Download Traffic Scheduling for CubeSats Swarms with Inter-Satellite Links

Yawen Zheng, Kwan-Wu Chin
School of Electrical, Computer and Telecommunications Engineering
University of Wollongong, NSW, Australia
Email: yz757@uowmail.edu.au, kwanwu@uow.edu.au

Luyao Wang
Center for Future Internet Technology
Beijing University of Technology, China
Email: luyaowang1990@bjut.edu.cn

***Abstract*—CubeSats are small sized satellites constructed using commercial off-the-shelf components and weigh no more than 1.3 Kg. They are ideal as an educational tool as well as low-cost platforms for space research. A key advantage of CubeSats is their ability to form swarms. A fundamental problem, however, is maximizing the total downloaded data from a CubeSats swarm to ground stations. The key challenges include intermittent contact time and duration with ground stations, and limited resources. Henceforth, this paper aims to study data transmission policies from CubeSats to ground stations. Unlike prior works, we quantify the benefits of transferring data over Inter-Satellite Links (ISLs). We investigate four satellite-to-ground ($s-g$) pairing rules; namely, Pair Transmit Most Data (PTMD), Random Pair (RP), Shortest Contact Duration (SCD), and Round Robin (RR). Our results indicate that ISLs help CubeSats ensure they have more data to download to ground stations. Moreover, the use of ISLs simplifies $s-g$ pairing because any of the aforementioned rules can be used to yield similar amount of downloaded data.**

***Index Terms*—CubeSats, Inter-Satellite Communications, Traffic Download, Transmission Policies**

1. Introduction

CubeSats are small, highly capable and compact satellites. A basic CubeSat, also called 1U (Unit), is a $10\times10\times10$ cm^3 satellite that weighs no more than 1.3 Kg with a volume of one litre. Multiple CubeSats can be connected together to form a 3U CubeSat; see Figure 1. A CubeSat can also be classified as a large picosatellite with a wet mass between 0.1 and 1 Kg or a nanosatellite with a wet mass between 1 and 10 Kg. Advantageously, CubeSats can be launched as auxiliary payloads of existing missions and multiple CubeSats can be deployed simultaneously. In fact, a number of CubeSats have been launched from the International Space Station (ISS); see [1], [2] and references therein for details.

To date, CubeSats have been used (i) as a teaching tool [3], and (ii) in various science missions, including earth observation, atmospheric science, and space weather to name a few; see [4] for more details. They can also function as a low-cost, fault-tolerant Synthetic Aperture Radar (SAR) [5]. Apart from that, CubeSats are becoming critical for deep-space missions [6]. In particular, a swarm of CubeSats can be deployed throughout our solar system and function as a distributed sensor network. For example, NASA plans to release a swarm of CubeSats in the Jovian system to complement their mission on Europa [7]. In another project, NASA intends to use two 1U CubeSats as relays between Mars and Earth [8].

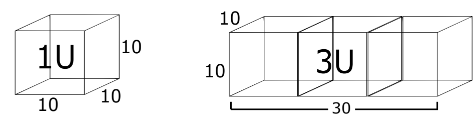

Figure 1. 1U and 3U CubeSats.

A swarm of CubeSats has many advantages. First, CubeSats are able to pool their resources together and complete a task cooperatively. For example, if the task at hand is to photograph a given area and it is cloudy, then the first CubeSat can inform other CubeSats to photograph the area continously or use CubeSats that are at a different vantage point. Second, compared to using only one CubeSat, a swarm of CubeSats has higher fault tolerance and as they are able to share workload, each CubeSat is only required to operate for a short period of time. Thirdly, CubeSats can help each other forward data to ground stations. For example, in Figure 2, CubeSats may exchange data over an Inter-Satellite Link (ISL) to ensure the maximum amount of data is transferred to a ground station. This is especially advantages when some CubeSats have a higher energy level or/and better channel conditions or/and are in contact with a ground station sooner or/and longer.

A key problem is optimizing the data transferred by CubeSats to ground stations. This is significant because the usefulness of Cubesats is directly proportional to the amount of scientific measurements or surveillance data that can be downloaded to ground stations. To do this, we need to address a key *pairing* problem: given a set of CubeSats that are visible to a ground station, which CubeSat should be connected to it? One important constraint is that each ground station can only be paired with *one* CubeSat at a time. Our goal is thus to determine the *best* CubeSat and ground station pair that will lead to a higher total download. Moreover, we aim to make use of ISLs. Specifically, if a CubeSat has insufficient data to saturate its downlink, then

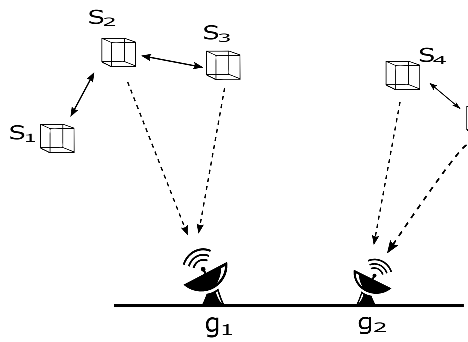

Figure 2. A CubeSat swarm

it picks and transfers data from an unpaired CubeSat. For example, in Figure 2, assume S_3 is paired with g_1 for one second and the downlink is capable of 9600 bps. Assume both S_3 and S_2 have 1000 bits of data. We see that S_3 is unable to fully utilize its downlink capacity. To this end, S_3 can transfer some data from S_2 to better utilize its downlink to ground station g_1.

