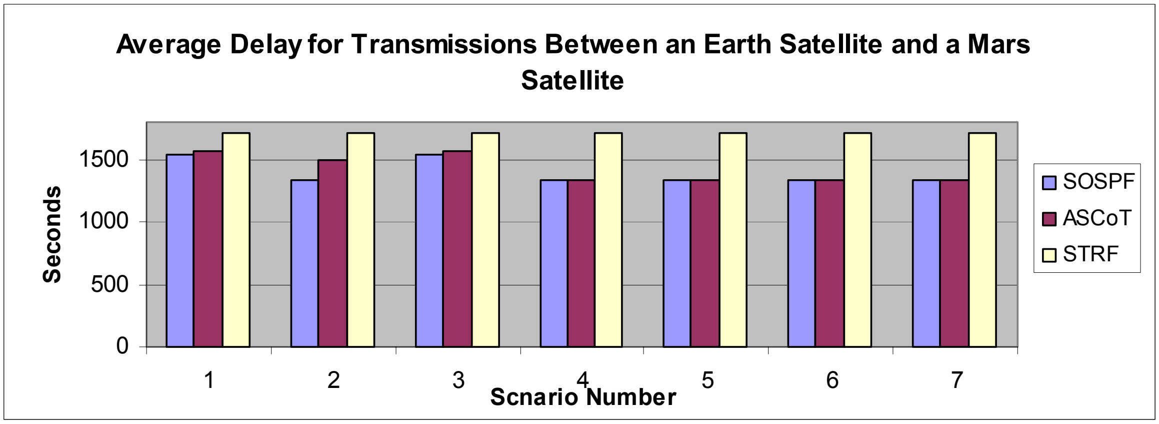

Figure 46: The average end-to-end delay

The simulation is run with seven scenarios which are illustrated in Figure 45. Each scenario is a combination of a number of SOSPF routers orbiting Mars and Earth.

This evaluation is comparing the proposed shortest delay intermittent pathway (SDIP) routing algorithm with two other space routing algorithms described earlier, ASCoT routing algorithm [34] and space-time routing framework (STRF) routing algorithm [47].

Figure 46 shows the average end-to-end delay where in scenario one and three, the SDIP routing algorithms shows improvement over the other two algorithms.

In scenario one, there is only one SOSPF router orbiting Mars which means that with limited resources, SOSPF performs better than the other algorithms. Moreover, just by adding one more SOSPF router around Mars in a new satellite constellation area, the SDIP improves even more over the other two algorithms.

On the other hand, when there are three or more SOSPF routers orbiting Mars and Earth, the SDIP and ASCoT space routing algorithm performs exactly the same. STRF did not perform well in all of these scenarios because that the hop count is the main routing decision factor and hops are treated equally.

In this research, it is concluded that the same goals are achieved with a limited number of routers on board satellites as if higher number of them exists.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Routing in space differ from Earth-like routing given the continuous relative motion of communication satellites with respect to each other and the volatile nature of the space environment. To address these design issues this research propose a routing protocol, Space Open Shortest Path First (SOSPF), which has algorithms which are tailored to fit this space environment.

7.1 Contributions

This research comprises the Space Open Shortest Path First (SOSPF) space communication protocol and the Shortest Delay Intermittent Pathway (SDIP) routing algorithm which is part of the SOSPF communication stack.

SOSPF has the following key features:

1. *Define Logical Areas:* The protocol divides the Solar System which includes celestial objects and spacecrafts into logically separate areas. Within each area, traffic cost is maintained and the routing information of one area can be summarized and propagated outside the area with reduced overhead. Further security is ensured by preventing intruders inside area whose traffic must be secured.

2. *Predictable Mobility*: In order to improve the accuracy of the routing tables, SOSPF incorporates a spacecrafts mobility prediction component that helps extrapolate and pre-calculate routing tables ahead of time, for faster communication.

3. *Distribute Routing Protocol*: The proposed routing protocol is first distributed routing protocol in a space environment. It encompasses dynamic error detection in space and maintain up-to date routing information which includes, but is not limited to changes in spacecraft trajectory, available bandwidth, or any parameter that can affect the spacecrafts routing capability.

The research's second contribution is the designing of the Shortest Delay Intermittent Pathway (SDIP) routing algorithm which provides scheduling solutions for intermittent links. Not only does the proposed SDIP routing algorithm handle routing in the space routing domain well, but it can also be used for any routing domain that suffers intermittent yet predictable disconnection. One example of this would be a network where certain routers are configured to be down at certain times.

