Directed Connectivity Analysis and Its Application on LEO Satellite Backbone

by

Junhao Hu
B.Sc., Northwest University, China, 2016

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Directed Connectivity Analysis and Its Application on LEO Satellite Backbone

by

Junhao Hu
B.Sc., Northwest University, China, 2016

Supervisory Committee

Dr. Lin Cai, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Kin Fun Li, Departmental Member
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. Lin Cai, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Kin Fun Li, Departmental Member
(Department of Electrical and Computer Engineering)

ABSTRACT

Network connectivity is a fundamental property affecting network performance.

Given the reliability of each link, network connectivity determines the probability that a message can be delivered from the source to the destination. In this thesis, we study the directed network connectivity where the message will be forwarded toward the destination hop by hop, so long as the neighbor(s) is (are) closer to the destination. Directed connectivity, closely related to directed percolation, is very complicated to calculate. The existing state-of-the-art can only calculate directed connectivity for a lattice network up-to-the size of 10×10. In this thesis, we devise a new approach that is simpler and more scalable and can handle general network topology and heterogeneous links. The proposed approach uses an unambiguous hop count to divide the networks into hops and gives two steps of pre-process to transform hop-count ambiguous networks into unambiguous ones, and derive the end-to-end connectivity. Then, using the Markov property to obtain the state transition probability hop by hop.

Second, with tens of thousands of Low Earth Orbit (LEO) satellites covering the Earth, LEO satellite networks can provide coverage and services that are otherwise not possible using terrestrial communication systems. The regular and dense LEO satellite constellation also provides new opportunities and challenges for network protocol design. In this thesis, we apply the directed connectivity analytical model on LEO satellite backbone networks to ensure ultra-reliable and low-latency (URLL) services using LEO networks, and propose a directed percolation routing (DPR) algorithm to lower the cost of transmission without sacrificing speed. Using Starlink

constellation (with 1,584 satellites) as an example, the proposed DPR can achieve a few to tens of milliseconds latency reduction for inter-continental transmissions compared to the Internet backbone, while maintaining high reliability without link-layer retransmissions.

Finally, considering the link redundancy overhead and delay/reliability tradeoff, DPR can control the size of percolation. In other words, we can choose a part of links to be active links considering the reliability and cost tradeoff.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

First and foremost, I would like to express special thanks to my supervisor, Dr. Lin Cai. She taught me that there is no high-quality answer without high-quality questions. I would like to thank her for sharing the interesting research problem and her research experience without reservation. Thanks to her mentoring and patience, and I can finally finish my research and keep improving my work. I am also very grateful for always encouraging me and helping me get through the difficulties. It will be always my great honor to work with her.

I am also particularly grateful to Prof. Jianping Pan for his great help and valuable comments on my research and final thesis. His guidance on how to do research, how to schedule things, and how to treat others always keep close to my heart.

Besides, I would like to thank all my friends in the CN Lab, Dr. Yue Li, Dr. Yongming Zhang, Dr. Wen Cui, Dr. Hamed Mosavat, Dr. Chenchen Zhao, Dr. Xiaolong Lan, Lei Zhao, and Bin Pan. I really enjoy spending quality time together with you. Sadly, the pandemic prevented us from getting together more.

I would also like to thank Prof. Kin and Prof. Chester for taking the time to review my thesis and provide constructive comments.

These days in Victoria are the most unforgettable time for me.

Junhao Hu

DEDICATION

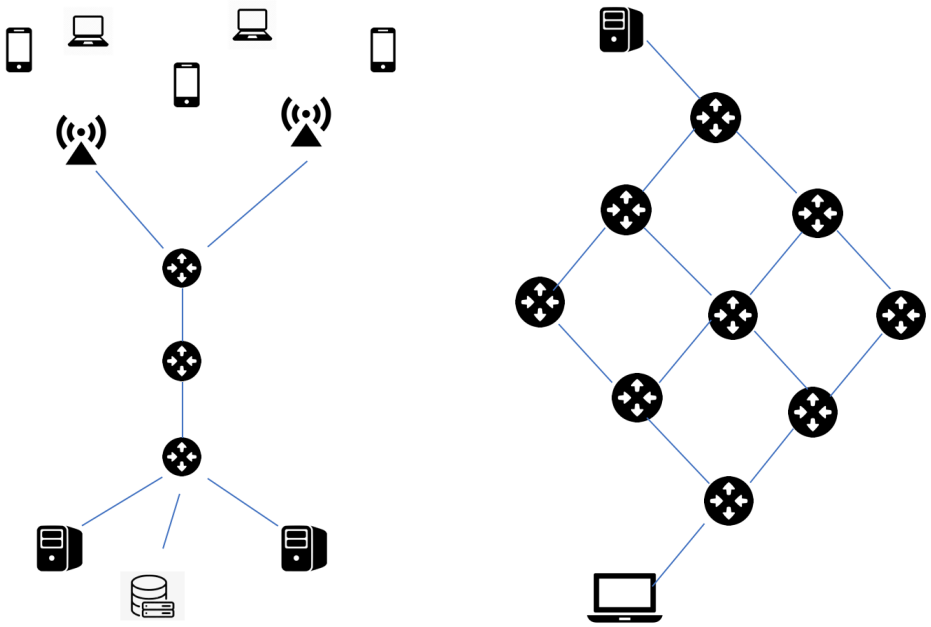
To my parents and supervisor!

Chapter 1

Introduction

Many emerging applications require Ultra-Reliable and Low-Latency (URLL) services, and they will be a driving force for the future growth of communication networks. First, for many Internet-of-Things applications, real-time sensing and control information needs to be exchanged among machines or algorithms which are less intelligent and not error-resilient, demanding high-reliability and in-time/on-time services. Here are the delay requirements for such real-time control applications: industry 4.0 (a few ns to a few ms); in cellular systems, Common Public Radio Interface (≤ 100 μs), and inter-site coordinated multipoint (Co-MP) (≤ 250 μs); smart grid (< 5 ms); vehicular communications for autonomous driving (a few ms). Second, applications such as high-frequency trading need to compete with each other at the ms level to profit from the high correlation of financial data distributed globally. For these applications, reducing a milli-second in delay can lead to million-dollar profits.

Compared to the previous delay-sensitive multimedia applications such as voice/video over IP, the above applications not only have a more stringent delay requirement but also cannot tolerate packet loss. The combination of high reliability and low latency brings tremendous challenges to packet-switching communication networks. It also attracts extensive research and standardization activities, covering both the backbone, access networks, and specialized/dedicated network systems. For the Internet community, IETF Deterministic Networking (DetNet) and IEEE 802.1 Time-Sensitive Networking (TSN) working groups closely cooperate with each other, aiming to support end-to-end service guarantee [15, 3]. In cellular systems, the 3GPP standard organization investigates how to provide URLL services with the 5G New Radio (NR) radio access networks, 5G core networks, and 5G fronthaul/backhaul [1]. For data center networks, how to apply priority flow control to ensure URLL services has



(a) Tree topology (b) 2×2 grid

Figure 1.1: Tree vs. mesh networks.

also been heavily investigated.

Given the traffic statistics and communication link properties, how to reserve/allocate resources in a single link to ensure latency/reliability has been extensively investigated. In a tree-topology network shown in Fig. 1.1(a), there is a single path for each flow, so we can ensure the URLL services of each link along the path to ensure the end-to-end performance. However, this approach is ineffective for mesh networks. In a 2×2 mesh network shown in Fig. 1.1(b), there are 6 "shortest" paths (in terms of hop count) between the source and destination pair. In a 4×4 and 9×9 one, the number of "shortest" paths is 70 (eight choose four) and $48,620$ (eighteen choose nine), respectively. Given the many possible paths, a key issue not fully explored yet is to deliver the message over multiple paths to ensure low latency and high reliability simultaneously.

How to quantify and then ensure network connectivity and reliability for large-scale, mesh topology networks is an open issue. If delivering a message over a mesh network using all possible paths, the end-to-end delay is the minimum of all paths. Given each link has only a certain probability to successfully deliver a packet within a bounded delay (e.g., without involving link-layer retransmission), the end-to-end reliability over the network is a key issue. Since there exists many dependent paths

in a two-dimensional network, the calculation of connectivity is prohibitively complex if using the traditional probability theory, i.e., $O(2^{\binom{2n}{n}})$ for an $n \times n$ lattice network. Note that $2^{\binom{2n}{n}} \approx 7.2e75$ for $n = 5$. To tackle the high complexity of the problem, a recursive approach was devised which can give the directed connectivity of two-dimensional lattice networks up-to the size of ten-by-ten [20]. This work published in 2014 is also the best known result. How to obtain closed-form solution for a larger network remains open.

In this thesis, we propose a new, more efficient approach that can calculate a two-dimensional lattice of the size of 15×15, and it can handle both regular and irregular topology networks. First, we pre-process a network to a directed graph with unambiguous hop count (from the source) for each vertex, so all vertexes with the same hop count can be grouped into a set. Then, we construct a Markov process to decouple the end-to-end reliability problem by calculating the connectivity between any neighbor sets, which results in much lower complexity compared to the existing state-of-the-art solution. The proposed method is named Hop-State Algorithm (HSA).

Furthermore, to reduce link cost without violating the service guarantee, two metrics are proposed to assist a trimming process to select a part of the network to deliver the message under the reliability constraint. According to analysis different networks with same number of links, two intuitionistic metrics are proposed to quantitative analysis of the end-to-end reliability. Therefore, less active links can be selected to achieve a certain threshold reliability by backward deduction.

As an application scenariothanks to the reusable rocket technologies and the development of small-size, light-weight, and low-cost satellites, launching Low Earth Orbit (LEO) satellites becomes more economical. With tens of thousands of LEO satellites covering Earth, LEO satellite networks can provide coverage and services that are otherwise not possible using terrestrial communication systems. How to provide URLL communication services over long distances has been one of the most challenging and profitable networking problems. For instance, in high-frequency trading (flash trading), the opportunity to realize a profit may be available for only a few to a fraction of a millisecond before parity is achieved. Since the propagation speed of light in the space is about 50% higher than that in fiber optical links, using LEO as the backbone for long-distance communication is anticipated to save tens of milliseconds for inter-continent transmissions. URLL services thus are critical to the profitable operation of LEO networks.

However, the current Internet architecture and TCP/IP protocol stack were designed to provide best-effort connectivities for arbitrary network topologies, and they do not take the advantages of the regular and dense LEO satellite constellation, and cannot well-support URLL communication services. For instance, routing strategies on the Internet are based on various ways to implement shortest-path routing, so the optimal path can be used to minimize the metrics such as the number of hops, delay, or other link costs [12, 18]. The reliability of transmissions highly depends on link-layer retransmissions using the automatic repeat request (ARQ) protocols. In LEO, the distance between satellites are hundreds to thousands of km away, so propagation delay over a single link can take up-to tens of ms, harmful to URLL services. Furthermore, satellites at different locations may experience severely uneven loads.

In the literature, several new routing strategies have been proposed for LEO satellite networks, aiming to balance the load or taking multi-path to enhance the throughput [17, 12, 18]. How to using redundant satellite links for URLL communication services remains an open issue, which motivates this work.

Given the large bandwidth of inter-satellite optical communication links, we aim to achieve URLL communication by directed percolation (DP) of packets in the network. Since the LEO network exhibits a grid structure, packets can take many possible links to reach the destination, so long as the next hop is closer to the destination. For example, Toronto is in the northeast of San Francisco, so the packet from San Francisco can be delivered towards north/east to reach Toronto. DP is a fundamental problem heavily investigated by Physics, Chemistry, Material Science, etc. To the best of our knowledge, this is the first application of DP for backbone networks supporting URLL services.

