



HAL
open science

Network Storage Analysis via Semiring Geometry

William Bernardoni, Robert Kassouf-Short, Robert Cardona, Brian Heller,
Justin Curry, David Spivak, Juan A Fraire

► **To cite this version:**

William Bernardoni, Robert Kassouf-Short, Robert Cardona, Brian Heller, Justin Curry, et al.. Network Storage Analysis via Semiring Geometry. 2024 IEEE Aerospace Conference, Mar 2024, Big Sky, France. pp.1-19, 10.1109/AERO58975.2024.10521207 . hal-04711322

HAL Id: hal-04711322

<https://hal.science/hal-04711322v1>

Submitted on 26 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Network Storage Analysis via Semiring Geometry

William Bernardoni
Case Western Reserve University
wrb37@case.edu

Robert Kassouf-Short
NASA Glenn Research Center
robert.s.short@nasa.gov

Robert Cardona, Brian Heller, and Justin Curry
University at Albany, State University of New York
{rlcardona, bheller, jmc Curry}@albany.edu

David Spivak
Topos Institute
dspivak@gmail.com

Juan A. Fraire
Inria Lyon - CONICET
juan.fraire@inria.fr

Abstract—The long-term goal of space networking studies is to provide the foundation needed to support a Solar System Internet (SSI). We anticipate that such an SSI would be composed of science and exploration satellites, ground stations, routing satellites, and ultimately humans and robots on the surfaces of different planets throughout our solar system. Interplanetary communications networks will need to handle the delays, disruptions, and disconnections inherent to space communications. The architecture of Delay Tolerant Networking (DTN) provides protocols and strategies to support these communications plans. At its core, DTN relies on a store-carry-forward approach to provide robust communications in the presence of delays and disruptions.

In the past, routing models in DTN have focused on the forwarding aspects of store-carry-forward. Methods such as Contact Graph Routing, Contact Multigraph Routing, and Probabilistic Routing Protocol using the History of Encounters and Transitivity (PRoPHET) provide solutions for how to choose where to forward bundles through a network. However, these routing models often function bundle-by-bundle allowing them to set aside storage needs for each node. In order to bring our vision of a SSI to reality, we will need to be able to predict and incorporate storage needs that satellites and rovers will require. Modeling and predicting storage needs is the first step to being able to optimize storage needs and also optimize communication footprints on the size, mass, and power needs of future satellites.

In this paper, we introduce a novel semiring model for contact-based routing protocols that includes a means of determining storage needs. Through proper analysis of the semiring structure, we show how to determine optimal storage structures in satellite networks. In addition, we run our analysis on simulated satellite networks to demonstrate the potential for working with these semiring models in a computational framework. We conclude by indicating future directions for semiring analysis in space communications.

TABLE OF CONTENTS

| | |
|---|----|
| 1. INTRODUCTION..... | 1 |
| 2. BACKGROUND: ROUTING IN DTN | 2 |
| 3. BACKGROUND: SEMIRINGS | 4 |
| 4. BACKGROUND: THE ALGEBRAIC PATH PROBLEM | 5 |
| 5. SEMIRING MODELLING OF COMMUNICATION WINDOWS..... | 6 |
| 6. SEMIRING MODELLING OF TRANSMISSION DURATIONS UNDER LIMITED STORAGE | 9 |
| 7. EXPERIMENTAL SETUP | 15 |
| 8. RESULTS | 15 |
| 9. CONCLUSION | 16 |
| 10. FUTURE WORK | 17 |

REFERENCES 17

1. INTRODUCTION

At its core, Delay Tolerant Networking (DTN) rests on a Store-and-Forward method of transmitting packets [1]. Much has been written on the ‘Forward’ piece of this puzzle, specifically on methods for routing in networks using DTN. Routing methods include Contact Graph Routing (CGR)[2], [3], Contact Multigraph Routing (CMR) [4], [5], Probabilistic Routing Protocol using the History of Encounters and Transitivity (PRoPHET) [6], [7], and even several methods using machine-learning models [8], [9], [10], [11]. However, the ‘Storage’ piece of the puzzle has been left for others to determine. In particular, many routing papers begin with the assumption that technical specifications for the satellites involved – such as planned orbits or computing power – will be predetermined by trade studies and mission requirements decided by whoever constructs the satellites.

Our goal in this paper is to introduce a means of computing storage requirements for networks using DTN. The key insight that we build upon is that the core mathematical structure governing path problems in networks is the idempotent semiring. As seen in [12], the solutions to path problems in networking can often be algebraized using semirings.

As encoding scheme can vary the bandwidth of a satellite link, we approach the problem of storage and throughput in the time domain. Storage, in our sense, can be thought of as the sum total of “layovers” taken between contacts. This perspective has the advantage of being unbound to specific systems – while our chief focus in this paper is on satellite networking, the techniques introduced in this paper can also be applied to other problems such as job scheduling or freight routing.

As seen in [12] and [13], many routing problems can be expressed using the language of semirings as matrix equations over varying number systems. By finding the solutions to these matrix equations, a problem known as the algebraic path problem, optimal routing strategies can be determined. This transforms a variety of combinatorial problems into linear algebra problems over the correct number system.

In this paper we introduce semirings which allow delay tolerant networking to be modeled by the algebraic path problem. These semirings have a myriad of potential applications and uses, as they give an algebraic and equational language to DTN. In this paper we will demonstrate an additional use – by analyzing the inherent geometry of these semirings we can derive a closed form method of determining the maximum transmission durations and corresponding storage requirements in a delay tolerant network.

All methods introduced in this paper are easily computable. C++ code implementing these theorems and methods can be found at the GitHub repository, wrberardini/Semiring-Geometry. Python code implementing these theorems and methods can be found at the GitHub repository <https://github.com/TheaMAS/sat-parser>.

After establishing the problem of routing in delay tolerant networks and demonstrating the value of semirings in addressing these problems, our first major contribution is Theorem 5.6 where we demonstrate that the Nevada construction requires five numbers to uniquely define each object. Then, we modify our semiring to incorporate storage considerations. In this next portion, we present Theorem 6.10 which yields a strict bound on the storage required to achieve a given transmission duration, alongside Theorem 6.11 which describes a means of determining how the storage must be distributed within the network. We then end the paper by setting up a fifteen node Earth-Moon simulated system and demonstrating how to apply our analysis methods to this system.

A reader versed in routing in DTN may skip section 2.

A reader versed in semirings may skip section 3, and 4, noting only that in this paper we will deal exclusively with additively idempotent semirings.

2. BACKGROUND: ROUTING IN DTN

Transferring data between DTN nodes necessitates a routing framework that adeptly manages space communication's inherent delays and disruptions. Unlike routing in static and mobile Internet networks, which are characterized by persistent and close-to-zero latency end-to-end paths, routing in DTNs demands algorithm adaptations, time-dynamic scheduling, and specific graph models.

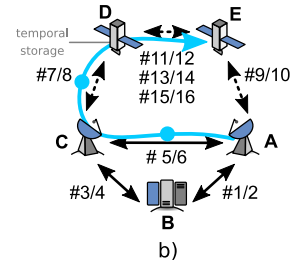
Non-Deterministic Routing Procedures—In DTNs with uncertain connectivity, routing strategies can adopt various approaches based on the network's characteristics and requirements. Simple methods like "flooding" might be utilized, which involve indiscriminately sending messages to all neighboring nodes. Alternatively, more controlled strategies like "epidemic" routing can be employed, which selectively share messages with nodes during encounters to reduce redundancy [14]. Another method in this category includes "Spray and Wait" [15], which involves a controlled replication of messages to specific nodes. In scenarios where connectivity patterns can be inferred, probabilistic inference methods such as PROPHET [6], [7] and MaxProp [16] might be leveraged. These methods utilize historical data and probabilistic metrics to predict future encounters and interactions among nodes, aiding in making informed routing decisions amidst the network's uncertainties.

Contacts and Contact Plans—In space, routing for DTN utilizes a "contact plan," a predictive model of future connectivity opportunities, or "contacts" between nodes, derived from precise orbital propagators and communication system models [17]. This contact plan, which outlines the expected resources and connectivity windows available for data transport, becomes the backbone for routing decisions in space systems.

Routing Phases—The routing process in space DTN is typically delineated into three interconnected stages: planning,

| # | snd. | rcp. | s | e | r | ω |
|-------|------|------|----|----|---|----------|
| 1/2 | A | B | 0 | 60 | 1 | 1 |
| 3/4 | B | C | 0 | 60 | 1 | 1 |
| 5/6 | A | C | 0 | 60 | 1 | 1 |
| 7/8 | C | D | 0 | 30 | 1 | 1 |
| 9/10 | A | E | 10 | 20 | 1 | 1 |
| 11/12 | D | E | 0 | 10 | 1 | 1 |
| 13/14 | D | E | 30 | 40 | 1 | 1 |
| 15/16 | D | E | 50 | 60 | 1 | 1 |

a)



b)

Figure 1: Example space network represented by a) a Contact plan table, and b) a static graph of the topology.

routing, and forwarding. During the *planning* phase, a centralized entity, such as mission control, generates the contact plan, either distributed to the space DTN nodes for distributed routing computation or retained for centralized routing calculation. Subsequently, in the *routing* phase, the contact plan is utilized as input for the CGR algorithm, which computes optimal paths to network destinations, considering the next hop and the optimal data delivery time, route volume limit, and valid transmission interval. Finally, the *forwarding* phase selects the best route from the computed route tables, considers real-time local conditions, and manages the data (or "bundle") transmission or storage until the next viable contact.

Topology Model: Contacts and Routes

Contact—A contact can be formally represented as $(C_{snd,rcp}^{s,e}, r, \omega)$, where it is defined as a time interval $(s; e)$ during which data is anticipated to be transmitted by DTN node *snd* (the sender) at rate r , ensuring reception by node *rcp* (the recipient) with a latency of ω . The time values, s and e , can be articulated either in absolute units, such as Gregorian Coordinated Universal Time (UTC), or in relative time, measured from a reference epoch.

In Fig. 1 (adapted from [3]), a table enumerates each contact, identified by a number (#1...16). Contacts $C_{A,B}^{0,60}$, $C_{B,C}^{0,60}$, and $C_{A,C}^{0,60}$ exemplify permanent links, such as those between mission control and ground stations interconnected through the Internet. Meanwhile, contacts $C_{C,D}^{0,30}$ and $C_{A,E}^{10,20}$ represent episodic Ground to Space Links (GSLs), and contacts $C_{D,E}^{0,10}$, $C_{D,E}^{30,40}$, and $C_{D,E}^{50,60}$ identify episodic Inter-Satellite Links (ISLs).

Note that contacts in space communications are defined as unidirectional due to the prevalence of one-way transmission in this context. Therefore, two unidirectional contacts are paired to represent bidirectional communication within a contact plan. Additionally, owing to the one-way light time (*owlt*, or ω), the start time of a contact typically does not synchronize with the reverse channel in a bidirectional link [3].

Route—A route, denoted as R_{src}^{dst} , where a bundle has node *src* as its current location and node *dst* as its destination, is defined as a **sequence of contacts**, also referred to as *hops*, within a contact plan. This definition adheres to the following conditions: a) *src* is the sending node for the initial contact, b) *dst* is the receiving node for the final contact, c) The receiving node for contact i becomes the sending node for contact $i + 1$, and d) The end time for contact $i + 1$ is not

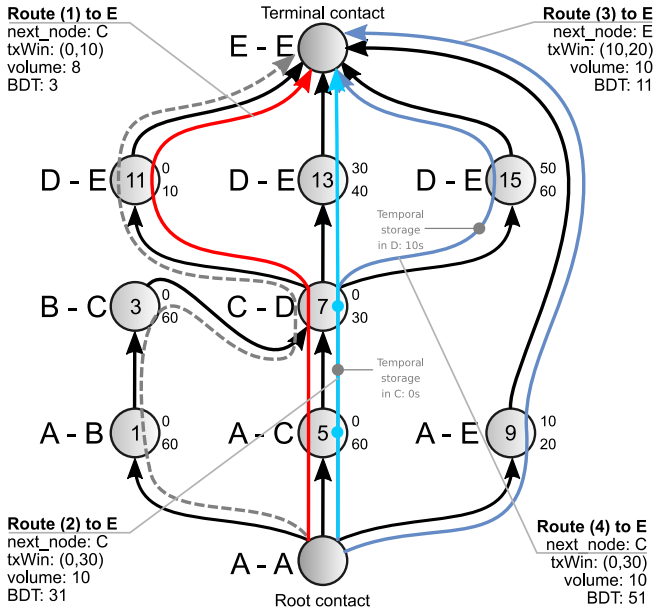


Figure 2: Contact graph CG_A^E with routes R_A^E for the topology in Fig. 1.

earlier than the start time for contact i [18].