7.2 Limitations and Future Work

The SOSPF routing protocol requires additional features which are not covered in this research. Those features can be categorized as the following:

- Defining *Neighboring Routers List*

  At the time of this research there is no specification on how to configure the *Neighboring Routers List* which specifies which areas an SOSPF router belongs to and for how long. The configuration of this list requires optimal placement heuristics which involves time and location in space which is a prime research topic in the near future for the SOSPF routing protocol to perform efficiently.

- Inserting new SOSPF router

  The original OSPF protocol is a chatty protocol which means that it could congest the routing domain with many flooding messages. SOSPF follows the same footsteps of OSPF in this regard. As the number of SOSPF routers increase, the network becomes more vulnerable to updates. In a naïve method, if all new SOSPF routers added to all celestial object areas possible, the number of message when an update message is issued will grow exponentially. A future research, which allows new SOSPF router to be added intelligently, will maintain a linear growth of the number of message when an update message is issued.

- Added security

  In SOSPF, members of one area can secure a few of their parameters using AM-LSA which advertises connectivity information prior joining an area. This scenario is the only case where security is thought of in this research. Although security remains a critical issue in space, security was not a prime concerned at the time of this research. Added security is future research topic which has to be dealt with.

- Support for big packets

  In 4.6.1, it is illustrated that packet *s*, that has a delay that is more than the active time period of path *p*, will not use the *p* in its routing decision. There a number of way to resolve this issue:

1. Break the packet into smaller pieces , or
2. Start transmitting *s* before *p*'s active time period is up in a way that *s* arrives at its destination during *p*'s active time period.

These solution and others may be future research topic.

- Defining space colony data structure

  The configuration and definition of space colony was specified in this research. In the near future, space colonies and the devices within them have to be defined. Moreover, the interaction between space colonies and other spacecraft has to be defined as well.

Further research might be implemented along the above directions.

REFERENCES

[1] AL-JAMMAZ, K., DIAZ, A., HERNANDEZ, F., IYER, S., ORTIZ, C., RICHARDSON, B., RODRIGUEZ, M., WHITESIDES, AND G., THANGAVELU, M. 2003. Elements for a sustainable lunar colony in the south polar region. In *Proceedings of the International Lunar Conference 2003* (Waikoloa, Hawaii, United States). American Astronautical Society and Space Age Publishing Co. by Univelt Inc., San Diego CA, 2004, 515-523.

[2] BAKER, F., CHANDRA, M., WHITE, R., MACKER, J., HENDERSON, T., AND BACCELLI, E. 2003. Problem statement for OSPF extensions for mobile ad hoc routing. IETF Internet Draft, September, 2003. http://www.ietf.org/internet-drafts/draft-baker-manet-ospf-problem-statement-00.txt

[3] BALL, J.R., AND EVANS C.H. 2001. *Safe Passage: Astronaut Care for Exploration Missions*. National Academy Press.

[4] BANTAN, N. 2006. Space OSPF. In *the Fifth Space Internetworking Workshop – presentation paper* (Baltimore, Maryland, United States). September 2006. http://www.hynet.umd.edu/Space_Internet_Workshop/pres/A004.pdf

[5] BANTAN, N. AND KHAN, J. 2007. Shortest path first routing algorithm for space. To appear in *Proceedings of The 25th AIAA International Communications Satellite Systems Conference* (Seoul, South Korea). April 2007.

[6] BANTAN, N. AND KHAN, J. 2007. SOSPF- a new routing protocol for space. To appear in *Proceedings of The 25th AIAA International Communications Satellite Systems Conference* (Seoul, South Korea). April 2007.

[7] BANTAN, N. AND KHAN, J. 2006. Space OSPF - shortest delay intermittent pathway routing with mobile routers. In *the Fifth Space Internetworking Workshop – presentation paper* (Baltimore, Maryland, United States). September 2006. http://www.hynet.umd.edu/Space_Internet_Workshop/pres/A008.pdf

[8] BASU, A. AND RIECKE, J. 2001. Stability issues in OSPF routing. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications* (San Diego, California, United States). SIGCOMM '01. ACM Press, New York, NY, 225-236.

[9] BELL, P.R., AND JABBOUR K. 1986. Review of point-to-point network routing algorithms. *IEEE Communications Magazine* 24, 1 (January 1986), 34–38.

[10] BLACK, U. 1995. *TCP/IP &Amp; Related Protocols (2nd Ed.)*. McGraw-Hill, Inc.

[11] BOARD ON PHYSICS AND ASTRONOMY (BPA) AND ENGINEERING AND PHYSICAL SCIENCES (DEPS) 2006. *Handbook of Frequency Allocations and Spectrum Protection for Scientific Uses*. The National Academic Press.

