L2D2: Low Latency Distributed Downlink for Low Earth Orbit Satellites

Deepak Vasisht[†‡], Jayanth Shenoy[†], Ranveer Chandra[‡]

[†]University of Illinois Urbana-Champaign, [‡] Microsoft

ABSTRACT

Large constellations of Low Earth Orbit satellites promise to provide near real-time high-resolution Earth imagery. Yet, getting this large amount of data back to Earth is challenging because of their low orbits and fast motion through space. Centralized architectures with few multi-million dollar ground stations incur large hour-level data download latency and are hard to scale. We propose a geographically distributed ground station design, L2D2, that uses low-cost commodity hardware to offer low latency robust downlink. L2D2 is the first system to use a hybrid ground station model, where only a subset of ground stations are uplink-capable. We design new algorithms for scheduling and rate adaptation that enable low latency and high robustness despite the limitations of the receive-only ground stations. We evaluate L2D2 through a combination of trace-driven simulations and real-world satellite-ground station measurements. Our results demonstrate that L2D2's geographically distributed design can reduce data downlink latency from 90 minutes to 21 minutes.

CCS CONCEPTS

• **Computer systems organization** → *Dependable and fault-tolerant systems and networks*; • **Networks** → **Network components**;

KEYWORDS

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

ACM Reference Format:

Deepak Vasisht[†‡], Jayanth Shenoy[†], Ranveer Chandra[‡]. 2021. L2D2: Low Latency Distributed Downlink for Low Earth Orbit Satellites. In *ACM SIGCOMM 2021 Conference (SIGCOMM '21), August 23–28, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3452296.3472932

1 INTRODUCTION

Low Earth Orbit (LEO) satellites mark a new frontier in communications and sensing research. Multiple companies [7, 20, 21, 28] have committed to invest tens of billions of dollars to deploy constellations of hundreds of small satellites such as CubeSats. These satellites aim to serve two main objectives: communication and Earth observation. Communication satellites provide low-latency, high-bandwidth, and universal internet connectivity. Earth observation satellites continuously orbit the earth (see Fig. 1) and collect imagery (aka eyes in the sky). Unlike traditional Earth observation satellites, new deployments consist of large LEO constellations of cheap CubeSats. These satellites aim to build a near real-time map of the earth and use it for real-time analysis of agriculture [2, 35], geological systems [62], disease spreads [5], natural disasters [4, 11], and geopolitics [44, 45].

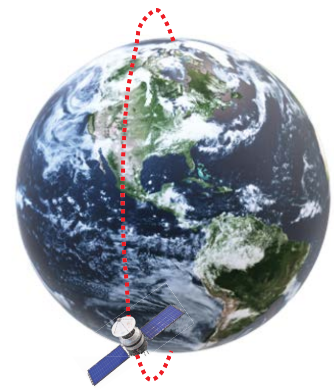


Figure 1: Earth observation satellites typically operate in polar low-earth orbits. As the Earth rotates under them, they scan different parts of the Earth during each orbit. A single satellite may image the Earth across several days, so operators plan to rely on large constellations to achieve collective imaging frequency of minutes to hours.

Today, about 45% of LEO satellites in orbit [59] are used for Earth observation. These satellites collect hundreds of Gigabytes of data every day and need to transmit this data [51] to Earth. This is challenging for LEO satellites because their low orbits mean that they move fast with respect to an observer on Earth. For any ground observer, the satellite is visible for around ten minutes and has four to five good contacts every day (see Fig. 2). As a result, satellite companies deploy a few highly specialized (multi-million US dollars) ground stations [34] that can download large quantities of data in a short period [18, 19]. This design for ground stations suffers from multiple shortcomings:

- **Downlink Latency:** While large constellations [7, 20, 30] promise to collect data every hour to few hours (in some cases, minutes [30]), this data must wait at the satellite before it comes in contact with a ground station. This adds latency of one to several hours, which can be crippling for time-sensitive applications like natural disaster management (e.g. forest fires, floods, etc.) and crop monitoring.

SIGCOMM '21, August 23–28, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8383-7/21/08...$15.00
https://doi.org/10.1145/3452296.3472932

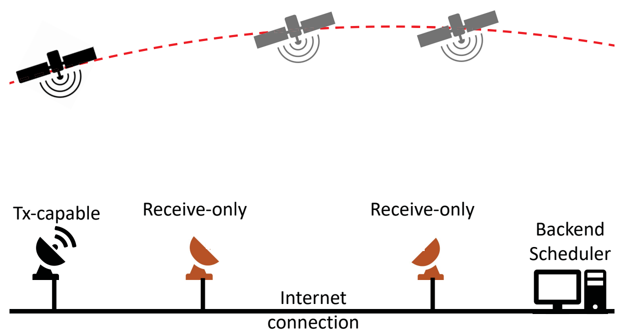


Figure 2: L2D2 is a geographically distributed ground station design. It uses a mix of transmit-capable and receive-only ground stations, to enable low latency, high fidelity data transfer from LEO satellites.

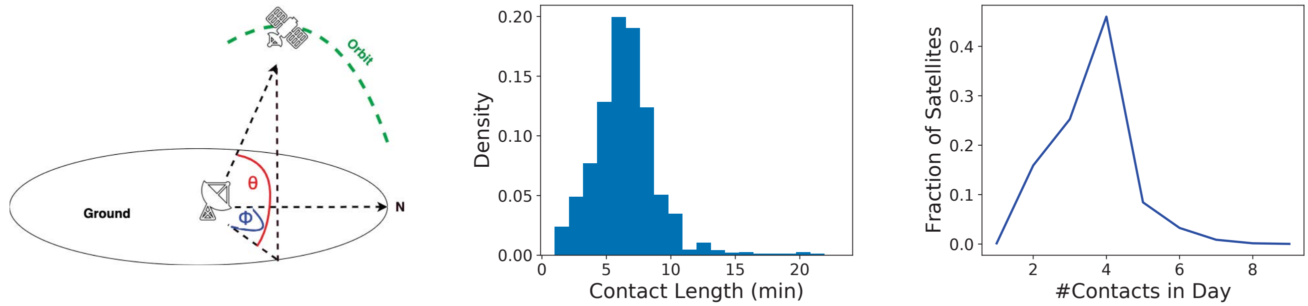
- **Scaling:** When the constellation size is small, the ground stations are under-utilized as they are used for a few minutes per satellite contact. As the promised large constellation sizes materialize, the ground stations are bottle-necked by bandwidth and contention, at which point satellite companies must deploy new ground stations. In addition to the high cost of designing and maintaining these ground stations, they also suffer from deployment delays and million-dollar costs due to regulatory requirements. For example, Amazon Ground Stations could not transmit data for at least a year after their public announcement due to licensing delays [27].
- **Robustness:** The centralized architecture relies on few ground stations that are prone to hardware failures and weather-related connectivity issues. At high frequencies used by the ground stations (8 GHz and above), the links are prone to attenuation of up to 10dB due to rain and clouds [6]. Some LEO satellites have reported up to 88% packet loss [17].
- **High Cost of Entry:** 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 (tens of thousands USD), the ground station becomes the bottleneck.

In this paper, we present a new ground station architecture for LEO satellites: L2D2. L2D2 has two key characteristics: (a) it uses a large set of geographically distributed low-cost ground stations. It relies on commodity equipment (e.g. small antennas) that can be deployed at rooftops and backyards instead of specialized hardware. (b) 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 [18, 19, 34]. Therefore, a majority of the ground stations in L2D2 are receive-only, removing the regulatory burden of setting up new L2D2 stations. In practice, this design choice leads to three key challenges:

- **Adaptive Downlink Scheduling:** Given the large number of satellites and ground stations, we need to dynamically schedule satellite-ground station contacts while accounting for orbits, link quality, and weather conditions. In our dataset with 259 satellites, a ground station may see up to 100 satellites (median 14) at the same instant. However, in a typical setting, a ground station communicates with one satellite at a single point in time. Therefore, we need to identify the optimal satellite-ground station matching across time and space. Furthermore, it must account for switching delays, i.e. a satellite cannot communicate while switching from one ground station to another. We observe that this scheduling problem is a variant of the circuit scheduling problem studied in the context of datacenters [10, 24, 39, 42, 57], which is known to be NP-hard. We leverage this observation to design a new (approximate) greedy algorithm for this scheduling problem that supports multiple objective functions – throughput, mean latency, and peak latency.
- **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 [6]. In typical wireless systems, the optimal rate is selected using feedback from the receiver. How does one perform rate selection in the absence of such feedback? To solve this problem, we build a new link quality predictor that leverages historical data, predicted weather conditions, and orbital dynamics to predict the ideal datarate. The design of this predictor is non-trivial because of the local multipath experienced by satellite signals (e.g. solar panels close to antennas). Unlike state-of-the-art models for satellite signal propagation, our model can account for such ground station and satellite-specific variations.
- **Satellite Feedback:** How does a satellite know if its data has been received at a receive-only ground station and is safe to delete from its storage without acknowledgments? L2D2 uses delayed acknowledgments that are relayed through transmit-capable stations (3-4 in L2D2's network) to the satellite, when the satellite flies over these stations. These acknowledgments allow the satellite to delete data that has been received and re-transmit lost data.

