UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

# DELAY- AND DISRUPTION-TOLERANT ROUTING ALGORITHMS TO SUPPORT HUMAN ACTIVITY ON MARS

by

**Jason Jack Kamps**

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

March 2024

# SUMMARY

---

**DELAY- AND DISRUPTION-TOLERANT ROUTING ALGORITHMS TO SUPPORT HUMAN ACTIVITY ON MARS**

by

**Jason Jack Kamps**

| | |
|---|---|
| Supervisor: | Dr F. Palunčić |
| Co-supervisor: | Prof B.T. Maharaj |
| Department: | Electrical, Electronic and Computer Engineering |
| University: | University of Pretoria |
| Degree: | Master of Engineering (Computer Engineering) |
| Keywords: | Contact Graph Routing, Delay- and disruption-tolerant networking, DTN Routing, Interplanetary Internet, Schedule-Aware Bundle Routing. |

Deep-space activity is expected to increase rapidly in the coming decades. Most notably, crewed missions to Mars will take place. With humans venturing light minutes away from Earth for the first time, communication becomes challenging. Humans have specific communication needs that become difficult to support in deep space where large propagation delays, high error rates, and intermittent connections are prevalent. Delay- and disruption-tolerant networking (DTN) and the Bundle Protocol provide a reliable communication solution in such challenging environments. The overall performance of DTN protocols is highly dependent on their routing algorithms.

With Mars being humanity's next target in our exploration of the Solar System, this study deals with finding and examining the most suitable routing protocols in the context of Earth-Mars communication. Realistic scenarios of space missions are constructed to enable the comparison of various DTN routing algorithms in simulation.

Routing algorithm performance is analysed, and an enhancement to Contact Graph Routing (CGR) is proposed to address a deficiency of the algorithm, improving routing performance in networks featuring parallel channels.

The comparative analysis of routing algorithms provides valuable insight into their suitability for supporting crewed missions to Mars. This insight, along with the proposed CGR enhancement, contributes to the broader goal of a robust interplanetary Internet.

# Acknowledgements

I would like to express my gratitude to the following people and organisations for their invaluable support:

- My supervisor, Dr. Filip Palunčić, for his expert guidance and feedback.
- The SENTECH Chair in Broadband Wireless Multimedia Communications for their financial support.
- My wife, Jenna-Lee Kamps, for her unending encouragement.

# LIST OF ABBREVIATIONS

| | |
|---|---|
| APP | application |
| ARQ | automatic repeat request |
| CCSDS | Consultative Committee for Space Data Systems |
| CGR | Contact Graph Routing |
| COM | communication protocol |
| DSN | Deep Space Network |
| DTN | delay- and disruption-tolerant networking |
| ECMP | equal-cost multi-path routing |
| ESTRACK | European Space Tracking |
| FEC | forward error correction |
| FFA | Ford–Fulkerson algorithm |
| ION | Interplanetary Overlay Network |
| IP | Internet Protocol |
| ISL | inter-satellite link |
| LEO | low Earth orbit |
| MEX | Mars Express |
| MOC | mission operations centre |
| MRN | Mars Relay Network |
| MRO | Mars Reconnaissance Orbiter |
| MVN | Mars Atmosphere and Volatile Evolution |
| ODY | Mars Odyssey |
| PBAT | projected bundle arrival time |
| PRoPHET | Probabilistic Routing Protocol using History of Encounters and Transitivity |
| RF | radio frequency |

| | |
|---|---|
| RTT | round-trip time |
| SABR | Schedule-Aware Bundle Routing |
| SDR | spacecraft data recorder |
| SSC | solar superior conjunction |
| STK | Systems Tool Kit |
| TCP | Transmission Control Protocol |
| TGO | Trace Gas Orbiter |

# TABLE OF CONTENTS

# CHAPTER 1 INTRODUCTION

## 1.1 PROBLEM STATEMENT

### 1.1.1 Context of the problem

In the coming decades, deep-space activity is expected to increase rapidly. People will begin to live and work throughout the Solar System. Research bases and perhaps small colonies will develop on the Moon and Mars [1]. As the commercial space industry continues to develop, countless spacecraft will be launched into deep space [2]. These commercial activities are expected to include uncrewed missions such as asteroid mining and crewed missions such as space tourism. Deep-space missions require robust communication with Earth, a significant challenge given the large propagation delays, high error rates, and frequent link disconnections associated with space communication.

In order to support this growing deep-space activity, a networked system similar to the Internet is required, where data is communicated automatically from source to destination via arbitrary nodes. A networked system would allow communication infrastructure to be shared between multiple missions, provide redundant data paths, and shorten individual connections. As more missions and people are launched into deep space, data and bandwidth requirements increase. Just as the Internet connects people and devices on Earth, an interplanetary network is required to connect people and devices throughout the Solar System.

The creation of standards to support an interplanetary Internet began in the early 2000s when it was realised that Internet transport protocols are not able to operate in the challenging environment of space [3]. Instead, a new standard was developed to support delay- and disruption-tolerant networking (DTN). This standard, commonly known as the Bundle Protocol [4], acts as an overlay network operating between the application and transport layers in the protocol stack. The Bundle Protocol transports units of information, bundles, from source to destination, selecting lower-level protocols that are most suitable for each leg of a bundle's journey.

In the challenging environment of space, traditional routing protocols fail. Since end-to-end connectivity is not guaranteed, the Bundle Protocol employs a "store and forward" approach whereby bundles are incrementally transferred through a network to the destination as connections become available. In a given network, there are variety of possible routes from source to destination, and the overall performance of the network is highly dependent on the employed routing algorithm. To maximise the probability that a bundle is delivered, some routing algorithms generate many copies of bundles and transmit them via different routes in the hope that one is delivered successfully. Other algorithms route bundles more discriminately, carefully selecting routes based on knowledge of upcoming connections between nodes.

In humanity's exploration of the Solar System, Mars is an obvious first target. It is relatively nearby, it has a wealth of potential science to be performed, and it has many resources necessary to support human activity. To date, many unmanned missions have visited Mars in the form of orbiters, landers, and rovers. Communication in these endeavours is manually scheduled on a per-mission basis, typically with direct radio links between Earth and the spacecraft. Recent missions to Mars now make use of the Mars Relay Network [5]. This system uses Martian orbiters to act as relays between surface spacecraft and the Deep Space Network (DSN) on Earth, significantly increasing data throughput compared to a direct connection to the rovers. The rovers transmit data to the orbiters using a short-range, high-capacity channel. The data is stored and retransmitted when the orbiters establish high-capacity channels with the DSN using their more powerful antennae.

Communication in these unmanned missions to Mars typically consists of the transmission of control information and the return of scientific data. Human activity on Mars will result in unique requirements and challenges. Firstly, large quantities of data will need to be communicated. Humans necessitate the need for voice and video communication. Voice and video are the foundation of communication in existing crewed missions in low Earth orbit [6]. Crewed missions to Mars will also lead to many research endeavours, resulting in a significant volume of scientific data to be returned to Earth. Secondly, crewed missions also necessitate the need for reliable, constant, and low-latency communication. In the event of an emergency, for example, bidirectional communication must be available at the lowest possible latency. In such scenarios, delays incurred by communication link downtime, network congestion, or solar conjunctions are unacceptable.

### 1.1.2 Research gap

DTN routing has become an independent research area [7], resulting in a diverse range of available protocols. These protocols include epidemic routing [8], Spray and Wait [9], Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) [10], and multiple variants of Contact Graph Routing (CGR) [11]. In the context of future interplanetary networks to support human communication, it is not clear which routing protocol is most suitable. There also remains potential for adaptions and enhancement of existing protocols to be explored in this context.

Existing work compares DTN routing protocols in the context of pedestrians and vehicles roaming in a city [12]. Other works explore the performance of CGR in satellite scenarios [13] and in deep-space scenarios [14]. The comparison of a variety of disparate DTN routing algorithms is yet to be performed in the context of interplanetary networks. The optimisation of algorithms for the network topologies and communication requirements of crewed missions to Mars also remains an avenue for research.

The identification of the best-performing routing algorithm, as well as the potential improvement of such algorithm, will surely have effects on the productivity and safety of humans on Mars. If their communication requirements are better met, they will be able to interact better with information repositories, mission personnel, and their families on Earth. Additionally, performant routing protocols provide more reliable and timely communication, which is important in emergencies.

## 1.2 RESEARCH OBJECTIVE AND QUESTIONS

Three questions capture the research problem:

1. What are the communication requirements for human activity on Mars?
2. Which DTN routing algorithm(s) best meet the needs of human activity on Mars?
3. Can existing routing algorithms be adapted or improved to better support human activity on Mars?

The crux of the research problem is encapsulated by the second question. However, this question cannot stand alone. First, the communication needs of humans living and working on Mars must be established. This is not trivial to achieve since no humans have been to Mars. There are numerous plans for crewed Mars missions, some more ambitious than others. In establishing communication requirements, a realistic idea of the human presence on Mars in the near future must be determined, along with the communication infrastructure expected to support such presence. Next, to accurately investigate and

compare DTN routing algorithms, the scenario in which they are applied must be properly defined. This requires the diligent construction of simulations that emulate the expected mission characteristics and the corresponding communication infrastructure. Finally, during the investigation of the second question, the merits and shortfalls of each routing algorithm will be discovered. The identification of these characteristics may prove useful in the development of an improved DTN routing protocol.

## 1.3 APPROACH

To answer the research questions, the following procedure is followed. First, a broad literature study is conducted to determine the expected human presence on Mars in the near future. The plans of national space agencies and private space companies are considered. Best estimates for the number of missions, number of humans per mission, mission durations, and mission locations are determined. In addition, the communication infrastructure expected to support these missions is also determined. This includes the presence or absence of relay spacecraft in locations such as Martian orbit, cislunar space, and Lagrange points. The expected data rates, communication mediums (radio or optical), node storage capacities, and other similar details are considered. Once an estimate of the expected human activity on Mars is determined, the communication requirements of this human activity are established. These requirements include bandwidth, latency, and uptime.

After the requirements and expected infrastructure are established, realistic scenarios are constructed in which to perform simulations. This process begins with the configuration of physics simulations in Ansys Systems Tool Kit (STK), an advanced simulation tool used to model and analyse terrestrial and space systems. The locations of nodes in the scenario's topology are configured into the STK simulation, which then propagates the movement of the nodes over time, providing accurate node-node line of sight information. This information, combined with details of propagation delays and link capacities, forms a contact plan: a representation of the communication opportunities between nodes over time in dynamic networks.

With a contact plan, network simulations are then performed using DtnSim [15], a network simulation tool implemented in OMNeT++ [16], designed specifically for DTN investigations. Scenarios are simulated with various routing algorithms, and the resulting performance metrics (such as bundle delivery ratio, delivery cost, and latency) are analysed. To aid the analysis of the performance metrics, the maximum theoretical throughput of a given network is desired. This is determined through the application of the Ford-Fulkerson algorithm, a graph-theory-based method of calculating the maximum flow in a network.

The empirical data produced by the simulations provide quantitative evidence supporting the suitability of particular routing protocols in a given scenario. The software simulation approach also allows repeated tests where scenarios and parameters can be modified based on previous findings. An iterative approach can thus be employed whereby a working hypothesis can be repeatedly updated and tested in new scenarios.

## 1.4 RESEARCH GOALS

The research goals of this dissertation are multifaceted but are ultimately focused on the comparison and improvement of DTN routing algorithms in the context of crewed missions to Mars. First, the study endeavours to characterise the communication requirements essential for human activity on Mars. This involves an analysis of proposed missions to the planet to ascertain an understanding of the infrastructure that will be placed in deep space.

Secondly, this dissertation seeks to construct realistic scenarios in which to perform networking simulations. The results of these simulations will provide a basis for the comparison of routing algorithms. Through the simulation of different scenarios and varying iterations thereof, the performance of each algorithm and the factors that influence performance can be determined.

Finally, an important research goal is the novel enhancement of existing DTN routing algorithms to better align with the network topologies and communications requirements of human activity on Mars. The knowledge obtained through the analysis of routing algorithm performance in the constructed scenarios is to guide the development of such improvements.

## 1.5 RESEARCH CONTRIBUTION

This dissertation makes contributions to the field of DTN communication by comparing a diverse array of routing algorithms in the context of Martian scenarios. The research evaluates protocols such as epidemic routing, Spray and Wait, PRoPHET, and many CGR variants, offering an understanding of their performance in the challenging environment of interplanetary communication. By extending the analysis beyond terrestrial and satellite scenarios to the complexities of deep-space missions, this research addresses a gap in the current body of knowledge.

A key highlight of the research is a proposed enhancement to CGR, a pivotal DTN routing protocol. Although CGR exhibits respectable performance within the deterministic networks prevalent in deep space, there is room for improvement. Notably, CGR does not consistently utilise all available network resources and tends to underutilise the bandwidth and redundancy of parallel channels [17]. The proposed enhancement improves CGR's performance by distributing network load across all

available routes, performing most effectively in networks featuring concurrent paths and multiple data sources.

The comparative analysis of a range of routing algorithms provides valuable insights into their suitability for deep-space missions. The empirical evidence generated through simulations forms a basis for future advancements in the design and optimisation of DTN routing protocols, contributing to the broader goal of establishing a robust interplanetary Internet.

## 1.6 RESEARCH OUTPUTS

J.J. Kamps, F. Palunčić, B.T. Maharaj, "A Load-balancing Enhancement to Schedule-Aware Bundle Routing," *International Journal of Satellite Communications and Networking*, (submitted).

## 1.7 OVERVIEW OF STUDY

In Chapter 2, a review of the literature sets the stage for this study, providing a thorough exploration of DTN, particularly in the context of the interplanetary Internet. The Bundle Protocol is introduced, followed by a review and examination of existing DTN routing protocols. The literature review also explores the communication requirements and network infrastructure expected to accompany crewed missions to Mars, providing a foundation for scenario construction.

Chapter 3 discusses the research methodology employed in this study. The chapter details the process of scenario construction and the development of the simulation platform. This involves the selection of simulation tools and criteria for algorithm evaluation.

In Chapter 4, the experimental results are presented and discussed, offering a detailed analysis of the performance of various DTN routing algorithms in different scenarios. Performance metrics are examined, and graphical representations of the results contribute to an understanding of the suitability of each algorithm for crewed missions to Mars.

In Chapter 5, an enhancement to CGR is introduced, elucidating its functionality and demonstrating, via simulations, its capacity to improve performance in deep-space networks.

Chapter 6 presents concluding remarks and offers possibilities for further research.

# CHAPTER 2 LITERATURE REVIEW

## 2.1 CHAPTER OVERVIEW

This chapter presents the literature study and theoretical background for this dissertation. Section 2.2 discusses the challenges faced by communication systems in space which has led to the development of new protocols, explored in Section 2.3.

Section 2.4 presents the theoretical background of various DTN routing protocols that are investigated in subsequent chapters. The mechanics of CGR, a prominent routing protocol in deterministic networks, is examined in detail.

Section 2.5 provides an overview of forthcoming human expeditions to Mars, revealing mission attributes and anticipated infrastructure based on a review of existing literature. These findings help inform the parameters of the simulated scenarios constructed in Chapter 3.

## 2.2 SPACE COMMUNICATION CHALLENGES

Communication in deep space is vastly different to terrestrial communication. Typically, terrestrial Internet has high data rates with similar upload and download speeds, latencies measured in milliseconds, and continuous, on-demand, end-to-end connectivity. There are several challenges that DTN protocols need to consider:

### 2.2.1 Propagation delays

Space is vast. Relative to distances on Earth, distances between points in the Solar System can be several orders of magnitude greater. Since communication is constrained by the speed of light, which is the rate at which electromagnetic waves propagate, it can take minutes or even hours for signals to reach locations within the Solar System. For example, the distance between Earth and Mars varies between 55 million km and 400 million km, resulting in a communication round-trip time (RTT) of between 6 minutes and 42 minutes [18].

Besides making near-real-time communication or conversations impossible, large propagation delays

cause traditional Internet protocols to fail or become exceptionally inefficient [19]. Although latency in space is unavoidable, data must still be delivered reliably and within a predicted time frame [20].

### 2.2.2 Intermittent connectivity

Orbital mechanics and the movement of spacecraft result in the frequent interruption of point-to-point connections between communication nodes. For example, as satellites orbit, they are eclipsed by their planet, preventing communication with nodes on the distant side of the planet (occultation) [21]. The rotation and changing position of planets also cause link disruptions. Communication assets such as antennae stations and rovers located on a planet's surface rotate with the planet and are exposed to only a portion of the sky at any given moment.

Similarly, solar conjunctions can disrupt direct communication between nodes for weeks at a time. For example, every 26 months, Mars is located on the opposite side of the Sun to Earth, disrupting communication for approximately two weeks [22].

Intermittent connectivity is a significant challenge for communication protocols. Fortunately, the connectivity status of each link in a space network can often be predicted using knowledge of the orbital characteristics of each node.

### 2.2.3 Power constraints

Spacecraft need to be energy efficient. Many run on solar power and batteries with limited energy production and storage capabilities. Energy restrictions limit the transmission power of communication equipment, reducing signal strength. It is important for transmission and routing protocols to transmit sparingly, limiting overhead and avoiding unnecessary use of energy [23].

Power constraints of spacecraft and the large propagation distances also limit the data throughput of communication links. Terrestrial communication channels can often be measured in Gb/s. In deep-space channels, however, data rates are often only measured in kb/s [4]. Furthermore, spacecraft are typically able to receive more data than they can transmit. In some instances, communication links may be simplex. Asymmetric bandwidth and simplex links disrupt traditional transmission protocols which rely on feedback from the receiver.

### 2.2.4 High error rates

Due to the vast distances in space, signal attenuation results in high error rates. This challenge can be mitigated through a combination of techniques including forward error correction (FEC) and automatic repeat request (ARQ) [4]. These methods should be implemented carefully as FEC consumes additional bandwidth and ARQ significantly increases the latency of transmission.

## 2.3 DTN PROTOCOLS

The Internet Protocol (IP) is not suited for DTN as it requires continuous end-to-end connectivity [3]. To create a communication network that can operate reliably given the challenges of deep space, new protocols have been developed.

The Transmission Control Protocol (TCP) underlies most applications on the Internet. In TCP, a three-way handshake is performed before data is transmitted, lasting a duration of 1.5 RTTs [19]. Consider communication between Earth and Mars with a maximum RTT of approximately 42 minutes. In such a scenario, the handshake would consume 63 minutes before data transmission. Ignoring TCP's two-minute timeout which makes this communication impossible, TCP is impractical and inefficient when nodes are a significant distance apart.

### 2.3.1 DTN architecture

Since end-to-end connectivity is never guaranteed in DTN, data must be transmitted iteratively from node to node until it reaches its final destination. This method is known as store-and-forward [24]. It requires that each node in the network have storage capabilities. As seen in Figure 2.1, the store-and-forward method has the potential to significantly reduce latency and improve throughput in intermittently connected networks. This figure depicts the progression of data through a network comprising a source, destination, and two intermediary nodes. Under end-to-end IP, all available links (green) between nodes must be simultaneously available before data (blue) can be transmitted to the destination. Store-and-forward allows data to progress towards the destination as individual links become available.

Reliable end-to-end communication is ensured by employing custody-based transmission [26], where for any given unit of data, a node takes responsibility for it. The custody of the unit of information is only transferred to the next node once it has been confirmed that this unit has been correctly transmitted to the node.

In DTN, the addressing of devices on the network needs to be flexible. The exact location of the destination device is not always known at the time of original transmission. Since a message may take hours to reach its destination, the destination device may have moved to a new location on the destination network, with a new local address. Therefore, late binding is required, which means that the address can be defined or redefined at the source, in transit, or near the destination [27].

Security in DTN is also important. Unauthorised transmissions waste valuable bandwidth and node storage. Each node must therefore prevent the propagation of unauthorised traffic to conserve network

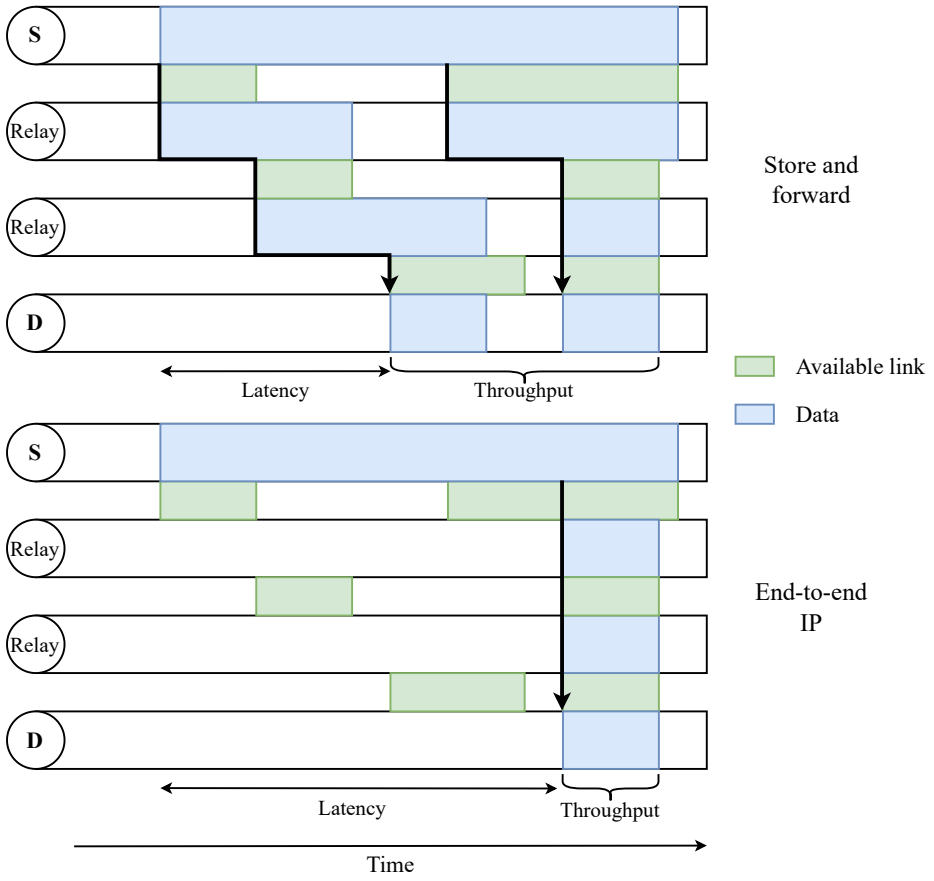



**Figure 2.1.** Store-and-forward visualisation in comparison to traditional end-to-end IP. Data is transmitted from source (S) to destination (D) through intermittent links between nodes. Adapted from [25] NASA, public domain, via Wikimedia Commons.

resources [4]. Authentication and encryption of data, although vital, must not require excessive overhead or processing cycles [20].

Given the challenges of DTN, advanced routing protocols are required. Routing in DTN must consider a large volume of information including network topologies, link uptime schedules, and traffic information. Unlike terrestrial Internet routing protocols which operate on the current, well-known, state of a network's topology, DTN protocols must operate in dynamic, intermittent networks where the current state of the entire network cannot be completely known [28]. Importantly, DTN routing protocols must manage congestion. Store-and-forward nodes have limited storage capacity which, in addition to bandwidth constraints, limits the throughput of the nodes.

### 2.3.2 Bundle Protocol

The Bundle Protocol [29] is an implementation of DTN architecture, allowing data to be transmitted through networks that experience large propagation delays and intermittent connections between nodes, also known as challenged networks [30]. The protocol acts as an overlay network, operating between the application and transport layers in the protocol stack.

Data managed by the Bundle Protocol is split into bundles: self-contained units of information that contain the actual data to be transmitted, along with additional metadata and control information. Each bundle consists of a primary block containing a variable-length, flexible, end-point identifier, along with sufficient information to allow the block to make progress through the network. Additional blocks contain the application data. Bundles can be fragmented into smaller bundles during transmission [3]. These fragments can be further fragmented or reassembled - allowing adaptability to current network conditions.

Bundling satisfies Burleigh's fundamental principles of DTN [4]. The first principle, a postal model of communications, states that data must be transmitted asynchronously, without waiting for confirmation of previous message reception. Per the principle, data units should also be self-contained, proactively including information such as authentication keys alongside the actual data, to avoid dialogue between the source and destination. The remaining two principles speak to the adaptability of the protocol to disparate environments and terseness, both of which the Bundle Protocol satisfies.

## 2.4 ROUTING PROTOCOLS

Just as traditional Internet transmission protocols fail in challenged networks, traditional routing protocols are also not suitable. Routers and their algorithms select the best path for packets using knowledge of the nodes and connections within a certain network. This information is stored in routing tables. In challenged networks, many links are unavailable for periods of time. Under Internet routing protocols, packets are dropped when their destination is absent from a routing table [31]. Additionally, due to propagation delays, updates regarding the connectivity of a network may be obsolete when arriving at a distant router [11]. These factors reveal that routing protocols designed specifically for DTN are required.

A variety of DTN routing protocols have been developed, and it remains an active research area [7]. In space networks, the motion of each node is typically defined by known orbital mechanics and is predictable. As such, the future topology of these networks can be computed. These deterministic networks allow the transmission of bundles to be scheduled in advance, meticulously forwarding

bundles between nodes. Conversely, some network topologies are stochastic in time. Routing in these networks is often based on flooding the network with copies of a bundle.

A variety of routing algorithms are investigated. An overview of candidate DTN routing algorithms is given below:

### 2.4.1 Epidemic routing

Epidemic routing [8] is a flooding-based method that operates by disseminating messages throughout the network akin to the spread of a contagion. Each node carries and transmits copies of a message to encountered nodes. The protocol, while ensuring timely message delivery, is criticised for its wasteful utilisation of network resources. The inherent flooding nature leads to significant bandwidth and energy consumption as multiple copies traverse the network. Epidemic routing is most suitable for transmitting a limited number of critical bundles, particularly in highly stochastic networks.

### 2.4.2 Spray and Wait

Spray and Wait [9] is a simple but effective routing algorithm that aims to achieve the delivery rate of flooding-based schemes while seeking the network efficiency of forwarding-based routing. Spray and Wait operates by replicating and broadcasting $L$ message copies to $L$ distinct neighbours only at the source node. These neighbouring nodes simply wait for an encounter with the destination. This technique is more efficient than epidemic routing, but its performance can be slow in large networks.

An alternative form of Spray and Wait, Binary Spray and Wait, is designed to disseminate messages faster through a network. As with regular Spray and Wait, the source begins with $L$ message copies, transmitting half of its copies to the next node it encounters. In fact, any node with $n > 1$ message copies transfers $\lfloor \frac{n}{2} \rfloor$ of its copies when it encounters a node with no copies, until it only has one copy remaining. Binary Spray and Wait is shown to be an optimal routing technique in stochastic networks where node movement is independent and identically distributed.

### 2.4.3 PRoPHET

Epidemic routing and Spray and Wait are both resource-intensive as they make little or no attempt to limit message replication. These strategies, particularly Binary Spray and Wait, are effective when communication opportunities between nodes occur randomly. In realistic scenarios, however, these encounters are not completely random. The Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) [10] aims to exploit the predictability of connections between nodes by storing probabilistic metrics at each node that indicate how likely it is for the node to deliver a

message to each destination. The probabilistic metrics are calculated using the history of encounters between nodes. Nodes share these metrics at each encounter, forming an understanding of the best routes through the network. By avoiding bundle duplication, PRoPHET is able to make efficient use of network resources compared to flooding-based techniques.

### 2.4.4 Contact Graph Routing

In deterministic networks, Contact Graph Routing (CGR) [11] is a prominent protocol used for the routing of Bundle Protocol bundles. CGR operates using contact plans which characterise the time-varying connectivity in a network. In CGR, each node in the network constructs a contact graph from contact plan messages, allowing each node to compute logical bundle forwarding decisions automatically. Despite its suitability in deterministic networks, CGR has some known limitations. The protocol may overbook forthcoming connections, causing congestion [32]. Furthermore, CGR has been found to overlook parallel channels, which if used, could improve path redundancy and increase bandwidth [17].

CGR has evolved over time as enhancements are proposed and improvements are implemented. For example, one proposed extension uses local traffic information to predict congestion using known path capacities [33]. The technique was shown to improve data delivery ratios and latencies, while minimally increasing processing complexity. Other enhancements to CGR include overbooking management [34] and extensions to support both deterministic and stochastic networks through opportunistic routing [26].

CGR is said to be especially advantageous for networks in space, as it can take advantage of the highly predictable contacts between nodes [35]. A detailed overview of CGR is thus presented here. More specifically, the latest version of CGR standardised by the Consultative Committee for Space Data Systems (CCSDS) [36] is explored. This version is known as Schedule-Aware Bundle Routing (SABR).

SABR, summarised in Figure 2.2, has three main stages:

1. **Planning**: A contact graph is constructed from a contact plan to represent the evolving communication opportunities within the network.
2. **Routing**: Using the contact graph, routes that offer the earliest arrival time are computed and placed in a candidate route list after being validated.
3. **Forwarding**: The best path from the candidate route list is selected for each bundle based on

current conditions. Bundles with the same destination may have different routes depending on the time they are transmitted, their priority, and local traffic conditions.

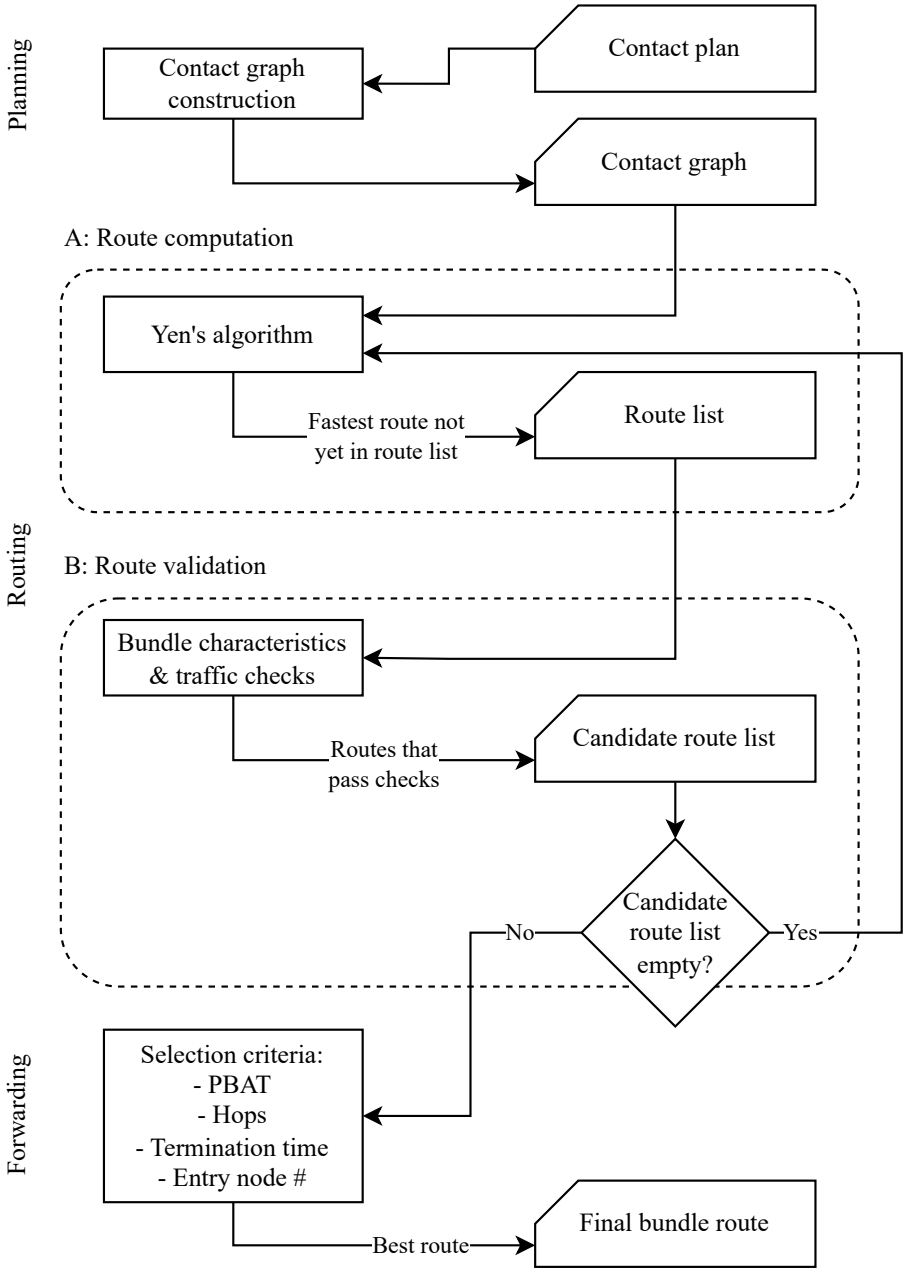



**Figure 2.2.** Overview of SABR.

#### 2.4.4.1 Planning

In certain networks, especially space-based networks, the upcoming communications opportunities between nodes are known. For example, orbital propagation software can accurately predict the line-of-sight periods between each node over time. These periods are contacts: intervals during which data can be transmitted from one node to another at a certain rate.