[12] BOARD ON PHYSICS AND ASTRONOMY (BPA) AND ENGINEERING AND PHYSICAL SCIENCES (DEPS) 2004. *Solar and Space Physics and Its Role in Space Exploration*. The National Academies Press.

[13] CAESAR, M., AND REXFORD, J. 2005. BGP routing policies in ISP networks. *IEEE Network Magazine* 19, 6 (November 2005), 5–11.

[14] CHANG, H.S., KIM, B.W., LEE, C.G., MIN, S.L., CHOI, Y., YANG, H.S., KIM, D.N., AND KIM, C.S. 1998. FSA-based link assignment and routing in low-Earth orbit satellite networks. *IEEE Transactions on Vehicular Technology* 47, 3 (August 1998), 1037–1048.

[15] CHANG-HSU, C., AND WU, E.H. 2003. AMRST: adaptive multicast routing protocol for satellite–terrestrial networks. *Computer Networks* 43, 6 (December 2003), 713–734.

[16] CHOUDHURY, G. L., MAUNDER, A. S., AND SAPOZHNIKOVA, V. D. 2001. Faster link-state IGP convergence and improved network scalability and stability. In *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks* (Tampa, Florida, United States). LCN ’01. IEEE Computer Society, Washington, DC, 149-158.

[17] CLAYTON, D. 2006. Travel to Mars via asteroid? *Colony World* (October, 2006). http://www.colonyworlds.com/2006/10/travel-to-mars-via-asteroid.html.

[18] COLTUN, R., FERGUSON, D., AND J. MOY. 1999. *OSPF for IPv6*. RFC. RFC Editor.

[19] CORMEN, T. H., LEISERSON, C.E., RIVEST, R.L. 1990. *Introduction to Algorithms, (1st edition)*, MIT Press and McGraw-Hill.

[20] D'ARRIGO, P., AND SANTANDREA, S. 2006. APIES: Exploring the asteroid belt with satellite swarms. In *Proceedings of the first Workshop on Innovative System Concepts – presentation paper* (Noordwjik, The Netherlands). February 2006. http://www.esa.int/gsp/ACT/doc/ACT-PRE-4200-InnSys-DArrigo.pdf

[21] DEERING, S. AND HINDEN, R. 1998. *Internet Protocol, Version 6 (IPv6) Specification*. RFC. RFC Editor.

[22] DEMETRESCU, C., AND ITALIANO, G. 2004. Engineering shortest path algorithms. In *Proceedings of the 3rd International Workshop on Experimental and Efficient Algorithms* (Angra dos Reis, Brazil). WEA'04. 191-198.

[23] DURRESIA, A., DASH, D., ANDERSON, B. L., KANNAN, R., KOTA, S., AND JAIN, R. 2004. Routing of real-time traffic in a transformational communications architecture. In *Proceedings of the 20004 IEEE Aerospace Conference* (Big Sky, Montana). 1086-1104.

[24] FERGUSON, H. C. 2004. Robotic infrastructure for Lagrange-point observatories. In *the NASA for their Strategic Capabilities Roadmap Workshop – White Paper*. (Washington, District of Colombia, United States). http://www-int.stsci.edu/~ferguson/Ferguson/moon-mars_files/LL1robot.pdf

[25] FERREIRA, A., GALTIER, J., AND PENNA, P. 2001. *Topological Design, Routing And Handover In Satellite Networks*, John Wiley & Sons, Handbook of Wireless Networks and Mobile Computing.

[26] FLOYD, W. 1962, Algorithm 97: shortest path. *Communications of the ACM* 5, 6 (June 1962), 345–370.

[27] FRANCK, L., AND MARAL, G. 2000. Candidate algorithms for routing in a network of inter-satellite links. In *Proceedings of the 18th American Institute of Aeronautics and Astronautics* (Oakland, California, United States). 805–815.

[28] FRANCK, L., AND MARAL G. 2002. Signaling for inter-satellite link routing in broadband non-GEO satellite systems. *Computer Networks* 39, 1 (May 2002), 79–92.

[29] FRANCK, L., AND MARAL, G. 2002. Static and adaptive routing in ISL networks from a constellation perspective. *International Journal Of Satellite Communications* 20, 6 (November/December 2002), 455–475.

[30] FU, A., MODIANO, E., AND TSITSIKLIS, J. N. 2003. Optimal energy allocation and admission control for communications satellites. *IEEE/ACM Trans. Netw.* 11, 3 (June 2003), 488–500.

