We formulate message scheduling optimization models for a communications network architecture containing nanosatellites, gateways, and remote users. In this architecture, messages are to be scheduled at gateways and nanosatellites during certain contact time windows to be delivered to their associated remote users using a store-and-forward approach. Two deterministic models and one probabilistic model are presented which derive message paths and schedules while addressing short contact windows, limited energy capacity and uncertainty in link quality. In the probabilistic model, uncertainty in link quality is incorporated by representing the demand to each user with a random variable and formulating it as a chance-constrained programming model. In order to compare the resulting strategies of the optimization models, a discrete-event scheduling simulator with message flow and energy dynamics is created. A simple greedy strategy is also derived by finding earliest message arrival paths to remote users based on the contact windows only. Probabilistic simulations are performed using the discrete-event simulator to compare the various strategies. Simulation results demonstrate that a well-balanced path usage and minimum average total delivery time is achieved with the strategy of the probabilistic model when compared with the deterministic models and the greedy strategy.

INTRODUCTION

Low cost, low earth orbiting nanosatellites (nanosats) are finding new applications in defense, public and commercial services [1]. Applications using nanosats include remote sensing, weather monitoring, science, and communications. In this paper, a communication network application using nanosats where nanosats serve as message ferries between ground gateways and remote users is considered. We assume gateways are connected to a central command and control center (CCC) and the schedules of gateways and nanosats are coordinated through the CCC. Remote users are located in regions without direct access to terrestrial connectivity. This paper discusses some current challenges in scheduling nanosats to enable timely communications between gateways and remote users and presents several optimization models to determine routing decisions while accounting for short contact time windows, limited energy capacity of nanosats, and uncertainty in link quality when delivering messages.

Nanosats are constrained by their small size and cannot contain large batteries. There is a maximum energy level that each nanosat can store. A nanosat cannot send messages if its energy storage falls below a minimum threshold required for message transmission. There is also uncertainty in the stability of the connection between nanosats and remote users due to environmental and communication systems noises. Multiple attempts may be needed to achieve a successful 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.

The objective of this research is to develop and analyze scheduling optimization methods that can assist with the timely delivery of messages using nanosats. We develop 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 CCC to gateways and nanosats to remote users. Two deterministic optimization models are developed, with and without energy constraints that incorporate threshold values on the energy level in the battery. The connection between nanosats and remote users may not always be well established. A probabilistic model is also developed to model the uncertainty in knowing if a message was successfully delivered. In the probabilistic model, the uncertainty in link quality is modeled using a chance constraint. The decision variables for sending messages are binary variables, and in general, it is a challenge to solve integer programs for most practical-sized problems. The deterministic optimization models formulated in this paper are shown to fit the form of a minimum cost flow network problem so integrality is ensured and there are algorithms that can be used to solve these specialized models quickly.

Comparisons are made to consider the differences between a basic deterministic model without energy constraints, a deterministic model with energy constraints and the chance-constrained model. Comparisons are also made to a simple greedy strategy to highlight the benefit of optimization.

RELATED WORK

Previous work has looked into solving routing and scheduling problems involving nanosats in a given network architecture. Wakayama et al. looks at a scheduling methodology for sending messages from one nanosat to multiple ground nodes [2]. Messages are assumed to be already stored in the nanosat and the problem is how to schedule timely delivery given energy constraints and short contact intervals [2]. In this paper, the complete routing of messages from gateways to nanosats to remote users is addressed.

Wakayama et al. considers the "store-and-forward" approach for message delivery [2]. 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 [3]. 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. This paper focuses on a "store-and-forward" approach.

In [2], Wakayama et al. 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 are integer valued, then the optimization model satisfies the integrality property [5]. This implies that the optimization problem can be solved without constraining the variables to be integer-valued, and the optimal solution will be integer-valued. 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:

\[
\text{minimize} \quad \sum \text{cost}_{\text{flow}} \times \text{flow} \quad (1)
\]

subject to

\[
\text{flow out} - \text{flow in} = \begin{cases} 
\text{supply} & \text{if source node} \\
\text{demand} & \text{if sink node} \\
0 & \text{if intermediate node}
\end{cases} \quad (2)
\]

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 [6]. Although initially the model in this paper 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.

OPTIMIZATION MODELS

Two deterministic demand network models, one probabilistic demand network model and a simple greedy strategy are formulated in this section. The models 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.