As such, the entries in a contact plan take the form:

```
contact <start time> <end time> <source> <destination> <rate>
```

Contacts are unidirectional since there is no guarantee that the data rates are the same in each direction. Transmission power is often significantly lower on spacecraft relative to terrestrial equipment. Furthermore, contacts may begin and end at different times depending on the direction. In scenarios with significant propagation delay, nodes often preemptively transmit data just before they have a line of sight. When nodes orbit a planet, this phenomenon results in asymmetric contact plans [35]. In some cases, the data rate can vary throughout the duration of a contact. This can be taken into account in the contact plan by specifying multiple consecutive contacts with different rates.

As mentioned above, contact plans are computed in advance and subsequently distributed to all nodes. This computation is typically performed at a central location such as mission control. The plans are distributed well in advance to account for any delays during propagation through the network.

A contact graph, a structure that efficiently represents the time-evolving topology of a network, can be constructed from a contact plan. An example contact graph, illustrated in Figure 2.3, represents the contact plan given in Table 2.1.

**Table 2.1.** Simple contact plan between five nodes over one minute. Adapted from [35], with permission.

| Start time (s) | End time (s) | Source node | Destination node | Data rate (bytes/s) |
|---|---|---|---|---|
| 0 | 60 | A | B | 1 |
| 0 | 60 | B | C | 1 |
| 0 | 60 | A | C | 1 |
| 0 | 30 | C | D | 1 |
| 10 | 20 | A | E | 1 |
| 0 | 10 | D | E | 1 |
| 30 | 40 | D | E | 1 |
| 50 | 60 | D | E | 1 |

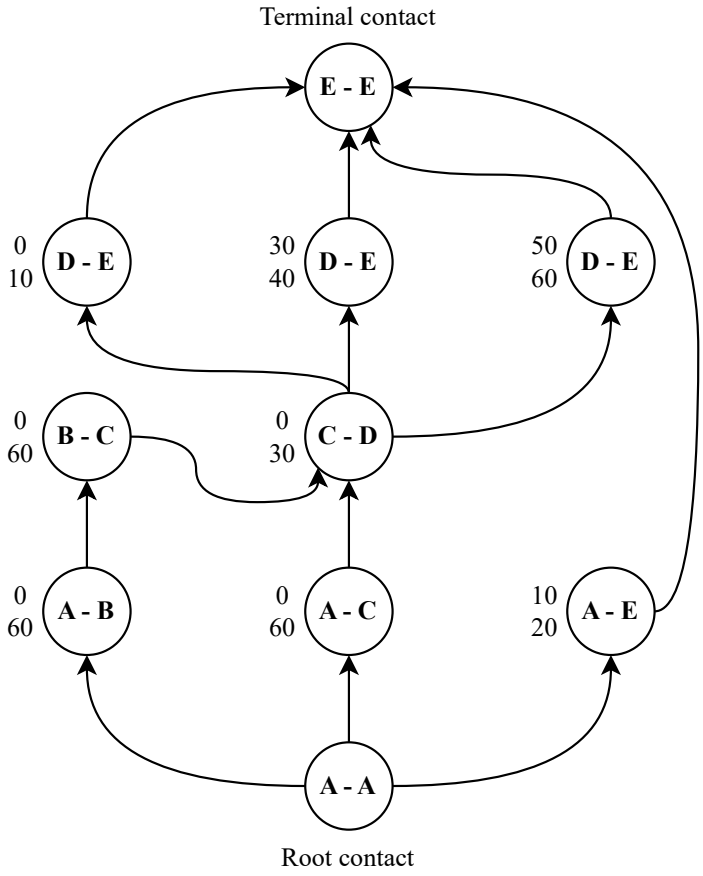



**Figure 2.3.** Contact graph of the contact plan given in Table 2.1. Adapted from [35], with permission.

Notice that the vertices of the contact graph are contacts, not network nodes. The edges show periods of time where data must be stored at a node between contacts. For example, after data is transferred from node C to node D between 0 s and 30 s, the data must be stored at node D (outgoing edges of the C-D vertex), before it can be transferred to node E by a subsequent contact. Only a contact graph is required for the computation of routes from a source node to a destination.

#### 2.4.4.2 Routing

**Part A: route computation**

The contact graph is searched to find the shortest paths between source and destination. The details of this search are not defined in the SABR standard and are left as an implementation matter. Yen's $K$ Shortest Path algorithm, however, is recommended by the standard and is ubiquitous in modern CGR implementations.

Yen's algorithm [37] returns the $K$ shortest paths between nodes in a graph. For $K = 1$, any shortest path algorithm is used, such as Dijkstra's algorithm [38]. To find additional short paths ($K > 1$), routes

are systematically explored by removing edges one at a time to force alternative paths. Yen's algorithm is described further in Appendix A.

It is not typically sufficient to compute a single route to each destination. Multiple routes are required since routes expire, have limited volume, and communication links may fail unexpectedly [35]. Contacts have a finite duration, thus the validity of any single route ends with the expiration of any one of its contacts. A contact may be exhausted for bundles of a certain priority, and an alternative must be used. Finally, multiple routes allow for redundancy, and, if exploited correctly, additional concurrent bandwidth.

At first, only a single route is computed and stored in a list of possible routes. Yen's algorithm provides an efficient method to discover additional routes when such routes are required.

**Part B: route validation**

Next, each route in the route list is examined to determine if it is suitable given the characteristics of the current bundle and local traffic information. A route is unsuitable if it meets any of the following criteria [36]:

- The best-case delivery time is after the bundle's expiration time.
- The route's first node is in the list of excluded nodes (helps avoid backward propagation).
- The route includes the current node at a later point in time (avoid routing loops).
- The route's existing backlog results in an earliest transmission opportunity that is after the end of the route's initial contact.
- The route's projected bundle arrival time (PBAT) is after the bundle's expiration time.
- A contact in the route is expected to have insufficient capacity for the bundle given its priority (the route is depleted).

All suitable routes are placed in the candidate route list. At this stage, if the candidate route list is empty, Yen's algorithm is performed again to find alternative routes.

#### 2.4.4.3 Forwarding

After establishing a list of candidate routes, the best route is chosen according to the following criteria:

- The route with the earliest PBAT.
- If equal, the route with the fewest number of contacts (fewest hops).
- If equal, the route with the latest termination time.

- If equal, the route with the smallest entry node ID (arbitrary selection).

The route with the earliest PBAT is thus selected and if there are multiple routes with the same PBAT, then tiebreakers are employed to select a single route.

If a bundle is marked as critical, then it is forwarded to all neighbouring nodes for which there is a candidate route to the destination. This process requires multiple executions of the route computation and route validation to find additional candidate routes.

## 2.5 HUMAN MISSIONS TO MARS

### 2.5.1 Planned Missions

The prospect of human missions to Mars has garnered significant attention, leading to many proposals. These proposals range from near-term to long-term missions. The near-term missions envisioned to commence as soon as the early 2030s involve a team of 4-6 astronauts embarking on a journey that will last 2-3 years and entail a stay of approximately 500 days on the Martian surface. The long-term missions are more ambitious and entail the establishment of a permanent human settlement on Mars, complete with robust infrastructure and a larger population. However, these ambitions are only likely to be realised several decades into the future.

The latest NASA Design Reference Mission [39] envisions the dispatch of a team of six astronauts to the Martian surface for a 500-day stay, with a travel time of approximately 180 days in each direction. The objectives of this mission include the pursuit of evidence for ancient life, the investigation of the Martian surface and interior evolution, and the preparation for a sustained human presence on the planet.

NASA has also devised a deep-space habitation strategy [40], which outlines its plan for the progressive development of deep-space habitats, starting from Low Earth Orbit to cislunar space, and ultimately, Mars. This strategy includes the deployment of crews of 4 astronauts for durations ranging from 300 to 1100 days in deep space.

The Mars Base Camp concept developed by Lockheed Martin [41] entails a 6-astronaut orbital mission, which may also involve brief visits to the Martian surface. The mission duration is expected to be 970 days, with 300 days spent in Martian orbit.

A proposal using solar electric propulsion [42] entails the transportation of a 6-member crew to Mars in 190 days, followed by a 528-day stay on the surface, with a return journey of similar duration. Solar

electric propulsion offers substantial advantages in terms of reduced mass compared to traditional chemical propulsion.

SpaceX has a grand vision of establishing a self-sufficient human settlement on Mars and advancing humanity as a multi-planetary species [43]. Their strategy is anchored on the utilisation of their Starship vehicle (currently under development), which is capable of transporting large quantities of equipment and dozens of people to the surface of Mars. The first phase will involve uncrewed Starship missions carrying supplies, followed by crewed missions transporting 10-20 people per Starship. The company's inaugural mission is expected to involve 20-40 individuals in two crewed Starships, with the aim of doubling the number of landed Starships with each subsequent trip. The scope and speed of SpaceX's plans set them apart from other organisations, as they aim to swiftly establish substantial infrastructure and a permanent human presence on Mars [44], unlike smaller, scientifically focused exploratory missions.

### 2.5.2 Infrastructure

In addition to the parameters of the proposed human missions, it is crucial to understand the infrastructure that will support these missions. The simulated scenarios must accurately represent the communication equipment and network topologies that are expected to be employed. Literature has been consulted to determine the expected communication infrastructure that will support crewed missions to Mars.

#### 2.5.2.1 Current infrastructure

Presently, communication with Martian spacecraft is carefully managed on a per-mission basis. Ground stations, such as the Deep Space Network (DSN) or the European Space Tracking (ESTRACK) network, communicate directly with spacecraft on and around Mars. In many cases, orbiters serve as intermediaries between Earth and surface spacecraft such as landers and rovers. The Mars Relay Network (MRN) consists of 5 orbiters that support communication with various surface spacecraft, including InSight, Curiosity, and Perseverance. Due to mass, power, and volume constraints, landed missions typically have smaller and less powerful antennae, resulting in inferior communication capabilities relative to orbiters. The MRN addresses this problem by providing short-range, high-capacity data links between surface spacecraft and orbiters, with the orbiters assuming responsibility for the long-distance communication with Earth [45]. This vastly increases data throughput between surface spacecraft and Earth since their inferior long-range antennae need not be used. Furthermore, the MRN provides additional communication opportunities as the orbiters are able to relay data from the far side of Mars.

With humans venturing to the Red Planet for the first time, Martian communication infrastructure must be significantly improved. The MRN is an ageing network and will not support the increased demands of human missions [46]. At present, the MRN consists of orbiters with scientific missions; their communication relay functions are of secondary importance. A dedicated communication infrastructure in Martian orbit is required to increase data rates and redundancy to support the more stringent communication requirements of humans.

Using mission activity modelling, Tai [47] expects that the maximum demands of early human missions will be 28.4 Mb/s in the Earth-to-Mars direction (uplink) and 211.9 Mb/s in the Mars-to-Earth direction (downlink). These data rates are consistent with NASA's working requirements [48]. Currently, the MRN only returns a combined 3.5 Gb per solar day on Mars (sol) from the robotic spacecraft [49]. The discrepancy in data rate requirements between human and robotic missions is based on the type of data that each category of mission requires. Unlike robotic spacecraft, astronauts require voice and video communication [50]. Each astronaut will also capture engineering data, images, and videos that must be returned to Earth [51]. As with the Apollo missions and current activity on the International Space Station, public interest in human space endeavours also demands the live broadcast of this activity.

#### 2.5.2.2 Future infrastructure

There are various network configurations and technologies that can be implemented to achieve human communication requirements. These will be briefly discussed.

Mars has already entered the "networked era" with the MRN. This network can be improved by placing dedicated communication relays into Martian orbit. Additionally, future scientific missions can be given secondary communication relay functionality to expand the network [52].

Early crewed missions are likely to be supported by orbiters in areostationary orbits [5][47][51]. A satellite in such an orbit is in a circular orbit above the equator, with a period equal to the rotational period of Mars. Thus, areostationary orbiters appear stationary in the sky from the ground. Two relay orbiters can be placed in an areostationary orbit with coverage of a candidate Mars base, depicted in Figure 2.4. The orbiters will remain in permanent contact with the Mars base and Earth, except during occultations [51]. At any given moment, at least one orbiter will maintain a direct line of sight with Earth. When neither is occluded, the pair will provide redundant and concurrent communication channels. This configuration ensures that the Mars base maintains continuous contact with Earth, even when the base is on the far side of Mars.

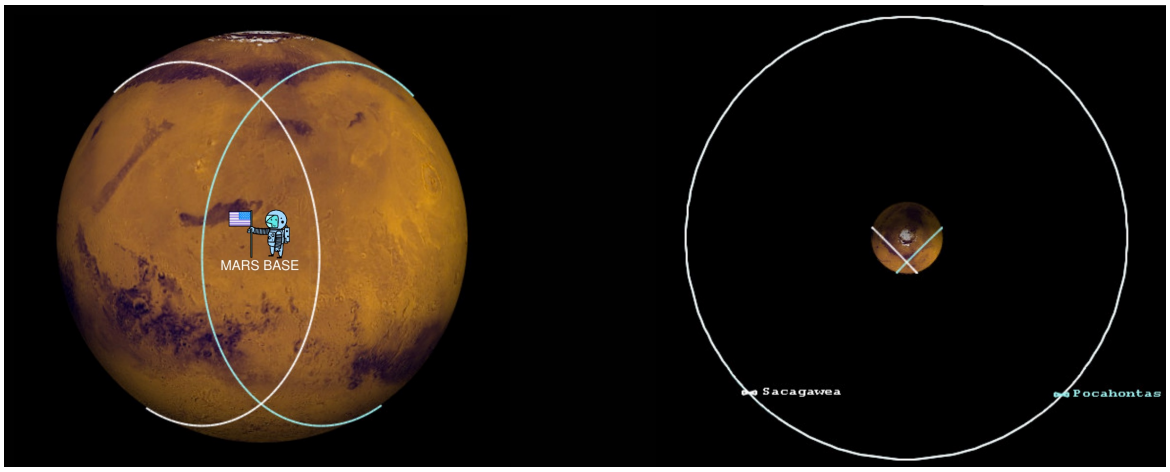
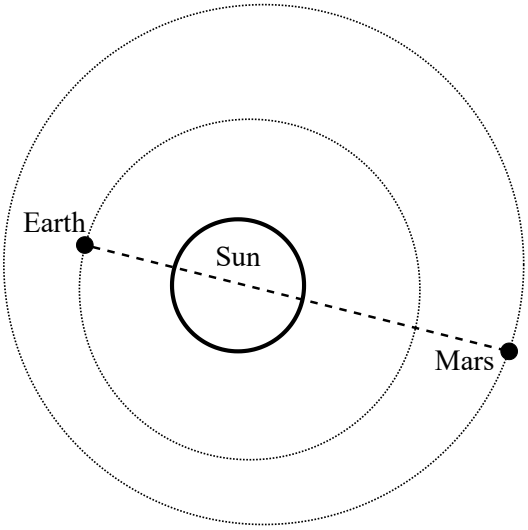



**Figure 2.4.** Two areostationary orbiters. Taken from [51] © 2005 IEEE.

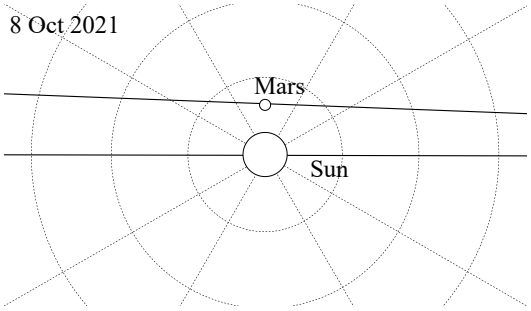
Communication will be interrupted, however, when a solar superior conjunction (SSC) occurs. These events, depicted in Figure 2.5(a), occur approximately every 26 months and can interrupt communication for multiple weeks at a time [53]. Electromagnetic signals are severely disrupted by charged particles from the Sun's corona, particularly those with high data rates. Since the orbital planes of Earth and Mars are not perfectly aligned, SSCs do not always result in occultations, as seen in Figure 2.5(b) and Figure 2.5(c), but communication is nevertheless severely disrupted. Basic communication can often be maintained through conjunctions without occultations [21], but the disruption is unacceptable for human missions. To mitigate interruptions and maintain sufficient communication, a relay must be employed to bypass the Sun.

A strong candidate for relay locations is a Lagrange point within the inner Solar System. Each planet has five Lagrange points (L1 through L5), as seen in Figure 2.6, with itself and the Sun where gravitational forces balance such that objects are gravitationally locked to the planet at that point. Therefore, an object placed in these locations will remain stationary with respect to the planet as it orbits the Sun. Signals can be relayed around the Sun by a spacecraft located at an L4 or L5 point. Although the Lagrange points of Mercury, Venus, and Mars can be utilised, the Earth-Sun L5 point is most appealing [53]. It is considerably cheaper to launch to Earth Lagrange points, and minimal additional communication delay is incurred relative to points at Mercury or Venus. Earth's L4 point is less desirable because it contains a Trojan asteroid. Lagrange relays would primarily be used during SSCs and would provide redundancy to direct communication during non-interference periods.

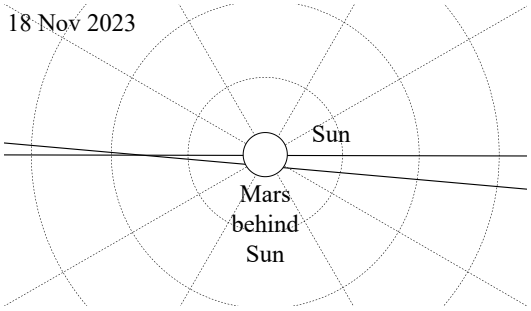
An alternative SSC mitigation solution would be to place relay spacecraft in a heliocentric orbit near Mars [55]. One particular orbit that has the same period as Mars, with a different eccentricity and inclination than that of Mars, would cause the spacecraft to rise above Mars, then fall behind Mars




**(a)** Conjunction geometry.




**(b)** Conjunction without occultation, as viewed from Earth.




**(c)** Conjunction with occultation, as viewed from Earth.

**Figure 2.5.** Mars-Sun-Earth Superior Conjunctions. Adapted from [21] © 2002 IEEE.

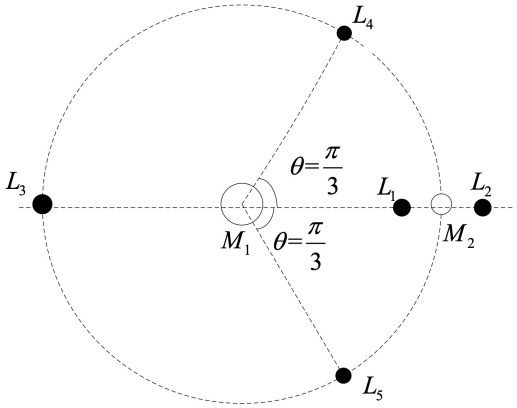



**Figure 2.6.** Lagrange points, where $M_1$ is the Sun and $M_2$ is the Earth. Taken from [54] © 2017 IEEE.

before dropping below and then pulling ahead of Mars by approximately 20 million kilometres in each direction. From the perspective of Earth, the spacecraft would maintain sufficient angular separation from the Sun to avoid SSC-induced interference.

Communication between Earth and Mars is challenging, in part, due to the large distance between the planets. With an average distance of 225 million km, powerful communication equipment is necessary to overcome signal attenuation. One solution that aims to improve communication between the planets involves a network of relay satellites placed between the planets [56]. The network would consist of multiple concentric rings of relays in heliocentric orbit between the orbits of Earth and Mars. To ensure that there are always relays between the planets, multiple evenly spaced satellites would be placed in each ring. The resulting network topology, depicted in Figure 2.7, would allow Earth-Mars communication signals to undertake shorter hops between nodes, avoiding the attenuation of long-distance links, and thereby improving data rates. Considering the number of satellites necessary and their orbits, this would be an expensive endeavour and may not be appropriate for early missions to Mars.

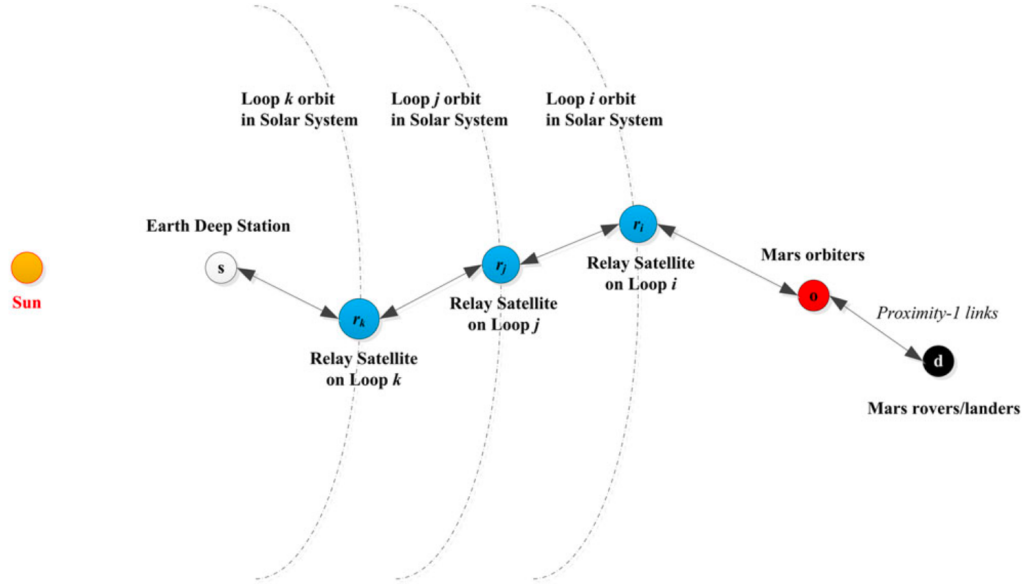



**Figure 2.7.** Multiple-hop relay constellation. Taken from [56], with permission.

The data rate between Earth and Mars is an important metric in the design of the interplanetary communication system. As discussed, some estimates put the expected demand at approximately 30 Mb/s uplink and 250 Mb/s downlink. Other estimates suggest that this is not sufficient, particularly if high-resolution video dominates the bandwidth [57]. Some NASA documentation indicates 1 Gb/s is desired by 2030 [58]. Noreen [51] predicts 440 Mb/s downlink is required while Howard [53]

indicates demand could reach 2 Gb/s in specific video-heavy scenarios. To achieve these data rates over interplanetary distances, new communication technologies will likely be employed.

Optical communications is a promising technology that is expected to provide the requisite increase in bandwidth. Compared to traditional radio frequency (RF) systems, optical communication systems offer significantly greater capacity due to the higher optical frequencies and the narrow beam divergence of lasers [59]. This is achieved while being smaller, lighter, more power efficient, and less susceptible to interference [57]. In addition to the high cost of the transceivers, optical communication may, however, be restricted by cloud obstruction, although this can be avoided by placing the transceivers in orbit. Williams [60] shows that data rates of 1 Gb/s can be achieved from Martian distances. In time, long-haul links between Earth and Mars are expected to be primarily comprised of optical communications, supplemented by RF communications.

With an increasing number of relay-enabled spacecraft and dedicated communication spacecraft launched into deep space, communication management becomes increasingly challenging. The current method of communication management is not sustainable. For example, within the MRN, links between spacecraft are manually scheduled [5]. Surface spacecraft do not automatically or opportunistically use overhead orbiters. Furthermore, there is no cross-communication between orbiters and no data exchange between surface spacecraft except via Earth. The use of the Bundle Protocol would solve these problems, allowing data transfers to occur without advanced planning or human interaction.

The automation enabled by the Bundle Protocol would create a user-agnostic network available for use by any spacecraft that can interact with it [46]. Additionally, data transfer rates in an interplanetary network managed by the Bundle Protocol, and supplemented with a suitable routing protocol, would no longer be limited to the rate of a single link. Instead, the rate would depend on the overall throughput of the network [50].

Once the MRN is supplemented by additional infrastructure and relay spacecraft are placed throughout the inner Solar System, the richer connectivity would result in multiple disparate end-to-end paths between source and destination [5]. With the Bundle Protocol operating in such a network, high-capacity, redundant communication becomes possible. Selecting the optimal path in a network is a challenging problem and the performance is dependant on the employed routing protocol.

### 2.5.3 Importance of Timely Communication

Compared to the robotic spacecraft currently on Mars, humans have vastly different communication requirements. To enable crewed missions to Mars, interplanetary communication must be improved. Unlike robots, human lives are important. As a result, astronauts' physical and mental well-being must be monitored at all times [57]. A psychological support network from Earth must be available during these long and isolating missions. Engineers on Earth must also be continuously available to provide technical assistance.

Simulations such as MARS-500 reveal the importance of timely and continuous communication [61]. In MARS-500, a crew of 6 spent 520 days in isolation to simulate a mission to Mars. A communication delay was implemented and the crew's psychophysiological health was monitored. It was found that communication with family, friends, and the ability to receive news about what was happening in their country was paramount to the participants' mental health [62]. Timely responses from "Earth" were revealed to be important in preventing depressive disorders. As communication delays increased, the importance of maintaining contact with family, friends, and psychological support groups increased.

## 2.6 CHAPTER SUMMARY

The literature study and theoretical background presented in this chapter lay the foundation for the dissertation. Examination of literature was crucial to understand the communication framework and protocols that exist to support communication in intermittently connected networks, particularly those intended for space environments. Additionally, a review of literature was conducted to gain insights into the details of planned human missions to Mars. This exploration aids in the construction of simulation scenarios, a topic further discussed in Chapter 3.

# CHAPTER 3 RESEARCH METHODOLOGY

## 3.1 CHAPTER OVERVIEW

This dissertation employs the method of computer simulations to model realistic scenarios of space missions. Communication experiments are performed in these simulations to test the performance of routing algorithms. This process involves the construction of scenarios using parameters informed by literature and the creation of a simulation platform to implement the scenarios and perform investigations.

The following steps encapsulate the research methodology:

1. **Scenario construction**: The first step is to construct a realistic scenario of a space mission. This involves identifying the key elements of the mission, such as the size of the mission and the position of spacecraft. The construction of the primary scenario is revealed in Section 3.2.
2. **Simulation platform**: The next step is to create a simulation platform to implement the scenario. This platform models the physics of the space environment as well as the communication network. Section 3.3 outlines the creation of the simulation platform.
3. **Analysis**: The final step is to analyse the data generated by the simulations to determine the most suitable routing algorithm for the given scenario. The analysis also assists with identifying the strengths of each routing protocol and potential avenues for improvement. Section 3.4 presents the metrics used for analysis.

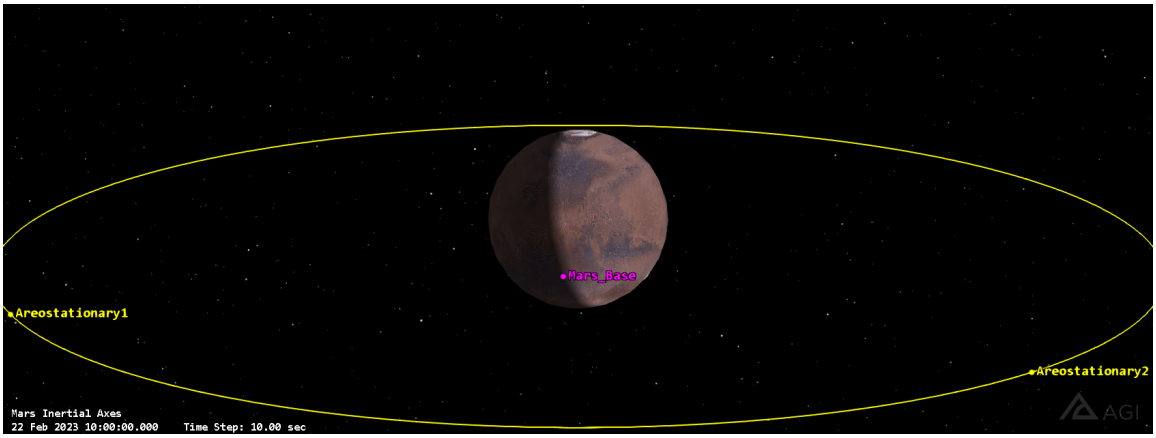
## 3.2 SCENARIO CONSTRUCTION

The primary scenario used in this dissertation is representative of an early crewed mission to Mars. As such, it consists of a single Mars base that will accommodate 4 to 6 astronauts executing a 500-day mission on the surface between two consecutive launch windows. The scenario simulations, however, will model a small interval of the overall mission.

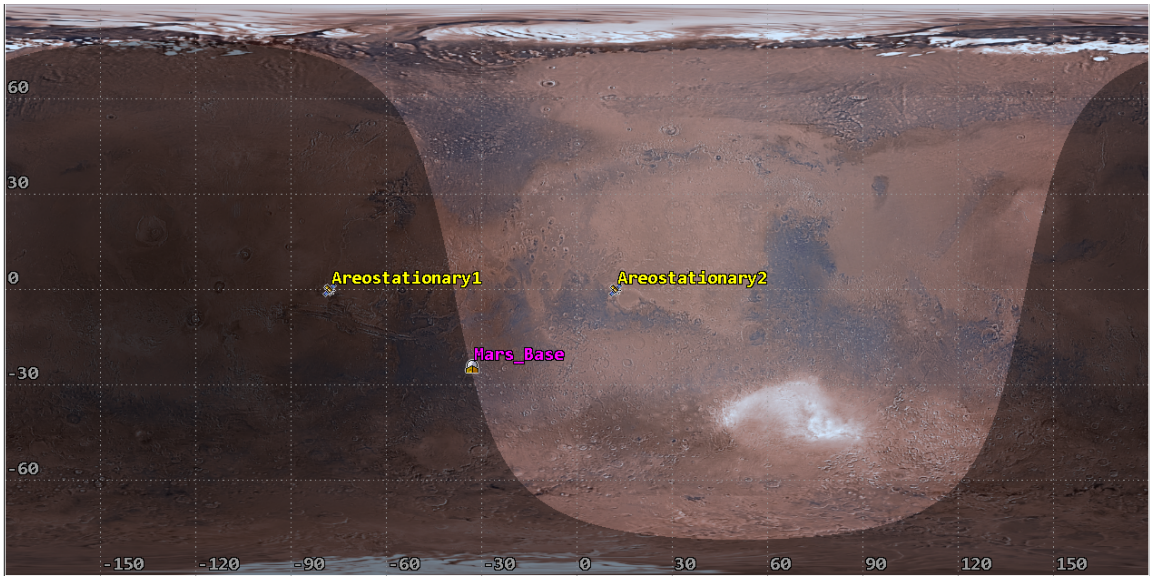
### 3.2.1 Mars base and areostationary relays

The placement of the Mars base is made with consideration of scientifically significant locations. Eberswald Crater, for example, is located among an ancient lake and river delta making it a strong candidate for early scientific and exploratory missions. It was voted close second in the selection of a landing location for the Curiosity rover [63].

The base location has a significant impact on communications. For example, only mid-latitude bases can be effectively supported by areostationary orbiters, which orbit above the equator. Eberswald




**(a)** 3D visualisation of the areostationary relay positions relative to the Mars base.




**(b)** 2D map showing the Mars base at Eberswald Crater relative to the points directly beneath the areostationary relays.

**Figure 3.1.** Locations of the Mars base and areostationary relays, visualised with STK.

Crater is conveniently located near the equator and is thus selected at the location of the scenario's Mars base.

An areostationary orbiter would maintain permanent contact with the Mars base, relaying communications to Earth even when the base is hidden from Earth's view. Two areostationary orbits are required to ensure that at least one relay maintains line of sight with Earth when Mars occludes the other [51]. When neither is occluded, the two relays enable additional capacity and redundancy. With a Mars base location of 24°S 33°W, the areostationary relays are to be placed in equatorial orbits above the longitudes 78°W and 12°E (offset by 45° from the Mars base). At an altitude of 7074.5 km, the orbiters would appear stationary in the sky from the Martian surface - allowing simple, fixed antennae to be used at the Mars base. The location of the base and its areostationary relays is shown in Figure 3.1.

