COMP4034 - Advanced Computer Networks Coursework

Callum Gooding

November 2020

Contents

Acronyms and Abbreviations

API Application Programming Interface 7, 9

DARPA Defense Advanced Research Projects Agency 4

DTN Delay Tolerant Network 3, 4, 16, 17

IPN Interplanetary Internet 4, 5, 16, 17

JPL Jet Propulsion Laboratory 4, 9

LLCD Lunar Laser Communications Demonstration 11

MANET Mobile Ad Hoc Network 4

OCW ONE Configuration Writer 7–9

ONE Opportunistic Network Environment 3, 4, 7–9, 11

oppnet opportunistic network 3, 16

OSS ONE Space Scenario 9

SV Summary Vector 5

Chapter 1

Introduction

1.1 Opportunistic Networks

Opportunistic networks (oppnets) are broadly defined by several main features:

- Multiple heterogeneous wirelessly connected nodes [21]
- A network topology that changes both temporally and spatially
- Each node only has the ability to communicate with nodes to which it is directly connected
- Network-wide communication is only possible with message passing, using the store, carry and forward paradigm [5]
- Nodes acting mainly as both end users and infrastructure of the network, in comparison to traditional networks where infrastructure elements such as switches and routers are largely distinct from the sources and destinations of data, such as mobile phones and computers [5]

A main advantage of oppnets is their lack of dependence on centralised infrastructure. This greatly increases the spatial range of any given node as it needs only to be within range of another node within the network. This is contrary to traditional networks where a node must be in range of an often static infrastructure element.

Diversity of nodes within an oppnet can also be leveraged to provide greater value than the sum of each individual element. In their article [21], Lilien et al. describe a hypothetical situation in which an earthquake strikes a city and nodes with differing capabilities such as imaging, vehicle database querying, vehicle remote diagnostics, and responder scheduling to coordinate a rescue operation.

By their nature, the nodes of an oppnet are often mobile and hence are subject to a variety of restrictions such as mass, size and cost. These restrictions often impose limits on the capabilities of each node. For example a mobile phone, one of the most common types of node in a modern day oppnet, may suffer from a smaller buffer size, less compute capability and a limited untethered battery life. Most crucially however, the range of each node is often the most limiting factor for a node. A typical cell tower can have a range of up to 150km [24] whereas peer to peer WiFi (WiFi Direct™), with a maximum range of 200 feet, and Bluetooth, which can only reach up to 30 feet [11].

1.2 The ONE Simulator

The Opportunistic Network Environment (ONE) Simulator [20] is a discreetly time-sliced event simulator that models agents, or nodes, in a Delay Tolerant Network (DTN). The simulator models node movement, inter-node contacts, routing and message handling and allows for reports to be collected to study various metrics such as delivery probability and round trip time [19].

```
## Scenario settings
Scenario.name = default_scenario
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1
# 43200s == 12h
Scenario.endTime = 43200
```

Listing 1: ONE Simulator configuration file snippet defining properties of the scenario

Simulation scenarios can be defined by configurations files. These files take the form of Java properties files [27]. These files modify the values of classes within the simulator and can also define which modules are used. Listing 1 shows a section of the default settings that define the property values for the scenario.

The ONE simulator is open source software. This means it is easy to implement new behaviours by simply extending an abstract module class and defining new implementations.

1.3 Interplanetary Internet

Four days before space shuttle Columbia broke apart over Texas in February 2003, the CANDOS [37] experiment successfully demonstrated that generalised extraterrestrial network communication was possible by sending a file between the shuttle and NASA's Goddard Space Center [17]. Since then, the need for a generic communications backbone within space has become evermore apparent.

In the spring of 1998 [7], Vint Cerf, one of the two developers of the TCP/IP suite [6], and others at NASA's Jet Propulsion Laboratory (JPL), began developing a new suite of protocols designed with the lack of predictability among node connections in mind. This research, along with earlier research into Mobile Ad Hoc Networks (MANETs) sponsored by Defense Advanced Research Projects Agency (DARPA) led to advances in generalised DTNs such as a 2003 SIGCOMM paper [10] outlining the motivations behind adopting a delay tolerant architecture as a solution for "Challenged Internets".

Efforts into creating specialised DTNs have since been grouped under the term Interplanetary Internet (IPN).

Chapter 2

Protocols

This paper uses three naive routing protocols to asses their usefulness in the IPN: Epidemic [34], Spray and Wait [29] and PRoPHET [22]. This chapter will provide a brief outline of these protocols.

2.1 Epidemic

The Epidemic routing protocol promises eventual message delivery but makes no "intelligent" choices and takes no notice of host metadata such as previous contacts or buffer capacity. The protocol is simple in nature due to it's lack of assumptions of network topology but this simplicity often leads to large scale congestion when the hosts' buffers fill up.