In this paper, we aim to quantify the benefits of ISLs in maximizing the amount of data downloaded to ground stations, and also whether the use of ISLs affects the rule used to pair a CubeSat and a ground station; so called $s-g$ or downlink pairing rule. The rules of interest include,

- **Pair Transmit Most Data (PTMD)**. This rule activates $s-g$ links that have the highest data flow.
- **Random Pair (RP)**. Pair a CubeSat with each ground station randomly.
- **Shortest Contact Duration (SCD)**. Picks the CubeSat with the shortest contact duration.
- **Round Robin (RR)**. The $s-g$ link that has been paired the least gets preference.

Our results show that ISLs help improve the total amount of data downloaded to ground stations. Interestingly, with ISLs, all downlink pairing rules achieve similar results. This implies that any downlink ($s-g$) rule can be used as ISLs help CubeSats improve the utilization of their downlinks.

Next, in Section 2, we discuss related works and identify key gaps. Then in Section 3 we define our notations before presenting the problem in Section 4. After that Section 5 discusses link selection rules. This is followed by our evaluation methodology in Section 6. We then present our conclusions in Section 7.

2. Related Works

To date, references [9], [10], [11] and [12] have considered the problem of maximizing data download from CubeSats. Both works in [9] and [10] propose a Mixed Integer Linear Program (MILP) to calculate the maximum possible downloaded data from a CubeSat swarm. They also proposed a heuristic that pairs CubeSats to ground stations that can download the most data. They, however, did not consider ISLs and did not evaluate their heuristic against other rules. The works in [11] and [12] indicate that ISLs can help download data to single ground device. Lv et al. [11] study joint routing and link scheduling to achieve the maximum downloading data throughput. The authors propose a novel scheme that uses ISLs to offload data among satellites before they contact with one ground device. Based on [11], the authors developed another iterative optimization algorithm that schedules both data offload among satellites and data downloading from satellites to the ground device. This optimization algorithm constructs a bipartite graph that finds the maximal matching in each time slot to offload data from nodes with more data to nodes with spare time. Spangelo et al. [13] study the maximum download capacity afforded by a collection of ground stations. They did not consider pairing CubeSats and ground stations, and transmission over ISLs. A number of works have considered a constellation of satellites. For example, in [14], the authors consider forwarding data over a two-tier satellite network comprising of *workers* and *messengers* satellites; only messenger satellites have a downlink to ground stations. The key problem is ensuring workers satellites do not overwhelm the queue of messenger satellites. Their aim is not to maximize data download. Also, ISLs are only used between workers and messengers; i.e., not between messengers. Hence, unlike our work, ISLs are not used to maximize data download from messenger satellites. The authors of [15] consider maximizing traffic transmitted to a Geostationary Earth Orbit (GEO) satellite that has continuous visibility to a ground station. In our case, we do not aim to forward *all* data to one satellite. Moreover, each CubeSat has opportunities to be in the coverage of at least one ground station over time. In [16], the authors characterize the downlink of a swarm of 50 CubeSats that operate on a lunar orbit. The swarm functions as a space-based telescope where CubeSats sample cosmic noise; sampled data are transmitted over ISLs and processed in a distributed manner. They assume one ground station whereas we consider multiple ground stations and also how satellites pair with each ground station.

There is a body of works that aims to schedule the time window in which one or more satellites are required to observe a given target or an area. The observed data must then be downloaded to a ground station within a given time window. These works aim to maximize a certain reward; see [17] and references therein. These works, however, do not consider ISLs. A closely related set of works is on the deep space network scheduling problem [18]. This problem considers requests from spacecrafts to pair with a ground station. The goal is to maximize the number of successful requests. This problem, however, does not consider maximizing data download or assume spacecrafts are connected via ISLs. In [19], the authors consider the problem of downloading one file from a Master CubeSat. The file is first divided into chunks that are then distributed slave CubeSats. Each slave CubeSat then transmits the received chunks to a ground station. Our problem is different because we consider downloading data from *all* CubeSats.

We note that many works have considered using satellite networks as a *communication* backbone. Examples include reference [20], which considers the problem of admitting and rerouting calls, and reference [21] aims to maximize the throughput between source and destination satellite pairs. These works, however, do not consider CubeSats nor aim to maximize the total data downloaded from satellites.

In summary, whilst there are a number of works that aim to pair satellites with ground stations in order to meet one or more objectives, none of them have considered ISLs to improve data download. This paper thus fills this gap.

3. Preliminaries

We now define formally the network under consideration. There is a set of nodes $V = \mathcal{S} \cup \mathcal{G}$, where $\mathcal{S}$ is the set of CubeSats and $\mathcal{G}$ is the set of ground stations. All ground stations are connected by the Internet, and are able to forward any downloaded data to a central server. Moreover, all ground stations are aware of the contact time and duration of CubeSats. This information is readily available because the orbit of CubeSats is known. The set of links is denoted as E, where $(i, j) \in E$ is a directed link between node i and j, where i and j can either be a CubeSat or a ground station. Although this paper focuses only on downlinks, we note that ground stations have an uplink to CubeSats; for example, the CubeSats developed for the QB50 project [22] have an uplink with data rates between 1.2 and 9.6 kbps. CubeSats can thus be informed of their transmission schedule and also the contact time and duration with ground stations.