The main contributions of this thesis are three-fold. First, we propose a new routing strategy, named Directed Percolation Routing (DPR), where each satellite routes a packet over several output links towards the destination. The reliability and latency of the service can be ensured given the redundant transmissions, instead of relying on link-layer retransmissions. Second, we develop a performance analysis framework to quantify the reliability and latency of DPR, when different numbers of links are used for packet transmission. Third, using the Starlink constellation as an example, we conduct extensive simulations to obtain the delay distribution and reliability performance of DPR, compared with the optimal shortest-path routing and multi-path routing. Simulation results show that with the proposed DPR, the inter-continent propagation delay can be reduced by at least 4 to 21 ms, while the

reliability can be several orders higher than single-path optimal routing. The analysis and simulation results reveal the tradeoff between the transmission cost and the service quality in terms of delay and reliability, which can be used to set DPR system parameters according to the service requirements.

The rest of the thesis is organized as follows. Sec. 2 gives the related work. The new HSA approach to derive the directed connectivity of the network is given in Sec. 3. Sec. 4 presents the case study, applying directed connectivity analysis and link selection strategy in LEO satellite backbone networks, and proposes a directed percolation routing for providing inter-continental URLL services. Sec. 5 applies the HSA approach in two-dimensional lattice networks to compare its performance and validate its correctness with the existing work handling 2-D lattice networks. In Sec. 6, we focus on the performance evaluation with several experiments on the performance comparison, and how to select links from the network to achieve high connectivity/reliability is also given, followed by the concluding remarks and further research issues in Sec. 7.

Chapter 2

Related Work

The connectivity of in two-dimensional multi-hop networks is a fundamental problem [4]. Due to the random location of the nodes in such networks, the multi-hop forwarding has to be characterized probabilistically. The approximate formula is presented for the probability of network connectivity [8]. With the assumption of a uniform distribution of nodes, the exact analytical expressions of the probability of connectivity are obtained in one-dimensional networks but only approximate bounds for the connectivity in two-dimensional networks [5]. The connectivity of message propagation in the two-dimensional ladder case is also derived [19].

On the other hand, geometric algorithms are also used in wireless sensor networks [7]. Different techniques based on stochastic geometry and the theory of random geometric graphs (including point process theory, percolation theory, and probabilistic combinatorics) have led to different results on connectivity, capacity, outage probability, and other fundamental limits of wireless networks [10]. Due to the physical space of nodes, networks show a unique geometric characteristic such as triangles, rectangles, and hexagons [21, 22, 23]. Although hexagons and rhombuses are also used, the square lattices network is most widely used in city scenarios. In addition, extra nodes can be distributed to improve connectivity by exploring geometric structures for sensor network operation and design.

Another approach to compute the probability of network connectivity is percolation theory [9]. Given a destination, messages flooding to the certain directions in geographical forwarding which is similar to a directed percolation process. For messages with a given destination, or vehicles traveling in certain directions, geographical forwarding is often deployed to minimize the network overhead due to flooding [19]. Thus directed percolation becomes a utilitarian model in such scenarios, and most ex-

isting work applies the results from isotropic or directed percolation on square lattices. However, the DP problem only cares about the existence of a giant component, while DC has to determine the exact connectivity to each vertex, which is more relevant to network performance [20].

The most related existing work is presented in [20]. Considering a 2D lattice topology, a recursive decomposition approach is developed by extending the 2D ladder connectivity to establish the analytical expression. Instead of splitting up a network into parts, this approach decomposes lattice in one path and the union of all other paths. Although it can quickly determine the network connectivity without lengthy simulation, the maximum result of this approach is 10*10 lattice due to its huge path number.

This thesis also studies the DC problem on square lattices. Given the complexity of the DC problem on square lattices, this thesis studies another solution based on hop number in the network. We try to establish a research model to address the most significant challenge of DC, to determine the exact connectivity to each vertex. If we want to separate networks into parts, the probability of vertexes is not mutually exclusive. Different from the existing work, we analyze the whole network and separate its links into groups instead of derivative a small part from the whole lattice. We also explore the exact connectivity expression as a function of the link reliability probability. In addition, our new approach extends to more general DC problems with variable link reliability probabilities, irregular shape, and three-dimensional networks.

In many routing algorithms, virtual topology is widely used to control the dynamical satellite networks. Duan[6] proposed a distributed processing method based on the position information to selects the 'nearest' satellite. To maintain satellite communication continuity, it reduces the propagation delay while keeping the handover times acceptable and maintain a lower complexity. However, the 'nearest' satellite shows sustainable and rarely handover in Starlink. [12] divided the whole space into small cubes (i.e., net grids) and then, satellites can be located by net grids instead of coordinates. Then, the routing complexity is significantly reduced.

Besides, this thesis also focus how to avoid busy links on LEO satellite backbone networks. Many routing algorithms focused on how to minimize end-to-end propagation delay. Due to the unbalanced traffic load in dynamical inter-satellite links (ISLs), some links may become heavily loaded even congested while others remain underutilized. To avoid the congestion, an explicit load balance (ELB) scheme is used to requests congested satellites to decrease the sending rate[17]. Unfortunately,

a part of link congestion still can not be effectively prevented. [13] proposed a State-Aware and Load-Balanced (SALB) routing model to divides the occupancy rate of each queue into several levels and each level corresponds to a link-state.

Multipath routing has been used in satellite networks [18]. This thesis proposed a Network Coding based Multipath Cooperative Routing (NCMCR) protocol for LEO satellite networks to improve the throughput. This source-based and destination-based multipath cooperative routing algorithms deliver different parts of data flow along multiple link-disjoint to decrease the end-to-end (e2e) delay. However, this algorithm flooding messages through detour links which bring redundant transmissions.

Chapter 3

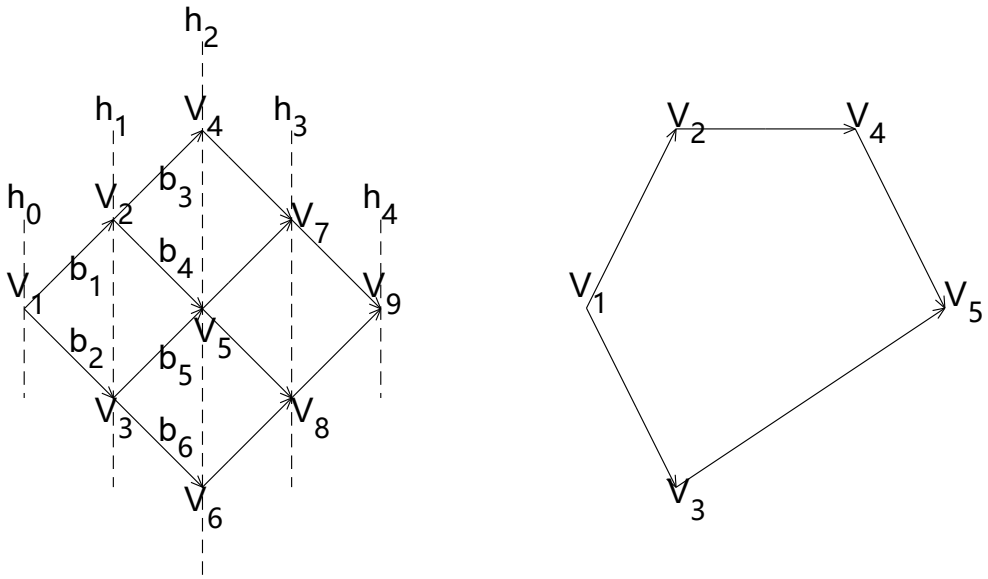
Directed Connectivity Analysis

3.1 System Model

In a network with the DPR strategy, a message from the source node can take all possible directed paths to reach the destination. This message propagation process mimic filtering of fluids through porous materials along a given direction, due to the effect of gravity. To avoid the blindly flooding cost, we assume that each message will be delivered over each link once. In other words, if a router receives a duplicated message, it will discard the message. Then, the link cost to deliver the message depends on the number of active links in the network. As shown in Fig. 3.1(a), at the source and all intermediate nodes, the message will be duplicated and forwarded to the neighboring nodes if the neighbor is closer to the destination[1]. This network can be viewed as a directed graph where vertexes are network nodes (e.g., routers) and edges are the links between the vertexes. There will not be any loop given the directed percolation routing in this directed graph. At any moment, each link or edge (used interchangeably in this thesis) can be connected with a probability. This connectivity probability can also be defined as the link reliability, or the probability that a packet can be successfully delivered over the link. (The physical meaning of the connectivity probability can be defined in different applications.)

To calculate the end-to-end connectivity in a directed graph, we first define hop count of each vertex or node (used interchangeably in this thesis), and then pre-process the network graph based on hop count.

[1]To make it simple, we use geography distance to decide which node is closer to the destination, and assume each link between two nodes are in a straight line, which is the case for wireless communication systems. We can extend the definition of distance to link distance for wired networks.



(a) unambiguous network (b) unambiguous network

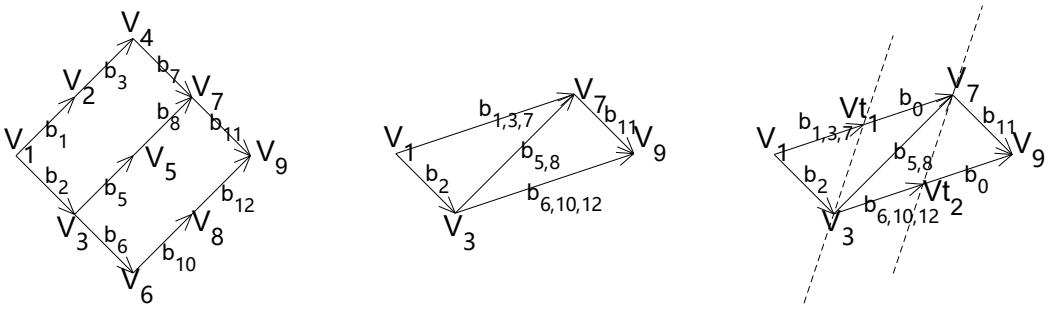
Figure 3.1: Unambiguous and ambiguous samples

3.1.1 Unambiguous Hop Count

Node with unambiguous hop count: If every directed path from the source to reach a vertex has the same number of hops, the vertex has an *unambiguous* hop count. For example, in Fig. 3.1(a), V_1 is the source, and there are two paths to reach vertex V_5, i.e., $V_1 \rightarrow V_2 \rightarrow V_5$ and $V_1 \rightarrow V_3 \rightarrow V_5$. The hop count of both paths is two, so V_5 has an unambiguous hop count of two.

Network with unambiguous hop count: If all vertexes in a network have an unambiguous hop count, we define the network as an unambiguous hop count network; otherwise, it is an ambiguous one. For example, in Fig. 3.1(a), the source vertex V_1 has the hop count of 0, the hop count of vertex V_2 and V_3 are both 1, that of vertex V_4, V_5, and V_6 are 3, and so on. Thus, this network has unambiguous hop count. In Fig. 3.1(b), the hop count to reach vertex V_5 can be either 2 or 3 when taking different directed paths, so it is an ambiguous hop-count network.

In an unambiguous hop-count network, the vertexes can be grouped into sets G_h where h is the hop count to reach the vertexes from the source. For instance, in Fig. 3.1(a), the dotted line h_i passes all vertexes in group G_i.



(a) (b) (c)

Figure 3.2: Pre-process network graph

3.1.2 Pre-process Network Graph

We take two steps below to pre-process the network graph. First, to simplify the analysis, we can convert a network to an equivalent one with a simpler topology by combining chained links passing through the nodes with a single link in and a single link out. For example, as shown in Fig. 3.2 (a), V_2 and V_4 each has a single link in and a single link out, so the links in and out from them, b_1, b_3, and b_7 can be combined into one, named $b_{1,3,7}$ as shown in Fig. 3.2 (b). The connectivity probability of $b_{1,3,7}$ equals the multiplication of the connectivity probabilities of b_1, b_3, and b_7.

Second, an ambiguous hop-count network can be converted to an unambiguous one as follows. For a vertex, if it can be reached by several paths with different hop counts, virtual links (each has the connectivity of 1) should be inserted into the shorter paths until all paths have the same hop count to reach the vertex. For instance, the network in Fig. 3.2(b) is an ambiguous hop-count network. Two paths to reach vertex V_7 are $V_1 \rightarrow V_7$ and $V_1 \rightarrow V_3 \rightarrow V_7$. Thus, we add a virtual link b_0 into the first path by adding a virtual vertex V_{t_1}, as shown in Fig. 3.2(c). Similarly, we add a virtual vertex V_{t_2} in the bottom link between V_3 and V_9 to make V_9 unambiguous. Then, the network becomes an unambiguous one as shown in Fig. 3.2(c).