Each host maintains a buffer of messages that it can then forward. These messages either originate from the node or have been forwarded by other nodes. This buffer is arranged as a hash table keyed by the ID of each message. As well as this buffer, each node also maintains a Summary Vector (SV) of bits, each of which are set to 1 if the corresponding entry in the hash table is occupied, and 0 otherwise. This then means that the length of the SV is equal to the amount of available spaces in the hash table.

When two nodes A and B establish a connection, an anti-entropy [9] session is started. Node A first transmits its SV, SV_A, to node B. Node B then calculates $SV_A \wedge \neg SV_B$, determining the messages that A has and B needs. This vector is then transmitted back to node A and the requested messages are then sent back to node B.

To avoid unnecessary connections between hosts, a table of recently contacted hosts can also be maintained such that a host may refuse an anti-entropy session with another if one has taken place within a configurable time period. Additionally, other policies may be implemented when a host receives its requested messages. As it is under no obligation to accept these messages, a host may have a policy of ignoring messages over a certain size, or those that have undergone an amount of hops, or those of a certain age.

2.2 Spray and Wait

A key benefit of epidemic routing is the speed with which a message is disseminated among nodes. This ensures a much higher delivery probability than simply using direct transmission. However, this mass dissemination gives rise to congestion, clogging up the buffers of transient nodes and preventing them from contributing to the overall message delivery probability.

Spray and Wait offers a hybrid approach. This approach consists of two phases:

1. Spray phase: during this phase, L copies of a message are given to L relays
2. Wait phase: if the message does not reach its destination during the spray phase, only direct transmission of the message is allowed

With no limiting heuristics, the spray phase is identical to epidemic routing. However, policies determining the exact nature of the spray phase can serve to increase delivery probability. For example, Source Spray and Wait allows only the source node to directly disseminate to other nodes during the spray phase.

A better alternative to this is Binary Spray and Wait. This method dictates that at any given node with n copies of a message, $n/2$ copies are given to any connecting node and the rest kept for the initial node. If during the spray phase any given node has only one copy, it spreads using direct transmission.

In their comparison of Spray and Wait and other routing protocols [29], Spray and Wait routing achieves a 1.4 - 2.2 times faster delivery than all assessed protocols except for Epidemic under low traffic and 1.8 - 3.3 times faster delivery than all assessed protocols in high traffic. This comparison also highlights the tenancy for Epidemic routing to promote high levels of congestion.

2.3 PRoPHET

One key failing of both Epidemic and Spray and Wait is their lack of desire to make use of pathways that are more likely to result in message delivery. In a completely randomised scenario, this is less of a weakness as such pathways are unknowable. However in most real-world scenarios, connections are more predictable and hence so are the paths between nodes. This predictability allows for performance gains to be made when leveraging this increased "knowableness".

Message passing in PRoPHET is simple: if a node encounters a node with a greater probability of delivering a message, it passes it to that node. The main body of computation occurs in the calculation of these probabilities. At initialisation, the delivery predictability between any nodes A and B is $P(A, B) = 0$. Each node stores the predictability between itself and all other discovered nodes. If a node does not store a probability for any other node, it is taken to be 0.

At each encounter between node A and B, three rules are followed to allow for updating of the delivery predictability:

1. The predictability of node A encountering node B is increased:

$$P(A, B)' = P(A, B) + (1 - P(A, B)) \times L_{encounter} \tag{2.1}$$

2. The predictabilites for all nodes C other than encountered node B are "aged":

$$P(A, C)' = P(A, C) \times \gamma^k \tag{2.2}$$

3. B's predictabilities for nodes D are transferred to node A, taking advantage of the transitive properties of predictability:

$$P(A, D)' = P(A, D) + (1 - P(A, D)) \times P(A, B) \times P(B, D) \times \beta \tag{2.3}$$

For these rules, $L_{encounter}$, γ^k and β are constants that can be tuned for the specific scenario.

Chapter 3

The Scenario

This scenario considers a future in line with, but beyond that of the long term considerations described in [2]. Specifically, long term human presence on the surface or in the orbit of Mercury, Venus and Mars, either in the form of traditional colonies or autonomous drone spacecraft. Additionally, the scenario considers interplanetary space travel to be a commodity.

The assumptions and simplifications made for this scenario include negligible occlusion of bodies and linear orbital transfer trajectories. The latter, whilst a significant simplification from an orbital mechanics perspective, is assumed to be an acceptable analogue to a more complex transfer trajectory such as the Hohmann transfer orbit [14] or Bi-elliptic transfer [4] from a networking perspective. The former is assumed due to the complexities of simulating such an effect and is made feasible by the implication of network relay satellites positioned at the Sun–Earth's Lagrangian points L1, L2, L4 and L5.

3.1 The ONE Configuration Writer

In order to generate settings for ONE, this paper presents a library written in the Dart programming language [8] developed by Google. Dart is a multi-platform language and can target ARM, x86, JavaScript and the Dart VM.

