Optimization and Scheduling Methodologies to Enable Low Earth Orbit Nano-satellite Communication

Michelle Song

A thesis
submitted in partial fulfillment of the
requirements for the master's degree of

Master of Science

University of Washington

2018

Committee:

Zelda B. Zabinsky, Chair

Cherry Wakayama

Youngjun Chou

Program Authorized to Offer Degree:
Industrial & Systems Engineering

University of Washington

Abstract

Optimization and Scheduling Methodologies to
Enable Low Earth Orbit Nano-satellite Communication

Michelle Song

Chair of the Supervisory Committee:
Professor Zelda B. Zabinsky
Industrial & Systems Engineering

Communications with low earth orbit (LEO) nano-satellites (nanosats) are challenging due to the short contact time intervals with ground nodes and uncertainty in successful delivery of messages due to the varying signal-to-noise ratio (SNR) in the environment. This thesis presents optimization models to enable minimum-delay store-and-forward communications between terrestrial gateways and remote users (e.g., ships) via nanosats. Optimization models are formulated to enable timely delivery of messages between nanosats and remote users. Unit-sized messages destined for remote users must be routed from gateways to nanosats to final remote destinations. The connection between nanosats and remote users may not always be well established. The uncertainty in knowing if a message needs to be sent again or was successfully delivered is modeled using a chance constraint in the optimization model. A network flow program is formulated to optimize the scheduling and routing of messages from a central command and control center (CCC) to gateways to nanosats and ultimately to the remote user. The decisions are chosen to minimize the total message delivery time while considering the nanosat contact time windows with gateways and remote users and the solar charging time windows of the nanosats. Although the scheduling and routing decisions are binary variables, the optimization models are shown to satisfy the integrality property. Therefore the relaxed network model can be solved much faster than a binary integer

problem. Results on a realistic problem are presented. Comparisons are made to consider the difference between a basic deterministic model, a model with energy constraints, and a chance-constrained model (with and without energy constraints). Comparisons are also made to simple greedy heuristics to demonstrate the value of optimization.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

To my advisors, Zelda B. Zabinsky and Cherry Wakayama - Thank you for the guidance that you have provided to me throughout my master's thesis journey. I am thankful for your patience and understanding while helping me progress through my graduate studies. I am grateful for the opportunity to have worked with two incredible role models; your passion towards research has inspired and motivated me throughout the past year.

To my committee, Youngjun Chou - Thank you for your support, advice, and guidance during my graduate studies. I am grateful for the time you have taken to review my thesis.

To the Industrial and Systems Engineering Faculty and Staff - Thank you for all your support and help throughout the past two years.

To my fellow (past and present) graduate students - Thank you for helping me through classes, research, and life in Seattle. I am grateful for all the friendships that I have formed and the guidance that everyone has given me.

To San Diego State University Research Foundation and SPAWAR - Thank you for your contribution towards my education and for granting me the opportunity to work on this research project.

To my family - Thank you for the unwavering support and faith in me throughout my education journey and for being shining examples of pursuing graduate degrees.

DEDICATION

To my family, who inspired me to continue my education

Chapter 1
INTRODUCTION

Low cost, low earth orbit nano-satellites (nanosats) are finding new applications in defense, public and commercial services [7]. Applications using nanosats include remote sensing, weather monitoring, science, and communications. A constellation of nanosats may potentially be used to relay messages between ground nodes, including gateways and remote hard-to-reach areas. This thesis discusses some current challenges in scheduling nanosats to enable timely communications, and presents several optimization models to determine routing decisions while accounting for short contact time windows and uncertainty in link quality when delivering messages.

1.1 Research Motivation

Small satellites including nanosats orbit the earth at a much lower altitude than traditional communication satellites (Figure 1.1). The low earth orbit limits nanosats to short contact time intervals with ground nodes. In this thesis, two types of ground nodes, gateways and remote users, are considered. The connectivity between gateways and remote users is achieved with a "store-and-forward" approach using nanosats which serve as message ferries from gateways to remote users. There are many ways to route messages for delivery; however messages should be delivered in a timely manner. There is also uncertainty in the stability of the connection between nanosats and remote users due to environmental and communication systems noises. Simulation software, for example, Systems Tool Kit (STK), can provide data on when a nanosat comes into contact with a remote user, however this connection is not always stable enough to successfully transmit a message [2]. Nanosats are also constrained by their small size and cannot contain large batteries. There is a maximum energy level that

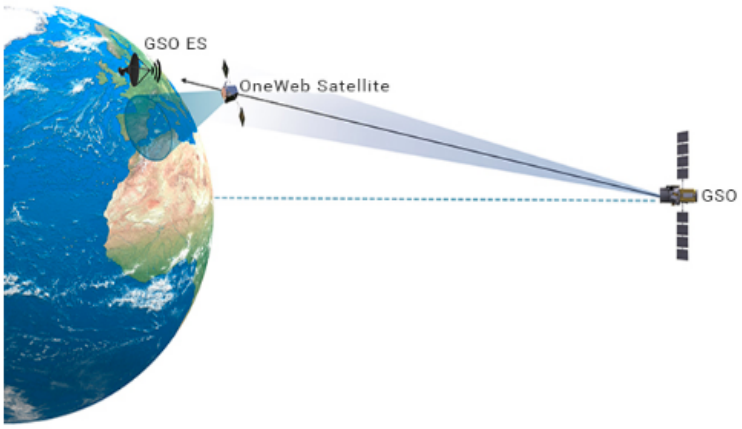

Figure 1.1: Comparison of orbit altitude between a small satellite and a traditional communication satellite [6]. GSO refers to a satellite that is in a geosynchronous orbit (orbital period matches the Earth's rotation) and ES is the earth station that receives signals from the GSO satellite.

each nanosat can store. The simulation software provides data on when there is sunlight to charge a nanosat's battery using its solar panels. A nanosat cannot send messages if its energy storage falls below a minimum threshold required for message transmission. These challenges necessitate the formulation of energy aware optimization models to derive efficient message delivery schedules for gateways and nanosats in a store-and-forward architecture.

1.2 Research Contribution

The objective of this research is to develop and analyze optimization and scheduling methods that can assist with the timely delivery of messages using nanosats. This thesis develops models to select an optimal route and schedule to minimize total message delivery times. We formulate network optimization models to optimize the scheduling and routing of messages from a central command and control center (CCC) to gateways and nanosats to remote users. Two deterministic optimization models are developed, with and without energy constraints to incorporate threshold values on the energy level in the battery. The connection between nanosats and remote users may not always be well established; the uncertainty in knowing if

a message was successfully delivered is modeled using a chance constraint in the optimization model. The decision variables for sending messages are binary variables, and in general, it is a challenge to solve integer programs quickly. The deterministic and chance-constrained optimization models formulated in this thesis are shown to fit the form of a minimum cost flow network problem so integrality is ensured and there are quick algorithms that can be used to solve these models.

Comparisons are made to consider the differences between a basic deterministic model without energy constraints and a deterministic model with energy constraints. Comparisons are also made to consider the differences between the deterministic models and the chance-constrained models (with and without energy constraints). Comparisons are also made to greedy heuristics to highlight the benefit of optimization.

1.3 Organization

Chapter 2 discusses background on nanosats, previously solved routing and scheduling problems related to nanosat constellations, simulation tools used to define nanosat constellations, and minimum cost flow network problems. Chapter 3 presents the formulation of two deterministic optimization models, with and without energy constraints, and illustrates with a small example. Chapter 3 also shows that the deterministic models can be cast as minimum cost network flow problems, ensuring integrality. Chapter 4 presents the formulation of probabilistic optimization models, gives small examples, and provides a min-cost flow representation of these models. Chapter 5 describes the results from a realistic problem. Chapter 6 compares greedy policies to the optimization policies formulated in Chapters 3 and 4 and gives simulation results of the different policies. Chapter 7 discusses a maximum flow formulation of the models discussed in Chapters 3 and 4 that can be used to analyze the maximum number of messages that can be supported by a given constellation of nanosats. Chapter 8 summarizes the overall results and conclusions and discusses future work.

Chapter 2

BACKGROUND

The small satellite and nanosat segments of the space industry have been growing rapidly in recent years. These satellites are dramatically smaller, lighter, less expensive to build and launch than their traditional counterparts. With multiple technology advances, these low-cost small satellites are finding exciting applications of interest to defense, public and commercial sectors, including remote sensing, weather monitoring, unique science experiments, and communications. In this thesis, a communication network application using nanosats where nanosats serve as message ferries between ground gateways and remote user terminals is considered. Previous work has looked into solving routing and scheduling problems involving nanosats in a given network architecture [9]. Wakayama et al. looks at a scheduling methodology for sending messages from one nanosat to multiple ground nodes [9]. In Wakayama et al., messages are assumed to be already in the nanosat and the problem is how to schedule timely delivery given energy constraints and short contact intervals [9]. In this thesis, the complete routing of messages from gateways to nanosats to remote users is addressed.

Wakayama et al. assumes the "store-and-forward" approach for message delivery [9]. The "store-and-forward" approach is when there is no communication between nanosats and any message that is sent to one nanosat must be delivered to its final destination by that same nanosat. In contrast, Cahoy et al. considers crosslinks between nanosats [4]. Crosslinks refer to nanosat-to-nanosat communication; a message sent to one nanosat can be transferred to another nanosat before being delivered to the final destination. Not all nanosats may have the capability to communicate with other nanosats. This thesis focuses on a "store-and-

forward" approach.

The Central Command and Control center (CCC) is a centralized system that has access to large power supplies and readily available internet connectivity. Gateways are terrestrial ground nodes that are part of the centralized CCC network. The CCC initiates the route for a message by first choosing a gateway on the ground. Gateways are part of a centralized internet-connected system, so messages may be broken up between different gateways. Gateways are considered to be stationary and have knowledge of the contact windows with the nanosats in the constellation. Remote users (e.g., ships) are located in regions which may not have access to large power supplies and do not have internet connectivity. This work focuses on a "store-and-forward" communication network architecture, where messages originate at the CCC and gateways, travel through nanosats, and are delivered to their final destination (a remote user) (Figure 2.1) [1].

The STK (Systems Tool Kit) software tool allows users to define a constellation of nanosats and ground nodes (gateways and remote users). The STK can determine the time duration of each nanosat-to-ground node contact window by the ground node's location and the nanosat's orbital parameters. The contact window is the time during which a nanosat can communicate with ground node. During a contact window, nanosats can send and/or receive messages to and from gateways and remote users. Figure 2.2 shows an example of a nanosat's orbit and ground contact range. It can be seen that the nanosat comes into contact with a gateway at GS-5 during its orbit. The STK can also determine the solar charging time windows for nanosats. The solar charging time windows are the periods of time when a nanosat is able to charge its batteries via sunlight using solar panels.

Nanosats can send and receive messages from both gateways and remote users. Uplink refers to gateway-to-nanosat communication and downlink refers to nanosat-to-remote user communication. Full Duplex refers to two-way communication where a nanosat can simul-

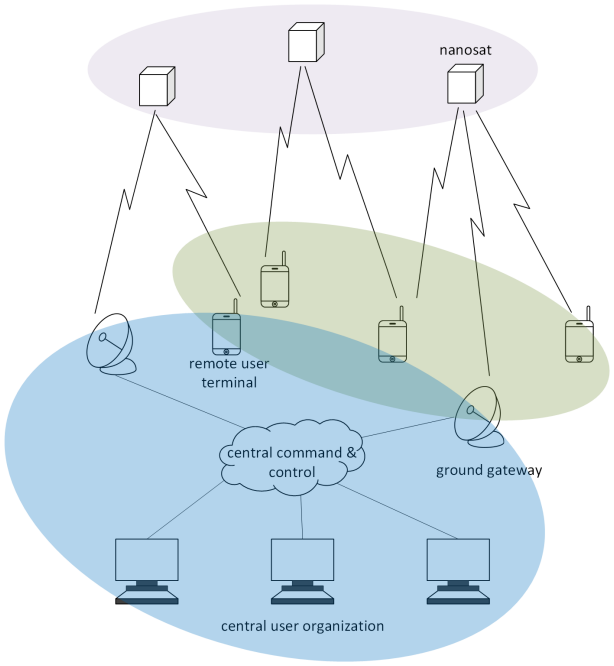

Figure 2.1: Representation of network architecture.

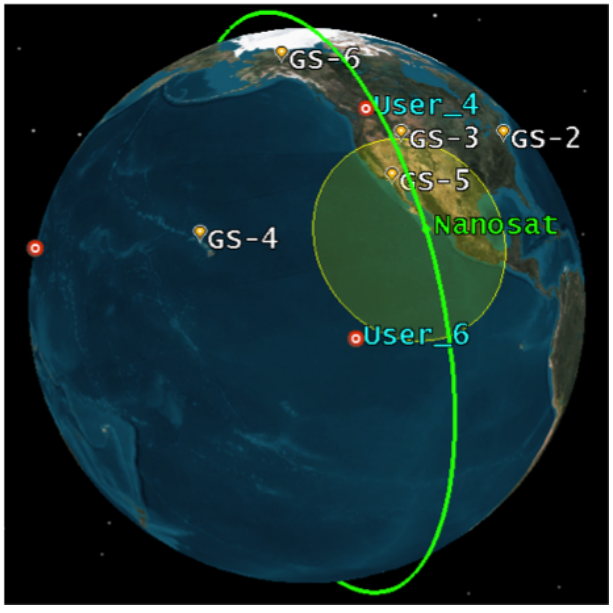

Figure 2.2: An example of a nanosat's orbit and ground contact range [9]. The nanosat cannot come in contact with ground nodes GS-2 and GS-4 at this time.

taneously send and receive messages. Half Duplex refers to one-way communication where a nanosat can either send or receive a message at a given point in time. This thesis assumes full duplex is available.