With the two-step pre-processing, we can obtain the simplified unambiguous network with the same connectivity as the original one. In the following, we can focus on the connectivity analysis of unambiguous networks.

3.2 Connectivity Analysis

3.2.1 Label States for Each Hop Set

For an unambiguous network, we can partition all the vertexes into sets, G_h, based on their hop count h from the source. Considering whether or not each vertex received the message in a vertex set, there are 2^n states, where n is the number of vertexes in the vertex set. For instance, the vertex set G_1 in Fig. 3.1(a) have 2 vertexes V_2 and V_3. Thus, one of the following three states should be reached to ensure a connected network: V_3 receives the message but V_2 not, V_2 receives the message but V_3 not, and both of them receive the message.

Obviously, if a message can reach its destination, the message must reach at least one vertex in each vertex set G_i, where i is smaller than or equal to the hop count of the destination. The state of G_i depends on the states of G_{i-1} and the connectivity of links between the two sets only, and we do not need to consider how the state of G_{i-1} is reached. This is in fact an important Markov property that can help us to simplify the connectivity analysis substantially.

To denote the state of each set, we first use a binary number to label whether or not a vertex receives the message by 1 and 0, respectively. Then the sequence of the binary labels of all vertexes in a set can be converted to its state label. For instance, the three states of G_1 in the above example can be labelled by state $01_2 = 1$, $10_2 = 2$, and $11_2 = 3$, respectively.

Denote all states for the message reaching at least one vertex in set G_h as S_k^h, where k is the state label. We have $1 \leq k \leq 2^{r_h} - 1$ where r_h is the number of vertexes in G_h. Using this notation, S_5^2 implies that in G_2, the first and third vertexes receive the message, and the rest not.

3.2.2 State Transition Probability

We define $P(S_k^h)$ as the probability of reaching state S_k^h and $P(b_n)$ as the probability of link b_n is connected.

Considering the links between two neighbour vertex sets, the state transition probability from S_k^h to $S_{k'}^{h+1}$ is denoted as $H_{k,k'}^h$.

For instance, as shown in Fig. 3.1(a), for state S_k^2, given four links b_3, b_4, b_5, and b_6, and the states in G_1, we can obtain the states S_k^2 $(1 \leq k \leq 2^{r_h} - 1)$ in G_2 as follows.

For S_1^1, V_2 is connected to the source, V_1, so we need to consider its outgoing links b_3 and b_4. If b_3 is connected but b_4 is not, V_4 is connected, and thus state S_1^2 can be reached from S_1^1. We have $H_{1,1}^1 = P(b_3)(1 - P(b_4))$.

For S_2^1, only V_3 is connected and it cannot deliver messages to V_4.

For S_3^1, V_2 and V_3 both received the message, so b_3, b_4, b_5, and b_6 all should be considered. When b_3 is connected but others are not, only V_4 received message. This state can be expressed as S_1^2. We have $H_{3,1}^1 = P(b_3)(1-P(b_4))(1-P(b_5))(1-P(b_6))$.

Then, $P(S_1^2)$ is given by

$$P(S_1^2) = P(S_1^1) \cdot H_{1,1}^1 + P(S_3^1) \cdot H_{3,1}^1. \tag{3.1}$$

Similarly, all connectivity probabilities of states in G_3 can be given, as shown in Fig. 3.3(a).

Using the Markov property, given the states in the G_h, and the connectivity of links between G_h and G_{h+1}, the probability to reach states in G_{h+1} is given by

$$P(S_{k'}^{h+1}) = \sum_{k=1}^{2^{r_h}-1} P(S_k^h) \cdot H_{k,k'}^h. \tag{3.2}$$

Next, we elaborate how to obtain the state transition probability $H_{k,k'}^h$. Using the binary sequence of state k, we define vertex set A includes those who have received the message and can send it to their next hop as

$$A = \{Va_1, Va_2, Va_3, ..., Va_\alpha\}, \tag{3.3}$$

where α is the number of vertexes in A.

For instance, in Fig. 3.3(b), to reach $H_{7,10}^h$, $7 = 111_2$ means that V_1^h, V_2^h, V_3^h can send messages to vertexes in G_{h+1}. Given the links from G_h to G_{h+1}, we have $E = \{Ve_1, Ve_2, Ve_3, ..., Ve_\epsilon\}$ who may receive the message, where ϵ is the number of vertexes in E.

In the above example, given $\alpha = 3$, $A = \{V_1^h, V_2^h, V_3^h\}$, and thus $E = \{V_1^{h+1}, V_2^{h+1}, V_3^{h+1}, V_4^{h+1}\}$ with $\epsilon = 4$. Using the binary number of k', we have vertexes $C = \{Vc_1, Vc_2, Vc_3, ..., Vc_\delta\}$ ($\delta > 0$) who may receive the message corresponding to this state. Obviously, C should be a subset of E.

We define set $I = \{Vl_1, Vl_2, Vl_3, ..., Vl_\iota\}$ ($\iota \geq 0$), which contains the vertexes in E but not in C, i.e., $E = C \cup I$ and $\emptyset = C \cap I$. In the example shown in Fig. 3.3(b),

$10 = 1010_2$, so $C = \{V_2^{h+1}, V_4^{h+1}\}$ ($\delta = 2$). Correspondingly, $I = \{V_1^{h+1}, V_3^{h+1}\}$ ($\iota = 2$).

From the definitions, vertexes in C receive the message, and vertexes in I do not receive the message. Define $P(C_i)$ as the probability of at least one message is received by vertex Vc_i, and $P(I_j)$ as the probability of vertex Vl_j receiving no message. We have

$$H_{k,k'}^h = \Pi_{i=1}^{\delta} P(C_i) \cdot \Pi_{j=1}^{\iota} P(I_j). \tag{3.4}$$

For set A defined in (3.3), define $\{Va_{c1}, Va_{c2}, ..., Va_{cp_i}\}$ as the vertexes who are linked with Vc_i, and p_i is the number of these links, as shown in Fig. 3.3 (c). For the example shown in Fig. 3.3(b), $\{V_1^h, V_2^h\}$ ($p_2 = 2$) are linked with V_2^{h+1} and the two links are b_1 and b_2. [2] Following the same principles, for V_4^{h+1}, it is linked with $\{V_3^h\}$ ($p_4 = 1$) through link b_6. These links are denoted as $bc_{i,\kappa}$ ($1 \leq \kappa \leq p_i$), so the connectivity probability of these links are $P(bc_{i,\kappa})$, respectively. For V_{C_i} to be reached, at least one link of $bc_{i,\kappa}$ should be connected, so we have

$$P(C_i) = 1 - \Pi_{\kappa=1}^{p_i}(1 - P(bc_{i,\kappa})). \tag{3.5}$$

Following the approach, using the example in Fig. 3.3(b), $C = \{V_2^{h+1}, V_4^{h+1}\}$ has two vertexes. The first vertex V_2^{h+1} can be reached through links b_2 and b_3, so $P(C_1) = 1-(1-P(b_2))(1-P(b_3))$. The other vertex V_4^{h+1} can only receive the message from link b_6 because vertex V_4^h is not connected, so $P(C_2) = 1-(1-P(b_6)) = P(b_6)$.

As shown in Fig. 3.3(d), similarly, vertex Vl_j can receive messages from $\{Va_{l1}, Va_{l2}, ..., Va_{lq_j}\}$, where q_j is the number of these links. The probability of link $bl_{j,\kappa}(1 \leq \kappa \leq q_j)$ is connected is denoted by $P(bl_{j,\kappa})$. For vertexes in I which are not connected, we have

$$P(I_j) = \Pi_{\kappa=1}^{q_j}(1 - P(bl_{j,\kappa})). \tag{3.6}$$

In the example in Fig. 3.3(a), group I has two vertexes, V_1^{h+1} and V_3^{h+1}. If V_1^{h+1} is not connected, link b_1 is not connected, so $P(I_1) = 1-P(b_1)$. If V_3^{h+1} is not connected, links b_4 and b_5 are both disconnected, we have $P(I_2) = (1 - P(b_4))(1 - P(b_5))$.

[2] Note that p_i is equal to or less than 2 in lattice networks, but in other networks, e.g., in triangle networks, p_i can be larger.

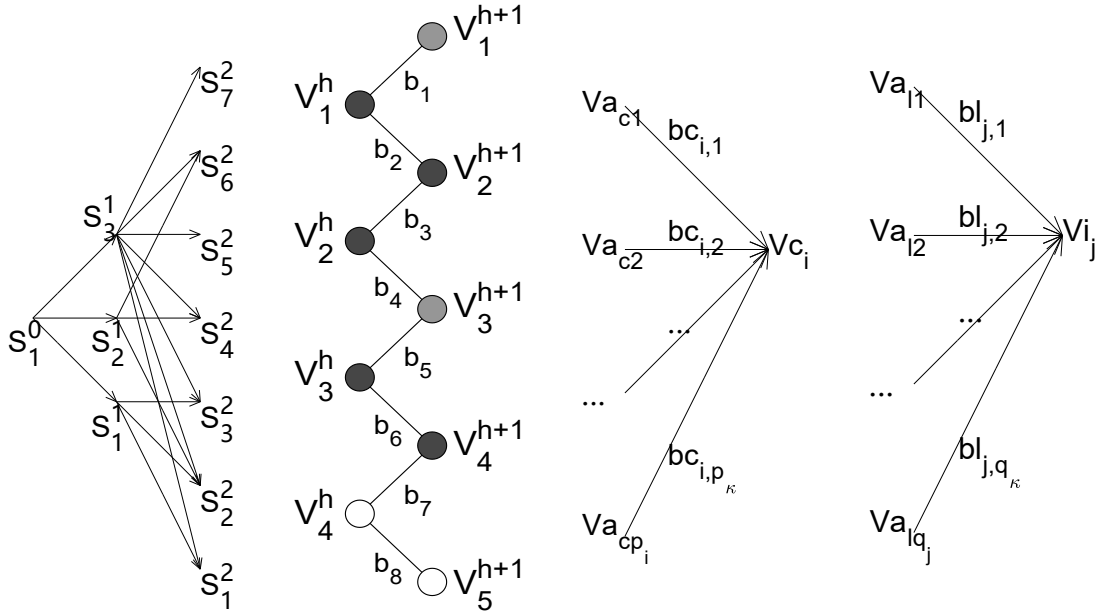


Figure 3.3: State transitions

Then the $H^h_{k,k'}$ can be expressed as

$$\begin{aligned} H^h_{k,k'} = \Pi^{\delta}_{i=1}[1 - \Pi^{p^i}_{j=1}(1 - P(bc_{i,j}))] \\ \times \Pi^{\iota}_{i'=1}\Pi^{q^{i'}}_{j'=1}(1 - P(bl_{i',j'})). \end{aligned} \tag{3.7}$$

The result state transition probability of the example in Fig. 3.3(a) is

$$\begin{aligned} H^h_{7,10} =& P(C_1)P(C_2)P(I_1)P(I_2) \\ =& [1 - (1 - P(b_2))(1 - P(b_3))]P(b_6) \\ & (1 - P(b_1))(1 - P(b_4))(1 - P(b_5)) \end{aligned}$$

3.2.3 Hop-State Algorithm (HSA) for Connectivity Calculation

Our connectivity calculation is summarized in the Hop-State Algorithm (HSA) shown in Algorithm 1, with the termination condition when the final hop reaches the destination. When we calculate for a network with one destination, the final hop set has one vertex only, and the corresponding connected state probability is also the

probability that the network is connected.

