Spectrumize: Spectrum-efficient Satellite Networks for the Internet of Things

Vaibhav Singh, Tusher Chakraborty, Suraj Jog, Om Chabra[†], Deepak Vasisht[†], and Ranveer Chandra

Microsoft Research, [†]UIUC

Abstract

Low Earth Orbit satellite constellations are gaining traction for providing connectivity to low-power outdoor Internet of Things (IoT) devices. This is made possible by the development of low-cost, low-complexity pico-satellites that can be easily launched, offering global connectivity without the need for Earth-based gateways. In this paper, we report the space-to-Earth communication bottlenecks derived from our experience of deploying an IoT satellite. Specifically, we characterize the challenges posed by the low link budgets, satellite motion, and packet collisions. To address these challenges, we design a new class of techniques that use the Doppler shift caused by the satellite's motion as a unique signature for packet detection and decoding, even at low signal-to-noise ratios and in the presence of collisions. We integrate these techniques into our system, called Spectrumize, and evaluate its performance through both simulations and real-world deployments. Our evaluation shows that Spectrumize performs $3\times$ better compared to the classic approach in detecting packets with over 80% average accuracy in decoding.

1 Introduction

The emergence of the "new space" era has led to the growing popularity of small satellites aimed at connecting Internet of Things (IoT) devices. These satellites offer connectivity to devices where terrestrial networks do not exist, for example, in farms, forests, oceans, and others. With their small size and low-complexity hardware, these satellites are easy to build and launch into orbit, resulting in the rapid growth of the IoT satellite industry. Over a dozen companies [3–6, 33, 35] are investing heavily in deploying constellations of hundreds of these small IoT satellites to provide connectivity using the Direct-to-Satellite (DtS) model. This allows IoT devices on the ground to uplink data directly to orbiting satellites, without requiring deployment of gateway devices. The satellites collect and downlink data to ground stations which have terrestrial backhaul for cloud-based data aggregation.

The main focus of IoT satellites is to achieve a low price point, setting them apart from traditional earth observation and broadband connectivity satellites. These satellites are smaller in size and optimized for IoT applications, which typically require low data rates, usually in the range of hundreds of bits per second, and operate at a bandwidth of around a hundred kilohertz. Therefore, IoT satellites are equipped with simple connectivity hardware, such as half-wave dipole cable

(a) *Our satellite and 4 others from the same constellation*

(b) *Our industry collaborator's ground station*

Figure 1: *Our real-world testbed: an IoT satellite launched in collaboration with FOSSA [7], and multiple ground stations.*

antennas with no beamforming capability or inter-satellite link (ISL) [43]. Similarly, ground stations (GS) and IoT devices on the ground are designed to be simple and can be deployed anywhere using small antennas that are either omnidirectional or have limited directionality. In conclusion, the connectivity design and hardware of IoT satellites closely resemble that of state-of-the-art terrestrial IoT networks.

We collaborated with FOSSA Systems [7], an IoT satellite company, to launch an IoT pico-satellite and investigate the characteristics of this emerging network class. We utilized their ground stations in Spain to facilitate data downloads from the satellite. To our surprise, despite using standard hardware, we encountered significant packet loss during the downlink from the satellite to the ground station. Specifically, we found that 65.28% of the packets transmitted by the satellite were lost and remained undecoded by the ground station. This paper stems from the challenges of enabling a robust downlink from low-cost, low-power IoT satellites. In particular, we observe that the following three challenges limit our ability to decode downlink transfer from IoT satellites.

Challenge 1: Low link budget – The link budget for satellite to ground communication is low because of the large distance (around 500 km), limited directionality of antennas on satellites, ground stations without beamforming capabilities, and atmospheric attenuation. We observe an average link budget of only -133.78 dBm during a satellite pass over a ground

station. Increasing transmission power is not feasible due to power constraints on the satellite and regulatory restrictions. Though the use of a high-end setup like a phased array antenna on the ground can increase the link budget, it requires a large antenna setup and is expensive, especially for IoT satellites that operate on lower frequency bands like VHF and UHF. Lower link budget leads to frequent packet drops and a low data rate. It is particularly problematic for the downlink transmissions because ground stations also experience increased noise floor due to terrestrial transmissions.