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 considered 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 the starting reference time when an optimization problem is solved.
7. Remote users’ movements are negligible relative to nanosats and are considered stationary.

**Deterministic Demand and no Energy Constraints (P1)**

A model with deterministic demands and no energy constraints (P1) is formulated. 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 (Systems Tool Kit) simulation results for a given constellation. The STK software tool allows users to define a constellation of nanosats and ground nodes (gateways and remote users) and perform coverage and sunlight analyses to generate contact time windows and solar charging time windows [7].

Model (P1) has five sets: gateways, nanosats, remote users, messages, and time intervals. Since each message is unit-sized, the units used for the models are 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. 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 needed because the contact windows between nanosats and gateways/remote users are discretized for the entire time horizon. There are four decision variables that represent the flow of messages between gateways, nanosats, and remote users.

**Sets**

- $g$: gateways
- $n$: nanosats
- $r$: remote users
- $m$: messages
- $k$: time intervals

**Parameters**

- $\tau_k$: start time of contact time interval $k$

**Decision Variables**

- $w_{mg} = \begin{cases} 
1 & \text{if message } m \text{ is sent to gateway } g \\
0 & \text{otherwise} 
\end{cases} \forall m, g$
- $v_{mgnk} = \begin{cases} 
1 & \text{if message } m \text{ is transmitted from gateway } g \text{ to nanosat } n \text{ in time interval } k \\
0 & \text{otherwise} 
\end{cases} \forall m, g, n, k$
- $u_{mnrk} = \begin{cases} 
1 & \text{if message } m \text{ is delivered from nanosat } n \text{ to remote user } r \text{ in time interval } k \\
0 & \text{otherwise} 
\end{cases} \forall m, n, r, k$
- $z_{mnk} = \begin{cases} 
1 & \text{if message } m \text{ remains in nanosat } n \text{ in time interval } k \\
0 & \text{otherwise} 
\end{cases} \forall m, n, k$

**Model (P1)**

minimize $\sum_m \sum_n \sum_r \sum_k \tau_k \times u_{mnrk}$ \hspace{1cm} (3)

subject to

$w_{mg} = \sum_n \sum_k v_{mgnk} \forall m, g$ \hspace{1cm} (4)

$\sum_g v_{mgn1} = z_{mn1} + \sum_r u_{mn1r} \forall m, n$ \hspace{1cm} (5)

$\sum_g v_{mgnk} + z_{mn(k-1)} = z_{mnk} + \sum_r u_{mnrk} \forall m, n, k = 2, \ldots, K - 1$ \hspace{1cm} (6)

$\sum_g v_{mgnk} + z_{mn(k-1)} = \sum_r u_{mnrk} \forall m, n$ \hspace{1cm} (7)

$\sum_m \sum_n u_{mnrk} \leq 1 \forall r, k$ \hspace{1cm} (8)

$\sum_m \sum_r u_{mnrk} \leq 1 \forall n, k$ \hspace{1cm} (9)

$\sum_m \sum_g v_{mgnk} \leq 1 \forall n, k$ \hspace{1cm} (10)

$\sum_m \sum_r v_{mgnk} \leq 1 \forall g, k$ \hspace{1cm} (11)
\[ \sum_{g} w_{ng} = 1 \quad \forall m \quad (12) \]
\[ \sum_{v} \sum_{k} u_{vmrk} = 1 \quad \forall m \quad (13) \]
\[ \sum_{g} \sum_{v} \sum_{k} v_{mngk} = 1 \quad \forall m \quad (14) \]
\[ \sum_{m} \sum_{v} \sum_{k} u_{vmrk} = d_{r} \quad \forall r \quad (15) \]
\[ v_{mngk}, u_{vmrk}, w_{mg}, z_{mnk} \in [0,1] \quad (16) \]

The objective function in (3) minimizes the sum of the delivery times of messages. Constraint (4) is the flow constraint for gateways; constraints (5), (6), and (7) are the flow constraints for nanosats. Constraint (8) ensures that a remote user receives at most one message in each time period. Constraint (9) ensures that a nanosat sends at most one message in each time period. Constraint (10) ensures that a nanosat receives at most one message in each time period. Constraint (11) ensures that a gateway sends at most one message in each time period. Constraint (12) ensures that each message is delivered to gateways once. Constraint (13) ensures that each message is delivered to remote users once. Constraint (14) ensures that each message is delivered to nanosats once. Constraint (15) ensures that user demand, \( d_{r} \), is met. Constraint (16) restricts the decision variables to be binary.