Consider the example provided in Figs. 1, where one of the quickest routes, namely route (1) $R_A^E = \{C_{A,C}^{0,60}, C_{C,D}^{0,30}, C_{D,E}^{0,10}\}$, is emphasized. An alternative route, route (2) $R_A^E = \{C_{A,C}^{0,60}, C_{C,D}^{0,30}, C_{D,E}^{30,40}\}$, can be derived using the same nodes but with a different final contact. As mentioned, DTN does not assume consistent end-to-end connectivity. Consequently, a route may necessitate temporary storage at intermediary nodes (for instance, at node D until time 30 in route (2)). Moreover, contact propagation delays (ω) are cumulatively added to storage times to compute the best delivery time (BDT) for each route. For instance, route (1) demonstrates a $BDT = 3$, even with immediate transmission at each node, due to the $\omega = 1$ in each of its three constitutive contacts. On the other hand, route (2) provides a $BDT = 31$ since node D will store the data until time 30. Storage and propagation times within a route are calculated during the routing procedure.

A routing procedure refers to the method through which a sequence of contacts is calculated based on a specified contact plan. The Contact Graph Routing (CGR) [2], [3] variants are among the most developed and considered below. However, alternatives such as Contact Multigraph Routing (CMR) [4], [5] and Shortest-Path Tree Approach for Routing in Space Networks (SPSN) [19] have been proposed as viable methodologies. In general, these procedures can be executed in a distributed manner on the DTN node, thereby utilizing the most recent traffic and topological information. Alternatively, the computations might be conducted centrally, within a ground-based mission operations and control center [20].

Contact Graph—Contact graphs, depicted in Fig. 2 (adapted from [3]), offer a structured approach to model the connectivity and routes in space DTN. A contact graph, when considering a destination node dst from a source node src , is conceptualized as a directed acyclic graph $CG_{src}^{dst} = (V, E)$, where the vertices V represent contacts $C_{snd,rcp}^{s,e}$ within the

contact plan. The edges E in a contact graph symbolize episodes of data retention at a node i , spanning the time between the end of a preceding contact and the start of the subsequent one. A CG is constructed by assigning a vertex to each contact in the contact plan, representing a transmission from src node to dst node, directly or indirectly through other contacts. Subsequently, edges are added between contacts with corresponding destination and source nodes. Finally, notional contacts from node src to itself and from node dst to itself (a.k.a. root and terminal contacts) are included in the CG . Fig. 2 illustrates CG_A^E and R_A^E , based on the contact plan example shown in Fig. 1.

Note that each pair of src and dst nodes utilizes a unique CG data structure. While this may appear a disadvantage, it aligns well with its distributed nature on flight computers with limited computing power. Computing on-demand routes to a single destination is often more pertinent than to all or multiple destinations.

Pathfinding—The primary benefit of utilizing CG data structures lies in their compatibility with conventional shortest-path algorithms. Specifically, Dijkstra’s shortest path algorithm can be modified to identify the optimal path from a source to a destination node [21]. The detailed adaptation of the modified Dijkstra’s algorithm is provided in [3]. The time complexity of the CGR Dijkstra call is $\mathcal{O}(|C|\log(|C|))$ (with min-priority queues and Fibonacci Heaps), where $|C|$ is the size of the contact plan. The computed path can then be employed in one of two ways: it can dictate the next hop in a distributed routing scenario, such as in the Interplanetary Overlay Network (ION) [22], or it can define the entire route path in source or centralized routing approaches [23]. This flexibility and adaptability of contact graph data structures facilitate efficient and effective routing in diverse network configurations.

Route Table—A single path is insufficient in practice since (i) routes have an expiration, (ii) routes possess limited volume, (iii) routes can be overbooked for specific priority classes, and (iv) routes may not be feasible due to uncertainties or failures. Therefore, a list of routes is needed. To this end, a series of route table management methods have been explored in [24]. The accurate method for constructing the route table involves utilizing Lawler’s modification of Yen’s algorithm [25]. This approach, implemented in ION v3.7 and persisting through to the current version at the time of writing (v4.0), employs Yen’s algorithm to execute a Dijkstra search within a nested loop. This delivers a set of the K optimal routes ($len([R_S^D]) = K$), with K being supplied as an argument.

Despite the noteworthy progress in deterministic routing for DTNs through CGR, storage utilization has, to the best of the author’s knowledge, yet to be thoroughly integrated as a variable or constraint within the overarching routing process. In its core functionality, routing delivers calculated sequences of contacts to allow efficient data transmission across the network. Moving forward in this work, we delve into the development of storage models, employing semiring geometry to render route selection aware of storage considerations. The following sections will unfold these models in detail, exploring their potential to seamlessly intertwine storage considerations with the established routing mechanisms, thereby paving the way toward a more integrated and efficient DTN management.

3. BACKGROUND: SEMIRINGS

Definition 3.1. A **semiring** is a tuple $(S, \oplus, \otimes, 0_S, 1_S)$, consisting of a set S , two binary operations on S , $\oplus, \otimes : S \times S \rightarrow S$, and two distinguished elements $0_S, 1_S \in S$ such that:

1. $(S, \oplus, 0_S)$ is a unital commutative monoid, i.e. for all $a, b, c \in S$:

(a) \oplus is associative: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

(b) \oplus is commutative: $a \oplus b = b \oplus a$

(c) 0_S is the unit of \oplus : $a \oplus 0_S = a = 0_S \oplus a$

2. $(S, \otimes, 1_S)$ is a (*potentially noncommutative*) monoid, i.e. for all $a, b, c \in S$:

(a) \otimes is associative: $(a \otimes b) \otimes c = a \otimes (b \otimes c)$

(b) 1_S is the unit of \otimes : $a \otimes 1_S = a = 1_S \otimes a$

3. S satisfies the distributive laws:

(a) $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

(b) $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$

4. Multiplication by 0_S annihilates:

$$0_S \otimes a = 0_S = a \otimes 0_S$$

Remark 3.2. Semirings are sometimes referred to as **Rigs** as they are **rings** without the **negatives**.

All rings are semirings, but we will be concerned instead with an orthogonal class of semirings - *idempotent semirings*.

Definition 3.3. A semiring S is **idempotent** if for all $a \in S$

$$a + a = a$$

Remark 3.4. Idempotent semirings are, in a sense, as far from rings as a semiring can be. The only semiring which is both idempotent and a ring is the zero semiring $\{0\}$.

There are many examples of idempotent semirings.

Example 3.5. The **boolean semiring**, denoted \mathbb{B} , is the set $\{0, 1\}$ equipped with operations:

$$a \oplus b = a \text{ OR } b$$

$$a \otimes b = a \text{ AND } b$$

and units:

$$0_{\mathbb{B}} = 0 \quad 1_{\mathbb{B}} = 1$$

The boolean semiring appears critically and frequently in computer science.

Example 3.6. The **tropical min-plus semiring**, denoted \mathbb{T} , is the set $\mathbb{R} \cup \{\infty\}$ equipped with operations

$$a \oplus b = \min(a, b)$$

$$a \otimes b = a + b$$

and units

$$0_{\mathbb{T}} = \infty \quad 1_{\mathbb{T}} = 0$$

The geometry associated with the tropical semiring is a rich field of study. It has created many powerful tools to address both abstract problems: such as the classification of isotopy classes of real curves of degree seven [26]; as well as practical applications such as enabling statistical methods in the field of phylogeny [27], train scheduling, hyperplane arrangements, and many other problems which may be read about in [28] [29][30].

Example 3.7. A closely related idempotent semiring is the **max-min semiring**. The max-min semiring is the set $[0, \infty]$ equipped with operations

$$a \oplus b = \max(a, b)$$

$$a \otimes b = \min(a, b)$$

and units

$$0_{[0, \infty]} = 0 \quad 1_{[0, \infty]} = \infty$$

In this paper we will deal with idempotent semirings which satisfy a somewhat strict condition, being closed under arbitrary infinite sums. Here the orthogonality with rings becomes apparent. Most rings that appear in practical applications are not closed under arbitrary infinite sums, however many idempotent semirings are.

Definition 3.8. A semiring is **complete** if it has infinite sums and satisfies infinite distributivity. That is, for any index set I , there is an infinitary sum operation \sum_I , such that:

$$a \otimes \left(\sum_{i \in I} a_i \right) = \sum_{i \in I} (a \otimes a_i)$$

$$\left(\sum_{i \in I} a_i \right) \otimes a = \sum_{i \in I} (a_i \otimes a)$$

Example 3.9. The tropical semiring, the max-min semiring, and the boolean semiring are all complete semirings.

Definition 3.10. Let $(S, \otimes, \oplus, 0_S, 1_S)$ be a complete semiring and X a set. We say that the **matrix semiring over X with coefficients in S** , denoted $M_X(S, \otimes, \oplus, 0_S, 1_S)$ – sometimes shortened to $M_X(S)$ – is the set of functions:

$$X \times X \rightarrow S$$

with addition defined pointwise:

$$(f \oplus g)(a, b) = f(a, b) \oplus g(a, b)$$

and multiplication:

$$(f \otimes g)(a, b) = \bigoplus_{c \in X} f(a, c) \otimes g(c, b)$$

The additive unit is the function:

$$0_{M_X(S, \otimes, \oplus, 0_S, 1_S)}(a, b) = 0_S$$

And the multiplicative unit is the function:

$$1_{M_X(S, \otimes, \oplus, 0_S, 1_S)}(a, b) = \begin{cases} 1_S & a = b \\ 0_S & a \neq b \end{cases}$$

Any function $f \in M_X(S)$ can be represented in a $|X|$ -dimensional matrix with entries in S . For this reason, we may often use matrix notation to represent elements of $M_X(S)$. For example, if $A \in M_X(S)$, we will denote $A(i, j)$ as $A_{i,j}$ for $i, j \in X$.

Remark 3.11. If X is a finite set then we do not need to require that S is a complete semiring however in Section 5 and Section 6 when we introduce semirings to model delay tolerant networking we will primarily be working with semirings of the form $M_{\mathbb{R}}(S)$, and so we will primarily work over complete semirings.

Proposition 3.12.

If S is a complete semiring, then $M_X(S)$ is complete.

If S is an idempotent semiring, then $M_X(S)$ is idempotent.

We encourage readers interested in learning more about idempotent semirings, ordered semirings, and non-ring semirings in general to read the texts [13] and [31].

4. BACKGROUND: THE ALGEBRAIC PATH PROBLEM

With an appropriate choice of semiring, finding the weights of shortest paths in a network turns into solving a particular matrix equation.

Definition 4.1. Let G be a directed (multi)graph with vertex set V , equipped with a weight function

$$w : E \rightarrow S$$

where S is some semiring. We can associate with this graph a matrix in $M_V(S)$, that we will call the **weighted adjacency matrix**, denoted A , where

$$A_{ij} = \sum_{(ij) \in E} w(ij)$$

That is, the (i, j) -th element of our matrix is the sum of the weights of the edges from node i to node j .

Definition 4.2. Let p be a path in our graph. The **weight** of p is the **product** of the weights of the edges in p , that is, if $p = (a_1 b_1)(a_2 b_2) \dots (a_n b_n)$:

$$w(p) = \prod_{i=1}^n w(a_i b_i)$$

Proposition 4.3. *Let P_{ij}^n be the set of paths of length exactly n in G from node i to node j .*

The (i, j) th entry of the n^{th} power of A is the sum of the weights of all paths of length exactly n between node i and node j . That is,

$$A_{ij}^n = \sum_{p \in P_{ij}^n} w(p)$$

In the context of an idempotent semiring this takes on an additional meaning. Idempotent semirings carry a natural

order, and we can use this A^n matrices to find paths of minimal weight.