The ONE Configuration Writer (OCW) allows for the programmatic definition of all of the configurable elements of the ONE. The core of the writer is an array of `ConfigModule`s. A module has two main components:

- `settings`: A map with strings as keys and another map as a value. These second tier maps use strings as keys and any data as values.
- `writeables`: An array of functions that can return a future [3]. These futures are used to allow for the writer to execute any asynchronous operations like network or file access.

The various modules, implemented as subclasses of the class in Listing 2, allow for a simple Application Programming Interface (API) to be used to define the configurable properties of ONE. This allows for more fine tuned control, confidence in the generated configurations and easier versioning of setups.

```dart
abstract class ConfigModule {
  Iterable<FutureOr Function(String outputPath)> get writeables => null;

  Map<String, Map<String, dynamic>> get settings => {};
}
```

Listing 2: Simplified `ConfigModule` implementation as an abstract class in Dart.

```
{
    'btInterface': {
        'type': 'SimpleBroadcastInterface',
        'transmitSpeed': 250000,
        'transmitRange': 10,
    },
    'highspeedInterface': {
        'type': 'SimpleBroadcastInterface',
        'transmitSpeed': 10000000,
        'transmitRange': 1000,
    },
}
```

Listing 3: Dart map literal representing the data in Listing 4.

```
btInterface.type = SimpleBroadcastInterface
# Transmit speed of 2 Mbps = 250kBps
btInterface.transmitSpeed = 250k
btInterface.transmitRange = 10

# High speed, long range, interface for group 4
highspeedInterface.type = SimpleBroadcastInterface
highspeedInterface.transmitSpeed = 10M
highspeedInterface.transmitRange = 1000
```

Listing 4: Snippet of ONE configuration file in the resultant format.

3.1.1 Settings

The two tiered map data structure was chosen to effectively model the data contained within the configuration file. The first map tier key represents the setting namespace and the second map tier key represents the setting name.

During writing, the library collates the maps from the `ConfigModule`s and adds a future to the `writeables` array that encodes and writes the data to the properties file.

The OCW also properly encodes large numerical values using the appropriate suffixes. For example, the integer 1000 is converted to the string `1k`.

3.1.2 Writeables

The second, and more powerful, feature of the OCW is the `writeables`. This allows modules to specify files to be written that are then referenced by the configuration file. This feature may be best described by the examples provided in this section.

External Path Movement

This module allows for the programmatic specification of a path to be used as a movement model within ONE. A list of paths, expressed as lists of points, can be passed to the module and location trace and activity trace files are written.

External Events Queue

When ONE cannot be relied upon to accurately simulate a type of event, such as a connection event, this module allows for a list of events to be provided which are then written in the expected format along with the correct file reference in the main configuration file.

3.2 The Space Configuration Writer

While the OCW provides an API with which the ONE can be configured, the ONE Space Scenario (OSS) provides a set of modules that can be used to configure ONE with a space based scenario. The OSS allows for the configuration of the planets used, the real-world time range used and the amount of simulated ships. This section outlines the methodology used.

3.2.1 Data Acquisition

The OSS requires the position data of the specified bodies in order to use them as hosts. In order to obtain this data, the JPL HORIZONS [15] web interface is used. This tool provides the Cartesian coordinates of any tracked body in the solar system and uses JPL's DE431 [1] ephemeris data.

3.2.2 Travel Time Calculation

In order to simulate ships between the planets, the travel time between any given planets at any given time must first be calculated. These are all calculated as a batch in order to determine the best time to travel between any two planets, the maximum time must be known, hence necessitating all times to be known.

The challenge of calculating travel times between bodies comes from the fact that they are both moving. This means that the distance between the bodies at any given instant cannot be used as a basis for time estimation. Instead, the distance between the bodies at two different times must be used.

Error Derivation

The travel time estimation problem can be reformulated to ask at what time t' must the ship arrive at body B providing it leaves body A at time t. This problem can be expressed as such:

$$T_{AB}(t, t') = t' - t \tag{3.1}$$

Where T_{AB} is a function that provides the time taken to travel from A at time t to B at time t'. This equation can be rearranged to give:

$$(T_{AB}(t, t') + t - t')^2 = error^2 \tag{3.2}$$

The goal of any solver to this problem is to choose a t' that minimises the $error^2$ term for any t. For this paper, travel is modelled linearly with constant speed. However this model is configurable in the OSS. The mathematical definition of T_{AB} is hence given, in this instance, as:

$$T_{AB}(t, t') = \frac{|\mathbf{B_{t'}} - \mathbf{A_t}|}{v} \tag{3.3}$$

Where:

- $\mathbf{B_{t'}}$ is a vector representing the position of B at time t'
- $\mathbf{B_{t'}}$ is a vector representing the position of B at time t'
- v is the constant speed at which a ship travels from A to B. In this paper, the value of v is taken to be 2,091,240 km/d, the speed of the Mars 2020 rover [33] as of 2020-12-14.

As an example, t is set to be 2000-01-01, A is Earth, B is Mars and t' ranges from 2000-01-01 to 2001-01-01. This example is shown in Figure 3.1.