Each CubeSat has two half-duplex radios [2]. One radio is used for ISLs and operates over S-band (2.45 GHz). ISLs do not interfere because those that do in each time t can be tuned to an orthogonal channel. Also, CubeSats can send HELLO messages over their ISLs radio to discover other CubeSats [1]. Another radio is for communication with ground stations. For example, the CubeSats that are part of the QB50 project have a downlink radio that operates in the Very High Frequency (VHF) band. Each ground station can only be paired with one CubeSat at a time [10].

The graph or network at time t is denoted as $G^t(V^t, E^t)$, where $V^t \subseteq V$ and $E^t \subseteq E$. The set of CubeSats and ground stations at time t is denoted as $\mathcal{S}^t$ and $\mathcal{G}^t$, respectively. We assume time is discrete with each slot indexed by t. Each slot has duration τ. We write $\mathcal{N}^t(i)$ as the set of neighbors in which a node i can communicate with at time t. Also each link $(i, j) \in E^t$ has rate r^t_{ij}.

Each CubeSat i receives e^t_i amount of energy (in Joules) from its solar panels at every time t. Its battery, denoted as E^t_i, has capacity $\mathcal{B}$. We thus have for all time t, CubeSat i has at most $E^t_i = \text{MIN}(E^{t-1}_i + e^t_i, \mathcal{B})$ of energy. Each CubeSat has a constant power consumption rate of p_c incurred by its subsystems; e.g., Attitude Determination and Control (ATDC). The transmission and reception cost over a downlink and ISL is respectively p_{txDL} and p_{rxDL}, and p_{txISL} and p_{rxISL}; all in J/bit. Hence, for each CubeSat i, we must ensure,

$$E^{t+1}_i = E^t_i - p_c\tau - C^t_{i,DL} - C^t_{i,ISL} \geq 0 \quad (1)$$

where $C^t_{i,DL}$ and $C^t_{i,ISL}$ correspond to the total energy used over a downlink and an ISL, respectively. Specifically, we have $C^t_{i,DL} = f^t_{i,txDL} p_{txDL}$, and $C^t_{i,ISL} = f^t_{i,txISL} p_{txISL} + f^t_{i,rxISL} p_{rxISL}$. Note, only $f^t_{i,txISL}$ or $f^t_{i,rxISL}$ is non zero in each t because CubeSats use a half-duplex radio and can only transmit/receive to/from at most one CubeSat at any time. Also, $f^t_{i,txDL}$ is zero if a CubeSat is not paired with a ground station at time t. Moreover, the amount of data transmitted is limited by r^t_{ij}, where (i, j) is either an ISL or a downlink.

CubeSat i receives d^t_i bits at each time t, e.g., from sensors, into its data storage D^t_i with capacity D_{max}. This means the data storage D^t_i of CubeSat i evolves as

$$D^t_i = D^{t-1}_i + d^t_i + f^{t-1}_{i,rxISL} - f^{t-1}_{i,txDL} \quad (2)$$

The foregone expression must satisfy the following conditions: $D^t_i \geq 0$ and $\text{MIN}(D^t_i, D_{max})$. In words, a CubeSat cannot transmit more data than what it has and if it received more than D_{max} bits, then $D^t_i - D_{max}$ bits are lost.

4. The Problem

We are now ready to define our problem. Given $G^t(V^t, E^t)$, where $t = 1, \ldots T$, we aim to maximize the amount of data downloaded by ground stations; i.e., $\text{MAX} \sum_{t=1}^{T} \sum_{i \in \mathcal{S}} f^t_{i,DL}$. Here, T is an arbitrary value and can be the time when the CubeSat swarm orbits the earth once or a planning horizon. Each CubeSat's energy and storage buffer must evolve as per (1) and (2) and meet their respective constraints. Moreover, each ground station must only be paired with one CubeSat and vice-versa at each time t. Also, each CubeSat must either transmit or receive over an ISL at each time t.

The key problem is selecting the downlinks and ISLs in each t. Consider Figure 3. Assume CubeSats do not have any energy constraint and $\tau = 1$. At $t = 1$, CubeSat S_1 can forward all its 100 bits to S_3 via the ISL between S_1 and S_3, and thus allowing S_3 to transfer 200 bits to g_2. Also note that g_1 will be paired with S_2. The total data downloaded at $t = 1$ is therefore $50 + 200 = 250$ bits. At $t = 2$, all CubeSats receive a further 100 bits of data, and all of them are within the visibility of ground station g_2 but not g_1. At this time, we see that the best pairing is between S_1 and g_2, which yields 100 bits of data.