Algorithm 1 Hop-State Algorithm (HSA) for Connectivity

$h = 0$ # hop count number
for not reach the destination **do**
    $k' = 0$ # state number
    for $k' < 2^{r_{h+1}}$ **do**
        $k' = k' + 1$
        $P(S_{k'}^{h+1}) = 0$
        $k = 0$ # state number
        for $k < 2^{r_h}$ **do**
            $k = k + 1$
            if The corresponding C belongs to E **then**
                # add the probability contributing to state k' in hop $h+1$ from state k in hop h
                $P(S_{k'}^{h+1}) = P(S_{k'}^{h+1}) + P(S_k^h) \cdot H_{k,k'}^h$
            end if
        end for
    end for
    $h = h + 1$
end for

Chapter 4

Low Earth Orbit (LEO) Satellite Networks

4.1 Case Study: Starlink LEO Lattice Network

Low Earth Orbit (LEO) satellite networks have attracted a lot of attention in recent years and it is a motivating example for our HSA approach. The Starlink project by SpaceX proposes an LEO network with thousands of satellites. In the first phase, 1,584 satellites will be launched in 72 orbital planes with 22 satellites each. Neighbour satellites in the same orbital plane and two nearest satellites in neighbour orbital planes can communicate with Inter-Satellite-Links (ISLs). The structure is similar to the lattice networks where each satellite is a vertex in the lattice and each ISL is the edge. The method of creating grid topology of satellites was introduced by our previous work [11].

Given the lattice network topology of LEO, we can apply directed percolation routing to ensure high-reliability and low-latency inter-continental network services. In the following, we first introduce the directed percolation routing used in LEO networks and devise the strategy to configure the routing scope to ensure high reliability with low cost.

4.1.1 Directed Percolation Routing (DPR)

Note that the LEO backbone is drastically different from the traditional Internet backbone. First, all satellites are flying at high speed but at a predefined orbit and speed. Satellites in the same orbital plane have a low relative speed, while those

in different orbital planes have a high relative speed. Links between satellites in the same orbital plane are named Satellite-Link (SL) and those between satellites in neighbour orbital planes are Orbit-Link (OL). Each satellite maintains relatively stable SL links and highly dynamic OL links. Second, the locations of all satellites at any time instant are known and they form a grid topology, as shown in Fig. 4.1. Also, the distance and link capacity of all ISLs at any time instant can be estimated. Third, each ISL can be up to a thousand km long, which results in a high propagation delay. Fourth, depending on the current locations, the traffic volumes to different satellites are highly uneven. These prominent features and the needs to support URLL services request a re-visit to the network architecture and routing protocol design.

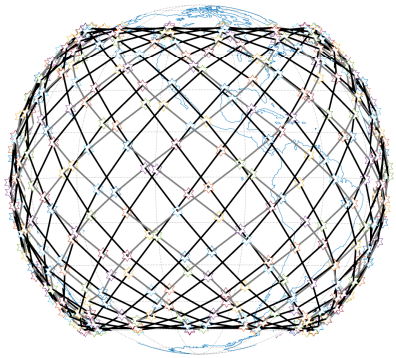


Figure 4.1: LEO network topology

Traditional Internet routing aims to find the shortest path to minimize the end-to-end path cost, using centralized or distributed algorithms. In LEO, ISL links are long-distance and highly dynamic, and using link-layer retransmissions to ensure per-hop and then per-path reliability will lead to large delay variation, not desirable for URLL services. On the other hand, given the grid topology of satellites, there are many paths between a pair of satellites with similar costs or latency. It is desirable to take the advantage of the grid topology and the many available links in the network for URLL services.

Given the same angular velocity of the satellites with the same orbit altitude, the satellites in the same orbital plane are almost static relatively, and each uses two SL links to connect to their neighbors in the same orbital plane. Each satellite can connect to its nearest satellites in the two neighbour orbital planes uses two OL links. as shown in Fig. 4.1, the black lines indicate SL links and gray lines indicate OL links.

A simple method is to use directed flooding to send the packet over the whole grid between the source/destination satellites. If the grid starting from the source satellite

and ending at the destination satellite is of size $m \times n$, directed flooding means that each satellite (acting as a router in the LEO backbone network) will relay any packet to the neighboring satellite that is closer to the destination of the packet. Using the Starlink constellation as an example, each satellite maintains four ISL links, two with the neighboring satellites in the same orbital plane and two with those in the neighbor orbital planes. Given the high connectivity of the LEO backbone, such a flooding strategy is prohibitively costly, as the destination may receive up to $\binom{m+n}{n}$ copies of the same packet, as there are $\binom{m+n}{n}$ paths in the $m \times n$ grid.

To avoid such flooding costs, we can ensure that each packet is transmitted only once per link, so packets will be percolated in the grid towards the destination. Thus, we name it Directed Percolation Routing (DPR). Using DPR, each router in the grid will forward a packet to each of its neighbor satellites which are closer to the destination once. Since the same packet may reach a satellite through more than one links, each satellite needs to store the identity of packets being transmitted for a period of time, i.e., a timer is set for a packet identity stored, and it will be removed from the buffer when the timer expires. Meanwhile, when a satellite receives a packet from its neighbor, it will be compared with the stored packet identities. If the new arrival finds a match, the new arrival will be dropped and no further action is needed; otherwise, the new arrival's identity will be stored, and the packet will be sent to the neighbor satellite(s) who is (are) closer to the destination. According to our analysis, a timer of 50 ms is sufficient, so the storage and comparison cost introduced by DPR is limited.

With DPR, considering the $m \times n$ grid where each satellite has four ISLs, each satellite will receive at most two copies of the same packet, and it will send at most two copies out. DPR's total transmission cost of a packet in an $m \times n$ grid equals the number of links, $2mn + m + n$, which is much more affordable compared to flooding. The load is also evenly distributed to all links in the grid. Even if the movement of the satellites may lead to topology change from time to time, the location-based DPR routing can make the decision based on the current locations of neighboring satellites. Thus, the DPR routing based on locations is simple and robust.

A major benefit of DPR is that, given the redundant but evenly distributed transmissions, the end-to-end reliability can be substantially improved, without relying on link-layer retransmissions. It is particularly desirable in LEO networks for URLL services as the link-layer retransmission is very time-consuming. When random nodes or links failure occurs, the redundant transmissions from the other part of the network

can maintain the URLL services.

Furthermore, the DPR can be flexibly configured to reduce the transmission cost by removing some overly-congested or undesirable ISLs from the percolation paths to make a tradeoff between transmission cost and service quality. Such a tradeoff relies on a thorough analysis of DPR performance given the network topology and each ISL's characteristics. In the following section, we will develop the analytical model to study the DPR performance.

To quantify the performance of DPR, we first analyze the network topology using Starlink parameters. The topology can determine the SL and OL link distance and propagation delay. Then, based on the path loss model, we can obtain the link reliability. Applying the directed connectivity analytical model in the previous section, the end-to-end DPR reliability can be obtained.

4.2 LEO Analysis

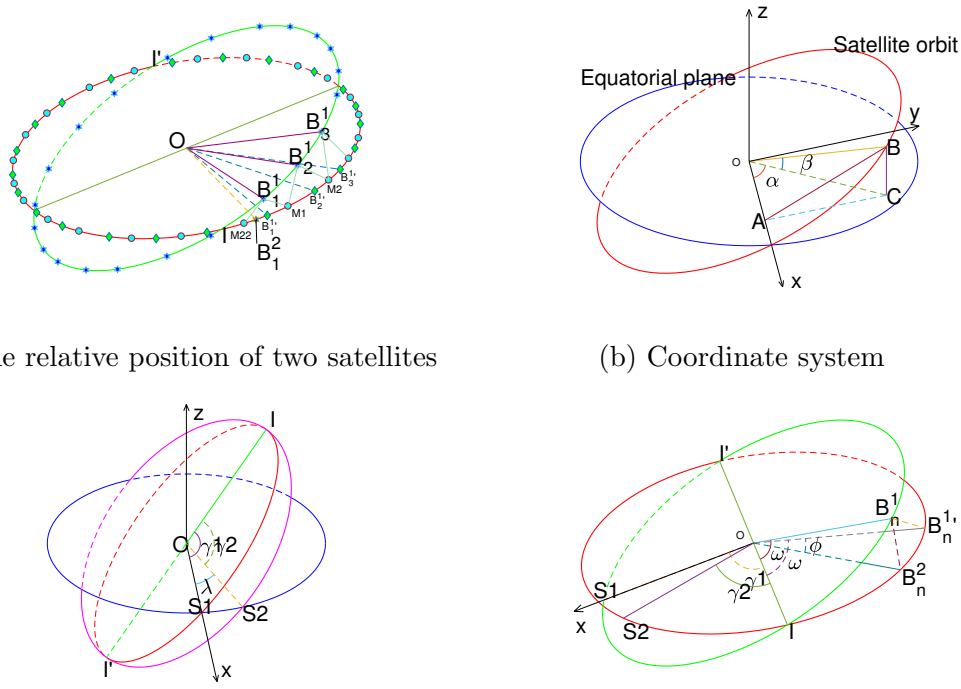
To quantify the performance of DPR, we first analyze the network topology using Starlink parameters [2].

4.2.1 LEO Topology

As satellites' movement is predictable, the LEO topology is predictable. According to Kepler's third theorem, the orbital period is $T = 2\pi\sqrt{r^3/\mu}$, where $\mu = 398600.441$ km^3/s^2 on Earth, and r is the radius of Earth plus the altitude of the orbit. For Starlink, we have the period of satellites equal to 5738.82 s. Since each packet can live in the LEO backbone network for at most several hundred milliseconds, the topology of satellite networks is assumed static during the lifetime of a packet. According to the movement patterns of satellites, any source satellite can easily estimate the network topology at any time.

The LEO topology is shown in Fig. 4.1, where the hexagram symbol denotes satellite. The black lines are SLs and the gray lines are OLs. Starlink has 72 orbital planes, while we draw 24 here for a clear illustration. Assuming the satellites are evenly distributed in an orbit, the length of each SL is $2r\sin(\pi/m_s)$, where $m_s = 22$ (for Starlink) is the number of satellites in an orbital plane.

As shown in Fig. 4.2(a), for circular orbits of Earth, there is an intersection line of any two orbits (red orbit and green orbit) through the geocenter. The two intersection points of the two orbits are denoted by I and I'. Denote the 22 satellites in the



(a) The relative position of two satellites

(b) Coordinate system

(c) Two neighbour orbital planes and the Equatorial plane

(d) The relative distance of two satellites

Figure 4.2: Distance model of satellites in neighboring orbital planes

green orbit by $B_1^1, B_2^1, B_3^1, ..., B_{22}^1$, marked in the blue hexagram. Their projections on the red orbit are denoted by $B_1'^1, B_2'^1, B_3'^1, ..., B_{22}'^1$, marked in green diamond, where $\angle IOB_n^1 = \angle IOB_n'^1 (1 \leq n \leq 22)$. In arcs $\overset{\frown}{B_n'^1 B_{n+1}'^1} (1 \leq n \leq 21)$, the middle points are $M_n (1 \leq n \leq 21)$, and in arc $\overset{\frown}{B_{22}'^1 B_1'^1}$ the middle point is M_{22}. We have 22 arcs which are $\overset{\frown}{M_n M_{n+1}} (1 \leq n \leq 21)$ and $\overset{\frown}{M_{22} M_1}$. Only one satellite will be located in each arc when the satellites are evenly distributed in the orbital plane, denoted by $B_1^2, B_2^2, B_3^2, ..., B_{22}^2$. Satellite B_n^1 and B_n^2 are the matched satellites in the neighbour orbital planes and their links are OLs. Next, we calculate the distance between two satellites and the OL distance.

4.2.2 Satellite Distance

We first define a convenient coordinate system for satellites to calculate their distance. Let the geocenter (point O) be the origin. The x-axis is on the lines of intersections between the equatorial plane and the satellite orbit plane. The z-axis is perpendicular to the equatorial plane. The y-axis is perpendicular to x- and z-axes, constituting

the right-handed system. Satellites are located in the spherical surface whose sphere center is O and the radius is r. The longitude ranges from 0 at the meridian, which passes through the forward x-axis, to π eastward and $-\pi$ westward, and the latitude ranges from 0 at the Equator to $\pi/2$ at the North Pole and $-\pi/2$ at the South Pole.