[31] GANGALE, T. 2005. Marssat: assured communication with Mars. In *New Trends in Astrodynamics*, Annals of the New York Academy of Sciences, annual Princeton conference 1065, 296-310.

[32] GARVIN, J. 2004. Special: sibling rivalry: a Mars/Earth comparison. *NASA Goddard Space Flight Center: Scientific Visualization Studio.* (April 2004). http://svs.gsfc.nasa.gov/stories/mars_earth_20040421/index.html.

[33] GAVISH, B., AND KALVENES, J. 1997. The impact of intersatellite communication links on LEOs performance. *Telecommunication Systems Journal* 8, 2–4 (1997), 159–190.

[34] GNAWALI, O., POLYAKOV, M., BOSE, P., AND GOVINDAN, R. 2005. Data centric, position-based routing in space networks. In *Proceedings of the 26th IEEE Aerospace Conference* (Big Sky, Montana, United States). 1322- 1334.

[35] GUMMADI, K., GUMMADI, R., GRIBBLE, S., RATNASAMY, S., SHENKER, S., AND STOICA, I. 2003. The impact of DHT routing geometry on resilience and proximity. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications* (Karlsruhe, Germany, August 25 - 29, 2003). SIGCOMM '03. ACM Press, New York, NY, 381-394.

[36] HUITEMA C. 2000. *Routing in the Internet* (2nd edition), Prentice-Hall.

[37] HUITEMA C. 1995. *Routing in the Internet*, Prentice Hall.

[38] IVANCIC,W., STEWART, D., SHELL D., WOOD, L. PAULSEN, P., JACKSON, C., HODGSON D., NORTHAM, J., BEAN, N., MILLER, E., GRAVES, M., AND KURISAKI, L. 2005. Secure, network-centric operations of a space-based asset: Cisco router in low Earth orbit (CLEO) and virtual mission operations center (VMOC). *The NASA Technical Report Server*. Technical Report. Report Number: NASA/TM—2005-213556., NASA Glenn Technical Report Server.

[39] JET PROPULSION LABORATORY AT CALIFORNIA INSTITUTE OF TECHNOLOGY 1999. Mars Climate Orbiter/Mars Polar Lander Mission Overview. *Mars Polar Lander: Searching for Water on Mars*. http://mars.jpl.nasa.gov/msp98/mission overview.html.

[40] KANDUSN, G., SVIGELJZ A., AND MOHORCIC, M. 2004. The impact of different scheduling policies on traffic class dependent routing in intersatellite link networks. *International Journal Of Satellite Communications and Networking*, 22, 5 (September/October 2004), 533–546.

[41] KHAN, J., ELLIS, B., MCFARLAND L., PUTT, C. 2005. A commentary on the nature of mobility and connectivity in space and implications on the design of space network protocols. *The NASA Technical Report Server*. Technical Report. Report Number: TR2005-08-01., NASA Glenn Technical Report Server.

[42] KRISHNA, P., VAIDYA, N. H., CHATTERJEE, M., AND PRADHAN, D. K. 1997. A cluster-based approach for routing in dynamic networks. *SIGCOMM Computer Communication Review* 27, 2 (April 1997), 49-64.

[43] KUIPERS, F., WANG, H., AND VAN MIEGHEM, P. 2005. The stability of paths in a dynamic network. In *Proceedings of the 2005 ACM Conference on Emerging Network Experiment and Technology* (Toulouse, France). CoNEXT'05. ACM Press, New York, NY, 105-114.

[44] LYON, G. 2000. The solar wind-magnetosphere-ionosphere system. *Science*, 288, 5473 (June 2000), 1987–1991.

[45] MALKIN, G. 1998. *RIP Version 2*. RFC. RFC Editor.

[46] MALKIN, G. 1994. *RIP Version 2 - Carrying Additional Information*. RFC. RFC Editor.

[47] MERUGU, S., AMMAR, M. H., AND ZEGURA, E. W. 2004. Routing in space and time in networks with predictable mobility. *Georgia Institute of Technology*. Technical Report Number: GIT-CC-04-07.

[48] MINDEN, G., EVANS, J., BALIGA, S., RALLAPALLI, S., AND SEARL, L. 2002. Routing in space based internets. In *the 2002 Earth Science Technology Conference* (Pasadena, California, United States).

[49] MOHORCIC, M., SVIGELJ, A., KANDUS, G., AND WERNER, M. 2002. Performance evaluation of adaptive routing algorithms in packet-switched intersatellite link networks. *International Journal of Satellite Communications* 20, 2 (February 2002), 97 – 120.

[50] MOY, J. 1994. *Multicast Extensions to OSPF*. RFC. RFC Editor.