Challenge 2: Packet collisions and spectral inefficiency – LEO IoT satellites have limited antenna directionality and can transmit signals across vast areas on Earth, referred to as their footprint. These footprints can span up to a million square kilometers of surface area. Since multiple satellites operate within the same frequency band, their transmissions collide at ground stations located within the overlapping footprints (Fig. 2). For example, consider the Swarm constellation consisting of 170+ satellites [9]. A ground station situated in a polar region, ideal due to the satellites' orbital dynamics, falls within the view of an average of 7 satellites simultaneously. Consequently, satellites may encounter a high collision rate when downloading data to the ground station. As a preemptive measure, ground stations are scheduled in advance to communicate with only one satellite, even when in the line of sight of multiple satellites. This scheduling strategy, while necessary, has a significant impact on spectral efficiency.

Challenge 3: Satellite motion – The satellite-ground link exhibits fluctuations in link quality due to two distinct types of motion: (a) the orbital motion of the satellite introduces variations in the distance between the ground device and the satellite, and (b) localized satellite motion, often characterized by tumbling. This phenomenon is particularly pronounced in small-sized IoT satellites, as they lack an altitude control system to stabilize their motion. While large broadband satellites tackle this issue through real-time bit-rate adaptation, it presents a more formidable challenge with IoT satellites due to their lower data rates and limited bandwidth. This challenge is especially evident in receive-only distributed ground stations employed for downloading data from the satellite [2, 48]. As a result, the satellite chooses conservative bit rates for transmission that can work even for the lowest signal-to-noise ratio (SNR). This leads to sub-optimal spectrum efficiency.

This paper presents Spectrumize, a novel approach designed to enhance the spectral efficiency of IoT satellite downlinks, thereby enhancing reliability, robustness, and scalability in satellite-based IoT networks. Spectrumize stands out by not treating satellite motion as a problem but as a unique enabler for satellite-based IoT networks. The core concept behind Spectrumize hinges on the predictability of satellite motion, which results in predictable Doppler shifts in the signal unique to each satellite. Consequently, we leverage the temporal variation of these Doppler shifts as signatures, akin

to codes in CDMA. Much like CDMA, Doppler signatures empower us to elevate SNR and demultiplex simultaneous transmissions, resulting in higher spectral efficiency.

SNR boosting: As mentioned previously, distinguishing the signal received from a satellite from noise at a ground station can be challenging. Traditional methods of packet detection using preamble-based correlation are often ineffective because of the signal's frequency shifts due to Doppler effects. We have made two key observations: first, Doppler shifts between satellites and ground stations are predictable due to stable orbital paths, and second, satellites typically transmit a periodic train of packets. To address this, Spectrumize employs a series of preambles, each adjusted with the appropriate Doppler correction, for correlation with the received signal to facilitate packet detection. In our empirical studies, we confirm that both Doppler shift correction and signal repetition are necessary for successful packet detection, even in the presence of hardware errors like carrier frequency offsets.

Collision resolution: Spectrumize introduces a new approach to address packet collision challenges in satellite IoT networks, particularly those caused by overlapping satellite footprints. This approach relies on three distinguishing factors: the unique Doppler shift of each satellite-ground link, the symbol reception time, and the received signal intensity. By leveraging the distinct Doppler shift patterns experienced by colliding packets from different satellites, we use the Doppler signature as a self-contained code to demultiplex signals from multiple transmissions. As a result, a ground station can simultaneously download data from multiple satellites, leading to a significant enhancement in spectral efficiency.

We conduct evaluations of Spectrumize through simulation, emulation, and real-world testbed, which encompasses our satellite in orbit. This testbed also involves other satellites from the same constellation and multiple ground stations provided by our industry collaborator, FOSSA. Utilizing this testbed, we characterize the nature of satellite-ground transmissions for IoT satellites and validate our design. The results demonstrate that Spectrumize outperforms state-of-the-art methods, achieving a $3\times$ enhancement in packet detection with an average decoding accuracy exceeding 80%.