Each satellite's altitude and latitude are affected by the inclination of its orbit. As shown in Fig. 4.2(b), for a satellite located at point B, set its longitude and latitude angles as α and β. Point C is the projection of B in the equatorial plane (the blue circle). Line BA is perpendicular to OA, so CA is also perpendicular to OA. In $\triangle OBC$, $\angle BOC = \beta$, $|OB| = r$, so $|OC| = r\cos\beta$, $|BC| = r\sin\beta$. In $\triangle ABC$, the inclination of the satellite orbit is $\angle BAC = \theta$. In Starlink, $\theta = 53^\circ$ in all orbits, so $|AC| = r\frac{\sin\beta}{\tan\theta}$. In $\triangle OAC$, $\angle AOC = \alpha$, so $\sin\alpha = \frac{\tan\beta}{\tan\theta}$. In $\triangle OAB$, set $\angle AOB$ as the initial phase angle γ, $\sin\gamma = \frac{\sin\beta}{\sin\theta}$.

In our coordination system, the coordinate of B is $(r\cos\alpha\cos\beta, r\frac{\sin\beta}{\tan\theta}, r\sin\beta)$. For any two satellites with altitude and latitude angles of (α_1, β_1) and (α_2, β_2), respectively, their distance $L = r[(\cos\alpha_1\cos\beta_1 - \cos\alpha_2\cos\beta_2)^2 + (\sin\alpha_1\cos\beta_1 - \sin\alpha_2\cos\beta_2)^2 + (\sin\beta_1 - \sin\beta_2)^2]^{1/2}$.

4.2.3 Distance of Orbit-Links (OLs)

Point I is an intersection of two orbits. As shown in Fig. 4.2(c), γ_1 (γ_2) is the angle between OI and OS_1 (OS_2). We denote the latitude angle of I by β_I and the longitude angle by α_I. We have $\sin\alpha_I = \sin(\alpha_I - n\lambda)$ for $(n = 1, 2, ..., m_o - 1)$, where $m_o = 72$ is the number of orbital planes and $\lambda = 2\pi/m_o$. If two orbits are neighbours, $n = 1$. Then $\alpha_I = (\pi + \lambda)/2$. Set $\angle IOS1 = \gamma_1$ and $\angle IOS2 = \gamma_2$. We have $\gamma_2 = \arcsin[\sin(\arctan(\sin(\frac{\pi-\lambda}{2})\tan\theta))/\sin\theta]$ and $\gamma_1 = \pi - \gamma_2$.

For a satellite in the green orbital plane, B_n^1, its coordinate is (α_1, β_1), and its matched satellite in the red orbital plane, B_n^2, has the coordinate of (α_2, β_2). Define $\angle IOB_n^1 = \omega$, so $\angle IOB_n'^1 = \omega$ $(0 \le \omega < 2\pi)$. Define the angle between the two orbital planes as $\angle B_n'^1 OB_n^2 = \phi$, in the range of $-\frac{2\pi}{2m_s} < \phi \le \frac{2\pi}{2m_s}$. Next, we have $\angle IOB_n^2 = \omega + \phi$. As shown in Fig. 4.2(d), $\sin(\gamma_1 + \omega t) = \sin\beta_1/\tan\theta$ and $\sin(\gamma_2 + \omega t + \phi) = \sin\beta_2/\tan\theta$. Next, we have $\sin\alpha_1 = \tan\beta_1/\tan\theta$ and $\sin(\alpha_2 - \lambda) = \tan\beta_2/\tan\theta$ (keep α_1 and $\gamma_1 + \omega t$ in the same quadrant, and α_2 and $\gamma_2 + \omega t + \phi$ in the same quadrant). Finally, plugging in the formula of distance between two satellites in Sec. 4.2.2, we can calculate the OL distance.

4.2.4 Bond Probability and End-to-End Reliability

In Starlink, the satellite height is 550 km above Earth, still inside the atmosphere, so we can use the propagation channel model in Sec. 1. Given the OL and SL link distances obtained, and plugging into the channel model in Sec. 1, we can obtain the SNR of each ISL.

The demodulation error probability with M-ary pulse-position modulation is given by[14]

$$P_e = \frac{(M-1)}{(\pi \mathrm{SNR})^2} \exp(-\mathrm{SNR}/4). \tag{4.1}$$

The bit error rate is related to P_e by

$$\mathrm{BER} = 2^{k-1} P_e/(2^k - 1), \tag{4.2}$$

where $k = \log 2M$. Finally, the packet loss rate in the link can be expressed as

$$P_l = \exp\left(N \log(1 - P_e)\right), \tag{4.3}$$

where N is the packet length.

In percolation theory, each link is called a bond, and the probability that a link is connected is called the bond probability, p. Since with DPR, each packet is transmitted over a link once, the packet loss rate of a link thus equals one minus the bond probability. Given the lattice grid and the bond probability of each link, the end-to-end reliability (or connectivity) can be calculated using the recursive numerical method given in [20]. Based on the end-to-end reliability analysis, we can further tune on/off links in the network to make a tradeoff of transmission overhead and performance. Such a tradeoff can be observed from the performance evaluation below.

Chapter 5

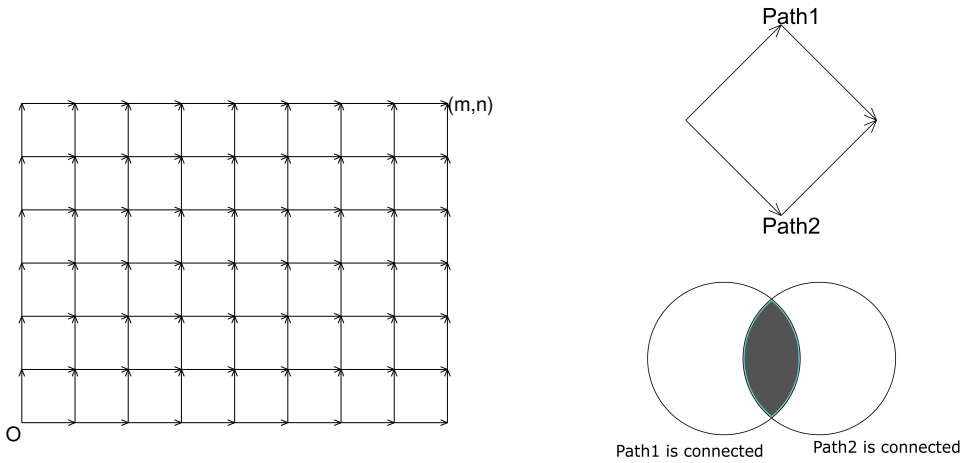
Performance Analysis

Different network scenarios correspond to the different topologies. As lattice networks have a wide range of applications, we here focus on analyzing 2-dimensional (2D) lattice networks in this section. We first present the complexity analysis of the proposed method for 2D lattice networks and compare it with the state-of-the-art. We then analyze the complexity and efficiency in dealing with irregular networks.

5.1 Connectivity of 2D Lattice Networks

In an $m \times n$ lattice network for directional percolation routing as shown in Fig. 5.1(a), each path from the source (at the origin) to the destination (at (m, n)) has m west-east links and n south-north links. Given the combinations of an arbitrary sequence of m west-east links and n south-north links, the number of the paths between the source and destination equals $\binom{m+n}{m}$.

For the source and destination to be connected, at least one of these paths should be connected. Using the Principle of Inclusion-Exclusion (PIE), to calculate the network connectivity (i.e., the source and destination are connected), we need to sum the probabilities of each path is connected, minus the probabilities that any pair of paths are connected, plus the probabilities that any triple of paths are connected, and so on. For instance, a 1×1 lattice network has two path as shown in Fig. 5.1(b). The reliability of the network is the sum of probabilities of two path minus the probabilities of two path both connected, as shown in the gray area. The complexity is dominated



(a) $m \times n$ (b) PIE principle

Figure 5.1: Lattice networks

by the number of path combinations, i.e.,

$$\sum_{i=1}^{\binom{n+m}{n}} \binom{\binom{n+m}{n}}{i} = 2^{\binom{n+m}{n}} - 1. \tag{5.1}$$

Thus the total complexity of the PIE approach is $O(2^{\binom{n+m}{n}})$, which is extremely high.

In [20], a recursive approach (RAC) to calculate the connectivity of lattice networks was developed. In this approach, the network is decomposed step by step, and the connectivity of a network (whether with the lattice topology or a decomposed tower topology) can be calculated based on the decomposed network connectivity. For an $m \times n$ lattice network, the complexity of the RAC algorithm in (5.2) is given by

$$\begin{aligned} &\sum_{i=1}^{n-1} \left[\binom{m+i}{i} - \binom{m+i-1}{i-1} \right] \cdot [1 + (m-1)(n-i)] \\ =&\sum_{i=1}^{n-1} \binom{m+i-1}{i} \cdot [1 + (m-1)(n-i)]. \end{aligned} \tag{5.2}$$

Using the big O notation, the complexity of this algorithm for an $n \times n$ network is $O\left(n^2 \frac{2n!}{n!(n+1)!}\right)$, which is asymptotically $O(\sqrt{n}4^n)$, much lower than that of PIE.

In our proposed approach, decoupling the network vertexes into sets with the same hop count, links are also divided into $m + n$ groups, where the links in each group

connecting two neighboring vertex sets. We then can consider cases one by one in an $m \times n$ ($m \geq n$) lattice. The complexity of the proposed hop-state algorithm as

$$2\sum_{i=1}^{n-1}\left[(2^i-1)(2^{i+1}-1)\right] + (m-n)\cdot(2^n-1)^2. \tag{5.3}$$

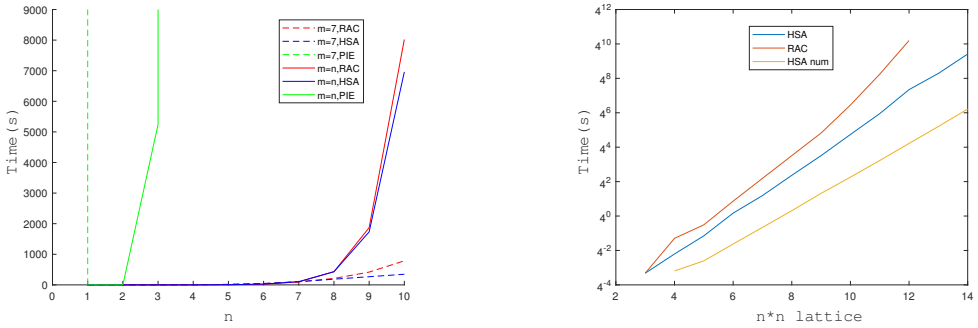
In other words, the complexity of the proposed algorithm is $O(4^n)$ for a regular $n \times n$ lattice. Comparing to the state-of-the-art in [20], we achieve a $\sqrt{n}$ reduction.

The theoretical detailed complexity performance comparisons for regular $m \times n$ lattice networks are given in Fig. 5.2(a). Here, the x-axis represents n. The dashed lines are for the cases $m = 7$, the solid lines are for the cases $m = n$. From the figure, the complexity (super-)exponentially grows with $\min m, n$ for all approaches. The green line is the PIE approach which has explosive growth and it reaches our computation capacity for 4×4 lattice networks. For the other two approaches, they can handle larger networks so long as $\min m, n$ is reasonably small. The red lines are for the existing RAC approach and the blue line is the simulation running time of our approach. Comparing to [20], the proposed HSA algorithm has substantially lower complexity and is more scalable.

The connectivity of $n \times n$ lattice network is a polynomial expression of p. For instance, the connectivity of 2×2 network is $p^4-6p^6-4p^7+2p^8+4p^9+2p^{10}-4p^{11}+p^{12}$. We have compared the connectivity results of lattice networks up-to 10×10 using HSA and RAC, and they give the same results. This validate the correctness of the two approaches.