Wakayama et al. [9] shows that an optimization model of a network of remote users and a single nanosat can be represented as a minimum cost network flow problem. Minimum cost network flow problems have the feature that when the problem data and constraints are integer valued, then the optimization model satisfies the integrality property [3]. This implies that the optimization problem can be solved without constraining the variables to be integer-valued, and given integer data, will provide an optimal integer-valued solution. Therefore minimum cost network flow problems can be solved with efficient algorithms that are much faster than solving a general (NP-hard) mixed integer problem. A minimum cost network flow problem is typically in the form:

$$\begin{aligned} &\text{minimize} \quad \sum((cost_{flow}) * (flow)) \quad \forall flow \\ &\text{subject to} \\ &flow\ out - flow\ in = \begin{cases} supply & \text{if source node} \\ -demand & \text{if sink node} \\ 0 & \text{if intermediate node.} \end{cases} \end{aligned} \tag{2.1}$$

Multi-commodity minimum cost network flow problems do not necessarily satisfy the integrality property that single-commodity minimum cost network flow problems satisfy due to shared resources [8]. Although initially the model in this thesis was cast as a multi-commodity minimum cost flow network problem, it has sufficient structure to be reformulated as a single commodity minimum cost flow network problem, and this ensures the integrality property.

Chapter 3

DETERMINISTIC OPTIMIZATION MODELS AND SMALL EXAMPLES

Two deterministic network models are formulated in this chapter. The models only consider messages being sent from gateways to remote users. Messages are discretized into unit-sized messages (referred to as unit-messages) to avoid being broken up between different gateways.

Section 3.1 discusses the model formulation for deterministic demand with no energy constraints (P1) and gives a small example. Section 3.2 discusses the minimum cost flow representation of (P1). Section 3.3 discusses the model formulation for deterministic demand with energy constraints (P2) and gives a small example. Section 3.4 discusses the minimum cost flow representation of (P2).

3.1 Deterministic Demand without Energy Constraints (P1)

A model with deterministic demand and no energy constraints is formulated in (P1). In this model, the messages to be delivered to the remote users are referred to as demands. This model follows the framework of a network flow problem. Messages flow from the CCC to gateways to nanosats to remote users. The connections between nanosats and ground nodes are known in advance using STK simulation results.

A few key assumptions are made:

1. There are no crosslinks between nanosats. This means that connectivity between ground nodes and remote users is achieved with a "store-and-forward" approach.

2. There is two-way communication (Full Duplex) between ground nodes and nanosats such that nanosats can simultaneously receive and send messages.

3. The time for "control requests" is negligible (e.g., the handshake protocol is negligible).

4. The energy consideration for uplinks is negligible to nanosats.

5. Nanosats have sufficiently large memory capacity so memory capacity will not be considered as a constraint.

6. All messages will be in queue at the CCC at time zero.

7. Remote users' movements are negligible relative to nanosats and are considered stationary.

(P1) has five sets: gateways, nanosats, remote users, messages, and time intervals, see Table 3.1. Since each message is unit-sized, the units used for the models will be in terms of "unit-messages". For example, the length of each discretized time interval is the amount of time it takes to deliver one unit-message, typically 10 or 100 seconds. There are five parameters used in model (P1): the start time of each contact time interval, the destination of each message, the contact time intervals between gateways and nanosats, the contact time intervals between nanosats and remote users, and the deterministic demand at each remote user. The parameter for the start time of each contact time interval is necessary because the contact windows between nanosats and gateways/remote users are discretized for the entire time horizon. Figure 3.1 shows an example of contact time windows for two ground nodes, two nanosats, and two remote users. Discretizing the time horizon allows us to disregard the large time windows where no contact is possible. Figure 3.2 shows the discrete time intervals associated with the same contact time windows in Figure 3.1. There are four decision variables that represent the flow of messages between gateways, nanosats, and remote users.

Table 3.1: Sets, parameters, and decision variables used in Model (P1).

Sets		
g	gateways	$g = 1, 2, ..., G$
n	nanosats	$n = 1, 2, ..., N$
r	remote users	$r = 1, 2, ..., R$
m	messages	$m = 1, 2, ..., M$
k	time intervals	$k = 1, 2, ..., K$
Parameters		
τ_k	Start time of contact time interval k	
$destination_{mr}$	1 if destination of message m is remote user r 0 otherwise	
$connection^n_{gk}$	1 if the connection between gateway g and nanosat n exists at time τ_k 0 otherwise	
$connection^r_{nk}$	1 if the connection between nanosat n and remote user r exists at time τ_k 0 otherwise	
d_r	(deterministic) demand to remote user r; can also be defined as the number of messages to remote user r	
Decision Variables		
v_{mgnk}	1 if message m is delivered from gateway g to nanosat n in time interval k 0 otherwise	
u_{mnrk}	1 if message m is delivered from nanosat n to remote user r in time interval k 0 otherwise	
w_{mg}	1 if message m is delivered to gateway g 0 otherwise	
z_{mnk}	flow of message m in nanosat n between time k and $(k+1)$	

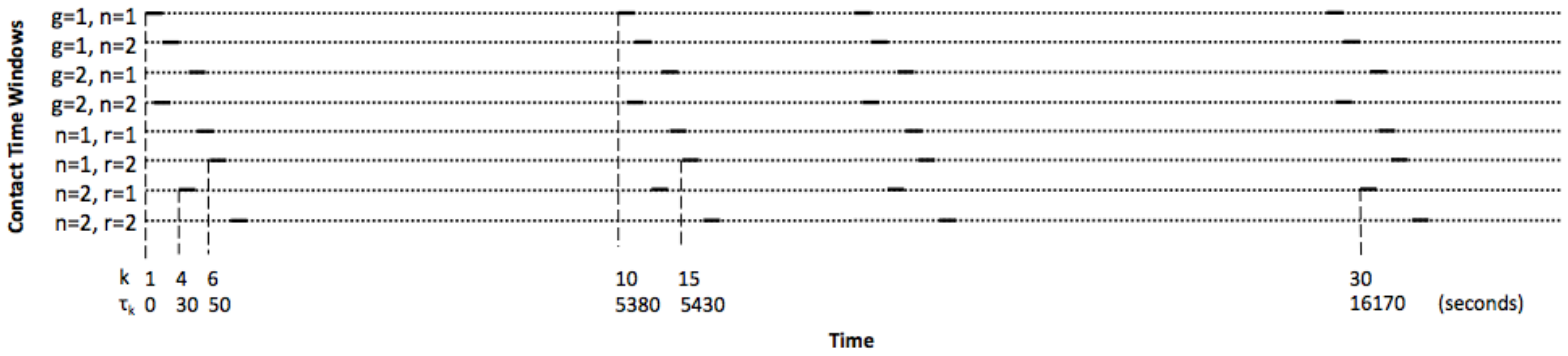

Figure 3.1: Contact time windows for a small example with two ground nodes, two nanosats, and two remote users, e.g., (g=1, n=1) represents (ground node 1, nanosat 1) and (n=1, r=1) represents (nanosat 1, remote user 1).

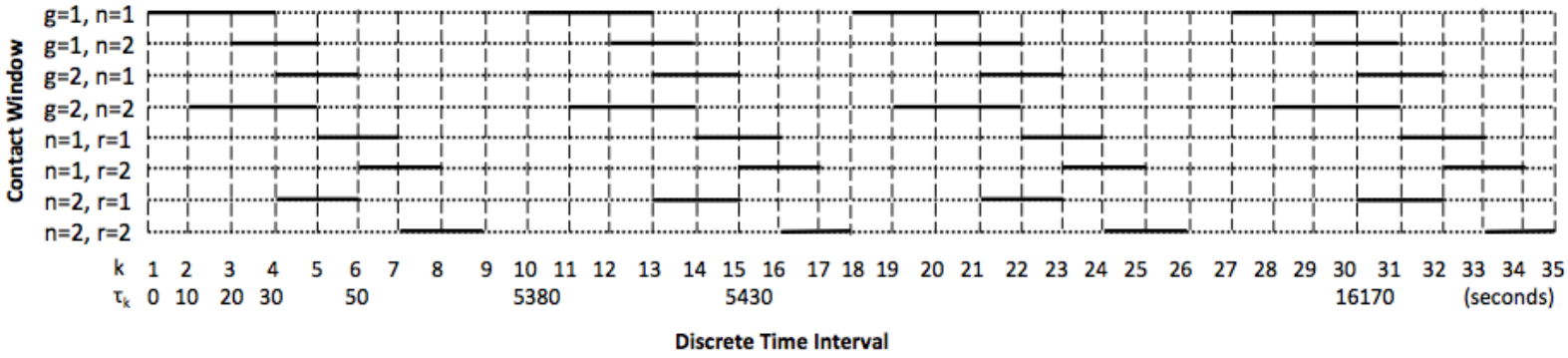

Figure 3.2: Discretized contact time intervals for small example (as in Figure 3.1) two ground nodes, two nanosats, and two remote users, e.g., (g=1, n=1) represents (ground node 1, nanosat 1) and (n=1, r=1) represents (nanosat 1, remote user 1).

Figure 3.3 shows an example of the network flow formulation using two ground nodes, two nanosats, and two users. In general, the maximum number of decision variables is M*G*N*K for v_{mgnk}, M*N*R*K for u_{mnrk}, M*G for w_{mg}, and M*N*K for z_{mnk}. However, not all combinations of v_{mgnk} and u_{mnrk} exist, but all combinations of w_{mg} and z_{mnk} exist. In the small example, there are 2 gateways, 2 nanosats, 2 remote users, 6 messages, and 36 time intervals. This leads to a total of 2,172 decision variables. Larger nanosat/ground node network will have a much larger number of decision variables.

The following equations are used to define the decision variables. The CPLEX model creates a decision variable for all possible combinations of v_{mgnk} and u_{mnrk}, and $destination_{mr}$ is a binary value equal to one if message m is to be sent to remote user r, $connection^{n}_{gk}$ and $connection^{r}_{nk}$ are also binary values equal one when contact between nanosats and gateways/remote users is possible at time τ_k,

$$u_{mnrk} \leq destination_{mr} \quad \forall(m,n,r,k) \tag{3.1}$$

$$v_{mgnk} \leq connection^{n}_{gk} \quad \forall(m,g,n,k) \tag{3.2}$$

$$u_{mnrk} \leq connection^{r}_{nk} \quad \forall(m,n,r,k). \tag{3.3}$$

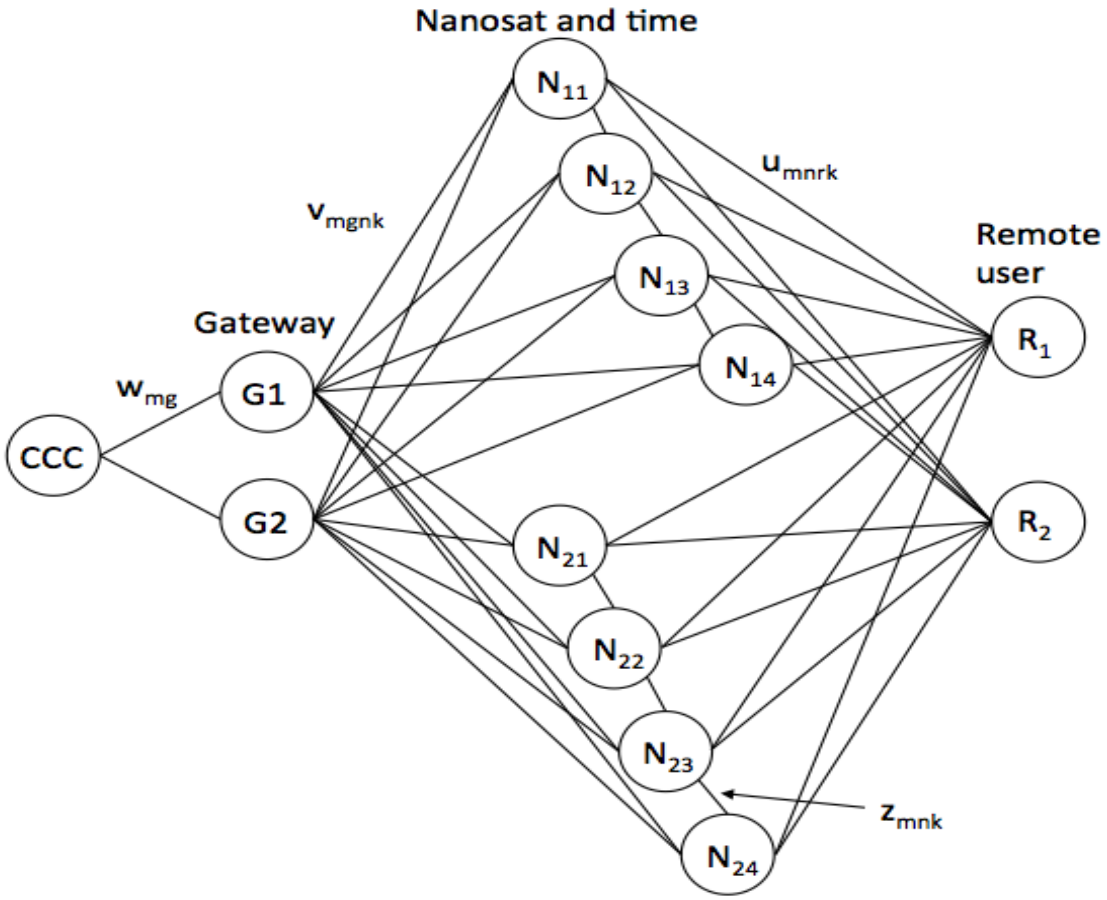

Figure 3.3: This is a network flow representation of the model. CCC is Central Command and Control, G_g denote ground nodes, N_{nk} denotes nanosat n at time interval k, R_{rk} denotes remote user r at time interval k. The decision variables w_{mg}, v_{mgnk}, u_{mnrk} represent the flow between CCC, G_g, N_{nk}, and R_{rk} respectively.