Definition 4.4. Let S be an idempotent semiring. The **canonical ordering** of S is given by:

$$a \leq b \iff a \oplus b = a$$

Remark 4.5. This ordering is given in analogy with the min-plus tropical semiring where

$$a \oplus b = \min(a, b)$$

Here we get that $a = a \oplus b \iff a \leq b$.

Some authors present the reverse ordering as the canonical ordering.

Remark 4.6. Under this ordering we get that for any finite set X (and if S is complete, any infinite set)

$$\inf(X) = \sum_{x \in X} x$$

Corollary 4.7. *If G is a graph weighted in an idempotent semiring, then*

$$A_{ij}^n = \inf_{p \in P_{ij}^n} w(p)$$

We would like to find the least weight between two nodes over any path. That is:

$$\inf_{n \in \mathbb{N}} \inf_{p \in P_{ij}^n} w(p)$$

Remark 4.6 gives us a method to find this minimal weight.

Definition 4.8. The **Kleene star** of a matrix A , denoted A^* is:

$$A^* = I + A + A^2 + A^3 + \dots = \sum_{n=0}^{\infty} A^n$$

Corollary 4.9. *The infimum over all weights of paths between nodes i and j is the i, j th entry of the Kleene star A , that is, if P_{ij} is the set of all paths between nodes i and j , we get:*

$$A_{ij}^* = \inf_{p \in P_{ij}} w(p)$$

Remark 4.10. When the canonical ordering on S is not totally ordered this does not exactly correlate to the “minimal weighted path”, instead A_{ij}^* should be thought of as the minimal weight achievable by a sum of paths.

We will develop a semiring in the next section that encodes contact windows—times in which nodes can communicate. The entry A_{ij}^* will then give us all possible times that the nodes i and j can communicate, possibly using several paths. Our contact window semiring will not be totally ordered, so there may not be a single path p such that $w(p) = A_{ij}^*$. Instead one should think of A_{ij}^* as the weight achievable if one has the option of traversing any path in the graph.

In order to calculate A^* we may potentially need to calculate an infinite sum. While there are many possible conditions to impose which can force the sum needed to compute A^* will converge in a finite number of steps (see [12] for further details), in general we will compute the finite sums which converge towards A^* , we call these the cumulants of A .

Definition 4.11. We say that the n th cumulant of a matrix A is the matrix:

$$A^{(n)} = I + A + A^2 + \dots + A^{n-1} + A^n = \sum_{i=0}^n A^i$$

When dealing with idempotent semirings there is a useful computational identity one can use to quickly calculate the cumulants of a matrix.

Proposition 4.12.

$$A^{(n)} = (I + A)^n$$

Definition 4.13. We say that a matrix A stabilizes in n steps if

$$A^* = A^{(n)}$$

The task of finding A^* from a matrix A is called the **algebraic path problem**. Many problems in computer science can be reduced to an instance of the algebraic path problem. A good introductory text on computational methods and applications of solving the algebraic path problem is the text [12]. A more theoretical view into the connection between graph problems and semirings is the text [13].

We will spend the rest of this paper developing semirings and methods from those semirings to apply this algebraic framework to delay tolerant networking.

5. SEMIRING MODELLING OF COMMUNICATION WINDOWS

In this section we will develop a semiring which answers the question “When can two nodes talk in a delay tolerant network?” We will then use this semiring as a basis to analyze the amount of storage necessary in a network to enable a desired transmission duration.

Any model that seeks to capture the behavior of a delay tolerant network needs to be able to describe:

1. **Contact windows:** As nodes move, obstructions and downtime may appear. As a result nodes have certain windows in which they can communicate.
2. **Delays:** Our primary application is in the analysis of communications for deep space networks. At a solar system scale large delays begin to appear. For instance, it can take between three and twenty three minutes for a message sent from Earth travelling at the speed of light to reach Mars. As we expand deeper into space we need our methods and models to be able to handle increasing communication delays.
3. **Store and Forward behavior:** Satellites may not act *exclusively* as relays. They are capable of storing messages to bridge gaps in communications.

There are models which can handle contact windows, such as the TVG model introduced in [32]; and there are models to handle contact windows with delays, such as the bent pipe semiring (aka the “propagation delay semiring” in [32]).

In this section we will introduce a computable semiring which can be used to model all three necessary behaviors of a delay tolerant network. The composite behaviors possible in networks that allow for store and forward behavior arise

as multiplicative words in this semiring. In this model, our question of “When can two nodes talk in a delay tolerant network?” reduces to solving a matrix equation over this semiring. This approach gives us an algebraic toolkit to analyze these delay tolerant networks.

Definition 5.1. The **Universal Contact Semiring**, which was first introduced using a different formulation in [32], is the subsemiring of $M_{\mathbb{R}}(\mathbb{B})$ consisting of upper triangular matrices. That is, matrices over the boolean semiring where $A_{ij} = 0$ if $i > j$.

Remark 5.2. As we will see, the universal contact semiring presented here focuses on arrival times of messages, whereas the version in [32] focuses on delays. These perspectives are isomorphic, but the multiplication rule presented in [32] is significantly more complicated than the matrix multiplication used here. This reformulation is significant enough to motivate the separate and novel presentation given here.

Remark 5.3. The canonical order of the universal contact semiring can be understood as follows:

Let A and B be two matrices in the universal contact semiring. The statement that $A \leq B$ with respect to the canonical ordering on the universal contact semiring, is the statement that: “If I can send a message at time i and have it arrive at time j using the path represented by B , then I can send a message at time i and have it arrive at time j using the path represented by A .”

That is, if $A \leq B$ then A has a larger “contact window”. This follows from considering Remark 4.5 and recalling that addition in the Boolean semiring is a logical OR.

If we have a matrix with entries in the universal contact semiring, then computing the Kleene star of that matrix is the problem of finding all possible send and receive times in the corresponding contact graph.

The universal contact semiring is too vast to allow every element to be represented in a computer. Indeed if one looks at upper triangular matrices where the support (non-zero entries) encodes the picture of a fractal or some other endlessly detailed pattern of 0’s and 1’s, then there is no hope to also perform operations on these, such as addition and multiplication. However, we note (1) for most practical applications, arbitrarily complex elements of the UCS can be approximated in finite space up to any desired precision and (2) the subsemirings encountered when modelling delay tolerant networks do not require approximation and admit simple closed form algebraic representations.

To (1)’s point, the paper [33] provides algorithms and theoretical guarantees on fidelity for representing an $\mathbb{R} \times \mathbb{R}$ matrix (with Boolean entries) using a “pixelated” version—thus motivating the notion of a *pixel array*. The paper [33] goes on to provide robust and efficient algorithms for multiplying finite collections of pixelated $\mathbb{R} \times \mathbb{R}$ matrices.

This paper will be primarily concerned with embracing strategy (2), whose goal is to find computable subsemirings that encapsulate the behaviors specific modelling DTN. To that end, we identify certain “building block” $\mathbb{R} \times \mathbb{R}$ matrices that capture the behavior of contact graph routing. We will then see that we can describe these building blocks, as well as all compositions of these building blocks in simple to compute closed forms without needing approximation.

As we now describe, these building block matrices are built to

model **contact windows**—a boolean encapsulation of when messages can be sent between two nodes, without regard to data rate—and **storage**—the ability for a node to bridge gaps between contact windows.

In Section 6 these building blocks are then generalized to model transmission duration and other, more realistic, demands of DTN architectures.

In our first, most elementary treatment, a contact window should be regarded as an interval of time where a message can be sent, along with a fixed delay associated with sending the message. We denote a contact window in the form: $([s, e] : \omega)$ where s is the start of the interval in which a message can be sent, e is the end of that interval, and ω is the fixed time delay associated with that contact.

In the universal contact semiring, we may associate to a contact window $([s, e] : \omega)$ a matrix X of the form:

$$X_{ij} = \begin{cases} 1 & j = i + \omega, i \in [s, e] \\ 0 & \text{otherwise} \end{cases}$$

An example is depicted below in Figure 3.

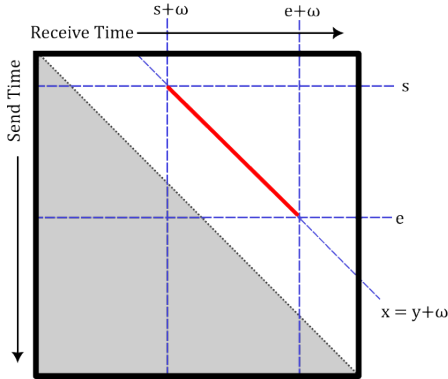


Figure 3: The boolean matrix associated with a contact, $([s, e] : \omega)$

Notation 1. We will refer to matrix associated to a contact window by the contact window itself. So we will say $([s, e] : \omega)_{ij}$ to refer to the i, j th element of the matrix associated with the contact window $([s, e] : \omega)$, i.e.,

$$([s, e] : \omega)_{ij} = X_{ij}$$

$([s, e] : \omega)_{ij}$ is 1 if and only if we can send a message along our contact window at time i and receive it at time j .

Unlimited storage can be modelled via the upper triangular matrix:

$$S_{ij} = \begin{cases} 1 & j \geq i \\ 0 & j < i \end{cases}$$

An example is depicted below in Figure 4.

This encoding reflects the observation that with infinite storage and uninterrupted connectivity, a message at any time in the past can be sent to any time in the future.

Using contact windows and storage matrices we can build an algebraic object that we can use to analyze contact graphs.

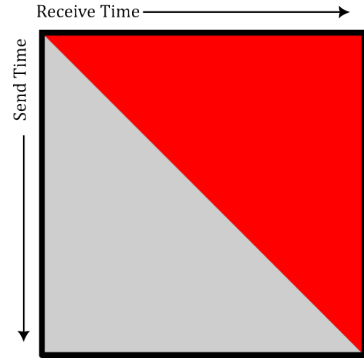


Figure 4: The storage matrix, S

Now that we have defined the matrices associated to contact windows and storage, we can examine their compositions. For instance the times in which we can utilize a contact and then store are given by multiplying the corresponding contact window and storage matrices.

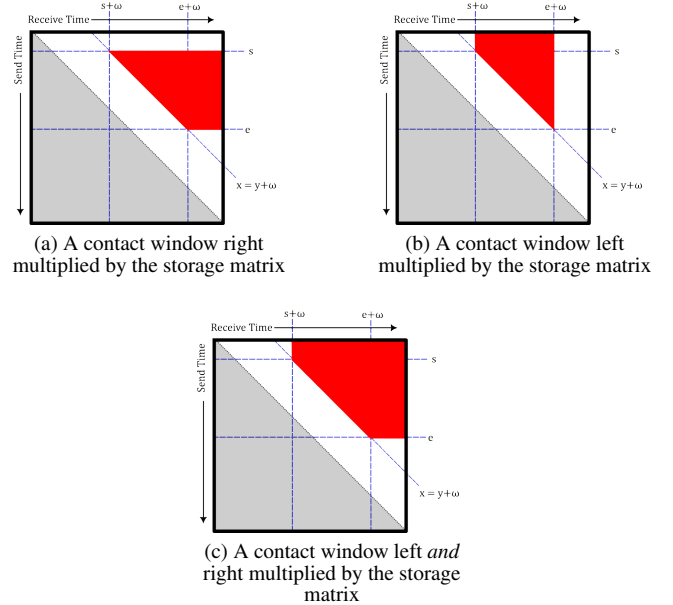


Figure 5: Various combinations of contact windows and storage matrices

Any semiring which contains these two behaviors then gives us a method in which we can find the possible windows that we may communicate across a delay tolerant network.

We examine the smallest semiring which contains both contacts and the storage matrix.

Definition 5.4. The **Store-and-Forward Semiring** is the subsemiring of the universal contact semiring consisting of finite sums and products of contacts, $([s, e] : \omega)$, and the storage matrix S .