Note that the coefficient of the connectivity expression becomes extremely large when the size of the network is large. For example, in 10×10, the biggest coefficient is $5.0537 * 10^{36}$ and in 18×18, the biggest coefficient is $2.8092 * 10^{125}$. Due to the coefficient sensitivity, these coefficients should be exact integers. Thus, it cost a lot of time at Bignum calculation. The blue line is HSA for calculating analytical expressions and the red line is for the RAC approach.

If we know the numerical link probability p of each link, the numerical calculation of connectivity is much faster. Orange line in Fig. 5.2 (b) gives the running time for numerical calculation of connectivity using the proposed HSA.



(a) The cost estimation (b) The cost actually by Log

Figure 5.2: Complexity comparison, for lattice network connectivity

5.2 Connectivity for Irregular Lattice Networks

In some realistic network scenarios, due to the physical limits or given the trade-off of link cost and connectivity performance, not all links in a lattice network can be available for data transmission. We compare the connectivity probability and the corresponding computation complexity for different irregular lattice networks.

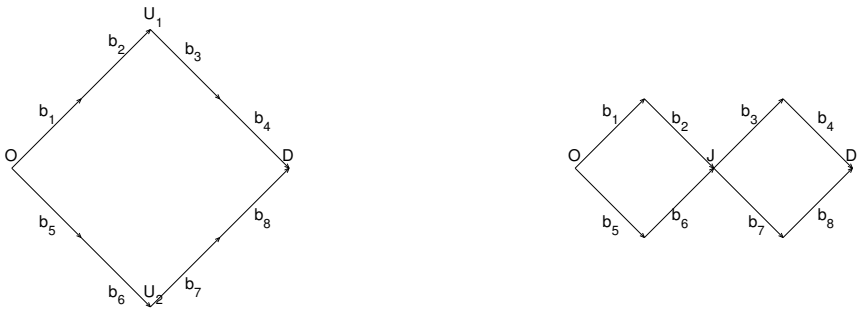
5.2.1 Joint vs. disjoint vertexes

If two paths passing the same vertices other than the source and destination, we named the vertices as joint vertices. As shown in 5.3, the source vertex is O and the destination vertex is D, the paths in Fig. 5.3(a) do not pass any common vertex other than O and D, but paths in Fig. 5.3(b) are joint at vertex J.

Given the same number of links, if there are more joint vertexes, the total number of directed paths will be higher. For example, Figs. 5.3(a) and (b) can be viewed as selecting 8 edges in a 2×2 lattice networks for delivering messages from O to D. The number of four-hop paths in Fig. 5.3(a) is two, and that in Figs. 5.3(b) is four, whose result is in higher connectivity according to the PIE principle.

Therefore, comparing the connectivity of two same-size networks with the same number of links, we conjecture that the one with more joint vertexes has higher connectivity. It is difficult to quantify how joint paths affecting the overall connectivity. We define a simple metric, $Jd = \frac{A_b}{A_v}$, to describe the degree of joint vertexes in a network, where A_b is the number of links and A_v is the number of vertexes in the network. We then conjecture that the same-size network with a larger Jd can have higher connectivity. This can be used as a guideline when we have the freedom to

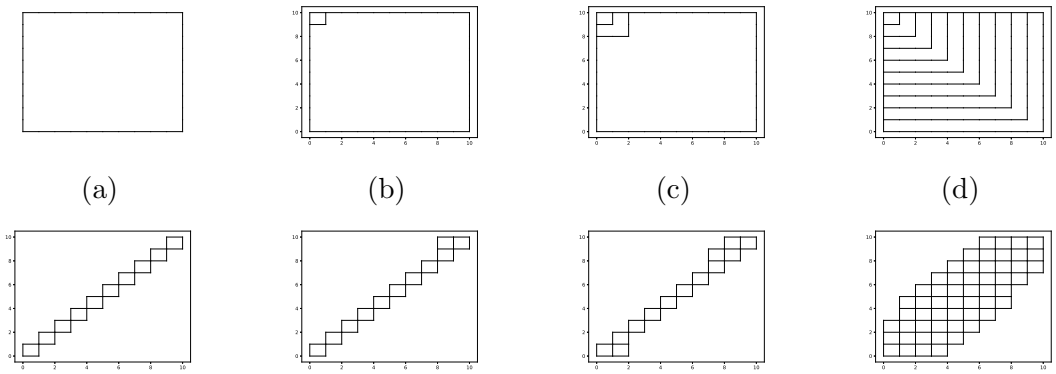
choose a limited number of links to be activated in a network while maintaining high connectivity/reliability.



(a) Disjoint vertex (b) Joint vertex

Figure 5.3: Networks with vs. without joint vertex

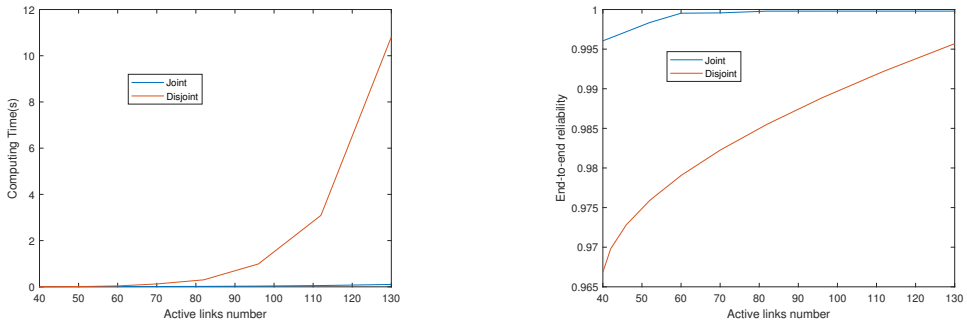
Given b_{num} active bonds, Jd has a maximum of $\frac{2b_{num}}{b_{num}+m+n+2}$ ($b_{num}-m-n$ should be even). When Jd is the maximum, we named the network as joint networks. We next give some examples to demonstrate the above guideline. We create networks with few joint vertices that follow the style of Fig. 5.4(a), (b), (c), and (d). Then, we create joint vertices networks with the same active link number that follows the style of Fig. 5.4(e), (f), (g), and (h). It takes more time to calculate the connectivity for disjoint networks than the joint networks, as shown in Fig. 5.5(a). Setting all link probabilities as 0.99, the end-to-end connectivity of networks is consistent with the theoretical results. Also, the joint networks have better connectivity than the disjoint ones, as shown in Fig. 5.5(b), which validates the proposed guideline.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5.4: Activate links of disjoint and joint networks



(a) Running time (b) Connectivity

Figure 5.5: Comparing computing time and connectivity of joint and disjoint networks

5.2.2 Variance of group size for each hop count

In addition to Jd, another metric that affects the connectivity is the variance of set size of different hop counts. For instance, as shown in Fig. 5.6, both networks are joint networks but the reliability of the network in (a) is probably higher than the network in (b) because of the bottleneck marked by the red circle. In order to quantify the degree of the bottleneck, we define the variance of the number of vertices in each hop set, σ^2, as another metric.

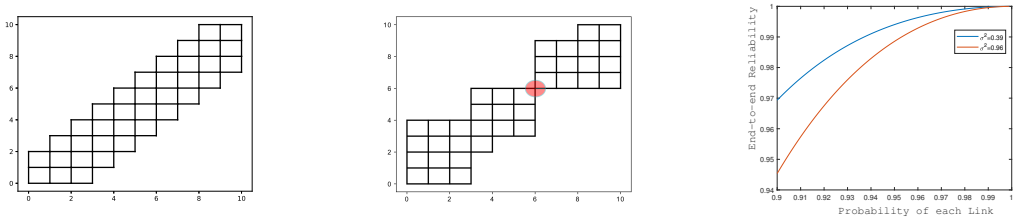
Using the variance formula,

$$\sigma^2 = \frac{\Sigma_1^{h_{\max}}(r_h - \bar{r})^2}{h_{\max} - 1} \tag{5.4}$$

Where $\bar{r}$ is mean value of all hop set, and $h_{\max}$ is the hop number of the destination. For example, for the networks in Fig. 5.6, the hop set sizes (for G_0, ... G_h are $1, 2, 3, 3, 3, 3, 3, 3 \ldots$ and $1, 2, 3, 4, 4, 3, 3 \ldots$ individually, and their variances are approximately 0.39 and 0.96, respectively. There end-to-end reliabilities of different link probability from 0.9 to 1 is shown in Fig. 5.6(c). Obviously, smaller variance leads to better results.

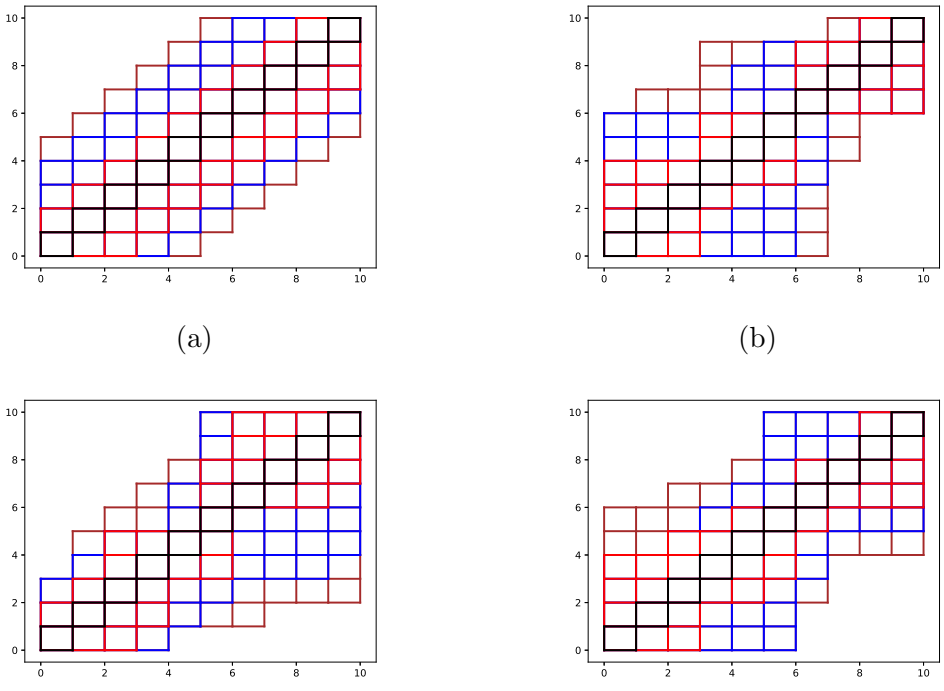
We run the simulation using a joint network following the style of Fig. 5.7(a) and some random joint networks following the style of Fig. 5.7(b), (c), and (d). First, black links become active, then red links become active, and then blue and brown links become active. Links of the same color have the same number. The simulation results are shown in Fig. 5.8. A larger variance often leads to some large size sets, resulting in longer computing time, as shown in Fig. 5.8(a). A smaller variance can generally lead to higher reliability, as shown in Fig. 5.8(b). Thus, we can rely on the

metric to select the links to achieve higher reliability.