Summary of Contributions:

- We characterize the communication challenges posed by the constraints in a deployed IoT satellite network.

- We propose a new 'Doppler signature as a code' approach to boost satellite decoding and demultiplexing performance.

- We evaluate Spectrumize in a real-world testbed including satellites in orbit.

2 Motivation from Real-world Experience

IoT satellites constitute a rapidly growing category of low-cost, low-complexity mini-satellites. As of 2022, approxi-
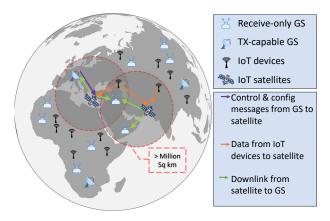
Figure 2: *Overview of end-to-end satellite IoT communication using Direct-to-Satellite technology.*

mately two thousand such satellites have been launched into orbit [8]. Swarm, FOSSA, Astrocast, EchoStar, Myriota, and more than ten other companies are leading this growing satellite IoT industry [3–6, 35]. To gain a better understanding of the industry practices, challenges, and the need for innovation in this burgeoning industry, we collaborated with an industry leader, FOSSA Systems [7], to launch an IoT satellite shown in Fig. 1a. Our collaborator designed and launched the satellite according to industry standards, without any design input from us. We utilized the satellite as a real-world testbed for our research along with the collaborator's ground station to communicate with the satellite, creating a testbed that accurately represents the ecosystem of today's satellite IoT industry. In this section, we delve deeper into the practices of the satellite IoT network industry and share our firsthand experiences with the challenges it presents.

2.1 Satellite as the Global IoT Gateway

In traditional terrestrial wide-area IoT networks such as LoRaWAN, IoT devices follow a star-of-stars topology with a gateway aggregating data from the IoT devices. In contrast, in satellite IoT networks, the IoT gateway is carried by the satellites themselves. This is achieved through direct-to-satellite (DtS) communication, where IoT devices on the earth send data directly to the overhead satellites. As the satellite orbits around the Earth, it can collect data from different locations and work as the global IoT gateway (Fig. 2). Similarly, our satellite carries a SX1302 LoRa gateway produced by Semtech [10]. The satellite has dimensions of around $10cm \times 5cm \times 5cm$ – this class of satellites is smaller than cube-satellites and is referred to as pico-satellite. Due to its small size, the satellite's power is restricted, with a maximum average power of 1.6 watts produced by its solar panel and stored in rechargeable batteries. To maintain low hardware complexity and cost-effectiveness, the satellite does not have any active altitude control system. It orbits Earth in a low earth orbit (LEO), typically between 450 to 550 kilometers

above Earth's surface, allowing it to observe different parts of the planet over time. Because of its orbital motion, the satellite can only communicate with ground devices for brief periods, typically lasting 5-9 minutes for a location on Earth. There are usually only 2-3 such communication windows per day for a location. As a result, a constellation of 150+ satellites is required to ensure 24/7 global coverage.

2.2 Ground Stations

To communicate with our satellite, we rely on two ground stations provided by our collaborator (Fig. 1b), which we use for downloading data and configuring the satellite through Telemetry, Tracking, and Command (TT&C) communication. To keep costs low, these ground stations use low-end connectivity hardware, including omnidirectional or directional Yagi antennas, rather than phased array antennas (Fig. 1b). The directional Yagi antennas have a beamwidth of 30° and a gain of 12 dBic, and they are coupled with rotators that can move them to point towards the satellite when it is in view. While we primarily use Yagi antennas, several other companies opt for omnidirectional antennas, which can further simplify their designs [2, 25]. The antenna is linked to a software-defined radio (SDR) to process the communication signal. The idea of using SDR-based ground stations is also popular in the ground station as a service industry [14, 29]. Furthermore, due to the short contact time and low data rate of the sat-ground communication link, there is currently a significant effort to develop receive-only distributed ground stations that can reduce data latency in satellite IoT networks [1, 2, 48]. For example, SatNOGS is an open-source global network of ground stations equipped with low-cost SDR and Raspberry Pi [1]. Our satellite also broadcasts health beacons targeting these distributed ground stations.