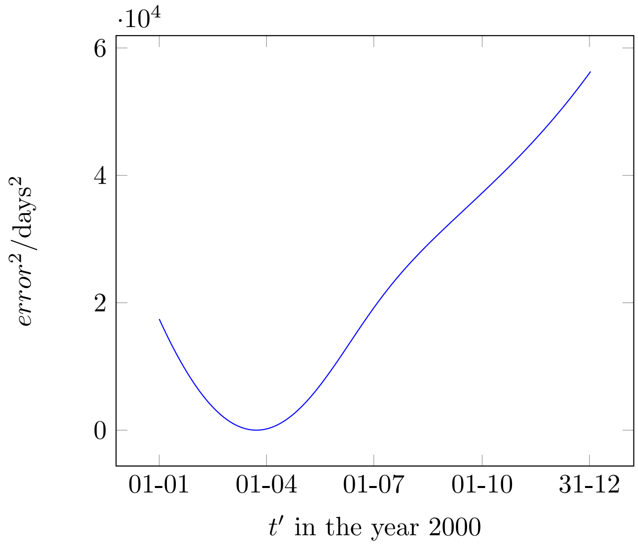

Figure 3.1: Graph showing $error^2$ between Earth and Mars as t' ranges from 2000-01-01 to 2001-01-01, with t as 2000-01-01.

Error Minimisation

Upon first glance, Newton's method [26] seems the simplest solution to this problem. Indeed, it does find the minimum and in better speed than a linear search. However, in the case of calculating all of the travel times for t over the whole simulation period, a linear search is made more effective by starting the search at the t' found for $t-1$. This optimisation means only a small number of guesses for t' are required for any t. This method also removes the requirement for T_{AB} to be differentiable by t'. Whilst the travel time model used for this paper is easily differentiated, a more complex model may suffer more from this requirement.

This optimisation hence gives us $\hat{T}_{AB}(t)$, the time taken to travel along the optimal path between A and B when leaving at time t. This is calculated as the difference between t and t' when the $error^2$ term from Equation 3.2 is minimised.

3.2.3 Ship Simulation

With the travel times between all pairs of planets at all times calculated, the ships that travel between these planets are now simulated. Each ship has a chance of deciding to travel at any given time slice. Once a ship has decided to travel, it calculates the optimality of all of the trips possible from the planet on which it is currently situated and the most optimal trip is hence taken. The optimality of any given trip is calculated as such:

$$O_{AB}(t) = 1 - \frac{\hat{T}_{AB}(t) - \min_{t'} \hat{T}_{AB}(t')}{\max_{t'} \hat{T}_{AB}(t') - \min_{t'} \hat{T}_{AB}(t')} \tag{3.4}$$

Explained intuitively, this calculation evaluates to 0 when the trip time at t is equal to the maximum trip time across the simulation, 1 when its equal to the minimum and linearly interpolated in between.

These trips are prefaced with random delays to reduce the change of ships bundling up. This includes the first trip.

3.2.4 Interfaces

The scenario makes use of three types of interface:

- Planet to planet optical

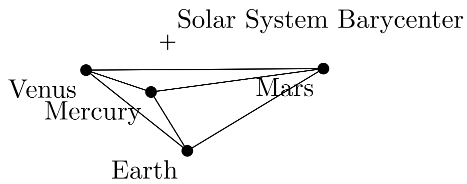

Figure 3.2: Positions of considered planets on 01-01-2000 with optical interface connections depicted.

Region	**Domestic Bandwidth Production** [16]	**Adjustment** [36]	**Adjusted Production**
Europe	39.2 Tbit/s	1.05	41.2 Tbit/s
Asia-Pacific	7.88 Tbit/s	8.4	66.2 Tbit/s
Latin America	7.35 Tbit/s	1.05	7.7 Tbit/s
North America	5.52 Tbit/s	21	116 Tbit/s
Africa	0.977 Tbit/s	1.05	1.03 Tbit/s
		Total	232 Tbit/s

Table 3.1: Breakdown of domestic bandwidth production in December 2020 by global region with assumed adjustments.

- Ship to planet optical
- Ship to ship radio

This section will provide the motivations and an overview of these interfaces.

Planet to Planet Optical

One of the largest technological bottlenecks involved in deep space exploration is communication. As an example, a mission to map the Martian surface would succeed in mapping only two per cent of the planet. With a space-based communications laser however, the rate could be increased 100 fold [12].

One of the most promising recent examples of space based optical communication is the Lunar Laser Communications Demonstration (LLCD) [28]. This was NASA's first attempt at demonstrating optical communications between a terrestrial terminal and an lunar orbiting spacecraft. The demonstration boasted a downlink speed of up to 622 Mbps with a space terminal that was lighter, smaller and more power efficient than an RF based terminal with equivalent capabilities [23].

This scenario considers a set of orbital communication terminals stationed at each considered planet. This set is represented by a network interface attached to hosts representing each planet. The scenario also considers enough redundancy in these terminals to allow constant connection between each possible pair of planets. Figure 3.2 shows the optical connections between the considered planets.