[51] MOY, J. 1998. *OSPF Version 2*. RFC. RFC Editor.

[52] MURPHY, J., AND CHOW, E. 1996. Future predictions for DSN traffic. *JPL Pasadena*. Technical Report.

[53] NASA 2004. The vision for space exploration. *The NASA Technical Report Server*. Technical Report. Report Number: NP-2004-01-334-HQ.

[54] NORDSTRÖM, E., GOLD, R., AND Gunningberg, P. 2005. Mythbusters: whatever you thought about MANET routing, think again.... *Uppsala University*. Technical Report.

[55] PERKINS, C. 1996. *IP Mobility Support*. RFC. RFC Editor.

[56] POTORTÌ, F., AND FRANCK, L. 2002. GaliLEO: a simulation tool for leo satellite constellations. *Transactions of the Society for Computer Simulation International* 78, 9 (September 2002), 543-551.

[57] PRATT, S.R., RAINES, R.A., FOSSA, C.E., AND TEMPLE, M.A. 1999. An operational and performance overview of the Iridium low earth orbit (LEO) satellite system. *IEEE Communication Surveys & Tutorials* 2, 2 (Second Quarter 1999). 2-10.

[58] REKHTER, Y. AND LI, T. 1995. *A Border Gateway Protocol 4 (Bgp-4)*. RFC. RFC Editor.

[59] REYNOLDS, A. 2002. *Redemption Ark*. Orion Publication.

[60] SHAIKH, A., VARMA, A., KALAMPOUKAS, L., AND DUBE, R. 2000. Routing stability in congested networks: experimentation and analysis. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols For Computer Communication* (Stockholm, Sweden). SIGCOMM '00. ACM Press, New York, NY, 163-174.

[61] SHEN, C.C., AND RAJAGOPALAN, S. 2002. Challenges in interplanetary MANETs. *University of Delaware*. Technical Report.

[62] SHEN, C.C., RAJAGOPALAN, S., BORKAR, G., AND JAIKAEO, C. 2004. A flexible routing architecture for ad hoc space networks. *Computer Networks* 46, 3 (October 2004), 389-410.

[63] SHEN, N. 2004. *Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels*. RFC. RFC Editor.

[64] SHOSTAK, A.B. 2005. *Moving Along: Far Ahead (Tackling Tomorrow Today)*. Chelsea House Publications.

[65] SIDDIQI, A., AND NANDY,B. 2005. Improving network convergence time and network stability of an OSPF-routed IP network. In *Proceedings of the 2005 IFIP Networking Conference* (Waterloo, Ontario, Canada). 469-485.

[66] STEENSTRUP, M. 1995. *Routing in Communication Networks* (1st edition). Prentice-Hall.

[67] SUN, Z., CRUICKSHANK, H., IYENGAR, S., HOWARTH, M. P., CLAVEROTTE, L., AND GOMEZ, R. 2003. IP multicast over satellite. In *Proceedings of the 21st International Communications Satellite Systems Conference and Exhibit* (Yokohama, Japan). Paper Number: AIAA-2003-2378.

[68] TANENBAUM A. S. 1989. *Computer networks* (2nd edition). Prentice-Hall.

[69] TORRENCE, M. 2006. List of satellites. *NASA International Laser Ranging Service*. http://ilrs.gsfc.nasa.gov/satellite_missions/list_of_satellites/.

[70] VAHDAT, A., AND BECKER, D. 2000. Epidemic routing for partially-connected Ad Hoc networks. *Duke University*. Technical Report CS-200006.

[71] WELZL, M. 2000. Adaptive multimedia communication over satellite routed IP. In *Proceedings of the 2000 IEEE International Conference on Communication* 2 (New Orleans, Louisiana, United Sates). 827-830.

[72] WESSMAN, P. 1996. Existing routing protocols and IPv6. *WES*. http://www.tuxick.net/linux/ip6routing.html.

GLOSSARY

ABR	Area Border Router
AM-LSA	Area Membership-LSA
AM-Timer	Area Membership Timer
AS	Autonomous System
ASBR	AS boundary router
ASCoT	Autonomous Space Communications Technology
BGP	Border Gateway Protocol
COBR	Celestial Object Border Router
EGP	Exterior Gateway Protocols
EIGRP	Enhanced Interior Gateway Routing Protocol
GGP	Gateway-Gateway Protocol
IBRR	Interrogation-Based Relay Routing protocol
IGP	Interior Gateway Protocol
IP	Internet Protocol
IPv6	Internet Protocol Version Six
IS-IS	Intermediate System-to-Intermediate System
LSA	Link State Advertisement
LSD	Link State Database