2.3 Communication

Our IoT satellite communicates with ground devices in three ways: 1) the satellite collects data from IoT devices, 2) the satellite communicates with ground stations for TT&C and downloading data to ground stations, and 3) the satellite sends data/beacons to distributed ground stations. In this section, we focus on communication between the satellite and ground stations. The satellite is equipped with a half-wave dipole cable antenna and does not have beamforming capability. It employs a Semtech SX1262 radio to communicate with the ground stations, which supports both LoRa and 2-FSK modulation [21]. We use the 401 - 402 MHz band assigned by the local regulators for Earth exploration satellites, meteorological aids, and meteorological satellites. The uplink communication (ground station to satellite) takes place on 401.3 MHz, while the downlink communication (satellite to ground station) occurs on 401.9 MHz. The satellite also sends a health beacon every 30-120 seconds on 401.7 MHz. The

channel bandwidth is max 125 kHz with LoRa modulation and 39 kHz with 2-FSK.

Before a satellite-ground station communication takes place, the contact is scheduled in advance using the satellite's two-line element (TLE) and the ground station's location [47]. During the contact, the communication is initiated by sending specific commands from the ground station to the satellite. Once the satellite receives a command, it relays it back to the ground station. The ground station confirms the command if it matches the one it sent, and then the command is executed. Depending on the command, the satellite either updates its onboard configuration or sends data to the ground station. When the satellite sends data, it does so in a stream of N data packets, where N is defined in the command. If the satellite does not have N packets to send, it sends all the packets that it has. Since the channel is not expected to be used for multiple satellite-ground communications at the same time, the packets are transmitted at a plain interval without any random back-off. A constant bit rate is maintained during communication, given the low bandwidth and short contact time.
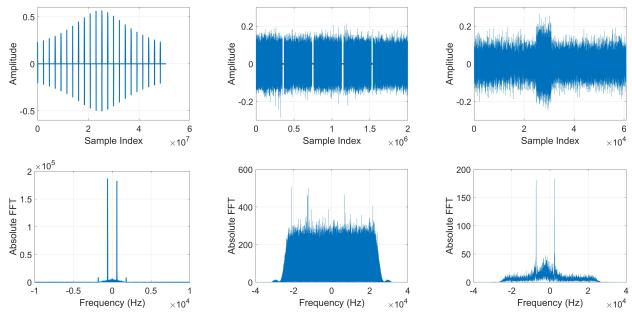
2.4 Our Experience and Challenges

Following the successful launch of our satellite, we commenced the commissioning phase which involves several stages such as initial orbit determination and tracking, first contact with the satellite, sequential activation and checks of spacecraft subsystems, and calibration of payloads, sensors, and control systems. Ideally, this phase should take about two weeks to complete. Unfortunately, we encountered some issues from the first contact with the satellite which extended the duration of the commissioning phase by several weeks. It was persistent across other satellites launched together. The issue primarily appeared to be related to the downlink communication, prompting us to conduct further investigations to identify the root cause.

The expected signal from our satellite using 2-FSK modulation can be seen in Fig. 3a(top) in the time domain, which was emulated in the lab using our satellite's RF setup. When we transfer the signal in the frequency domain using Fast Fourier Transformation (FFT) as shown in Fig. 3a(bottom), we see two distinct peaks with the perfect frequency deviation configured in 2-FSK modulation. In Fig. 3b, we can see the signal recorded by the SDR of our ground station during a satellite pass of approximately 10 minutes. According to the config of our satellite's regular beacon transmission, we expect to receive 10-14 packets. However, in the received signal shown in Fig. 3b, we do not see any clear signs of the packets as we saw in the emulated signal (Fig. 3a). When we transfer the received signal to the frequency domain using FFT, as shown in Fig. 3b(bottom), we do not see any clear peaks corresponding to 2-FSK modulated packets. It is apparent that the packets are buried under the heavy noise and interference on the ground. Nevertheless, we attempted to run a classic

