A Distributed and Hybrid Ground Station Network for Low Earth Orbit Satellites

Deepak Vasisht, Ranveer Chandra
Microsoft
{deepak.vasisht,ranveer}@microsoft.com

ABSTRACT

Low Earth Orbit satellites for Earth observation have become very popular in recent years due to their ability to take high-resolution images of the Earth at high revisit rates. These satellites collect hundreds of Gigabytes of imagery during their orbit. This data needs to be downloaded using ground stations on Earth. However, due to the low altitudes, the satellites move fast with respect to a ground station on Earth, and consequently, have a few minutes time window to download the data to a single station. We propose a geographically distributed ground station design, DGS that improves robustness and reduces downlink latency. DGS is the first system to use a hybrid ground station model, where only a subset of ground stations are uplink-capable. This paper evaluates the feasibility of this design using simulations and empirical measurements.

CCS CONCEPTS

• **Networks** → **Network architectures**; *Network design principles*; • **Computer systems organization** → *Dependable and fault-tolerant systems and networks.*

KEYWORDS

Satellite Networking, Earth Observation, Ground Station Architecture, Distributed Ground Station

ACM Reference Format:
Deepak Vasisht, Ranveer Chandra. 2020. A Distributed and Hybrid Ground Station Network for Low Earth Orbit Satellites. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks (HotNets '20), November 4–6, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3422604.3425926

1 INTRODUCTION

In recent years, we have seen a meteoric rise of interest in Low Earth Orbit (LEO) satellites. Multiple companies [11, 12, 17, 34] have committed to deploy constellations of hundreds of cubesats (small-sized satellites) in low Earth orbits. Today, around 75% of satellites in orbit around the Earth are LEO satellites [34]. The reduced cost of satellite component manufacturing and the lower cost of launching them through ride-sharing arrangements has precipitated the increase in low-volume LEO satellites. These satellites broadly serve two purposes: communication and Earth observation. Communication satellites (for example, StarLink by SpaceX) enable connectivity either at lower latency than traditional Internet access methods, or serve underserved regions like remote rural areas or IoT sensors deployed in the wild.

Earth observation satellites provide high-resolution images of Earth in various parts of the spectrum (visible, infra-red, radio waves, etc.) at high revisit rate. The low orbit enables high-resolution imagery, the density of satellites enables high revisit rates, and the miniaturization of imaging hardware enables multi-spectral or hyper-spectral imagery. Around 45% of the LEO satellites in orbit today are classified as Earth observation satellites [34]. These satellites typically orbit at an altitude of 300 to 600 kilometers in polar orbits, take high-resolution images, and transmit them back to ground stations. They collect hundreds of Gigabytes of data in a single pass over the Earth and need to transmit this data back [29].

Today, every satellite operator deploys few highly specialized (multi-million US dollars) ground stations [22] to download data from these satellites. Due to the low orbit, the satellites move fast with respect to an observer on Earth and can only communicate with a ground station for less than ten minutes in a single pass [9, 10]. This necessitates the requirement of high-end high-fidelity ground stations that can download large quantities of data in a short period.

This design for ground stations suffers from multiple bottlenecks. First, the ground stations are under-utilized when the constellation size is small. As the constellation size grows to hundreds, the system suffers from contention since multiple satellites become visible at the same time to the ground station. Second, the centralized link is a single point of failure. At high frequencies used by the ground stations (8 GHz and above), the links are prone to weather related variation (attenuation of 10-25 dB due to rain and clouds [3]). Some LEO satellites have reported up to 88% packet loss [8]. Third, this design adds latency to the downlink. Any data collected in orbit can only be downloaded when the satellite comes in contact with the ground station. This system adds latency from several hours to a few days for the data to be accessible. This latency is crucial for time-sensitive applications of satellite data like flood modeling and forest fires. Finally, the cost of licensing and setting up a ground station is prohibitive for new entrants (like academic research satellites). Given the reduced costs of satellites (few tens of thousands USD), the ground station becomes the bottleneck.

In this paper, we present DGS, a distributed ground station design for LEO satellites. Instead of a few high-fidelity ground stations per constellation, DGS uses a large set of geographically distributed low-complexity ground stations that are managed by individual contributors. A geographically distributed design has many advantages. Our design can automatically scale to varying demands.