Remark 5.5. Given a set of contacts $\mathcal{C} = \{C_{AB}^{s,e}\}$ with an associated time delay function $\omega : \mathcal{C} \rightarrow [0, \infty)$, we can build a matrix, X , over the store-and-forward semiring where the AB entry of X is the sum of the contact windows for contacts

between nodes A and B in our network, i.e.

$$X_{AB} = \sum_{C_{AB}^{s,e} \in \mathcal{C}} ([s, e] : \omega(C_{AB}^{s,e}))$$

If a node A has storage capabilities then we add

$$X_{AA} = S$$

The matrix X gives us a method of analyzing the communications possible in our contact graph. The AB entry of X^n will be a $\mathbb{R} \times \mathbb{R}$ boolean matrix with a 1 at entry i, j if there is a route in n contact or storage steps from node A to node B that allows one to send a message at time i and receive at time j .

The AB entry of the Kleene star of X , X_{AB}^* tells us all pairs of send and receive times that can be attained over any path from node A to node B in our network.

In order to compute X^n or X^* we need to know how to compute finite additions and multiplications of contact windows and the storage matrix. Adding two of these matrices is just taking the union of their support. We will see that all of the basic shapes that can occur in these matrices are simple polygons, and so their union can be computed efficiently. We will show that in this semiring we can also quickly compute the multiplication of elements.

The Store-and-Forward semiring has a surprisingly simple algebraic structure. Multiplicative sentences in this semiring can be computed quickly, and easily, and can be summarized with just five pieces of information.

Theorem 5.6. *Every element of the Store-and-Forward semiring can be written as a finite sum of elements in either the form of a contact*

$$([s, e] : \omega)$$

or a *nevada*

$$([s_1, e_1] : \omega_1)S([s_2, e_2] : \omega_2)$$

Remark 5.7. We call elements of the form

$$([s_1, e_1] : \omega_1)S([s_2, e_2] : \omega_2)$$

nevadas, as their general form is in the shape of the state of Nevada. We indicate these elements as nevadas with a lower case n to avoid any confusion with the state of Nevada.

Proof. We note that the product of two contact windows is a contact window:

$$\begin{aligned} ([s_1, e_1] : \omega_1) * ([s_2, e_2] : \omega_2) &= \\ ([s_1, e_1] \cap [s_2 - \omega_1, e_2 - \omega_1] : \omega_1 + \omega_2) \end{aligned}$$

The storage matrix is multiplicatively idempotent:

$$S * S = S$$

The identity matrix can be written as the contact window

$$((-\infty, \infty) : 0)$$

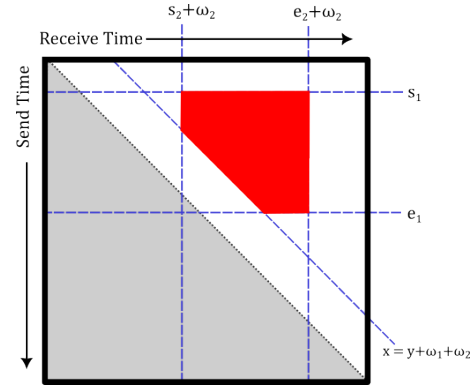


Figure 6: The general form of an element of the store-and-forward semiring of the form $([s_1, e_1] : \omega_1)S([s_2, e_2] : \omega_2)$ is the shape carved out by five hyperplanes: $y = s_1$, $y = e_1$, $x = s_2 + \omega_2$, $x = e_2 + \omega_2$, and $x = y + \omega_1 + \omega_2$

and that the following S -conjugation formula holds

$$S * ([s, e] : \omega) * S = ((-\infty, e] : 0)S([s, \infty) : \omega)$$

From these three facts we can reduce any multiplicative strings of contact windows and storages into either a single contact window, or a nevada. \square

Remark 5.8. We can uniquely determine a nevada with five numbers, as each nevada can be written in the canonical form:

A nevada of the form

$$([s_1, e_1] : \omega_1)S([s_2, e_2] : \omega_2)$$

is equal to the nevada

$$([s_1, \min(e_1, e_2 - \omega_1)] : 0)S([\max(s_1, s_2 - \omega_1), e_2 - \omega_1] : \omega_1 + \omega_2)$$

From this we note that each nevada admits a unique form

$$([s_1, e_1] : 0)S([s_2, e_2] : \omega)$$

with $s_2 \geq s_1$ and $e_1 \leq e_2$.

This tells us that the communication window of any path through a contact network, no matter how complicated, can be described in at most five numbers.

Example 5.9. Consider the product:

$$([0, 10] : 5)SS([3, 6] : 2)S([1, 8] : 1)([0, 8] : 2)$$

We can rewrite this

$$\begin{aligned} &([0, 10] : 5)SS([3, 6] : 2)S([1, 8] : 1)([0, 8] : 2) \\ &= ([0, 10] : 5)SS([3, 6] : 2)S([1, 7] : 3) \\ &= ([0, 10] : 5)S([3, 6] : 2)S([1, 7] : 3) \\ &= ([0, 10] : 5)((-\infty, 6] : 0)S([3, \infty) : 2)([1, 7] : 3) \\ &= ([0, 1] : 5)S([3, 5] : 5) \end{aligned}$$

These formulas give us a quick computational tool to determine all possible communication times within a contact graph.

6. SEMIRING MODELLING OF TRANSMISSION DURATIONS UNDER LIMITED STORAGE

As we build satellite networks, an important problem is that of finding the minimal amount of storage necessary to give us a certain amount of data throughput in our network. Hard drives take weight, and excess weight is anathema to satellites.

In this section we will make one simplifying assumption. We will assume that the bandwidth across each link in our network is identical, or very nearly so. In a satellite network this is not an unreasonable assumption, and the method build in this paper can be expanded to apply to networks with non-constant or non-identical bandwidths - however quite a bit of bulk must be added to the theorems to handle the loss of this assumption: rather than the simple formula we will derive by the end of this section, the storage requirements and throughput possible become solutions of tropical polynomial equations of potentially high degree.

Assuming identical bandwidth, it then is enough to model *transmission duration*. We will introduce a semiring which models both limited storage capabilities as well as limited transmission durations, and then use the properties of that semiring to determine the minimal amount of storage necessary in our network in order to obtain maximal transmission durations.

Remark 6.1. All of our units in this section will be units of time.

We call **transmission duration** the amount of time we can contiguously stream a message along a path in our contact network and we call **storage** the total amount of time that a message must buffer at each node. Our notion of storage can be thought of as the “layover time” that a message experiences along a route.

Assuming a constant bandwidth we can use these numbers to determine the amount of data transmit-able in bits (bandwidth times transmission duration), and the amount of storage in bits necessary at each node (the maximum of bandwidth times the temporal storage and the data transferred).

While the constructions introduced may be technical, the result of this technical analysis will be that we can summarize these transmission duration enhanced contact matrices and their multiplication with seven numbers, as we will see in remark 6.15. This results in a simple and computable method for modeling and analysing the transmission durations possible in a delay tolerant network.

Throughout this section we will build the tools needed to analyze an arbitrary path through a contact network. We will take as an input an ordered list of contact windows, $\{([s_i, e_i] : \omega_i)\}$ given by a path through a contact graph, and associate with it a transmission duration enhanced nevada. From this nevada we will derive the optimal transmission duration and storage requirements along that path. The method used to produce these paths can be any routing algorithm, or it can be generated by the semiring itself by using the matrix construction given in remark 5.5.

Notation 2. We will use the following conventions:

- s_i will be used to denote the start time of the i th contact window along a path
- e_i denotes the end time of the i th contact window along a

path

- ω_i denotes the delay associated with the i th contact along a path, i.e. a message sent along this contact at time t will arrive at time $t + \omega_i$

- Ω_i is the i -cumulant delay, that is:

$$\Omega_i = \sum_{n=1}^i \omega_n$$

- α_i represents the storage capability, measured in units of time, associated with the i th node in a path.

- A_i is the i -cumulant storage, that is:

$$A_i = \sum_{n=1}^i \alpha_n$$

We also define $A_0 = 0$

- We use \hat{s}_i and \hat{e}_i to represent the times that a message would need to be initially sent to achieve the start or end time of a given contact window, that is:

$$\hat{s}_i = s_i - \Omega_{i-1} \quad \hat{e}_i = e_i - \Omega_{i-1}$$

Definition 6.2. The **Transmission Duration Enhanced, Storage Limited Store-and-Forward Semiring** is the sub-semiring of $M_{\mathbb{R}}([0, \infty], \max, \min, \infty, 0)$ generated by matrices of the form of contact windows:

$$([s, e] : \omega)_{ij} = \begin{cases} e - i & j = i + \omega, i \in [s, e] \\ 0 & \text{otherwise} \end{cases}$$

for $s \in [-\infty, \infty), e \in (-\infty, \infty], \omega \in [0, \infty)$. As well as matrices representing potentially limited storage:

$$(S_{\alpha})_{ij} = \begin{cases} \infty & i \leq j \leq i + \alpha \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha \in [0, \infty]$

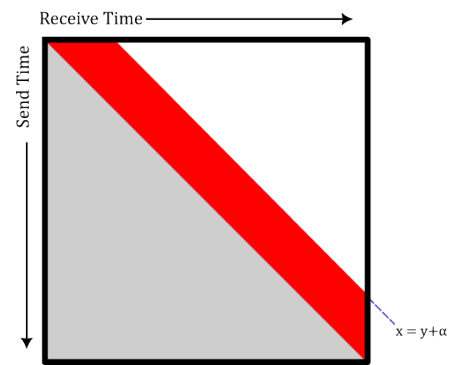


Figure 7: The support of the matrix S_{α}

Remark 6.3. Given an element X in this semiring. X_{ij} is the maximal duration of time we can stream a message such that we send the first bit at time i and it is received at time j .

Given two elements X and Y . $(X + Y)_{ij}$ is the maximum duration between X_{ij} and Y_{ij} , and $(X * Y)$ is the maximum duration of a message allowed if we compose the links described by X and Y , first taking the links in X and then in Y .

We will now analyze the composite behavior that appears in this storage limited setting, analyzing what happens as we vary the amount of storage allocated in the network, and use that to determine the minimal amount of storage necessary to attain the maximum transmission duration.

We begin by analyzing the behavior that comes from storing first, and then sending a message along a contact.

Proposition 6.4.

$$(S_\alpha([s, e] : \omega))_{ij} = e - j + \omega$$

When

$$j - \omega \in [s, e] \\ i \leq j - \omega \leq i + \alpha$$

and 0 elsewhere.

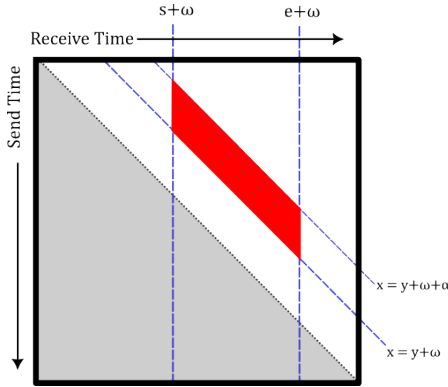


Figure 8: The support of the matrix $S_\alpha([s, e] : \omega)$

From Proposition 6.4, we can then find the behavior that occurs in a “simple nevada”.

Proposition 6.5.

$$([s_1, e_1] : \omega_1)S_\alpha([s_2, e_2] : \omega_2)_{ij} = \min(e_1 - i, e_2 - j + \omega_2)$$

When

$$i \in [s_1, e_1] \cap [s_2 - \omega_1 - \alpha, e_2 - \omega_1] \\ j - \omega_2 \in [s_2, e_2] \cap [s_1 + \omega_1, e_1 + \omega_1 + \alpha] \\ i + \omega_1 + \omega_2 \leq j \leq i + \omega_1 + \omega_2 + \alpha$$

and 0 elsewhere.

We may now induct on Proposition 6.5 to find a general case.