preamble-based packet detection by running correlation between the received signal and the known packet preamble. Ideally, we should have seen a prominent correlation peak corresponding to the beginning of each packet in the signal, translating to 10-14 peaks in our case. However, as shown in Fig. 4, we hardly see a prominent correlation peak. We then conducted a manual inspection of the received signal in search of the packets and found 12 packets in the signal, as shown in Fig. 3c after zooming in on the signal. We ran FFT on this part of the signal and found clear 2-FSK peaks in the frequency domain, confirming that the packet was transmitted from the satellite. In aggregate, the SNR of the received signal was so low that the classic preamble correlation-based technique failed in detecting packets. This experience led us to dig deeper into the reasoning behind our experience.

To begin with, the link between the satellite and ground station has a low link budget due to several factors, such as the omnidirectional radiation pattern of the satellite's antenna, the long distance between the two (hundreds of kilometers), the satellite's power limitation, and regulatory restrictions on the power flux density of satellite transmission on the ground. For instance, in the case discussed, the link budget is -112 dBm with the satellite's transmission EIRP of 22 dBm, which is comparable to other satellite service providers like Swarm reporting -108.24 dBm [43]. However, these values are calculated by assuming an elevation angle of $50°$. The elevation angle has a direct impact on path loss and consequently, the link budget. In other words, a lower elevation angle means the satellite is closer to the horizon, resulting in a lower link budget. Using data from the TinyGS, an opensource distributed ground station network for IoT satellites, over a six-week period, we found that the average elevation angle across the ground stations is $26.68°$ indicating that the link budget is lower in the real world than the value calculated for nominal elevation angle of $50°$. Furthermore, the link budget in reality is even lower due to environmental factors such as weather and surrounding infrastructure that cause signal attenuation. The average RSSI across the TinyGS ground stations was found to be -133.78 dBm, which is -128.12 dBm for our ground stations. Note that this data had packet survival bias since we have the RSSI values for only the detected packets, whereas, the packets having a RSSI below the minimum detectable intensity (MDI) of ground station radio or very poor SNR values, are not detected and hence not recorded. Additionally, we noticed that the rotation of the satellite around its own axis, known as tumbling, leads to the antenna depointing, resulting in decreased signal strength.

The ability to detect and decode packets despite having an RSSI above the MDI depends on the level of noise and interference at the ground station, which can originate from both terrestrial and non-terrestrial sources. Interference from other satellite constellations is common since IoT satellites often use a band designated for Earth exploration, meteorological, and mobile satellites. In addition, interference from

(a) *Emulated signal containing 2-FSK packets transmitted by satellite. Time domain (top) and frequency domain (bottom).*

(b) *Signal received at the ground station in real-world. Time domain (top) and frequency domain (bottom).*

(c) *Packet found after manual inspection by zooming in the signal. Time domain (top) and frequency domain (bottom).*

Figure 3: *Our experience in processing signal received from our satellite at the collaborator's ground station.*
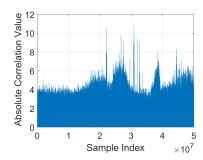

Figure 4: *Packet detection from a real-world signal using single preamble correlation. Results: no clear correlation peak at the beginnings of packets.*

terrestrial communication can significantly affect the satellite signal due to the low link budget, and operation in the ISM band can make them susceptible to direct terrestrial interference [15, 28, 31, 59]. Besides, the use of low-cost and low-complexity hardware in ground stations also contributes to the noise. As a combined effect of these factors, the SNR value gets extremely poor and becomes negative [19]. To highlight the extent of the poor SNR values, the average SNR at our ground station locations was found to be as low as −18 dB, while it was −17.11 dB across TinyGS ground stations.