LSR	Link State Routing
NASA	National Aeronautics and Space Administration
NBMA	Non-Broadcast-Multi-Access
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSPFv3	Open Shortest Path First version three
PLS	Positional Link-trajectory State
QoS	Quality of Service
RIP	Routing Information Protocol
SCBR	Satellite Constellation Border Router
SDIP	Shortest Delay Intermittent Pathway
SOSPF	Space Open Shortest Path First
SPF	Shortest Path First
SR-LSA	Space Router- Link State Advertisement
STRF	Space-Time Routing Framework
ToS	Type of Service

APPENDIX A

There are six orbital parameters which defines the location of an SOSPF router in space at any time. Those parameters are:

- Semi-major axis (a) : is half of the largest diameter distance between opposite points of an ellipse
- Eccentricity (e) : is the center-foci axis (c) divided by the semi-major axis (a), i.e.,

$$e = \frac{c}{a} \tag{48}$$

- Longitude of the ascending SOSPF router (Ω): is the angle from planet's line toward the vernal equinox point (Y) to the ascending SOSPF router (A) of a satellite's orbit along the plane of planets equatorial plane.
- Argument of perigee (ω): is the angle from planet's line toward the ascending SOSPF router (A) of a satellite's orbit eastward along the satellite's orbital plane to the perigee (P).
- Inclination of orbit (inc): is the angle of the satellite's orbit of the planet's equatorial plane at the ascending point (A).
- True angle (v) : is the angle from the line of the center of planet toward the perigee (P)

to the orbiting object.

We must define other parameters which assist in the definition of those six parameters

- Center-foci axis (c): is the distance from a foci to center of orbit
- Perigee (P): is the point of the orbit where the orbiting satellite is at its closest point to the planet.
- Ascending SOSPF router (A): is a point in the orbit of an object where it crosses the plane of reference from South to North in the direction of motion. (Usually the equator).
- Vernal Equinox point (Y): is the point where the center of the Sun, moving northwards, crosses the Earth's equator.

Figure 47 shows the parameters which are related to the elliptical orbit of satellite S_i. Figure 48 shows the parameters which are related to the orbital planes of planet S_k and satellite S_i Figure 49 shows the true angle of satellite S_i orbiting plane S_k.