HotNets '20, November 4–6, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8145-1/20/11…$15.00
https://doi.org/10.1145/3422604.3425926

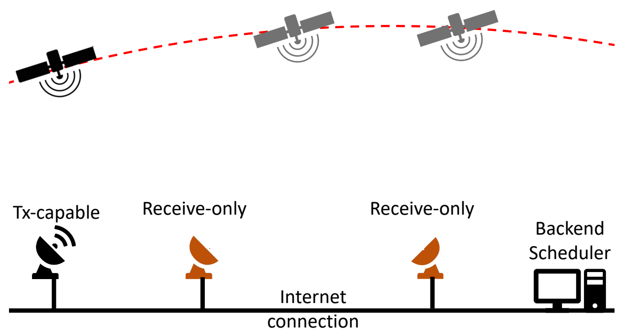


Figure 1: DGS is a geographically distributed ground station design. It uses a mix of transmit-capable and receive-only ground stations, to enable low latency, high fidelity data transfer from LEO satellites.

Allowing the ground station to be distributed also allows DGS to relax the requirement of high throughput on individual links. High throughput can be achieved by leveraging the geographical diversity as opposed to a single link. Moreover, DGS is more robust to weather variations. Downlink can be dynamically scheduled so that cloudy weather in one part of the world is offset by clear weather in the other. Finally, since a satellite is likely to encounter many more ground stations, it can offload latency-sensitive data faster.

Furthermore, we make the observation that the primary data mode for Earth observation satellites is downlink; the uplink is infrequently used for control traffic alone. In fact, ground stations today support Gbps downlink but only hundreds of Kbps uplink [9, 10, 22]. In DGS, we build on this observation to limit a majority of the ground stations to be receive-only. A very small number of DGS ground stations are capable of uplink, thereby significantly reducing the licensing overhead required to establish a ground stations. The uplink-capable ground stations communicate with the satellites and upload a plan for the data-dump as the satellite orbits around the Earth. The satellite then dumps the data at the locations pre-specified by the uploaded plan. We believe this design choice is essential for ease of setting up ground station downlink-only links by individual operators.

Enabling such a design, however, needs us to solve the following challenges:

- **Adaptive Downlink Scheduling:** We present a new scheduling framework that schedules satellite-ground station links by accounting for orbits, link quality, and weather conditions. Our scheduler allows for objective functions that can optimize for throughput, mean latency, peak latency, etc.
- **Rate Selection:** The satellite downlink rate depends on the channel conditions at a given location. For instance, rain can attenuate the downlink signal by 10 to 20 dB in X, Ku, and Ka bands used for satellite downlink [3]. In typical systems, the optimal rate is selected using feedback from the receiver. In our receive-only design, we need to predict the link quality and optimal rate. We build a link quality estimator that can use weather conditions in conjunction with satellite's expected orbit to compute the expected channel quality.
- **Ack-free Downlink:** Given that the receive-only ground stations cannot send immediate acks to the satellite, how do the satellites know if the downlink was successful? DGS relays the acks from the receive-only ground stations to transmit-capable ground stations through Internet links, collates them, and uploads these delayed acks to the satellite through the transmit-capable stations when the satellite is in contact with the transmit-capable station.

We evaluate DGS using a combination of simulations and real-world data from deployed open-source SatNOGS ground stations [25]. In SatNOGS, amateur radio enthusiasts manage ground stations to observe data from LEO satellites. We use this data to model individual links and geographical distribution. We model 259 satellites and 173 ground stations in the SatNOGS database. A summary of our results is:

- **Latency:** DGS reduces the mean latency of data download from 58 minutes to 12 minutes and the 90-th percentile from 293 minutes to 44 minutes as compared to a baseline method that uses high-end ground stations with 10X more link capacity than DGS's ground stations.
- **Data Transfer and Backlog:** In our experiment, we download over 250 TB of data in DGS from 259 satellites. In an experiment with each satellite collecting 100 GB per day, DGS reduces the median backlog (data not delivered) for a satellite from 8.5 GB to 1.9 GB (99-th percentile: 80.7 GB to 16.7 GB).
- **Adaptability:** We demonstrate that DGS's scheduling can adapt to different value functions. A throughput-optimized system doubles the 90-th percentile latency as compared to a latency-optimized system.

As the LEO satellite deployments increase, a distributed framework is essential to enable a scalable, performant, and robust ground station design. This work is inspired from the past shifts in computing from singular highly specialized hardware to distributed low-complexity components. In DGS, we present an initial set of tools to support this new paradigm.