In order to provide a value for the transfer speed of the interface, we use an estimate of the global domestic bandwidth production. This value is taken as a weighted sum of the regions defined in Table 3.1. and is calculated to be 232 Tbit/s. However, as ONE allows only for speeds of up to the Java maximum integer value, all speeds are scaled. As such, the value used in the simulator is 2147483647 B/s [32].

Planet to Ship Optical

This scenario also considers ship based optical communications to be a commodity. As such, it is assumed that enough planet based optical communication terminals are present to allow for an arbitrary number of connections between a planet and a set of ships. However, a ship is considered to be able to support a single optical connection. This is due to space, mass and power considerations.

Due to the limited capacity of the ship's interface, a ship is considered connected to only the closes planet on this interface. The absolute speed of this interface is taken to be 1 Tbit/s [18] which is scaled to 9256395 B/s.

Ship to Ship Radio

To supplement the optical communication interfaces, the scenario also equips each ship with a high gain radio transceiver. This transceiver is intended for ship to ship communication.

As the considered radio transceivers are directional, their angle is nontrivial to calculate. The globally optimal orientations of transceivers corresponds to a minimum spanning tree configuration where edges correspond to connections between nodes. However, as a simplification, a locally optimal metric is used that optimises for average amount of connection per node.

In order to calculate the optimal transceiver angle of a ship with position $\mathbf{s_0}$, a modified brute force approach is used. In order to reduce the amount of possible angles, only angles corresponding to unique combinations of connections are considered. This results in only as many combination as the amount of ships within radio range of the source ship. The optimality of an angle is given by the amount of ships visible with the transceiver field of view and range and is hence calculated as such:

$$O(\theta) = |\{\mathbf{s} \in S : |atan2(\mathbf{s}_y - \mathbf{s_0}_y, \mathbf{s}_x - \mathbf{s_0}_x) - \theta| > \alpha\}| \tag{3.5}$$

Where:

- $O(\theta)$ is the function giving the optimality of transceiver angle θ
- S is the set of vectors representing the positions of neighbouring ships
- $\mathbf{s_0}$ is the vector representing the position of the considered ship
- α is the transceiver half-angle

Hence the tranciever angle of a ship is given by:

$$\underset{\theta}{\operatorname{argmin}}\ O(\theta) \tag{3.6}$$

The angle is then refined by setting it to the average of the angles of the set of connected ships.

The speed for this interface is given as 506 Mbit/s [25]. This value is then scaled to 4683 bits/s. For the transceiver half angle a value of 10° is used and for transceiver range a value of 0.25 AU is used.

Chapter 4

Results

For result collection, body position data from 01-01-2000 to 31-12-2000 is used. For each router, 5 simulations were run for each a unique ship simulation was conducted and the results for each averaged. In order to ensure consistency in the difference in order of magnitude between transfer speed and buffer size, the planet nodes were given an buffer size of 1 Gbit and the ship nodes a buffer size of 1 Mbit. The PRoPHET router used a β value of 0.9 and a Γ value of 0.999 [35]. The Spray and Wait router was configured to spread a maximum of 50 messages in a binary fashion. These metrics are compared in Figures 4.1 and 4.2.

4.1 Analysis

4.1.1 Delivery Probability

As is to be expected, the Epidemic router results in a very low delivery probability. This is due to the buffer being quickly flooded. This effect is shown in Figure 4.3.

Figure 4.3 also shows greatly increasing buffer occupancy as time continues with both Spray and Wait and PRoPHET. This effect would result in reduced effectiveness of these protocols at greater timescales. This would suggest these protocols are suffering from the effects of lack of node density. As such, a higher delivery probability may be achieved as a result of higher node density. This possibility is confirmed in Figure 4.4.

4.1.2 Median Latency

Upon first glance, Epidemic routing seems to provide optimal median latency. However when viewed in the context of the delivery probabilities, it can be seen that the only messages that contribute to the median latency are those that were trivial to deliver due to the proximity of the destination node to the host node. This follows as such messages would largely not be subject to the impedance as a result of high buffer occupancy.

When comparing the remaining protocols, Spray and Wait and PRoPHET, the latter, and more advanced protocol, results in a lower median latency as it can more effectively take advantage of the predictability of encounters.

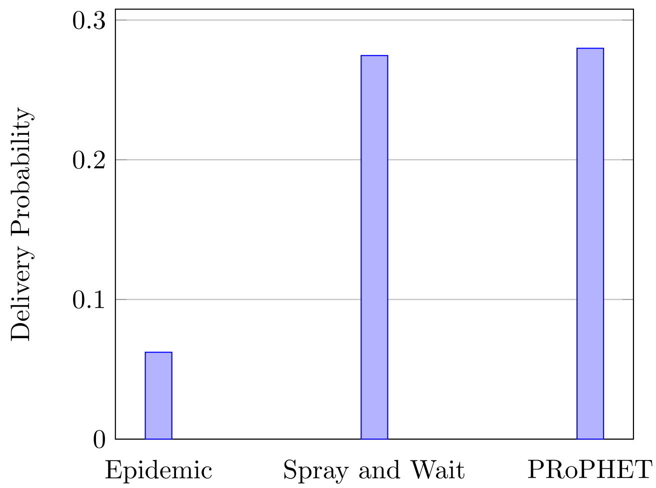

Figure 4.1: Comparison of delivery probabilities.