5. Link Selection Policies

Our interest is the link selection policy used in each time t. The key function is *SelectLinks()*; see Algorithm 1. It first pairs CubeSats with ground stations according to either PTMD, RP, SCD, or RR; line-1. These rules are explained later. We term CubeSats selected by the said rule as *paired* CubeSats. Conversely, those without a downlink at time t are termed *unpaired*, and they are stored in the set N^-; line-2. Amongst the paired CubeSats, we then determine whether they saturate their downlink; line 4-8. Specifically, in line-5, we store the link (a, b) in the set L^- those CubeSats that

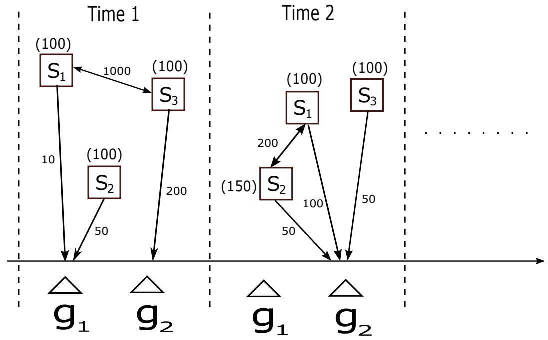

Figure 3. The download traffic scheduling problem. The number next to each link indicates its data rate (bits/s). The notation (x) indicates a CubeSat has x bits of data for download. At each t, each CubeSat i receives $d_i^t = 100$ bits of data. In this example, there is no energy constraint.

are unable to saturate their downlink and have a positive E_i^t value after accounting for the energy used to transfer D_i^t to their paired ground station. For these CubeSats, in line 10-17, we pair them with an unpaired CubeSat. The goal is to increase their available data for download; i.e., D_i^t. In line-11, for a CubeSat i in L^-, we first obtain all its unpaired neighbors. If there is at least one such CubeSat, then in line 13, we identify the CubeSat n that can transfer the *most* data to CubeSat i. The link (n, i) is then added into the solution set L, and line 16 removes the chosen CubeSat n from further consideration.

Algorithm 1: SelectLinks()

Input : $G^t(V^t, E^t)$
Output: Paired nodes in time t

/* Apply PTMD, RR, RP, or SCD */
1 $\mathcal{L}$ = SelectDownlinks(V^t, E^t)
/* The set of unpaired CubeSats */
2 $N^- = \{a \mid (i,j) \in \mathcal{L} \wedge a \neq i \wedge i, a \in \mathcal{S}^t\}$
3 $L^- = \emptyset$
/* Find unsaturated links */
4 **for** $(i,j) \in \mathcal{L}$ **do**
5   **if** $D_i^t < MIN(r_{ij}^t \tau, \frac{E_i^t}{p_{txDL}}) \wedge (E_i^t - D_i^t p_{txDL}) > 0$ **then**
6     $L^- \cup (i,j)$
7   **end**
8 **end**
/* Select ISLs */
9 $L = \emptyset$
10 **for** $(i,j) \in L^-$ **do**
11   $\Gamma = \{a \mid a \in \mathcal{N}^t(i) \wedge a \in N^-\}$
12   **if** $\Gamma \neq \emptyset$ **then**
13     $n = \arg\max_{a \in \Gamma} \text{MIN}(D_a^t, r_{ai}\tau, E_a^t p_{txISL})$
14     $L \cup (n, i)$
15     $\mathcal{N}^- \setminus n$
16   **end**
17 **end**
18 return$(\mathcal{L} \cup L)$

Figure 2 shows how each ground station in time t is paired with a Cubesat. Lines 1-3 construct the set of downlinks. It then iterates through each ground station and determines the *best* CubeSat to pair with. Here, *best* is determined using one of the following rules:

- **Pair the Most Data (PTMD)**: select the CubeSat in S with the largest D_i^t value.
- **Random Pair (RP)**: a random CubeSat from S is chosen.
- **Round Robin (RR)**: define I as a vector where the i-th element records the number of times CubeSat i has been paired with a ground station from time $t = 1$ to $t - 1$. The *best* CubeSat n is one with the smallest $I(n)$ value.
- **Shortest Contact Duration (SCD)**: choose a CubeSat that has the shortest contact duration with g.

After selecting the best Cubesat n, the downlink (n, g) is added to the solution set and CubeSat n is removed from consideration in subsequent iterations.

Algorithm 2: SelectDownlinks()

Input : V^t, E^t
Output: Paired s-g links

1 $\mathcal{S}^t$ = GetCubeSats(V^t)
2 $\mathcal{G}^t$ = GetGroundStations(V^t)
3 $\mathcal{L} = \{(i,j) \mid i \in \mathcal{S}^t \wedge j \in \mathcal{G}^t\}$
4 $L = \emptyset$
5 **for** $g \in \mathcal{G}^t$ **do**
6   $S = \{i \mid i \in \mathcal{N}^t(g) \cap \mathcal{S}^t\}$
7   $n = BestCubeSat(S)$
8   $L \cup (n, g)$
9   $\mathcal{S}^t \setminus n$
10 **end**
11 return(L)

6. Evaluation

The experiments are conducted in Matlab. The parameter values are based on the QB-50 space mission [10]; see Table 1. The contact time and duration of each CubeSat with a ground station and each other is generated randomly. That is, the start time and contact duration between CubeSats and ground stations are random. We conducted two sets of experiments comparing the case with and without ISLs. We plot the average of 50 runs.