### 3.2.2 Mars Relay Network

In addition to the areostationary orbiters, the MRN will be used to relay communications from the Mars base to Earth. The five orbiters that comprise the MRN are less important to the scenario's mission as they have less bandwidth and do not maintain constant contact with the Mars base. Nevertheless, the MRN is already established and available for use, providing additional communication opportunities at little additional cost. The denser network and additional communication channels created by the inclusion of the MRN also allow for a more robust evaluation of routing algorithms since the pool of available routes is greater.

**Table 3.1.** Characteristics of the MRN orbiters [64].

| | **ODY** | **MRO** | **MVN** | **TGO** |
|---|---|---|---|---|
| **Orbital characteristics** | Sun-synchronous.<br>Altitude: 400 km.<br>Inclination: 93° | Sun-synchronous.<br>Altitude: 375 km.<br>Inclination: 93° | Elliptic.<br>Apogee: 6200 km<br>Perigee: 150 km.<br>Inclination: 75° | Circular.<br>Altitude: 400 km.<br>Inclination: 74° |
| **Uplink data rate (kb/s)** | 512 | 4096 | 4096 | 4096 |
| **Downlink data rate (kb/s)** | 64 | 256 | 256 | 256 |

Table 3.1 summarises the characteristics of the MRN orbiters. These characteristics will be used within the simulation platform.

The orbits of Mars Reconnaissance Orbiter (MRO), Mars Odyssey (ODY) and Trace Gas Orbiter (TGO) are relatively close to the Martian surface, thus communication opportunities with these orbiters last less than 15 minutes per flyover. The elliptical orbits of Mars Atmosphere and Volatile Evolution (MVN) and Mars Express (MEX) often result in much longer contact periods, at the expense of greater ranges [49]. Note that MEX is not included in the scenario as it is a backup relay.

### 3.2.3 Deep Space Network

The MRN communicates with the DSN on Earth. There are three DSN facilities located in Goldstone (United States), Madrid (Spain), and Canberra (Australia). These locations ensure that any distant spacecraft is always in view of at least one facility, as seen in Figure 3.2.

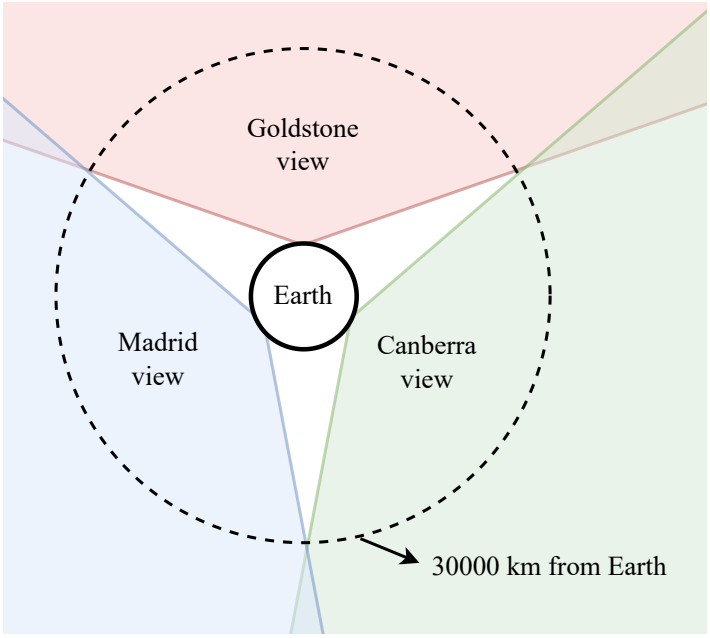


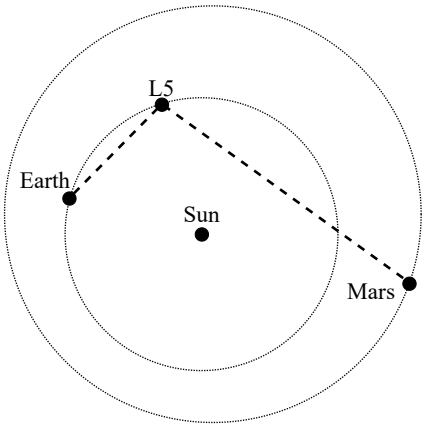

**Figure 3.2.** Field of view of the Deep Space Network antennas, looking down on Earth from above the North Pole. Adapted from [65] "SimonOrJ", CC BY-SA 3.0.

Each facility consists of at least three 34-meter beam waveguide antennas and one 70-meter antenna. The antennae can be arrayed to improve performance. Additionally, the DSN antennae can receive multiple signals from different spacecraft simultaneously. This is known as Multiple Spacecraft Per Aperture and is an important feature of the DSN in the context of the current scenario since multiple Martian spacecraft will be in the DSN's field of view at any given moment [66]. Until more advanced communication infrastructure (such as optical) is established on Earth, it can be assumed that the full

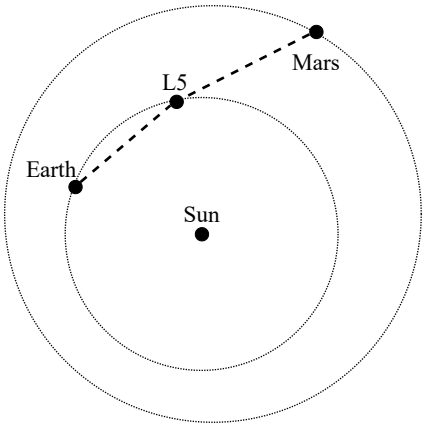
capabilities of the DSN will be deployed to support early crewed missions to Mars, ensuring maximum communication uptime and bandwidth. In this scenario, the DSN will be the Earth's gateway to the interplanetary network, connecting the mission operations centre (MOC) to the Martian spacecraft and the Lagrangian relay.

### 3.2.4 Lagrangian relay

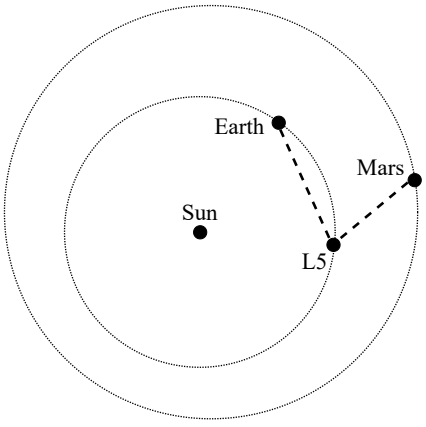
A communication relay in an Earth-Sun Lagrange point will be employed to maintain communication between Earth and Mars during SSCs. As previously discussed, the Earth-Sun L5 point is the most appealing. Figure 3.3(a) reveals how the relay creates a route to circumvent the Sun during SSCs.




**(a)** L5 relay providing a route around the Sun during a SSC.




**(b)** L5 relay acting as a booster between Earth and Mars.




**(c)** Position of Earth, Mars, and the L5 relay in the primary scenario.

**Figure 3.3.** Various Earth-L5-Mars configurations that occur throughout the orbital progression of Earth and Mars.

When there is no conjunction, the relay simply provides an alternative route between the planets. This route is often less desirable than a direct route between Earth and Mars, and thus only provides benefits when the direct routes are congested or unavailable. At certain points in the orbits of Earth and Mars, however, L5 is located between the planets as seen in Figure 3.3(b). This configuration briefly enables a superior communication channel as the relay is able to boost signals with no additional propagation delay compared to a direct route.

In the primary scenario, a time period is selected where there is no SSC and the L5 relay does not serve as a signal booster. This configuration is representative of the most common manner in which the L5 relay functions in the network. The scenario begins on 23 February 2023, when the planets are in the configuration shown in Figure 3.3(c). On this date, Mars is 162 million km from Earth, which results in a propagation delay of approximately 540 seconds.

### 3.2.5 Network topology

A summary of the scenario's topology is given in Figure 3.4. Notice that all Martian nodes can communicate with both the DSN and the L5 relay. This allows the L5 relay to take over from the DSN during SSCs. The four MRN orbiters and three DSN facilities are simplified down to a single node in this representation.

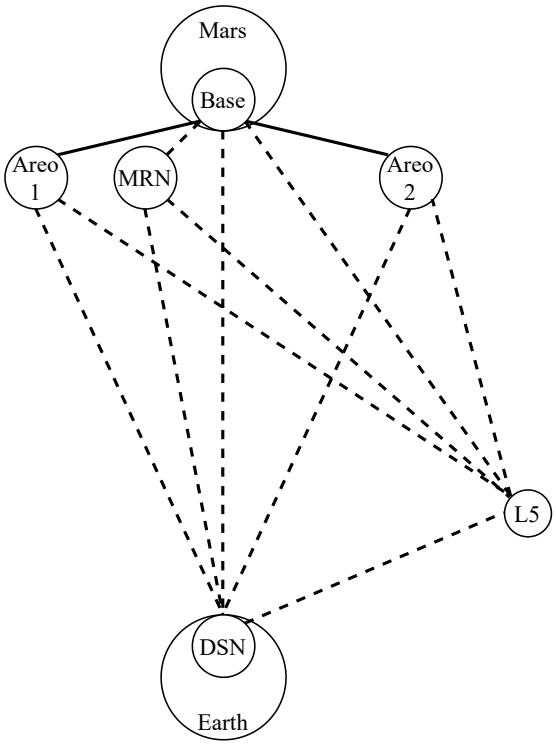



**Figure 3.4.** Simplified scenario topology. Dashed lines represent intermittent contacts. Solid lines represent permanent contacts. MRN and DSN nodes are represented by a single symbol.

**Table 3.2.** Primary scenario connectivity rules.

| **To →**<br>**From ↓** | **DSN** | **L5 relay** | **Areostat-ionary** | **MRN** | **Mars base** |
|---|---|---|---|---|---|
| **DSN** | | Uplink | Uplink | Uplink if better than Mars base | Uplink if better than MRN |
| **L5** | Downlink | | Uplink | Uplink if better than Mars base | Uplink if better than MRN |
| **Areostat-ionary** | Downlink | Downlink | | No contact | Uplink |
| **MRN** | Downlink if better than L5 | Downlink if better than DSN | No contact | | Uplink |
| **Mars base** | Downlink if better than L5 | Downlink if better than DSN | Downlink | Downlink | |

While Figure 3.4 shows all the possible contacts between nodes, not all contacts are available at any given moment. This is partially due to the dynamic positions of nodes and changing lines of sight, but also due to the limitations of the communication equipment installed on certain spacecraft. For example, the orbiters of the MRN are not designed to communicate with multiple nodes simultaneously. They only have a single high-gain antenna which is highly directional. As a result, the orbiters can only communicate with a single distant target at one time, meaning that simultaneous communication with the DSN and the L5 relay is not possible, even if a line of sight exists with both.

As such, communication rules must be implemented to indicate which specific contacts are permitted when multiple are available. Certain nodes, particularly future spacecraft, can support simultaneous contacts. Recall that the DSN can receive simultaneous communication from multiple spacecraft. The communication rules implemented in the primary scenario are presented in Table 3.2.

## 3.3 SIMULATION PLATFORM

There are two primary components of the simulation platform. First, a physics simulation is required to construct a contact plan. With a contact plan, a networking simulation can then be performed. The overall simulation platform is summarised in Figure 3.5.

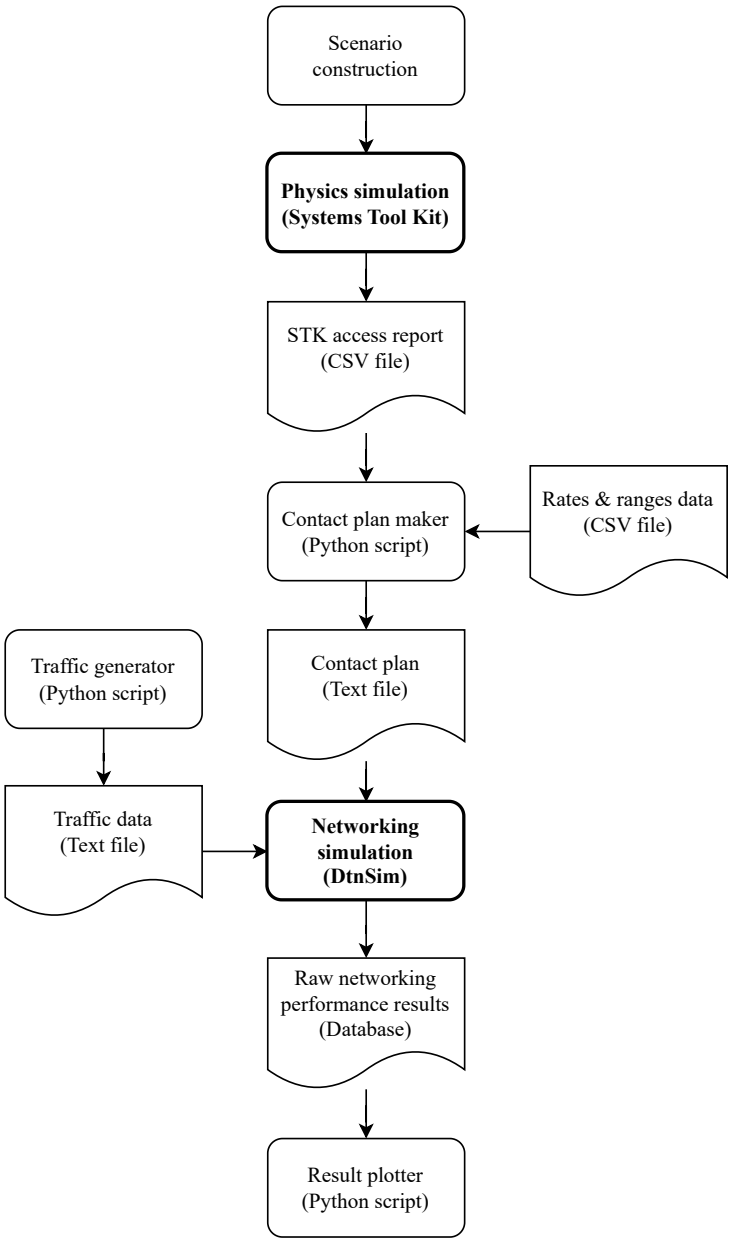



**Figure 3.5.** Summary of simulation platform.

### 3.3.1 Physics simulation

Unlike traditional network topologies on Earth, each node in an interplanetary network is highly dynamic. Planets rotate and satellites orbit, resulting in an ever-changing topology. Fortunately, the movement of each node can be computed using a physics simulation, allowing for the accurate characterisation of the changing topology. Simulation software models the movement of planets, ground nodes, and spacecraft. Link analysis can be performed to understand precisely when each node has a line of sight with other nodes, as well as the exact distance between nodes. With this information, a contact plan can be created.

Systems Tool Kit (STK), developed by Analytical Graphics, Inc. is selected for use as the physical simulation platform due to its graphical method of scenario configuration and use in literature [13] [67]. A free licence of STK was granted, allowing limited use of STK's features. The available features, however, are sufficient for the simulation of the primary scenario.

The configuration of the physics simulations begins with the placement of nodes. Many existing nodes such as the antennae in the DSN are preconfigured and can simply be imported into the scenario. The Mars base is placed at the desired coordinates within the Eberswald Crater. The MRN orbiters, areostationary relays, and the L5 relay are each configured manually by specifying the parameters of their orbits, illustrated in Figure 3.6.

After each node has been configured, link analysis can be performed using STK's Access tool. This tool allows the line of sight between each node to be computed, depicted in Figure 3.7. STK simulates the movement of each node over a given period, producing an access report that indicates the start time and stop time of each available link. An extract of an access report is shown in Table 3.3.

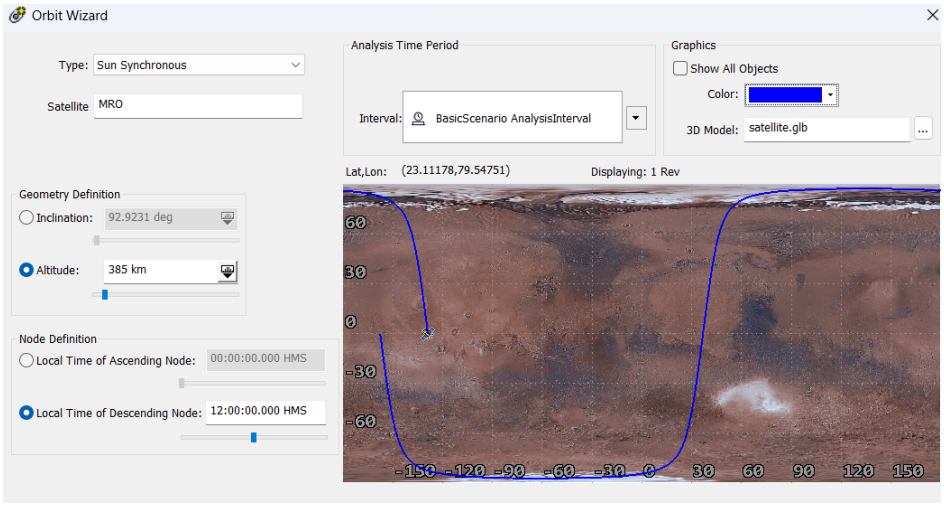



**Figure 3.6.** Configuration of the MRO in STK.

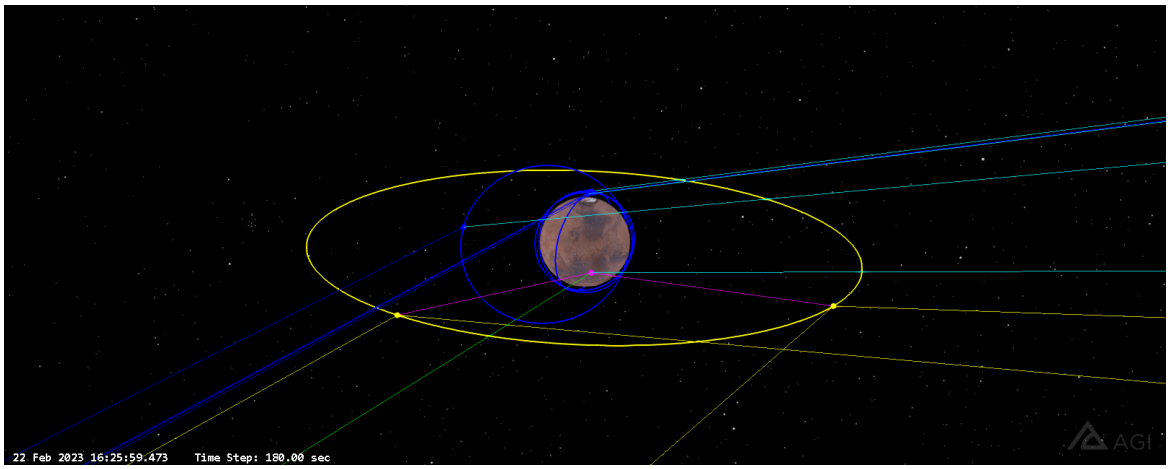



**Figure 3.7.** STK visualisation showing the available links between Martian nodes, DSN stations, and the L5 relay. The links that exit at the bottom of the frame extend to Earth while the links that exit to the right extend to the L5 relay.

**Table 3.3.** Extract of an STK access report showing intervals of downlink availability from the Mars base to the Canberra DSN facility.

| Start Time (UTC) | Stop Time (UTC) | From Object | To Object |
|---|---|---|---|
| 22 Feb 2023 10:00:00 | 22 Feb 2023 13:28:52 | Mars Base | DSN Canberra |
| 23 Feb 2023 07:51:16 | 23 Feb 2023 13:26:23 | Mars Base | DSN Canberra |
| 24 Feb 2023 08:30:22 | 24 Feb 2023 13:23:55 | Mars Base | DSN Canberra |
| 25 Feb 2023 09:09:29 | 25 Feb 2023 10:00:00 | Mars Base | DSN Canberra |

The access report is combined with data rate information and propagation delay information (also obtained from STK) to form a contact plan for the scenario.

Entries in a contact plan take the form:

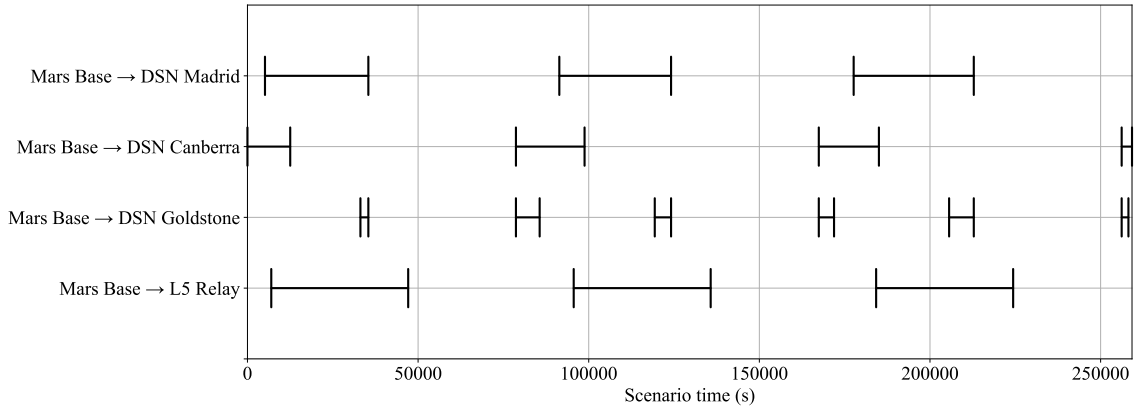
```
a contact <start time (s)> <end time (s)> <source node ID>
<destination node ID> <data rate (MB/s)>

a range <start time (s)> <end time (s)> <source node ID>
<destination node ID> <propagation delay (s)>
```

A Python script was created to perform this procedure, producing a contact plan as illustrated below:

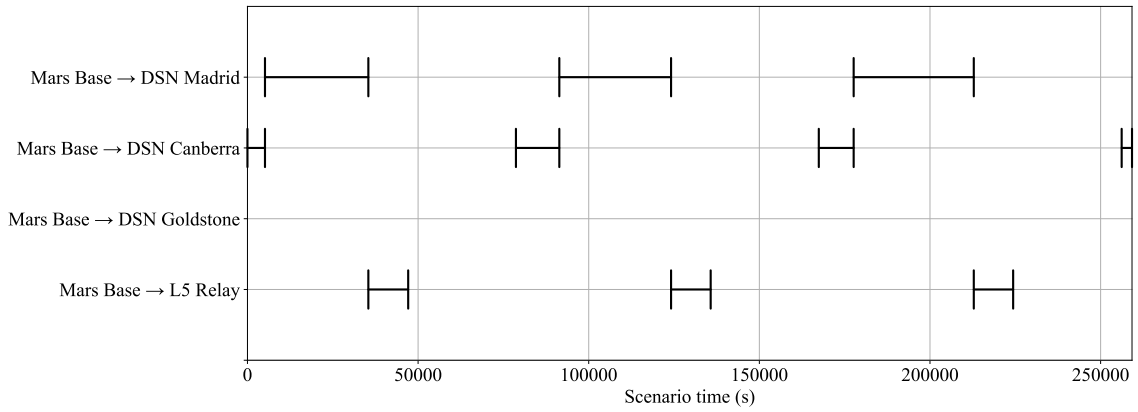
```
a contact +0 +5442 5 2 6.25
a contact +73988 +91688 5 2 6.25
a contact +162731 +177936 5 2 6.25
a contact +251475 +259200 5 2 6.25
a range +0 +259200 5 2 545
```

An important part of this script is rule enforcement. As briefly discussed, nodes cannot necessarily communicate with all available nodes simultaneously. For example, a directional antenna at the Mars base cannot communicate with both the DSN and the L5 relay as it cannot be pointed at both Earth and L5. As such, the connectivity rules of Table 3.2 are enforced, along with more specific rules that describe the priority of nodes within the MRN and DSN.

The result of this rule enforcement can be seen in Figure 3.8, where the downlink rules between the Mars base, L5 relay, and the DSN are enforced. Notice that the Madrid facility has the greatest priority




**(a)** Before rule enforcement.




**(b)** After rule enforcement.

**Figure 3.8.** Downlink availability between the Mars base, L5 relay, and the DSN before and after connectivity rule enforcement.

in this scenario, with the Canberra facility only being used when Madrid is unavailable. The Goldstone facility, with the lowest priority, is not used as either Madrid or Canberra is concurrently available. The L5 relay (with a longer, indirect route) is only used when the Mars base is on the far side of the planet relative to Earth and, as such, no DSN facility is available.

Rule enforcement ensures that the contact plan more accurately reflects the contacts that are realistically available by removing links that theoretically exist, but cannot be utilised in practice. The custom Python script developed to generate contact plans is designed to be generic, allowing the contact plans of new scenarios to be easily computed.

### 3.3.2 Networking simulation

A networking simulation is employed to evaluate the performance of routing protocols. This involves the construction of the network topology, the generation of data at specific nodes, and the analysis of bundle propagation through the network. The simulation needs to use contact plans to construct dynamic network topologies.

Various DTN simulation tools are available, presented in Table 3.4. An overview of each tool is also given below:

1. **DTN2**: An experimental platform where researchers can validate DTN protocol designs. It caters to researchers but DTN2 code is also production-ready and can be used in real-world networks [68].

2. **ION**: The Interplanetary Overlay Network is an implementation of DTN developed by NASA [69]. Although it can be used for advanced simulations, its primary purpose is to run on embedded flight hardware in space.

3. **The ONE simulator**: The Opportunistic Network Environment simulator is specifically designed to evaluate DTN routing and application protocols [70]. It is primarily useful for simulating terrestrial opportunistic networks such as vehicular networks.

4. **DtnSim**: An event-driven simulation, implemented in OMNeT++, that is designed to study various aspects of space-based DTN including routing, forwarding, and scheduling [15]. DtnSim integrates with flight software such as ION and has many preconfigured routing algorithms.

**Table 3.4.** Candidate networking simulation tools.

| Software tool | Released | Developer(s) | Language | Operating system | Primary purpose |
|---|---|---|---|---|---|
| **DTN2** | 2004 | Delay Tolerant Networking Research Group | C++ | Linux | Research and deployment |
| **ION** | 2007 | NASA | C++ | OS independent | Deployment on flight hardware |
| **The ONE simulator** | 2009 | Ari Keränen, Jörg Ott, Teemu Kärkkäinen | Java | OS independent | Research (terrestrial opportunistic DTNs) |
| **DtnSim** | 2017 | Juan Fraire, Pablo Madoery | C++ | Linux | Research |

DtnSim is selected as the simulation tool as it is designed for research and is suited specifically for space networks similar to the primary scenario. DtnSim employs the Bundle Protocol to emulate communication within intermittently connected networks. It was developed as an alternative to tools focusing on opportunistic routing and tools that run in real time. The tool is event-driven and can evaluate space communication systems at accelerated speeds [13]. This is important for space systems since simulations often need to span orbital periods, requiring simulation durations of multiple days. The DtnSim source code is freely available and can be easily modified to add new routing protocols or alter existing protocols. This is crucial in the evaluation of routing protocol enhancements.

DtnSim employs a layered architecture, depicted in Figure 3.9, where each node has an application (APP), DTN, and communication protocol (COM) module [15]. The APP module handles data generation and consumption while the DTN module focuses on providing delay-tolerant multihop transmission, incorporating a local storage unit in each node for in-transit bundles. Routing algorithms operate within the DTN module, deciding when and where to forward bundles. The DTN module also models the intermittent nature of space networks using a network's contact plan. The COM module emulates wireless links, transferring bundles between nodes. This module is also where bundles experience the effects of data rate constraints, propagation delays, and contention.

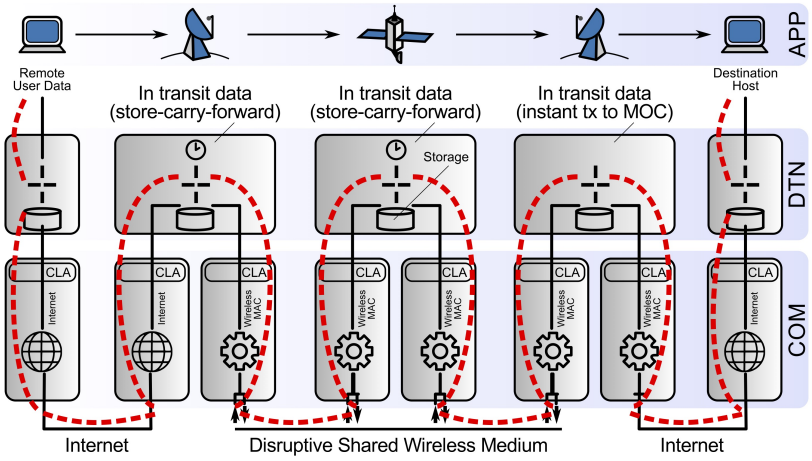
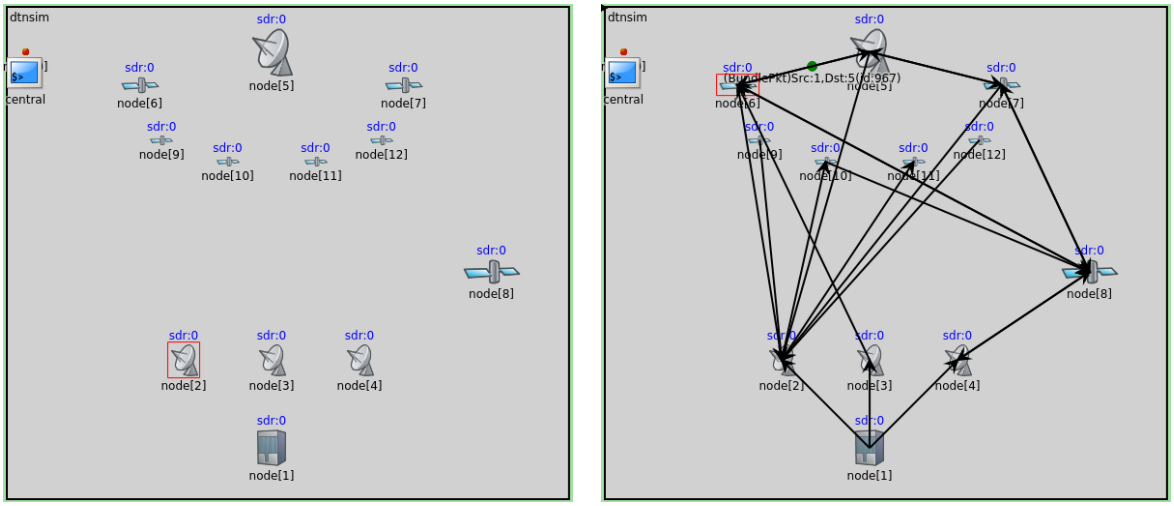



**Figure 3.9.** DtnSim node architecture. Taken from [15] © 2017 IEEE.

The first step in the setup of a DtnSim simulation is the placement of nodes, followed by the configuration of the actual topology using a contact plan. The topology of the primary scenario is shown in Figure 3.10. The placement of nodes is purely visual and does not impact the simulation; the contact plan informs the relationship between nodes. The setup of DtnSim simulations is performed in an OMNeT++ configuration file (.ini) which specifies the contact plan file, network traffic file, and the desired routing algorithm.




**(a)** Placement of nodes. **(b)** Links between nodes.

**Figure 3.10.** Configuration of the primary scenario's network topology in DtnSim. Node 1 is the MOC, nodes 2 to 4 are the DSN facilities, node 5 is the Mars base, nodes 6 & 7 are the areostationary relays, node 8 is the L5 relay, and nodes 9 to 12 are the MRN relays. The links between nodes, informed by the scenario's contact plan, are displayed during simulation execution.

In the primary scenario, data is generated at both the MOC (node 1) and the Mars base (node 5). The frequency, size, and destination of bundles are configurable. In each investigation, a particular size and frequency of bundles is generated to place a particular load on the network. A series of simulations can be performed with each subsequent simulation generating bundles of greater size and frequency, allowing the network performance to be analysed at a range of network loads.

A custom Python script is developed to automate the creation of simulation configuration. The number of simulations can be specified, along with the desired network load measured in Mb/s. A series of network traffic files are produced, which are referenced by a series of OMNeT++ configuration files generated from a template. An example of a network traffic file and OMNeT++ configuration file is given in Appendix B.