Our design has many advantages. Its geographical distribution ensures that a satellite encounters ground stations more frequently and can offload latency-sensitive data sooner. Allowing the ground station to be distributed also allows L2D2 to relax the requirement of extremely high throughput on individual links, therefore enabling the use of commodity components. Moreover, L2D2 is more robust to failures and weather variations since the impact of individual failures can be minimized by re-routing the data. For example, downlink can be dynamically scheduled so that cloudy weather in one part of the world is offset by clear weather in the other. Finally, L2D2 can allow new entrants to schedule communication for their satellites by providing a software abstraction.

To evaluate L2D2, we use a four-step approach. First, we collect real-world link quality measurements at five ground stations from four satellites operating in the X-band and Ka-bands. We are releasing this new dataset to the community. We use this dataset to validate individual components of our design. Second, to evaluate the distributed design at scale, we leverage the open-source SatNOGS ground stations operated by amateur radio operators. We compile this data from the publicly-hosted SatNOGS database. We use a network of 259 satellites and 173 ground stations deployed by different entities. These ground stations listen to satellite beacons at lower frequencies and do not correctly model the satellite downlink behavior in X-band or Ka-bands. Therefore, in step three, we model



(a) Elevation θ (red) and Azimuth Φ (blue)

(b) Contact Length

(c) #Contacts in Day

Figure 3: LEO Satellite Background: (a) Satellite elevation defines link quality. Higher elevation provides better signal due to shorter distance. (b,c) are reported on our dataset in Sec. 4. A ground station has around four good contacts with a satellite in a day with most contacts lasting less than ten minutes.

the link quality behavior of each of these ground stations as if they were a randomly chosen ground station from our X-band stations in the first step. Finally, we ask, in simulation, how this network would operate if each satellite had 100 GB of data to download per day. Based on this analysis, we summarize our results below:

- **Link Estimation:** L2D2's link estimation algorithm achieves a median error of 0.39 dB in predicting the link quality as compared to state-of-the-art ITU models that achieve 2.39 dB median error (90th percentile 1.74 dB vs 6.03 dB). This translates to a datarate loss of 6.5% for L2D2 while the ITU model suffers 40.8% loss.
- **Latency:** L2D2 reduces the mean latency of data download from 90 minutes to 21 minutes and the 90-th percentile from 323 minutes to 71 minutes as compared to a baseline method that uses high-end ground stations with 10X more link capacity than L2D2's ground stations.
- **Data Transfer and Backlog:** L2D2 downloads over 250 TB of data in a day from 259 satellites. In an experiment with each satellite collecting 100 GB per day, L2D2 reduces the median backlog (data not delivered) for a satellite from 7.6 GB to 3.4 GB (90-th percentile: 26.5 GB to 7.2 GB).

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 by the past shifts in computing from singular highly specialized hardware to distributed low-complexity components. In designing L2D2, we make the following contributions:

- We present a new distributed hybrid design that uses low-cost receive-only ground stations to ensure low latency data transfer from LEO satellites.
- We design a novel scheduler for satellite-ground station links that accounts for temporal variation & switching delay.
- We build the first data-driven blind rate adaptation algorithm for LEO satellites.
- We evaluate L2D2 using measurements and large-scale simulations performed using multiple satellites and ground stations.

2 BACKGROUND

The recent interest in satellites has been precipitated by the low cost of manufacturing small satellites like cubesats and launching them into low earth orbits (300 to 500 miles above Earth surface) using rideshare agreements. Together, these factors have ensured that cheap hardware can be deployed in space for tens of thousands of dollars. This is in contrast to costs of tens of millions for traditional satellites [17]. In this paper, we focus on Earth observation satellites. The primary goal of these satellites is to get very frequent, near real-time data about Earth.

Imaging Equipment: Earth observation satellites serve multiple high-value applications like agriculture, forestry, smart traffic, natural disaster response, and geopolitical analysis. As such, they carry a wide array of sensing mechanisms in different parts of the spectrum. The most prominent ones are visible imagery and SAR-imagery [7] (synthetic aperture radar), but sensing in various parts of the spectrum (infrared, microwave, ultraviolet) is also prevalent [59]. As orbits get lower and equipment becomes precise, the resolution of this imagery has been steadily improving. Of late, several constellations offer meter-level pixel resolution [7, 20, 30].

Orbital Dynamics: The LEO satellites typically operate in polar orbits (see Fig. 1). One orbit period is around 90 minutes for satellites in LEO orbits. Since the orbit period is different from the period of Earth's rotation, the satellite observes a different part of Earth during each orbit. A single satellite can image the Earth across several days. Therefore, Earth observation satellites operate in constellations to achieve frequent image capture. The planned constellation sizes consist of hundreds of satellites [7, 20, 30] to get a new image every few minutes to few hours.

The location of the satellite with respect to an observer on Earth is defined by two angles: azimuth and elevation (Fig. 3a). The azimuth defines the angle with respect to the north-south direction in the Earth-surface plane. The elevation is the angle measured perpendicular to the Earth's surface. An elevation of 90 degrees corresponds to the satellite directly overhead. Due to their low orbits, an observer on Earth would see a LEO satellite rise from the horizon, travel through the sky, and fade below the horizon on the other end within few minutes. An observer would get four to five such contact periods in one day (see Fig. 3), with each contact achieving different peak elevation. When the satellite gets to a higher elevation, it is closer to the Earth, can deliver a stronger signal to a ground station on the Earth, and has a longer contact period.

Data Communication: For a traditional ground station on Earth, the uplink comprises primarily of low rate TT&C (tracking, telemetry, and control) data. Consequently, the typical design of ground stations uses a narrowband uplink (tens to hundreds of kbps) and a high bandwidth downlink (hundreds of Mbps to tens of Gbps) [18, 19, 34]. This design choice also accounts for the choice of spectrum. Public documents [22, 23, 52] show that the uplink uses the lower frequency and lower-bandwidth S-band (2025-2110 MHz) [1] 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 [19].

Today, most satellite operators deploy their own ground stations [18, 19]. These ground stations cost millions of dollars to deploy and maintain due to three factors: (a) high-end specialized equipment, (b) expensive licensing process for transmission, (c) large antenna sizes require acquisition of dedicated space. In contrast, L2D2's ground stations operate with commodity hardware, use small antennas deployable on rooftops, and majority of them are receive-only.

State-of-the-Art: The best known ground station design today can achieve a data rate around 1.6 Gbps by combining six frequency-polarization channels at the optimal satellite-ground station link [19]. The 1.6 Gbps link in [19] can download data up to 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 four to six passes per ground station per day, but the passes have varying quality. The typical amount of data that needs to be downlinked to image the Earth every day can go up to few Terabytes per day [13]. Multiple satellites need to collaborate to make this happen.

3 DISTRIBUTED HYBRID GROUND STATION ARCHITECTURE

We propose a new distributed hybrid ground station architecture, L2D2, for Earth observation satellites. An overview of L2D2 is in Fig. 2. L2D2 consists of multiple ground stations spread across the globe. Each of these ground stations is connected to the Internet and communicates with a centralized scheduler. L2D2 ground stations have three distinctive characteristics:

- **Geographically Distributed:** L2D2 ground stations are spread across the globe, either maintained by independent individuals, volunteers, or corporations. This geographic distribution of ground stations 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 monitoring 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 [34] that are both expensive and time-consuming. In L2D2, 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 require the cost and complexity of individual ground stations to be low. As such, the individual ground stations in L2D2 do not have high gain, specialized equipment, but rely on commodity hardware that is easily deployable on rooftops and backyards. This decreases the capacity of individual links by an order of magnitude or more, but the reduced capacity is compensated through geographic diversity.

[1]Different organizations have slight variations in the definition of bands.

Overview: In L2D2, a scheduler estimates the trajectory of a satellite for a fixed future time-interval (e.g. one day). 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 (4-5 in our network). 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 the next contact with a transmit-capable station (Sec. 3.3).