6.1. No ISLs

We first consider the case without ISLs. Figure 4 shows the total downloaded data for each $|\mathcal{G}|$ value, with $|\mathcal{S}| = 10$ and $|\mathcal{G}|$ ranges from one to six. We see that increasing the number of ground stations will give CubeSats more opportunities to be paired with ground stations. Hence, more data can be downloaded. Referring to Figure 4, the

Symbol	Description	Values
$\|\mathcal{S}\|$	Number of CubeSats	10
$\|\mathcal{G}\|$	Number of ground stations	6
e_i^t	Energy arrivals	10 J/s
p_c	Power consumption of sub-systems	2 J/s
$\mathcal{B}$	Battery size	8000 J
D_{max}	Data storage size	8×10^9 bits
p_{txISL}	Transmit power over ISLs	0.0005 J/bit
p_{rxISL}	Receive power over ISLs	0.0005 J/bit
p_{txDL}	Transmit power over downlinks	0.001 J/bit
p_{rxDL}	Receive power over downlinks	0.001 J/bit
d_i^t	Data arrivals	1600 bps
r_{ISL}	Data rate of ISLs	12800 bps
r_{DL}	Data rate of downlinks	9600 bps
I_{sg}	Start time (secs) when CubeSat s sees ground station g	$[100, 300]$
Z_{sg}	Contact duration (secs) of CubeSat s with ground station g	$[1, 100]$
T	Experiment time period	350s

TABLE 1. SIMULATION PARAMETERS. THE SYMBOL s AND g CORRESPOND TO A CUBESAT AND A GROUND STATION, RESPECTIVELY.

performance of SCD is worse than PTMD; see Figure 5 and 6. This is expected because SCD may choose CubeSats that are unable to saturate their downlink. RR has similar performance to PTMD because Cubesats accumulate data whilst waiting their turn to be paired with a ground station. Consequently, when they are paired with a ground station, they have sufficient data to ensure a high download rate.

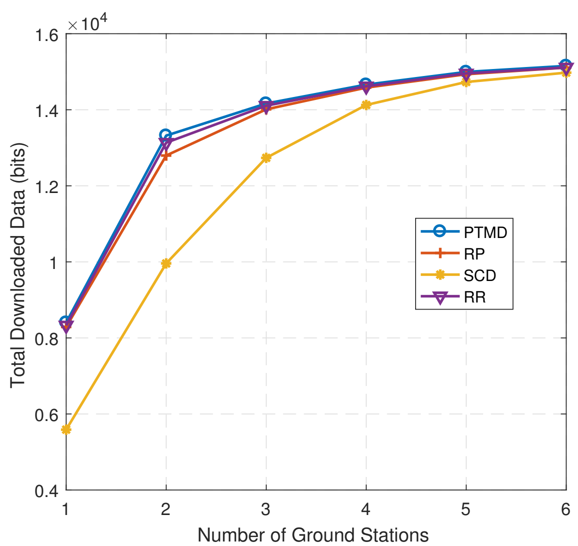

Figure 4. No ISLs case – a comparison of four downlink rules

Figure 5 and 6 show the average downlink utilization for PTMD and SCD, respectively; both are cumulative distribution function (CDF) plots of the percentage of links with at most a given average utilization. We remark that both RP and RR exhibit similar behaviors to PTMD and are thus omitted. Note that when there are more ground stations, there are also more downlinks. As we increase the number of ground stations, e.g., six ground stations (6G), we see that a high number of links (90%) have less than 20% link utilization when using PTMD. In contrast, when there is one ground station (1G), fewer links (less than 20% for PTMD and 40% for SCD) have utilization less than 90%. Comparing the two figures, for the 1G case, in PTMD, we can see 10% downlinks have an average utilization of 30%, but 90% downlinks have 100% average utilization. For SCD, 40% of the downlinks have an average utilization of 10%, with the remaining 60% downlinks fully saturate the downlink capacity. In contrast, when using PTMD, we find that more than 80% of the links have an average utilization that exceeds 90%. When using SCD, fewer downlinks have a high utilization. Referring to Figure 6, for 1G, 40% have average utilization less than 10%. This grows to 85% when there are six ground stations. The reasons is because CubeSats do not have enough time to transmit all their stored data; hence, downlinks cannot be fully used in each time slot. On the other hand, for PTMD, the average downlinks utilization is higher than SCD because PTMD picks downlinks with the highest data.

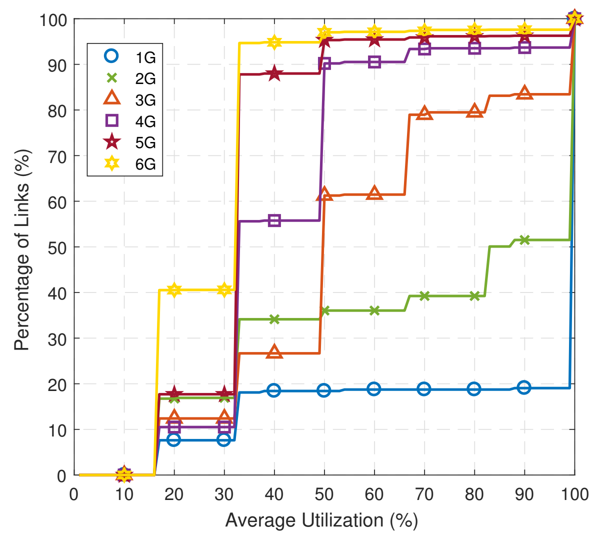

Figure 5. Downlink utilization – PTMD

6.2. With ISLs

Figure 7 illustrates the average ISLs utilization with varying number of ground stations. The number of CubeSats is set to ten. We can see that when there are fewer than three ground stations, SCD yields much higher utilization as compared to PTMD. However, when the number of ground stations is larger than five, the average ISLs utilization is approximately the same and concentrates around 10%. The reason is because CubeSats have many download opportunities. As a result, CubeSats have little spare data and energy to support data transmission over ISLs.