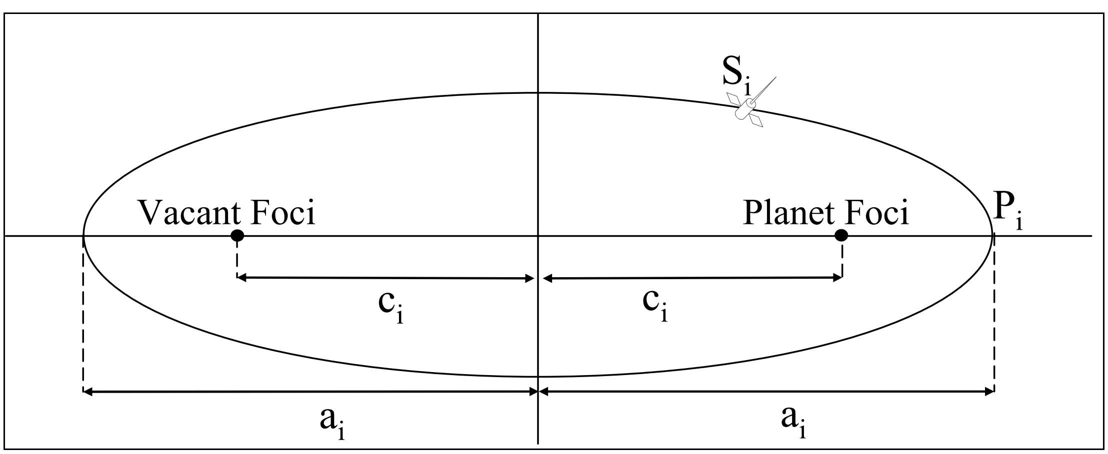

Figure 47: All the elliptical parameters

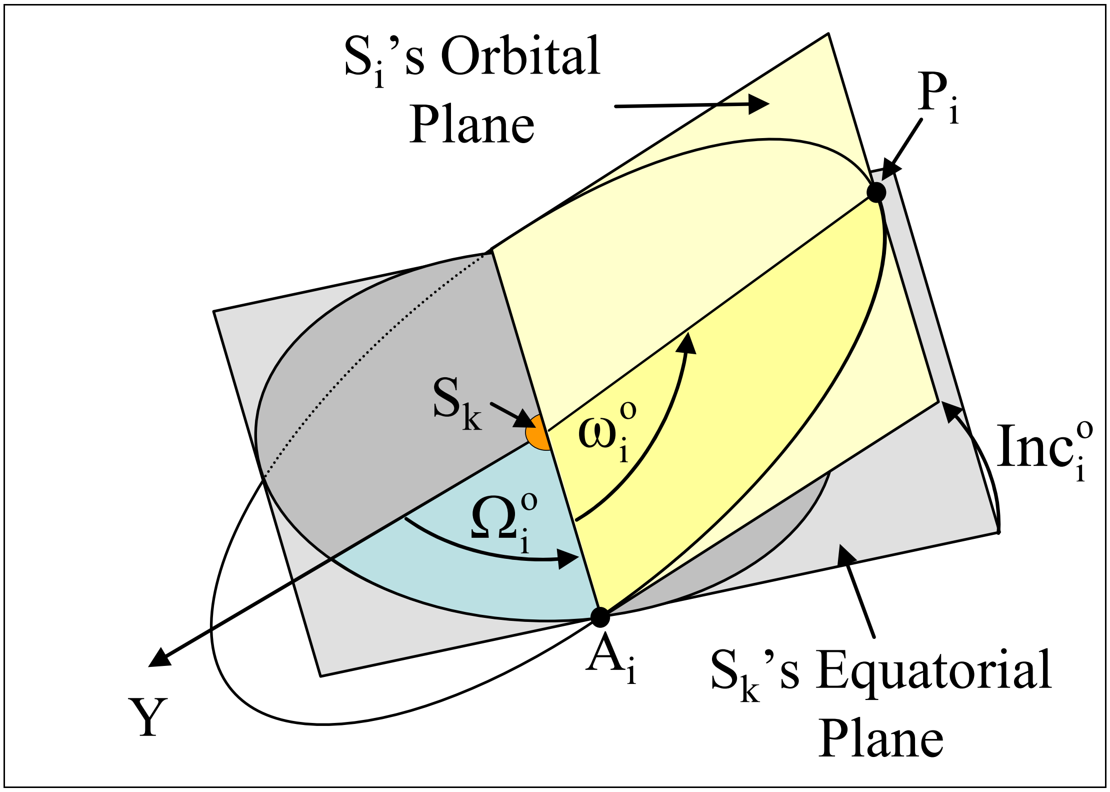

Figure 48: All orbital parameter of a planet

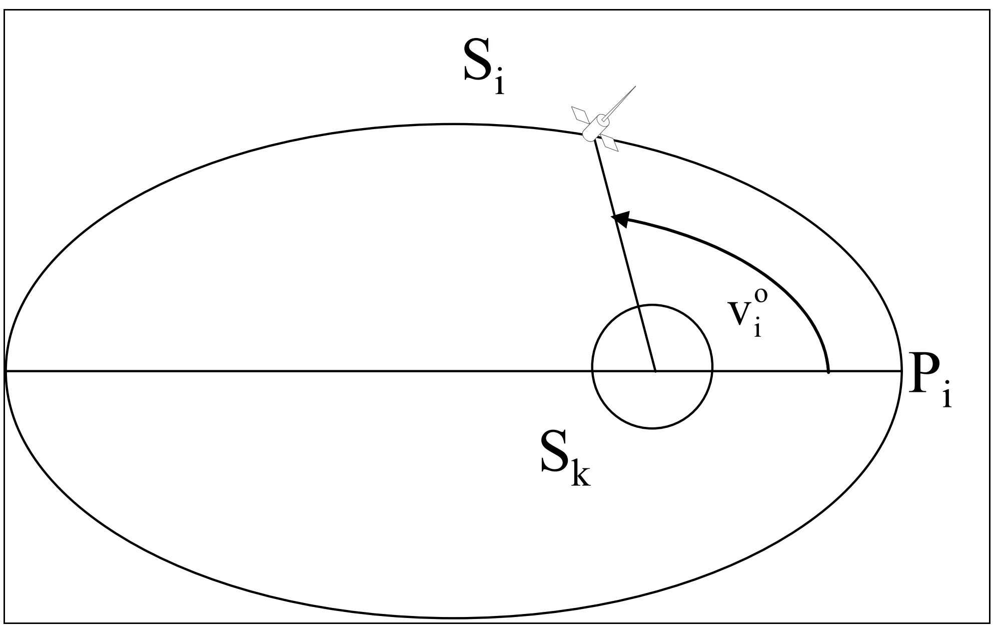

Figure 49: The view of the true angle

APPENDIX B

B.1 Area Field

Each area is defined with a 128 bit field, called Area Field which is divided into 5 sections. Each section represents a level of the area hierarchy such that the leftist field represents the identity of the routing domain's backbone level (level 0) and the right most field represents the lowest level of the area hierarchy (see Figure 50).