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. Before we delve deeper into L2D2's downlink scheduler, we note that scheduling downlink for a distributed hybrid architecture like L2D2 is fundamentally different from scheduling for a centralized architecture with a small number of ground stations and satellites. In centralized architectures, it is rare for multiple satellites to compete for a ground station's time. This is because the number of satellites is small (typically only from a single provider) and each satellite-ground station contact lasts just ten minutes. The goal of L2D2 is to serve as a ground station fabric spread across the planet that can downlink data from satellites from multiple providers. At such scale, conflicts become the norm. In our dataset (Sec. 4), we see up to 100 satellites (median 14) competing for a ground station. More importantly, innovative scheduling benefits the robustness of our architecture. When a satellite-ground station link is going to be weakened by hardware constraints, multipath effects, or bad weather, we can predict it in advance and schedule the satellite to downlink at a different ground station.

Intuitively, we aim to find a scheduling algorithm that can optimize a pre-defined value function across time and across all satellite-ground station pairs. As we formalize this problem below, we face two high-level challenges. First, for a satellite, switching between ground stations is not instantaneous. They may have to steer their antenna either mechanically or electronically and incur cost in terms of delays, lost throughput, etc. Second, the problem of identifying the best schedule across space-time with switching delays is known to be NP-hard. We present an algorithm that tackles these challenges below.

Problem Formulation: 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) [29]. TLE is a standard representation for satellite orbits that contains the satellite identifier and orbit parameters. The TLEs are time-varying and are updated over time. For LEO satellites, satellite location prediction using TLEs is accurate to within a kilometer if done a few days in advance. 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^t that it intends to send to ground stations. X_i^t varies with time t as the satellite collects more data and downlinks some of it to a ground station. We define a value function, ϕ such that for any subset $x \subset X_i^t$, $\phi(x, t)$ denotes the value of transmitting that data to Earth. This value function can capture different objectives. For instance, it can capture throughput or latency requirements. 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.

Orbit Calculations: L2D2 obtains the most recent TLE data for each satellite and uses it to compute the future orbit of the satellite. At each instance of time, we compute if a satellite is above the horizon for a ground station. If a satellite is above the horizon, L2D2 computes the distance, the elevation, and the azimuth angle of the satellite with respect to that ground station. These parameters are fed to a link estimation algorithm described in 3.2.

Scheduling Algorithm: We use the link estimation algorithm to predict expected data rate that a satellite-ground station link can achieve and use it form the matrix, D^t such that D_{ij}^t is the data that can be transmitted between satellite, s_i and ground station, g_j at time instant t. Therefore, we can define a value matrix, Φ^t, such that $\Phi_{ij}^t = \phi\left(\min(X_i^t, D_{ij}^t), t\right)$. If satellite, s_i, has enough data to transmit, it transmits D_{ij}^t, otherwise it transmits all the data it has, $X_i^t \cdot \Phi_{ij}^t$ defines the value of transmitting that data.

Our objective is to match satellites to ground stations at each time step to maximize the corresponding value function. Specifically, this objective function can be defined as finding a sequence of permutation matrices, P^t, $t = 0, ..., T$, where T is the time interval at which the schedules are generated (for example, one day). Recall, a permutation matrix is a binary matrix with exactly one value in each row (and each column) set to one. Therefore, pre-multiplication by a permutation matrix shuffles the rows of any matrix. In our case, the permutation matrix, P^t, shuffles the matrix, Φ^t. Then, we consider the sum of the diagonal values of the resultant matrix to compute the net value achieved by the system at time, t. The goal of our algorithm is to identify a sequence of permutation matrices that can maximize the value over time. Specifically, we aim to optimize:

$$\max_{P^0, P^1, \ldots P^t} \sum_t Tr(P^t \Phi^t) \tag{1}$$

where $Tr(\cdot)$ defines the trace (sum of diagonal values) of the matrix. We pad the matrix Phi^t to make it a square matrix with each dimension $max(M, N)$. Intuitively, iff $P_{ji}^t = 1$, then ground station, g_j is matched to satellite, s_i, at time t.

Algorithm 1: Greedy Algorithm for Schedule with Switching Cost

input : A sequence of value matrices: Φ^t, $t = 1, 2, .., T$. The bonus factor for a sticky schedule: b.
output: A sequence of permutation matrices: P^t, $t = 1, 2, .., T$

```
for t = 1 to T do
    if t > 1 then
        B^t = b P^{t-1 ⊤} · Φ^t
        where ⊤ is matrix transpose and · is dot product
        P^t = arg max_{P^t} Tr(P^t (Φ^t + B^t))
    else
        P^t = arg max_{P^t} Tr(P^t Φ^t)
    end
end
```

Switching Penalty: Note that Eq. 1 does not penalize a satellite for switching between ground stations. In practice, switching between satellite-ground station pairs incurs non-trivial costs. This is because many ground stations rely on mechanical steering. Therefore, we must institute a penalty for switching delay when a satellite performs the switch.

We note that this scenario is similar to the circuit switching problem encountered in the context of data centers [10, 24, 39, 42, 57]. In such problems, switch reconfiguration incurs a delay, therefore the schedules have to be incentivized to be sticky. This problem is known to be NP-hard and has approximate-optimal greedy solutions. However, such circuit switching formulations penalize the entire system for reconfiguration delays. In our case, the reconfiguration impact is localized to the satellite making the switch. Therefore, we adapt the greedy algorithms used for circuit switching to our problem with localized penalties for switching.

Specifically, we define a bonus matrix B^t, such that $B_{ij}^t = b \times \Phi_{ij}^t$ iff $P_{ji}^{t-1} = 1$, and zero otherwise. Note, $B_{i,j}$ is positive and non-zero at time t only when the connection between satellite, s_i, and ground station g_j existed in the prior time step. Here, $b \in R^+$ is the bonus factor assigned for sticking to the original matching and is inversely proportional to the cost of switching. Note that this bonus is applied

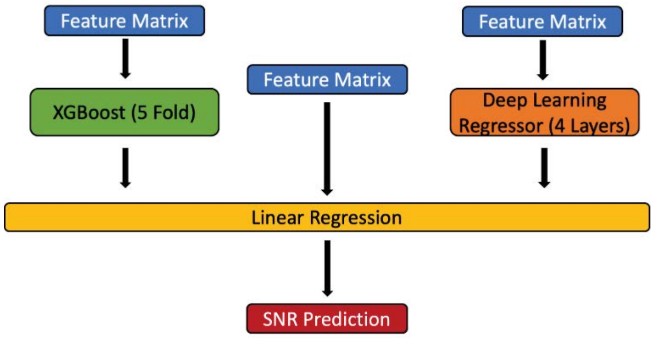


Figure 4: L2D2 Link Prediction Model Architecture

multiplicatively. Therefore, a satellite that has a good connection to a ground station has a higher incentive to stick to its current match, unlike a satellite that has a poor connection. Then, at each time step, we define the greedy algorithm to pick P^t such that:

$$P^t = \arg\max_{P^t} Tr(P^t(\Phi^t + B^t)) \tag{2}$$

Finally, how do we solve Eq. 2 to identify the optimal P^t? We can map this to a maximum bipartite graph matching problem. In L2D2, we use the Hungarian algorithm [38, 47] to identify the best match and hence P^t. We present a description of the algorithm in Algorithm 1. L2D2 uses the satellite-ground station match identified by this algorithm to schedule links over time.

3.2 Rate Adaptation

For a typical wireless link, the ideal data rate for data transmission is estimated either using direct feedback from the receiver or through indirect feedback e.g. acknowledgments. For example, in the state-of-the-art radio design for cubesats [19], the radios use DVB-S2 protocol for communication and use adaptive coding and modulation to adapt to different satellite-ground station link conditions. The satellite radio selects the best modulation and coding configuration based on feedback received from the ground station.

In L2D2, a majority of our ground stations are receive-only. As such, we cannot rely on feedback from the ground station to identify the right data rate to downlink data. This is crucial because if the satellite chooses too low a data rate, it would waste opportunities to downlink more data. Conversely, picking a data rate that is higher than what the link can support would lead to high packet loss. Furthermore, we need to identify the amount of data that will be transmitted during a contact *before* the communication begins to identify the optimal satellite-ground station links using Algorithm 1.