Theorem 6.6. Let $n \geq 2$,

$$([s_1, e_1] : \omega_1)S_{\alpha_1}([s_2, e_2] : \omega_2)S_{\alpha_2} \dots S_{\alpha_{n-1}}([s_n, e_n] : \omega_n)_{ij}$$

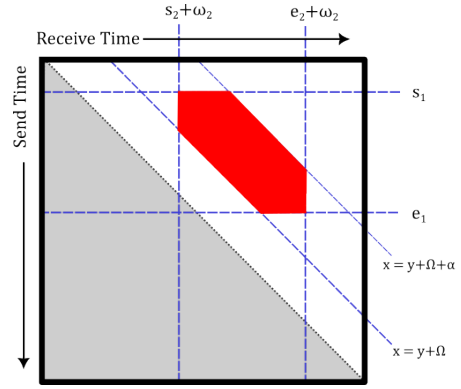


Figure 9: The support of the matrix $([s_1, e_1] : \omega_1)S_\alpha([s_2, e_2] : \omega_2)$, where $\Omega = \omega_1 + \omega_2$

is the minimum of the four quantities

$$\begin{aligned} & \min_{\ell=1}^{n-1} \left(e_\ell - \sum_{k=1}^{\ell-1} \omega_k \right) - i \\ & \min_{\ell=2}^{n-1} \left(e_\ell - \max_{k \leq \ell} \left(s_k + \sum_{f=k}^{\ell-1} \omega_f \right) \right) \\ & \min_{\ell=2}^n \left(e_\ell + \sum_{k=\ell}^n \omega_k + \sum_{k=\ell}^{n-1} \alpha_k \right) - j \\ & \min_{\ell=2}^{n-2} \left(e_\ell + \sum_{k=\ell}^{n-1} \omega_k + \sum_{k=\ell}^{n-2} \alpha_k - \max_{k=1}^{n-1} \left(s_k + \sum_{f=k}^{n-1} \omega_f \right) \right) \end{aligned}$$

For

$$\begin{aligned} i & \in \bigcap_{\ell=1}^n [s_\ell - \sum_{k=1}^{\ell-1} (\omega_k + \alpha_k), e_\ell - \sum_{k=1}^{\ell-1} \omega_k] \\ j - \omega_n & \in \bigcap_{\ell=1}^n [s_\ell + \sum_{k=\ell}^{n-1} \omega_k, e_\ell + \sum_{k=\ell}^{n-1} (\omega_k + \alpha_k)] \\ i + \sum_{\ell=1}^n \omega_\ell & \leq j \leq i + \sum_{\ell=1}^n \omega_\ell + \sum_{\ell=1}^{n-1} \alpha_\ell \end{aligned}$$

and 0 elsewhere.

Proof. Note that we get the $n = 2$ case from Proposition 6.5.

We can then induct by multiplying the above form by an element of the form $S_{\alpha_n}([s_{n+1}, e_{n+1}] : \omega_{n+1})$, whose coefficients we know from Proposition 6.4 \square

Remark 6.7. We can rewrite this in a simpler form.

First note that the fourth minimand is always greater than the third, as $j \geq \max_{\ell=1}^n s_\ell + \sum_{k=\ell}^n \omega_k$, so we may omit the fourth minimand.

Rewritten using notation 2 the three relevant minimands become:

$$\begin{aligned} & \min_{\ell=1}^{n-1} (\hat{e}_\ell) - i \\ & \min_{\ell=2}^{n-1} \left(\hat{e}_\ell - \max_{k \leq \ell} \hat{s}_k \right) \\ & \min_{\ell=2}^n (\hat{e}_\ell + \Omega_n + A_{n-1} - A_{\ell-1}) - j \end{aligned}$$

Our bounds on i and j can be written:

$$\begin{aligned} i & \in \bigcap_{\ell=1}^n [\hat{s}_\ell - A_{\ell-1}, \hat{e}_\ell] \\ j & \in \bigcap_{\ell=1}^n [\hat{s}_\ell + \Omega_n, \hat{e}_\ell + \Omega_n + A_{n-1} - A_{\ell-1}] \\ i + \Omega_n & \leq j \leq i + \Omega_n + A_{n-1} \end{aligned}$$

These three minimands give us a nice lemma to lower bound other coordinates of our matrix once we know one.

Lemma 6.8. *Let*

$$((s_1, e_1) : \omega_1) S_{\alpha_1} \dots S_{\alpha_{n-1}} ([s_n, e_n] : \omega_n)_{ij} \geq X$$

If $Y \geq 0$ then for $i' = i + Y$ and $j' = j + Y$ we get:

$$((s_1, e_1) : \omega_1) S_{\alpha_1} \dots S_{\alpha_{n-1}} ([s_n, e_n] : \omega_n)_{i'j'} \geq X - Y$$

To find the maximum possible transmission duration, we can take the nevada associated to a path

$$((s_1, e_1) : \omega_1) S_{\alpha_1} \dots S_{\alpha_{n-1}} ([s_n, e_n] : \omega_n)$$

and let our storage become unbounded. The maximum entry in the corresponding matrix is then the maximum possible transmission duration.

Theorem 6.9. *Let $((s_1, e_1) : \omega_1), ([s_2, e_2] : \omega_2), \dots, ([s_n, e_n] : \omega_n)$ be an ordered list of contact windows.*

In a non-storage constrained setting, the maximum transmission duration through the composite of these contact windows is

$$\min_{\ell=1}^n \left(\hat{e}_\ell - \max_{k \leq \ell} (\hat{s}_k) \right)$$

Proof. Let $\alpha_i \rightarrow \infty$ for each i , our three minimands become

$$\begin{aligned} & \min_{\ell=1}^{n-1} (\hat{e}_\ell) - i \\ & \min_{\ell=2}^{n-1} \left(\hat{e}_\ell - \max_{k \leq \ell} (\hat{s}_k) \right) \\ & \hat{e}_n + \Omega_n - j \end{aligned}$$

To maximize this we only need to consider the minimum possible values of i and j :

$$\begin{aligned} i & \geq s_1 \\ j & \geq \max_{\ell=1}^n \hat{s}_\ell + \Omega_n \end{aligned}$$

Put together this gives us our maximum possible transmission duration. \square

Our question then becomes what is the minimal A_{n-1} such that there exists an i, j where

$$((s_1, e_1) : \omega_1) S_{\alpha_1} ([s_2, e_2] : \omega_2) S_{\alpha_2} \dots S_{\alpha_{n-1}} ([s_n, e_n] : \omega_n)_{ij}$$

is equal to the maximal transmission duration

$$\min_{\ell=1}^n \left(\hat{e}_\ell - \max_{k \leq \ell} (\hat{s}_k) \right)$$

i.e. what is the minimal required storage necessary to attain maximal transmission duration.

Theorem 6.10. *Let an ordered series of contact windows $((s_1, e_1) : \omega_1), ([s_2, e_2] : \omega_2), \dots, ([s_n, e_n] : \omega_n)$ be given.*

Fix some τ such that $0 \leq \tau \leq \min_{\ell=1}^n \left(\hat{e}_\ell - \max_{k \leq \ell} (\hat{s}_k) \right)$

Let:

$$\nu = \max(0, \tau - \left(\min_{\ell=1}^{n-1} \hat{e}_\ell - \max_{k=1}^n \hat{s}_k \right))$$

There is an assignment of storage, $\alpha_1, \dots, \alpha_{n-1}$ such that it is possible to stream a message through the series of contact windows for a length of time τ if and only if

$$A_{n-1} \geq \nu$$

Proof. Let $\alpha_1, \dots, \alpha_n$ be such that there exists an i, j where

$$((s_1, e_1) : \omega_1) \alpha_1 \dots \alpha_{n-1} ([s_n, e_n] : \omega_n)_{ij} = \tau$$

This tells us

$$\begin{aligned} \min_{\ell=1}^{n-1} \hat{e}_\ell - i & \geq \tau \\ \min_{k=2}^n (\hat{e}_k + \Omega_n + A_{n-1} - A_{k-1}) - j & \geq \tau \end{aligned}$$

We note that $i \geq j - \Omega_n - A_{n-1}$ and $j \geq \max_{k=1}^n \hat{s}_k + \Omega_n$ and so our equation

$$\min_{\ell=1}^{n-1} \hat{e}_\ell - i \geq \tau$$

tells us

$$\min_{\ell=1}^{n-1} \hat{e}_\ell - \max_{k=1}^n \hat{s}_k + A_{n-1} \geq \tau$$

or

$$A_{n-1} \geq \tau - \left(\max_{\ell=1}^{n-1} \hat{e}_\ell - \max_{k=1}^n \hat{s}_k \right) = \nu$$

Thus if $\alpha_1, \dots, \alpha_{n-1}$ are such that we may attain a stream of length τ then $A_{n-1} \geq \nu$

Let $A_{n-1} \geq \nu$ we will show there is an assignment of $\alpha_1, \dots, \alpha_n$ and an i such that for

$$j = \max_{\ell=1}^n (\hat{s}_\ell + \Omega_n)$$

attains a transmission duration at least τ , i.e

$$((s_1, e_1] : \omega_1) S_{\alpha_1} \dots S_{\alpha_{n-1}} ([s_n, e_n] : \omega_n)_{ij} \geq \tau$$

We can see that this holds for $n = 2$, with

$$i = \min(\min_{\ell=1}^n \hat{e}_\ell - \tau, \max_{\ell=1}^n \hat{s}_\ell)$$

as our bound is $A_1 \geq \nu$, and for any $A_1 = \alpha_1 \geq \nu$ we get

$$((s_1, e_1] : \omega_1) S_{\alpha_1} ([s_2, e_2] : \omega_2)_{ij} \geq \tau$$

We can then induct on n to show that the above i and j work in general.

Let the above hold for $n - 1$, and let $A_{n-1} \geq \nu$.

Let

$$\begin{aligned} j &= \max_{\ell=1}^n (\hat{s}_\ell + \Omega_n) \\ \bar{\tau}_{\max} &= \min_{\ell=1}^{n-1} (\hat{e}_\ell - \max_{k \leq \ell} \hat{s}_k) \\ \bar{\tau} &= \min(\bar{\tau}_{\max}, \tau + \delta) \\ \bar{\nu} &= \max(0, \bar{\tau} - (\min_{\ell=1}^{n-2} \hat{e}_\ell - \max_{k=1}^{n-1} \hat{s}_k)) \\ \delta &= \max_{\ell=1}^n \hat{s}_\ell - \max_{\ell=1}^{n-1} \hat{s}_\ell \\ t &= \bar{\tau} - \tau \\ \alpha_{n-1} &= \max(0, \tau + \delta - \bar{\tau}_{\max}) \end{aligned}$$

We note that $\bar{\tau} \leq \bar{\tau}_{\max}$, and so for any $A_{n-2} \geq \bar{\nu}$ by our inductive hypothesis there is an assignment of storage $\alpha_1, \dots, \alpha_{n-2}$ and an i such that for $k = \max_{\ell=1}^{n-1} \hat{s}_\ell + \Omega_{n-1}$ we get

$$((s_1, e_1] : \omega_1) S_{\alpha_1} \dots S_{\alpha_{n-2}} ([s_{n-1}, e_{n-1}] : \omega_{n-1})_{ik} \geq \bar{\tau}$$

We will show we can use this assignment and extend it with α_{n-1} to create the proper storage assignment and message length for A_{n-1} .

To do so we first show

$$A_{n-1} - \alpha_{n-1} \geq \bar{\nu}$$

Let $\tau + \delta \geq \bar{\tau}_{\max} = \bar{\tau}$, then $\alpha_{n-1} = \tau + \delta - \bar{\tau}_{\max}$

$$\begin{aligned} A_{n-1} - \alpha_{n-1} &\geq \nu - \alpha_{n-1} \\ &= \max(-\alpha_{n-1}, \bar{\tau}_{\max} - \min_{\ell=1}^{n-1} \hat{e}_\ell + \max_{k=1}^{n-1} \hat{s}_k) \\ &= \max(-\alpha_{n-1}, \bar{\tau} - \min_{\ell=1}^{n-1} \hat{e}_\ell + \max_{k=1}^{n-1} \hat{s}_k) \\ &= \bar{\nu} \end{aligned}$$

We note

$$\bar{\tau}_{\max} - \min_{\ell=1}^{n-1} \hat{e}_\ell + \max_{k=1}^{n-1} \hat{s}_k \geq 0$$

as $\bar{\tau}_{\max} \geq \min_{\ell=1}^{n-1} \hat{e}_\ell - \max_{k=1}^{n-1} \hat{s}_k$ by definition.