Model (P1)

objective function

$$\min \sum_m \sum_n \sum_r \sum_k (\tau_k * u_{mnrk}) \tag{3.4}$$

subject to

$$w_{mg} = \sum_n \sum_k v_{mgnk} \quad \forall(m,g) \tag{3.5}$$

$$\left(\sum_g v_{mgnk}\right) = z_{mnk} + \sum_r u_{mnrk} \quad \forall(m,n,k=1) \tag{3.6}$$

$$\left(\sum_g v_{mgnk}\right) + z_{mn(k-1)} = z_{mnk} + \sum_r u_{mnrk} \quad \forall(m,n,k=2,...,K-1) \tag{3.7}$$

$$\left(\sum_g v_{mgnk}\right) + z_{mn(k-1)} = \sum_r u_{mnrk} \quad \forall(m,n,k=K) \tag{3.8}$$

$$\sum_m \sum_n u_{mnrk} \leq 1 \quad \forall(r,k) \tag{3.9}$$

$$\sum_m \sum_r u_{mnrk} \leq 1 \quad \forall(n,k) \tag{3.10}$$

$$\sum_m \sum_g v_{mgnk} \leq 1 \quad \forall(n,k) \tag{3.11}$$

$$\sum_m \sum_n v_{mgnk} \leq 1 \quad \forall(g,k) \tag{3.12}$$

$$\sum_g w_{mg} = 1 \quad \forall m \tag{3.13}$$

$$\sum_n \sum_r \sum_k u_{mnrk} = 1 \quad \forall m \tag{3.14}$$

$$\sum_g \sum_n \sum_k v_{mgnk} = 1 \quad \forall m \tag{3.15}$$

$$\sum_m \sum_n \sum_k u_{mnrk} = d_r \quad \forall r \tag{3.16}$$

$$v_{mgnk}, u_{mnrk}, w_{mg}, z_{mnk} \in \{0, 1\} \tag{3.17}$$

The objective function in (3.4) minimizes the sum of the delivery times of messages. Constraint (3.5) is the flow constraint for gateways; constraints (3.6), (3.7), and (3.8) are the flow constraint for nanosats. Constraint (3.9) ensures that a remote user receives at most one message in each time period. Constraint (3.10) ensures that a nanosat sends at most one message in each time period. Constraint (3.11) ensures that a nanosat receives at most one message in each time period. Constraint (3.12) ensures that a gateway sends at most one message in each time period. Constraint (3.13) ensures that each message is delivered to gateways once. Constraint (3.14) ensures that each message is delivered to remote users once. Constraint (3.15) ensures that each message is delivered to nanosats once. Constraint (3.16) ensures that user demand, d_r, is met. Constraint (3.17) restricts the decision variables to be binary. Equations (3.1), (3.2), and (3.3) are added to ensure contact time windows are respected.

Small Example with (P1)

A small example is created with 2 gateways, 2 nanosats, 2 remote users, 6 messages, and 36 time units. CPLEX is used to solve this small example using (P1). The contact time windows (with values for τ_k) between nanosats and gateways and remote users are shown in Table 3.2. The contact times in Table 3.2 correspond to contact times in Figure 3.2; only the first 18 time intervals are shown in the table. The demand at each remote user, d_r, is shown in Table 3.3.

Model (P1) is solved for the small example. All six messages were delivered within eight time intervals with the last message being delivered 70 seconds after the start time. The results for (P1) are shown in Table 3.4 and Figure 3.4.

Table 3.2: Contact time windows between gateways and nanosats.

	Gateway 1		Gateway 2		Nanosat 1		Nanosat 2	
time (k, τ_k)	Nano 1	Nano 2	Nano 1	Nano 2	Remote User 1	Remote User 2	Remote User 1	Remote User 2
1, 0	Yes							
2, 10	Yes			Yes				
3, 20	Yes	Yes		Yes				
4, 30	Yes	Yes	Yes	Yes			Yes	
5, 40		Yes	Yes	Yes	Yes		Yes	
6, 50			Yes		Yes	Yes	Yes	
7, 60					Yes	Yes		Yes
8, 70								Yes
9, 80								Yes
10, 5380	Yes							
11, 5390	Yes			Yes				
12, 5400	Yes	Yes	Yes	Yes				
13, 5410	Yes	Yes	Yes	Yes			Yes	
14, 5420		Yes	Yes	Yes	Yes		Yes	
15, 5430			Yes		Yes	Yes	Yes	
16, 5440					Yes	Yes		Yes
17, 5450						Yes		Yes
18, 5460								Yes

Table 3.3: Message demand for each remote user.

Remote User (r)	1	2
Demand (d_r)	3	3

Table 3.4: Results for the small example showing the path that each message takes and the times when the message was delivered to a nanosat and user.

Message	Gateway	Nanosat	Time delivered to Nanosat (k)	Remote User	Time delivered to Remote User (k)
1	2	2	2	1	6
2	1	1	2	1	5
3	1	1	1	2	6
4	2	1	6	2	8
5	2	2	4	2	7
6	1	2	3	1	4

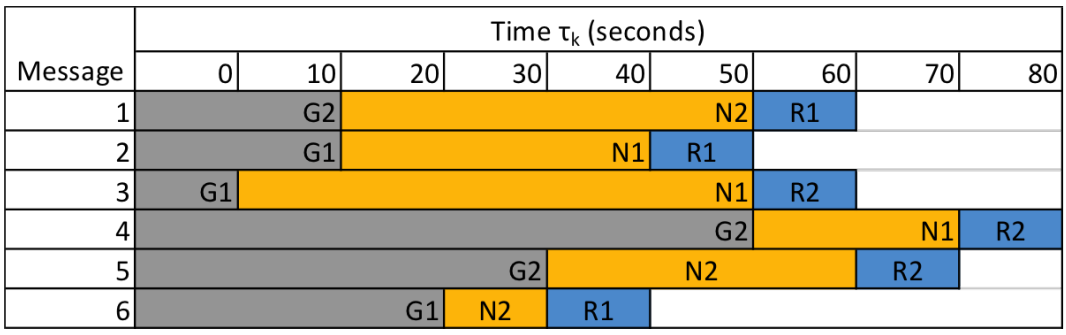

Figure 3.4: Visual representation of (P1) small example results with two gateways G1 and G2, two nanosats N1 and N2, and two remote users R1 and R2. Each line represents a message ($m = 1, 2, ..., 6$). For example, message 1 was delivered from gateway 2 to nanosat 2 at time 10 seconds. Message 1 was then delivered from nanosat 2 to remote user 1 at time 50 seconds.

3.2 Minimum Cost Flow Representation of (P1)

A minimum cost flow network problem, shown in (2.1), satisfies the integrality property if the right hand side of the constraints are integer valued [3]. Model (P1) can be shown to satisfy the integrality property by creating additional dummy nodes for messages and representing it in the form of (2.1). An equivalent model to (P1) is created with nodes CCC, G_{mg}, N_{mnk}, and U_{mr}. The flow decision variables are denoted $f_{(i),(j)}$ where i and j are in the node sets. The demand at each remote user r of message m is given to be one, i.e. d_{mr}. The capacity of flow from nodes i to j is given to be one for all flows. The supply at CCC is $\sum_m \sum_r d_{mr}$. The cost for flows not from nodes N_{mnk} to R_{mr} is zero; cost for flows from nodes N_{mnk} to R_{mr} is τ_k. The flow variables $f_{(CCC),(G_{mg})}$, $f_{(G_{mg}),(N_{mnk})}$, $f_{(N_{mnk}),(R_{mr})}$, and $f_{(N_{mnk}),(N_{mn(k-1)})}$ can be mapped to w_{mg}, v_{mgnk}, u_{mnrk}, and z_{mnk} from (P1). The equivalent problem written with the expanded network flow is given as:

objective function

$$\min \sum_m \sum_n \sum_r \sum_k \tau_k * f_{(N_{mnk}),(R_{mr})} \tag{3.18}$$

subject to

$$\sum_m \sum_g f_{(CCC),(G_{mg})} = \sum_m \sum_r d_{mr} \tag{3.19}$$

$$\sum_n \sum_k f_{(G_{mg}),(N_{mnk})} - f_{(CCC),(G_{mg})} = 0 \quad \forall G_{mg} \tag{3.20}$$

$$f_{(N_{nmk}),(N_{mn(k+1)})} + f_{(N_{mnk}),(R_{mr})} - \sum_g f_{(G_{mg}),(N_{mnk})} = 0 \quad \forall k = 1 \tag{3.21}$$

$$f_{(N_{nmk}),(N_{mn(k+1)})} + f_{(N_{mnk}),(R_{mr})} - f_{(N_{mn(k-1)}),(N_{mnk})} - \sum_g f_{(G_{mg}),(N_{mnk})} = 0 \quad \forall k = 2, ..., (K-1) \tag{3.22}$$

$$f_{(N_{mnk}),(R_{mr})} - f_{(N_{mn(k-1)}),(N_{mnk})} - \sum_g f_{(G_{mg}),(N_{mnk})} = 0 \quad \forall k = K \tag{3.23}$$

$$-\sum_{n}\sum_{k} f_{(N_{mnk}),(R_{mr})} = -d_{mr} \tag{3.24}$$

$$0 \leq f_{(i),(j)} \leq 1 \qquad \forall \text{nodes } (i)\&(j). \tag{3.25}$$

Constraint (3.19) ensures that flow into the CCC is equal the total number of messages at all remote users and corresponds with constraint (3.16). Constraint (3.20) ensures that the flow into each gateway is equal to the flow coming out of the gateway and corresponds to constraint (3.5) in (P1). Constraints (3.21), (3.22), and (3.23) ensure that the flow into each nanosat is equal to the flow coming out of the nanosat and corresponds to constraints (3.6), (3.7), and (3.8) of (P1). Constraint (3.24) ensures that the flow to each remote user is equal to the demand at that remote user and corresponds to constraints (3.13), (3.14), and (3.15). Constraint (3.25) ensures that the flow capacity on each arc and corresponds to constraints (3.9), (3.10), (3.11), (3.12), and (3.17). Figure 3.5 shows an example of the min-cost flow representation of (P1) using two gateways, two nanosats, two remote users, and two messages.

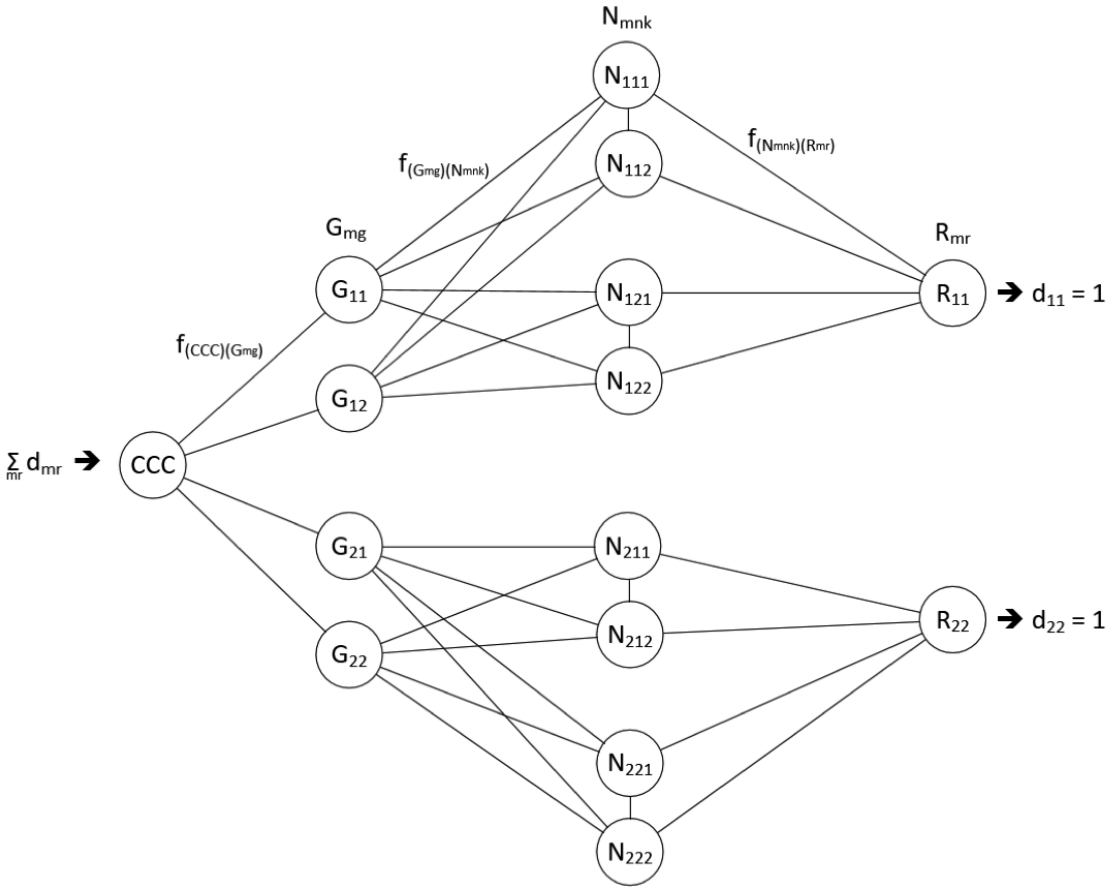

Figure 3.5: Minimum cost network flow representation of (P1)

3.3 Deterministic Demand with Energy Constraints (P2)

Model (P2) differs from (P1) in that energy constraints are considered. Four additional parameters are included, e_{min}, e_{max}, E_{n0}, and δ_{nk}, where e_{min} and e_{max} represent the minimum and maximum energy capacity for nanosats in terms of number of unit-messages, E_{n0} is the remaining energy for nanosats at τ_0 in terms of number of unit-messages, and δ_{nk} is the energy harvested during time interval k in terms of number of unit-messages. A nanosat charges enough energy to send one unit-message during each unit-message time interval. The amount of energy (in terms of unit-messages) charged during each time interval k is calculated based on the solar charging time windows obtained from the STK Scheduler. Figure

3.6 shows the ground node to nanosat contact windows that correspond to Figures 3.1 and 3.2 for the small example. It also shows two sets of solar charging time windows used in small examples (A) and (B). In (P2), h_{nk} is an intermediate (decision) variable to capture spilled energy or extra energy not needed for message delivery.