To solve this challenge, we propose a new rate adaptation algorithm that does not rely on observed link quality but uses predicted link quality. We build a new prediction framework that can estimate the signal strength and signal-to-noise ratio for a signal received at the ground station, and use it to identify the ideal rate for data transmission. Intuitively, the loss in signal strength as the signal travels from a satellite to the Earth happens because of three reasons: (a) Propagation loss: This increases with distance and signal frequency (logarithmic). This loss increases as the elevation of the satellite decreases. Lower elevation implies that the satellite is closer to the horizon and as a result, the signal has to travel a longer distance. (b) Weather-related Loss: Clouds, rain, and snow induce additional attenuation for the signal. This can vary from zero to 10 dB depending on the weather and signal frequency [6]. Higher frequencies experience this effect more severely. (c) Hardware effects: The design of the antennas and circuits in the transmitter and receiver introduces a link-specific loss.

At a high level, the propagation loss can be estimated using orbit calculations and the hardware effects can be calibrated for. There are also standard models for understanding weather-related loss [31–33]. Therefore, in principle, a combination of these will help predict the link quality accurately. However, as we demonstrate in Sec. 6, this combination proves to be empirically inaccurate. Our analysis of real measurements reveals that the surroundings of the satellite and ground stations play an important role in link quality. Specifically, the presence of reflectors (like solar panels) on satellites, their relative positions, and their motion stability impacts the signal strength at the receiver. Similarly, the presence of buildings, trees, etc. in the proximity of the ground station causes obstacles or reflections on the receiver's end.

Since this behavior varies with the satellite and ground station, we take a data-driven approach to estimating link quality. Specifically, we develop a machine learning model for each satellite-ground station link that can learn the quirks of each link and its temporal variation. Our model uses (a) satellite orbital position relative to the ground station and (b) weather conditions at the ground station to predict the signal strength of the link. L2D2's choice to use machine learning is motivated by two factors: (a) the reflectors and obstacles near satellites vary per satellite and are hard to model without observed data, and (b) signal strength data is easily available at ground stations. Therefore, these models can be easily trained using data measured at the ground station during the first few days after setup.

Model Architecture: L2D2 uses an ensemble model to predict signal strength. Ensemble models are known to be highly performant because they combine multiple model architectures that can extract varied insights. Specifically, L2D2 uses two model architectures: (a) gradient boosted regression trees and (b) deep learning-based regression. The tree ensemble fits on five folds of the input feature matrix. We use 1000 trees as weak estimators with a maximum tree depth of 8. We use the XGBoost implementation [12]. The deep learning regression model has 4 layers. Each layer uses rectified linear unit (ReLu) activation functions to incorporate non-linearity. We stack the output of these two models and feeds into a final linear regression model that outputs the signal strength estimate. Fig. 4 illustrates the design.

Input Features: L2D2 uses features that describe (a) satellite orbital positions: elevation and azimuth angle of the satellite with respect to the ground station, and (b) weather: precipitation intensity. As expected, due to their high frequencies, Ka-band satellites are more sensitive to weather. Therefore, we augment the dataset with cloud cover and precipitation probability in addition to precipitation intensity. X-band satellites are not as susceptible to atmospheric variation so we do not include cloud cover and precipitation probability for them. The models can be trained to either output signal strength or the signal to noise ratio. The output is measured on the log scale, in dB or dBm, as is the norm in wireless systems.

Loss Function: We train the models using the mean squared error loss. However, if we use the output of the model directly to estimate the data rate, it is expected to make roughly equal errors in terms of over-estimating and under-estimating the signal strength. In reality, over-estimation errors for data rate lead to packet losses (100% loss) while under-estimation errors just lead to a partial data rate loss. Therefore, we learn another buffer parameter, ϵ, from the data. We subtract ϵ from the model output and choose ϵ (during training) such that the data rate prediction error is minimized. This value (typically 1-2 dB) helps us reduce the data rate loss even further. Finally, note that we do not train the regression model to directly predict the data rate because it is a discrete parameter and leads to a non-differentiable loss function.

Induction of New Ground Stations: In L2D2, we expect new ground stations to join the network frequently. Therefore, it would be inefficient if we wait to train a new model for each satellite before we can start using this ground station effectively. Therefore, we devise a transfer learning strategy that uses models learnt at other ground stations for a given satellite to infer a model for a new ground station, thereby reducing the startup time for a new ground station.

We describe several modifications to Fig. 4 for transfer learning in order to estimate the link between a newly added ground station g_i and an existing satellite s_k. Let us assume that we already have models trained for satellite s_k and ground station g_j. We freeze the gradient boosted tree ensemble and the top three layers of the deep learning regression framework, based on the pre-trained model $(s_k - g_j)$. Then, we use a small number of measurements on a new ground station, g_i, to re-train the last layer of the deep learning regressor and the linear regression model shown in Fig. 4. This method allows us to encode the generic features of the satellite s_k in the frozen layers that are transferred from the other ground station, and helps us quickly get to the ground station specific features. In doing so, we transfer our model from one ground station to another and reduce the induction time for a new ground station.

3.3 Satellite Feedback

Since a majority of L2D2 ground stations are receive-only, they cannot transmit acknowledgments (transport-layer or application-layer) back to the satellite. This has two implications: (a) the satellite cannot re-transmit data when packets get lost, e.g., due to errors in link estimation, (b) the satellite cannot remove data from its storage. This will, naturally, be untenable as cubesats have limited storage capacity. Therefore, we must identify a mechanism to communicate this information with the satellite.

To achieve this objective, we deploy a delayed-relayed acknowledgment mechanism. Specifically, every L2D2 ground station checks its received imagery and creates a bitmap for the received data. In this bitmap, each bit denotes if a packet was received successfully. This bitmap is then broadcast to all transmit-capable stations (typically three to five stations) over the Internet. Note that, these bitmaps are small and can be transferred at low latencies over the Internet. This bitmap is then conveyed to the satellite when it comes in contact with a transmit-capable station. This can be few minutes to few hours after the actual packet transmission. On receiving the acknowledgment, a satellite deletes the data that was successfully received at the ground station and places the unacknowledged data back in its transmit queue. Note that, with centralized architectures, the satellites have to store data for a few orbits anyway, so our approach does not increase the storage requirement on a satellite (see Sec. 6). In addition, L2D2 ground stations receive the data from satellites at a lower latency than traditional architectures.

3.4 Discussion and Open Questions

Satellite Transmit Power: Typical satellites transmit data at a higher power than required for successful signal transmissions. This is done to avoid small link quality fluctuations due to orbits, obstructions, weathers, etc. L2D2's link estimation algorithm can identify such fluctuations in advance, and therefore reduce the need for extra power in transmission, resulting in power savings at the satellite.

Satellite Power Management: Transmissions from satellite radios consume significant power. A design like L2D2 requires satellites to transmit data more frequently. Therefore, future iterations of L2D2 should incorporate power budgets of the satellite radio in the scheduler design optimization.

Edge compute on the ground station: Past proposals [17] 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 L2D2 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 L2D2 by enabling each ground station to split power between multiple satellites, thereby increasing the data downlink efficiency. A similar question arises when ground stations can leverage multiple frequencies to communicate with different satellite constellations. We leave the exploration of these new optimizations to future work.

Backward Compatibility: L2D2'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 due to lack of public documentation.

Economic and Security Implications: L2D2's adoption hinges on appropriate economic incentives for operators to deploy ground stations and a security framework to prevent data misuse. This is an exciting direction for future research.

4 EXPERIMENTAL SETUP

We evaluate L2D2 using a combination of real-world link quality measurements and simulations as we describe below.

4.1 Link Quality Measurements

To evaluate L2D2's capability to predict the quality of individual links, we collect real-world data from 16 ground station-satellite

Figure 5: L2D2 Setup: Ground stations (red dots) used for evaluating L2D2.

X-band Satellites	NOAA-20/JPSS1, AQUA, TERRA, SNPP
Ka-band Satellites	JPSS [50]
X-band Ground Station Locations	Wisconsin, Hawaii, Guam, Florida
Ka-band Ground Station Locations	Antarctica [50]

Table 1: Dataset Details: We collect link quality data at five ground station locations from four satellites. We use this data to validate our design.

pairs operating in the X-band and one ground station-satellite pair operating in the Ka-band (see Table 1). Recall, X-band is the most popular downlink band for Earth imagery satellites today. The ground stations used to obtain this dataset cover a large geographical spread: Wisconsin, Hawaii, Antarctica, Guam, and Florida. These ground stations collect data from 5 satellite links (X-band: NOAA-20/JPSS1, AQUA, TERRA, SNPP & Ka-band: NOAA-20/JPSS1) and measure the signal strength and SNR of the received signal in dBm, along with the elevation and azimuth angles of the satellite. In total, we compile data for 30 days worth of passes for each X-band ground station-satellite pair and 6 days worth of passes for the Ka-band ground station-satellite pair [2]. We augment our dataset using weather data – precipitation intensity, precipitation probability, and cloud cover obtained using the Dark Sky weather API [16].