### 3.3.3 Simulation timeline

The primary scenario is configured to begin on 23 February 2023 at 10:00 UTC, with a data generation period of 24 hours. An additional 24 hours is simulated to allow for bundle progression and delivery. A 24-hour generation period ensures that the scenario is tested in a wide range of network conditions. There is little variation in the patterns of contacts from day to day, but over a single day, the network conditions change dramatically as the planets rotate, making it important to simulate the network over a minimum of 24 hours.

In the case of PRoPHET simulations, an additional 72 hours is simulated before bundle generation to allow the algorithm to initialise. Recall that PRoPHET makes use of the history of contacts between nodes to inform routing decisions. The timeline of the primary scenario's PRoPHET simulations is given in Figure 3.11.

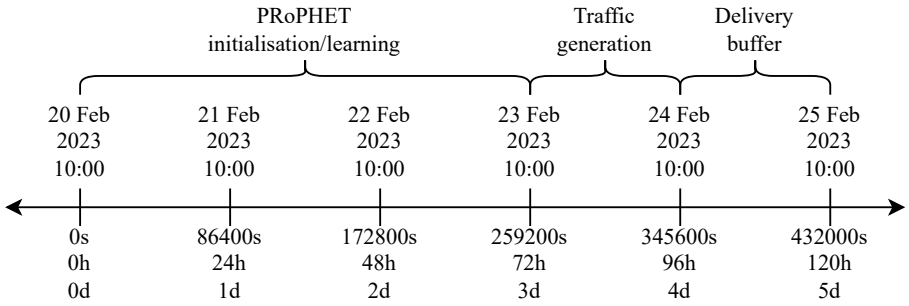



**Figure 3.11.** Timeline of the primary scenario's simulations in the specific case of PRoPHET simulations. When other routing algorithms are simulated, the simulations begin at 10:00 UTC on 23 February 2023, without the initial 72-hour period.

## 3.4 ANALYSIS

### 3.4.1 Metrics

DtnSim records many parameters during simulation execution that can be studied to examine network performance. The following metrics are derived from the recorded parameters and are used to evaluate routing algorithms:

- **Bundle latency**: This crucial metric represents the time it takes for a bundle to traverse the network from source to destination. In DTNs, successful routing algorithms are those that minimise latency despite congested conditions and networks with multiple dynamically changing routes. A sub-optimal route selection within a deep-space network can significantly increase latency as two candidate routes may differ in length by several light seconds or minutes. Furthermore, poor routing decisions may cause a bundle to be transferred to an unconnected node, delaying it as it awaits an onward contact.
- **Delivery ratio**: Analogous to packet loss, the delivery ratio can be described as the ratio of bundles that are received at the destination node to bundles that were originally transmitted at the source. Successful routing algorithms are those that maintain high delivery ratios despite congested and disrupted networks. Undelivered bundles hinder overall performance since these bundles need to be retransmitted.
- **Aggregate network load**: This metric is simply a count of the number of node-to-node bundle transmissions that take place throughout a simulation. It is an elegant indicator of the efficiency of a routing algorithm. Efficient algorithms have high delivery ratios with a low aggregate network load. Inefficient algorithms duplicate bundles, transmitting multiple copies throughout the network, resulting in a multitude of individual transmissions for each delivered bundle. Additional processing and transmission energy is required to distribute these redundant bundles, and network congestion is increased.
- **Throughput**: This metric reveals the data transfer rate achieved by a routing algorithm in a DTN. It encapsulates the efficacy of the algorithm in delivering bundles. Throughput is influenced by route selection and network resource utilisation. Low throughput suggests potential inefficiencies stemming from suboptimal route decisions and network congestion. Evaluating throughput in conjunction with other metrics provides a comprehensive understanding of a routing algorithm's overall performance in challenging DTN environments.

DtnSim results are stored in a SQLite database. Limited analysis can be performed by inspecting the database or by using the plotting tools in OMNeT++. Instead, a custom Python script is developed

to process DtnSim databases, extract relevant metrics, and plot them. None of the desired metrics are available directly in the database therefore the script must combine several database entries. As a simple example, no database entry exists for "delivery ratio." Instead, the script uses the number of bundles generated at the source and the number of bundles received at the destination to determine this ratio. The plotting script is designed to plot the results of multiple simulations simultaneously, allowing the effect of an independent variable (such as data generation rate) to be visualised.

### 3.4.2 Ford-Fulkerson algorithm

To aid in the analysis of routing protocols in a given network, it is useful to know the maximum throughput of the network between a source node and a destination node. Networks introduce multiple paths between source and destination, increasing overall throughput due to the presence of multiple parallel routes. The calculation of the maximum throughput in a network is not trivial and is a classic optimisation problem in graph theory. Multiple paths, intermediate nodes, and bottlenecks make it difficult to determine the overall data rate by inspection. Knowing the maximum throughput provides a baseline for performance evaluation, putting the measured throughput into perspective.

The first algorithm to solve the maximum flow problem is the Ford–Fulkerson algorithm (FFA) [71]. Many improvements to the FFA have been proposed, including new algorithms, primarily to reduce its computational complexity. The Edmonds–Karp algorithm [72], a near-direct implementation of the FFA, is used in this research. The algorithm is described in Appendix C.

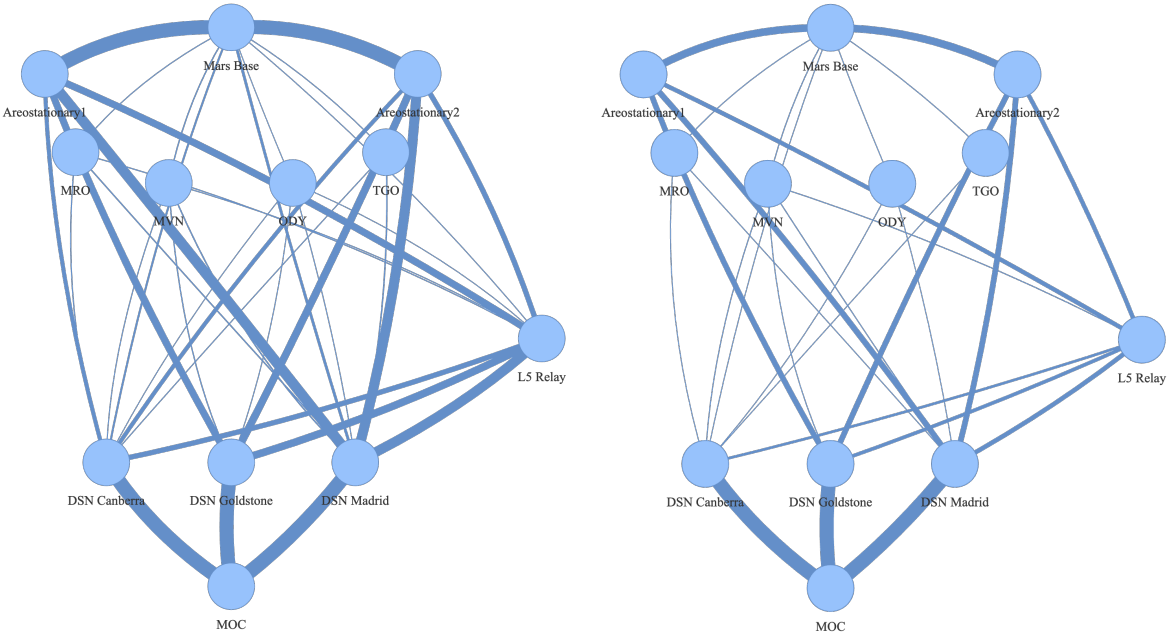
The FFA cannot be applied directly to an intermittently connected network since it operates on a static graph representation of a given network. As such, modifications must be made before applying the FFA. To use the FFA on an intermittently connected network, all possible links between nodes in the network are made available with a time-averaged data rate (effective data rate of the link over time). For example, a link with a data rate of 90 Mb/s that is available only 20% of the time, is given a data rate of 18 Mb/s in the FFA graph.

This network simplification means that the resulting maximum flow is not perfectly accurate and is simply an upper bound of the network's average throughput over the duration of the entire primary scenario. At any point in the scenario, the actual throughput may be higher (when contacts are available and aligned) or lower (when contacts are unavailable or misaligned). The average throughput over the duration of the scenario, however, cannot exceed the calculated maximum throughput.

The timing (alignment) of contacts in a network has a significant impact on bundle progression. In intermittently connected networks, contacts may go unused when no bundles are supplied to these

contacts due to inopportune contact timing. The conversion to a network with 100% contact uptime gives such a network an advantage, inflating the expected average throughput. Despite this, the upper bound is still valuable and can be used in performance analysis as an optimistic benchmark.

The graphs used to compute the primary scenario's maximum throughput are given in Figure 3.12.




**(a)** Downlink. **(b)** Uplink.

**Figure 3.12.** Network graphs of the primary scenario used in the Ford-Fulkerson maximum throughput algorithm. Edge thickness is proportional to the effective data rate of the link over time.

The time-averaged maximum throughput between the Mars base and MOC over the course of the primary scenario was calculated to be 1741 Mb/s in the downlink direction and 501 Mb/s in the uplink direction.

## 3.5 CHAPTER SUMMARY

This chapter details the methodology used in the dissertation, which involves computer simulation to model realistic space mission scenarios and test routing algorithms' performance. The key steps outlined in this chapter include scenario construction, simulation platform creation, and result analysis.

The primary scenario is presented, which centres around a small crewed mission to Mars. Areostationary relays and the Mars Relay Network facilitate communication between the Mars base and Earth. The Deep Space Network and a Lagrangian relay complete the interplanetary network.

The simulation platform combines physics and networking simulations using System Tool Kit and DtnSim, respectively. System Tool Kit models the movement of planets and spacecraft, creating a contact plan. DtnSim emulates communication in a network, allowing routing algorithms to be evaluated using metrics such as bundle latency, delivery ratio, aggregate network load, and throughput. The Ford-Fulkerson algorithm is employed to determine the maximum throughput of the network, providing a benchmark for performance evaluation.

Overall, the chapter outlines a structured methodology for conducting communication simulations in interplanetary networks.

# CHAPTER 4 RESULTS AND DISCUSSION

## 4.1 CHAPTER OVERVIEW

This chapter presents the networking simulation results of routing algorithms tested in various DTN scenarios. The performance metrics of the algorithms are presented and analysed.

Replication-based algorithms are first evaluated in the primary Martian scenario of Section 3.2, followed by the evaluation of CGR algorithms in the same scenario. A new satellite network scenario is introduced to further evaluate the CGR algorithms in a different network configuration.

## 4.2 REPLICATION-BASED ROUTING ALGORITHMS

First, replication-based routing algorithms are compared. These algorithms duplicate data bundles, distributing copies throughout the network. This replication allows for favourable bundle delivery rates, as only one bundle needs to reach the destination. A disadvantage, however, is that network resources are not used efficiently. As such, replication-based algorithms are typically not scalable, with networks becoming congested as traffic increases.

Epidemic routing, PRoPHET, Spray and Wait, and Binary Spray and Wait are compared at increasing data generation rates in the primary scenario. Data is generated at the Mars base and flows in the downlink direction to the MOC on Earth. In the following simulations, the data is generated in bursts spaced evenly throughout the 24-hour data-generation period.

Multiple simulation iterations are performed with increasing data generation rates, to place increasing demands on the network. In this first set of simulations to investigate replication-based algorithms, data is generated at rates from 25 Mb/s to 1500 Mb/s. Network performance metrics are plotted as a function of the data generation rate.

### 4.2.1 Unlimited node storage

First, the performance of the algorithms is investigated in a configuration where the storage capacity of each node's spacecraft data recorder (SDR) is unlimited. Nodes are thus able to store all received

bundles indefinitely while awaiting an onward contact. This configuration benefits replication-based algorithms as the duplication of bundles does not saturate the storage capacity of nodes. Network congestion can therefore only occur due to saturation of communication links.

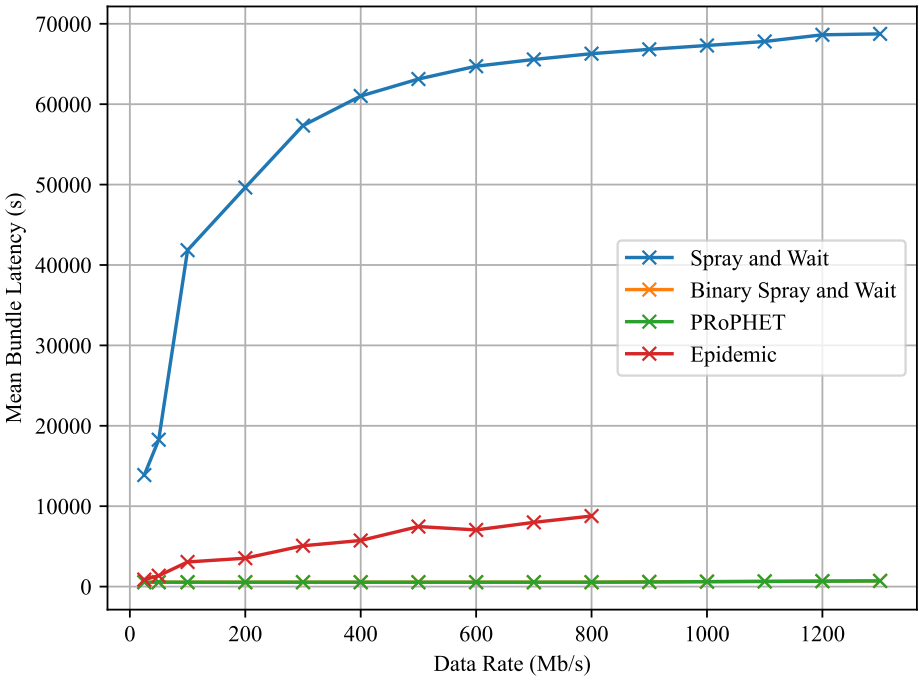
Figure 4.1 shows the mean bundle latency of the algorithms as the demands on the network are increased. It is immediately evident that Epidemic routing and Spray and Wait perform poorly, with high latencies even at low data generation rates. Figure 4.1(b) reveals that PRoPHET and Binary Spray and Wait maintain a latency of 545 s, in line with the propagation delay between the planets in the primary scenario. The performance of these two algorithms begins to degrade beyond the data generation rate of 800 Mb/s, with delivery latency increasing proportionally to the data generation rate.

The poor performance of epidemic routing can be attributed to its flooding-based nature. It quickly saturates the transmission capacity of contacts. This is supported by Figure 4.3, where it is revealed that epidemic routing makes the most transmissions in the network for each delivered bundle relative to the other algorithms. The quantity of duplicated bundles also caused lengthy runtime durations of epidemic routing simulations therefore the algorithm was not simulated in this configuration beyond 800 Mb/s.

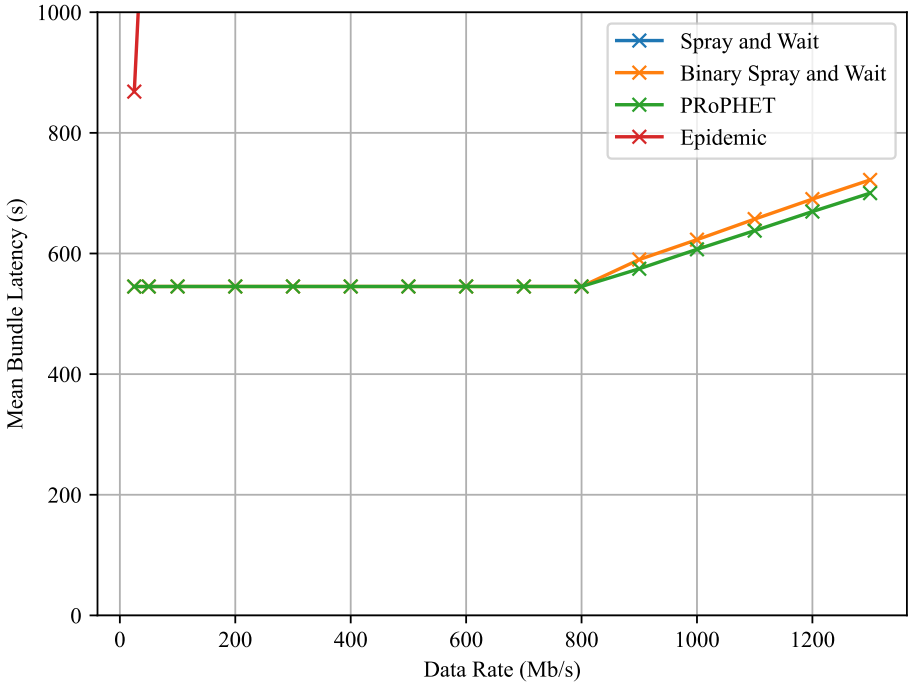
Figure 4.2, showing the ratio of delivered bundles to sent bundles, reveals that all but one algorithm maintains a 100% delivery ratio. Spray and Wait performs poorly as the algorithm is inherently inadequately suited to the network topology of the primary scenario. Spray and Wait only distributes bundles to the source's immediate neighbours, with the bundles progressing no deeper into the network, unless these neighbours have direct contact with the destination. This limits the possible paths to the MOC significantly. Bundles are only delivered during narrow windows of opportunities when the Mars Base is in contact with the DSN. The limited capacity of this path means that not all bundles can be delivered, and those that are delivered are severely delayed.

Binary Spray and Wait significantly improves upon Spray and Wait by making a simple modification (described in Section 2.4.2) that allows bundles to progress deeper into networks. This improves performance in the context of the primary scenario as all paths between source and destination are utilised.

PRoPHET and Binary Spray and Wait perform perfectly up to 800 Mb/s, with 100% delivery ratios and minimum latency. Figures 4.3 and 4.4 provide additional metrics to gain insight into the performance of algorithms.




**(a)** Full perspective.




**(b)** Narrow perspective.

**Figure 4.1.** Mean bundle latency of replication-based algorithms in the primary scenario with unlimited SDR capacity.

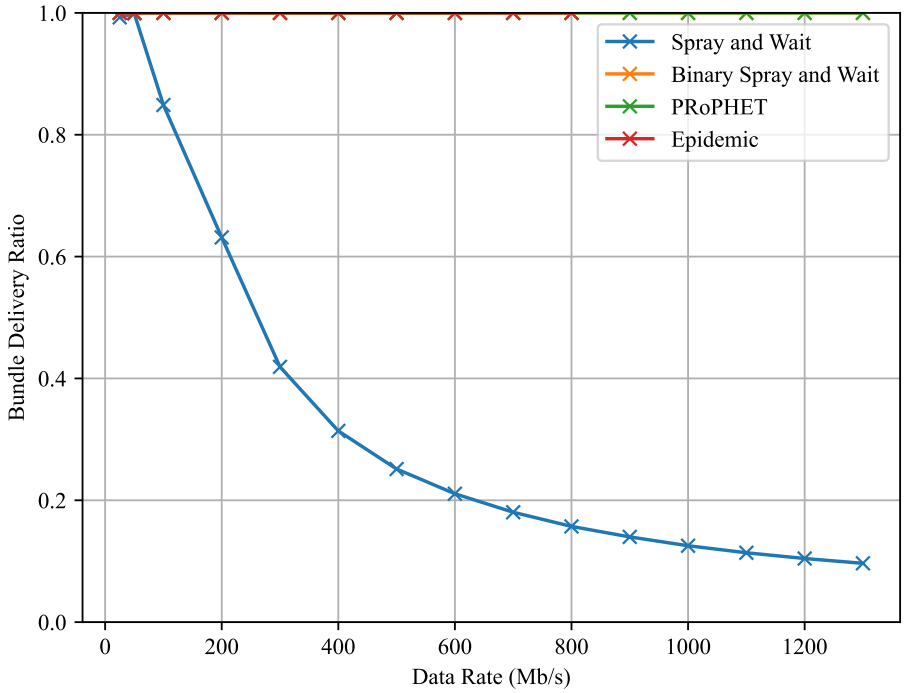



**Figure 4.2.** Bundle delivery ratio of replication-based algorithms in the primary scenario with unlimited SDR capacity.

Figure 4.3 shows the efficiency of the algorithms by revealing the number of bundles transmitted between nodes in the network for each bundle that reaches the destination. It is evident that the two best-performing algorithms, PRoPHET and Binary Spray and Wait, place the lowest demands on the network's capacity. These algorithms transfer approximately ten bundles between nodes in the network for every delivered bundle. Epidemic routing, as expected, transmits the greatest number of bundles, resulting in network congestion. Notice that at lower data generation rates, epidemic routing is able to transmit and flood the network with more bundles than it can at higher data generation rates.

Figure 4.4 shows the time-averaged quantity of data stored throughout the entire network in node SDRs. In this configuration with no SDR constraints, the data stored in the network increases proportionally to the data generation rate. Notice that PRoPHET is the most storage-efficient algorithm in this investigation, with epidemic routing consuming the most storage resources, as expected. Without a limit on storage capacity, the quantity of data stored in the network increases without bound. This is not practical or scalable in reality. In the next investigation, the same routing algorithms are compared but in a network configuration with a fixed SDR capacity.

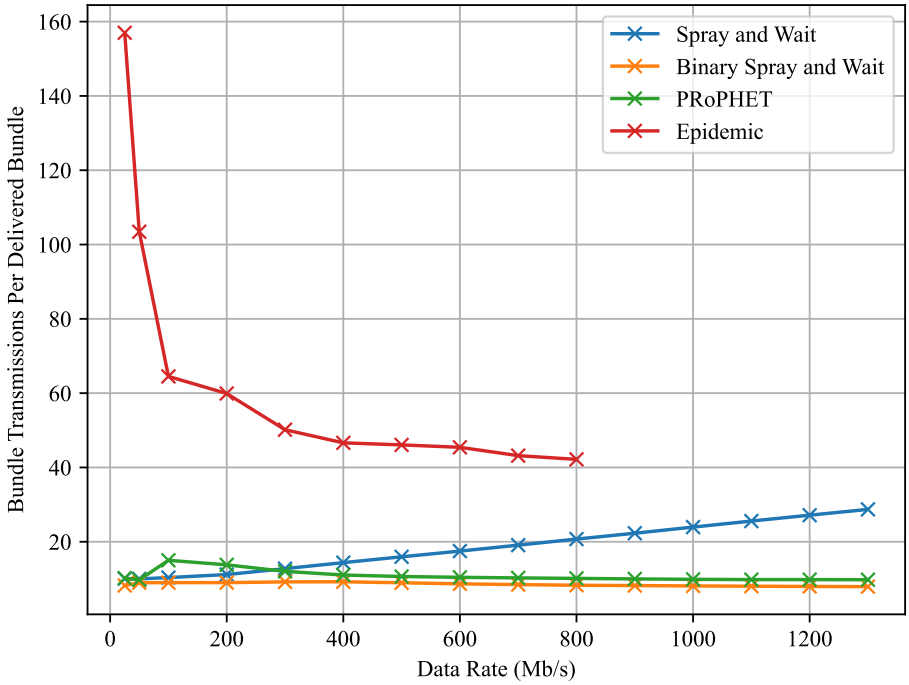



**Figure 4.3.** Bundle transmission count per delivered bundle of replication-based algorithms in the primary scenario with unlimited SDR capacity.

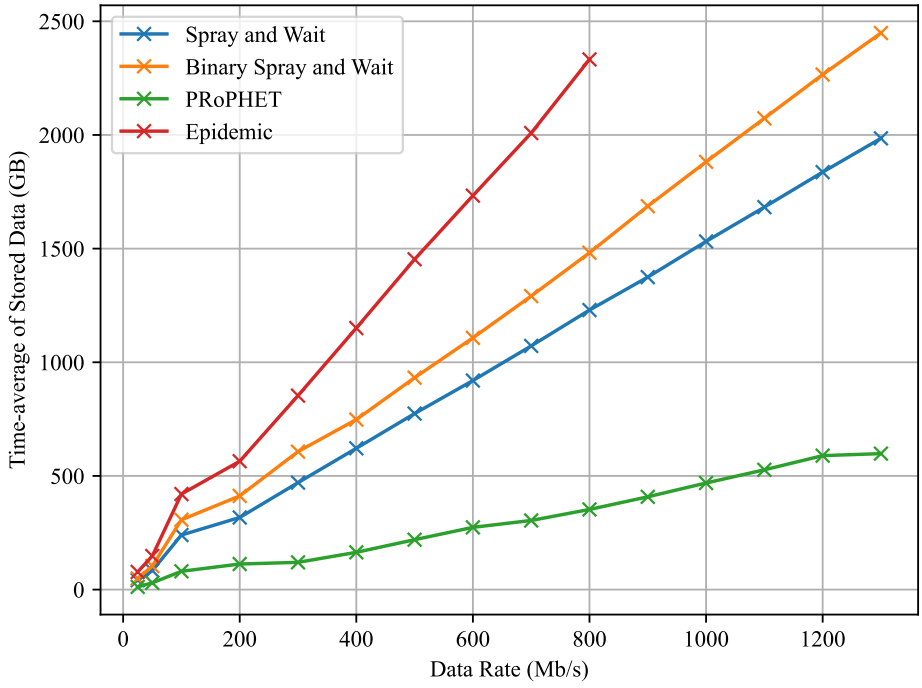



**Figure 4.4.** Time-averaged quantity of data stored in the network by replication-based algorithms in the primary scenario with unlimited SDR capacity.

### 4.2.2 Limited node storage

The replication-based routing algorithms are now compared in a more realistic scenario where the nodes have limited storage capacities. The configuration is identical to the previous scenario configuration, except for the SDR capacities. The performance is expected to decrease significantly as the large number of duplicated bundles produced by the algorithms will quickly saturate the available storage at each node.

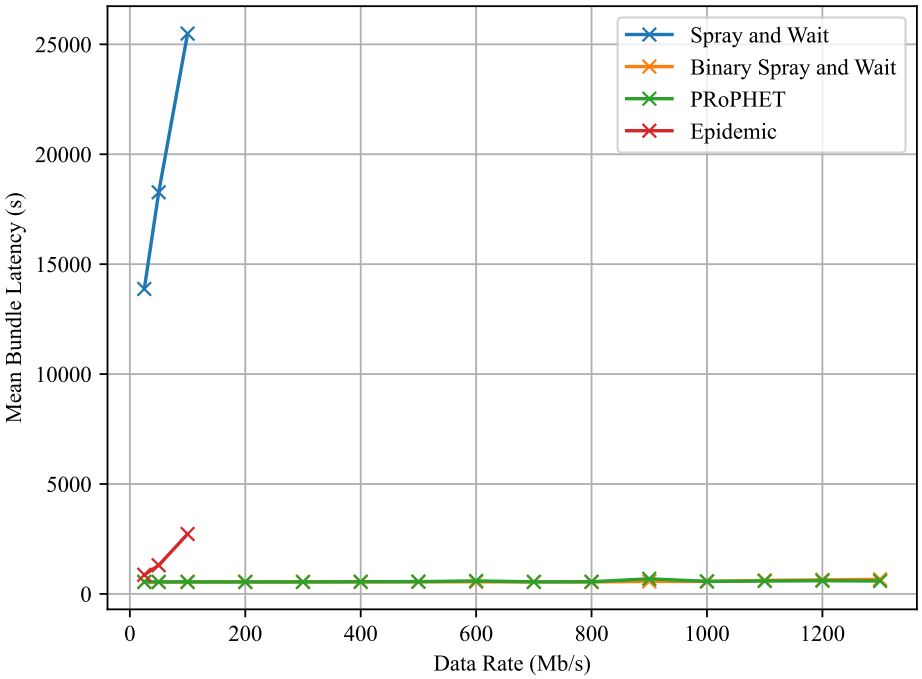
An SDR capacity of 60 GB is selected for this investigation. Each node can therefore store 60 GB of bundles before bundles are deleted. Recall that the method of data generation in this preliminary investigation involves the generation of data in bursts spaced throughout the 24-hour generation period. Data is generated for 1.4 hours of the 24-hour generation period. The 60 GB constraint would correspond to a 1 TB constraint in a network with continuous data generation. When an SDR reaches capacity, it is cleared to make space for newer bundles.

Figure 4.5 shows the mean bundle latency of the algorithms as the demands on the network are increased. Again, it is immediately evident that Epidemic routing and Spray and Wait perform poorly, with high latencies at low data generation rates.

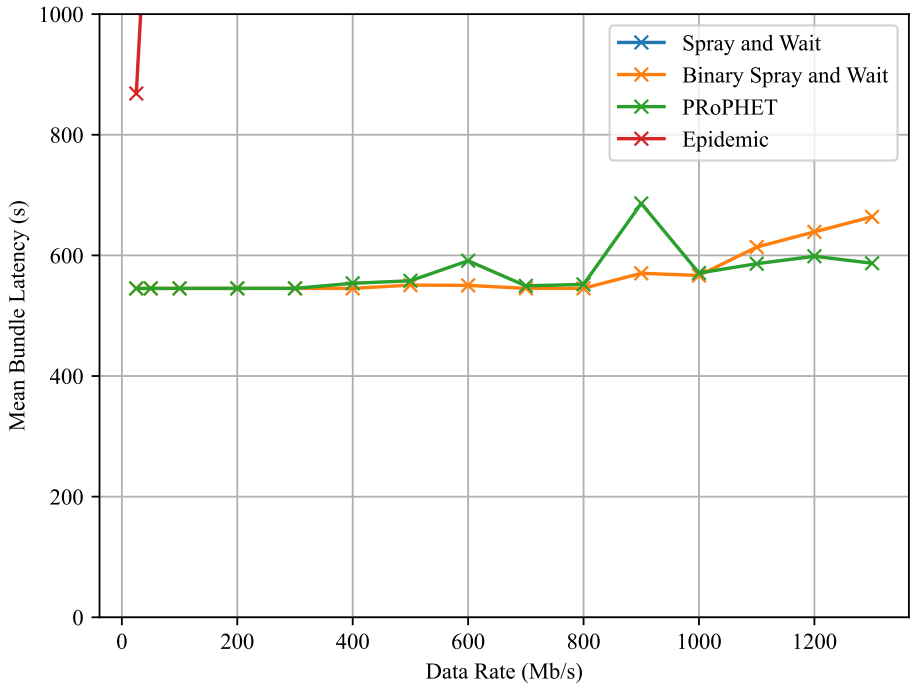
Note that performance metrics are only plotted at data generation rates where an algorithm maintains a high delivery ratio. As the delivery ratio decreases, other metrics such as latency become less monotonic due to inconsistencies caused by undelivered and deleted bundles. For example, PRoPHET in Figure 4.5(b) exhibits an unusual spike at 850 Mb/s due to bundles that are delivered (with delay) at 850 Mb/s but are not delivered at higher data generation rates.

Figure 4.5(b) reveals that PRoPHET and Binary Spray and Wait maintain a latency of 545 s, in line with the propagation delay between the planets in the primary scenario. The performance of these two algorithms begins to significantly degrade beyond the data generation rate of 800 Mb/s. At this data generation rate, the networks can no longer support the volume of bundles produced by the algorithms, resulting in undelivered bundles (see Figure 4.6) and increased latency for bundles that reach the destination.

The poor performance of Spray and Wait is again due to its shallow penetration into the network. Unlike the unlimited node storage scenario, Epidemic routing's delivery ratio quickly decreases as the network load is increased. Epidemic routing quickly fills SDRs, resulting in bundle deletions and many undelivered bundles. Figure 4.8 shows that Epidemic routing has the greatest SDR usage at low data rates.