2 BACKGROUND AND RELATED WORK

LEO satellites are primarily used for communication and Earth observation. The satellites orbit the Earth and communicate with ground stations. For Earth observation satellites, the imaging equipment (may incorporate sensors in infrared, ultraviolet, microwave, and other parts of the spectrum) captures high-resolution images of Earth frequently and must send the images back to Earth. The uplink comprises primarily of TT&C (tracking, telemetry, and control) data and is fairly low bandwidth. Consequently, the typical design of Earth observation satellites uses a narrowband uplink (tens to hundreds of kbps) and a high bandwidth downlink (hundreds of Mbps to tens of Gbps) [9, 10, 22]. This design choice also reflects in the choice of spectrum. Public documents [14, 15, 30] show that the uplink uses the lower frequency and lower-bandwidth S-band (2025-2110 MHz) while the downlink uses the higher frequency X-band (8025-8400 MHz). Some designs are also exploring higher frequencies (Ku band – 12 to 18 GHz and Ka band – 26.5 to 40 GHz) for downlink [10].

Today, most satellite operators deploy their own ground stations, preferably close to the Earth's poles [9, 10]. Since Earth observation LEO satellites typically operate in polar orbits, deploying ground stations around the pole increases the time of contact between the satellite and the ground station during each pass. The best known ground station design can achieve a data rate around 1.6 Gbps by combining six frequency-polarization channels at the best satellite-ground station link [10]. A typical contact (a pass) between the satellite and the ground station lasts for seven to ten minutes. The 1.6 Gbps link in [10] can download data upto 80 GB in a single pass. Note that, the max rate can only be sustained when the satellite is at the shortest path. As the satellite reaches closer to the horizon, the link quality degrades and the satellite has to downgrade its rate. Each satellite can do two-to-three passes per ground station per day, but the passes have varying quality. In some passes, the satellites pass closer to horizon, hence the link quality degrades. The typical amount of data that needs to be downlinked to image the Earth everyday can go up to few Terabytes per day [4]. Multiple satellites need to collaborate to make this happen.

The downlink becomes less reliable and constrained as constellation size grows. This problem of constrained downlink on LEO satellites has been tackled previously in both academia and industry. [8] proposes offloading some computation to the satellites to reduce the downlink load. For instance, in a workload that needs images of buildings, the satellites could pre-filter building images before downlink to the ground stations. However, this design runs contrary to the business model for Earth observations satellites that sell the observed data to customers, who then run the end application. In the absence of *a priori* knowledge of the end application, the pre-filtering on the satellite might reject important information relevant to the user. In contrast, DGS downlinks all the data to the ground using a hybrid ground station design.

In the industry, multiple efforts [1, 23, 24] have emerged recently to rent out time on individual ground stations to satellite operators by the minute. This is a welcome trend in enabling access to new satellite operators. This investment opens up the possibility of new abstractions like distributed ground station architectures in the future. In DGS, we investigate the tools that will be required for such a distributed design. VERGE [26] is perhaps the closest design to DGS. In [26], Lockheed Martin is planning to deploy low cost S-band parabolic antennas in a geographically distributed manner. Each antenna will stream raw RF measurements to the cloud, where a software defined receiver will decode this data. In contrast, DGS co-locates compute alongside the antenna and the decoded & processed data is sent to the cloud. This significantly reduces the backhaul capacity required to support the ground station (by orders of magnitude). Furthermore, it allows for edge compute workloads that can prioritize data upload to the cloud in an efficient manner. One direct impact of this design choice is that [26] is limited to lower bandwidth S-band downloads, as opposed to X-band downloads that are common for earth observation.

Finally, DGS is inspired by past work in open source ground station designs [5, 25] and deployments of these stations [6, 25, 28, 31]. These deployments have fostered research in scheduling, mission control, and other aspects of ground station design [2, 32, 33, 35]. Most of these designs are limited to low frequency, low data rate regimes for experimental satellites that transmit small amounts of data. In DGS, we differ along three axes: distributed design framework, high frequency and high bandwidth data downloads, mix of transmit-capable and receive-only ground stations.

3 DISTRIBUTED GROUND STATIONS

We propose a distributed architecture for ground station networks: DGS. An overview of DGS is in Fig. 1. DGS consists of multiple ground stations spread across the globe. Each of these ground stations is connected to the Internet. DGS ground stations have three distinctive characteristics:

- **Geographically Distributed:** DGS ground stations are spread across the globe, either maintained by independent individuals, volunteers, or corporations. This geographic distribution of data centers has two advantages. First, it enables satellites to follow a dynamic downlink schedule. If the link from satellite α to ground station i is expected to encounter clouds, then it could downlink data at a different ground station j that falls along its path. Second, the geographic distribution reduces latency in the data downlink process. This allows the download plan to be cognizant of the latency-sensitivity of the data. For instance, in latency sensitive applications like forest fires and floods, the sensitive data can be downlinked in tens of minutes in a geographically distributed network, but will take hours to days in a centralized architecture.
- **Hybrid:** As noted in Sec. 2, the data communication for Earth Observation satellites is primarily downlink. Moreover, enabling uplink on a ground station requires following complex licensing requirements [22] that are both expensive and time-consuming. In DGS, we allow for a majority of the nodes to be receive-only, i.e. they do not transmit any data. This is an important design choice for making the system scalable. At the same time, this design choice opens up a lot of interesting systems problems that we discuss below.
- **Low-complexity:** For the system to be deployed at scale, we expect the average cost and complexity of individual ground stations to be low. As such, the individual ground stations in DGS do not (necessarily) have high gain, specialized equipment. This reduces the SNR of individual links, but this is compensated through geographic diversity.