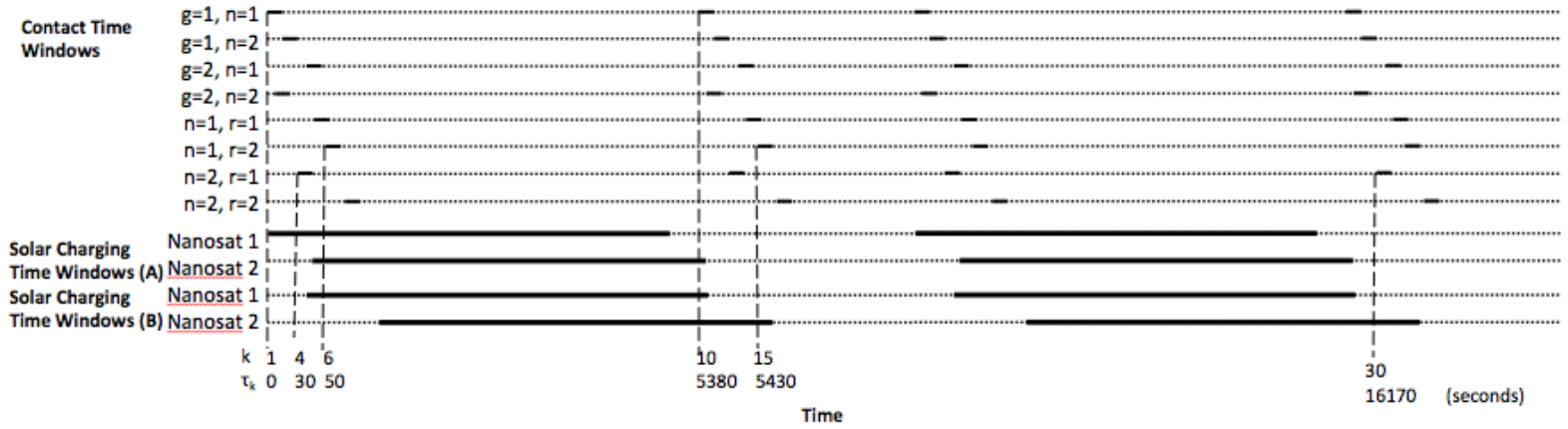

Figure 3.6: Contact time windows for small example as in Figure 3.1, with additional nanosat solar charging time windows, for small examples (A) and (B).

Model (P2)

objective function

$$\min \sum_{m}\sum_{n}\sum_{r}\sum_{k}(\tau_k * u_{mnrk}) \tag{3.26}$$

subject to

Constraints (3.5 - 3.17)

$$e_{nk} = E_{n0} - \sum_{m}\sum_{r} u_{mnrk} + \delta_{nk} - h_{n,k-1} \quad \forall(n, k = 1) \tag{3.27}$$

$$e_{nk} = e_{n(k-1)} - \sum_{m}\sum_{r} u_{mnrk} + \delta_{nk} - h_{n,k-1} \quad \forall(n, k = 2, ...K) \tag{3.28}$$

$$e_{min} \leq e_{nk} \leq e_{max} \quad \forall(r, k) \tag{3.29}$$

$$h_{nk} \geq 0 \tag{3.30}$$

The objective function in (3.26) minimizes the sum of the delivery times of messages. Constraints (3.5) - (3.17) are from (P1). Constraints (3.27) and (3.28) are the energy constraints which ensures that energy is stored or used according to nanosat specifications. Constraint (3.29) ensures that a nanosat energy level stays within the appropriate levels. Constraint (3.30) ensures that h_{nk} is collecting extra/spilled energy.

Small Example (A) with (P2)

The same parameters from the small example for (P1) are used. Table 3.5 shows the amount of energy that each nanosat gains during each time period k. Values for e_{min}, e_{max}, and E_{n0} are shown in Table 3.6. The results of small example (A) are shown in Table 3.7 and Figure 3.7. The six messages were all delivered within 70 seconds after the start time. For this example, the energy constraints do not make a difference on message delivery times, however the routes are slightly different. In particular, messages 2 and 4 had different routes.

Table 3.5: Energy charged during each time interval k (in terms of unit-messages) for small example (A) with P2.

k	1	2	3	4	5	6	7	8	9
$\delta_{n=1,k}$	1	1	1	1	1	1	1	1	352
$\delta_{n=2,k}$	1	1	1	1	1	1	1	1	358
k	10	11	12	13	14	15	16	17	18
$\delta_{n=1,k}$	1	1	1	1	1	1	1	1	1
$\delta_{n=2,k}$	1	1	0	0	0	0	0	0	0
k	19	20	21	22	23	24	25	26	27
$\delta_{n=1,k}$	1	1	1	1	1	1	1	1	352
$\delta_{n=2,k}$	1	1	1	1	1	1	1	1	358
k	28	29	30	31	32	33	34	35	36
$\delta_{n=1,k}$	1	1	1	1	1	1	1	1	1
$\delta_{n=2,k}$	1	1	0	0	0	0	0	0	0

Table 3.6: Values for e_{min}, e_{max}, and E_{n0}

e_{min}	1
e_{max}	10
E_{n0}	1

Table 3.7: Results for the small example (A) showing the path that each message takes and the times when the message was delivered to a nanosat and user.

Message	Gateway	Nanosat	Time delivered to Nanosat (k)	Remote User	Time delivered to Remote User (k)
1	2	2	2	1	6
2	1	2	3	1	4
3	1	1	1	2	6
4	2	1	6	2	8
5	2	2	4	2	7
6	1	1	2	1	5

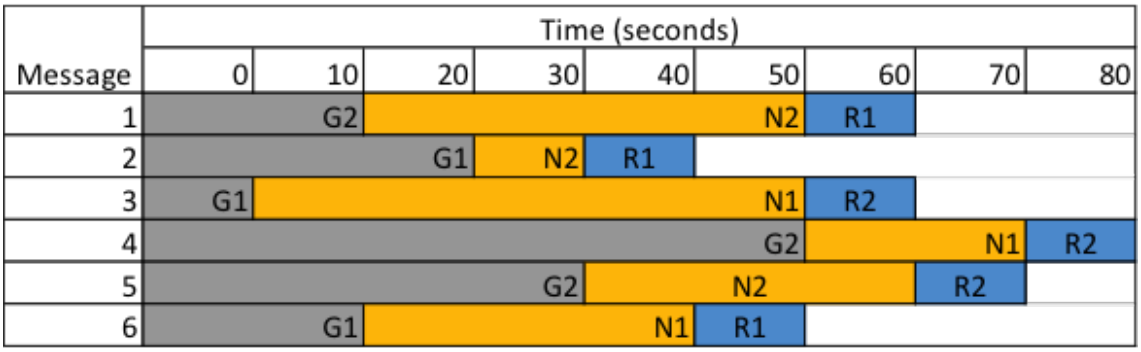

Figure 3.7: Visual representation of (P2) small example (A) results with two gateways G1 and G2, two nanosats N1 and N2, and two remote users R1 and R2.

Small Example (B) with (P2)

The same parameters from the small example (A) for (P1) are used in (B) except for δ_{nk}. New δ_{nk} values are shown in Table 3.8. The results are shown in Table 3.9 and Figure 3.8. Small examples (A) and (B) show that energy is not always a binding constraint. Solar charging time windows delay the delivery of messages in small example (B). Small example (A) shows that (P2) is able to deliver the messages within the same time from as (P1) but with slightly different routes. Small example (B) shows that message delivery times can be delayed if nanosats do not have sufficient energy. Both routes and delivery times for small example (B) differ from small example (A) for both (P1) and (P2).

Table 3.8: Energy charged during each time interval k (in terms of unit-messages) for small example (B) with P2.

k	1	2	3	4	5	6	7	8	9
$\delta_{n=1,k}$	0	0	0	0	1	1	1	1	352
$\delta_{n=2,k}$	0	0	0	0	0	0	0	0	162
k	10	11	12	13	14	15	16	17	18
$\delta_{n=1,k}$	1	0	0	0	0	0	0	0	0
$\delta_{n=2,k}$	1	1	1	1	1	1	1	0	0
k	19	20	21	22	23	24	25	26	27
$\delta_{n=1,k}$	0	0	0	0	1	1	1	1	352
$\delta_{n=2,k}$	0	0	0	0	0	0	0	0	162
k	28	29	30	31	32	33	34	35	36
$\delta_{n=1,k}$	1	0	0	0	0	0	0	0	0
$\delta_{n=2,k}$	1	1	1	1	1	1	1	0	0

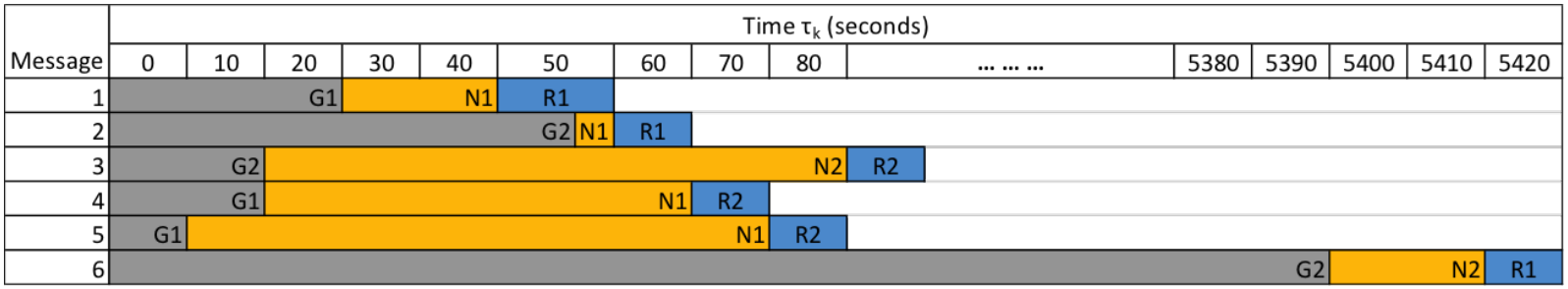

Figure 3.8: Visual representation of (P2) small example (B) results with two gateways G1 and G2, two nanosats N1 and N2, and two remote users R1 and R2.

Table 3.9: Results for the small example (B) showing the path that each message takes and the times when the message was delivered to a nanosat and user.