**(a)** Full perspective.




**(b)** Narrow perspective.

**Figure 4.5.** Mean bundle latency of replication-based algorithms in the primary scenario with limited SDR capacity.

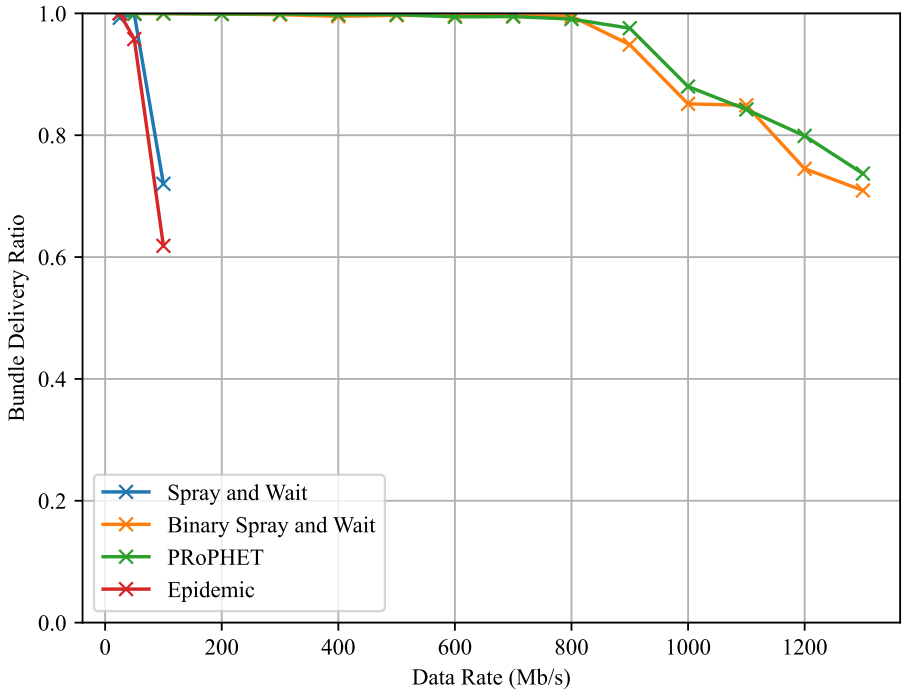



**Figure 4.6.** Bundle delivery ratio of replication-based algorithms in the primary scenario with limited SDR capacity.

Figure 4.6 confirms that PRoPHET and Binary Spray and Wait are the best-performing replication-based routing algorithms. The two algorithms perform similarly, maintaining a delivery ratio of close to 100% until 800 Mb/s, after which the delivery ratios significantly degrade. As noticed in Section 4.2.1, network congestion begins to impact PRoPHET and Binary Spray and Wait at 800 Mb/s. SDR storage constraints have limited detrimental effects on the latency and delivery ratio below 800 Mb/s in this scenario.

The load that each routing algorithm places on the network, shown in Figure 4.7, remains largely unchanged with the introduction of storage constraints. Notice the large difference in transmissions between Epidemic routing and the other algorithms. It is evident that a key to low latency and guaranteed delivery is the frugal use of network resources.

With the node storage constraints, the total quantity of data that can be stored in the network is limited. This is evident in Figure 4.8, where unlike Figure 4.4, there is no universal proportional relationship between data generation rate and quantity of stored data. Instead, the quantity of stored data is approximately proportional to the data rate at low generation rates, but then the average drops off at higher data rates where SDRs are regularly filled to capacity and cleared. Such bundle deletions cause detrimental effects to overall performance, particularly at higher data generation rates where it is

more likely that all of a particular bundle's copies are deleted.

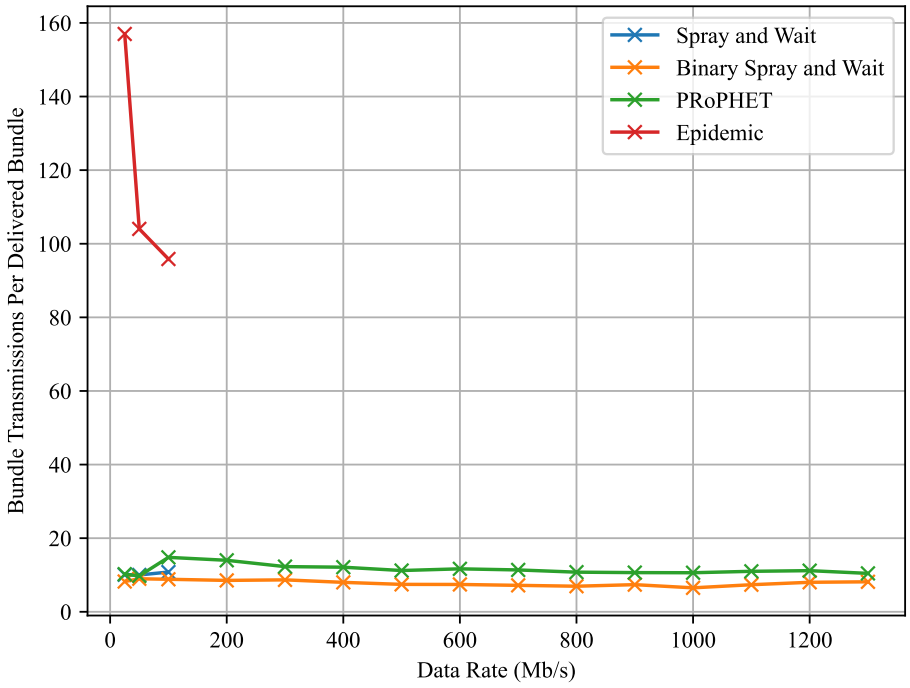



**Figure 4.7.** Bundle transmission count per delivered bundle of replication-based algorithms in the primary scenario with limited SDR capacity.

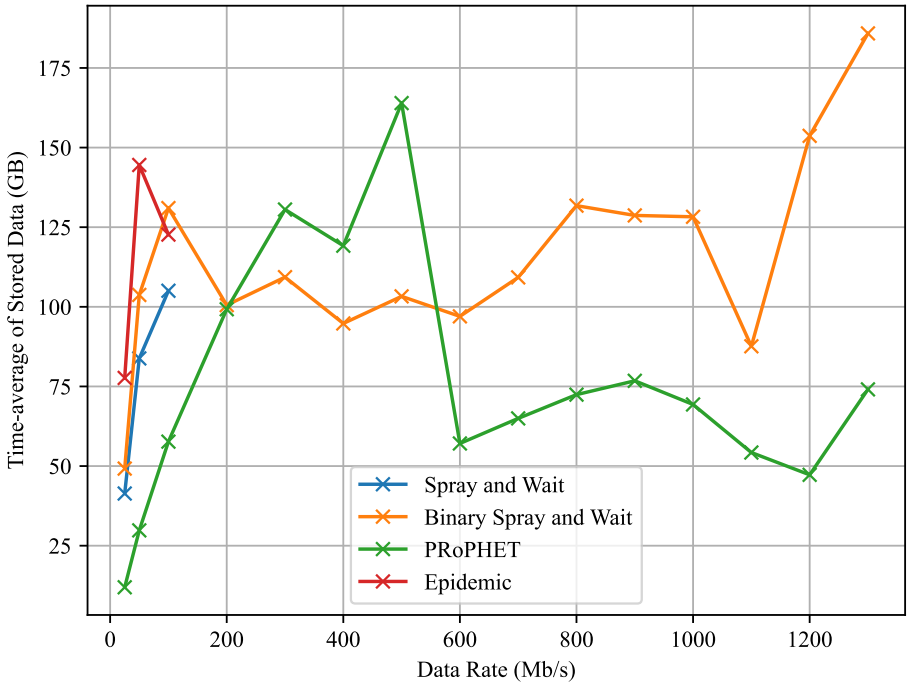



**Figure 4.8.** Time-averaged quantity of data stored in the network by replication-based algorithms in the primary scenario with limited SDR capacity.

## 4.3 CONTACT GRAPH ROUTING ALGORITHMS

Unlike replication-based routing algorithms, CGR operates by prudently forwarding bundles from node to node with no duplication. CGR has evolved in parallel with new versions of ION, incorporating new developments and enhancements. In this section, different versions of CGR are evaluated in the primary scenario.

The scenario configuration is identical to the configuration in which the replication-based algorithms are evaluated, but due to the improved simulation performance of the CGR algorithms, continuous data generation is employed. Continuous data generation provides a more realistic testing environment compared to bursts of generated data, at the cost of more computationally intensive simulations. Continuous data generation is also more intensive for the routing algorithms, as there are no periods to "recover" between bursts of generated data. All data must be routed in real time to avoid backlogs.

In this investigation, both uplink (Earth to Mars) and downlink (Mars to Earth) communications are evaluated. Bundles are generated at the Mars Base and MOC over the entire data generation period at rates from 100 Mb/s to 2000 Mb/s. Each node has an SDR capacity of 1 TB.

The candidate CGR algorithms are presented below:

- **CGR 3.5** (*cgrModel350* in DtnSim): This algorithm corresponds to the version of CGR implemented in ION v3.5.
- **CGR 3.5 Hop** (*cgrModel350_Hops* in DtnSim)*:* This algorithm corresponds to the version of CGR implemented in ION v3.5. It deviates from the standard algorithm by prioritising hop count over arrival time.
- **CGR 3.5 Probabilistic** (*cgrModel350_Probabilistic* in DtnSim)*:* This algorithm corresponds to the version of CGR implemented in ION v3.5 but with modifications inspired by both scheduled and opportunistic routing schemes. The modifications are intended to improve performance in scenarios with uncertain (or inaccurate) contact plans by considering an additional parameter that indicates the probability of delivery success via each route. A bundle is transmitted via the route with the earliest arrival time if the probability of success is sufficient, else it is transmitted via the route with the fewest hops [73].
- **CGR 3.6** (*cgrModelRev17* in DtnSim): This algorithm corresponds to the version of CGR implemented in ION v3.6 in 2017. CGR 3.6 prioritises low route latency, high route capacity, and short route length while CGR 3.5 only prioritises low route latency and short route length. Additionally, CGR 3.6, as implemented in DtnSim, provides significantly more configuration

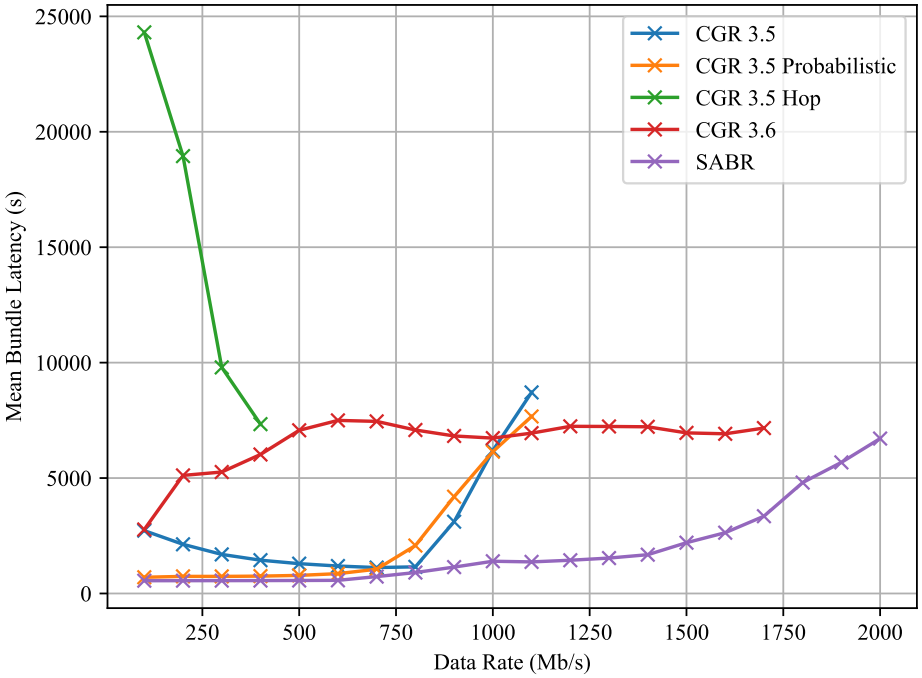
flexibility over CGR 3.5. Mechanisms such as anchoring [13] and volume awareness [74] can be configured. In this dissertation, the default CGR 3.6 configuration is used, but future work can be performed to investigate the optimal configuration(s) for interplanetary communication.

- **SABR** (*uniboCgr* in DtnSim): This algorithm corresponds to the latest version of CGR standardised by the CCSDS [36]. Unibo-CGR is an independent implementation of SABR [75]. The most significant improvement of SABR over CGR 3.6 is in its route computation strategy. Prior to ION v3.7 and SABR, CGR route computation operated using a series of Dijkstra searches on a contact graph to identify the best routes. To avoid the discovery of the same route in successive Dijkstra searches, the initial contacts of previously discovered routes are removed from the contact graph. This method of discovering unique paths becomes problematic in the presence of long-lasting initial contacts. Removal of such contacts could suppress many valid onward paths, leaving them undiscovered. Various route computation improvements have been proposed to address this issue [74], but Yen's algorithm (described in Appendix A) is recommended by the SABR standard. Yen's algorithm efficiently computes the $K$ best routes in a contact graph, ensuring a superior group of candidate routes for final route selection.

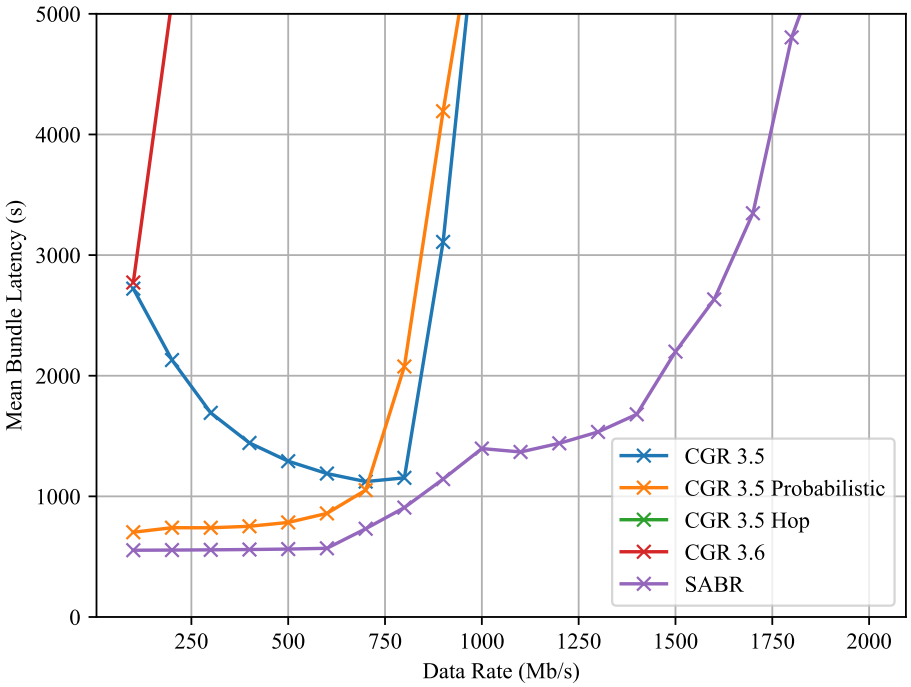
The latency and delivery ratio results in Figure 4.9 and Figure 4.10 reveal the downlink performance of the CGR algorithms in the primary scenario. It is clear that SABR has the best downlink latency across all bundle generation rates. Importantly, SABR's bundle delivery performance exceeds that of all the algorithms. CGR 3.6 has the next highest delivery ratio, but its bundle latency is poor in comparison to SABR.

CGR 3.5 and CGR 3.5 Probabilistic perform similarly. Their bundle latencies and delivery ratios begin to degrade at approximately 800 Mb/s. Before this threshold, CGR 3.5 Probabilistic maintains a bundle latency near that of SABR. CGR 3.5 has a higher bundle latency at low data generation rates which decreases as the data rate increases, eventually outperforming CGR 3.5 Probabilistic. This relationship between data rate and latency may appear counterintuitive but it is due to CGR 3.5's suboptimal route computation. Unlike its successors, CGR 3.5 does not feature more advanced route computation strategies such as anchoring or Yen's algorithm. Instead, CGR 3.5 uses Dijkstra searches to discover routes. To avoid the discovery of the same route in successive Dijkstra searches, the initial contacts of previously discovered routes are removed from the contact graph. This may leave the best route(s) undiscovered. In this scenario, CGR 3.5 discovers and selects routes that do not result in the best possible latency (approximately 545 s). As the data generation rate increases, the routes initially selected become saturated, forcing the algorithm to find and select alternative routes. These alternative

routes prove to have smaller overall latencies, reducing the mean bundle latency. This trend continues until 800 Mb/s, where latency dramatically increases and the delivery ratio declines.




**(a)** Full perspective.




**(b)** Narrow perspective.

**Figure 4.9.** Mean downlink bundle latency of CGR algorithms in the primary scenario.

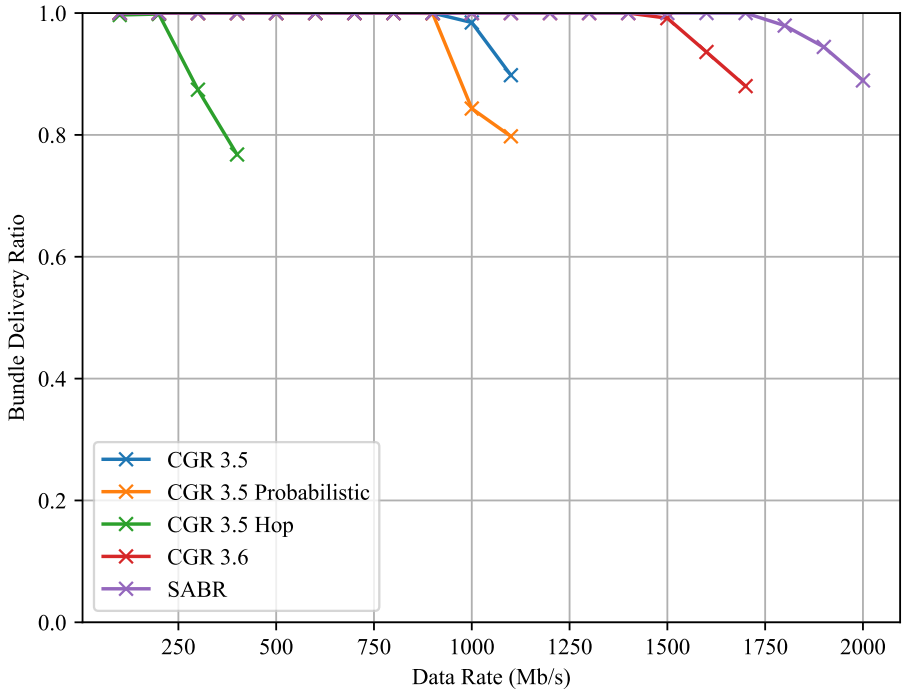



**Figure 4.10.** Downlink bundle delivery ratio of CGR algorithms in the primary scenario.

A similar relationship is seen in the case of CGR 3.5 Hop. CGR 3.5 Hop has very poor bundle latency because it prioritises routes with the smallest number of node-to-node transmissions. Such routes do not necessarily result in prompt bundle delivery, as is the case here. Again, as the bundle generation rate increases, more bundles are forced to take routes with greater numbers of hops, decreasing latency as these routes have smaller propagation delays.

The latency and delivery ratio results in Figure 4.11 and Figure 4.12 reveal the uplink performance of the CGR algorithms in the primary scenario. The reduced uplink performance relative to the downlink performance is due to the lower capacity of the physical links in the uplink direction. The uplink results confirm that SABR is the top-performing algorithm. As in the downlink direction, SABR maintains the best latency and delivery ratio relative to the other algorithms. CGR 3.5 and CGR 3.5 Hop have similar delivery ratio performance to SABR but with far worse latency.

It can be seen that the non-SABR algorithms exhibit similar performance in the uplink direction, compared to the disparate performances observed in the downlink direction. This is a result of the non-SABR algorithms' similar route computation strategies and the unique network topology in the uplink direction. At any given time, bundles generated at the MOC only have a single reasonable next node: the DSN station with a line of sight to Mars. This significantly reduces the number of candidate routes available to non-SABR algorithms, as these algorithms suppress the initial contact (the DSN

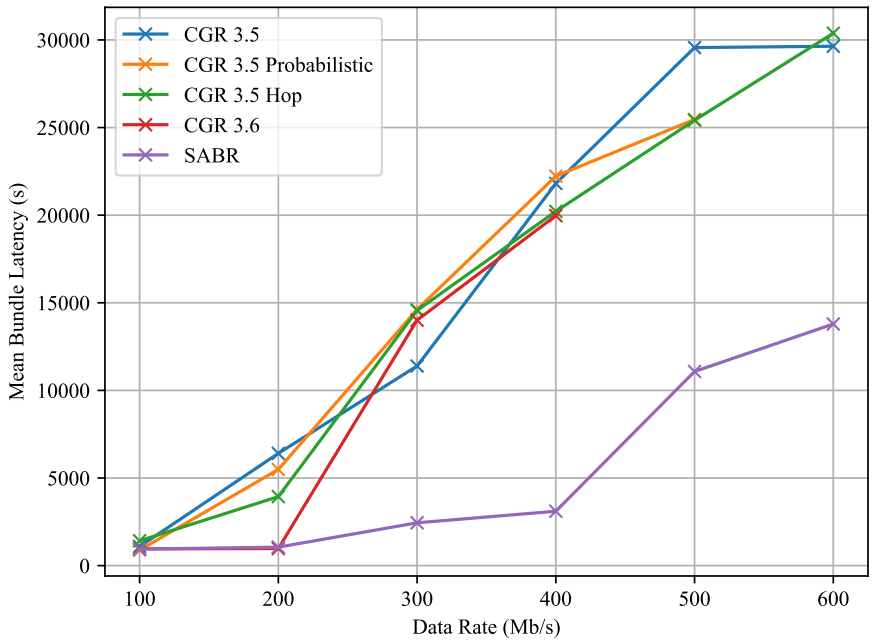



**Figure 4.11.** Mean uplink bundle latency of CGR algorithms in the primary scenario.

station) of previously discovered routes. Consequently, the algorithms have a limited, similar group of candidate routes from which to select for MOC bundles. SABR, with Yen's algorithm, discovers and considers a greater variety of routes that begin with the DSN station with a line of sight to Mars, allowing for better overall performance.

Figure 4.13 and Figure 4.14 provide additional metrics and are plotted only at data generation rates where the algorithms maintain high delivery rates. Figure 4.13 shows the efficiency of the algorithms by revealing the number of bundles transmitted between nodes in the network for each bundle that reaches the destination. Notice that all the CGR algorithms use far fewer transmissions to deliver each bundle relative to the replication-based algorithms. Figure 4.7 shows that the best-performing replication-based algorithms incur approximately ten bundle transmissions for every delivered bundle whereas CGR algorithms incur no more than five transmissions per bundle on average.

Figure 4.14 shows the time-averaged quantity of data stored throughout the entire network in node SDRs. As the data generation rate increases, more data is stored throughout the network. Notice that SABR uses the fewest storage resources, indicating that its bundles spend more time being transmitted between nodes than stored at nodes, compared to the other algorithms.

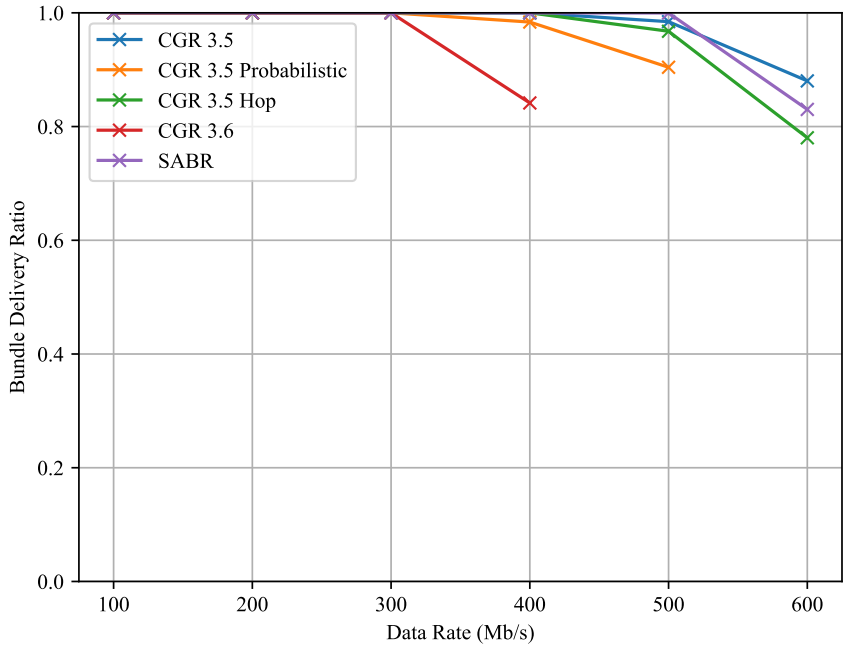



**Figure 4.12.** Uplink bundle delivery ratio of CGR algorithms in the primary scenario.

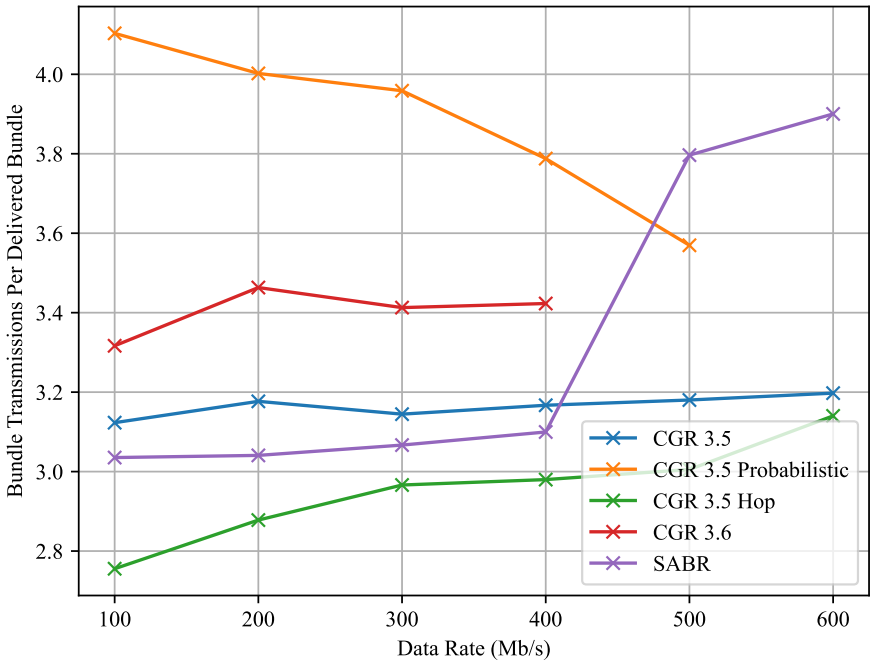



**Figure 4.13.** Bundle transmission count per delivered bundle of CGR algorithms in the primary scenario.

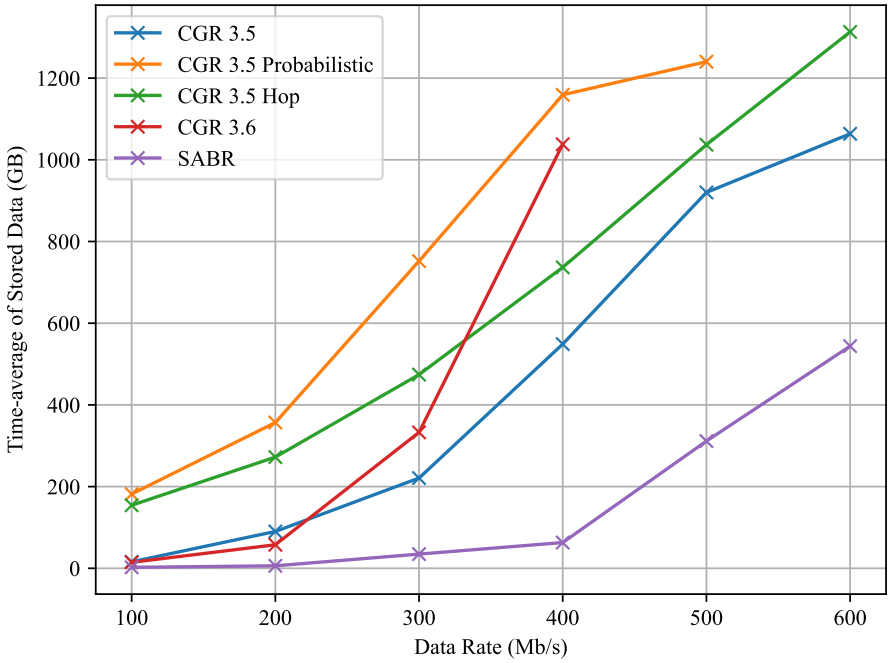



**Figure 4.14.** Time-averaged quantity of data stored in the network by CGR algorithms in the primary scenario.

### 4.3.1 Maximum throughput

The Ford-Fulkerson algorithm, described in Section 3.4.2, calculates the maximum throughput between two nodes in a network. The maximum throughput provides a baseline for performance evaluation, putting the measured throughput into perspective.

The time-averaged maximum throughput between the Mars base and MOC over the course of the primary scenario is calculated to be 1741 Mb/s in the downlink direction. Recall that this value is optimistic, and serves as an upper bound to the potential performance of a network. Given a saturated network at maximum throughput, the expected mean latency of bundles is calculated to be 581 s. This calculation is performed by considering the best-case propagation delay of each route, and the proportion of traffic that travels along each route when the network is at maximum throughput. This expected latency is also optimistic since it is calculated using a network that is permanently connected. In reality, intermittently connected networks often require bundles to wait at nodes for onward contacts. This increases the mean latency relative to permanently connected networks.

The time-averaged maximum throughput and corresponding expected latency are plotted in Figure 4.15, along with the mean downlink bundle latency of the CGR algorithms. Notice that SABR maintains a latency lower than the expected latency only until 600 Mb/s. Beyond this point, SABR's latency increases steadily. At the theoretical, time-averaged maximum throughput of the network, SABR's

latency is significantly higher than the expected latency. This is likely because there are better routing decisions available (SABR can still be improved), and because the benchmark latency and throughput are too optimistic (the maximum throughput calculation can be improved).

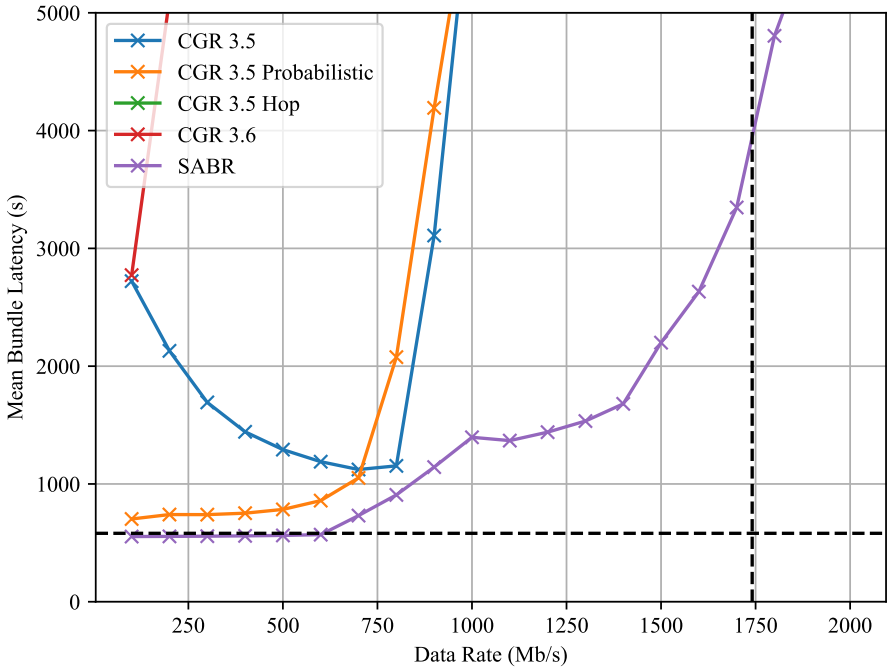



**Figure 4.15.** Mean downlink bundle latency of CGR algorithms in the primary scenario. The dotted horizontal line represents the expected mean latency of a saturated network. The dotted vertical line represents the theoretical maximum throughput of a saturated network.

A permanently connected network, even with adjusted link capacities, does not sufficiently represent an intermittently connected network when calculating the maximum throughput. Future work is required to obtain a more accurate benchmark of optimal DTN performance, allowing for a better understanding of the untapped performance of DTN routing algorithms.

## 4.4 REPLICATION-BASED AND CGR ALGORITHM COMPARISON

Thus far, replication-based and CGR routing algorithms have been evaluated in separate scenarios with different data generation schemes. It would, however, be beneficial to directly compare the two types of algorithms in the same scenario. Due to simulation runtime constraints, the replication-based algorithms cannot be evaluated in the continuous data generation scenarios of Section 4.3. Instead, the CGR algorithms can be evaluated alongside the replication-based algorithms in the burst-type data generation scenarios of Section 4.2. Figure 4.16 and Figure 4.17 show the top-performing replication-based algorithms' latency and delivery ratio performance alongside the best-performing CGR algorithm: SABR.