Overview: In DGS, a scheduler estimates the trajectory of a satellite for a fixed future time-interval. Then, it estimates the link quality between all satellite-ground station pairs using the link quality estimation method in Sec. 3.2. It, then, identifies an optimal match between satellites and ground stations at each time instant (Sec. 3.1). This schedule is distributed to all the ground stations over the Internet. The downlink schedule for each satellite is also uploaded to individual satellites when they come in contact with a transmit-capable ground station. Then, during their path, the satellites follow the planned schedule and downlink data to receive-only ground stations, which follow the shared schedule as well and point to the corresponding satellite. This data is then collated at the back-end and any missing pieces can be communicated to the satellite during next contact with transmit-capable stations.

The design of DGS poses several technical questions. At the core, DGS needs to route traffic from M satellites to N ground stations with time-varying (and weather dependent) links between the satellites and ground stations. This is complicated by the receive-only capability of a majority of the ground stations in the network. Below, we present our solutions to a subset of these challenges.

3.1 Downlink Scheduling

In this section, we formalize the problem of scheduling the satellite-ground station downlink and provide a mechanism to identify the right downlink schedule for DGS.

Before we delve deeper into DGS's downlink scheduler, we note that scheduling downlink for a distributed hybrid architecture like DGS is fundamentally different from scheduling for a centralized architecture with a small number of ground stations. In centralized architectures, it is rare for multiple satellites to compete for a ground station's time. This is because each pass (or satellite-ground station contact) lasts just ten minutes. Even when the conflict arises, it is typically handled by using multiple co-located antennas that can point to different satellites [4, 9, 10]. As constellations grow, this conflict becomes more likely. Similarly, as the density of ground stations increases, each satellite has multiple feasible downlink paths, which do not exist in a sparse centralized design. Finally, the existence of multiple feasible links allows DGS to incorporate and adapt to the time-variation in link quality, either due to elevation or weather.

Let us assume we have a set of satellites, $\mathcal{S} = \{s_1, s_2, ..., s_M\}$ and a set of ground stations, $\mathcal{G} = \{g_1, g_2, ..., g_N\}$. Each satellite, s_i is represented by its TLE (Two Line Element set) [18]. TLE is a standard representation for satellite orbits that contains a satellite identifier as well as orbit parameters. For LEO satellites, satellite location prediction using TLEs is accurate to within a kilometer if done a few days in advance. Note that the TLEs are time-varying and are updated over time. Similarly, each ground station, $g_j \in \mathcal{G}$, is represented by its latitude, longitude, ownership information, and data downlink constraints. The downlink constraints are represented as a M-bit bitmap, where bit i is 1 if data downlink from s_i is allowed. The downlink constraints ensure that ground station owners can maintain control over their resources (e.g. a ground station owner might want satellite operators to pay a subscription fee) or to maintain regulatory restrictions (e.g. some countries may not want to downlink data from satellites operated by their competitors).

Each satellite, s_i has a sequence of data bits X_i that it intends to send to ground stations. We also assume the existence of a value function, Φ such that for any subset $x \subset X_i$ and time t elapsed since the capture of the data, $\Phi(x, t)$ denotes the value of transmitting that data to Earth. This value function is generic enough to capture different objectives. For instance, if the system aims to minimize time between data capture and data transfer, $\Phi(x, t) = t$, or to minimize throughput, $\Phi(x, t) = |x|$ (|x| denotes the number of bits in x). Similarly, $\Phi(x, t)$ can be defined by the satellite operators to prioritize data based on geography, e.g. to honor service level agreements (SLAs) with customers. From a ground station perspective, the value function can be assigned by bidding for priority access. Given this problem definition, DGS's scheduler performs the following steps:

Orbit Calculations: First, DGS obtains the most recent TLE data for each satellite and uses it to compute the orbit of the satellite over time. At each instance of time, we compute if a satellite is above the horizon for each ground station. If a satellite is above the horizon, DGS computes the distance, the elevation, and the azimuth angle of the satellite with respect to the ground station.