We also note

$$\bar{\tau}_{\max} - \min_{\ell=1}^{n-1} \hat{e}_\ell + \max_{k=1}^{n-1} \hat{s}_k \geq \bar{\tau}_{\max} - \min_{\ell=1}^{n-2} \hat{e}_\ell + \max_{k=1}^{n-1} \hat{s}_k$$

and so

$$\bar{\tau}_{\max} - \min_{\ell=1}^{n-1} \hat{e}_\ell + \max_{k=1}^{n-1} \hat{s}_k \geq \max(0, \bar{\tau}_{\max} - \min_{\ell=1}^{n-2} \hat{e}_\ell + \max_{k=1}^{n-1} \hat{s}_k) = \bar{\nu}$$

Thus $A_{n-1} - \alpha_{n-1} \geq \bar{\nu}$.

Let $\bar{\tau} = \tau + \delta < \bar{\tau}_{\max}$, then $\alpha_{n-1} = 0$. We need to show that in this case $A_{n-1} \geq \bar{\nu}$

$$\begin{aligned} A_{n-1} &\geq \nu \\ &= \max(0, \tau - \min_{\ell=1}^{n-1} \hat{e}_\ell + \max_{k=1}^n \hat{s}_k) \\ &= \max(0, \tau + \delta + \max_{k=1}^{n-1} \hat{s}_k - \min_{\ell=1}^{n-1} \hat{e}_\ell) \\ &= \max(0, \bar{\tau} + \max_{k=1}^{n-1} \hat{s}_k - \min_{\ell=1}^{n-1} \hat{e}_\ell) \\ &= \bar{\nu} \end{aligned}$$

As $\bar{\tau} \geq \tau$ we may apply lemma 6.8, and we get that for $\bar{i} = i + t$ and $\bar{k} = k + t$ we have

$$((s_1, e_1] : \omega_1) S_{\alpha_1} \dots S_{\alpha_{n-2}} ([s_{n-1}, e_{n-1}] : \omega_{n-1})_{\bar{i}\bar{k}} \geq \tau$$

We will now show

$$S_{\alpha_{n-1}} ([s_n, e_n] : \omega_n)_{\bar{k}j} \geq \tau$$

once we do so our proof is complete, as then the product

$$((s_1, e_1] : \omega_1) S_{\alpha_1} \dots S_{\alpha_{n-1}} ([s_n, e_n] : \omega_n)_{\bar{i}j} \geq \tau$$

We examine Proposition 6.4 to show this last detail.

We note

$$e_n - j + \omega_n = \hat{e}_n - \max_{\ell=1}^n (\hat{s}_\ell + \Omega_n) \geq \tau$$

So it remains to show that \bar{k} and j are in the proper bounds.

The only nontrivial bound is:

$$\bar{k} \leq j - \omega_n \leq \bar{k} + \alpha_{n-1}$$

We note as $\bar{\tau}_{\max} \geq \tau$

$$\bar{k} + \omega_n = \max_{\ell=1}^{n-1} \hat{s}_\ell + \Omega_n + \bar{\tau} - \tau \leq \max_{\ell=1}^{n-1} \hat{s}_\ell + \Omega_n + \delta = j$$

For the upper bound we note that $t + \alpha_{n-1} = \delta$ and that $k + \delta = j$, and we see

$$\begin{aligned}\bar{k} + \omega_n + \alpha_{n-1} &= k + t + \omega_n + \alpha_{n-1} \\ &= k + \omega_n + \delta \\ &= j\end{aligned}$$

And so there is an assignment of storage for every $A_{n-1} \geq \nu$ which attains message length τ .

And so by induction our converse holds. \square

The inductive step in the converse can be used as a recursive algorithm to find the specific storage allocation necessary to attain a given transmission duration.

The previous theorem gives us the minimal total storage requirement, but in many applications it is important to know *where* the storage is needed.

We can find bounds on not only A_{n-1} , but each α_i .

Theorem 6.11. *Let an ordered series of contact windows $([s_1, e_1] : \omega_1), ([s_2, e_2] : \omega_2), \dots, ([s_n, e_n] : \omega_n)$ be given.*

Fix some τ such that $0 \leq \tau \leq \min_{\ell=1}^n \left(\hat{e}_\ell - \max_{k \leq \ell} (\hat{s}_k) \right)$

Let:

$$\begin{aligned}\gamma_1 &= \max\left(0, \tau - \left(\min_{\ell=1}^{n-1} (\hat{e}_\ell) - \max(\hat{s}_1, \hat{s}_2)\right)\right) \\ \gamma_i &= \max\left(0, \tau - \left(\min_{\ell=1}^{n-1} \hat{e}_\ell - \max_{j=1}^{i+1} \hat{s}_j\right)\right) \\ &\quad + \min\left(0, \tau - \left(\min_{j=i}^{n-1} \hat{e}_j - \max_{\ell=1}^n \hat{s}_\ell\right)\right)\end{aligned}$$

For any assignment of storage such that we may stream a message for time τ we know

$$\alpha_i \geq \max(0, \tau - \gamma_i)$$

For $i \geq 2$ and

$$\alpha_1 \geq \gamma_1$$

Proof. Let $\alpha_1, \dots, \alpha_n$ be such that there exists an i, j where

$$([s_1, e_1] : \omega_1) \alpha_1 \dots \alpha_{n-1} ([s_n, e_n] : \omega_n)_{ij} = \tau$$

This tells us

$$\begin{aligned}\min_{\ell=1}^{n-1} \hat{e}_\ell - i &\geq \tau \\ \min_{k=2}^n (\hat{e}_k + \Omega_n + A_{n-1} - A_{k-1}) - j &\geq \tau\end{aligned}$$

We know that $i \geq \max_{k=1}^n \hat{s}_k - A_{k-1}$ which tells us

$$\min_{\ell=1}^{n-1} \hat{e}_\ell - \max_{k=1}^n (\hat{s}_k - A_{k-1}) \geq \tau$$

As this holds for the maximum value, it holds for all k , and we get

$$\min_{\ell=1}^{n-1} \hat{e}_\ell - \hat{s}_k + A_{k-1} \geq \tau$$

or

$$A_{k-1} \geq \tau - \left(\min_{\ell=1}^{n-1} \hat{e}_\ell - \hat{s}_k\right)$$

As each α_i is nonnegative we get $A_{k-1} \geq 0$, and so

$$A_{k-1} \geq \max\left(0, \tau - \left(\min_{\ell=1}^{n-1} \hat{e}_\ell - \hat{s}_k\right)\right)$$

For each $j \leq k$ we get $A_{j-1} \leq A_{k-1}$ which allows us to refine this inequality further

$$A_{k-1} \geq \max\left(0, \tau - \left(\min_{\ell=1}^{n-1} \hat{e}_\ell - \max_{j=1}^k \hat{s}_j\right)\right)$$

As $\alpha_1 = A_1$ we get

$$\alpha_1 \geq \max\left(0, \tau - \left(\min_{\ell=1}^{n-1} \hat{e}_\ell - \max(\hat{s}_1, \hat{s}_2)\right)\right) = \gamma_1$$

Similarly we note that $j \geq \max_{k=1}^n \hat{s}_k + \Omega_n$ and so we get

$$\min_{k=2}^n (\hat{e}_k + \Omega_n + A_{n-1} - A_{k-1}) - \left(\max_{\ell=1}^n \hat{s}_\ell + \Omega_n\right) \geq \tau$$

As this is true for minimal k we get

$$(\hat{e}_k + \Omega_n + A_{n-1} - A_{k-1}) - \left(\max_{\ell=1}^n \hat{s}_\ell + \Omega_n\right) \geq \tau$$

for $2 \leq k \neq n$. We can rewrite this

$$A_{k-1} \leq A_{n-1} + \hat{e}_k - \max_{\ell=1}^n \hat{s}_\ell - \tau$$

We also note that if $j \geq k$ then $A_{k-1} \leq A_{j-1}$, which allows us to refine this inequality even further

$$A_{k-1} \leq A_{n-1} + \min_{j=k}^{n-1} \hat{e}_j - \max_{\ell=1}^n \hat{s}_\ell - \tau$$

We also know that $A_{k-1} \leq A_{n-1}$ so we may write

$$A_{k-1} \leq \min(A_{n-1}, A_{n-1} + \min_{j=k}^{n-1} \hat{e}_j - \max_{\ell=1}^n \hat{s}_\ell - \tau)$$

$$A_{k-1} \leq A_{n-1} + \min\left(0, \min_{j=k}^{n-1} \hat{e}_j - \max_{\ell=1}^n \hat{s}_\ell - \tau\right)$$

$$A_{k-1} \leq A_{n-1} - \max\left(0, \tau - \left(\min_{j=k}^{n-1} \hat{e}_j - \max_{\ell=1}^n \hat{s}_\ell\right)\right)$$

For $k \geq 2$ we know that $\alpha_k = A_k - A_{k-1}$, from the above inequalities we get:

$$\begin{aligned}\alpha_k &\geq \max\left(0, \tau - \left(\min_{\ell=1}^{n-1} \hat{e}_\ell - \max_{j=1}^{k+1} \hat{s}_j\right)\right) \\ &\quad + \max\left(0, \tau - \left(\min_{j=k}^{n-1} \hat{e}_k - \max_{\ell=1}^n \hat{s}_\ell\right)\right) - A_{n-1}\end{aligned}$$

or

$$\alpha_k \geq \gamma_k - A_{n-1}$$

We also know that $\alpha_k \geq 0$, so we get

$$\alpha_k \geq \max(0, \gamma_k - A_{n-1})$$

□

Example 6.12. Consider the path

$$([0, 3] : 0)S_{\alpha_1}([3, 4] : 0)S_{\alpha_2}([2, 7] : 0)$$

Our maximum transmission duration is $\tau = 1$

For $\tau = 1$, we get $\nu = \max(0, 1 - 3 + 3) = 1$

We get

$$\begin{aligned}\gamma_1 &= \max(0, 1 - 3 + 3) = 1 \\ \gamma_2 &= \max(0, 1 - 3 + 2) + \max(0, 1 - 4 + 3) = 0\end{aligned}$$

And so we know

$$\begin{aligned}A_2 &\geq 1 \\ \alpha_1 &\geq 1 \\ \alpha_2 &\geq \max(0, -A_2) = 0\end{aligned}$$

Example 6.13. Consider

$$([0, 3] : 0)S_{\alpha_1}([3, 7] : 0)S_{\alpha_2}([2, 4] : 0)S_{\alpha_3}([8, 11] : 0)$$

Here we get a maximum τ of 1, and for that τ , $\nu = 1 - 3 + 8 = 6$ and

$$\begin{aligned}\gamma_1 &= \max(0, 1 - 3 + 3) = 1 \\ \gamma_2 &= \max(0, 1 - 3 + 3) + \max(0, 1 - 4 + 8) = 6 \\ \gamma_3 &= \max(0, 1 - 3 + 8) + \max(0, 1 - 4 + 8) = 11\end{aligned}$$

Which tells us

$$\begin{aligned}A_3 &\geq 6 \\ \alpha_1 &\geq 1 \\ \alpha_2 &\geq 0 \\ \alpha_3 &\geq 5\end{aligned}$$

And so we can see that for minimal storage we allocate 1 unit of storage to α_1 and 5 to α_3 .

Example 6.14. Consider

$$([0, 1] : 0)S_{\alpha_1}([1, 3] : 0)S_{\alpha_2}([1, 3] : 0)S_{\alpha_3}([2, 3] : 0)$$

Here we get a maximum τ of 1, and for that τ we have

$$\begin{aligned}\nu &= \max(0, 1 - 1 + 2) = 2 \\ \gamma_1 &= \max(0, 1 - 1 + 1) = 1 \\ \gamma_2 &= \max(0, 1 - 1 + 1) + \max(0, 1 - 3 + 2) = 1 \\ \gamma_3 &= \max(0, 1 - 1 + 2) + \max(0, 1 - 3 + 2) = 2\end{aligned}$$

From here we get

$$\begin{aligned}A_3 &\geq 2 \\ \alpha_1 &\geq 1 \\ \alpha_2 &\geq 0 \\ \alpha_3 &\geq 0\end{aligned}$$

We note that our bounds here do not tell us where the second unit of storage is allocated, but if we examine our path we can note that the allocation is ambiguous – we may set either $\alpha_1 \geq 2$ or $\alpha_2 \geq 1$ in order to attain our maximum transmission duration.