Figure 8 and 9 show ISLs utilization for PTMD and SCD. There are ten CubeSats and ten ground stations. When there is one ground station (1G), ISLs are used less

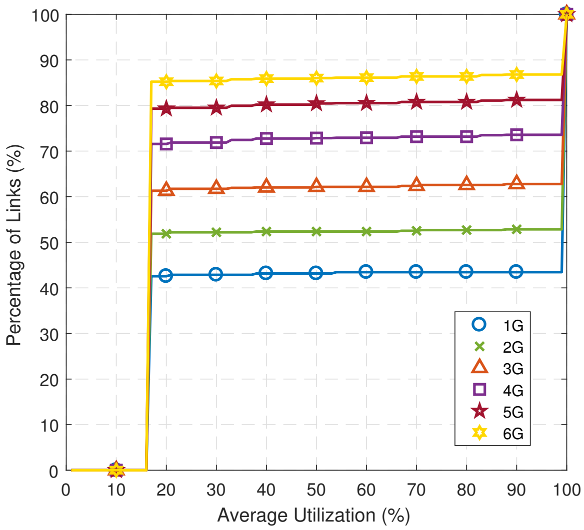

Figure 6. Downlink utilization – SCD

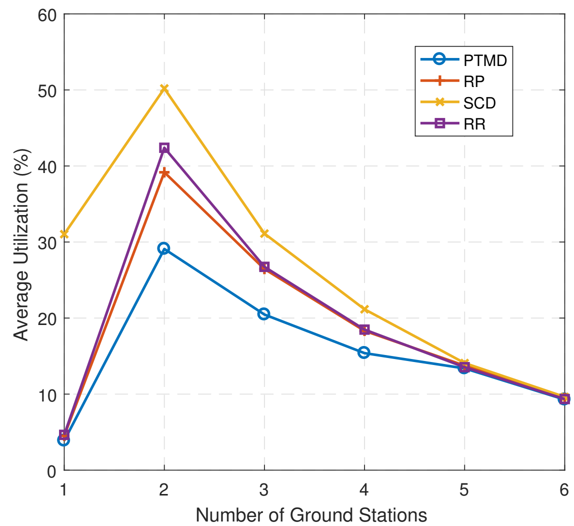

Figure 7. ISLs utilization for all four rules

frequently; referring to Figure 8, for PTMD, approximately 90% of the ISLs have an average utilization of around 10%. However, with more ground stations, e.g., 6G, only 30% of ISLs have utilization below 20%. More than 90% of ISLs have utilization higher than 20%. With increasing ground stations, CubeSats download more frequently, and hence, they need to transfer data over ISLs more frequently in order to experience a higher download rate. For example, for both PTMD and SCD, when there are six ground stations (6G), only 30% of ISLs have utilization less than 20%. The majority of ISls have higher than 20% utilization. SCD has a higher ISLs usage because a CubeSat with a short contact duration may not have sufficient data for its downlink. Consequently, it will need to transfer data from another CubeSat over an ISL. Moreover, when using SCD, CubeSats have more buffered data and unused energy due to the lack of transmission opportunities with ground stations, which lead to higher ISLs utilization on average.

Table 2 shows an experiment with ten CubeSats and two ground stations. We record the total endogenous data, amount of data transmitted over ISLs, and the total data that arrives at ground stations. We see that the total downloaded data is approximately the same. CubeSats that use PTMD rely less on ISLs because they have more buffered data for download. For SCD, data transmitted over ISLs almost doubles. As mentioned earlier, CubeSats using SCD may have insufficient data when chosen to download to a ground station. Moreover, they tend to have a high energy and untransmitted data. The performance of PTMD and RR is different. For example, when using RR, before CubeSats are paired with a ground station, they have already transmitted data to their neighbors. Therefore, these CubeSats have less data when it is their turn to transmit to a ground station. Consequently, they will rely more on ISLs in order to ensure high download rate.

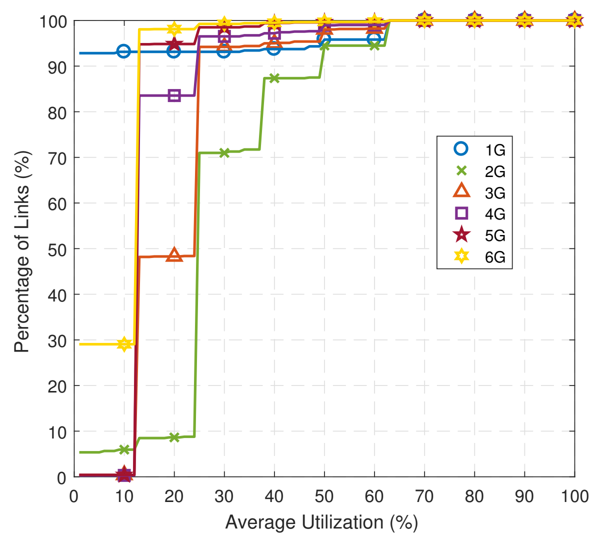

Figure 8. ISLs utilization – PTMD

Figure 10 shows the total downloaded data for varying $|\mathcal{G}|$ values. There are ten CubeSats. Observe that due to the use of ISLs, all downlink rules managed to achieve a high total download. In contrast, see Figure 4, without ISLs, there is gap between all four rules when CubeSats do not exploit ISLs. On the other hand, with ISLs, if a CubeSat has insufficient stored data to saturate its downlink, then it can transfer more data from a neighboring CubeSat.