Graph Construction: DGS, then, uses the parameters estimated above to compute a weighted graph with the ground stations, $\mathcal{G}$, and satellites, $\mathcal{S}$ as nodes. There is an edge between a ground station and a satellite if the satellite is above the horizon (elevation is greater than zero) for a ground station and this link adheres to constraints defined for the ground station. This graph constantly evolves with time. We determine the weight of each edge by estimating the value of using that link. At each time instant, we estimate the link quality (and corresponding data rate) using the method described in Sec. 3.2. Then, we compute the value corresponding to the data that the satellite can send on that link using Φ.

Matching: Finally, at each time instant, we need to match satellites to ground stations. Multiple ground stations can downlink data from a single satellite. Similarly, multiple satellites are visible to a single ground station at any instance of time. However, most current ground stations can only support point to point links. Therefore, we need to pick a subset of the edges such that we optimize the transmissions given the capacity constraints defined in our graph.

Note that the weighted graph defined above is a bipartite graph. Thus, we can define our problem as a bipartite matching problem. Our objective is to find a valid matching in the graph that we constructed above. We can use two approaches to solve this problem: (a) identify a stable matching, and (b) identify an optimal matching. An optimal matching optimizes the value achieved by the entire system, i.e. it downloads the most valuable data across all satellites and ground stations. However, we envision the DGS framework to be fragmented. Consequentially, an optimal matching leaves space for a satellite-ground station pair to achieve sub-optimal results for itself. Therefore, we choose to implement stable matching algorithm in our current implementation of DGS. A stable match ensures that if any satellite-ground pair breaks their assigned link and forms a link of their own, at least one of them will derive less value from the new link than what they previously obtained. We use the Gale-Shapley algorithm [16] to solve the stable matching problem in the bipartite graph. It converges in $O(K^2)$, where $K = max(N, M)$.

Finally, we run the stable matching algorithm at each time instance to capture the temporal variation of the links. We do not optimize for links across time. This optimization can further benefit DGS but we leave this to future work.

3.2 Link Quality Model

A key factor in the optimization process is a prediction of the link capacity between a satellite and a ground station. Typically, link quality estimation can be done at runtime using feedback from the ground station to the satellite. However, in DGS many ground stations cannot transmit feedback. Furthermore, in order to pre-select the right communication pairs, this link quality needs to be estimated before the communication begins.

The link quality between a satellite ground station pair depends on three factors: (a) the distance between the satellite and the ground station, (b) the weather conditions – rain and clouds can significantly attenuate the signal, (c) the hardware used by the ground station and the satellite. In free space, the loss, L experienced by the signal when it traverses distance, d, is given by:

$$L = \left(\frac{4\pi d f}{c}\right)^2 \quad (1)$$

The path loss, L, increases with distance (d) and signal frequency (f). c is the speed of light.

Earth's atmosphere causes additional attenuation. A portion of it is static for a given location, but another part of it is time-varying depending on weather conditions. In particular, at frequencies over few GHz, the time varying component can be significant (>10 dB at 10 GHz). The effect also depends on the distance the signal covers below the clouds (i.e. in the rain). We use weather forecasts for a region, in combination with well-studied models developed by the International Telecommunication Union [19–21], to predict this component of the loss. Finally, the hardware dependent loss is static for a satellite-ground station pair and can be calibrated for.

DGS uses the above method to compute the net SNR of the received signal and converts that into the corresponding expected data rate. Specifically, we leverage the specifications of the DVB-S2 protocol [13, 27] used for downlink in Earth observation satellites [9, 10] for our analysis. This allows us to identify how much data can be downlinked at any time instant between a satellite-ground station pair. The satellite maintains a priority queue and sends the data in the highest priority first order. The value (estimated using the value function Φ) of the data to be sent using this link serves as the weight of the corresponding satellite-ground station edge in the bipartite graph defined above.

3.3 Discussion and Open Questions

Ack-free Downlink: Since a subset of DGS ground stations are receive only, a satellite can discard data only when it has interacted with a transmit-capable ground station and received an acknowledgement for the downlinked data. This implies that DGS does not necessarily reduce a satellite's storage requirement. Today, satellites have to store data for an entire orbit anyway, so DGS does not increase this requirement either.

Edge compute on the ground station: Past proposals [8] have explored edge compute on the satellite to pre-filter downlinked data. Edge compute on the satellite requires hardware upgrades and is not agnostic to the underlying application. We believe DGS provides a new avenue for this line of work by enabling edge compute on the ground station. Ground stations can leverage edge compute techniques to deliver latency-sensitive data to the cloud faster and upload the other data at a lower priority.