Message	Gateway	Nanosat	Time delivered to Nanosat (k)	Remote User	Time delivered to Remote User (k)
1	1	1	3	1	5
2	2	1	6	1	6
3	2	2	2	2	9
4	1	1	2	2	7
5	1	1	1	2	8
6	2	2	11	1	13

3.4 Min-Cost Flow Representation of (P2)

Model (P2) can be shown to satisfy the integrality property by creating dummy nodes and recasting it in the form of (2.1). (P2) has the same nodes and edges as (P1). However, energy and messages share the same edges. An equivalent model is created with nodes e_{nk}, G_{mgnk}, N_{mnk}, R_{mr}, and E_{extra}. The CCC node in (P2) is now represented by the multiple e_{nk} nodes. The demand at each remote user r of message m is limited to be one. The supply to each e_{nk} node is δ_{nk} and E_{n0} (if $k = 1$). The flow decision variables are denoted $f_{(i),(j)}$ where i and j are in the node sets. Unlike (P1), the capacity of flow from nodes i to j is not given to be one for all flows. The demand at each remote user r of message m is given to be one, d_{mr}. The supply at CCC is $\sum_m \sum_r d_{mr}$. The cost for flows not from nodes N_{mnk} to R_{mr} is zero; cost for flows from nodes N_{mnk} to R_{mr} is τ_k. The capacity of flow from nodes i to j is given to be one for all flows except for the flow of energy. The flow of energy between the e_{nk} nodes is bounded by e_{min} and e_{max}. The flow variables $f_{(e_{nk}),(E_{extra})}$, $f_{(e_{nk}),(G_{mgnk})}$, $f_{(e_{nk}),(e_{n(k+1)})}$, $f_{(G_{mgnk}),(G_{mgn(k+1)})}$, $f_{(G_{mgnk}),(N_{mnk})}$, $f_{(N_{mnk}),(N_{mn(k+1)})}$, and $f_{(N_{mnk}),(R_{mr})}$, can be mapped to e_{nk}, h_{nk}, w_{mg}, v_{mgnk}, u_{mnrk}, and z_{mnk} from (P2). Figure 3.9 shows an example

of the min-cost flow representation of (P2) using two gateways, two nanosats, two remote users, and two messages. The equivalent problem written with the expanded network flow is given as:

objective function

$$\min \sum_{m}\sum_{n}\sum_{r}\sum_{k} \tau_k * f_{(N_{mnk}),(R_{mr})} \tag{3.31}$$

subject to

$$\delta_{nk} + E_{n0} = f_{(e_{nk}),(e_{n(k+1)})} + \sum_{m}\sum_{g} f_{(e_{nk}),(G_{mgnk})} + f_{(e_{nk}),(E_{extra})} \qquad \forall(n, k = 1) \tag{3.32}$$

$$\delta_{nk} + f_{(e_{n(k-1)}),(e_{nk})} = f_{(e_{nk}),(e_{n(k+1)})} + \sum_{m}\sum_{g} f_{(e_{nk}),(G_{mgnk})} + f_{(e_{nk}),(E_{extra})} \qquad \forall(n, k = 2, ..., (K-1)) \tag{3.33}$$

$$\delta_{nk} + f_{(e_{n(k-1)}),(e_{nk})} = \sum_{m}\sum_{g} f_{(e_{nk}),(G_{mgnk})} + f_{(e_{nk}),(E_{extra})} \qquad \forall(n, k = K) \tag{3.34}$$

$$f_{(e_{nk}),(G_{mgnk})} = f_{(G_{mgnk}),(G_{mgn(k+1)})} + f_{(G_{mgnk}),(N_{mnk})} \qquad \forall(m, g, n, k = 1) \tag{3.35}$$

$$f_{(e_{nk}),(G_{mgnk})} + f_{(G_{mgn(k-1)}),(G_{mgnk})} = f_{(G_{mgnk}),(G_{mgn(k+1)})} + f_{(G_{mgnk}),(N_{mnk})} \qquad \forall(m, g, n, k = 2, ..., (K-1)) \tag{3.36}$$

$$f_{(e_{nk}),(G_{mgnk})} + f_{(G_{mgn(k-1)}),(G_{mgnk})} = f_{(G_{mgnk}),(N_{mnk})} \qquad \forall(m, g, n, k = K) \tag{3.37}$$

$$\sum_{g} f_{(G_{mgnk}),(N_{mnk})} = f_{(N_{mnk})(N_{mn(k+1)})} + f_{(N_{mnk})(R_{mr})} \qquad \forall(m, n, r, k = 1) \tag{3.38}$$

$$\sum_{g} f_{(G_{mgnk}),(N_{mnk})} + f_{(N_{mn(k-1)})(N_{mnk})} = f_{(N_{mnk})(N_{mn(k+1)})} + f_{(N_{mnk})(R_{mr})} \qquad \forall(m, n, r, k = 2, ..., (K-1)) \tag{3.39}$$

$$\sum_{g} f_{(G_{mgnk}),(N_{mnk})} + f_{(N_{mn(k-1)})(N_{mnk})} = f_{(N_{mnk})(R_{mr})} \qquad \forall (m, n, r, k = K) \tag{3.40}$$

$$\sum_{n} f_{(N_{mnk})(R_{mr})} = d_{mr} \qquad \forall r \tag{3.41}$$

$$\sum_{n} \sum_{k} f_{(e_{nk}),(E_{extra})} = \sum_{n} \sum_{k} (\delta_{nk} + E_{n0}) - \sum_{m} \sum_{r} d_{mr}. \tag{3.42}$$

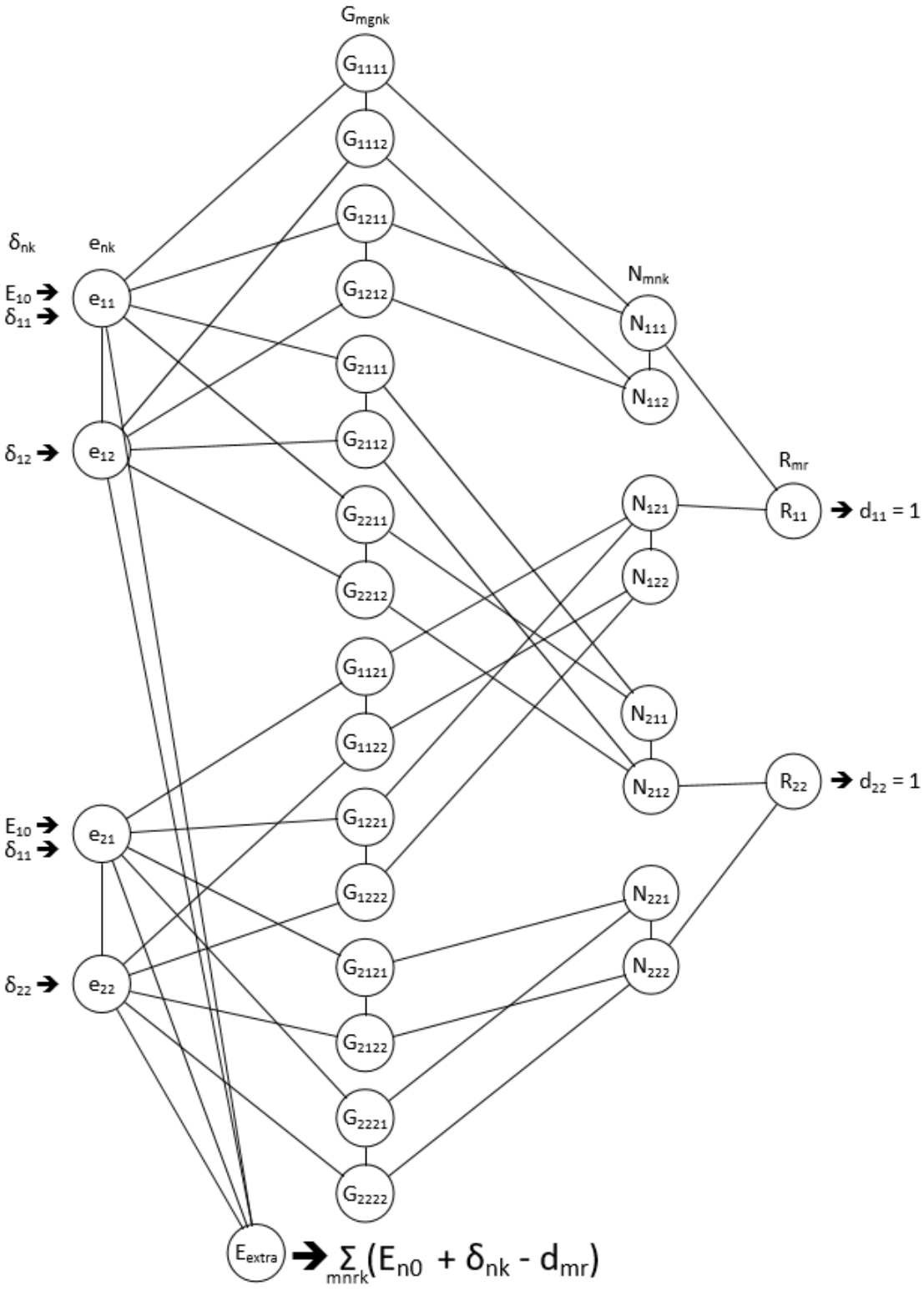

Figure 3.9: Minimum cost network flow representation of (P2).

Chapter 4

PROBABILISTIC OPTIMIZATION MODELS AND SMALL EXAMPLES

Two probabilistic network models are formulated in this chapter. Similar to Models (P1) and (P2), these models consider unit-messages being sent from gateways to remote users. Section 4.1 discusses the model formulation for probabilistic demand with no energy constraints (P3) and gives a small example. Section 4.1 also shows how the chance constraints are linearized. Section 4.2 discusses the minimum cost flow representation of (P3) that satisfies the integrality property. Section 4.3 discusses the model formulation for probabilistic demand with energy constraints and gives a small example. Section 4.4 discusses possible reasons why (P4) does not satisfy the integrality property.

4.1 Probabilistic Demand without Energy Constraints (P3)

The connections between nanosats and remote users are known, however the connection may not be strong enough for the nanosat to successfully deliver a message. There is uncertainty in knowing if a message was successfully delivered or if it needs to be sent again due to an unstable connection. This can be modeled by changing the deterministic demand d_r that was used in (P1) and (P2) to a random variable D_r.

A chance constraint similar to that used in Li & Zabinsky's supplier selection problem [5] is used to model the uncertainty of the quality of connection between the nanosat and remote user. The demand constraint (3.16) from (P1) is changed into a chance constraint (4.1) where user demand is a random variable, D_r,

$$P\left(\sum_m \sum_n \sum_k u_{mnrk} \geq D_r\right) \geq 1-\alpha \quad \forall r. \tag{4.1}$$

Linearizing Chance Constraint

Constraint (4.1) can be rewritten as $\sum_m \sum_n \sum_k u_{mnrk} \geq F_{D_r}^{-1}(1-\alpha)\ \forall r$, where F_{D_r} denotes the cumulative distribution function (cdf) of D_r and $0 \leq \alpha \leq 1$. Different distributions can be used to define $F_{D_r}^{-1}(1-\alpha)$. In this thesis, each message can be successfully delivered in three or less message delivery attempts. To capture the uncertainty of a successful transmission of a message, we let the probability that one attempt is needed to deliver a message be p_1, the probability that two attempts are needed to deliver a message is p_2, and the probability that three attempts are needed to deliver a message is p_3. The probabilities are given in (4.2).

$$p_i = \begin{cases} 0.7 & i = 1 \\ 0.2 & i = 2 \\ 0.1 & i = 3 \end{cases} \tag{4.2}$$

Note that the probability of a message being delivered in three or less attempts equals one. The D_r cdf for each remote user is calculated using a multinomial distribution. The multinomial distribution is used to calculate the probability of having n message delivery attempts. The number of message delivery attempts n ranges from d_r to $3d_r$ for each remote user r. A cumulative distribution function (cdf) is calculated for each remote user. It is assumed that α is 0.1, so the probability of meeting demand D_r must be greater than or equal to 0.9. The cdf for each remote user is used to determine the number of message delivery attempts needed to ensure that the probability of meeting demand D_r is be greater than or equal to 0.9. The smallest integer value that satisfies this probability is $F_{D_r}^{-1}(1-\alpha)$. Constraint (4.1) is equivalent to $\sum_m \sum_n \sum_k u_{mnrk} \geq F_{D_r}^{-1}(1-\alpha)\ \forall r$ and is now represented by r linear constraints.

Model (P3)

objective function

$$\min \sum_{m}\sum_{n}\sum_{r}\sum_{k}(\tau_k * u_{mnrk}) \tag{4.3}$$

subject to

Constraints (3.5) - (3.12), (3.17)

$$\sum_{g} w_{mg} \geq 1 \quad \forall m \tag{4.4}$$

$$\sum_{n}\sum_{r}\sum_{k} u_{mnrk} \geq 1 \quad \forall m \tag{4.5}$$

$$\sum_{g}\sum_{n}\sum_{k} v_{mgnk} \geq 1 \quad \forall m \tag{4.6}$$

$$\sum_{m}\sum_{n}\sum_{k} u_{mnrk} \geq F_{D_r}^{-1}(1-\alpha) \quad \forall r \tag{4.7}$$

The objective function in (4.3) minimizes the sum of the delivery times of messages. Constraint (4.4) ensures that each message is delivered to gateways at least once. Constraint (4.5) ensures that each message is delivered to remote users at least once. Constraint (4.6) ensures that each message is delivered to nanosats at least once. Constraints (4.4 - 4.6) differ from constraints (3.13) - (3.15) in (P1) and (P2) because each message can be sent more than once in (P3). Constraint (4.7) is the equivalent representation of the chance constraint (4.1) which ensures that the probability that remote user demand, D_r, is met will be greater than or equal to some large value, (1-α).

Small Example for (P3)

The same parameters from the small example for (P1) are used. Table 4.1 shows the values

used for constraint (4.7). Figure 4.1 shows the cdf used for both remote users 1 and 2. Three messages are destined to each remote users 1 and 2. These values are used in place of the demand constraint (3.16) for (P1). The results of the small example are shown in Table 4.2 and Figure 4.2. While there can be multiple delivery attempts for each message, only the message path that first delivers the message is used as the solution. The uncertainty of a successful message transmission resulted in both different message paths and longer delivery times.

Table 4.1: Integer value for $F_{D_r}^{-1}$ that ensures the demand D_r is met at least 90% of the time.