(a) (b) (c)

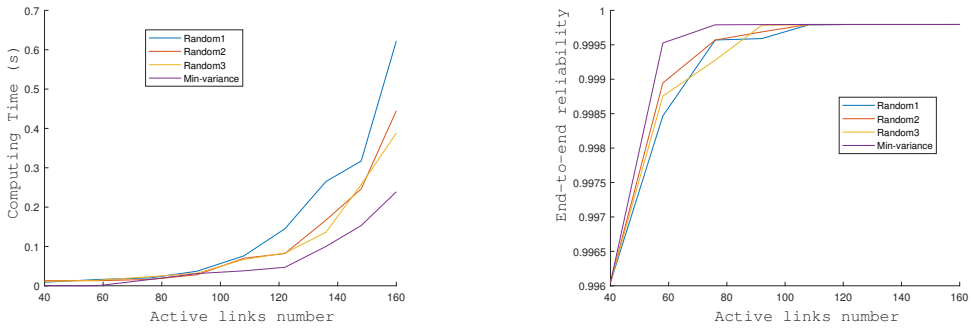
Figure 5.6: Networks with different variances in terms of sizes of hop count sets



(a) (b)

(c) (d)

Figure 5.7: Average and random joint networks



(a) Computing time (b) Connectivity

Figure 5.8: Simulation results, with different variances

Chapter 6

Performance Evaluation

6.1 Performance Evaluation of DPR

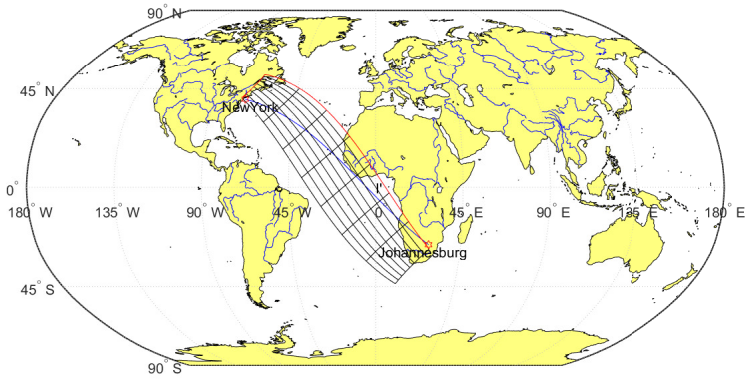
To evaluate the performance of DPR in LEO networks and validate the analysis, we conducted simulations using Starlink system parameters, and considered a few important inter-continent source-destination pairs.

Starlink covers the whole Earth except the Arctic and Antarctic, so all cities are always covered. Given the location of two cities (e.g., New York to Johannesburg and Toronto to Shanghai), at any time, we can easily identify the satellites covering the corresponding earth-stations. The quadrilateral satellite grid between New York and Johannesburg is plotted in Fig. 6.1(a), while the grid between Toronto and Shanghai contains a fold, as shown in Fig. 6.1(b). The blue line is the trace as the crow flies. The red line in the quadrilateral networks is a shortest path in the LEO network.

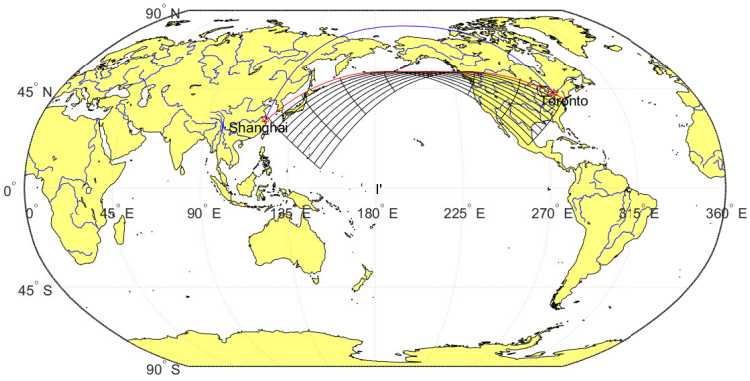
OL Distance – Following the analysis in Secs. 4.2.2 and 4.2.3, the distance of OLs can be calculated with a fixed gap ϕ. Fig. 6.2 shows the distance of OLs. Obviously, a smaller gap ϕ always has a shorter distance. Thus, $\phi = 0$ will lead to the shortest distance and the highest reliability.

Bond Probability – By calculating the distance of the SLs and OLs, we can obtain the bond probability (p) of each link using the formula in Sec. 4.2.4. The parameters are given in Table 6.1. The hardware parameters are from [16].

Fig. 6.3 shows the relationship between distance and bond probability with a fixed transmit power. From the figure, p is very distance sensitive. When the distance is above a threshold, p quickly drops to zero when the distance is further increased by 50 km. If we use a high transmit power, the system may waste energy substantially; if we use a low transmit power, some links with long distances can hardly deliver any



(a) New York-Johannesburg



(b) Toronto-Shanghai

Figure 6.1: Quadrilateral grid

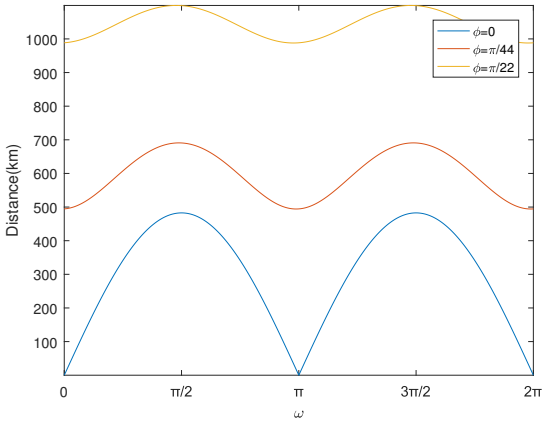


Figure 6.2: OL link distance

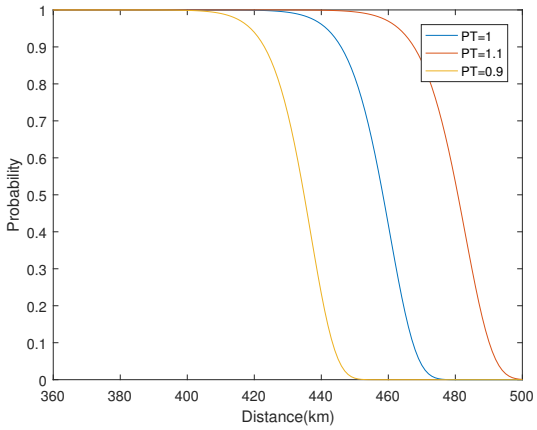


Figure 6.3: Probability of bonds

message successfully. To achieve reasonable performance, transmit power should be adjusted for different ISLs to keep p in a reasonable range. In the following, we set $0.9 \leq p \leq 0.999$.

Reliability – To investigate the tradeoff of link cost and performance, we can activate a part of ISLs in a grid. Given the nature of DPR, we use the number of active bonds (links) instead of path number for performance comparison. Here, the number of active bonds is proportional to the communication cost.

We consider the 10×7 quadrilateral grid network between Toronto and Shanghai as an example, where there are in total 157 links (bonds) in the grid. To deliver a packet from the source to destination, the least number of active bond number is 17 when a single 17-hop path is used. We chose the optimal single-path to obtain the path-reliability and tuned p for all links from 0.9 to 0.99, and we also investigated the scenario where the bond probability of each link is randomly chosen between 0.9 and 0.99. As shown in Fig. 6.4(a), using single-path routing, even when $p = 0.99$, the end-to-end reliability is only 0.85, not desirable for URLL. On the other hand, with the increase of the bond number, DPR achieves a higher reliability. When $p = 0.99$, using 61 active bonds can achieve a close to one reliability, so we can turn off the rest 97 to save cost. In the situation of congestion, we can inactivate those bonds with higher traffic loads. When p is as low as 0.9, we can still achieve above 0.98 end-to-end reliability when we activate about 100 bonds. Fig. 6.4 also shows that the simulation results match the analysis well. The dotted line shows DRP in heterogeneous bond probability (average of 0.95). We can found the performance is similar to the homogeneous bond probability case ($p = 0.95$) when there are more

Table 6.1: Simulation Parameter Settings

Parameters	Values
bandwidth (GHz)	400
pulse-position modulation	1024-ary
operating wavelength (nm)	1550
transmit power (dBm)	20
transmitter/receiver aperture diameter (mm)	150
transmitter/receiver optics efficiency	0.8
transmitter/receiver pointing error (μrad)	1.1
temperature (Kelvin)	289.85

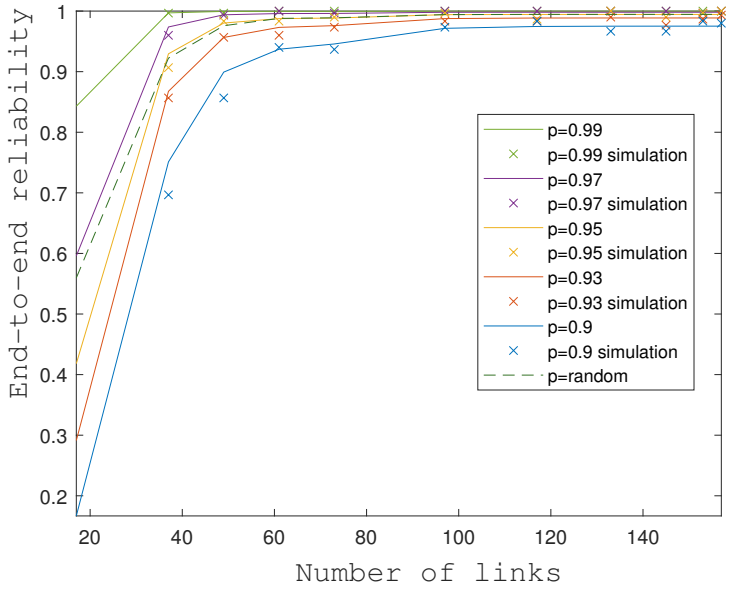
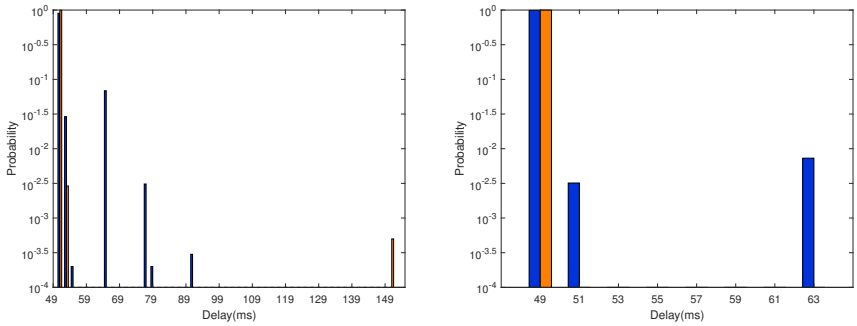


Figure 6.4: Active bond number vs. reliability

than 37 active bonds.

Delay – Next, we simulated the shortest path routing and DPR to evaluate the delay performance. For shortest single-path routing, we chose the path with the smallest delay between the two cities, and the lost packet will be retransmitted in the link layer. On the other hand, for the proposed DPR, the packet will be retransmitted only if timeout (when zero copy is reached in the destination and no end-to-end ACK is received by the source). For each setting, $10,000$ packets were transmitted and their delays were measured.

The resulting delay distribution when $p = 0.99$ and $p = 0.999$ are shown in Fig. 6.5(a) and (b), respectively, where the blue bars are for single-path and the orange bars for DPR. From the figures, not only the average delay of DPR is smaller,



(a) $p = 0.99$ (b) $p = 0.999$

Figure 6.5: Delay distributions, single path (blue) vs. DPR (orange)

Table 6.2: Reliability and Delay Performance

	Johannesburg – Rio de Janeiro	Shanghai – Toronto	Tokyo – Paris	New York – Canberra
Direct distance (km)	7100	11537	9887	16303
LEO path distance (km)	9405	14164	12096	18225
LEO grid	3×13	7×10	6×9	9×7
Loss rate w. DPR	$2.0e-6$	$2.0e-6$	$2.0e-6$	$2.0e-6$
Min delay reduction (ms)	4	11	9	21

more importantly, using the shortest-path routing, about 10% of packets suffering a delay jitter more than 20 ms when $p = 0.99$, while only 0.02% of packets suffering delay outage using DPR. When $p = 0.999$, about 1% of packets suffering a delay jitter around 14 ms using the shortest-path routing, and all packets can reach the destination within 50 ms using DPR. Thus, DPR is more desirable for URLL communication services.

We also chose four inter-continent pairs of Johannesburg and Rio de Janeiro, Shanghai and Toronto, Tokyo and Paris, and New York and Canberra to compare the performance in Table 6.2. To ensure URLL services, we have $p = 0.999$, and no link-layer or end-to-end retransmission is used. From Table 6.2, with DPR, the reliability crossing the LEO backbone is about 99.9998%. Furthermore, since the light propagation speed over the air is about 50% higher than that in fibre optical cables, even using the direct distance to under-estimate the propagation distance of fibre cables, the inter-continent backbone in LEO can still save at least 4 to 21 ms one-way propagation delay, making LEO a game changer for time-sensitive applications such as flash trading.