In a non-storage limited setting we do not need to remember the whole list of contacts to compute these transmission duration enhanced nevedas, instead we only need to keep track of seven numbers to determine all of the information held within the nevada.

Remark 6.15. Let A and B be ordered sequences of contact windows,

$$A = ([s_i^A, e_i^A] : \omega_i^A) \quad B = ([s_i^B, e_i^B] : \omega_i^B)$$

In a non-storage constrained setting can determine the matrices associated with A and B , as well as the maximal throughput and minimal storage requirement by just keeping track of the following seven numbers.

Let τ_X be the maximum possible throughput of sequence X , i.e.

$$\tau_X = \min_{\ell=1}^n \left(\hat{e}_\ell^X - \max_{k \leq \ell} \hat{s}_k^X \right)$$

Let ν_X be the corresponding storage requirement

$$\nu_X = \max \left(0, \tau_X - \min_{\ell=1}^{n-1} \hat{e}_\ell^X + \max_{k=1}^n \hat{s}_k^X \right)$$

Let E_X be the minimum time adjusted end point of all but the very final contact window in X

$$E_X = \min_{\ell=1}^{n-1} \hat{e}_\ell^X$$

Let S_X be the maximum start time of a contact window in X

$$S_X = \max_{\ell=1}^n \hat{s}_\ell^X$$

Let ϵ_X be the time adjusted end time of the final contact window in X

$$\epsilon_X = \hat{e}_n^X$$

Let σ_X be the time adjusted start time of the first contact window in X

$$\sigma_X = s_1^X$$

Let Ω_X be the total delay in X

$$\Omega_X = \sum_{\ell=1}^n \omega_\ell^X$$

From these seven numbers we can recover the entire structure of the corresponding matrix. The support of a sequence of contact windows X is the nevada

$$([\sigma_X, \min(E_X, \epsilon_X)] : 0)S([S_X, \epsilon_X] : \Omega_X)$$

The value at the i, j coordinate, when within the above nevada, is

$$\min(\tau_X, E_X - i, \epsilon_X - j)$$

And the storage required to obtain a transmission duration t , ($0 \leq t \leq \tau_X$) is

$$\max(0, \nu_X - \tau_X + t)$$

We can find these values for the multiplication of two such matrices as well

$$\tau_{AB} = \min(\tau_A, \tau_B, \min(E_B, \epsilon_B) - \Omega_A - S_A)$$

$$E_{AB} = \min(E_A, \epsilon_A, E_B - \Omega_A)$$

$$S_{AB} = \max(S_A, S_B - \Omega_A)$$

$$\epsilon_{AB} = \epsilon_B - \Omega_A$$

$$\sigma_{AB} = \sigma_A$$

$$\Omega_{AB} = \Omega_A + \Omega_B$$

$$\nu_{AB} = \max(0, \tau_{AB} - E_{AB} + S_{AB})$$

Thus in a non-storage constrained setting, we only need to track the above seven numbers in order to fully recreate the throughput enhanced storage matrices.

7. EXPERIMENTAL SETUP

To demonstrate our method, we constructed a scenario based upon the proposed LunaNet communications architecture. The basis for this architecture is derived from the LunaNet Interoperability Specification [34], the Interagency Operations Advisory Group Future Lunar Communications Architecture study [35], and ephemeris data from the Lunar Gateway Near-Rectilinear Halo Orbit computations [36]. Each satellite in the scenario is pulled from either of these resources, producing a total of 6 satellites. The scenario was constructed in Satellite Orbital Analysis Program (SOAP), and the contact plan data and distance information can be made available upon request.

On Earth, we identified 6 ground stations as assets. In the Deep Space Network (DSN), there are 3 ground stations in Goldstone, Madrid, and Canberra. As the DSN is a planned part of LunaNet communications, we included these three locations here. In addition to the DSN, we identified 3 potential ground sites for the Lunar Exploration Ground Stations (LEGS) antenna network. We identified locations at White Sands, Wallops, and Dongara as potential LEGS sites. Each ground station is connected to each other ground station with a 22ms internet connection at all times.

On the Lunar surface, we constructed 3 representative ground stations. The first is located near the South Pole of the moon, representing an estimate for a potential Lunar Base location. The other two are located on the far side of the moon, and each represents a Lunar Rover located inside a crater.

The input data to our algorithm comes in the form of contacts and distances between objects in the contact which is then translated to a one-way light time. Using SOAP, we extract the start and end times of line-of-sight availability between any pair of assets. These start and end times are used as the start and end of each contact interval between said assets. Similarly, we use SOAP to compute the distances between assets every 30 seconds. Then, the first entry for each distance

that falls within the contact interval is used to build a one-way¹⁵ light time that is then assumed constant for the remainder of the contact.

These final assumptions lead us to some caveats that are needed to frame our experiment. Each contact represents a potentially used contact between two assets, and does not necessarily represent an actual connection made. The advantage to this strategy is that we may analyze all potential paths that the data could potentially take through the scenario. In practice however certain paths are exclusionary. In order to make use of certain contact windows satellites may need to be realigned, removing the availability of other contact windows.

In addition, we assumed constant one-way light time values to determine the delays for each contact window. Over the day long window that we examined, the change in light time between any two assets was minimal, however in longer ranging simulations contact windows may need to be subdivided to better represent moving assets.

Our assumption of constant data rate dramatically simplifies the problem. Due to the difference in data rates arising from different modulation and coding schemes, even a constant one-way light time can represent different amounts of data to different assets. While this is easy to analyze for a single contact, transmissions over multiple hops where each hop may use a different data rate requires more care. The semiring analysis technique can be adapted to incorporate data rate information, but we use the simplifying assumption of constant data rate in order to ensure a closed algebraic form to our construction, and to give an analysis not reliant on a specific modulation or coding scheme.

8. RESULTS

C++ code implementing these theorems and methods can be found at the GitHub repository, wrberardonni/Semiring-Geometry. Both semirings described in section 5 and 6 are implemented as the CGRSemiring.

Python code implementing these theorems and methods can be found at the GitHub repository <https://github.com/TheaMAS/sat-parser>.

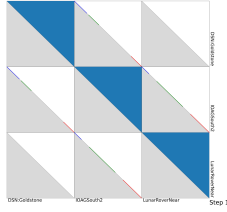
The code used to process this 15×15 contact matrix was ran on an old and ailing 2019 Lenovo Yoga C930 with a 4 GHz i7 processor. The contact graph consisted of 288 contacts, and the contact matrix stabilized at the 7th cumulant. The time necessary to compute the Kleene star of this contact matrix using the provided C++ code was 22.013 seconds.

As an example, we will analyse the communication link between IOAGSouth2 satellite and LunarRoverNear rover.

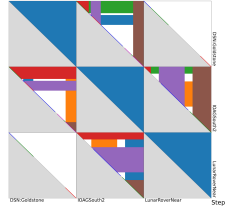
The maximum possible transmission duration from IOAG-South2 to LunarRoverNear is about 15.8 hours long, and requires 6.15 hours worth of storage.

At a high bitrate 6.15 hours of storage may be a physically unfeasible amount of storage in our network, however with no storage it is possible to send a message with a transmission duration of 10.32 hours between our lunar satellite and our lunar rover along a separate path through our contact network.

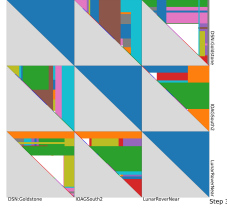
This network configuration is not symmetric however.



(a) A three by three slice of the first cumulant $A^{(1)}$.



(b) A three by three slice of the second cumulant $A^{(2)}$.



(c) A three by three slice of the third cumulant $A^{(3)}$.



(d) A three by three slice of the fourth cumulant $A^{(4)}$.

Figure 10: Three by three submatrices of the first four cumulants of the store and forward matrix A (see remark 5.5) in our experimental set up. Each colored region represents the window in time that a choice of path in our contact graph can be used.

The maximum possible transmission duration from LunarRoverNear to IOAGSouth2 is 10.32 hours long, this maximal transmission duration requires zero storage to attain. While we can increase the transmission duration from IOAGSouth2 to LunarRoverNear by adding storage into our network, we get no such benefit in the reverse direction.

Being able to detect this asymmetry is a powerful feature of the Store-and-Forward semiring. While the contacts that were used to generate the matrix were symmetric – a contact being available from node A to B meant that the same contact was available from B to A – the behavior of both delays and message forwarding allow us to see and analyse asymmetries present within the temporal behavior of the graph.

9. CONCLUSION

By analyzing the geometric structure of novel semiring models of delay tolerant networking, we were able to develop tools capable of analyzing both communication availability as well as the minimal storage requirements for a delay tolerant network.

The semirings introduced in sections 5 and 6 allow us to create both visualization tools for delay tolerant networks (see figures 10 and 11), as well as analyze potential storage requirements for the creation of the network (see figures 12 and 13).

There is far more that can be done with these semirings. The algorithms found in [12] can be used to make routing decisions for delay tolerant networks using these semirings,

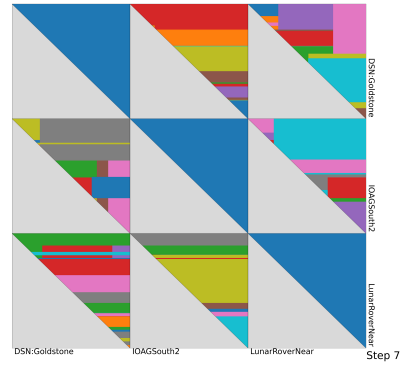


Figure 11: A three by three submatrix of the Kleene star of the store and forward matrices in our experimental set up. We can see that this network obtains coverage for almost the entire day, however certain pairs of nodes have far more “fragmented” coverage and require more paths available to maintain coverage.

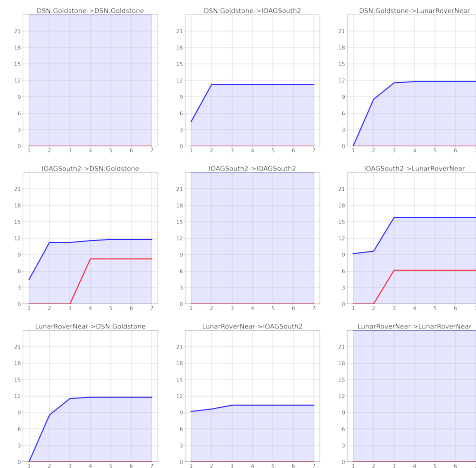


Figure 12: Maximum transmission duration (in blue) and required storage (in red) in hours versus the hop limit of a three node subset of the network.

and we believe that further analysis using this model of delay tolerant networking will give rise to future algorithms and approaches, such as in identifying network substructures and optimal decompositions.

We would also like to stress that while we introduced these methods in the context of satellite networks they are in no way specific to satellite networks. Any structure that has the core behaviors outlined at the beginning of Section 5 lends itself to analysis using the semirings introduced in this paper. In addition, as noted in [32], the Universal Contact Semiring encapsulates the behavior of many semirings commonly used in routing. As a result the framework used in this paper is well suited to modelling other network modalities, and even fused networks – our example scenario for instance is a fused satellite/internet topology.

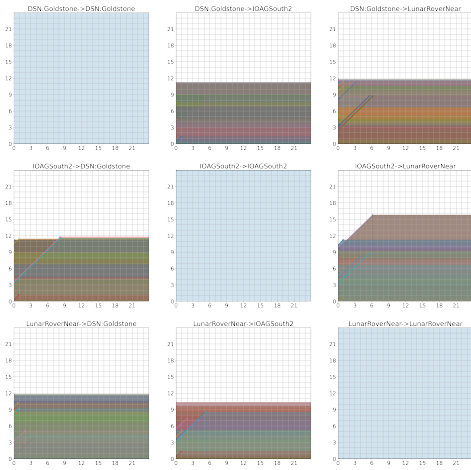


Figure 13: Transmission duration vs required storage in hours for a three node subset of our network. Each colored area represents a different path through the contact graph. We can see that even if a large amount of storage is needed for optimal transmission duration, even with little to no storage available we obtain high possible transmission durations

10. FUTURE WORK

We believe that there are many directions that this algebraic and semiring geometric perspective on delay tolerant routing may be further developed.