Moreover, leading zeros are omitted and a field with all zeros from the right is ignored when written. For example 0001:000003:000001:00000000:00000000 is written 1:3:1.

<table>
<tr><td colspan="2">Level 0 (16 bits)</td><td colspan="3">Level 1 (24 bits)</td><td colspan="3">Level 2 (24) bits)</td></tr>
<tr><td colspan="4">Level 3 (32 bits)</td><td colspan="4">Level 4 (32 bits)</td></tr>
</table>

Figure 50: The Area Field

B.2 Space Routers List

The *Space Routers List* consists of all the space objects (celestial objects and spacecrafts) that are used in the SOSPF routing protocol. Each object in space is described in this list to calculate its location and calculate their occlusion status. Each member of the *Space Routers List* has the following fields:

- *Type* identifies the type of the space object such as
    - 0 = Celestial object
    - 1 = Spacecraft
- *Calculating Method* describes how the object is calculated. All celestial objects and space crafts that are bound to orbiting a celestial object are to have the *Calculating Method* unless specified otherwise. The setting of the *Calculating Method* is as follow:
    - 0 = bound to the six orbital parameters
    - 1 = bound to trajectories defined outside the SOSPF routing protocol
- *ID* is an IPv4 address given to each space object. This IPv4 address is used for the purpose of identification only and has no correlation with IPv4 address assignment on Earth.
- *Semi-major axis* is half of the largest diameter distance between opposite points of an ellipse
- *Eccentricity* is the center-foci axis divided by the semi-major axis
- *Longitude of the ascending SOSPF router* is the angle from planet's line toward the vernal equinox point to the ascending SOSPF router of a satellite's orbit along the plane of planets equatorial plane.
- *Argument of perigee* is the angle from planet's line toward the ascending SOSPF router of a satellite's orbit eastward along the satellite's orbital plane to the perigee.
- *Inclination of orbit* is the angle of the satellite's orbit of the planet's equatorial plane at

the ascending point.

- *True angle* is the angle from the line of the center of planet toward the perigee

Number of entries		3
Entry # 1	Type	0
	Calculating Method	0
	ID	100.2.22.1
	Semi-major axis (AU)	0.20943951606750488280e1
	Eccentricity ^{o}	0.28189981821924448010e-3
	Longitude of the ascending SOSPF router ^{o}	0.41887903213500976560e1
	Argument of perigee ^{o}	0.24515445000000000000e7
	Inclination of orbit ^{o}	0.13993999455124139790e-2
	True angle ^{o}	0.59827148739760886020e-38
	True angle Data and Time	2006:08:30:23:14:50
Entry # 2	Type	1
	Calculating Method	0
	ID	99.1.122.12
	Semi-major axis (AU)	1.20943951606750488280e1
	Eccentricity ^{o}	0.28534223321924448010e-3
	Longitude of the ascending SOSPF router ^{o}	0.53422032135009765601e1
	Argument of perigee ^{o}	0.45324320000000000000e6
	Inclination of orbit ^{o}	0.12339994551241397900e-2
	True angle ^{o}	0.59827148739760886020e-24
	True angle Data and Time	2000:02:30:11:16:50
Entry # 3	Type	1
	Calculating Method	1
	ID	99.1.122.12
	Semi-major axis (AU)	0
	Eccentricity ^{o}	0
	Longitude of the ascending SOSPF router ^{o}	0
	Argument of perigee ^{o}	0
	Inclination of orbit ^{o}	0
	True angle ^{o}	0
	True angle Data and Time	0

Figure 51: An example Space Routers List

An example of a *Space Routers List* is shown in Figure 51.

B.3 Configured Parameters

MinLSArrival is the minimum time that must elapse between receptions of new LSA instances during flooding. The setting of *MinLSArrival* is dependant on the propagation delay between the pair. *MinLSArrival* is set during the database exchange process.

RouterDeadInterval is minimum time a router has to wait for a hello packet from a neighboring router prior considering the neighboring router down. The setting of *RouterDeadInterval* is dependant on the propagation delay between the pair. *RouterDeadInterval* is set during the database exchange process.

B.4 Constant Parameters

MaxTuple = 2^8-1

MaxMembershipPeriod = $2^{16} - 1$.

Maximum Stability Period = $2^{18} - 1$

InfDelay = $2^{32} - 1$

Although we have defined those constants, they can be modified if we see fit, in the future.