Beamforming: We assume that every ground station can connect to only one satellite at each point of time. Some modern designs of ground stations have explored beamforming at the ground station. This will be an interesting addition to DGS by enabling each ground station to split power between multiple satellites, thereby increasing the data downlink efficiency. We leave the exploration of this new optimization problem to future work.

Backward Compatibility: DGS's design is compatible with the DVB-S2 protocol used for data downlink. At this time, we cannot comment on compatibility with the software deployed on satellites in orbit due to lack of public documentation.

Figure 2: DGS Setup: Ground stations (red dots) used for evaluating DGS.

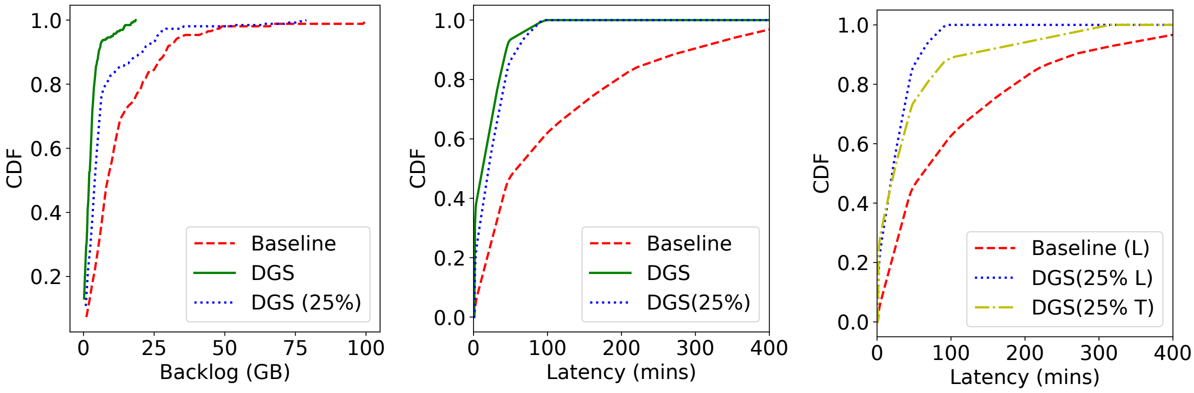
Economic and Security Implications: DGS's adoption hinges on appropriate economic incentives for operators to collect data and a security framework to prevent data misuse. This is an exciting direction to explore for future research.

4 EXPERIMENTAL EVALUATION

Setup: We evaluate DGS using data collected from deployments of the open-source SatNOGS ground stations [25]. SatNOGS is deployed by amateur radio enthusiasts using software defined radios. The ground stations listen to satellite broadcast signals e.g. from NOAA weather satellites. The observation data is logged in a public database. We download the data from all ground station-satellite links for a month long period. Note that, SatNOGS ground stations do not include any of the routing mechanism, link quality prediction, or other algorithms proposed in DGS. We use this data to validate our algorithms through a combination of simulation and real data. We select the ground stations that are operational and have made at least 1k observations. In the filtered dataset, we have 173 ground stations (Fig. 2) and 259 satellites.

For each ground station location and each time instance, we get the weather data using the Dark Sky weather API [7]. A majority of the ground stations operate in the sub-500 MHz frequency bands, and some (approx. 20%) support the L-band (1.5 to 1.75 GHz). Since Earth Observation satellites use the X-band (>8 GHz) [10, 22] to download their data, we cannot use the data from the SatNOGS database to report the SNR for satellite-ground station links. We use the SatNOGS measurements to validate other aspects of our design like orbit calculation, observation times, satellite-ground station link duration, etc. For SNR estimation, we use data rate estimates obtained using Sec. 3.2. We do not validate the high-frequency SNRs with hardware measurements – we leave this to future work. We validate the link quality model for lower frequencies using the SatNOGS measurements.

For simulating data transfer, each satellite generates 100 GB of data per day. Each satellite has the state-of-the-art radio described in [10]. However, since our ground stations are low-complexity, we do not use large dishes (5 m or more) typically used by commercial ground stations [22, 30]. We simulate our ground stations to have small, 1m diameter, dish antennas. This reduces the SNR of each station by 6 dB. Furthermore, our ground stations use a single-channel receiver.



(a) Data Backlog **(b) Latency** **(c) Value Function**

Figure 3: DGS outperforms the baseline in terms of (a) data delivered and (b) latency. (c) Changing the value function from latency (L) to throughput (T) increases the latency.

Baseline: We use the state-of-the-art ground station described in [10] as our baseline. This method uses 6 parallel channels as well as high-end receivers with 4m diameter dish antennas. As in [10], we model 5 such high-end ground stations across the planet. Each baseline ground station achieves 10x the median throughput achieved by a DGS node.