Lastly, we compare the difference between the total downloaded data with and without ISLs. Here, we fix $|\mathcal{S}|$ to ten. From Figure 11, after adding ISLs, SCD has the biggest difference. Due to limited contact duration, CubeSats using SCD have more buffered data and unused energy as compared to the other three rules. Consequently,

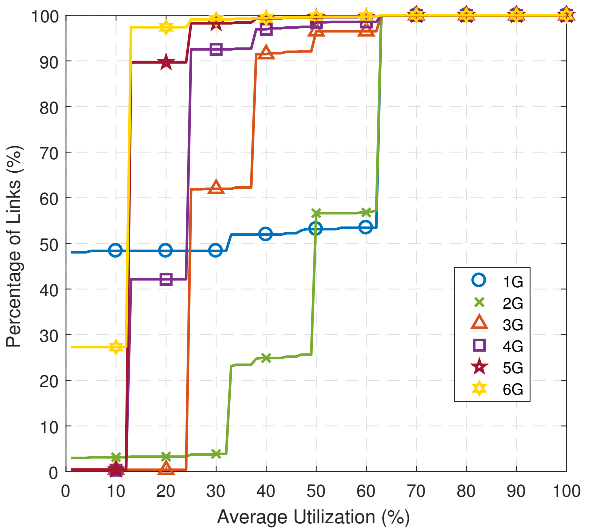

Figure 9. ISLs utilization – SCD

Pairing Rules	Total Downloaded Data (bits)	Data over ISLs (bits)	Endogenous Data (bits)
PTMD	16245	6730	9514
RP	16231	9290	6941
SCD	16231	12650	3581
RR	16245	9527	6718

TABLE 2. EXPERIMENT RESULT WITH TEN CUBESATS AND TWO GROUND DEVICES

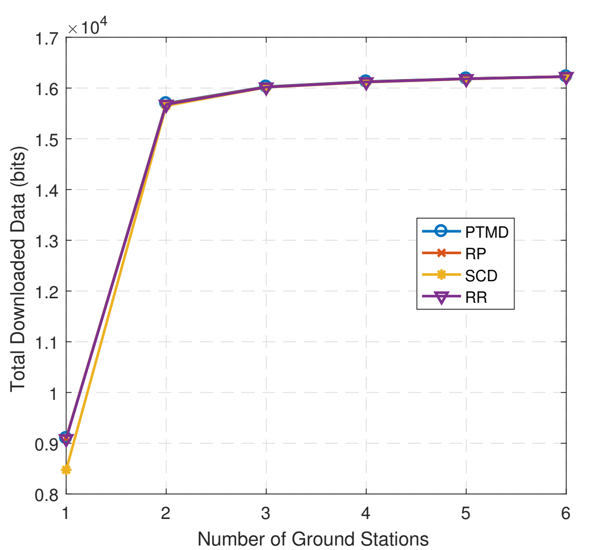

Figure 10. Total downloaded data for all four rules when ISLs are available.

CubeSats are able to download more data over ISLs to saturate their downlinks. Moreover, the difference in the total downloaded data of all four rules increase first before decreasing. When there is one ground station, CubeSats have limited opportunities to download to the ground station. In other words, the one ground station is the bottleneck. With more ground stations, CubeSats are afforded more download opportunities. This means they have less data to download with increasing number of ground stations. In fact, the maximum total downloaded data is attained with two or three ground stations; see Figure 10. Consequently, when there are many ground stations, regardless of the pairing rules used, CubeSats have less data to download. Nevertheless, the total downloaded data is still better than the case without ISLs because at the minimum a CubeSat can forward its own data as well as that of a neighbor in each pairing to a ground station.

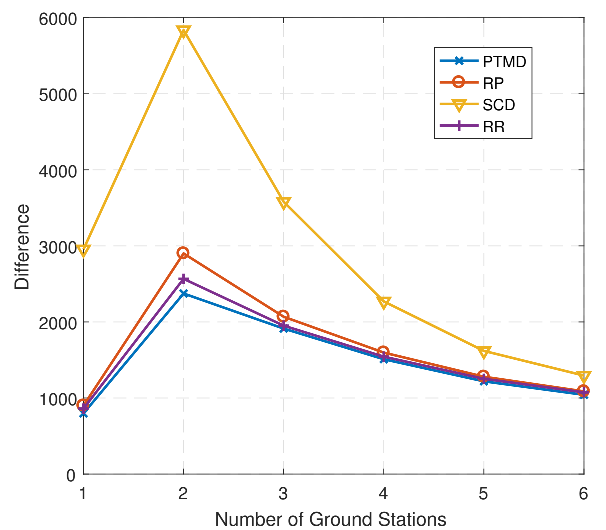

Figure 11. Difference in total downloaded data with and without ISLs

7. Conclusion

This paper has studied data download from a CubeSats swarm. In particular, it quantifies the benefits of ISLs. We find that using ISLs is advantageous because CubeSats with surplus data can flow to those with less data. Consequently, CubeSats are more likely to saturate downlink capacity. Moreover, all tested rules used to pair a CubeSat with a ground station yield the same performance. This indicates than when ISLs exist, it is unnecessary to develop the *best* rule for pairing a CubeSat with a ground station.