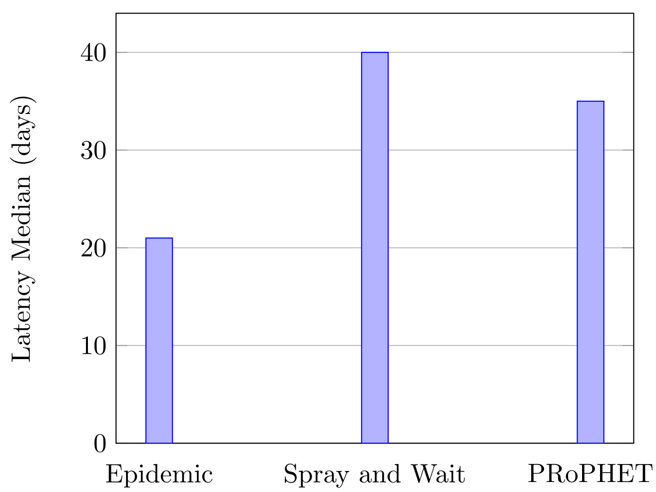

Figure 4.2: Comparison of median latencies.

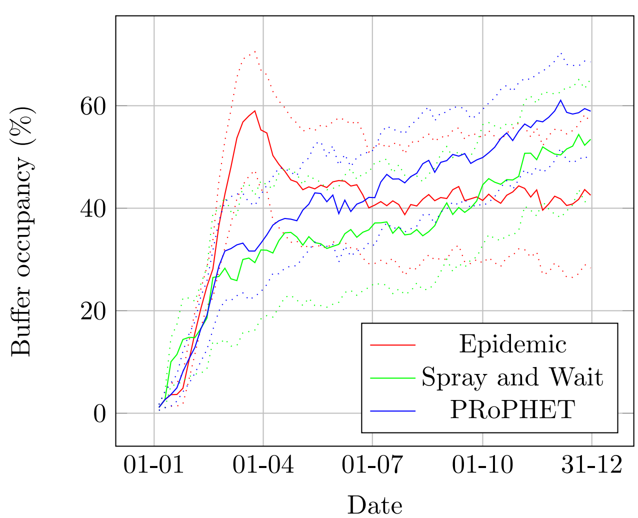

Figure 4.3: Graph showing buffer occupancy through time for each router, variance range are dotted.

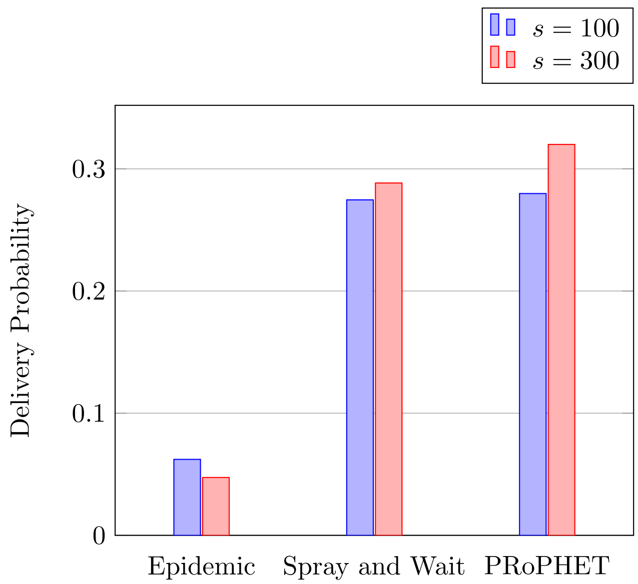

Figure 4.4: Graph comparing delivery probabilites as ship count s increases.

Chapter 5

Review

5.1 Space Based DTNs

In conventional networking, delays are an artefact of switch processing speed, the signal speed through various media or server response. In space however, delays are a core tenant of the infrastructure. Nodes are often so sparsely placed that a message must necessarily take from seconds to hours to travel from one node to the next, a limit imposed not by technical limitations but by the laws of nature themselves.

Additionally, whilst conventional networking may consider a disconnection to be a failure of the entire session, DTNs must account for these disconnections and be resilient to them. A varying network topology is one of the main design considerations one must allow for when considering a generalised networking approach to communication in space.

Space based DTNs, specifically IPNs, are an edge case of oppnets. They present challenges not present in terrestrial DTNs, beyond even the greatest two mentioned previously. Such extra challenges include the added complexity of gravitational lensing, occlusion by celestial bodies and signal attenuation over distance. These challenges are are not present or have negligible effect in the majority of DTN applications.