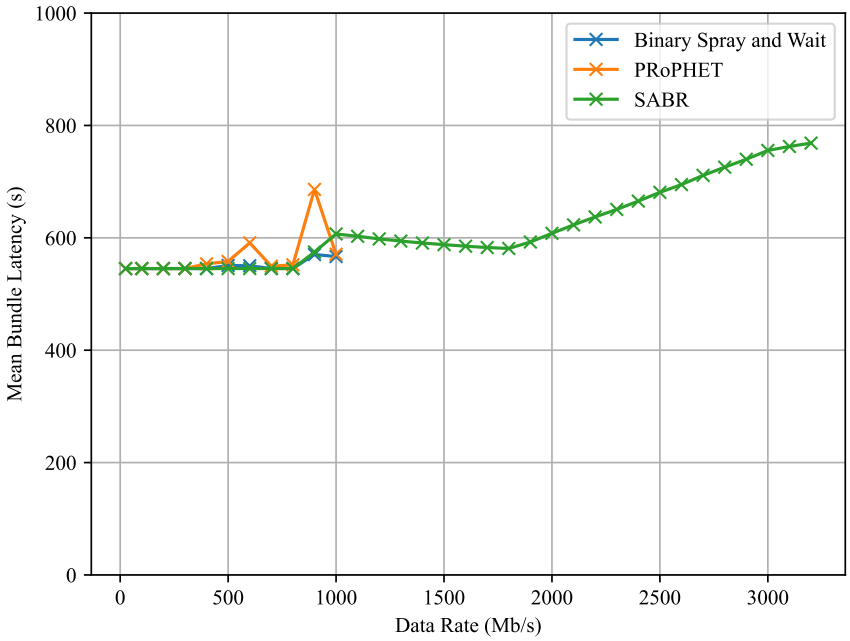



**Figure 4.16.** Mean bundle latency of the best-performing replication-based and CGR algorithms in the primary scenario.

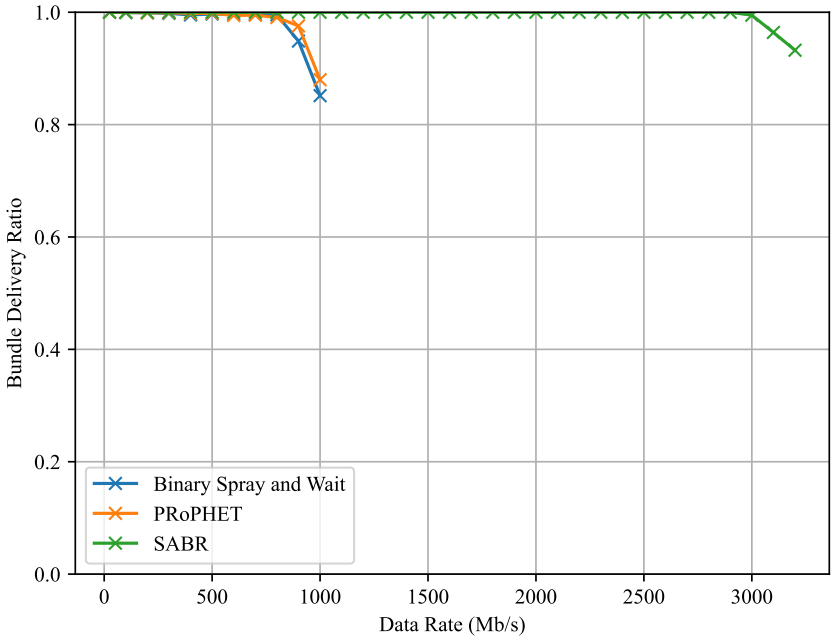



**Figure 4.17.** Bundle delivery ratio of the best-performing replication-based and CGR algorithms in the primary scenario.

It is evident that SABR has significantly greater success in bundle delivery relative to PRoPHET and Binary Spray and Wait. SABR's delivery ratio begins to degrade at 3000 Mb/s compared to the degradation of the replication-based algorithms at 800 Mb/s. CGR algorithms appear to be

better suited to deep-space scenarios, exploiting their knowledge of upcoming contacts to selectively forward bundles. However, future work should be performed to rigorously evaluate the performance of PRoPHET and Binary Spray and Wait in scenarios with continuous data generation.

## 4.5 SATELLITE NETWORK SCENARIO

In addition to the primary scenario (early crewed mission to Mars), the routing algorithms are also evaluated in realistic low Earth orbit (LEO) satellite network scenarios. These scenarios provide a completely different topology for algorithm performance analysis. Two satellite configurations, created by the authors of DtnSim [13], are employed: a sun-synchronous along-track and a Walker Delta configuration, as depicted in Figure 4.18.

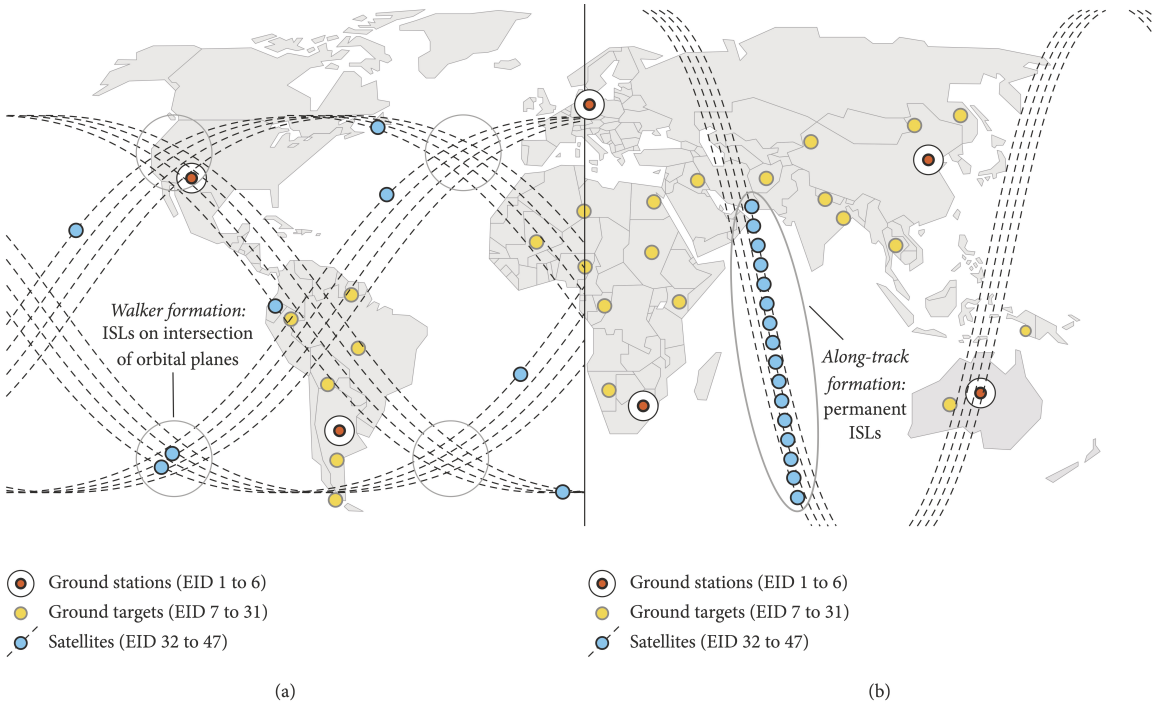



**Figure 4.18.** (a) Walker satellite formation and (b) along-track satellite formation. Taken from [13] © 2017 J. A. Fraire et al. CC BY 4.0.

A Walker formation and an along-track formation are two distinct configurations of satellites in LEO. In a Walker formation, the satellites are evenly spaced in a series of inclined orbital planes. The spacing between satellites results in a large coverage area of the satellite network and ensures that points on Earth are regularly serviced by an overhead satellite. Satellites can communicate with each other at the intersections of orbital planes.

An along-track formation consists of satellites that are positioned tightly in similar planes. The satellites orbit together as a train, with a permanent inter-satellite link (ISL) between adjacent satellites. A

limitation of this arrangement is the concentrated coverage area since only a narrow ground track is followed. Points on Earth thus have to wait longer before being serviced by the train of satellites.

Each configuration consists of 16 LEO satellites which facilitate communication between 6 ground stations and 25 ground targets. The ground targets can be seen as remote locations that need to communicate with a MOC. The MOC is reachable by the ground stations through the Internet. These relationships can be seen in Figure 4.19.

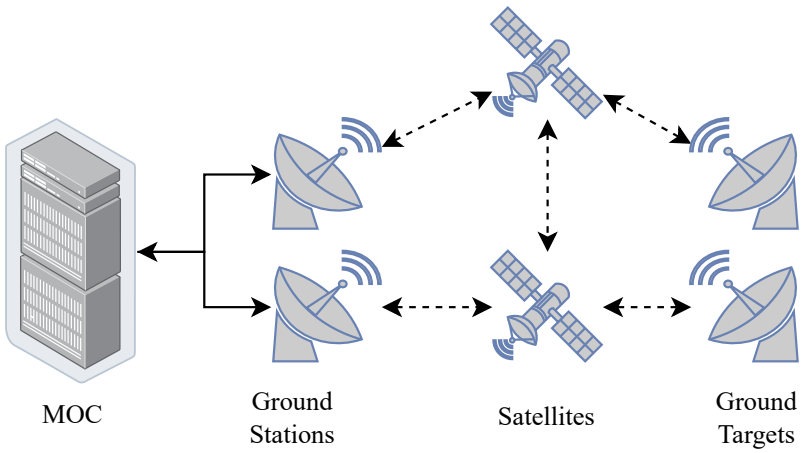



**Figure 4.19.** Simplified satellite network topology. Dotted lines indicate intermittent links (ISLs are continuous in along-track formation). Solid lines indicate continuous links.

The simulation has a duration of 24 hours. Data is generated by the MOC and all ground targets over the first five hours of the simulation. As in the primary scenario, the algorithms are evaluated over a range of data generation rates. Zero-latency, 100 kb/s links are assumed between nodes. Only the CGR algorithms are evaluated, as it is evident that replication-based algorithms are not well suited to space applications.

### 4.5.1 Along-track formation

In the along-track satellite network scenario, SABR is the best-performing algorithm in terms of both latency and bundle delivery ratio. Figure 4.20 and Figure 4.21 reveal that CGR 3.5, CGR 3.5 Probabilistic and SABR perform similarly at low data generation rates, but SABR stands out with the lowest latency and highest delivery ratio as the network load increases.

Notice that a 100% delivery ratio is never achieved by any algorithm in this scenario. This can be attributed to the limited coverage provided by the along-track formation. In the 24-hour scenario, the satellite train does not pass over certain ground targets at the appropriate time, leaving bundles undelivered. Nevertheless, the relative performance of the algorithms can still be investigated.

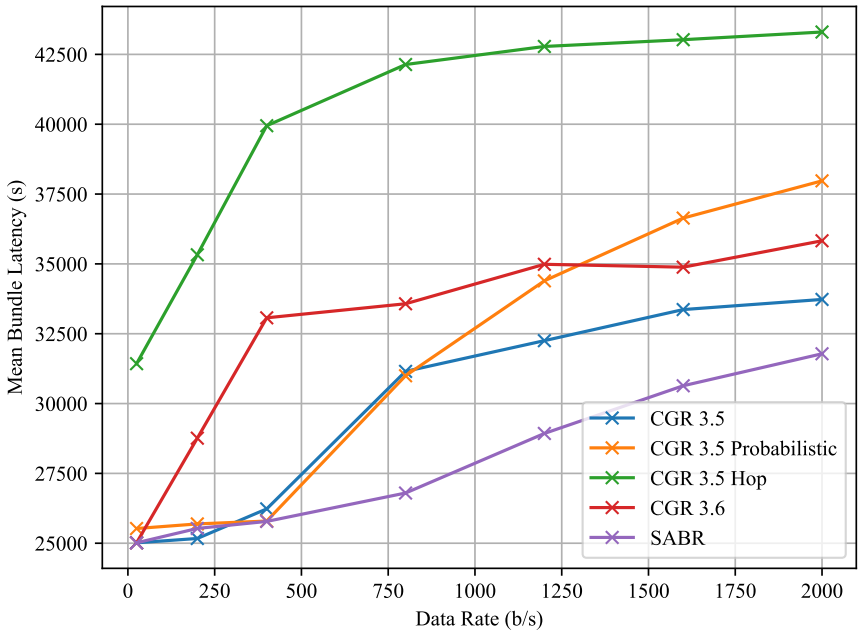



**Figure 4.20.** Mean bundle latency of CGR algorithms in the along-track satellite scenario.

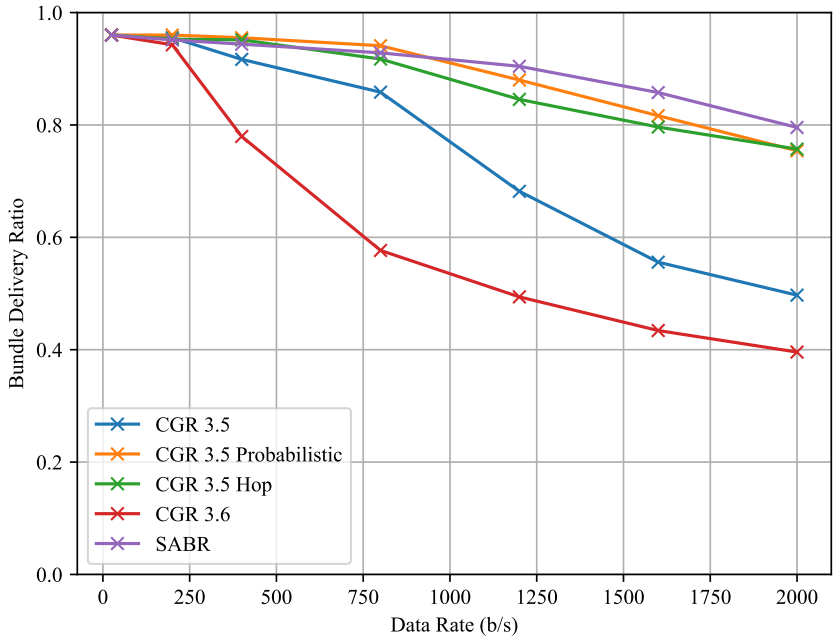



**Figure 4.21.** Bundle delivery ratio of CGR algorithms in the along-track satellite scenario.

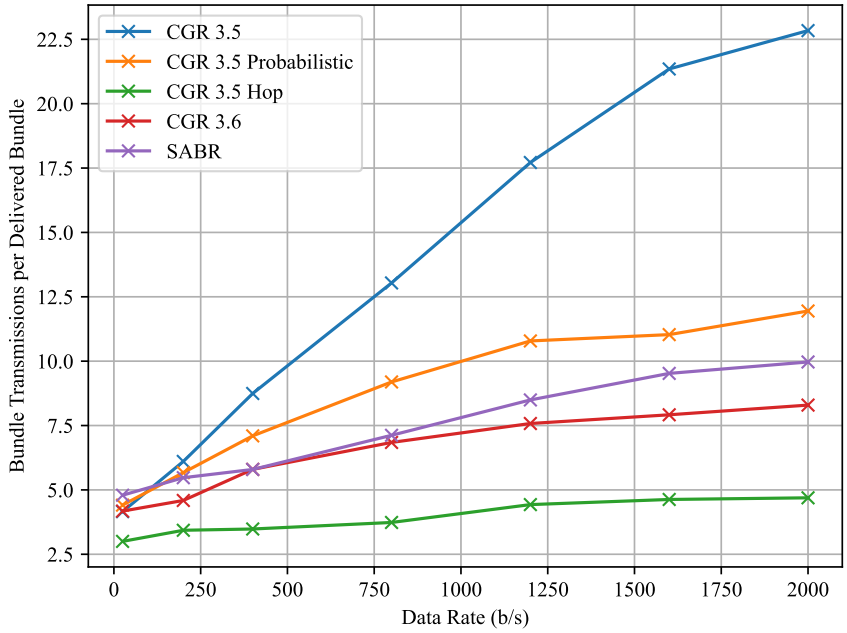



**Figure 4.22.** Bundle transmissions count per delivered bundle of CGR algorithms in the along-track satellite scenario.

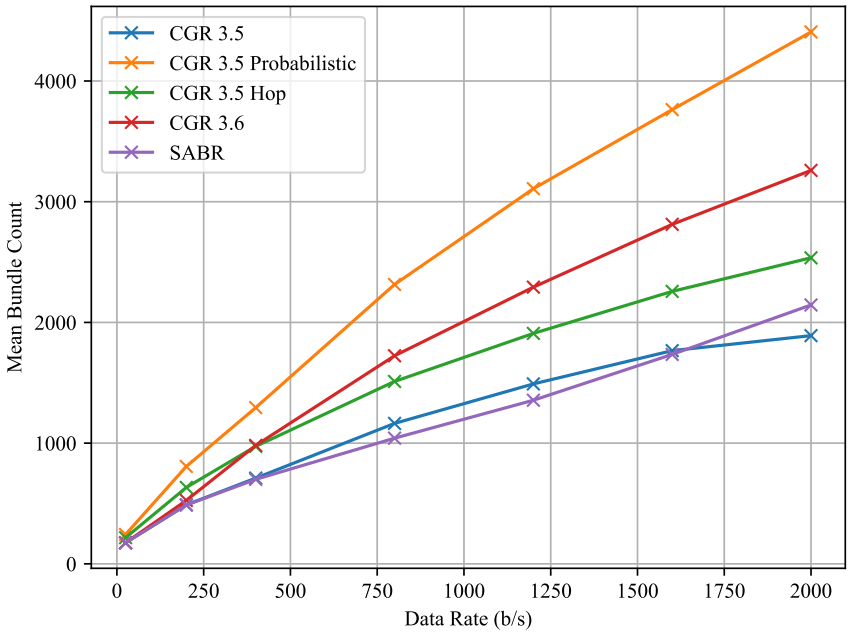



**Figure 4.23.** Time-averaged quantity of bundles stored by CGR algorithms in the along-track satellite scenario.

Notice that CGR 3.5 Hops has poor latency performance relative to the other algorithms. CGR 3.5 Hops attempts to minimise the number of hops, as evident in Figure 4.22. This technique is not effective in this scenario. Instead, it is beneficial to use additional hops to redistribute bundles between

satellites in the train to reduce congestion and queues at the satellites. In this scenario, a significant constraint is the limited communication opportunities between the satellite train and ground nodes. It is therefore advantageous to spread bundles between nodes in the train to minimise queues and maximise the use of the train's collective bandwidth when passing over a ground node. CGR 3.5 Probabilistic and SABR produce favourable delivery rates and latencies relative to CGR 3.5 Hops despite making many more transmissions.

SABR, again, makes frugal use of storage resources. Figure 4.23 shows that SABR and CGR 3.5 store the fewest number of bundles over time in this scenario, indicating that their bundles spend more time being transmitted between nodes than stored at nodes, compared to the other algorithms.

### 4.5.2 Walker formation

In the Walker satellite network scenario, there is no clear best-performing algorithm. Figure 4.24 and Figure 4.25 reveal that CGR 3.5, CGR 3.6, and SABR have similar latencies and delivery ratios across all bundle delivery rates. This is in contrast to the along-track satellite network scenario, and the primary scenario, where SABR is the clear best performer.

The similar performance of CGR 3.5, CGR 3.6, and SABR in the Walker scenario can be attributed to the specific characteristics of the Walker formation. The formation results in a more evenly distributed network topology, which makes it more difficult for any one algorithm to gain a significant advantage. In the Walker formation, there is typically a distinct best route from source to destination, with no similar alternative routes. CGR 3.5, CGR 3.6, and SABR identify the optimal route for almost all bundles. As such, the transmission efficiency and network storage usage in Figure 4.26 and Figure 4.27 are nearly identical for these algorithms.

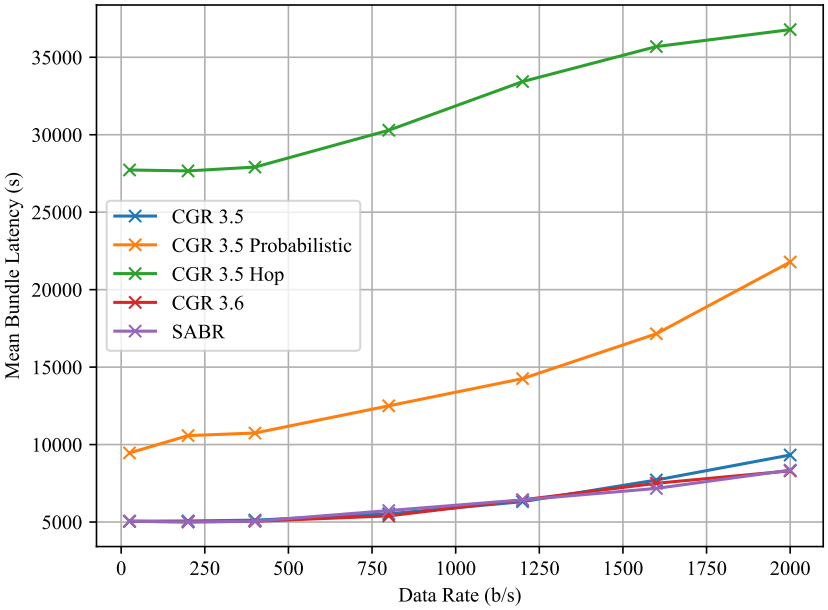



**Figure 4.24.** Mean bundle latency of CGR algorithms in the Walker satellite scenario.

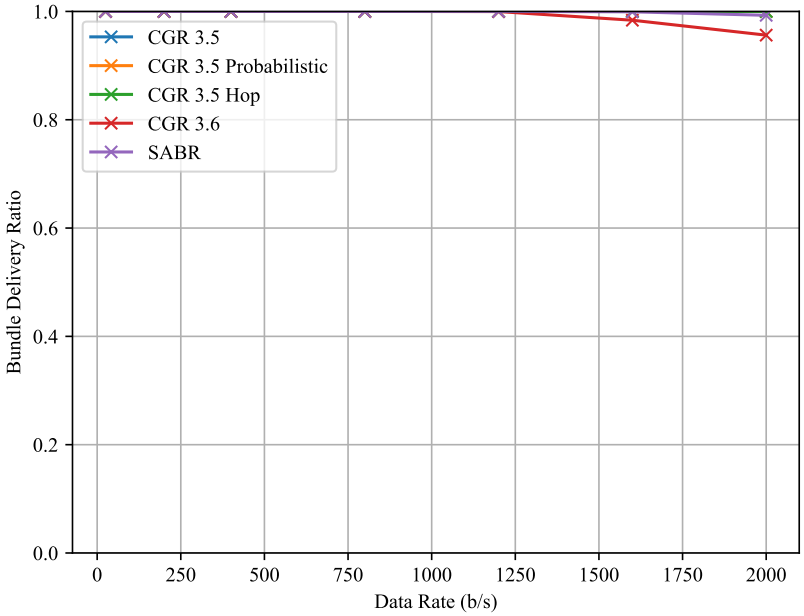



**Figure 4.25.** Bundle delivery ratio of CGR algorithms in the Walker satellite scenario.

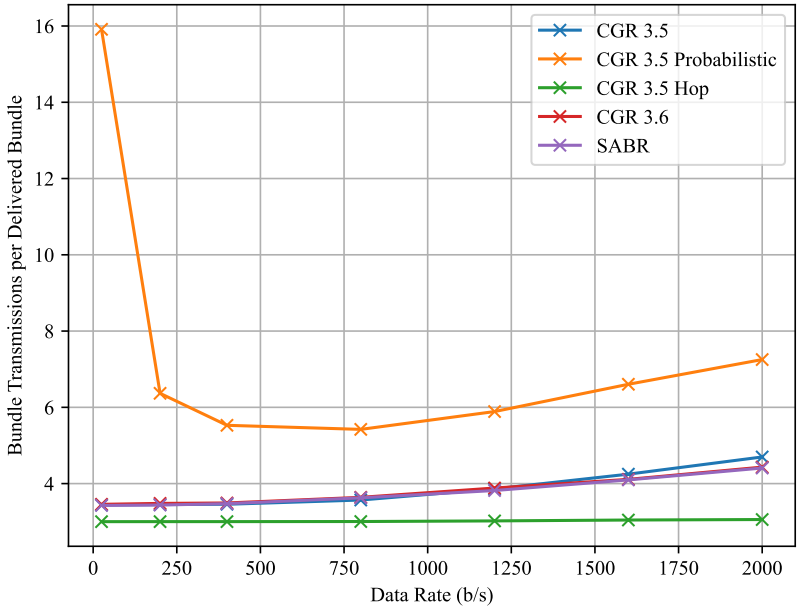



**Figure 4.26.** Bundle transmissions count per delivered bundle of CGR algorithms in the Walker satellite scenario.

Overall, the results from the Walker satellite network scenario reveal that in certain network configurations, CGR algorithms perform similarly. The differences between routing algorithms are best revealed when there are multiple candidate paths between source and destination.

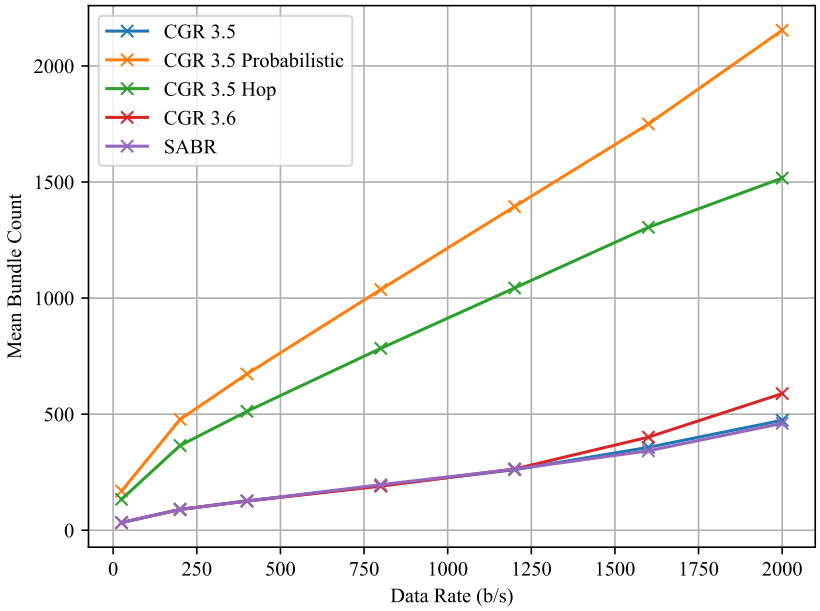



**Figure 4.27.** Time-averaged quantity of bundles stored by CGR algorithms in the Walker satellite scenario.

## 4.6 CHAPTER SUMMARY

This chapter presented the performance of replication-based routing algorithms and CGR algorithms. Replication-based algorithms were quickly deemed unsuitable for deep-space environments due to their inefficiency and inability to scale to meet the demands of high traffic loads in resource-constrained networks. The simulations demonstrated that replication-based algorithms led to network congestion and poor delivery ratios, particularly in scenarios with limited node storage capacities.

On the other hand, CGR algorithms emerged as far more promising solutions for space-based networks, leveraging knowledge of network contacts to optimise bundle delivery in challenging environments. Among the CGR algorithms evaluated, SABR consistently outperformed others, exhibiting superior latency and delivery ratios across various network configurations. The findings emphasise the critical importance of network resource efficiency, enabling effective communication in space missions, with SABR standing out as the overall best-performing algorithm for crewed missions to Mars.

# CHAPTER 5 TRAFFIC SPREADING ENHANCEMENT

## 5.1 CHAPTER OVERVIEW

This chapter presents a new enhancement to the SABR protocol. The enhancement, labelled "traffic spreading," is intended to improve performance in DTNs that contain competing candidate routes between source nodes and destination nodes.

First, the details of the SABR deficiency are presented, followed by the details of the SABR modifications required to implement traffic spreading. A reference scenario is introduced to showcase the operation of traffic spreading and its effect on routing in DTNs. Finally, SABR with the traffic spreading enhancement is evaluated in the primary Martian scenario of Section 3.2 and the satellite network scenario of Section 4.5.

## 5.2 SABR DEFICIENCY

During the simulations of the primary scenario, a potential avenue for enhancement to the SABR protocol was identified. It was realised that at a given node, SABR would select a single particular route for all bundles despite the availability of other similar routes. This characteristic becomes problematic if multiple independent nodes have access to the same set of similar routes. In such a scenario, each node will select the same route for each bundle, potentially causing downstream congestion.

Consider the following simple scenario as an example: two Mars bases on opposite hemispheres of Mars have access to two relay satellites in Martian orbit, illustrated in Figure 5.1. These relays communicate with a station on Earth. This example represents a simplified version of the primary scenario introduced in Section 3.2. In this example, there is no direct communication channel between the Mars bases. Similarly, there is no direct communication channel between the relays.

Consider the network in Figure 5.1(a). The propagation delay of both links between the source node and relays is equal. S1 has two candidate routes to D: red and green. There is no significant difference

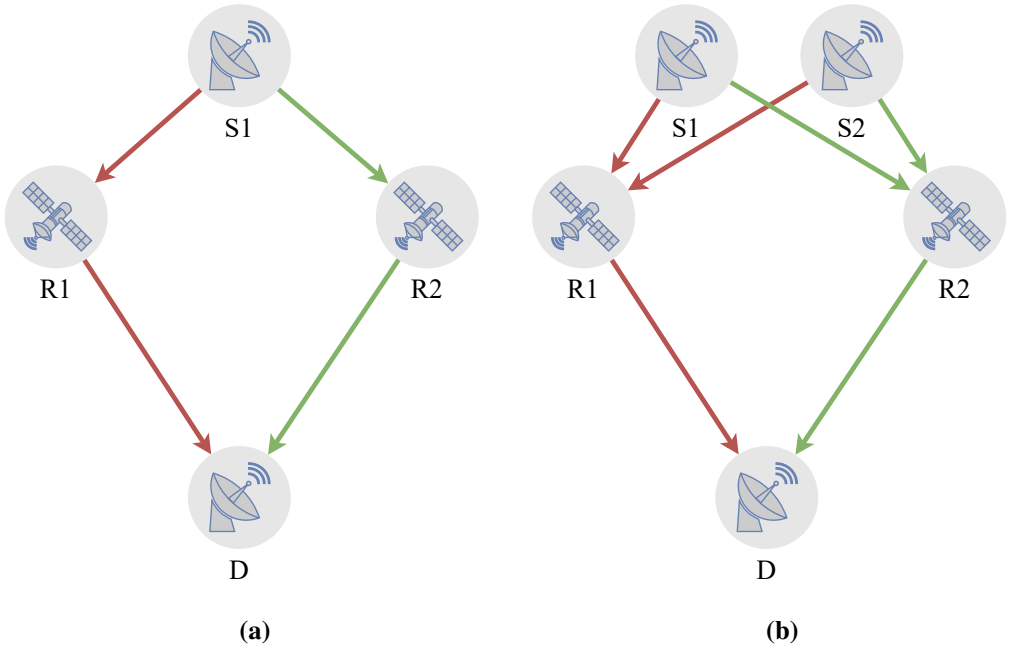



**Figure 5.1.** Simple networks comprising source node(s) (S1 and S2) with communication routes to a destination (D) via relays (R1 and R2).

between the red and green routes. Under the SABR protocol, the red route would be chosen for all bundles as it has the smallest entry node ID (see forwarding rules in Section 2.4.4.3). With a single source node, this is a valid route selection.

Figure 5.1(b) shows the network with the addition of a second source node, S2. Routing decisions at S2 are identical to those at S1. Thus, all bundles originating at S2 are routed along the red route. With all bundles from both S1 and S2 directed along the red routes, the long haul link between R1 and D has the potential to become congested while the equivalent green link between R2 and D remains unused.

Coordination between S1 and S2 or better SABR forwarding decisions are required to better utilise available network resources. While coordination between nodes is theoretically possible, this would require fundamental changes to the SABR protocol. Furthermore, while coordination in this scenario may be feasible, it may not be effective in all DTNs. In general, there is no guarantee of timely communication between nodes in a DTN. In an interplanetary network, source nodes could be light-minutes or even light-hours apart, preventing real-time coordination efforts.

Instead, modifications to SABR are proposed to uniformly route bundles when multiple similar routes are available. These modifications aim to distribute bundles among all available routes thereby improving network resource utilisation and reducing congestion.