Data Transfer: First, we compare the ability to downlink data from 259 satellites for the baseline high-fidelity ground stations and DGS. We compare two variants of DGS: DGS and DGS(25%). DGS uses all of 173 ground stations in the network to download data. In DGS(25%), we reduce the number of stations to 25% to isolate the benefit provided by geographic diversity alone. In DGS(25%), the total network capacity is lower than the baseline. We measure the amount of data not downloaded from the satellites at the end of the day and plot the cdf of this data backlog in Fig. 3a.

The median (90-percentile, 99-percentile) backlog for the baseline is 8.5 GB (28.9 GB, 80.7 GB). This means that for 10% of the satellites, 28.9 GB data is yet to be downloaded and for 1%, 80.7 GB data is in the backlog. In contrast, for DGS, the corresponding backlog is 1.9 GB (5.3 GB, 16.7 GB). DGS improves the backlog by a factor of 5 for the median as well as the 90-th and 99-th percentile. Even if we limit DGS to 25% of its stations, with total link capacity less than the baseline, the backlog is 3.9 GB (20.1 GB, 66.7 GB). This highlights that a subset of the gains are achieved because of geographic diversity alone, i.e. (a) geographic spread means less satellites conflict at a single ground station, and (b) distributed nature of DGS ensures that degradation of individual links, for example due to weather, do not severely impact the entire system.

Latency: We measure the time elapsed between data capture and data reception at the ground station for the three methods defined above. We plot the cdf of this latency in Fig. 3b. The baseline method achieves a median (90-percentile, 99-percentile) latency of 58 minutes (293 minutes, 438 minutes). In contrast, DGS achieves a latency of 12 minutes (44 minutes, 88 minutes). Even with 25% deployment, DGS achieves a latency of 20 minutes (58 minutes, 88 minutes). This result highlights a key benefit of DGS's geographically distributed design. Even when the overall link capacity of the system is lower, it achieves a way lower latency (4-5 times lower for different metrics). This is because a satellite is likely to encounter multiple ground stations during its orbit.

Adaptability of Value Function: Recall, we use the value function Φ to modulate the behavior of DGS. We ask if tuning the value function has any tangible effects? So far, for all of our results, we had tuned our value function to optimize for latency. We compare this to what happens if we tune our value function to optimize for throughput instead. To evaluate this, we plot the cdf of latency in Fig. 3c for three different methods: DGS(L) – DGS optimized for latency, DGS(T) – DGS optimized for throughput, baseline (L) – baseline optimized for latency. As shown, when DGS is optimized for throughput, the latency goes up – median goes up from 20 mins (90-th percentile: 58 mins) to 22 mins (90-th percentile: 119 mins). This shows that tuning the value function can indeed improve the intended outcome and that DGS is an agile framework for distributed ground station design. We can tune this value function to prioritize data for geographic regions, natural disasters, or just use a bidding system to bid on ground station time. Finally, note that even the throughput optimized system with 25% of ground stations used has a lower latency than the full baseline system optimized for latency. This, again, highlights the low-latency advantage of DGS due to its geographically distributed nature.

5 CONCLUSION

We present DGS, a novel ground station design for Earth observation satellites. DGS is built using a mix of transmit-capable and receive-only geographically-distributed ground stations. While we design DGS nodes to be low-complexity low-cost ground stations, we believe the methods presented in this work are applicable more broadly to other distributed ground station designs. Finally, as discussed in 3.3, we hope that this work initiates future work that tackles multiple open research questions towards realizing an agile, robust, and high-performance distributed ground station system.

ACKNOWLEDGEMENTS

We thank Peder Olsen for multiple brainstorming sessions. We are grateful to Anirudh Badam, Anuj Kalia, and anonymous reviewers for their feedback.

REFERENCES

[1] Amazon Inc. AWS Ground Station . https://aws.amazon.com/ground-station/.

[2] L. Barbulescu, A. E. Howe, J. P. Watson, and L. D. Whitley. *Satellite Range Scheduling: A Comparison of Genetic, Heuristic and Local Search.* 2002.

[3] S. Bonafoni, S. Shrestha, and D.-Y. Choi. Characterization of Rain Specific Attenuation and Frequency Scaling Method for Satellite Communication in South Korea. *International Journal of Antennas and Propagation*, 2017.

[4] K. Colton and B. Klofas. Supporting the Flock: Building a Ground Station Network for Autonomy and Reliability. Small Satellite Conference, 2016.

[5] J. Cutler. Ground Station Virtualization. The Fifth International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations, 2003.