4.2 Large Scale Simulation

Since the deployment of X-band ground stations today is limited, we rely on amateur ground stations deployed in lower frequencies to evaluate the scheduling aspects of our design. Specifically, we evaluate L2D2 using data collected from deployments of the open-source SatNOGS ground stations [41]. SatNOGS is deployed by independent amateur radio enthusiasts using software-defined radios. The ground stations listen to low-bandwidth satellite broadcast signals primarily from government and academic satellites e.g. from NOAA weather satellites. The observation data is logged in a public database. 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. 5) and 259 satellites. We download the data from all ground station-satellite links for a month-long period. Then, we model L2D2 ground stations to be positioned at the same location and interacting with the same set of satellites. Since our simulations are based on positional and orbital information from real-world ground stations and satellites, our evaluation accurately models the geographical distribution of a network that has independently evolved over time.

A majority of SatNOGS 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) [19, 34] to download their data, we cannot use the data from the SatNOGS database to get the SNR for satellite-ground station links. Therefore, we simulate data download behavior for each L2D2 ground station in the X-band. For SNR estimation, we use models trained in Sec. 3.2 to predict the SNR at each ground station. Specifically, each L2D2 ground station emulates the behavior (packet loss rates, SNR variation, reflections, etc.) of a randomly chosen ground station in Table 1.

For simulating data transfer, each satellite generates 100 GB of data per day. Since our ground stations are low-complexity, L2D2 ground stations do not use large dishes (5 m or more) typically used by commercial ground stations [34, 52]. We simulate our ground stations to have small, 1m diameter, dish antennas that can be deployed on rooftops or backyards and are similar to the setup deployed by SatNOGS operators. This reduces the SNR of each station by 6 dB. Furthermore, our ground stations use a single-channel receiver, as opposed to six-channel receivers in centralized designs [19]. Finally, L2D2 computes the data download plan at the granularity of a day. Finer granularity than a day is possible but we haven't explored this in L2D2.

4.3 Model Details

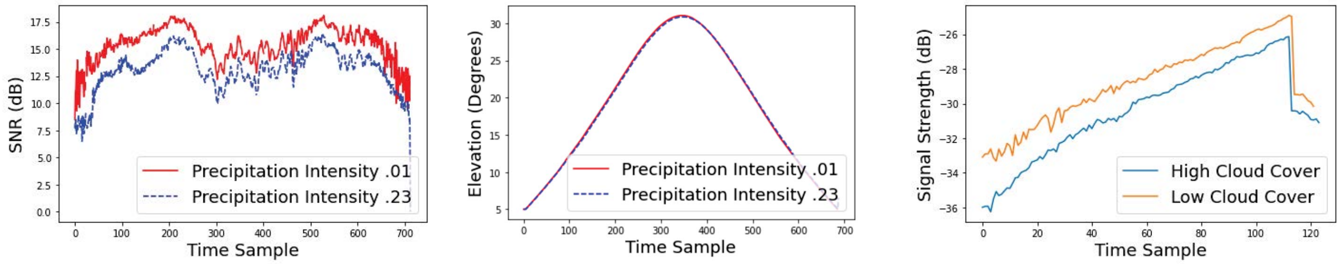
To train the link quality model, we use the last 25% of each dataset as our evaluation set. For X-band satellite, we use 3-weeks of training data and the last week as the test set, unless specified otherwise. We implement the deep learning component of our model in Keras and train our model using the Adam optimizer for 40 epochs in all trials. The model trains on an off-the-shelf Macbook in around 40 minutes and performs a prediction in sub- millisecond (without GPU). We convert estimated SNR to data rate using the 64K blocksize thresholds in a standard satellite data receiver [61].

4.4 Baselines

Our primary baseline to compare L2D2's performance is a centralized architecture that deploys the state-of-the-art ground stations described in [19]. This method uses 6 parallel channels as well as high-end receivers with 4m diameter dish antennas. As in [19], we model 5 such high-end ground stations across the planet. In contrast, each L2D2 ground station uses off-the-shelf components – net link capacity is $6dB$ lower per link and we do not use any parallel channels. Therefore, each baseline ground station achieves 10x the median throughput achieved by a L2D2 node. Furthermore, L2D2 ground stations experience packet loss due to errors in link prediction. In contrast, the baseline stations are transmit-capable and do not experience such loss.

Our evaluation for link estimation compares against a well-studied statistical model from the International Telecommunication Union (ITU) [31–33]. This model uses the distance between satellite

[2]We released these datasets at: https://github.com/ConnectedSystemsLab/L2D2.



(a) X-Band Signal Variation **(b) Satellite Position** **(c) Ka-Band Signal Variation**

Figure 6: Microbenchmarks: (a,b) The SNR and elevation of a satellite measured during two similar passes over a ground station. The SNR of the signal (generally) increases with elevation and decreases with (light) rain. (c) Ka-band satellites experience attenuation even with cloud cover (no precipitation).

and ground station, satellite elevation, and precipitation to mathematically model the link quality.

5 MICROBENCHMARKS

We present some microbenchmarks for L2D2 below. These sample findings are part of our real-world satellite measurement dataset.

5.1 Elevation & Link Quality

Fig. 6a presents the measured signal-to-noise ratio (in dB) for two sample passes of the JPSS1 satellite over the ground station in Guam. The elevation of the satellite is shown in Fig. 6b. As shown in the figure, as the satellite enters the view of the ground station, its signal strength is fairly low. Gradually, its signal strength increases as the elevation increases and then, goes back down as expected. The signal strength variation is around 10 dB which would be enough to cover the entire modulation span for satellite communication receivers like [61]. As specified earlier, the max elevation in a pass does not reach 90 degrees (overhead) in every pass. Note that, while the relationship between elevation and SNR generally holds, the middle of the pass experiences variations that aren’t explained by that. Such variations are caused by reflectors on satellites and near ground stations.

5.2 Impact of Weather on Link Quality

To show how weather impacts link quality, we plot the SNR of two passes in Fig. 6a with identical elevation measurements in Fig. 6b. The only difference between these passes is the weather where the red curve has light rain. As shown in the figure, rainy weather leads to decreased signal strength by 2 to 3 dB. Moreover, the effects on the signal due to weather are even more noticeable at higher frequencies. Fig. 6c illustrates the signal strength differences between two Ka-band satellite-ground links that are similar with respect to position, but varying with respect to weather. As shown in the figure, just a cloud cover (no precipitation) can cause a significant signal strength dip.

6 RESULTS

In this section, we describe L2D2’s evaluation.

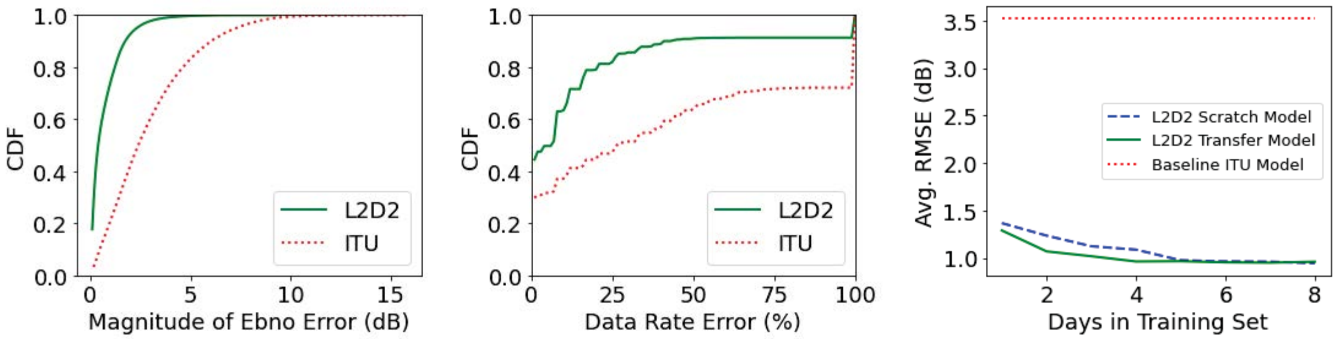
6.1 Link Quality Estimation

To evaluate our link quality estimation model (Sec. 3.2), we use the real-world ground station-satellite measurements described in Table 1. An overview of the results is in Fig. 7. We plot the CDF of the absolute error between predicted SNR and measured SNR across all satellite-ground station pairs in Fig. 7a. L2D2 achieves a median error of 0.39 dB and a 90-th percentile error of 1.74 dB. The median error is small enough that it is unlikely to cause the data rate prediction to cross from one configuration to another. In contrast, the state-of-the-art ITU models achieve a median error of 2.39 dB (90th percentile 6.03 dB), which is 6 times higher. We believe this is because the ITU models are designed for aggregate link behavior and hence do not capture the fine-grained variation caused by individual satellite-ground station design and surroundings. For example, they do not capture local multipath effects or antenna quality variations.