## 5.3 TRAFFIC SPREADING ENHANCEMENT

The proposed traffic spreading enhancement is a modification to Unibo-CGR [75], an independent implementation of SABR. The enhancement is summarised in Figure 5.2 and can be compared to the standard operation of SABR given in Figure 2.2.

SABR selects the same route for bundles with the same destination, provided the route has sufficient capacity. This is not an issue if there is only a single node producing data on the network. However, if there are additional data sources, each source may independently select routes that use the same contacts, resulting in congestion at these contacts. In DTN, there is no guarantee that nodes are aware of up-to-date traffic information throughout the entire network, and it is often impossible to know due to propagation delays. Thus, improved route selection is required to reduce potential congestion.

Traffic spreading aims to distribute bundles between all similar candidate routes. This is achieved by dividing the forwarding stage into two parts: (A) best route identification and (B) route selection.

### 5.3.1 Forwarding

#### 5.3.1.1 Part A: Best route identification

In ordinary SABR, the best route of all candidate routes is the route with the earliest PBAT. When there are multiple routes with the same PBAT, the tiebreaker rules result in the same route being selected for all bundles. With Traffic Spreading, all routes with similar PBATs are considered, and instead of tiebreak rules, a weighted round-robin [76] style distribution protocol is used.

From the candidate route list, the route with the earliest PBAT is identified. All other candidate routes are then evaluated and routes with similar PBATs are placed in a list of best routes. An adjustable tolerance is used to determine how similar a route must be. A tolerance of 1%, for example, would allow routes with a PBAT of 1% greater than the fastest route to be included in the best routes list.

#### 5.3.1.2 Part B: Route selection

To avoid the selection of the same route during each execution of the algorithm, the tiebreak rules are replaced by a random selector based on a random number generator. This results in the even distribution of bundles between all similar routes, reducing the likelihood of overloading a single contact. If each node distributes its outgoing bundles between all best routes, traffic will be spread more evenly, making efficient use of all network resources and avoiding the concentration of bundles at a minority of contacts.

The distribution of bundles can be improved by assigning weights to each route. Routes with more

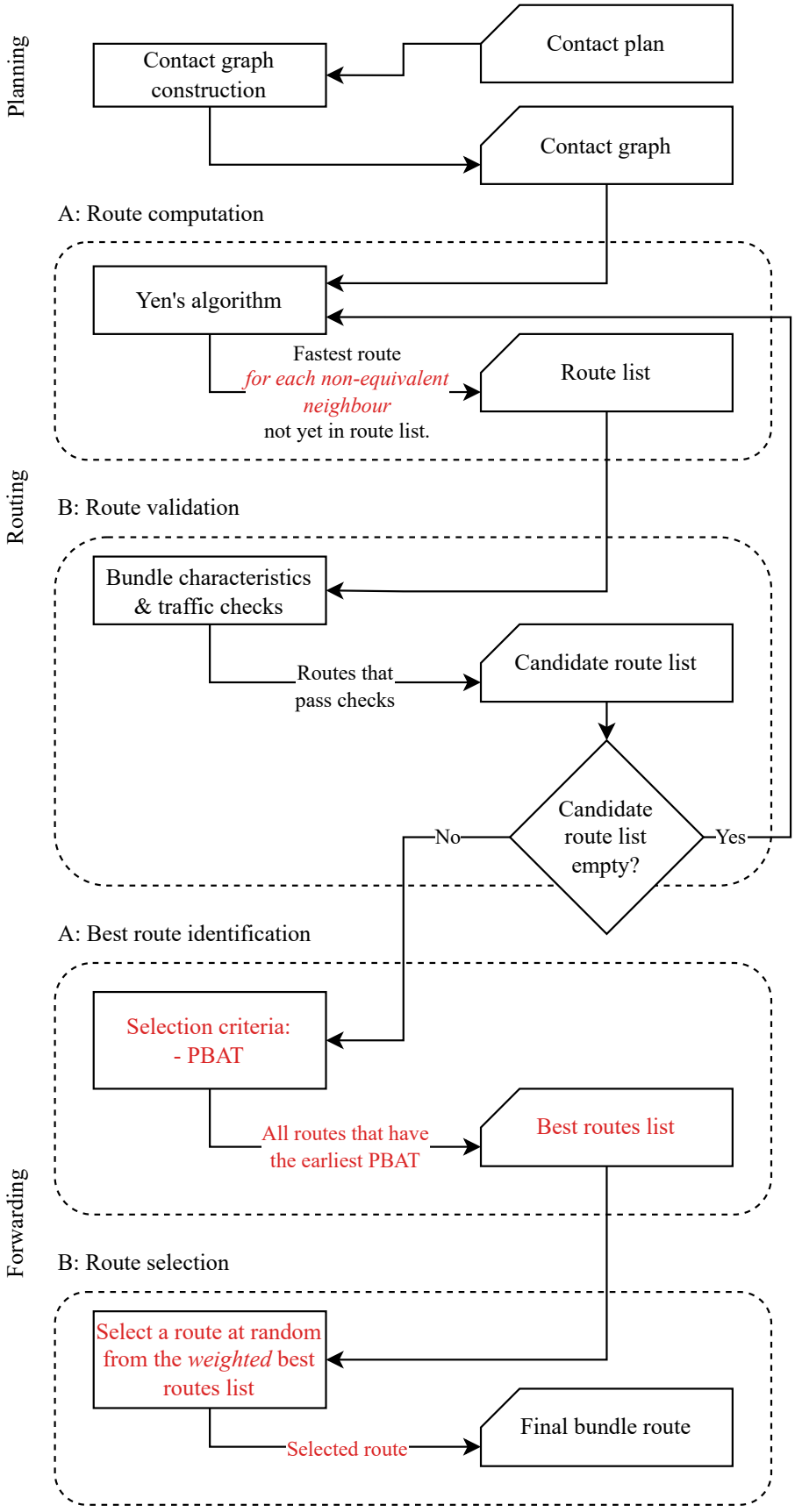



**Figure 5.2.** Overview of SABR with the traffic spreading enhancement.

capacity should be chosen more often to avoid saturating lower-capacity routes. As such, the best routes are weighted according to their route volume limit. This metric represents the expected available capacity of the most constrained contact (the bottleneck contact) of the entire route, calculated using the local traffic information at the current node.

### 5.3.2 Routing

Another modification to SABR is made in the routing stage, to improve the routes available for selection in the forwarding stage. During route computation, the route list is appended not with only the next fastest route to the destination, but with the next fastest route that begins with each neighbour of the current node, similar to the one-route-per-neighbour enhancement in [75]. This is intended to increase the quantity of candidate routes available for selection.

This step also excludes routes that begin with equivalent neighbours. An equivalent neighbour is a neighbouring node that can be reached with negligible delay and whose onward contacts are similar to the current node. This typically occurs when multiple nodes occupy a similar location. In the following reference scenario, relay 1 and relay 2 would be considered equivalent in the downlink direction if they shared an instantaneous link between them. This restriction is implemented to prevent repeated, "ping-pong", transmissions of bundles between equivalent nodes.

### 5.3.3 Existing load distribution schemes

Traffic spreading is inspired by features of existing load distribution schemes present in traditional networks. For example, Cisco routers have a software function labelled "load balancing" which allows routers to use multiple paths to a destination when forwarding packets [77]. Load balancing is implemented by default if a routing table has multiple routes to a destination. There are two types of load balancing: per-destination and per-packet load balancing. Traffic spreading is most similar to per-packet load balancing where each packet is forwarded along a different path in a round-robin fashion, guaranteeing equal load across all links.

Traffic spreading is also inspired by the routing strategy of equal-cost multi-path routing (ECMP) [78]. Under ECMP, packets to a single destination can be forwarded along several best paths of equal cost. Traffic spreading's weighted route selection is comparable to Palo Alto Networks' Weighted Round Robin ECMP extension that prioritises paths with greater link capacities and speed [79].

## 5.4 REFERENCE SCENARIO

A simple reference scenario is used to highlight the effect of traffic spreading on networks that exhibit symmetry. Although this reference scenario does not represent any particular existing/future scenario, it has some resemblance to the Martian scenarios of [51] and [80] and is inspired by the reference scenario of [75]. The scenario is intentionally simple to clearly demonstrate the traffic spreading enhancement. The symmetry present in this reference scenario is useful to test SABR's ability to find the best route in the presence of equivalent choices. With the expansion of the Interplanetary Internet, the quantity of equivalent routes is expected to increase.

Consider a scenario where two space assets are streaming data back to an MOC on Earth. Two relays exist to bridge the gap between the spacecraft and Earth. This simple scenario is depicted by the topology in Figure 5.3.

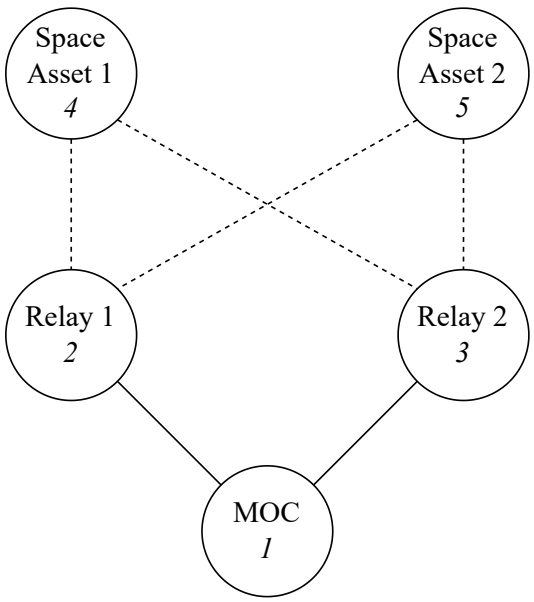



**Figure 5.3.** Reference scenario topology. Dotted lines indicate intermittent links. Solid lines indicate continuous links.

### 5.4.1 Contact plan

The reference scenario's contact plan is given in Table 5.1 and visualised in Figure 5.4.

Note that:

- The continuous contacts are declared as long contacts that span the duration of the scenario.
- The contacts are unidirectional and only downlink (space assets to MOC) contacts are represented in the table. Uplink contacts would be separate entries in the reverse direction.

**Table 5.1.** Reference scenario contact plan.

| Start time (s) | End time (s) | Source node | Destination node | Data rate (kb/s) |
|---|---|---|---|---|
| 0 | 100 | 5 | 2 | 256 |
| 0 | 100 | 4 | 2 | 256 |
| 20 | 80 | 5 | 3 | 256 |
| 20 | 80 | 4 | 3 | 256 |
| 120 | 220 | 5 | 2 | 256 |
| 120 | 220 | 4 | 2 | 256 |
| 160 | 210 | 5 | 3 | 256 |
| 160 | 210 | 4 | 3 | 256 |
| 0 | 300 | 2 | 1 | 256 |
| 0 | 300 | 3 | 1 | 256 |

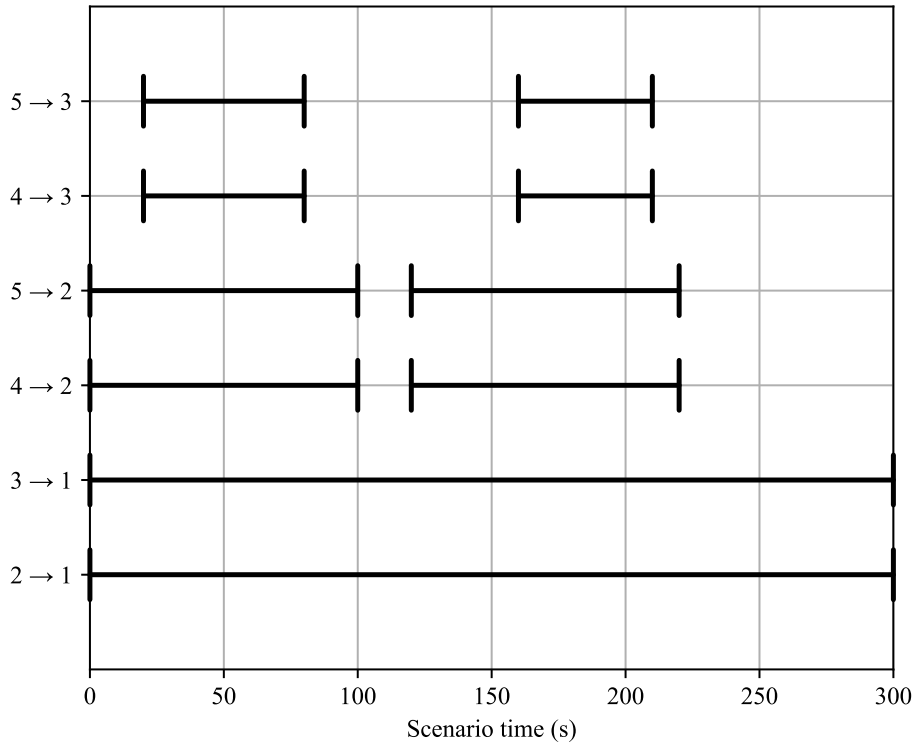



**Figure 5.4.** Visualisation of the reference scenario contact plan.

- An equal propagation delay (of 5 seconds) is inserted on all links, for simplicity. In this symmetrical reference scenario, the propagation delay has no effect on routing decisions.

### 5.4.2 Simulations

The reference scenario is configured in DtnSim to perform the requisite networking simulations. At each space asset, 15 bundles of 100 KB are produced over the first 75 seconds of the scenario, for an effective data generation rate of 320 kb/s. The downlink progression of each bundle is logged for analysis.

First, a SABR baseline is established by executing the scenario without the traffic spreading enhancement. The results in Figure 5.5 reveal that all bundles from both space assets are routed via relay 1.

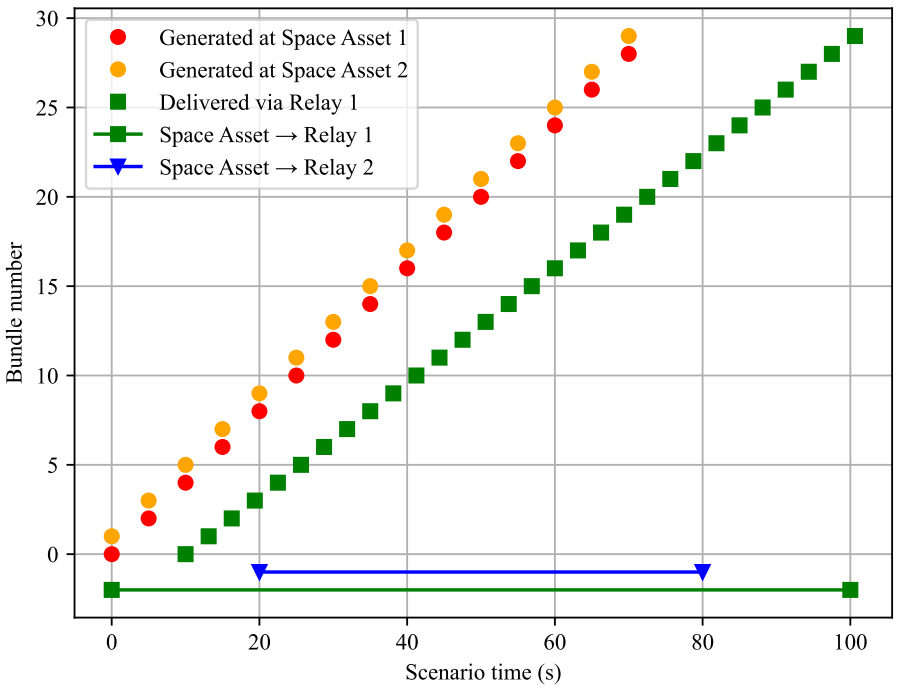



**Figure 5.5.** Generation and delivery times of bundles routed by SABR.

Without the presence of any queueing delays to impact PBAT, each source node selects relay 1 as the next node for each bundle, despite the availability of the nested contact to relay 2. The tie-break rules result in all bundles being routed to relay 1 as its contact has a later termination time. Notice that bundle latencies increase over time due to congestion at relay 1.

Next, the scenario is executed with traffic spreading enabled. A tolerance of 0% is set to only accept candidate routes with identical PBATs to the best route. In this ideal scenario, all routes from space

assets to the MOC have identical PBATs. The results in Figure 5.6 reveal that the bundles are now routed via both relays.

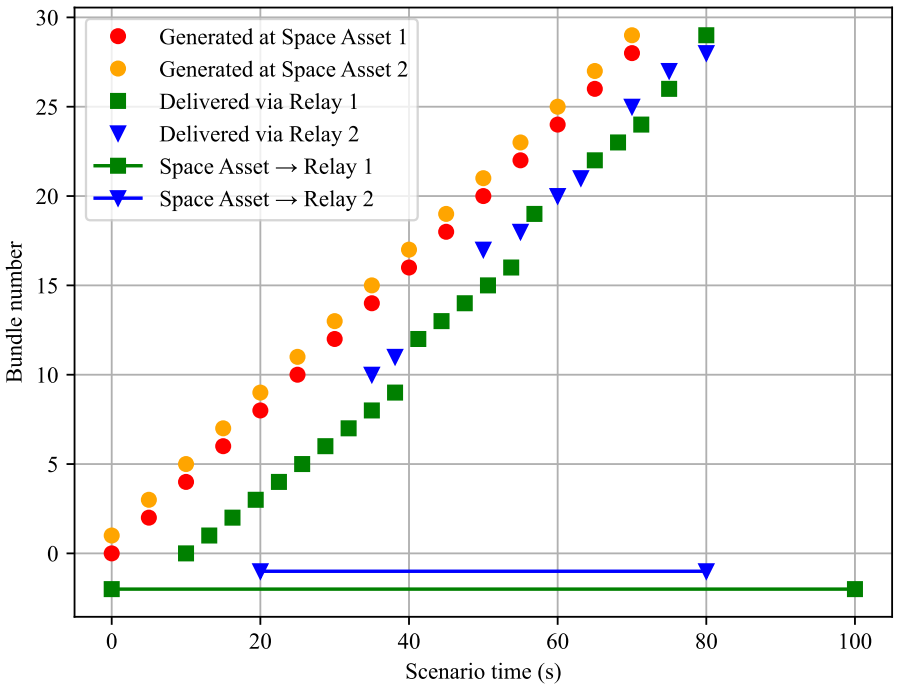



**Figure 5.6.** Generation and delivery times of bundles routed by SABR with the traffic spreading enhancement.

With the traffic spreading enhancement, bundles are distributed between the two relays, reducing congestion at relay 1, and reducing the average bundle delay. Notice that all bundles are delivered within the first 80 seconds of the simulation, compared to 100 seconds without traffic spreading.

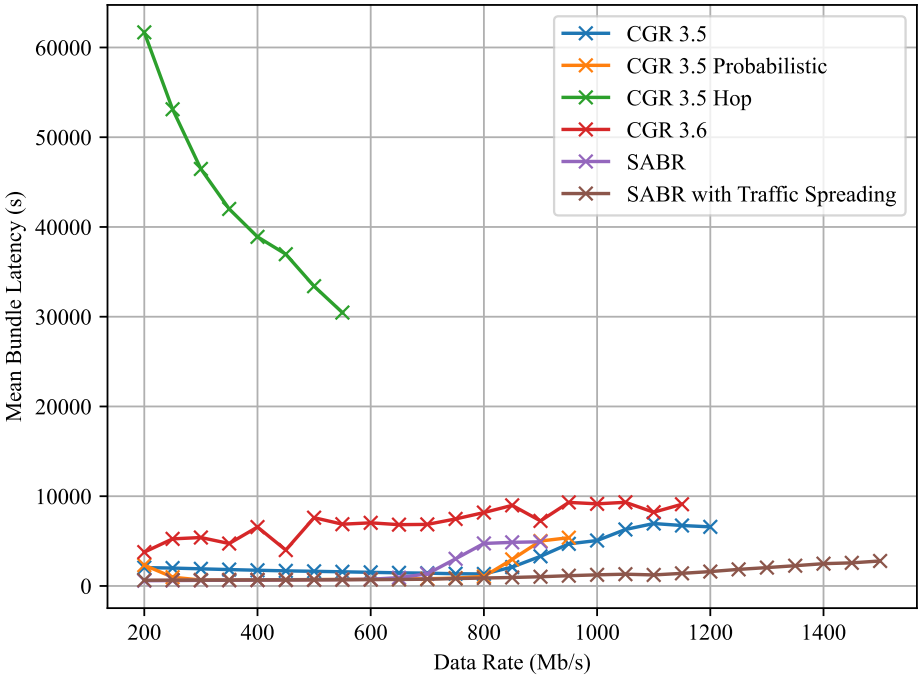
When the nested contact to relay 2 begins, bundles now have two concurrent routes to the MOC. The weighted random distribution of bundles between these routes results in a more efficient use of the overall network. Note that in this simplified scenario, a very small number of bundles are produced, resulting in a seemingly uneven distribution between the routes. In practice, when thousands of bundles are transmitted, the law of large numbers ensures that the quantity of bundles assigned to each route is proportional to the route's capacity.

## 5.5 PRIMARY SCENARIO

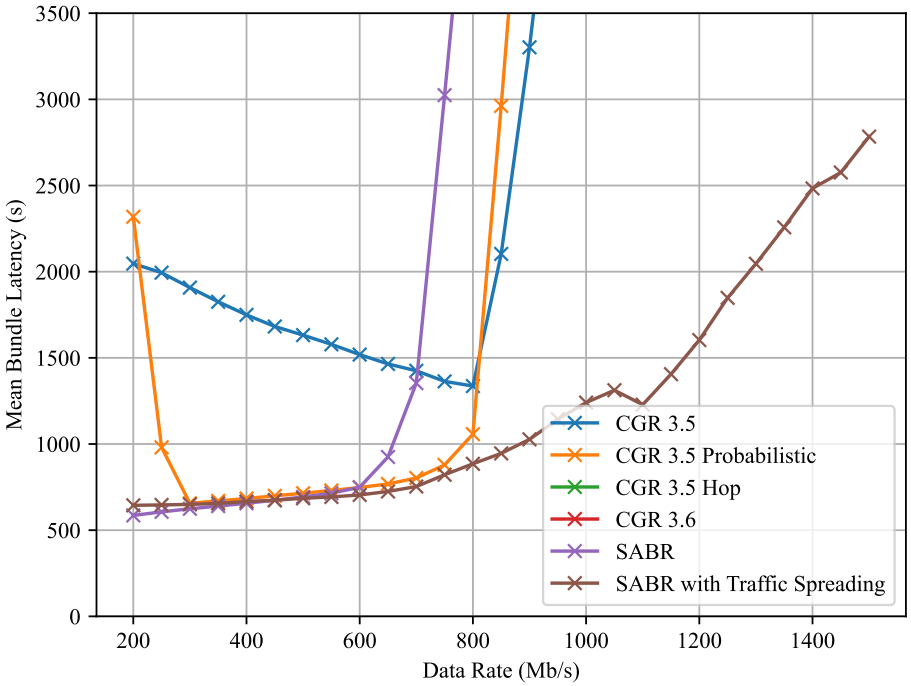
SABR with the traffic spreading enhancement was evaluated in the primary scenario introduced in Section 3.2. The scenario was modified to include a second data-producing Mars base with line of sight to the two areostationary orbiters. The exact location is insignificant to the investigation. The

algorithms' performances are evaluated in the downlink direction only. Bundles are generated at each Mars base at rates from 100 Mb/s to 750 Mb/s. Each node has an SDR capacity of 1 TB.

Figure 5.7 and Figure 5.8 reveal the latency and delivery ratio performance of the CGR algorithms. SABR and SABR with traffic spreading performed the best, achieving a combination of high delivery ratios and low latencies.




**(a)** Full perspective.




**(b)** Narrow perspective.

**Figure 5.7.** Mean bundle latency of CGR algorithms in the primary scenario, including SABR with traffic spreading.

CGR 3.5 has good delivery ratio performance, however its latency results are poor. CGR 3.5 Probabilistic arguably outperforms SABR, but it suffers high latencies at low data generation rates, making it

an unsuitable algorithm in this scenario. CGR 3.5 Probabilistic's bundle forwarding scheme considers arrival confidence in addition to arrival time and hop count, making the scheme less rigid than SABR. Its bundles are distributed more evenly than those of SABR, improving performance at certain data generation rates.

SABR with traffic spreading outperforms all, maintaining a flawless delivery ratio at data rates far higher than any other algorithm. Traffic spreading distributes bundles between nodes, avoiding a concentration of bundles at a small number of nodes, and therefore avoiding SDR saturation and subsequent bundle deletions.

The distribution of bundles between routes also reduces queuing delays, resulting in improved bundle latency performance. SABR with traffic spreading has the lowest mean latency of all the algorithms, except at low bundle generation rates, where standard SABR performs marginally better. This marginal difference can be attributed to the adjustable PBAT traffic spreading tolerance. In this scenario, the tolerance is set at 2%, allowing for the inclusion of routes which have a PBAT of up to 2% greater than the best route.

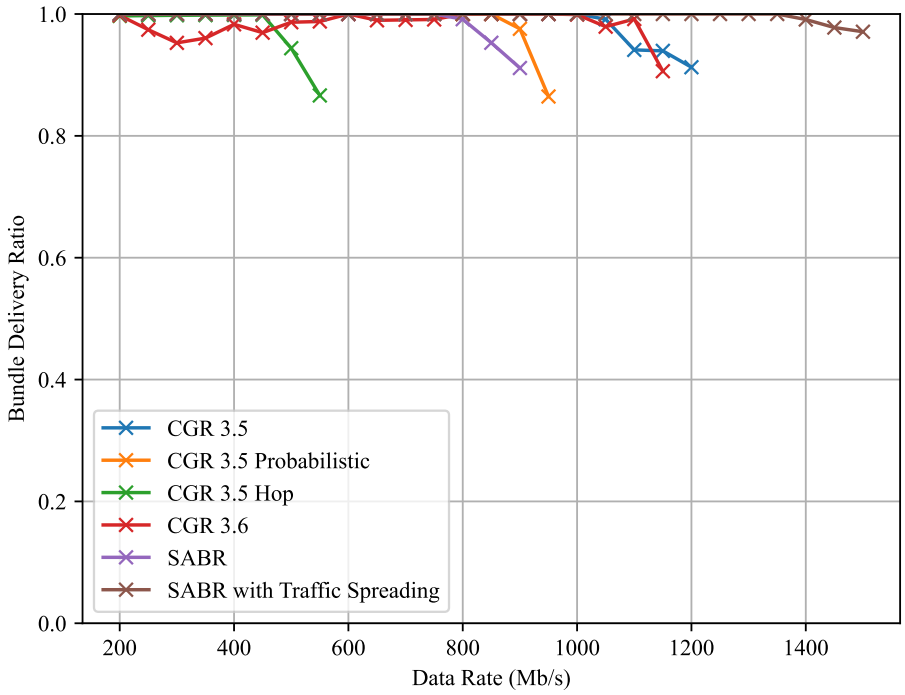



**Figure 5.8.** Bundle delivery ratio of CGR algorithms in the primary scenario, including SABR with traffic spreading.

## 5.6 SATELLITE NETWORK SCENARIO

SABR with traffic spreading was also evaluated in the satellite network scenarios introduced in Section 4.5. The effect of the enhancement was investigated in both the along-track and Walker satellite configurations.

### 5.6.1 Along-track formation

In the along-track formation, SABR with the traffic spreading enhancement performed well, maintaining low latencies and high bundle delivery ratios. The latency results, seen in Figure 5.9, reveal that SABR with traffic spreading caused slightly higher mean bundle latencies at lower data rates relative to standard SABR. This counterintuitive result is the consequence of traffic spreading's PBAT tolerance. At each hop, traffic spreading may select a route with a later delivery time than the best route, increasing the overall mean bundle latency. This consequence is most prominent at lower network loads, as the congestion-alleviating properties of traffic spreading are less effective. As the network load increased, SABR with traffic spreading maintained a low mean latency, eventually outperforming standard SABR.

Importantly, the bundle delivery ratio results in Figure 5.10 show that SABR with traffic spreading achieved the highest delivery ratio across all routing algorithms at all data generation rates.

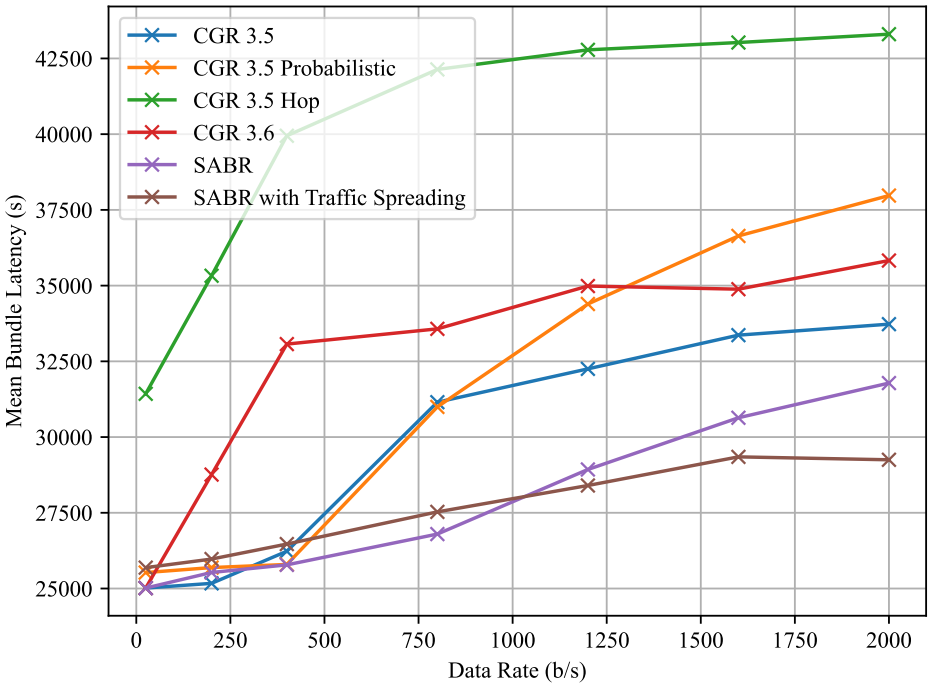



**Figure 5.9.** Mean bundle latency of CGR algorithms in the along-track satellite scenario, including SABR with traffic spreading.

The traffic spreading enchantment improved SABR performance in this scenario, particularly at high network loads. The along-track satellite train provides multiple similar paths from ground nodes to ground stations. Traffic spreading helps to distribute bundles evenly between nodes in the satellite train, reducing bundle concentration and congestion.

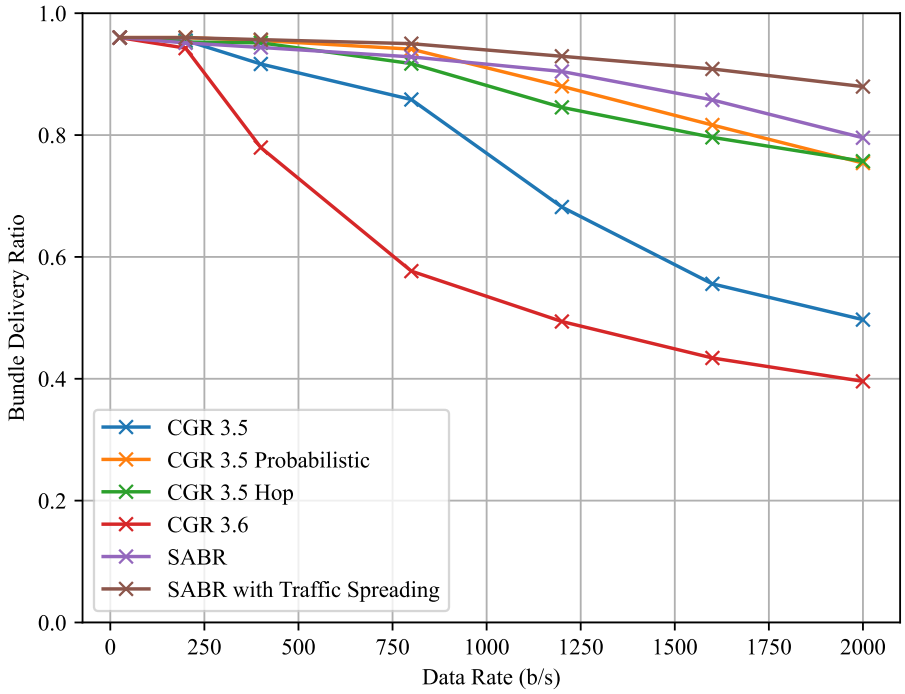



**Figure 5.10.** Bundle delivery ratio of CGR algorithms in the along-track satellite scenario, including SABR with traffic spreading.