A minimum cost flow network problem, shown in (1) and (2), satisfies the integrality property if the right-hand side of the constraints are integer-valued [5]. Model (P1) can be shown to satisfy the integrality property by creating additional dummy nodes for messages and representing it in the form of (1) and (2). An equivalent model to (P1) is created with nodes \( CCC, G_{mg}, N_{mnk}, \) and \( R_{mr} \). 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., \( d_{mr}=1 \). 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 \( \tau_{k} \). The flow variables \( f_{(CCC,G_{mg},G_{mg},(Nmnk), (Nmnk), (Rmr), (Nmnk), (Nmnk-k-1))} \) can be mapped to \( w_{mg}, v_{mngk}, u_{vmrk}, \) and \( z_{mnk} \) from (P1). The equivalent problem written with the expanded network flow is given below. An example of the minimum cost flow network representation using two gateways, two nanosats, two remote users and two messages is shown in Figure 1.

**Equivalent model of (P1)**

\[ \text{minimize} \quad \sum_{m} \sum_{v} \sum_{k} \tau_{k} \times f_{(Nmnk),(Rmr)} \quad (17) \]

\[ \text{subject to} \]

\[ \sum_{m} \sum_{v} \sum_{k} u_{vmrk} = d_{r} \quad \forall r \quad (18) \]
\[ \sum_{v} \sum_{k} f_{(Ng, (Nm), (G_{mg})}(N_{mnk}) = \sum_{m} \sum_{r} d_{mr} \quad (19) \]
\[ f_{(Nmnk),(Rmr)} - f_{(CCC),(G_{mg})}(N_{mnk}) = 0 \quad \forall G_{mg} \quad (20) \]
\[ \sum_{g} f_{(G_{mg})}(Nmnk) = 0 \quad \forall N_{mnk} \quad (21) \]
\[ -\sum_{k} f_{(Nmnk),(Rmr)} = -d_{mr} \quad (22) \]
\[ 0 \leq f_{(i,j)} \leq 1 \quad \forall \text{nodes} (i) \& (j) \quad (23) \]

**Figure 1: Minimum cost network flow representation of (P1)**

**Deterministic Demand and Energy Constraints (P2)**

Model (P2) differs from (P1) in that energy constraints are considered. Four additional parameters are included, \( \epsilon_{min}, \epsilon_{max}, E_{0}, \) and \( \delta_{nk} \), where \( \epsilon_{min} \) and \( \epsilon_{max} \) represent the minimum and maximum energy capacity for nanosats in terms of number of unit-messages, \( E_{0} \) is the initial energy level at nanosat \( n \) at \( t_{0} \) in terms of number of unit-messages, and \( \delta_{nk} \) 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 simulation. In (P2), \( \delta_{nk} \) is an intermediate (decision) variable to capture spilled energy or extra energy not needed for message delivery.
Model (P2)

Objective function (3)

Constraints (4) – (16)

\[ e_{nk} = e_{n(k-1)} - \sum_m \sum_r u_{mnrk} + \delta_{nk} - h_{n(k-1)} \]

\[ \forall n, k = 1, ..., K \quad (25) \]

\[ e_{n0} = E_{n0} \quad \forall n \quad (26) \]

\[ e_{\min} \leq e_{nk} \leq e_{\max} \quad \forall r, k \quad (27) \]

\[ h_{nk} \geq 0 \quad (28) \]

Objective function (3) and constraints (4) to (16) are from (P1). Constraint (25) is the energy constraints which ensures that energy is stored or used according to nanosat energy dynamics. Constraint (26) represents the initial energy level at each nanosat. Constraint (27) ensures that a nanosat energy level stays within the appropriate levels. Constraint (28) ensures that \( h_{nk} \) is collecting extra/spilled energy.

Model (P2) can also be shown to satisfy the integrality property by creating dummy nodes and recasting it in the form of (1) and (2). An equivalent representation of (P2) as a minimum cost network flow problem is given in Appendix.

**Deterministic Greedy Strategy**

We consider a straight forward greedy strategy that provides path assignments for messages. The greedy strategy takes the earliest arrival paths from the CCC to remote users based on contact windows between nanosats and ground nodes. The greedy strategy is derived by first identifying the nanosat with the earliest contact window for each remote user. Then, for each nanosat, the greedy strategy determines the gateway with the earliest contact window. This determines a path for every message from each gateway to nanosat to remote user. Note that the greedy strategy provides path solutions but does not schedule the message delivery. Message deliveries from gateways and nanosats can be scheduled using a first-in-first-out rule.