Remote User (r)	1	2
$F_{D_r}^{-1}$	6	6

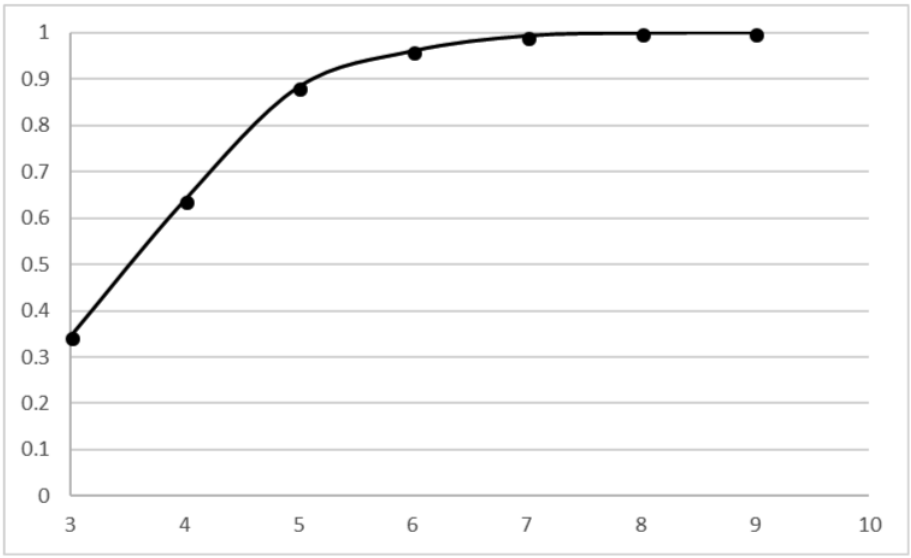

Figure 4.1: D_r cumulative distribution function for three messages.

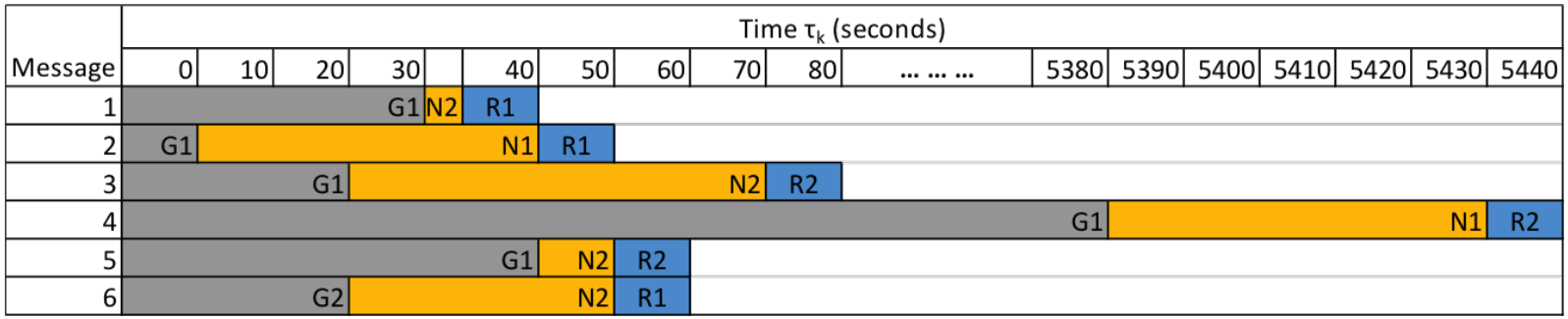

Figure 4.2: Visual representation of (P3) small example results with two gateways G1 and G2, two nanosats N1 and N2, and two remote users R1 and R2.

Table 4.2: Results for the small example showing the path that each message takes and the times when the message was delivered to a nanosat and remote user.

Message	Gateway	Nanosat	Time delivered to Nanosat (k)	Remote User	Time delivered to Remote User (k)
1	1	2	4	1	4
2	1	1	2	1	5
3	1	2	3	2	8
4	1	1	10	2	15
5	1	2	5	2	6
6	2	2	2	1	6

4.2 Min-Cost Flow Representation of (P3)

Model (P3) can be shown to satisfy the integrality property by using the min-cost flow representation of (P1), which is discussed in Section 3.2. Model (P1) is shown in standard form. Adding surplus variables to constraints (4.4) - (4.7) creates a standard form model that is equivalent to (P1). Since (P1) can be mapped to its respective min-cost flow representation, (P3) can also be mapped to the same min-cost flow representation.

4.3 Probabilistic Demand with Energy Constraints (P4)

This model includes both probabilistic demand and energy constraints. It uses parameters and variables that were used for (P2) and (P3).

Model (P4)

objective function

$$\min \sum_{m} \sum_{n} \sum_{r} \sum_{k} (\tau_k * u_{mnrk}) \tag{4.8}$$

subject to

Constraints (3.5) - (3.12), (3.17)

Constraints (3.27) - (3.30)

Constraints (4.4) - (4.7)

The objective function in (4.8) minimizes the sum of the delivery times of messages. Constraints (3.5) - (3.12), (3.17) are from (P1). Constraints (3.27) - (3.30) are the energy constraints from (P2). Constraints (4.4) - (4.7) are from (P3).

Small Example with (P4)

The same parameters from the small example for (P3) are used. The results of the small example are shown in Table 4.3 and Figure 4.3. While there can be multiple delivery attempts for each message, only the the message path that first delivers the message is used as the solution. (P3) and (P4) have similar message delivery times but the message delivery paths differ. In (P3), all but one message was delivered from Gateway 1. The message paths chosen by (P4) are more well balanced.

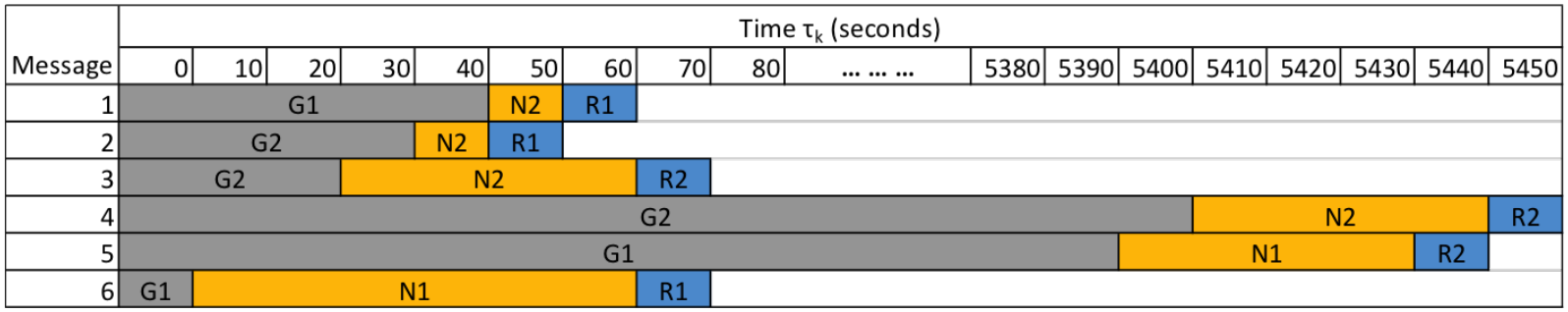

Figure 4.3: Visual representation of (P4) small example results with two gateways G1 and G2, two nanosats N1 and N2, and two remote users R1 and R2.

Table 4.3: Results for the small example showing the path that each message takes and the times when the message was delivered to a nanosat and remote user.

Message	Gateway	Nanosat	Time delivered to Nanosat (k)	Remote User	Time delivered to Remote User (k)
1	1	2	5	1	6
2	1	2	4	1	5
3	2	2	3	2	7
4	2	2	12	2	16
5	1	1	11	2	15
6	1	1	1	1	7

4.4 Min-Cost Flow Representation of (P4)

(P4) does not satisfy integrality because combining energy and multiple message delivery attempts eliminates the special problem format that allowed (P1), (P2), and (P3) to be modeled as single commodity network flow problems. For (P4), the flow arcs now share capacity between each message. For example, since each unique message m may have multiple message delivery attempts, those arcs now have a shared capacity between the same message.

Chapter 5

REALISTIC PROBLEM DESCRIPTION

Models (P1), (P2), (P3), and (P4) are run using data from a realistic problem. Simulation data from STK is used to determine contact time windows and solar charging windows. A 12-hour time window is simulated using 2 gateways, 5 nanosats, and 6 remote users. The time window is from 72,000 seconds to 115,200 seconds. The 12-hour time window is discretized into 335 time units ($k = 335$). A total of 21 messages were used. Three messages were destined for remote user 1, seven messages for remote user 2, three messages for remote user 3, five messages for remote user 4, one message for remote user 5, and two messages for remote user 6. Thirty-nine message delivery attempts were used for (P3) and (P4).

5.1 Results

Tables 5.1 - 5.4 show the optimization solution for realistic problems (P1), (P2), (P3), and (P4). Message paths and delivery times are shown in the tables. For (P1), the 21 messages were delivered within 218 time units. For (P2), the 21 messages were delivered within 260 time units. For (P3), the 21 messages were successfully delivered within 270 time intervals. For (P4), the 21 messages were successfully delivered within 280 time intervals.

Table 5.1: Message paths and delivery times for (P1).

Message	Gateway	Nanosat	Time Delivered to Nanosat (k)	Remote User	Time Delivered to Remote User (k)
1	1	4	3	2	178
2	1	4	89	5	218
3	1	3	92	2	92
4	2	3	88	4	88
5	2	2	64	4	67
6	1	4	2	3	215
7	1	4	41	1	179
8	2	3	91	2	91
9	1	4	43	1	181
10	2	3	89	2	89
11	1	4	42	2	180
12	1	4	1	4	177
13	2	2	66	4	66
14	2	2	63	3	64
15	2	3	90	2	90
16	1	4	39	4	216
17	2	3	87	6	87
18	1	4	7	6	217
19	1	4	8	1	182
20	2	2	65	2	65
21	1	1	4	3	4

Table 5.2: Message paths and delivery times for (P2).

Message	Gateway	Nanosat	Time Delivered to Nanosat (k)	Remote User	Time Delivered to Remote User (k)
1	1	4	178	2	3
2	2	2	67	5	66
3	2	3	90	2	90
4	2	3	91	4	91
5	2	4	260	4	260
6	2	3	87	3	87
7	1	4	218	1	88
8	1	1	4	2	4
9	2	3	88	1	88
10	1	4	216	2	6
11	1	4	215	2	7
12	1	4	180	4	8
13	1	4	177	4	41
14	2	2	64	3	64
15	2	4	259	2	259
16	1	4	217	4	2
17	1	4	179	6	1
18	2	3	89	6	89
19	2	2	66	1	61
20	1	3	92	2	92
21	2	2	65	3	62

Table 5.3: Message paths and delivery times for (P3).

Message	Gateway	Nanosat	Time Delivered to Nanosat (k)	Remote User	Time Delivered to Remote User (k)
1	1	4	42	2	218
2	2	5	19	5	265
3	1	4	2	2	259
4	1	4	40	4	92
5	1	4	224	4	260
6	1	4	7	3	89
7	2	2	61	1	267
8	1	4	39	2	90
9	2	5	21	1	270
10	1	4	8	2	64
11	1	5	149	2	269
12	1	4	3	4	221
13	2	1	41	4	217
14	1	4	1	3	182
15	1	4	43	2	216
16	2	3	90	4	91
17	1	4	41	6	67
18	2	1	40	6	219
19	2	2	63	1	215
20	1	4	87	2	87
21	1	1	4	3	4

Table 5.4: Message paths and delivery times for (P4).

Message	Gateway	Nanosat	Time Delivered to Nanosat (k)	Remote User	Time Delivered to Remote User (k)
1	1	4	43	2	280
2	2	3	89	5	259
3	2	1	41	2	283
4	1	4	6	4	181
5	1	4	40	4	89
6	2	1	34	3	216
7	1	4	89	1	219
8	2	1	83	2	270
9	1	4	41	1	66
10	1	4	1	2	217
11	1	2	33	2	65
12	1	4	2	4	179
13	1	4	39	4	180
14	1	5	10	3	265
15	2	1	40	2	215
16	1	4	3	4	67
17	1	1	4	6	4
18	1	4	90	6	92
19	1	4	8	1	91
20	2	3	86	2	88
21	1	2	32	3	64

5.2 Discussion

Thirty-nine message delivery attempts were made for (P3) and (P4). The number of total delivery attempts was calculated using the method detailed in Chapter 4. Model (P3) took longer than (P1) and (P2) to successfully deliver the 21 messages. Model (P4) took the longest to successfully deliver the 21 messages. Not all the message delivery times for (P3) and (P4) are implementable in reality. For example, in (P3) the first attempt to deliver message 8 has a delivery time of $k = 92$ and $k = 90$ to the nanosat and remote user respectively. This is a feasible solution given the network flow formulation of (P3), however this is not achievable in the real world. For each message at least one delivery attempt will be achievable because for the flow conservation constraints to be fulfilled, there must be at least one flow v_{mgnk} that is less than or equal to a flow u_{mnrk} for all m and n. Tables 5.3 and 5.4 show one message delivery time and path. If multiple delivery attempts were made, the path with the quickest feasible delivery time to the remote user is chosen.