The majority of objects humanity sends into space are presently and have been historically single use or single purpose. Many missions have been designed in a bubble, with no dependencies on external resources. As these missions approach the end of their lives, the resources deployed in order to achieve them are perfectly placed to become the backbone of an interplanetary internet. The networks build on top of these resources will benefit from the lessons learnt in terrestrial DTNs.

5.2 A New Simulator

Currently, the simulators used in research are designed for applications deployable today. As such, they do not cater to the specific challenges of the IPN. A new simulator, designed specifically for these challenges, would allow for better research into this field.

5.2.1 Temporal Variability

In a smaller simulation space, such as a neighbourhood or city, events mostly occur at intervals within the same order of magnitude. If the simulator does not account for the speed of messages in their transmission media, this effect is even more pronounced. This allows for simulators designed for these environments to benefit from this assumption by using constant time slicing or even a hybrid time slicing and scheduling approach. However, on the scale of the solar system, the temporal variability of events would be impossible to effectively model using a constant time slicing approach and would create unnecessary overhead with a hybrid approach.

Consider for example SpaceX's Starlink [30] constellation. SpaceX currently plans to launch 42,000 [13] at an altitude of 550 km in orbital planes of 53° [31] inclination of 60 satelites. This would result in a maximum inter satellite distance of around 724km, a distance light would take around 2.4ms to traverse. In contrast, light takes 5 hours to get to Pluto from the Sun. Events on both these scales in the same simulation makes time slicing an inappropriate approach.

A purely scheduling based approach would alleviate these issues and would hence be an effective approach in a novel simulator.

5.2.2 Time of Light Simulation

Many of today's DTN simulators make use of the assumption that messages are transferred instantly. Such an assumption is impossible with IPNs. The core problem to be solved when attempting to simulate this effect is similar to that presented in Section 3.2.2. However, due to the vast difference in speeds between light and network hosts, Newton's method is much more appropriate. The only requirement for this approach is knowledge of host velocities. This compute heavy yet resource light sub problem is also a good target for parallelism.

5.2.3 Gravitational Lensing and Body Occlusion

This additional consideration increases the complexity of the overall simulation by adding a photon simulation aspect to the time of light simulations. This problem could, however, take cues from solutions used by ray tracing and marching software. Using a variable step size, a common approach used in these fields, could allow for efficient yet accurate simulation of these effects.

References

[1] C. Acton. *Notes on the DE430 and DE431 Planet and Lunar Ephemerides*. Aug. 2013. URL: https://web.archive.org/web/20140116001551/http://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets/AAREADME_Planet_SPKs.txt.

[2] Ahmad Alhilal, Tristan Braud, and Pan Hui. "The Sky is NOT the Limit Anymore: Future Architecture of the Interplanetary Internet". In: *IEEE Aerospace and Electronic Systems Magazine* 34.8 (Aug. 2019), pp. 22–32. DOI: 10.1109/maes.2019.2927897. URL: https://doi.org/10.1109/maes.2019.2927897.

[3] Henry C. Baker and Carl Hewitt. "The Incremental Garbage Collection of Processes". In: *SIGPLAN Not.* 12.8 (Aug. 1977), pp. 55–59. ISSN: 0362-1340. DOI: 10.1145/872734.806932. URL: https://doi.org/10.1145/872734.806932.

[4] *Bi-elliptic transfer*. URL: https://en.wikipedia.org/wiki/Bi-elliptic_transfer (visited on 01/08/2021).

[5] Chiara Boldrini et al. "Opportunistic Networks". In: *Computer Communications* 48 (July 2014). DOI: 10.1016/j.comcom.2014.04.007.

[6] V. Cerf and R. Kahn. "A Protocol for Packet Network Intercommunication". In: *IEEE Transactions on Communications* 22.5 (1974), pp. 637–648. DOI: 10.1109/TCOM.1974.1092259.

[7] Susan D'Agostino. "To Boldly Go Where No Internet Protocol Has Gone Before". In: *Quanta Magazine* (Oct. 2020).

[8] *Dart Programming Language*. URL: https://dart.dev/ (visited on 01/08/2021).

[9] Alan Demers et al. "Epidemic Algorithms for Replicated Database Maintenance". In: *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing*. PODC '87. Vancouver, British Columbia, Canada: Association for Computing Machinery, 1987, pp. 1–12. ISBN: 089791239X. DOI: 10.1145/41840.41841. URL: https://doi.org/10.1145/41840.41841.

[10] Kevin Fall. "A Delay-Tolerant Network Architecture for Challenged Internets". In: *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM '03. Karlsruhe, Germany: Association for Computing Machinery, 2003, pp. 27–34. ISBN: 1581137354. DOI: 10.1145/863955.863960. URL: https://doi.org/10.1145/863955.863960.

[11] Eric Geier. "Discovering the Upcoming Wi-Fi Direct Standard". In: *Cisco Press* (Aug. 16, 2010). URL: https://www.ciscopress.com/articles/article.asp?p=1620205&seqNum=2.