A future work will be to model the problem mathematically as a MILP. Another interesting work is to model a CubeSat swarm as a time-varying graph and determine the minimum time required to download all data.

References

[1] P. Muri and J. McNair, "A survey of communication sub-systems for inter-satellite linked systems and cubesat missions," *IEEE Wireless Communications*, vol. 7, no. 4, pp. 290–305, Apr. 2012.

[2] R.Radhika, W. Edmonson, F. Afghah, R. Rodriguez-Osorio, F. Pinto, and S. Burleigh, "Survey of inter-satellite communication for small satellite systems: Physical layer to network layer view," *IEEE Comms Surveys & Tutorials*, vol. 18, no. 4, pp. 2442–2473, 2016.

[3] W. A. Shiroma, A. T. Ohta, and M. A. Tamamoto, "The University of Hawaii CubeSat: a multidisciplinary undergraduate engineering project," in *23rd Ann. Frontiers in Edu.*, Westminster, CO, USA, Feb. 2003.

[4] K. Woellert, P. Ehrenfreund, A. J. Ricco, and H. Hertzfeld, "Cubesats: Cost effective science and technology platforms for emerging and developing nations," *Elsevier Advances in Space Research*, vol. 47, no. 4, pp. 663–684, Feb. 2011.

[5] C. K. Pang, A. Kumar, C. H. Goh, and C. V. Le, "Nano-satellite swarm for SAR applications: Design and robust scheduling," *IEEE Transactions on Aerospace and Electronic System*, vol. 51, no. 2, pp. 853–865, Apr. 2015.

[6] E. Gibney, "Cubesats queue up for deep-space rides," *Nature*, vol. 535, no. 7, pp. 19–20, Jul. 2016.

[7] "JPL selects Europa cubesat proposals for study," https://www.jpl.nasa.gov/news/news.php?feature=4330, accessed: 2017-08-17.

[8] "Mars cube one (MarCO)," https://www.jpl.nasa.gov/cubesat/missions/marco.php, accessed: 2017-08-18.

[9] S. Spangelo, J. Cutler, K. Gilson, and A. Cohn, "Optimization-based scheduling for the single-satellite, multi-ground station communication problem," *Elsevier Computers and Operations Research*, vol. 57, no. 5, pp. 1–16, 2015.

[10] B. Lemay, J. Castaing, R. A. E. Zidek, A. Cohn, and J. Cutler, "An optimization-based approach for small satellite download scheduling with real-world applications," 2017, submitted to AIAAA Journal.

[11] T. Lv, W. Liu, H. Huang, and X. Jia, "Optimal data downloading by using inter-satellite offloading in leo satellite networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[12] X. Jia, T. Lv, F. He, and H. Huang, "Collaborative data downloading by using inter-satellite links in leo satellite networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1523–1532, March 2017.

[13] S. C. Spangelo, J. W. Cutler, A. T. Klesh, and D. R. Boone, "Models and tools to evaluate space communication network capacity," *IEEE Transactions on Aerospaace and Electronic Systems*, vol. 48, no. 3, pp. 2387–2404, Jul. 2016.

[14] Y. An, J. Li, W. F. B. Wang, Q. Guo, jin Li, X. Li, and X. Du, "EESE: Energy efficient communication between satellite swarms and earth stations," in *IEEE 16th Intl. Conf. on Adv. Communication Technology*, South Korea, Feb. 2014, pp. 845–850.

[15] J. Chen, L. Liu, and X. Hu, "Towards throughput-optimal routing algorithm for data collection on satellite networks," *Journal of Distributed Sensor Networks*, vol. 12, no. 7, pp. 1–7, Jul. 2016.

[16] A. Budianu, A. Meijerink, and M. J. Bentum, "Swarm-to-earth communication in OLFAR," *Elsevier Acta Astronautica*, vol. 107, pp. 14–19, Feb. 2015.

[17] P. Wang, G. Reinelt, P. Gao, and Y. Tan, "A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation," *Elsevier Computers and Industrial Engineering*, vol. 61, no. 2, pp. 322–335, Feb. 2011.

[18] B. Clement and M. D. Johnston, "The deep space network scheduling problem," in *Procs. of the National Conference on AI*, Pittsburgh, Pennsylvania, Jul. 2005.

[19] O. Challa and J. McNair, "Cubesat torrent: Torrent like distributed communications for cubesat satellite clusters," in *IEEE Milcom*, Orlando, FL, USA, Oct. 2012.

[20] W. Usaha and J. A. Barria, "Reinforcement learning for resource allocation in LEO satellite networks," *IEEE Trans. on Sys., Man and Cybernetics, Part B: Cybernetics*, vol. 37, no. 3, pp. 515–527, Jun. 2007.

[21] J. Sun and E. Modiano, "Routing strategies for maximizing throughput in LEO satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 2, pp. 273–283, Feb. 2004.

[22] H. Bedon, C. Negron, J. Llantoy, C. M. Nieto, and C. Asma, "Preliminary internetworking simulation of the QB50 cubesat constellation," in *IEEE LATINCOM*, Bogota, Columbia, Nov. 2010.