For (P1) and (P2) only four of the five nanosats were used to deliver all the messages; all five nanosats were utilized for (P3) and (P4). This may to due to the fifth nanosat's orbital parameters or to the fact that (P3) and (P4) required 39 message delivery attempts instead of the 21 attempts for (P1) and (P2). Figures 5.1 - 5.4 show a graphical representation of the message paths for (P1), (P2), (P3), and (P4). The message paths taken in (P3) and (P4) in Figures 5.3 and 5.4 are more well balanced than the paths taken in (P1) and (P2) in Figures 5.1 and 5.2.

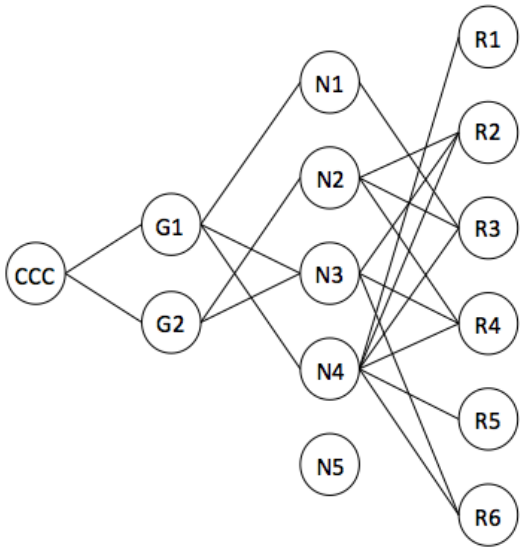

Figure 5.1: Network representation of (P1) message paths. CCC is the Central Command and Control Center, G1 representations gateway 1, N1 is nanosat 1, and R1 is remote user 1.

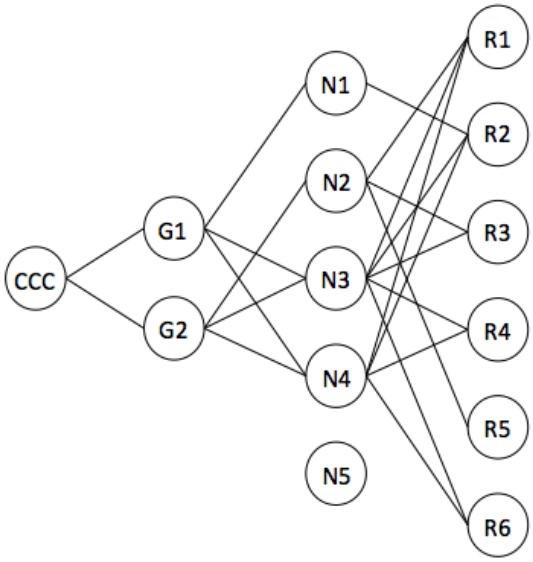

Figure 5.2: Network representation of (P2) message paths. CCC is the Central Command and Control Center, G1 representations gateway 1, N1 is nanosat 1, and R1 is remote user 1.

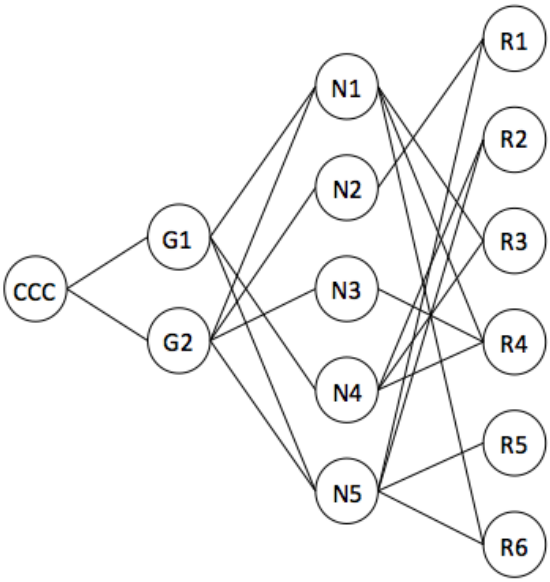

Figure 5.3: Network representation of (P3) message paths. CCC is the Central Command and Control Center, G1 representations gateway 1, N1 is nanosat 1, and R1 is remote user 1.

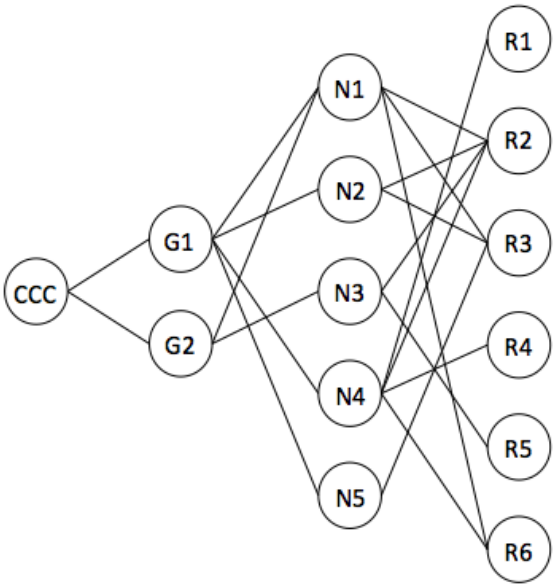

Figure 5.4: Network representation of (P4) message paths. CCC is the Central Command and Control Center, G1 representations gateway 1, N1 is nanosat 1, and R1 is remote user 1.

Chapter 6

COMPARISON OF GREEDY POLICIES AND OPTIMIZATION POLICIES

A simulation is created to compare different scheduling policies. Section 6.1 discusses the simulation framework and defines an easily implemented greedy policy for routing messages. In Section 6.2, the greedy policy is compared to the solutions of (P1) and (P2) for the small example (B) by evaluating the objective functions.. In Section 6.3, the greedy policy and solutions from (P1)-(P4) are simulated and the results are presented.

6.1 Simulation Framework

The simulation has six main components: the STK Simulation Database, Nanosat, Central Command and Control (C3), Ground Gateways, Remote Users, Main Simulation, and Performance Evaluation. A block diagram for the simulation framework is shown in Figure 6.1.

The STK simulation database is generated by the STK software. A scenario with a constellation of nanosats and various ground node locations for gateways and remote users is set up using STK. Coverage analysis is performed to generate contact time windows between nanosats and ground nodes (gateways and remote users). Solar charging time windows are generated for nanosats through sunlight analysis.

A nanosat object stores the database, which includes sunlight time windows and contact time windows between ground nodes and nanosats. Each nanosat object includes methods for updating the message queue, remaining energy, and message arrival and departure times as it receives and transmits messages. In the main simulation, several nanosat objects are

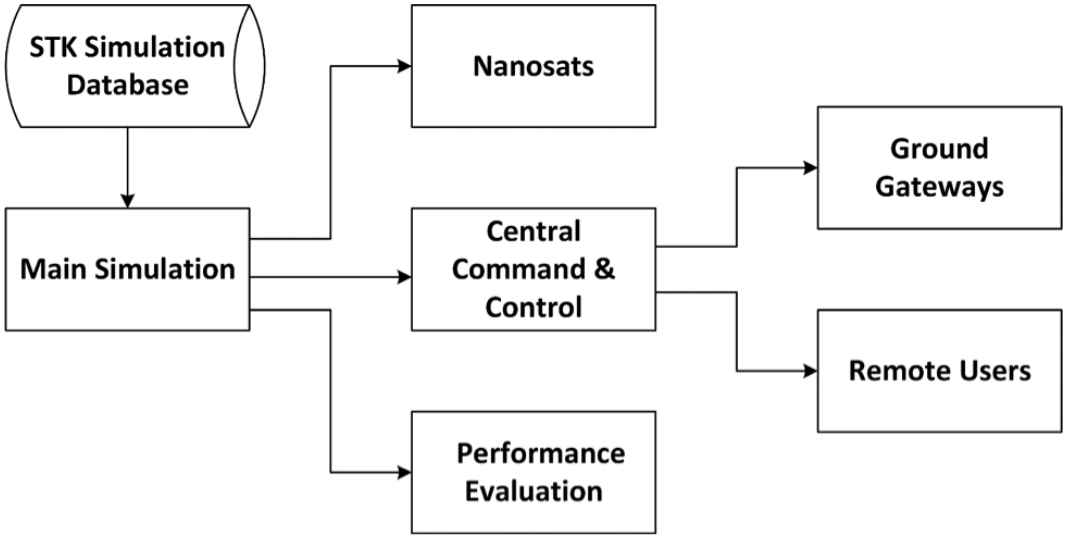

Figure 6.1: Block diagram of main scheduling simulation architecture.

constructed to form a constellation. Gateways and remote users include the identification of ground nodes which represent gateways and remote users respectively.

The C3 object stores the database for ground nodes and nanosats including contact time windows. The C3 object includes methods for message generation, route generation using a greedy policy or solving the optimization models (P1), (P2), (P3), and (P4), and updating message queues and arrival times at gateways and remote users. Messages are generated at C3 using a Poisson process with a specified mean message generation rate. The destination for each message is randomly assigned to a remote user from a set of remote users in the database.

The main simulation module executes a discrete event simulation. It creates a time-ordered event list is created based on the beginnings of contact time windows between ground nodes and nanosats. For each contact event, uploading (reception at nanosat from gateway) and/or downloading (transmission from nanosat to remote user) of messages are performed based

on the contact nodes (nanosat and gateway or remote user) and the message queue on the nanosat and/or gateway. With each download and upload event, message queues and arrival times of messages are updated. With each download event, remaining energy on the nanosat is updated.

Route generation (selection of gateway and nanosat for a message) by a greedy policy is based on the earliest delivery time to the remote user considering contact time windows only. Route generation by an optimal policy is based on solving a specified optimization model (P1), (P2), (P3), (P4). The optimization routes used are shown in Tables 5.1 - 5.4.

Two types of message delivery simulations are performed: deterministic and probabilistic. In the deterministic simulation, the download for each message happens only once (100% delivery success is assumed). In the probabilistic simulation, the number of downloads for each message is sampled from a given discrete probability mass function (e.g., Pr(number of downloads = 1) = 0.7, Pr(number of downloads = 2) = 0.2, Pr(number of downloads = 3) = 0.1). Based on the sampled value, a message may be transmitted from the nanosat one, two or three times. The message delivery time and the remaining energy on the nanosat are properly updated based on the number of transmissions.

The performance evaluation calculates the delay for each message which is calculated as the difference between the arrival time at the user and the message generation at the C3. This is done once all the messages have been delivered to remote users. The statistics on the delay performance of all messages for a specified scheduling and routing policy (greedy, (P1), (P2), (P3) or (P4)) are evaluated including mean, minimum, maximum, and standard deviation.

6.2 Greedy Policy with Small Example

The greedy policy was first evaluated under the model (P1) and obtained the same objective function value as the optimal solution presented in Section 3.6. The objective function value was the same as the original (P1) optimization objective function value, because there are many message paths that lead to the same objective function value so both (P1) and the greedy policy are optimal solutions to (P1). Then the greedy policy was evaluated under model (P2) with energy constraints, and the objective function value of 16,440 was higher than the (P2) optimization objective function value of 5,710. The greedy policy message paths can be seen in Figure 6.2. The visual results for (P2) optimization and greedy policy (P2) are shown in Figure 6.3

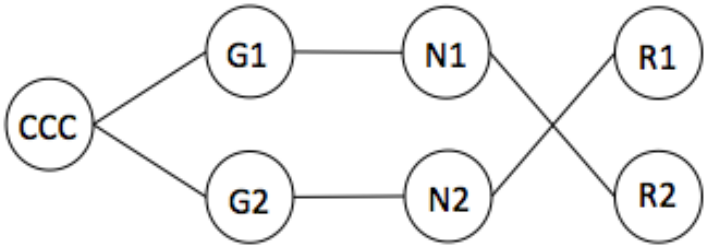

Figure 6.2: Network representation of greedy message path for small example. CCC is the Central Command and Control Center, G1 representations gateway 1, N1 is nanosat 1, and R1 is remote user 1.

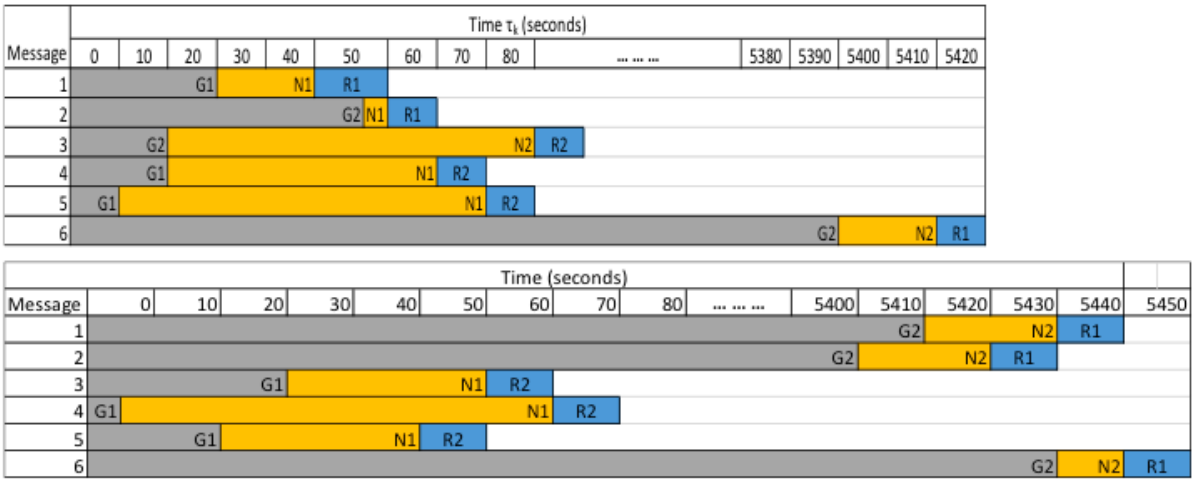

Figure 6.3: Visual representation of Greedy (P2) small example result with two gateways G1 and G2, two nanosats N1 and N2, and two remote users R1 and R2.