### 5.6.2 Walker formation

In the Walker formation, the traffic spreading enhancement provided no significant performance benefit. The mean latency results in Figure 5.11 and delivery ratio results in Figure 5.12 reveal that SABR with traffic spreading performed similarly to CGR 3.5, CGR 3.6, and SABR. The Walker formation provides a very limited number of low-latency routes between ground nodes and ground targets. These top-performing algorithms all identify the same best route for each bundle. Traffic spreading provides no benefit in such a network since there is typically only a single best route, with no competing alternatives.

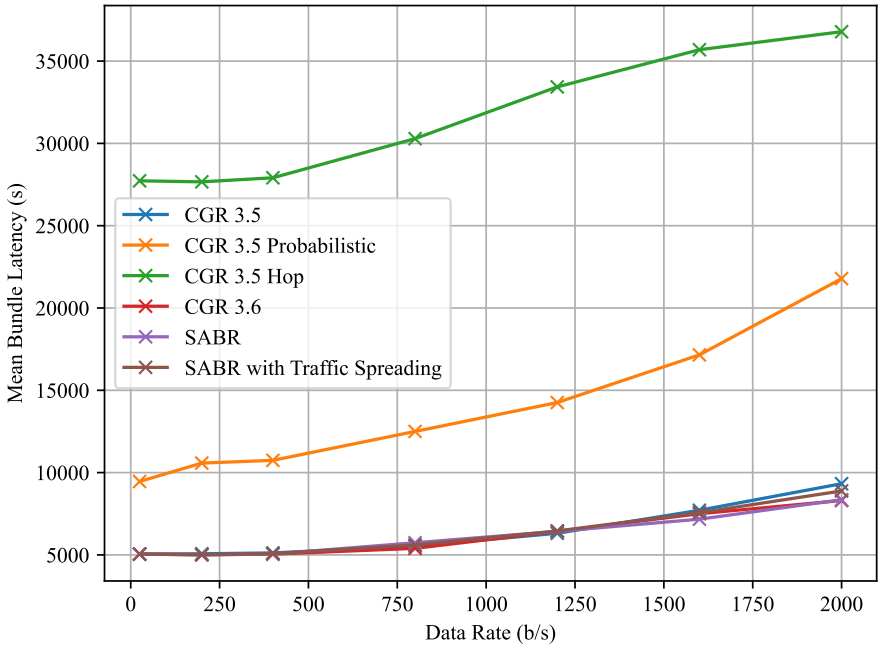



**Figure 5.11.** Mean bundle latency of CGR algorithms in the Walker satellite scenario, including SABR with traffic spreading.

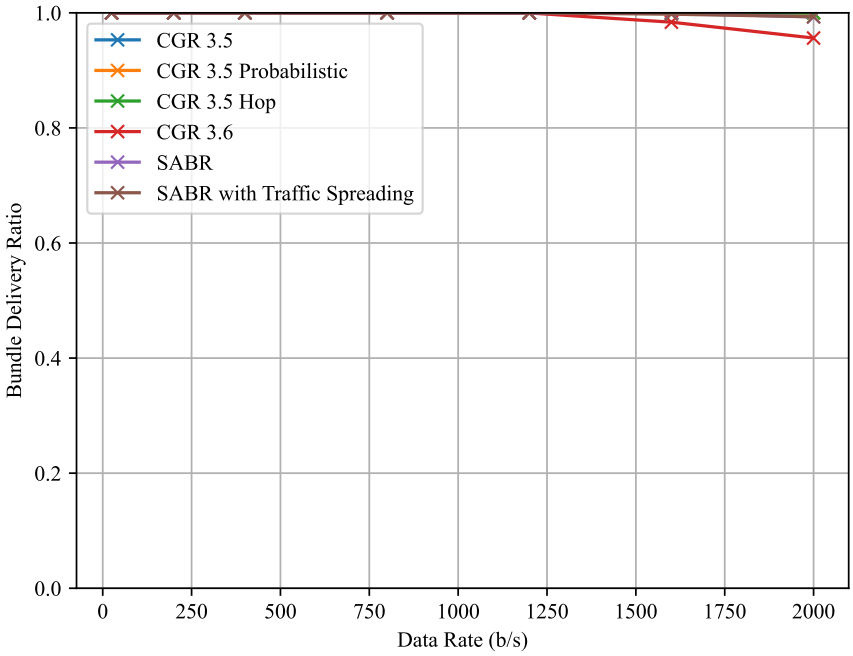



**Figure 5.12.** Bundle delivery ratio of CGR algorithms in the Walker satellite scenario, including SABR with traffic spreading.

## 5.7 CHAPTER SUMMARY

This chapter introduces an enhancement to the SABR protocol which improves performance in DTNs with competing candidate routes. It addresses a deficiency where SABR selects a single route for bundles, despite the availability of other routes, potentially leading to congestion. Through simulations, the chapter presents scenarios where multiple nodes independently transmit bundles along the same route, causing downstream bottlenecks. Traffic spreading aims to distribute bundles among similar candidate routes by modifying the SABR algorithm in both the forwarding and routing stages.

The modified forwarding stage begins with the identification of all routes with the earliest PBAT, forming a set of candidate routes. A weighted round-robin distribution scheme is employed to select a route for each bundle from the set of candidate routes. Route weights are assigned in proportion to the routes' available capacities. In the routing stage, modifications increase the number of candidate routes available for selection by computing routes that begin with each neighbour of the current node.

Traffic spreading is demonstrated in a simple reference scenario, followed by an evaluation in the primary Martian scenario and the satellite network scenarios. In the primary scenario, SABR with traffic spreading significantly outperforms other CGR algorithms, producing higher delivery ratios and lower latencies. In the satellite network scenarios, traffic spreading benefits along-track formations by evenly distributing bundles between nodes in the satellite train. However, in Walker formations with limited best-route alternatives, the enhancement's benefits are negligible.

Overall, traffic spreading proves to be a promising enhancement to SABR, improving performance by distributing bundles among competing routes to increase resource utilisation and alleviate congestion.

# CHAPTER 6 CONCLUSION

This dissertation explored the domain of interplanetary communication, concentrating on the communication requirements and routing algorithms pertinent to human activity on Mars. Through an examination of existing literature, coupled with computer simulations and analyses, the research aimed to address three research questions.

Firstly, the communication requirements for human activity on Mars were investigated. By examining proposed missions and anticipated infrastructure this study explored the bandwidth, latency, and uptime essential for effective communication in Martian environments. The study of literature formed a foundation for the development of realistic simulation scenarios.

Secondly, the dissertation tested various DTN routing algorithms to determine their efficacy in meeting the communication needs of Martian missions. The research methodology involved the construction of realistic space mission scenarios, implemented through the development of a simulation platform integrating physics and networking simulations. By analysing performance metrics such as delivery ratio and latency, the study provided quantitative evidence supporting the efficacy of specific routing protocols.

The study highlighted the suitability of CGR algorithms, particularly emphasising the performance of SABR, in space-based communication scenarios. SABR consistently outperformed other algorithms, exhibiting superior latency and delivery ratios across various network configurations.

Lastly, the research explored the prospect of enhancing existing DTN routing algorithms to better align with the challenges of Martian communication networks. This endeavour led to the development of “traffic spreading,” an enhancement to the SABR protocol aimed at mitigating congestion and improving resource utilisation in environments with competing candidate routes. SABR with the “traffic spreading” enhancement improved performance relative to standard SABR (and other CGR algorithms) by distributing bundles between parallel routes, reducing potential downstream congestion.

In conclusion, this dissertation contributes to the understanding of communication requirements and

routing algorithms relevant to human activity on Mars. Through simulations and analyses, the most suitable DTN routing algorithms were identified. A deficiency of SABR was addressed with the introduction of the "traffic spreading" enhancement. The findings underscore the critical importance of resource-efficient communication protocols in enabling successful crewed missions to Mars and contribute to the greater goal of advancing interplanetary communication technologies.

## 6.1 FUTURE WORK

The following topics have been identified as avenues for further research:

- This dissertation employed the FFA to determine the maximum throughput of DTNs. To use the FFA on DTNs, the intermittently connected networks had to first be simplified to permanently connected networks, reducing the accuracy and utility of the maximum throughput calculations. Future work is recommended to explore more accurate maximum throughput computations for intermittently connected networks, thereby providing a better benchmark for DTN routing algorithm performance analysis.
- PRoPHET and Binary Spray and Wait, two replication-based performance algorithms, exhibited promising performance in certain Martian scenario simulations. Simulation runtime constraints prohibited the rigorous evaluation of these algorithms. It is recommended that future work be conducted to comprehensively investigate the performance of PRoPHET and Binary Spray and Wait in the context of interplanetary communication. This study would assess whether these replication-based algorithms can match the performance of CGR algorithms.
- This dissertation presented "traffic spreading" as a general solution to SABR's suboptimal route selection when presented with multiple candidate routes. An alternative solution, based on node coordination, may also prove to be effective. Coordination between nearby data-producing nodes (such as Mars bases), could prevent route contention. Nodes would attempt to reach a consensus on bundle distribution between all available routes, thereby mitigating potential downstream congestion. Future research could develop a novel coordination scheme that is integrated with existing routing protocols.

# REFERENCES

[1] R. Zubrin, *Case for Mars*. New York, NY, USA: Simon and Schuster, Jun. 2011.

[2] G. Denis, D. Alary, X. Pasco, N. Pisot, D. Texier, and S. Toulza, "From new space to big space: How commercial space dream is becoming a reality," *Acta Astronautica*, vol. 166, pp. 431–443, Jan. 2020.

[3] A. McMahon and S. Farrell, "Delay- and Disruption-Tolerant Networking," *IEEE Internet Computing*, vol. 13, no. 6, pp. 82–87, Nov. 2009.

[4] S. Burleigh *et al.*, "Delay-tolerant networking: an approach to interplanetary Internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, Jun. 2003.

[5] G. Kazz, S. Burleigh, K.-M. Cheung, and B. Shah, "Evolution of the Mars Relay Network End-to-End Information System in the Mars Human Era (2030-2040)," in *SpaceOps 2016 Conference*, Daejeon, Korea, 16-20 May 2016.

[6] S. May, "The Space Network: Cell Towers for Astronauts," Sep. 2018. [Online]. Available: http://www.nasa.gov/audience/foreducators/stem-on-station/downlinks-scan.html

[7] K. Fall and S. Farrell, "DTN: an architectural retrospective," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 828–836, Jun. 2008.

[8] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," *Technical Report CS-200006, Duke University*, p. 14, Jun. 2000.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, Philadelphia, PA, USA, 26 Aug. 2005, pp. 252–259.

[10] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, Jul. 2003.

[11] S. Burleigh, "Contact Graph Routing," Internet Engineering Task Force, Internet Draft draft-burleigh-dtnrg-cgr-01, Jul. 2010. [Online]. Available: https://datatracker.ietf.org/doc/draft-burleigh-dtnrg-cgr

[12] T. Abdelkader, K. Naik, A. Nayak, N. Goel, and V. Srivastava, "A performance comparison of delay-tolerant network routing protocols," *IEEE Network*, vol. 30, no. 2, pp. 46–53, Mar. 2016.

[13] J. A. Fraire *et al.*, "Assessing Contact Graph Routing Performance and Reliability in Distributed Satellite Constellations," *Journal of Computer Networks and Communications*, vol. 2017, pp. 1–18, 2017.

[14] J. Segui, E. Jennings, and S. Burleigh, "Enhancing Contact Graph Routing for Delay Tolerant Space Networking," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, Houston, TX, USA, 5-9 Dec. 2011, pp. 1–6.

[15] J. A. Fraire, P. Madoery, F. Raverta, J. M. Finochietto, and R. Velazco, "DtnSim: Bridging the Gap between Simulation and Implementation of Space-Terrestrial DTNs," in *2017 6th International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, Henares, Madrid, Spain, 27-29 Sep. 2017, pp. 120–123.

[16] A. Varga, "Using the OMNeT++ discrete event simulation system in education," *IEEE Transactions on Education*, vol. 42, no. 4, p. 11, Nov. 1999.

[17] R. Lent, "Evaluation of Cognitive Routing for the Interplanetary Internet," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Taipei, Taiwan, 7-11 Dec. 2020, pp. 1–6.

[18] T. Iida, Y. Arimoto, and Y. Suzuki, "Earth-Mars communication system for future Mars human community: A story of high speed needs beyond explorations," *IEEE Aerospace and Electronic Systems Magazine*, vol. 26, no. 2, pp. 19–25, Feb. 2011.

[19] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald, "When TCP Breaks: Delay- and Disruption- Tolerant Networking," *IEEE Internet Computing*, vol. 10, no. 4, pp. 72–78, Jul. 2006.

[20] A. Könsgen and A. Förster, "Current state and future challenges in deep space communication: A survey," *it - Information Technology*, vol. 63, no. 4, pp. 219–234, Sep. 2021.

[21] D. Morabito and R. Hastrup, "Communicating with Mars during periods of solar conjunction," in *IEEE Aerospace Conference Proceedings*, vol. 3, Big Sky, MT, USA, 9-16 Mar. 2002, pp. 3–3.

[22] A. Alhilal, T. Braud, and P. Hui, "The Sky is NOT the Limit Anymore: Future Architecture of the Interplanetary Internet," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 8, pp. 22–32, Aug. 2019.

[23] K.-S. Wong and T.-C. Wan, "Current State of Multicast Routing Protocols for Disruption Tolerant Networks: Survey and Open Issues," *Electronics*, vol. 8, no. 2, p. 162, Feb. 2019.

[24] C. Caini, H. Cruickshank, S. Farrell, and M. Marchese, "Delay- and Disruption-Tolerant Networking (DTN): An Alternative Solution for Future Satellite Networking Applications," *Proceedings of the IEEE*, vol. 99, no. 11, pp. 1980–1997, Nov. 2011.

[25] NASA, "Delay / Disruption Tolerant Networking animation." Jun. 2016. [Online]. Available: https://commons.wikimedia.org/wiki/File:Delay-Disruption_Tolerant_Networking_(DTN).webm

[26] G. Araniti *et al.*, "Contact graph routing in DTN space networks: overview, enhancements and performance," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 38–46, Mar. 2015.

[27] J. Mukherjee and B. Ramamurthy, "Communication Technologies and Architectures for Space Network and Interplanetary Internet," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 881–897, 2013.

[28] P. Godha, S. Jadon, A. Patle, I. Gupta, B. Sharma, and A. Kumar Singh, "Architecture, an Efficient Routing, Applications, and Challenges in Delay Tolerant Network," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, 15-17 May 2019, pp. 824–829.

[29] K. Scott and S. Burleigh, "Bundle Protocol Specification," Internet Engineering Task Force, Request for Comments RFC 5050, Nov. 2007. [Online]. Available: https://datatracker.ietf.org/doc/rfc5050

[30] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, Karlsruhe Germany, 25-29 Aug. 2003, pp. 27–34.

[31] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys Tutorials*, vol. 8, no. 1, pp. 24–37, 2006.

[32] N. Bezirgiannidis, C. Caini, and V. Tsaoussidis, "Analysis of contact graph routing enhancements for DTN space communications," *International Journal of Satellite Communications and Networking*, vol. 34, no. 5, pp. 695–709, Sep. 2016.

[33] P. G. Madoery, J. A. Fraire, and J. M. Finochietto, "Congestion management techniques for disruption-tolerant satellite networks," *International Journal of Satellite Communications and Networking*, vol. 36, no. 2, pp. 165–178, 2018.

[34] N. Bezirgiannidis, C. Caini, D. Padalino Montenero, M. Ruggieri, and V. Tsaoussidis, "Contact Graph Routing enhancements for delay tolerant space communications," in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, Livorno, Italy, 8-10 Sep. 2014, pp. 17–23.

[35] J. A. Fraire, O. De Jonckère, and S. Burleigh, "Routing in the Space Internet: A contact graph routing tutorial," *Journal of Network and Computer Applications*, vol. 174, p. 102884, Jan. 2021.

[36] CCSDS, "Schedule-Aware Bundle Routing, Recommended Standard," Washington DC, USA, 2019. [Online]. Available: https://public.ccsds.org/Pubs/734x3b1.pdf

[37] J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[38] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[39] B. G. Drake, S. J. Hoffman, and D. W. Beaty, "Human exploration of Mars, Design Reference Architecture 5.0," in *2010 IEEE Aerospace Conference*, Big Sky, MT, USA, 6-13 Mar. 2010, pp. 1–24.

[40] J. C. Crusan, D. A. Craig, and N. B. Herrmann, "NASA's deep space habitation strategy," in *2017 IEEE Aerospace Conference*, Big Sky, MT, USA, 4-11 Mar. 2017, pp. 1–11.

[41] T. Cichan *et al.*, "Mars Base Camp: An Architecture for Sending Humans to Mars," *New Space*, vol. 5, no. 4, pp. 203–218, Dec. 2017.

[42] F. Chang Díaz, J. Carr, L. Johnson, W. Johnson, G. Genta, and P. F. Maffione, "Solar electric propulsion for human Mars missions," *Acta Astronautica*, vol. 160, pp. 183–194, Jul. 2019.

[43] J. L. Heldmann *et al.*, "Mission Architecture Using the SpaceX Starship Vehicle to Enable a Sustained Human Presence on Mars," *New Space*, vol. 10, no. 3, pp. 259–273, Sep. 2022.

[44] E. Musk, "Making Life Multi-Planetary," *New Space*, vol. 6, no. 1, pp. 2–11, Mar. 2018.

[45] C. Edwards, W. Banerdt, D. Beaty, L. Tamppari, and R. Zurek, "Relay Orbiters for Enhancing and Enabling Mars In Situ Exploration," Sep. 2009. [Online]. Available: https://mepag.jpl.nasa.gov/reports/decadal/CharlesDEdwards.pdf

[46] R. E. Gladden, C. H. Lee, C. D. Edwards, M. A. Viotti, and R. M. Davis, "A Dedicated Relay Network to Enable the Future of Mars Exploration," in *2021 IEEE Aerospace Conference*, Big Sky, MT, USA, 6-13 Mar. 2021, pp. 1–14.

[47] W. S. Tai, D. S. Abraham, and K.-M. Cheung, "Mars Planetary Network for Human Exploration Era – Potential Challenges and Solutions," in *The 15th International Conference on Space Operations (SpaceOps)*, Marseille, France, 28 May - 1 Jun. 2018.

[48] J. Breidenthal, M. Jesick, H. Xie, and C.-W. Lau, "Deep Space Relay Terminals for Mars Superior Conjunction," in *The 15th International Conference on Space Operations (SpaceOps)*, Marseille, France, 28 May - 1 Jun. 2018.

[49] R. Gladden *et al.*, "Preparing the Mars Relay Network for the Arrival of the Perseverance Rover at Mars," in *2022 IEEE Aerospace Conference (AERO)*, 5-12 Mar. 2022, pp. 01–19.

[50] L. Gu, L. Yu, W. Li, and K. Zhao, "A publish-subscribe networking architecture for future manned deep space exploration," *China Communications*, vol. 17, no. 7, pp. 38–51, Jul. 2020.

[51] G. Noreen *et al.*, "Integrated network architecture for sustained human and robotic exploration," in *IEEE Aerospace Conference Proceedings*, vol. 2005, Big Sky, MT, USA, 5-12 Mar. 2005, pp. 1266–1285.

[52] S. Burleigh *et al.*, "The interplanetary internet: A communications infrastructure for Mars exploration," *Acta Astronautica*, vol. 53, no. 4-10, pp. 365–373, Aug. 2003.

[53] R. L. Howard, "Lagrange-Based Options for Relay Satellites to Eliminate Earth-Mars Communications Outages During Solar Superior Conjunctions," in *ASCEND 2020*, Virtual Event, 16-18 Nov. 2020.

[54] B. Du, F. Gao, and J. Xu, "The analysis of topology based on Lagrange points L4/L5 of Sun-Earth system for relaying in Earth and Mars communication," in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, Guangzhou, China, 6-8 May 2017, pp. 533–537.

[55] T. Gangale, “MarsSat: Assured Communication with Mars,” *Annals of the New York Academy of Sciences*, vol. 1065, no. 1, pp. 296–310, 2005.

[56] P. Wan and Y. Zhan, “A structured Solar System satellite relay constellation network topology design for Earth-Mars deep space communications,” *International Journal of Satellite Communications and Networking*, vol. 37, no. 3, pp. 292–313, 2019.

[57] E. Koktas and E. Basar, “Communications for the Planet Mars: Past, Present, and Future,” *arXiv*, Nov. 2022.

[58] F. Miranda, “Antenna Technologies for Future NASA Exploration Missions,” in *2006 IEEE International Workshop on Antenna Technology Small Antennas and Novel Metamaterials*, White Plains, NY, USA, 6-8 Mar. 2006, pp. 7–10.

[59] H. Hemmati, A. Biswas, and I. B. Djordjevic, “Deep-Space Optical Communications: Future Perspectives and Applications,” *Proceedings of the IEEE*, vol. 99, no. 11, pp. 2020–2039, Nov. 2011.

[60] W. D. Williams *et al.*, “RF and Optical Communications: A Comparison of High Data Rate Returns From Deep Space in the 2020 Timeframe,” in *12th Ka and Broadband Communications Conference*, Naples, Italy, 27-29 Sep. 2020.

[61] C. Botella, R. Baños, E. Etchemendy, A. Garcia-Palacios, and M. Alcañiz Raya, “Psychological countermeasures in manned space missions: “EARTH” system for the Mars-500 project,” *Computers in Human Behavior*, vol. 55, pp. 898–908, Feb. 2016.

[62] I. B. Ushakov *et al.*, “Main findings of psychophysiological studies in the Mars 500 experiment,” *Herald of the Russian Academy of Sciences*, vol. 84, no. 2, pp. 106–114, Mar. 2014.

[63] E. Hand, “NASA picks Mars landing site,” *Nature*, vol. 475, no. 7357, pp. 433–433, Jul. 2011.

[64] C. D. Edwards, P. R. Barela, R. E. Gladden, C. H. Lee, and R. De Paula, “Replenishing the Mars relay network,” in *2014 IEEE Aerospace Conference*, Big Sky, MT, USA, 1-8 Mar. 2014, pp. 1–13.

[65] "SimonOrJ", “Field of view of the Deep Space Network antennas, looking down from the North Pole.” Jun. 2013. [Online]. Available: https://commons.wikimedia.org/wiki/File:DSNantenna.svg

[66] M. Browne Mwakyanjala, P. Hyvönen, J. de Oliveira, and J. van de Beek, "Feasibility of using a software-defined baseband for multiple space per aperture (MSPA) ground operations," *International Journal of Satellite Communications and Networking*, vol. 40, no. 2, pp. 133–147, 2022.

[67] D. R. Jackson, C. Caloz, and T. Itoh, "Leaky-Wave Antennas," *Proceedings of the IEEE*, vol. 100, no. 7, pp. 2194–2206, Jul. 2012.

[68] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Densmore, and R. Patra, "Implementing Delay Tolerant Networking," *Intel Research Berkeley Technical Report IRB-TR-04-020*, Dec. 2004.

[69] S. Burleigh, "Interplanetary Overlay Network: An Implementation of the DTN Bundle Protocol," in *2007 4th IEEE Consumer Communications and Networking Conference*, Washington DC, USA, 11-13 Jan. 2007, pp. 222–226.

[70] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the Second International ICST Conference on Simulation Tools and Techniques*, Rome, Italy, 2-6 Mar. 2009.

[71] L. R. Ford and D. R. Fulkerson, "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem," *Canadian Journal of Mathematics*, vol. 9, pp. 210–218, Jan. 1957.

[72] J. Edmonds and R. M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *Journal of the ACM*, vol. 19, no. 2, pp. 248–264, Apr. 1972.

[73] P. G. Madoery, F. D. Raverta, J. A. Fraire, and J. M. Finochietto, "Routing in Space Delay Tolerant Networks under Uncertain Contact Plans," in *2018 IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, 20-24 May 2018, pp. 1–6.

[74] J. A. Fraire, P. G. Madoery, A. Charif, and J. M. Finochietto, "On route table computation strategies in Delay-Tolerant Satellite Networks," *Ad Hoc Networks*, vol. 80, pp. 31–40, Nov. 2018.

[75] C. Caini, G. M. De Cola, and L. Persampieri, "Schedule-Aware Bundle Routing: Analysis and enhancements," *International Journal of Satellite Communications and Networking*, vol. 39, no. 3, pp. 237–249, Jun. 2021.

[76] J. He and J. Rexford, "Toward internet-wide multipath routing," *IEEE Network*, vol. 22, no. 2, pp. 16–21, Mar. 2008.

[77] Cisco, "Understand how Load Balancing Works," Dec. 2022. [Online]. Available: https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5212-46.html

[78] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," Internet Engineering Task Force, Request for Comments RFC 2992, Nov. 2000. [Online]. Available: https://datatracker.ietf.org/doc/rfc2992

[79] Palo Alto Networks, "ECMP Load-Balancing Algorithms," Nov. 2023. [Online]. Available: https://docs.paloaltonetworks.com/pan-os/10-1/pan-os-networking-admin/ecmp/ecmp-load-balancing-algorithms

[80] N. Alessi, C. Caini, T. de Cola, S. Martin, and J. P. Mayer, "DTN performance analysis of multi-asset Mars-Earth communications," *International Journal of Satellite Communications and Networking*, vol. 40, no. 1, pp. 11–26, 2022.

# APPENDIX A YEN'S ALGORITHM

Yen's algorithm [37] is a method used to find the $K$ shortest, loopless paths in a graph from a source node to a destination node. For $K = 1$, the algorithm operates identically to Dijkstra's shortest path algorithm [38]. For $K = 2$, the algorithm returns the shortest path and the second shortest path between a source node and a destination node.

The concept underlying Yen's algorithm is to iteratively find the shortest path between two nodes and remove edges along that path to explore alternative paths. By repeating this process, Yen's algorithm discovers multiple distinct shortest paths between the same pair of nodes.

The algorithm maintains two lists. List $A$ is the list of shortest paths and list $B$ is a list of candidate shortest paths. First, the shortest path from source to destination is found using Dijkstra's algorithm and appended to $A$. Each path in $A$ is evaluated, with each node in each path becoming a spur node. A spur node is created by removing its outgoing edge from the graph. The shortest path from the spur node to the destination is then computed and concatenated with the root path (the path from the source node to the spur node). This new path is appended to $B$. After each node in the original path is made a spur node, the shortest path in $B$ is appended to $A$. The process is repeated until $A$ is populated with $K$ routes.

Pseudo-code is presented in Algorithm 1.

**Algorithm 1** Yen's Algorithm

```
A[0] ← Dijkstra(graph, source, destination)
for k = 1 to K do
    B ← []
    for i = 0 to length(A[k − 1]) − 2 do
        spurNode ← A[k − 1][i]
        rootPath ← A[k − 1][0 : i]
        graphCopy ← graph
        for all path in A do
            if path starts with rootPath then
                graphCopy.removeEdge(path[i], path[i + 1])
            end if
        end for
        spurPath ← Dijkstra(graphCopy, spurNode, destination)
        candidatePath ← concatenate(rootPath, spurPath)
        if candidatePath not in B then
            B.append(candidatePath)
        end if
    end for
    A.append(shortestPath(B))
end for
return A
```

# APPENDIX B

## B.1 EXAMPLE OMNET++ SIMULATION CONFIGURATION FILE

```
[General]
user-interface = Qtenv
network = src.dtnsim
repeat = 1

# Save the results in sqlite format
**.vector-recording = false
outputscalarmanager-class = "omnetpp::envir::SqliteOutputScalarManager"

result-dir = results/run1

# Node ids are consecutive and start from 1 up to nodesNumber. They must match
    the node ids in the contact plan
dtnsim.nodesNumber = 12

# Node icons (see Omnet++ images folder)
dtnsim.node[1].icon = "server"  # MOC
dtnsim.node[2..4].icon = "receiverdish"  # DSN stations
dtnsim.node[5].icon = "receiverdish_l"  # Mars base
dtnsim.node[6..7].icon = "satellite"  # Areostationary orbiters
dtnsim.node[8].icon = "satellite_l"  # L5_Realy
dtnsim.node[9..12].icon = "satellite_s"  # MRN orbiters

# Use custom postions set in src/node/graphics/Graphics.cc
dtnsim.node[1..12].useCustomPositions = true

# Set SDR size
dtnsim.node[*].dtn.sdrSize = 1000000 # MB = 1 TB
```

```
28
29 # Contact plan
30 dtnsim.node[*].dtn.printRoutingDebug = false
31 dtnsim.central.contactsFile= "contactPlan/contact_plan.txt"
32
33 # Generate traffic
34 include traffic/100Mbps.ini
35
36 # Routing protocol
37 [Config uniboCgr]
38 dtnsim.node[*].dtn.routing = "uniboCgr"
```

## B.2 EXAMPLE DTNSIM SIMULATION TRAFFIC FILE

```
1  # Traffic generation: MOC to Mars base
2  dtnsim.node[1].app.enable=true
3  dtnsim.node[1].app.returnToSender=false
4  dtnsim.node[1].app.bundlesNumber="16432,"
5  dtnsim.node[1].app.start="259200,"
6  dtnsim.node[1].app.interval=5.258033
7  dtnsim.node[1].app.destinationEid="5,"
8  dtnsim.node[1].app.size="65.726707,"
9  # Traffic generation: Mars base to MOC
10 dtnsim.node[5].app.enable=true
11 dtnsim.node[5].app.returnToSender=false
12 dtnsim.node[5].app.bundlesNumber="16432,"
13 dtnsim.node[5].app.start="259200,"
14 dtnsim.node[5].app.interval=5.258033
15 dtnsim.node[5].app.destinationEid="1,"
16 dtnsim.node[5].app.size="65.726707,"
```

# APPENDIX C EDMONDS–KARP ALGORITHM

The Edmonds–Karp algorithm [72] is a near-direct implementation of the FFA which calculates the maximum flow between nodes in a graph. $G$ is a graph, consisting of directed edges between vertices. $c(u,v)$ is the capacity of an edge from $u$ to $v$, and $f(u,v)$ is the edge's current flow. The algorithm finds the maximum possible flow from source $s$ to destination $d$ where flow does not exceed the capacity of any edge. Initially, $f(u,v)$ is set to 0.

$G_f$ is the residual graph, which contains all of $G$'s edges $(u,v)$, along with fictitious reverse edges $(v,u)$, whose capacities are set to 0. These residual edges assist with finding the entire set of augmenting paths through the network that results in maximum flow.

An augmenting path is a path of edges from source to destination through $G_f$ where the unused capacity in each edge is greater than 0. Each augmenting path has a bottleneck value equal to the smallest available capacity of all the edges in the path. When an augmenting path is found, $G_f$ is augmented by updating the available capacity of the path's edges with this bottleneck value (increasing the flow of forward edges and decreasing the flow of residual edges). Augmented paths are repeatedly found until no augmenting paths remain between source and destination. The maximum flow is thus found and is equal to the sum of the flow of all edges that terminate at the destination. It is also equal to the sum of the bottleneck values of all augmenting paths.

The algorithm's pseudo-code is presented in Algorithm 2, returning the maximum flow $f_{max}$ through $G$ from $s$ to $d$ where $p$ is a path from $s$ to $d$. $BFS()$ is a breadth-first search algorithm used to explore the residual graph. The use of a breadth-first search algorithm distinguishes the Edmonds–Karp algorithm from the FFA.

**Algorithm 2** Edmonds–Karp Algorithm

1: $f_{max} \leftarrow 0$
2: **for all** edges $(u,v) \in G$ **do**
3: $c(u,v) \leftarrow$ `edge capacity`
4: $f(u,v) \leftarrow 0$
5: **end for**
6: **for all** edges $(v,u) \in G_f$ **do**
7: $c(v,u) \leftarrow 0$
8: $f(v,u) \leftarrow 0$
9: **end for**
10: **while** $p \leftarrow BFS(G_f)$ **do**
11: $c(p) \leftarrow \min(c(u,v) : (u,v) \in p)$
12: $f_{max} \leftarrow f_{max} + c(p)$
13: **for all** edges $(u,v) \in p$ **do**
14: $f(u,v) \leftarrow f(u,v) + c(p)$
15: $f(v,u) \leftarrow f(v,u) - c(p)$
16: **end for**
17: **end while**
18: **return** $f_{max}$