**A Small Example E1 using (P1), (P2) and Greedy Strategies**

A small example E1 is created with 2 gateways, 2 nanosats, 2 remote users, 6 messages, and 36 time units. Figure 2 shows the network representation for the small example with 4 time units. The contact time windows (with values for \( \tau_k \)) between nanosats and gateways and remote users are shown in Figures 3 and 4. The demand at each remote user, \( d_r \), is shown in Table 1.

Models (P1) and (P2) are solved for the small example using CPLEX [8]. The solutions for (P1) and (P2) are illustrated in Figure 5(a) and 5(b) respectively. With (P1), all six messages were scheduled tightly within 8 time intervals with the last message delivery being scheduled at 70 seconds after the start time. Note that the scheduled times may be different from the actual delivery times. Message delivery from the nanosat to the user cannot happen during a contact window if the nanosat does not have enough energy. The message delivery will then be postponed until a later contact time window between the nanosat and the user. With energy consideration in (P2), the last message transmission was scheduled at 5410 seconds. The message delivery times are delayed due to nanosats not having sufficient energy for message delivery at current contact windows and waiting to be recharged enough energy for delivery at next contact windows.

![Figure 2: Network representation for (P1)'](image)

<table>
<thead>
<tr>
<th>Table 1: Demand for Example E1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote User ((r))</td>
</tr>
<tr>
<td>Demand ((d_r))</td>
</tr>
</tbody>
</table>

The solutions using the greedy strategy is shown in Figure 4(d). Messages 1, 2 and 6 to user 1 are scheduled on gateway 2 and nanosat 2, and messages 3, 4, and 5 to user 2 are scheduled on gateway 1 and nanosat 1. The last message to user 1 is scheduled at 5410 seconds and the last message user 2 at 5430 seconds.

It is observed that without energy consideration model (P1) produces tightly packed message delivery schedules during contact windows. With energy consideration,
Figure 3: Contact time windows and solar charging time windows

Figure 4: Contact time windows after discretization

Figure 5: Visual representations of the solutions for E1: (a) (P1), (b) (P2), (c) (P3), and (d) greedy strategy. Two gateways are represented by G1 and G2, two nanosats by N1 and N2, and two remote users by R1 and R2.
model (P2) produces schedules that are more stretched out over multiple contact windows. The greedy strategy produces schedules that ignore other available contact opportunities to a particular remote user.

**Probabilistic Demand and Energy Constraints (P3)**

Similar to models (P1) and (P2), this model (P3) considers unit-messages being sent from gateways to remote users. 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 Li & Zabinsky’s supplier selection problem is used to model the uncertainty of the nanosat to remote user connection [9]. The demand constraint (15) from (P1) is changed into a chance constraint (29) where user demand is a random variable, $D_r$.

$$P(\sum_m \sum_n \sum_k u_{mnrk} \geq D_r) \geq 1 - \alpha \ \forall r \tag{29}$$

**Linearizing Chance Constraint**

Constraint (29) 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 $F_{D_r}^{-1}(1 - \alpha)$. We define a discrete probability distribution function to model the random variable representing the number of message delivery attempts. In this paper, each message can be successfully delivered in three or less message delivery attempts. 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 (30). Note that the probability of a message being delivered in three or less attempts equals one.

$$p_i = \begin{cases} 0.7 & i = 1 \\ 0.2 & i = 2 \\ 0.1 & i = 3. \end{cases} \tag{30}$$

The $D_r$ 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 greater than or equal to 0.9. The smallest integer value that satisfies this probability is $F_{D_r}^{-1}(1 - \alpha)$. Constraint (28) 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 (3)**

Constraints (4) – (11), (16), (25) – (28)

$$\sum_g w_{ng} \geq 1 \ \forall m \tag{31}$$

$$\sum_n \sum_r \sum_k u_{mnrk} \geq 1 \ \forall m \tag{32}$$

$$\sum_g \sum_n \sum_k v_{ngmnk} \geq 1 \ \forall m \tag{33}$$

$$\sum_m \sum_n \sum_k u_{mnrk} \geq F_{D_r}^{-1}(1 - \alpha) \ \forall r \tag{34}$$

Constraint (31) ensures that each message is delivered to gateways at least once. Constraint (32) ensures that each message is delivered to remote users at least once. Constraint (33) ensures that each message is delivered to nanosats at least once. Constraint (34) is the equivalent representation of the chance constraint (30) which ensures that the probability that remote user demand, $D_r$, is met will be greater than or equal to some large value, $(1 - \alpha)$.