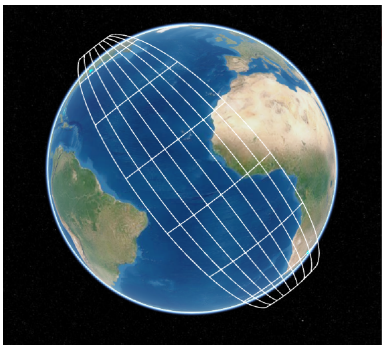
6.2 Performance Study and Reliability/Cost Trade-off

We next apply our approach to choose active links and obtaining the connectivity of the active network. Following this principle, it is clear that all lattice networks are regular multi-hop networks. We can build an efficient communication network with the least satellite resources and get the directed connectivity by HSA.

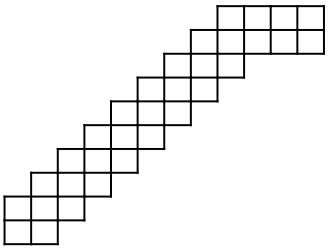
As shown in Fig. 6.6(a), here are all links between New York and Johannesburg so long as the next hop is closer to the aim satellite. Obviously, it is a typical lattice network with variable connectivity of each link. We consider the LEO satellite network as a topological two-dimensional grid which is a $7 * 8$ lattice network. Following keeping large Jd and small S^2, we make sure the number of active links between each neighbour hop are the same except triangles at both ends. Then, we can choose active links as shown in Fig. 6.6(b). If the network needs higher reliability, active links between each neighbour hop can be added as shown in Fig. 6.6(c). The final active links between satellites can be shown in Fig. 6.6(d).

Between New York and Johannesburg is the ocean, so satellites above the area are not busy. However, in other areas, for instance, between London and Tokyo, metropolises such as Paris, Vienna, Moscow, and Beijing are located there. The huge throughput of satellites above these metropolises may have some busy links. The red lines are busy links as shown in Fig. 6.8(a). The network when close to polar regions may bending occurs. However, the folding network also has the same lattice topological.

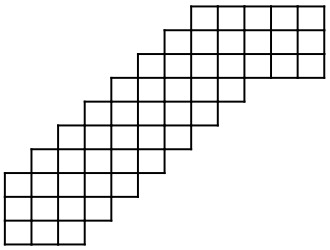
When we select links for message delivery, we can avoid the busy links to achieve better load balancing and performance. In the corners, the two links connected to the source and destination can be selected first, as shown in Fig. 6.7(a). In the middle, let the number of active links between neighbour hop be n_t. We start with one corner and choose the active links step by step. The end vertexes can be $n_t/2 + 1$ or $n_t/2$. When the end vertexes are $n_t/2$, we prefer the links in the middle unless they are busy links. In each step, as shown in Fig. 6.7(a), if all blue links are not busy and at least one green link is not busy, we can choose them; if one of the blue links or both green links are busy, we need to take a step back, and re-select the previous hop's links to probe for other better choices. With this recursive algorithm, the resulting link selection is shown in Fig. 6.7 (b), where the black links are selected for routing and the red links are the busy ones being avoided. After the recursive process, the



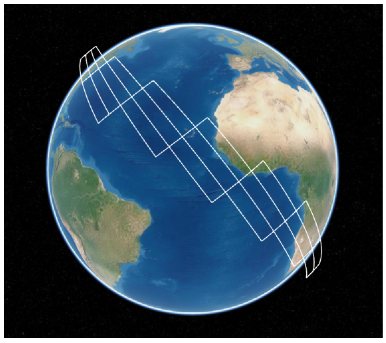
(a) All links between New York and Johannesburg



(b) Topological active links



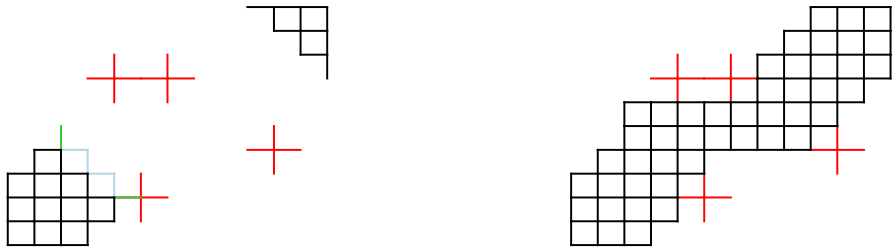
(c) More topological active links



(d) Active links between New York and Johannesburg

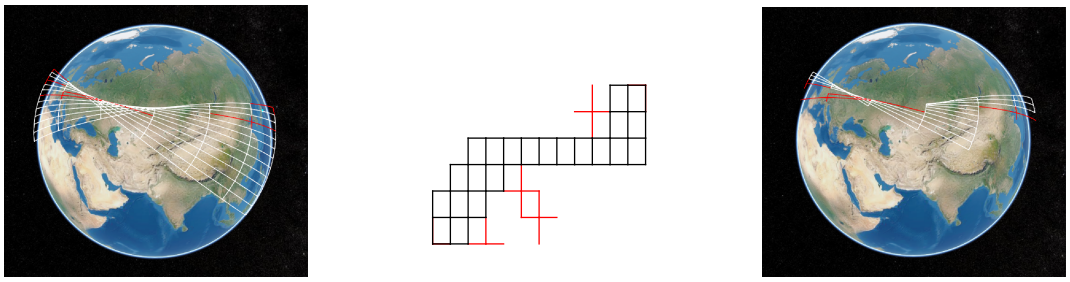
Figure 6.6: Simulate between New York and Johannesburg

topology of the active network is shown in Fig. 6.8(b) and its real active links are shown in Fig. 6.8(c). Our process to choose active links is shown in Algorithm 2.



(a) Corner of active links (b) Active links avoiding busy links

Figure 6.7: How to avoid busy links



(a) All links between London and Tokyo (b) Topological active links avoiding busy links (c) Active links between London and Tokyo

Figure 6.8: Simulate between London and Tokyo

Algorithm 2 Finding active links

Input: end vertexes
  for have not reach another corner's vertexes **do**
    if Blue links not have busy bonds **then**
      recall this code with input new end vertexes
    else
      if upper orange link is not busy bonds **then**
        recall this code with input new end vertexes
      else
        recall this code with input new end vertexes
      end if
    end if
  end for

Chapter 7

Conclusions

In this thesis, we can explore path diversity in a mesh network to ensure high reliability and low latency. Based on a Markov model, the closed-form network reliability as a polynomial expression of link reliability has been obtained using the proposed HSA, which is of lower computational complexity and more scalable than the state-of-the-art in the literature. Furthermore, we proposed two metrics that can be used for selecting the links in a network for routing to ensure performance while reducing link cost. From the analytical and simulation evaluation, exploring path diversity is an effective approach to ensure the reliability and delay performance for URLL services.

Then, we developed the DPR for the LEO backbone. First, we abstract the network into a quadrilateral topology. Each packet can be directly percolated in an $n \times m$ grid network to reach the destination. Furthermore, we can inactivate some ISLs to control the size of percolation to avoid congestion or reduce the transmission cost. From the analytical and simulation evaluation, the reliability and delay performance of DPR can support inter-continent URLL services.

Many open issues are beckoning further investigation. Given the unique features of the LEO satellite backbone, for instance, how to further reduce the computation complexity to analyze the reliability of an even larger network, how to optimize the link selection process to minimize the link cost given the reliability and delay constraints, and how to jointly optimize link parameters (such as transmission power, modulation, and coding configuration) and the link selection in routing for URLL services. For DPR, bonds can be heterogeneous and some bonds are more critical than the others, e.g., the two connecting to the source and the two to the destination. Applying a random forwarding probability in different parts of the networks to reduce overhead, ensure load balancing, and avoid congestion while maintaining the service

quality needs further investigation.

Bibliography

[1] 3gpp 33.825, study on the security of ultra-reliable low-latency communication (urllc) for the 5g system (5gs).

[2] Spacex gets ok to re-space starlink orbits. 20 December 2019.

[3] Ieee 802.1 time-sensitive networking task group. *technical report*, Release 16, 2019.

[4] M. Desai and D. Manjunath. On the connectivity in finite ad hoc networks. *IEEE Communications Letters*, 6(10):437–439, 2002.

[5] O. Dousse, P. Thiran, and M. Hasler. Connectivity in ad-hoc and hybrid networks. In *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1079–1088 vol.2, 2002.

[6] Chaofan Duan, Jing Feng, Haotian Chang, Bin Song, and Zhikang Xu. A novel handover control strategy combined with multi-hop routing in leo satellite networks. In *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 845–851, 2018.

[7] J. Gao and L. Guibas. Geometric algorithms for sensor networks. *Philosophical Trans. of the Royal Society A*, page 370(1958):27–51, 2012.

[8] A. Ghasemi and S. Nader-Esfahani. Exact probability of connectivity one-dimensional ad hoc wireless networks. *IEEE Communications Letters*, 10(4):251–253, 2006.

[9] Ingmar Glauche, Wolfram Krause, Rudolf Sollacher, and Martin Greiner. Continuum percolation of wireless ad hoc communication networks. *Physica A: Statistical Mechanics and its Applications*, 325(3):577–600, 2003.

[10] Martin Haenggi, Jeffrey G. Andrews, Francois Baccelli, Olivier Dousse, and Massimo Franceschetti. Stochastic geometry and random graphs for the analysis and design of wireless networks. *IEEE Journal on Selected Areas in Communications*, 27(7):1029–1046, 2009.

[11] Junhao Hu, Lin Cai, Chengcheng Zhao, and Jianping Pan. Directed percolation routing for ultra-reliable and low-latency services in low earth orbit (leo) satellite networks. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–6, 2020.

[12] Jian Li, Hancheng Lu, Kaiping Xue, and Yongdong Zhang. Temporal netgrid model-based dynamic routing in large-scale small satellite networks. *IEEE Transactions on Vehicular Technology*, 68(6):6009–6021, 2019.

[13] Xu Li, Feilong Tang, Long Chen, and Jie Li. A state-aware and load-balanced routing model for leo satellite networks. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, 2017.

[14] Arun K Majumdar. Free-space laser communication performance in the atmospheric channel. *Journal of Optical and Fiber Communications Reports*, 2, 2005.

[15] B. Varga N. Finn, P. Thubert and J. Farkas. Deterministic networking architecture. *Internet Eng. Task Force, Fremont, CA, USA*, Internet Draft draft-ietf-detnet-architecture-04, Oct 2017.

[16] Somanath Pradhan, P. K. Sahu, Rajat Kumar Giri, and Bijayananda Patnaik. Inter-satellite optical wireless communication system design using diversity techniques. In *2015 International Conference on Microwave, Optical and Communication Engineering (ICMOCE)*, pages 250–253, 2015.

[17] Tarik Taleb, Daisuke Mashimo, Abbas Jamalipour, Nei Kato, and Yoshiaki Nemoto. Explicit load balancing technique for ngeo satellite ip networks with on-board processing capabilities. *IEEE/ACM Transactions on Networking*, 17(1):281–293, 2009.

[18] Feilong Tang, Heteng Zhang, and Laurence T. Yang. Multipath cooperative routing with efficient acknowledgement for leo satellite networks. *IEEE Transactions on Mobile Computing*, 18(1):179–192, 2019.

[19] J. Pan Y. Zhuang and L. Cai. A probabilistic model for message propagation in two-dimensional vehicular ad-hoc networks. *Proc. IEEE INFOCOM'12*, page 1476–1484, 2012.

[20] Lei Zhang, Lin Cai, Jianping Pan, and Fei Tong. A new approach to the directed connectivity in two-dimensional lattice networks. *IEEE Transactions on Mobile Computing*, 13(11):2458–2472, 2014.

[21] Y. Zhuang and J. Pan. Random distances associated with hexagons. *arXiv:1106.2200*, 2011.

[22] Y. Zhuang and J. Pan. Random distances associated with rhombuses. *arXiv:1106.1257*, 2011.

[23] Y. Zhuang and J. Pan. Random distances associated with equilateral triangles. *arXiv:1207.1511*, 2012.