On practical grounds we believe that these methods could also be utilized to develop planning and analysis methods for tasks such as the shipping of livestock or spoilable material, as well as for manpower scheduling tasks such as the timing of experiments on the ISS. In addition, optimizations to the algorithm may enable storage considerations to be incorporated into DTN routing algorithms improving efficiency and reliability of selected routes.

Mathematically there are many avenues these methods can be expanded as well.

The representation of delay tolerant networks as matrices over particular semirings naturally leads to the question of whether these matrices have eigenvectors and eigenvalues and the implications of those eigenvectors and eigenvalues on the corresponding contact graphs, such as “Could we use the linear algebra of these semirings to create a ‘Delay Tolerant Katz centrality’?”

The connection with the pixel array work in [33] suggests that there are further categorical constructions that can be explored over these and related structures.

The task of computing the Kleene star of a matrix A , and solving the associated routing problems can also be phrased as solving the semiring matrix equation

$$X = AX + I.$$

While there is much developed theory in solving such linear equations over semirings, there is not a well developed sense of an algebraic geometry over semirings such as the ones in this paper. This leads to interesting theoretical questions and also practical ones. Optimizing the design of deep space satellite topologies can be expressed as parameter spaces and polynomial equations over these semirings, understanding the

structure of such spaces and how to solve such polynomial equations can be used to build further tools in the analysis and deployment of delay tolerant networks.

REFERENCES

- [1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, “RFC 4838, Delay-Tolerant Networking Architecture,” *IETF Network Working Group*, 2007. [Online]. Available: <https://tools.ietf.org/html/rfc4838>
- [2] G. Araniti, N. Bezirgiannidis, E. Berrane, I. Bisio, S. Burleigh, C. Caini, M. Feldmann, M. Marchese, J. Segui, and K. Suzuki, “Contact graph routing in dtn space networks: overview, enhancements and performance,” *IEEE Communications Magazine*, vol. 53, no. 3, pp. 38–46, March 2015.
- [3] J. A. Fraire, O. De Jonckère, and S. C. Burleigh, “Routing in the space internet: A contact graph routing tutorial,” *Journal of Network and Computer Applications*, vol. 174, p. 102884, 2021.
- [4] M. Moy, R. Kassouf-Short, N. Kortas, J. Cleveland, B. Tomko, D. Conricode, Y. Kirkpatrick, R. Cardona, B. Heller, and J. Curry, “Contact multigraph routing: Overview and implementation,” in *2023 IEEE Aerospace Conference*, 2023, pp. 1–9.
- [5] A. Hylton, M. Moy, R. Kassouf-Short, and J. Cleveland, “Multigraph-based routing in delay tolerant networks: An alternative to contact graph routing,” in *2023 32nd International Conference on Computer Communications and Networks (ICCCN)*, 2023, pp. 1–7.
- [6] A. Lindgren, A. Doria, E. Davies, and S. Grasic, “RFC 6693: Probabilistic Routing Protocol for Intermittently Connected Networks,” *IETF Network Working Group*, 2012. [Online]. Available: <https://tools.ietf.org/html/rfc6693>
- [7] S. Grasic, E. Davies, A. Lindgren, and A. Doria, “The evolution of a dtn routing protocol - prophetv2,” in *Proceedings of the 6th ACM Workshop on Challenged Networks*, ser. CHANTS ’11. New York, NY, USA: Association for Computing Machinery, 2011, p. 27–30. [Online]. Available: <https://doi.org/10.1145/2030652.2030661>
- [8] R. Lent, “Implementing a cognitive routing method for high-rate delay tolerant networking,” in *2023 IEEE Cognitive Communications for Aerospace Applications Workshop*, 2023, pp. 1–6.
- [9] M. Moore, R. Bull, S. Burleigh, J. Cook, J. Waszkiewicz, D. Cook, and J. Seif, “Reactive routing: Harnessing advanced software modems,” in *2023 IEEE Cognitive Communications for Aerospace Applications Workshop*, 2023, pp. 1–5.
- [10] A. Mody, B. Crompton, D. Tran, D. Giger, D. Simpson, D. Gormley, A. Smith, M. Kappes, D. Redelings, and T. Melodia, “Claire: Enabling heterogeneous communication network optimization for robust and resilient operations,” in *2023 IEEE Cognitive Communications for Aerospace Applications Workshop*, 2023, pp. 1–6.
- [11] D. Ta, R. Memon, J. Taggart, A. Tettamanti, S. Feaser, P. Torrado, and J. Smith, “Roaming dtn: Integrating unscheduled nodes into contact plan based dtn networks,” in *2023 IEEE Cognitive Communications for Aerospace Applications Workshop*, 2023, pp. 1–9.

- [12] J. S. Baras and G. Theodorakopoulos, "Path problems in networks," *Synthesis Lectures on Communication Networks*, vol. 3, 2010.
- [13] M. Gondran and M. Minoux, *Graphs, Dioids and Semirings: New Models and Algorithms*, ser. Operations Research/Computer Science Interfaces Series. Springer US, 2008.
- [14] Y. Wu, S. Deng, and H. Huang, "Performance analysis of epidemic routing in dns with limited forwarding times and selfish nodes," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 13, no. 3/4, pp. 254–263, July 2013. [Online]. Available: <http://dx.doi.org/10.1504/IJAHUC.2013.055474>
- [15] T. Spyropoulos, K. Psounis, and C. 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*, ser. WDTN '05. New York, NY, USA: ACM, 2005, pp. 252–259. [Online]. Available: <http://doi.acm.org/10.1145/1080139.1080143>
- [16] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, April 2006, pp. 1–11.
- [17] J. A. Fraire and J. M. Finochietto, "Design challenges in contact plans for disruption-tolerant satellite networks," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 163–169, 2015.
- [18] Consultative Committee for Space Data Systems (CCSDS), "Schedule-aware bundle routing (SABR) (blue book, recommended standard CCSDS 734.3-B-1," <https://public.ccsds.org/Pubs/734x3b1.pdf>, July 2019.
- [19] O. De Jonckère and J. A. Fraire, "A shortest-path tree approach for routing in space networks," *China Communications*, vol. 17, no. 7, pp. 52–66, 2020.
- [20] J. A. Fraire and E. L. Gasparini, "Centralized and decentralized routing solutions for present and future space information networks," *IEEE Network*, vol. 35, no. 4, pp. 110–117, 2021.
- [21] J. Segui, E. Jennings, and S. Burleigh, "Enhancing contact graph routing for delay tolerant space networking," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, December 2011, pp. 1–6.
- [22] S. Burleigh, "Interplanetary overlay network: An implementation of the dtn bundle protocol," 2007.
- [23] E. Birrane, S. Burleigh, and N. Kasch, "Analysis of the contact graph routing algorithm: Bounding interplanetary paths," *Acta Astronautica*, vol. 75, pp. 108 – 119, 2012.
- [24] 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, 2018.
- [25] E. L. Lawler, "A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem," *Management science*, vol. 18, no. 7, pp. 401–405, 1972.
- [26] O. Viro, "Patchworking real algebraic varieties," 2006.
- [27] A. Monod, B. Lin, R. Yoshida, and Q. Kang, "Tropical geometry of phylogenetic tree space: A statistical perspective," 2022.
- [28] B. Heidergott, G. J. Olsder, and J. van der Woude, *Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*. Princeton University Press, 2006. [Online]. Available: <http://www.jstor.org/stable/j.ctt7zv8k3>
- [29] D. Maclagan and B. Sturmfels, *Introduction to Tropical Geometry*, ser. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2015, vol. 161.
- [30] I. Itenberg, I. Itenberg, G. Mikhalkin, and E. Shustin, *Tropical Algebraic Geometry*, ser. Oberwolfach Seminars Series. Birkhäuser, 2007.
- [31] J. Golan, *Semirings and their Applications*. Springer Netherlands, 1999.
- [32] W. Bernardoni, R. Cardona, J. Cleveland, J. Curry, R. Green, B. Heller, A. Hylton, T. Lam, and R. Kassouf-Short, "Algebraic and geometric models for space networking," 2023.
- [33] D. I. Spivak, M. R. C. Dobson, S. Kumari, and L. Wu, "Pixel arrays: A fast and elementary method for solving nonlinear systems," 2017.
- [34] D. Israel and N. Babu, "Draft lunanet interoperability specification," July 2022. [Online]. Available: <https://ntrs.nasa.gov/citations/20220010998>
- [35] M. Cosby, W. Tai, M. Hose, F. D'Amico, J.-L. Issler, P. Jin, P. Kazakoff, M. Picard, N. Lii, M. Lanucara, A. Grop, D. Rovelli, A. R. Srinivas, H. Itoh, Y. Kaneko, D.-J. Park, D. Israel, J. Shier, and G. Iv, "The future lunar communications architecture," Interagency Operations Advisory Group Lunar Communications Architecture Working Group, Tech. Rep., January 2022.
- [36] D. Lee, "White paper: Gateway destination orbit model: A continuous 15 year nrho reference trajectory," Aug 2019. [Online]. Available: <https://ntrs.nasa.gov/citations/20190030294>



William Bernardoni is a PhD candidate at Case Western Reserve University. His focus is on the algebraic geometry of idempotent semirings. Prior to his PhD he studied computer science, and did research in statistical machine translation. His current research centers on extending ideas and theory from tropical and real algebraic geometry to work over combinatorial domains via the language of idempotent semirings.



Robert Kassouf-Short earned his PhD in mathematics from Lehigh University in 2018. He worked as a Visiting Assistant Professor of Mathematics at John Carroll University until he joined the Secure Networks, System Integration and Test Branch at NASA Glenn Research Center in 2020. His research interests lie in the intersection of abstract mathematics and real world applications. Currently, his focus is on the foundations of

networking theory and how to efficiently route data through a network using local information.



Robert Cardona is a PhD student in applied topology. He studied computer engineering and mathematics before going on to work as a software developer. He then obtained a masters at Freie Universität Berlin and continued on to study applied topology at Albany.



Brian Heller is a PhD student studying mathematics at University of Albany. Previously he worked as a software engineer and as an advisor in digital and emerging technologies. His research interests include category theory, mathematical logic, and algebraic geometry.



Justin Curry is an Associate Professor of Mathematics and Statistics at the University at Albany, SUNY. Before arriving at Albany in 2017, he was a Visiting Assistant Professor at Duke. Professor Curry earned his PhD in mathematics from the University of Pennsylvania in 2014, under the direction of Robert Ghrist. His research interests include the use of category theory in applied mathematics, with particular emphasis

on applied sheaf theory, and inverse problems in topological data analysis (TDA).



David Spivak received a PhD in mathematics from UC Berkeley in 2007; his thesis was in algebraic topology. Noticing category theory's impressive ability to organize and layer abstractions and to interconnect widely different disciplines, he set out to show that mathematics' most abstract field was also its most applicable. After spending ten years at MIT, funded in part by NASA, the US departments of defense and commerce,

and private companies, he co-founded Topos Institute, an independent non-profit research institute that works to shape technology for the public benefit by applying the language of category theory. David's work ranges from database integration to knowledge representation, from materials science to dynamical systems, all with a focus on compositionality, interoperability, and collective intelligence. He has written three books on applications of category theory.



Juan Fraire is a researcher and professor at INRIA (France) and CONICET-UNC (Argentina) and a guest professor at Saarland University (Germany). Core topics of his interest are near-Earth and deep-space networking and informatics, adding up to more than 100 published papers in international journals and leading conferences. Juan is the co-founder and chair of the Space-Terrestrial Inter-networking Workshop (STINT) and par-

ticipates in diverse joint projects with space agencies (e.g., NASA, ESA, CONAE) and companies in the space sector (e.g., D3TN, Skyloom).