**A Small Example E1 using (P3)**

The results of the small example E1 using (P3) are shown in Figure 4(c). Model (P3) anticipates the need to send messages more than once, and produces a hedging strategy with wiggle rooms. While there can be multiple delivery attempts for each message, only the message path that first delivers the message is used as the solution. Comparing the solution shown in 4(c) with that of 4(a) and 4(b), it is observed that different paths for messages are selected for different models. Model (P1) shows an optimistic approach to scheduling and message schedules are packed very tightly. With energy consideration, (P2) schedules messages in multiple contact windows. With energy and uncertainty consideration, model (P3) generates a more conservative schedule with wiggle rooms for message retransmissions.

**A LARGER REALISTIC EXAMPLE E2**

We now construct a larger example, E2, using realistic problem data. Simulation data from STK is used to determine contact time windows and solar charging windows for 2 gateways, 5 nanosats, and 6 remote
users. A 12-hour simulation time window is set 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) using the probabilistic method outlined in the previous section.

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 280 time intervals. Thirty-nine message delivery attempts were made for (P3). Model (P3) took the longest to successfully deliver the 21 messages. We also observe that not all the message delivery paths and times for (P3) are implementable in reality. For example, in (P3) solutions, the first attempt to deliver message 2 to the remote user from a nanosat occurs at \(k = 90\), which is before the delivery to the nanosat at \(k = 224\). However, for each message at least one realizable delivery attempt is achieved because for the flow conservation constraints to be fulfilled, there must be at least one flow \(V_{mngk}\) that is less than or equal to a flow \(u_{mnrk}\) for all \(m\) and \(n\).

Figure 6 shows a network representation of the message paths for (P1), (P2), (P3), and the greedy strategy. For (P1), (P2), and greedy only one gateway was used. In contrast, (P3) used both gateways and all five nanosats. The message paths taken in (P3) in Figures 6(c) are more well balanced than the paths taken in (P1), (P2) and greedy in Figures 6(a), 6(b) and 6(d).

**COMPARISON OF GREEDY AND OPTIMIZATION STRATEGIES USING SIMULATION**

A discrete event scheduling simulator is created to compare different scheduling strategies. The simulator has six main components: the STK Simulation Database, Nanosat, Central Command & Control (CCC), Ground Gateways, Remote Users, Main Simulation, and Performance Evaluation. A block diagram for the simulator is shown in Figure 7.

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.

The CCC object stores the database for ground nodes and nanosats including contact time windows. The CCC object includes methods for message generation, route generation using the greedy strategy or solving the optimization models (P1), (P2), and (P3), and updating message queues and arrival times at gateways and remote users. Messages are generated at CCC 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.

A discrete event simulator is constructed in the main simulation module. 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 the gateway and nanosat for a message) is performed based on solving a specified optimization model (P1), (P2), (P3), or by the greedy strategy. The simulator maintains message queues at gateways and nanosats and transmits and receives messages during contact windows so long as there is enough energy available on nanosats for transmission. Note that the message transmission times to nanosats and remote users by the simulator may be different from the schedules obtained by solving the optimization models. However, the path assignments and the ordering of message deliveries are the same as that of the optimization solutions.

A probabilistic message delivery simulation is performed. In the simulation, the number of downloads for each message is sampled from a given discrete probability mass function (as in (30), \(P(\text{number of downloads} = 1) = 0.7, P(\text{number of downloads} = 2) = 0.2, P(\text{number of downloads} = 3) = 0.1\)). 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 delivery attempts.

The performance evaluation calculates the delay for each message, i.e., the difference between the arrival time at the remote user and the message generation at the CCC. 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 strategy (greedy, (P1), (P2), or (P3)) are evaluated including average, minimum, maximum, and standard deviation.

**Simulation Results**

The simulation results using the parameters of the realistic problem E2 are discussed in this section. However, a 24-hour time window was used for the simulation (instead of the 12-hour time window used in Example E2). For each strategy, the simulation was run ten times with different samples of the number of downloads for each message and the results are shown in Table 3.