6.3 Simulation Results on Realistic Problem

The simulation ran using the parameters of the realistic problem that were presented in Chapter 5. However, a 24-hour time window was used for the simulation (instead of the 12-hour time window used in Chapter 5). The deterministic simulation where each message was successfully delivered on the first attempt was run once using the greedy policy and the optimal routes from (P1) and (P2). The results are shown in Table 6.1. The probabilistic simulation was run ten times using the greedy policy and the solutions from (P1), (P2), (P3), and (P4). The results are shown in Table 6.2.

The greedy policy performed better than simulated (P1) and simulated (P2) for the deterministic simulation in terms of total delivery time. This may be explained by the contact time not being discretized in the simulation. the 12-hour/24-hour time horizon difference between the realistic example and the simulation, or other small differences between the optimization problems and the simulaton. Simulated (P1) performed better in terms of the maximum delivery time. This means that all the messages were delivered in a shorter time

Table 6.1: Deterministic simulation results showing Greedy, simulated (P1), and simulated (P2) policies.

	Greedy	Simulated (P1)	Simulated (P2)
Total Delivery Time (seconds)	1,848,100	1,990,400	1,873,300
Minimum Delivery Time (seconds)	72,800	72,800	72,800
Maximum Delivery Time (seconds)	118,900	110,700	129,800

Table 6.2: Probabilistic simulation results showing Greedy, simulated (P1), simulated (P2), simulated (P3), and simulated (P4) policies. The average of 10 trials is shown.

	Greedy	Simulated (P1)	Simulated (P2)	Simulated (P3)	Simulated (P4)
Average Total Delivery Time (seconds)	1,999,760	2,078,320	1,998,210	2,056,150	1,925,410
Average Minimum Delivery Time (seconds)	72,830	75,690	72,850	72,830	73,510
Average Maximum Delivery Time (seconds)	129,520	113,410	135,660	138,620	126,270

for simulated (P1) than when using the greedy policy. There is abundant energy during the deterministic simulation due to the smaller number of message delivery attempts. Energy does not constrain the network when there is a small number of message deliveries.

On average, simulated (P4) performed the best for the probabilistic simulation in terms of the average total delivery time. The greedy policy performed better than simulated (P1) and simulated (P3). Neither (P1) nor (P3) take energy constraints into consideration, which may be why neither simulated policy performed better than the greedy policy. Simulated (P2) performed slightly better than the greedy policy in terms of average total delivery time, however it is not a significant improvement and on average, it took longer for all the messages to be delivered for simulated (P2) than for the greedy policy. A larger number of message delivery attempts can be accommodated using more well-balanced message paths. (P4) has a more well-balanced use of paths than the greedy policy. Refer to Figures 5.1 - 5.4 for the message paths taken for (P1), (P2), (P3), and (P4). The greedy policy only utilizes one of the two gateways and does not utilize nanosat 3. (P4) uses both gateways and all five nanosats (including nanosat 3). The greedy and (P4) message paths can be seen in Figure 6.4. Note that the initial, maximum, and minimum energy parameters for (P4) are not the same as simulated energy parameters. This is okay because in real life, we can plan for this but simulate something else.

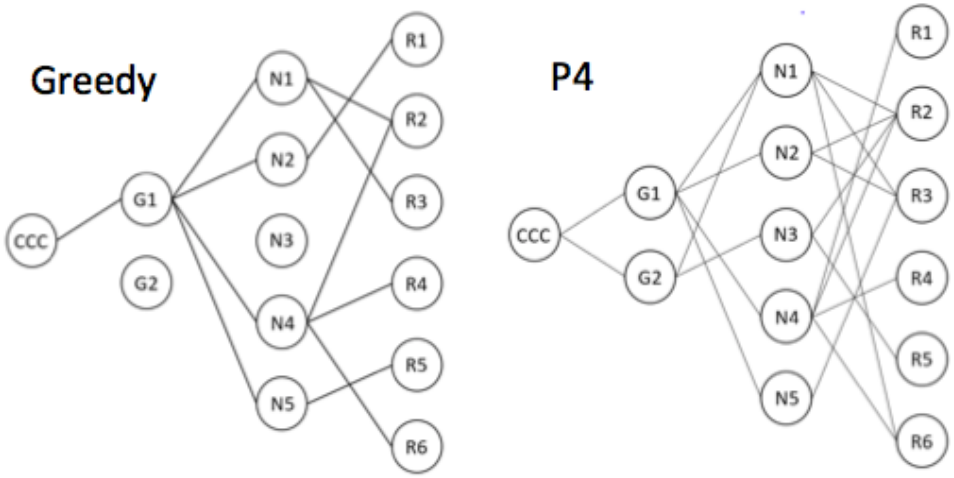

Figure 6.4: Network representation of greedy message paths and (P4) message paths. CCC is the Central Command and Control Center, G1 representations gateway 1, N1 is nanosat 1, and R1 is remote user 1.

Chapter 7

MAXIMUM FLOW

A maximum flow network problem is formulated to analyze the maximum number of messages that can be sent using a given constellation of nanosats and within a given time frame.

objective function

$$\max \sum_{m}\sum_{n}\sum_{r}\sum_{k} u_{mnrk} \tag{7.1}$$

subject to

$$\sum_{m}\sum_{g} w_{mg} \leq M \tag{7.2}$$

$$w_{mg} \geq \sum_{n}\sum_{k} v_{mgnk} \qquad \forall(m,g) \tag{7.3}$$

$$\left(\sum_{g} v_{mgnk}\right) \geq z_{mnk} + \sum_{r} u_{mnrk} \qquad \forall(m,n,k=1) \tag{7.4}$$

$$\left(\sum_{g} v_{mgnk}\right) + z_{mn(k-1)} \geq z_{mnk} + \sum_{r} u_{mnrk} \qquad \forall(m,n,k=2,...,K-1) \tag{7.5}$$

$$\left(\sum_{g} v_{mgnk}\right) + z_{mn(k-1)} \geq \sum_{r} u_{mnrk} \qquad \forall(m,n,k=K) \tag{7.6}$$

$$\sum_{m}\sum_{n} u_{mnrk} \leq 1 \qquad \forall(r,k) \tag{7.7}$$

$$\sum_{m}\sum_{r} u_{mnrk} \leq 1 \qquad \forall(n,k) \tag{7.8}$$

$$\sum_{m}\sum_{g} v_{mgnk} \leq 1 \quad \forall(n,k) \tag{7.9}$$

$$\sum_{m}\sum_{n} v_{mgnk} \leq 1 \quad \forall(g,k) \tag{7.10}$$

$$\sum_{g} w_{mg} \leq 1 \quad \forall m \tag{7.11}$$

$$\sum_{n}\sum_{r}\sum_{k} u_{mnrk} \leq 1 \quad \forall m \tag{7.12}$$

$$\sum_{g}\sum_{n}\sum_{k} v_{mgnk} \leq 1 \quad \forall m \tag{7.13}$$

$$e_{nk} = E_{n0} - \sum_{m}\sum_{r} u_{mnrk} + \delta_{nk} - h_{n,k-1} \quad \forall(n,k=1) \tag{7.14}$$

$$e_{nk} = e_{n(k-1)} - \sum_{m}\sum_{r} u_{mnrk} + \delta_{nk} - h_{n,k-1} \quad \forall(n,k=2,...K) \tag{7.15}$$

$$e_{min} \leq e_{nk} \leq e_{max} \quad \forall(n,k) \tag{7.16}$$

$$h_{nk} \geq 0 \tag{7.17}$$

$$v_{mgnk}, u_{mnrk}, w_{mg}, z_{mnk} \in \{0,1\} \tag{7.18}$$

The objective function in (7.1) maximizes the total number of message deliveries. Constraint (7.2) ensures that the total number messages delivered from the CCC to gateways is less than or equal to the total number of messages (M) waiting to be sent from the CCC. Constraint (7.3) is the flow constraint for gateways; constraints (7.4)-(7.6) are the flow constraint for nanosats. Constraint (7.7) ensures that an user receives at most one message in each time period. Constraint (7.8) ensures that a nanosat sends at most one message in each time period. Constraint (7.9) ensures that a nanosat receives at most one message in each time period. Constraint (7.10) ensures that a gateway sends at most one message in each time period. Constraint (7.11) ensures that each message is delivered to gateways at least once. Constraint (7.12) ensures that each message is delivered to remote users at least once. Constraint (7.13) ensures that each message is delivered to nanosats at leastonce. Constraints

(7.14) and (7.15) are the energy constraints which ensure that energy is stored or used according to nanosat specifications. Constraint (7.16) ensures that a nanosat energy level stays within the appropriate levels. Constraint (7.17) ensures that h_{nk} is collecting extra/spilled energy. Constraint (7.18) restricts the decision variables v_{mgnk}, u_{mnrk}, w_{mg}, z_{mnk}, to be binary.

The two small examples from Section 3.3 and the realistic problem from Chapter 5 are used for the maximum flow problem. Small example (A) has a maximum capacity of 32 messages while the small example (B) has a maximum capacity of 29 messages. The realistic problem has a maximum capacity of 58 messages. The solar charging time windows play a role in determining the maximum message delivery capacity for a given nanosat and ground node constellation, which is illustrated by difference in maximum capacity for small examples (A) and (B).

Chapter 8

CONCLUSIONS AND FUTURE WORK

Short contact intervals between ground nodes and nanosats, along with uncertainty in successful delivery of messages due to noise in the environment, create challenges when using low earth orbit nanosats for communication. This thesis presented four optimization models to enable store-and-forward communications between gateways and remote users via nanosats. Model (P1) was a deterministic demand model without energy constraints. In (P2), energy constraints were added to see the effects that energy would have on the network of nanosats and ground nodes. To model the uncertainty in successful delivery of messages, a probabilistic chance constraint was introduced in (P3). Model (P4) built upon the foundation of (P3) by including energy constraints. Models (P1), (P2), and (P3) were shown to satisfy integrality by formulating an equivalent single commodity minimum cost network flow problem for each model. This allowed for (P1), (P2), and (P3) to be solved quickly as linear programs as opposed to large integer programs. A maximum flow network problem was formulated to analyze the maximum number of messages that can be supported by a given constellation of nanosats with a time horizon.

Small examples were used to illustrate models (P1), (P2), (P3), and (P4). A realistic problem was solved using the four optimization models. The results showed that (P3) and (P4) had more well-balanced use of message paths, while (P1) and (P2) only used a small subset of message paths to deliver all the messages. The results from the realistic problem were used to simulate different policies for the simulation presented in Chapter 6. The simulation results showed that a greedy policy is sufficient when there is abundant energy in the system and a small number of message deliveries, and when messages are successfully delivered

on the first attempt. The probabilistic simulation, where messages may not be successfully delivered on the first attempt, showed that (P4) was the best policy. The greedy policy only utilized one of two gateways and therefore had limited ways to send messages to the remote users. The more well-balanced use of message paths for (P4) provided more flexibility in delivering messages to remote users.

Several assumptions were made when modeling message deliveries using a constellation of nanosats. This thesis assumed two-way communication between ground nodes and nanosats. In future work, models should be created for one-way communication where nanosats cannot simultaneously receive and send messages. The current models also assume a "store-and-forward" approach where nanosats do not communicate with one another. Crosslinks (where nanosats can communicate amongst each other) should be modeled and the effort on message delivery times should be compared to the "store-and-forward" approach. While this thesis did not allow message preemption, as done in [9], or message priority, future work should delve into the effects of message priority and preemption on the delivery times for a given nanosat constellation network.

BIBLIOGRAPHY

[1] Dariush Abbasi-Moghadam, Seyed Mehdi Hoseini-Nasab Hotkani, and Mojtaba Abolghasemi. Store and forward communication payload design for leo satellite systems. *Majlesi Journal of Electrical Engineering*, 10(3):7, 2016.

[2] AGI. Agi website. `http://www.agi.com/products/engineering-tools`, 2018 (accessed May 22, 2018).

[3] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to Linear Optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

[4] Kerri Cahoy and Andrew K Kennedy. Initial results from access: An autonomous cubesat constellation scheduling system for earth observation. 2017.

[5] Lei Li and Zelda B Zabinsky. Incorporating uncertainty into a supplier selection problem. *International Journal of Production Economics*, 134(2):344–356, 2011.

[6] OneWeb. Oneweb website. `http://www.oneweb.world/#technology`, 2018 (accessed May 22, 2018).

[7] SpaceWorks. Nano/microsatellite market forecast, 8th edition. *SpaceWorks Enterprises, Inc.*, 2018.

[8] Cliff Stein. Multicommodity flow. `http://www.columbia.edu/~cs2035/courses/ieor6614.S16/multi.pdf`, 2018 (accessed May 22, 2018).

[9] Cherry Y Wakayama, Peter J Yoo, and Zelda B Zabinsky. Energy-cognizant scheduling of store-and-forward communications with multiple priority levels in nanosatellite systems. 2016.