To dig deeper, we break this error down by individual satellite-ground station links. Table 2 reports the mean error for individual satellite-ground station links. There are two points to note here: first, L2D2 consistently outperforms the ITU models by a significant margin. Second, some pairs have higher error than others. This is because of the environmental characteristics (obstacles, reflections) of the setup. For instance, the satellites SNPP and JPSS have antennas close to moving solar panels which makes it harder to predict the link quality.

Data Rate Analysis: How do these errors translate to data rates? To evaluate this, we convert both the predicted SNR and actual SNR into their optimal data rates as described in Sec. 3.2. We plot the CDF of the percentage loss in data rate in Fig. 7b. Note that, if we over-predict the data rate, the error is hundred percent since the SNR cannot support that data rate. As shown, L2D2 significantly outperforms the ITU models. Specifically, we achieve a median data rate error of 6.25% (75-th percentile: 15.5%, 90-th percentile: 41%), and lose less than 10% of the packets. In contrast, the ITU baseline experiences a median data rate error of 41% (75-th percentile: 100%) and loses more than 30% of the packets. This shows that L2D2’s link estimation model is precise in its data rate prediction and can form the foundation of L2D2.

Transfer Learning Analysis: Next, we want to dive deeper into the transfer learning aspect of our model and understand its impact on newly added L2D2 ground stations. For this part of the evaluation, we attempt to learn the links for all satellites on the Guam ground station from the Wisconsin ground station. We pre-train four transfer learning models, one for each satellite, using



(a) SNR Error (b) Data Rate Error (c) Transfer Learning

Figure 7: Link Estimation Evaluation: (a) L2D2 achieves very accurate SNR prediction (0.39 dB median loss) as compared to the baseline ITU model (2.39 dB median loss). (b) This translates to a median data rate loss of 6.25% for L2D2, as opposed to 40.8% for ITU. (c) Transfer learning provides good initial predictions before learning a new model from scratch becomes beneficial.

	JPSS		**SNPP**		**Aqua**		**Terra**	
Error (dB)	L2D2	ITU	L2D2	ITU	L2D2	ITU	L2D2	ITU
Wisconsin	**1.05**	2.93	**1.16**	3.15	**0.29**	3.52	**0.38**	3.61
Florida	**1.58**	3.43	**1.65**	4.00	**1.07**	2.07	**0.40**	4.44
Guam	**1.18**	3.80	**1.59**	3.94	**0.34**	4.26	**0.34**	2.10
Hawaii	**1.27**	3.03	**1.42**	3.66	**0.61**	1.60	**0.66**	4.17
Antarctica	**1.20**	4.30						

Table 2: Mean Link Estimation Error (dB) for each satellite-ground station pair.

all 30 days of the Wisconsin dataset. We then incrementally fit on data from the Guam satellite pairs. Although long-term we find our model trained from scratch to be more effective in predicting the link quality, Fig. 7c demonstrates our transfer learning model's superior performance in the early days of training when the ground station has collected very few link quality measurements. In general, ground station-satellite pairs that were susceptible to high link quality variance due to multipath (JPSS and SNPP satellites) benefited significantly more from the transfer learning based approach.

6.2 Data Download

We evaluate the ability to downlink data from 259 satellites for the baseline high-fidelity ground stations and L2D2. We compare two variants L2D2 and L2D2(25%). L2D2 uses all the 173 ground stations in the network to download data. In L2D2(25%), we reduce the number of simulated stations to 25 % of the original (173) to isolate the benefit provided by geographic diversity alone. L2D2(25%) has a lower aggregate capacity than the baseline. We measure the amount of data not downloaded by the satellites at the end of the day and plot the cdf in Fig.8a. The median (90-percentile) backlog for the baseline is 7.6 GB (26.5 GB). This means that for 10% of the satellites, 26.5 GB data is yet to be downloaded. In contrast, for L2D2 the corresponding backlog is significantly less, 3.4 GB (7.2 GB). Even if we limit L2D2 to 25% of its stations, with aggregate capacity less than the baseline, the backlog is 3.8 GB (12.1 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 L2D2 ensures that degradation of individual links, for example due to weather, do not severely impact the entire system.

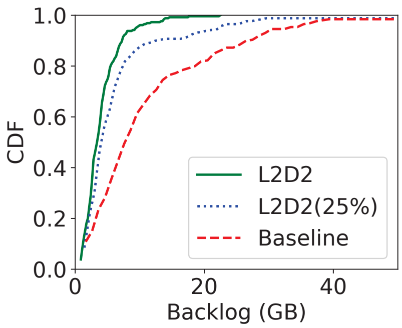
6.3 Latency

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. 8b. The baseline method achieves a median (90-percentile) latency of 90 minutes (323 minutes). In contrast, L2D2 achieves a latency of 21 minutes (71 minutes) – a 4.3X improvement over the baseline. Even with 25% deployment, L2D2 achieves a latency of 38 minutes (105 minutes). These results highlight that even when its overall link capacity is lower, L2D2 achieves a much lower latency because a satellite is likely to encounter multiple ground stations during its orbit. This result shows that distributing downlink over larger areas is beneficial for ground station design. Crucially, L2D2 reduces the latency of downlink to tens of minutes bringing us closer to the vision of near-realtime satellite imagery of the Earth.

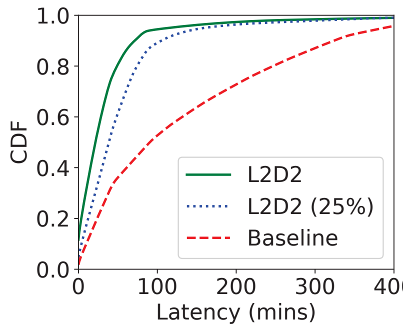
6.4 Scheduling

In Sec. 3.1, we describe a scheduling algorithm for L2D2. Now, we evaluate the impact of our scheduling algorithm within the L2D2 network. For this analysis, we use a random scheduler as a baseline. For each ground station, the random scheduler randomly picks one of the satellites that has a feasible link to the ground station. We report the results in Table 3. All the other components of L2D2 (like link estimation) are ported over to the random scheduler. As shown in the table, L2D2's optimized scheduler (L2D2-OPT) overperforms random scheduler (L2D2-Random) across multiple metrics (10% to 80% improvement). This difference is more prominent for L2D2(25%) as it is a more constrained system and therefore, benefits more from optimized scheduling.

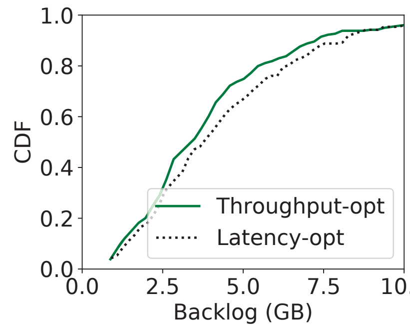
Adaptability of Value Matrix: Recall, we define the value matrix Φ to modulate the behavior of L2D2. We evaluate the effects of tuning the value matrix to optimize for latency vs throughput. Fig. 8c plots the backlog of both the value matrix variants. The result demonstrates that tuning that value matrix to optimize throughput reduces backlog by 11% as compared to a latency-optimized version. Having the ability to tune this value matrix allows us to control



(a) Data Backlog



(b) Latency



(c) Value Function

Figure 8: L2D2 outperforms the baseline in terms of (a) data delivered and (b) latency, even when the number of ground stations are reduced to 25%(L2D2(25%)). (c) Changing the value function from throughput to latency increases the backlog by 10%.

Method	Latency (min)	Backlog (GB)
Baseline	90 (323)	7.6 (26.5)
L2D2-Opt	21 (71)	3.4 (7.2)
L2D2-Random	26 (81)	3.7 (11.1)
L2D2(25%)-Opt	38 (105)	3.8 (12.1)
L2D2(25%)-Random	53 (230)	7.0 (30.7)

Table 3: Scheduler Evaluation: Median (90th percentile)

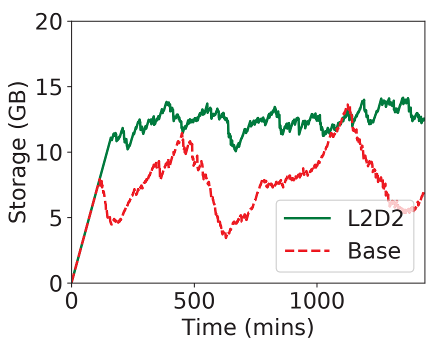


Figure 9: Median Storage on Satellite Over Time

prioritization of data for specific groups and tasks (i.e geographic regions, natural disasters, satellite bidding systems).