On average, simulated (P3) performed the best for the probabilistic simulation in terms of the average total delivery time. The greedy strategy performed better than simulated (P1) although the difference in performance was insignificant. Simulated (P2) performed slightly better than the greedy strategy in terms of average total delivery time. However, it took slightly longer for all the messages to be delivered for simulated (P2) than for the greedy strategy. A larger number of message delivery attempts can be accommodated using more well-balanced message paths. (P3) has a more well-balanced use of paths than the greedy strategy. The greedy strategy only utilizes one of the two gateways and does not utilize nanosat 3. (P3) uses both gateways and all five nanosats (including nanosat 3).

**Table 3: Probabilistic simulation results showing Greedy, simulated (P1), simulated (P2), and simulated (P3)**

<table>
<thead>
<tr>
<th></th>
<th>Greedy</th>
<th>Simulated (P1)</th>
<th>Simulated (P2)</th>
<th>Simulated (P3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Total Delivery Time (seconds)</td>
<td>1,999,760</td>
<td>2,078,320</td>
<td>1,998,210</td>
<td>1,925,410</td>
</tr>
<tr>
<td>Average Minimum Delivery Time (seconds)</td>
<td>72,830</td>
<td>75,690</td>
<td>72,850</td>
<td>73,510</td>
</tr>
<tr>
<td>Average Maximum Delivery Time (seconds)</td>
<td>129,520</td>
<td>135,410</td>
<td>135,660</td>
<td>126,270</td>
</tr>
</tbody>
</table>

**MAXIMUM FLOW NETWORK**

Maximum flow network problems involve determining a maximum amount of feasible flow in a constrained network. With slight modification, the network flow model was reformulated as a maximum flow network problem to find the maximum number of messages that could be sent for a given network of nanosats, gateways and users over a given time frame. The maximum flow problem of (P2) was applied to examples E1 and E2. Example E1 has a maximum capacity of 29 messages with 14 messages destined to remote user 1 and 15 messages for remote user 2. Example E2 has a maximum capacity of 58 messages over a time frame of 12 hours. 33 messages were sent to remote user 1, seven messages to remote user 2, one message to remote user 3, six messages to remote user 4, eight messages to remote user 5, and three messages to remote user 6. The unbalance in the number of
messages to different users may be due to the non-uniform contact opportunities between the users and the nanosats over the specified time frame.

CONCLUSIONS AND FUTURE WORK

Short contact intervals between ground nodes and nanosats, along with limited energy and uncertainty in successful delivery of messages due to noise in the environment, create challenges when using low earth orbiting nanosats for communications. This paper presents three optimization models to enable store-and-forward communications between gateways and remote users using nanosats. Model (P1) is 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). Models (P1) and (P2) were shown to satisfy integrality by formulating an equivalent single commodity minimum cost network flow problem for each model. This allowed for (P1) and (P2) to be solved quickly as linear programs as opposed to large integer programs. A maximum flow network problem is also formulated to analyze the maximum number of messages that can be supported by a given network of nanosats, gateways and users over a specified time frame.

A small example E1 and a larger realistic example E2 were used to illustrate optimization models (P1), (P2), and (P3). The results showed that (P3) produced a more well-balanced use of message paths, while (P1) and (P2) only used a small subset of message paths to deliver all the messages. A discrete-event simulator on message scheduling was used to implement and test different policies of optimization models and compare to a greedy strategy for E2. The probabilistic simulation results on an example network E2 showed that simulated (P3) is the best strategy in terms of delivery times and a more balanced usage of gateways and nanosats allowing the strategy more flexibility in delivering messages to remote users. The greedy strategy only utilized one of two gateways and therefore had limited ways to send messages to the remote users.

The current models consider a "store-and-forward" approach where nanosats do not communicate with one another. In future work, crosslinks (where nanosats can communicate amongst each other) will be modeled, and the computation complexity and message delivery performance will be compared to the "store-and-forward" approach. While this paper did not address message preemption or message priority, future work should delve into the effects of message priority and preemption on the delivery times for a given nanosat communications network.

ACKNOWLEDGMENTS

This work was funded under SPAWAR Naval Innovative Science and Engineering (NISE) program.