[6] J. W. Cutler, A. Fox, and K. Bhasin. Applying the lessons of Internet services to space systems. IEEE Aerospace Conference, 2002.

[7] Dark Sky. Dark Sky Weather API. https://darksky.net/dev.

[8] B. Denby and B. Lucia. Orbital edge computing: Nanosatellite constellations as a new class of computer system. ACM ASPLOS, 2020.

[9] K. Devaraj, R. Kingsbury, M. Ligon, J. Breu, V. Vittaldev, B. Klofas, P. Yeon, and K. Colton. Dove High Speed Downlink System. Small Satellite Conference, 2017.

[10] K. Devaraj, M. Ligon, E. Blossom, J. Breu, B. Klofas, K. Colton, and R. Kingsbury. Planet High Speed Radio: Crossing Gbps from a 3U Cubesat. Small Satellite Conference, 2019.

[11] A. Escher. Inside Planet Labs' new satellite manufacturing site. TechCrunch. https://techcrunch.com/2018/09/14/inside-planet-labs-new-satellite-manufacturing-site/, 2018.

[12] D. Etherington. SpaceX launches 58 more Starlink satellites and 3 Planet Skysats for first rideshare launch. TechCrunch. . https://techcrunch.com/2020/06/13/spacex-launches-58-more-starlink-satellites-and-3-planet-skysats-for-first-rideshare-launch/, 2019.

[13] ETSI EN 302 307. Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2). Technical report, European Standard (Telecommunication Series), 2009.

[14] European Space Agency. Sentinel 2: Core Ground Segment Design . https://sentinel.esa.int/web/sentinel/missions/sentinel-2/ground-segment/core-ground-segment.

[15] European Space Agency. Sentinel 6: Ground Segment Design . https://sentinel.esa.int/web/sentinel/missions/sentinel-6/ground-segment.

[16] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. Technical report, Brown University, 1960.

[17] C. Henry. Amazon moving Project Kuiper team to new R&D headquarters. SpaceNews. . https://spacenews.com/amazon-moving-project-kuiper-team-to-new-rd-headquarters/, 2019.

[18] F. R. Hoots and R. L. Roehrich. Models for Propagation of NORAD Element Sets. Technical report, Aerospace Defense Command, United States Airforce, 1980.

[19] International Telecommunications Union. ITU P.838: Specific attenuation model for rain for use in prediction methods. Technical report, 2019.

[20] International Telecommunications Union. ITU P.839 : Rain height model for prediction methods . Technical report, 2019.

[21] International Telecommunications Union. ITU P.840: Attenuation due to clouds and fog. Technical report, 2019.

[22] B. Klofas. Planet Labs Ground Station Network. Cubesat Developers Workshop, 2016.

[23] Kongsberg Satellite Services. Ground Station Services. https://www.ksat.no/services/ground-station-services/.

[24] Leaf Space. Leaf Line: Shared Ground Station Network. https://leaf.space/leaf-line/.

[25] Libre Space Foundation. SatNOGS: An Open Source Ground Station and Network. https://satnogs.org/, 2014.

[26] Lockheed Martin. Amazon Web Services And Lockheed Martin Team To Make Downlinking Satellite Data Easier And Less Expensive. https://news.lockheedmartin.com/news-releases?item=128622, 2018.

[27] A. Morello and V. Mignone. Dvb-s2: The second generation standard for satellite broad-band services. *Proceedings of the IEEE*, 2006.

[28] Y. Nakamura and S. Nakasuka. Ground station networks to improve operations efficiency of small satellites and its operation scheduling method. International Aeronautical Congress, 2006.

[29] Planet Labs. Planet Imagery Product Specifications. https://www.planet.com/products/satellite-imagery/files/Planet_Combined_Imagery_Product_Specs_December2017.pdf, 2018.

[30] Planet Labs Inc. Earth Station Application. FCC Request. Attachment A. . http://licensing.fcc.gov/myibfs/download.do?attachment_key=1073553.

[31] B. Preindl, H. Page, and V. Nikolaidis. Genso: The global educational network for satellite operations. International Aeronautical Congress, 2008.

[32] M. Schmidt. *Ground Station Networks for Efficient Operation of Distributed Small Satellite Systems*. PhD thesis, 2011.

[33] M. Schmidt and F. Zeiger. Design and Implementation of In Orbit Experiments for the Pico Satellite UWE-1". *International Astronautical Congress*, 2006.

[34] Union of Concerned Scientists. UCS Satellite Database. https://www.ucsusa.org/resources/satellite-database, 2020.

[35] N. Zufferey, P. Amstutz, and P. Giaccari. Graph colouring approaches for a satellite range scheduling problem . *Journal of Scheduling*, 2008.