[12] Hamid Hemmati. *Interplanetary Laser Communications*. Nov. 2007. URL: https://www.osa-opn.org/home/articles/volume_18/issue_11/features/interplanetary_laser_communications/.

[13] Caleb Henry. *SpaceX submits paperwork for 30,000 more Starlink satellites*. Oct. 2019. URL: https://spacenews.com/spacex-submits-paperwork-for-30000-more-starlink-satellites/.

[14] *Hohmann transfer orbit*. URL: https://en.wikipedia.org/wiki/Hohmann_transfer_orbit (visited on 01/08/2021).

[15] *HORIZONS System*. URL: https://ssd.jpl.nasa.gov/?horizons (visited on 01/08/2021).

[16] *Internet Exchange Point Growth by Region*. Dec. 2020. URL: https://www.pch.net/ixp/summary_growth_by_region (visited on 12/21/2020).

[17] Joab Jackson. "The Interplanetary Internet - IEEE Spectrum". In: *IEEE Spectrum* (Aug. 2005). URL: https://spectrum.ieee.org/telecom/internet/the-interplanetary-internet.

[18] Matilda Jordanova-Duda. *Datenreise per Laserstrahl*. Feb. 2019. URL: https://web.archive.org/web/20190218202024/https://www.vdi-nachrichten.com/Technik/Datenreise-Laserstrahl (visited on 12/27/2020).

[19] Ari Keränen. *Opportunistic Network Environment Simulator*. Apr. 2008.

[20] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. "The ONE Simulator for DTN Protocol Evaluation". In: *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. Rome, Italy: ICST, 2009. ISBN: 978-963-9799-45-5.

[21] Leszek Lilien et al. "Opportunistic networks: the concept and research challenges in privacy and security". In: *Mobile and Wireless Network Security and Privacy* (Jan. 2007). DOI: 10.1007/978-0-387-71058-7_5.

[22] A. Lindgren et al. *Probabilistic Routing Protocol for Intermittently Connected Networks*. Tech. rep. Aug. 2012. DOI: 10.17487/rfc6693. URL: https://doi.org/10.17487/rfc6693.

[23] *LLCD: 2013-2014*. June 2018. URL: https://www.nasa.gov/directorates/heo/scan/opticalcommunications/llcd/.

[24] *Mobile Phone Base Stations, How do mobile base stations work, Mobile Base Stations in Australia, Cell Tower, Mobile Phone Tower*. URL: https://mobilenetworkguide.com.au/mobile_base_stations.html (visited on 11/16/2020).

[25] *Newtec - IP Trunking and IP Backbones over Satellite*. URL: https://www.newtec.eu/frontend/files/userfiles/files/Corporate%5C%20application%5C%20note%5C%20App%5C%20Note%5C%20IP%5C%20Trunking%5C%20Rev1.pdf (visited on 12/28/2020).

[26] Isaac Newton. *De analysi per aequationes numero terminorum infinitas*. William Jones, 1669.

[27] *Properties (The Java™ Tutorials > Essential Classes > The Platform Environment)*. URL: https://docs.oracle.com/javase/tutorial/essential/environment/properties.html (visited on 11/17/2020).

[28] B. S. Robinson et al. "The lunar laser communications demonstration". In: *2011 International Conference on Space Optical Systems and Applications (ICSOS)*. 2011, pp. 54–57. DOI: 10.1109/ICSOS.2011.5783709.

[29] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks". In: *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking*. WDTN '05. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 2005, pp. 252–259. ISBN: 1595930264. DOI: 10.1145/1080139.1080143. URL: https://doi.org/10.1145/1080139.1080143.

[30] *Starlink*. URL: https://www.starlink.com/ (visited on 01/08/2021).

[31] *Starlink Satellite Constellation of SpaceX*. URL: https://directory.eoportal.org/web/eoportal/satellite-missions/s/starlink (visited on 12/31/2020).

[32] *The Java™ Tutorials - Primitive Data Types*. URL: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html (visited on 12/27/2020).

[33] *Trip to Mars - NASA Mars*. URL: https://mars.nasa.gov/mars2020/timeline/cruise/ (visited on 01/08/2021).

[34] Amin Vahdat and David Becker. "Epidemic Routing for Partially-Connected Ad Hoc Networks". In: *Technical Report* (June 2000).

[35] Luming Wan et al. "Routing Protocols For Delay Tolerant Networks: Survey And Performance Evaluation". In: *International Journal of Wireless & Mobile Networks* 7 (June 2015), pp. 55–69. DOI: 10.5121/ijwmn.2015.7305.

[36] Bill Woodcock. *How much data is transferred on the Internet on a daily basis?* July 2020. URL: https://www.quora.com/How-much-data-is-transferred-on-the-Internet-on-a-daily-basis.

[37] David Zillig. *CANDOS Experiment Overview*. Oct. 2002. DOI: 10.2514/6.2002-T3-26.