REFERENCES


APPENDIX

Optimization model (P2) is shown to be represented by a minimum cost flow network problem and thereby it satisfies the integrality property. Model (P2) has the same nodes and edges as (P1). However, energy and messages share the same edges. An equivalent model is created with nodes $e_{nk}, G_{mgk}, N_{nuk}, R_{nr},$ 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 one. The supply to each \( e_{nk} \) node is \( \delta_{nk} \) 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 \( \sum_{m} \sum_{n} d_{mr} \). The cost for flows not from nodes \( N_{mk} \) to \( R_{mr} \) is zero; cost for flows from nodes \( N_{mk} \) to \( R_{mr} \) is \( \tau_{K} \).

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 \( e_{\text{min}} \) and \( e_{\text{max}} \). The flow variables \( f_{(enk),(extra)} \), \( f_{(enk),(Gmgnk)} \), \( f_{(enk),(en(k+1))} \), \( f_{(Gmgnk),(Gmgnk+1)} \), \( f_{(Gmgnk),(Nmk)} \), \( f_{(Nmk),(Nmk(k+1))} \), \( f_{(Nmk),(Nmk(k+1))} \), and \( f_{(Nmk),(Rmr)} \) can be mapped to \( e_{nk} \), \( h_{nk} \), \( W_{mg} \), \( V_{mgnk} \), \( U_{mk} \), and \( z_{mk} \) from (P2). In the following, an equivalent minimum cost network flow model is given. A graphical representation of the minimum cost network flow model with two gateways, two nanosats, two remote users and two messages is shown in Figure 8.

**Equivalent model of (P2)**

\[
\text{minimize} \quad \sum_{m} \sum_{n} \sum_{r} \sum_{k} \tau_{K} \times f_{(N_{mk})}(R_{mr})
\]

subject to

\[
\delta_{n1} + E_{n0} = f_{(e_{n1}), (e_{n2})} + \sum_{m} \sum_{g} f_{(e_{nk}), (G_{mgnk})} + f_{(e_{nk}), (extra)} \quad \forall n
\]

\[
\delta_{nk} + f_{(e_{nk}), (e_{nk})} = f_{(e_{nk}), (en(k+1))} + \sum_{m} \sum_{g} f_{(e_{nk}), (G_{mgnk})} + f_{(e_{nk}), (extra)} \quad \forall n, k = 2, ..., K - 1
\]

\[
\delta_{nk} + f_{(e_{nk}), (e_{nk})} = \sum_{m} \sum_{g} f_{(enk), (G_{mgnk})} + f_{(enk), (extra)} \quad \forall n
\]

\[
f_{(e_{n1}), (G_{mg1})} = f_{(G_{mg1}), (G_{mg2})} + f_{(G_{mg1}), (N_{mnt})} \quad \forall m, g, n
\]

\[
f_{(e_{nk}), (G_{mgnk})} + f_{(G_{mgnk}), (G_{mgnk})} = f_{(G_{mgnk}), (G_{mgnk})} + f_{(G_{mgnk}), (N_{mnt})} \quad \forall m, g, n, k = 2, ..., K - 1
\]

\[
f_{(e_{nk}), (G_{mgnk})} + f_{(G_{mgnk}), (G_{mgnk})} = f_{(G_{mgnk}), (N_{mnt})} \quad \forall m, g, n
\]

\[
\sum_{g} f_{(G_{mg1}), (N_{mnt})} = f_{(N_{mnt}), (N_{mnt})} + f_{(N_{mnt}), (R_{mr})} \quad \forall m, g, r
\]

\[
\sum_{g} f_{(G_{mgk}), (N_{mnt})} + f_{(N_{mnt(k-1)}), (N_{mnt})} = f_{(N_{mnt(k-1)}), (R_{mr})} \quad \forall m, g, n, k = 2, ..., K - 1
\]

\[
\sum_{g} f_{(G_{mgk}), (N_{mnt})} + f_{(N_{mnt(k-1)}), (N_{mnt})} = f_{(N_{mnt(k-1)}), (R_{mr})} \quad \forall m, g, n
\]

\[
\sum_{n} f_{(N_{mk}), (R_{mr})} = d_{mr} \quad \forall r
\]

\[
\sum_{n} \sum_{k} f_{(e_{nk}), (extra)} = \sum_{n} \sum_{k} (\delta_{nk} + E_{n0}) - \sum_{m} \sum_{r} d_{mr}
\]

**Figure 8: Minimum cost network flow representation of (P2)**