6.5 Satellite Storage

Recall that, in L2D2, data is stored on a satellite until a delayed acknowledgment is received from a TX-capable ground station. The acknowledgment ensures that if any packets are dropped, the satellite can later resend the data. We would like to better understand the storage requirement on the satellites in L2D2. Fig. 9 plots the median data storage across satellites through one day. On average, we see that L2D2's storage requirement is 4 GB higher than a centralized architecture where all the data is acknowledged right after download. Although the mean storage requirement for L2D2 is higher, it is not significant since even the baseline at its peak stores data similar to L2D2's requirement (at around 500 and 1200 mins in Fig. 9). This is because even though the centralized architecture acks the data right after download, the data downlink is infrequent in a centralized architecture.

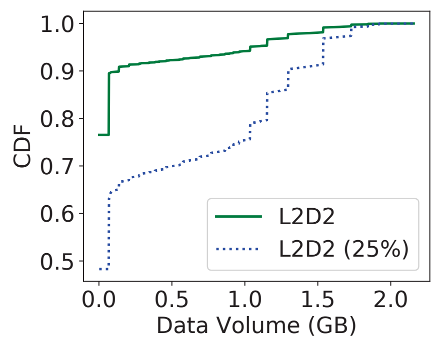


Figure 10: We plot the cdf of data downloaded across ground stations (segmented by minute).

6.6 Backhaul

In this paper, we focus on downloading data from satellites to ground stations. However, to generate insights for the end user, this data needs to be transported to the end applications operating in the cloud or at other end hosts. Given the quantity of the data, it is non-trivial to transport this data to end users and may incur more delays. We leave a detailed analysis of this delay to future work, but present a brief analysis of the backhaul requirements here.

To understand the backhaul requirements from ground stations to the cloud, we analyze the data downloaded in one-minute intervals at each ground station. We plot the distribution of data downloaded in each minute across all ground stations in Fig. 10. In L2D2, the median data download volume is 0 GB (90-th percentile: 0.14 GB, 99-th percentile: 1.536 GB). In L2D2(25%), the ground stations are more busy and the median data download volume is 0.07 GB (90-th percentile: 1.2 GB, 99-th percentile: 1.72 GB) in a minute. Supporting real-time upload to the cloud from these ground stations for the 99-th percentile would require nearly 30 MBps upload capacity. In contrast, for the baseline with five high capacity ground stations (cdf not shown in the figure for emphasis on the L2D2 graph), the median data downloaded is 0.2 GB (90-th percentile: 12.96 GB, 99-th percentile: 15.6 GB) in a minute. Supporting real-time upload from these ground stations would require 260 MBps upload capacity.

There are two interesting insights from this data. First, the traffic at ground stations is sparse and peaky. The median download volume is low because satellite-ground station links aren't always active. However, the peaks are relatively high – this happens when the satellite-ground station link is strong (for example, when the satellite is overhead). Second, as expected, the requirement for internet connectivity is higher for high-capacity ground stations. This is non-trivial to achieve, especially in polar regions or sparsely populated areas where ground stations are likely to be located. Based on this analysis, we believe the earth imaging application is ideally suited for edge computing at the ground station. An edge computing architecture could prioritize data upload from the ground station, thereby satisfying the demands of the latency-sensitive applications, while using the 'quiet' periods at the ground station to upload the less urgent data. We leave this exploration to future work.

7 RELATED WORK

The problem of downlink constraints on new constellations of LEO satellites has been tackled in both academia [17–19, 54] and industry [1, 20, 34, 40, 46]. A large part of this work focuses on eking out more performance from individual satellite-ground station links. Such a design suffers from the limitations of a centralized architecture described before. Recently, [17] proposed 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, L2D2 downlinks all the data to the ground using a hybrid ground station design.

In the industry, multiple efforts [1, 36, 40, 46] 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 but suffers from similar regulatory and equipment challenges as centralized architectures [27]. However, this investment opens up the possibility of new abstractions like distributed ground station architectures in the future. In L2D2, we investigate the tools that will be required for such a distributed design. VERGE [43] is perhaps the closest design to L2D2. In [43], 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, L2D2 co-locates compute alongside the antenna and the decoded & processed data is sent to the cloud. This significantly reduces the backhaul capacity and cost 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 [43] is limited to lower bandwidth S-band downloads, as opposed to X-band downloads that are common for earth observation.

The scheduling problem for satellite-ground station links has been tackled in [8, 9, 25, 55, 58]. These systems do not account for varying link quality over time and/or limit themselves to single satellite and multiple ground stations. In contrast, L2D2 presents a scheduler for multi-satellite, multi-ground station configuration while accounting for varying link qualities and switching delays.

Prior work on satellite-ground link quality estimation has mainly been carried out on simulated data that does not capture the complexities of real-world signals like reflections close to the transceiver [26]. Some research efforts incorporate real-world link quality measurements in their design exclusively with low-frequency links (UHF,S-band) [37, 49]. However, in the context of Earth observation satellite networking, L2D2's link estimation model based on X-band data is more applicable since Earth observation satellites more commonly operate in this high frequency range and such links are more prone to weather-effects. L2D2 also outperforms prior statistical models for link quality prediction [31–33, 37]. L2D2 achieves this by using a data-driven approach that accounts for multipath effects and occlusions.

Finally, L2D2 is inspired by past work in open source ground station designs [14, 41] and deployments of these stations [15, 41, 48, 53]. These deployments have fostered research in scheduling, mission control, and other aspects of ground station design [3, 55, 56, 63]. Most of these designs are limited to low frequency, low data rate regimes for experimental satellites that transmit small amounts of data. In L2D2, we differ along three axes: distributed design framework, high frequency and high bandwidth data downloads, mix of transmit-capable and receive-only ground stations.

We note that L2D2 builds on a previous workshop paper [60] and differs along three axes: (a) new scheduling framework that accounts for switching delays, (b) new data-driven link estimation algorithm, and (c) extensive evaluation on a real-world dataset.

8 CONCLUSION

We present L2D2, a novel ground station design for Earth observation satellites. L2D2 is built using a mix of transmit-capable and receive-only geographically-distributed ground stations. L2D2's geographical diversity along with a new scheduling and rate adaptation algorithm ensure that L2D2 can download data from large constellations of LEO satellites in minutes, as opposed to hours for traditional architectures. While we design L2D2 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 3.4, 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.

ETHICS STATEMENT

This work does not raise any ethical issues.

ACKNOWLEDGMENTS

We thank anonymous reviewers and our shepherd, Barath Raghavan, for their valuable feedback. We are grateful to Peder Olsen for the multiple brainstorming sessions about this research and to Anirudh Badam, Anuj Kalia, Haitham Hassanieh, Romit Roy Choudhury, Yu Tao, and Zikun Liu for providing feedback on early versions of this paper. Jayanth Shenoy is funded by the NSF Graduate Research Fellowship Program under Grant No. DGE–1746047.

REFERENCES

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

[2] Bruno Aragon, Rasmus Houborg, Kevin Tu, Joshua B. Fisher, and Matthew McCabe. 2018. CubeSats Enable High Spatiotemporal Retrievals of Crop-Water Use for Precision Agriculture. *Remote Sensing* (2018).

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

[4] Panagiotis Barmpoutis, Periklis Papaioannou, Kosmas Dimitropoulos, and Nikos Grammalidis. 2020. A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing. *Sensors* (2020).

[5] A. Bhattachan, N. Skaff, S. Vimal, J. Remais, and D. P. Lettenmaier. 2019. Using geospatial datasets to characterize mosquito larval habitats in the Los Angeles Basin. In *AGU Fall Meeting Abstracts.*

[6] Stefania Bonafoni, Sujan Shrestha, and Dong-You Choi. 2017. Characterization of Rain Specific Attenuation and Frequency Scaling Method for Satellite Communication in South Korea. *International Journal of Antennas and Propagation* (2017).

[7] Capella Space. [n. d.]. Capella Space. https://www.capellaspace.com/. ([n. d.]).

[8] Rogerio Atem de Carvalho. 2019. Optimizing the Communication Capacity of a Ground Station Network. *Journal of Aerospace Technology and Management* (2019).

[9] Jeremy Castaing. 2014. Scheduling Downloads for Multi-Satellite, Multi-Ground Station Missions *(Small Satellite Conference).*

[10] G. Celik, S. C. Borst, P. A. Whiting, and E. Modiano. 2016. Dynamic scheduling with reconfiguration delays. *Queueing Systems* 83, 1 (2016), 87–129.

[11] Kejie Chen, Jean-Philippe Avouac, Saif Aati, Chris Milliner, Fu Zheng, and Chuang Shi. 2020. Cascading and pulse-like ruptures during the 2019 Ridgecrest earthquakes in the Eastern California Shear Zone. *Nature Communications* (2020).

[12] Tianqi Chen and Carlos Guestrin. 2016. XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug 2016). https://doi.org/10.1145/2939672.2939785

[13] Kyle Colton and Bryan Klofas. 2016. Supporting the Flock: Building a Ground Station Network for Autonomy and Reliability *(Small Satellite Conference).*

[14] James Cutler. 2003. Ground Station Virtualization *(The Fifth International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations).*

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

[16] Dark Sky. [n. d.]. Dark Sky Weather API. https://darksky.net/dev. ([n. d.]).

[17] Bradley Denby and Brandon Lucia. 2020. Orbital Edge Computing: Nanosatellite Constellations as a New Class of Computer System *(ACM ASPLOS).*

[18] Kiruthika Devaraj, Ryan Kingsbury, Matt Ligon, Joseph Breu, Vivek Vittaldev, Bryan Klofas, Patrick Yeon, and Kyle Colton. 2017. Dove High Speed Downlink System *(Small Satellite Conference).*

[19] Kiruthika Devaraj, Matt Ligon, Eric Blossom, Joseph Breu, Bryan Klofas, Kyle Colton, and Ryan Kingsbury. 2019. Planet High Speed Radio: Crossing Gbps from a 3U Cubesat *(Small Satellite Conference).*

[20] Anna Escher. 2018. Inside Planet Labs' new satellite manufacturing site. TechCrunch. https://techcrunch.com/2018/09/14/inside-planet-labs-new-satellite-manufacturing-site/. (2018).

[21] Darrell Etherington. 2019. 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).

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

[23] European Space Agency. [n. d.]. Sentinel 6: Ground Segment Design . https://sentinel.esa.int/web/sentinel/missions/sentinel-6/ground-segment. ([n. d.]).

[24] Klaus-Tycho Foerster, Manya Ghobadi, and Stefan Schmid. 2018. Characterizing the Algorithmic Complexity of Reconfigurable Data Center Architectures. In *Proceedings of the 2018 Symposium on Architectures for Networking and Communications Systems (ANCS '18).*

[25] C. Fuchs and F. Moll. 2015. Ground station network optimization for space-to-ground optical communication links. *IEEE/OSA Journal of Optical Communications and Networking* (2015).

[26] K. Harb, F. Yu, and Samir Abdul-Jauwad. 2012. Performance Evolution in Satellite Communication Networks Along with Markovian Channel Prediction.

[27] Mark Harris. 2019. Is Amazon's Satellite Ground Station Service Ready for Primetime? . IEEE Spectrum. (2019).

[28] Caleb Henry. 2019. Amazon moving Project Kuiper team to new R&D headquarters. SpaceNews. . https://spacenews.com/amazon-moving-project-kuiper-team-to-new-rd-headquarters/. (2019).

[29] Felix R. Hoots and Ronald L. Roehrich. 1980. *Models for Propagation of NORAD Element Sets.* Technical Report. Aerospace Defense Command, United States Airforce.

[30] ICEYE. [n. d.]. ICEYE: Every Square Meter, Every Hour. https://www.iceye.com/. ([n. d.]).

[31] International Telecommunications Union. 2019. *ITU P.838: Specific attenuation model for rain for use in prediction methods.* Technical Report.

[32] International Telecommunications Union. 2019. *ITU P.839 : Rain height model for prediction methods .* Technical Report.

[33] International Telecommunications Union. 2019. *ITU P.840: Attenuation due to clouds and fog.* Technical Report.

[34] Bryan Klofas. 2016. Planet Labs Ground Station Network *(Cubesat Developers Workshop).*

[35] J. Kong, Y. Ryu, R. Houborg, and M. Kang. 2019. Monitoring canopy photosynthesis in high spatial and temporal resolution using CubeSat imagery. In *AGU Fall Meeting Abstracts.*

[36] Kongsberg Satellite Services. [n. d.]. Ground Station Services. https://www.ksat.no/services/ground-station-services/. ([n. d.]).

[37] S. Krinninger, B. Preindl, Lars Mehnen, I. Klinkert, and Jens Dalsgaard Nielsen. 2010. Measuring and modeling link quality in ground station networks for small satellite operations. 4 (01 2010), 3280–3290.

[38] H. W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* (1955).

[39] Janardhan Kulkarni, Stefan Schmid, and Paweł Schmidt. 2020. Scheduling Opportunistic Links in Two-Tiered Reconfigurable Datacenters. (2020). arXiv:cs.DC/2010.07920

[40] Leaf Space. [n. d.]. Leaf Line: Shared Ground Station Network. https://leaf.space/leaf-line/. ([n. d.]).

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

[42] He Liu, Matthew K. Mukerjee, Conglong Li, Nicolas Feltman, George Papen, Stefan Savage, Srinivasan Seshan, Geoffrey M. Voelker, David G. Andersen, Michael Kaminsky, George Porter, and Alex C. Snoeren. 2015. Scheduling Techniques for Hybrid Circuit/Packet Networks *(ACM CoNEXT).*

[43] Lockheed Martin. 2018. 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).

[44] Jean-François Maystadt. 2019. Breaking Ground: Automating the Detection of Refugee Settlement Establishment and Growth through Landsat Time Series Analysis with a Case Study in Northern Uganda. (2019).

[45] Jean-François Maystadt, Valerie Mueller, Jamon Van Den Hoek, and Stijn van Weezel. 2020. Vegetation changes attributable to refugees in Africa coincide with agricultural deforestation. *Environmental Research Letters* (2020).

[46] Microsoft. [n. d.]. Azure Orbital. https://azure.microsoft.com/en-us/services/orbital/. ([n. d.]).

[47] James Munkres. 1957. ALGORITHMS FOR THE ASSIGNMENT AND TRANSPORTATION PROBLEMS. (1957).

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

[49] Jens Frederik Dalsgaard Nielsen, Bastian Preindl, Lars Mehnen, and F. Rattay. 2009. A Ground Station Link Quality Measurement Satellite. In *Proceedings of 2nd European CubeSat Workshop, ESA/ESTEC, Noordvjik, The Netherlands, 2009.*

[50] NOAA's Office of Satellite and Product Operations. 2020. 2020: Ka-band radio frequency received signal strength as measured by the Common Ground System (CGS) transmitted from the Joint Polar Satellite System-1 satellite during telemetry playback over McMurdo Station October 1-7. (2020).

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

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

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

[54] K. Riesing, H. Yoon, and K. Cahoy. 2017. A portable optical ground station for low-earth orbit satellite communications. In *IEEE International Conference on Space Optical Systems and Applications (ICSOS).*

[55] Marco Schmidt. 2011. *Ground Station Networks for Efficient Operation of Distributed Small Satellite Systems.* Ph.D. Dissertation.

[56] Marco Schmidt and Florian Zeiger. 2006. Design and Implementation of In Orbit Experiments for the Pico Satellite UWE-1". *International Astronautical Congress* (2006).

[57] Roy Schwartz, Mohit Singh, and Sina Yazdanbod. 2019. Online and Offline Greedy Algorithms for Routing with Switching Costs. (2019). arXiv:cs.DS/1905.02800

[58] Sara Spangelo, James Cutler, Kyle Gilson, and Amy Cohn. 2015. Optimization-based scheduling for the single-satellite, multi-ground station communication problem. *Computers & Operations Research* (2015).

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

[60] Deepak Vasisht and Ranveer Chandra. 2020. A Distributed and Hybrid Ground Station Network for Low Earth Orbit Satellites *(ACM HotNets).*

[61] Viasat. [n. d.]. HI-BEAM Transceiver DVB-S2 High-Speed Modem. https://medit-tech.it/PDF/hi-beam_modem_127_web_0.pdf. ([n. d.]).
[62] M. Wiseman and A. Bradley. 2019. Impact of the length of the sea ice-free summer season on Alaskan Arctic coastal erosion rates. In *AGU Fall Meeting Abstracts*.
[63] Nicolas Zufferey, Patrick Amstutz, and Philippe Giaccari. 2008. Graph colouring approaches for a satellite range scheduling problem . *Journal of Scheduling* (2008).