In addition to external interference, IoT satellites also face interference from other satellites within their own constellation. This is due to the omnidirectional radiation pattern of a satellite's antenna, which results in a footprint that can cover millions of square kilometers. The footprints of adjacent satellites from the same constellation overlap with each other as

shown in Fig. 2. As mentioned earlier, ground stations are strategically placed in locations where they can maximize the number of satellite passes, such as the polar regions. With over 170 IoT satellites in a constellation like Swarm, a ground station in such an optimal location spends 83% of its time in the overlapping footprint of two or more satellites. At any given time, the ground station can see up to 23 satellites with a median of 4 (see Appendix A). Now, the satellites having overlapped footprints create interference with each other as they transmit to a ground station in an overlapping region. As a solution, when downloading data from a satellite at a ground station, only one satellite is scheduled for communication at a time, despite multiple satellites being within view. The ground station sends a command to the scheduled satellite to initiate data transfer while keeping the others silent. This approach, while addressing interference issues, results in significantly reduced spectral efficiency. This impact is even more pronounced for ground stations located in polar regions, which are almost all the time in overlapping footprints of multiple satellites.

3 Spectrumize's Algorithm

In this section, we present our technique for detecting and decoding the packets received from IoT satellites that are severely buried under the noise floor (Fig. 3b (top)). We found that conventional technique of correlating with a single preamble exhibits poor performance (Fig. 4), where the peaks corresponding to packet start time are obscured by the
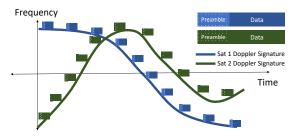
Figure 5: *Unique Doppler Signatures allow us to disentangle the collisions from two different satellites at a ground station.*

overwhelming noise.

To enhance the prominence of the peaks associated with packet start times, it is crucial to increase the correlation score of the genuine packet within the noisy received signal. To do so, we leverage two unique observations specific to the IoT satellite domain.

1. Downlink from satellite to ground stations are contention-free transmissions: Traditional wireless networks, such as WiFi and cellular, operate on a contention-based system where transmitters employ carrier sense before transmitting packets. Consequently, packet transmissions in traditional wireless networks do not typically adhere to a periodic or predetermined pattern. However, the dynamics in IoT satellite networks differ. Techniques like carrier sense are ineffective due to the vast footprint of satellites on the Earth's surface, making it possible for distant satellites to inadvertently collide at a ground station. As a result, the current industry standard for deploying downlink communication protocols from IoT satellites to ground stations adheres to a contention-free schedule.

More specifically, the IoT satellite network employs a centralized downlink scheduler that assigns ground stations to receive packets from specific satellites during each satellite's contact period with the ground station. The IoT satellite receives this transmission schedule in advance, along with commands that specify the number of packets, denoted as N, that it can transmit. During the satellite's contact period with its designated ground stations, it transmits a train of N data packets, each periodically spaced. It's essential to note that because the satellite does not perform carrier sensing before transmitting data packets, its packet transmissions follow a deterministic periodic pattern, as illustrated in Fig. 5. The key insight of Spectrumize is that this known periodic sequence of N packets can be harnessed to construct a longer virtual *preamble train*. This virtual preamble train serves to enhance the correlation score of the target signal, elevating it above the noise floor and facilitating the detection of packets.

2. Orbital motion of satellites contribute unique Doppler signatures at the physical layer: The second pivotal insight underpinning Spectrumize revolves around the exploitation of the orbital motion of the satellite during its contact period with a ground station, which introduces a distinctive Doppler signature unique to each satellite-ground station link. This

signature is influenced by factors such as the satellite's position and velocity w.r.t. the stationary ground station. Fig 5 provides an example of two satellites, each characterized by its unique Doppler signature. These Doppler signatures effectively introduce an additional layer of physical modulation over the existing data packets transmitted. In essence, this process is akin to the codes used in CDMA. Similarly, we can harness this well-known temporally varying Doppler signature of the satellite-ground station link to enhance signal SNR and to demultiplex simultaneous transmissions.

We next detail our packet detection and collision resolution algorithm, followed by techniques to address hardware imperfections like CFO and time jitter in the packet sequence.

3.1 Packet Detection

Following the transmit schedule and command from the ground station, the satellite transmits N data packets to the assigned ground station during its contact period. Each individual data packet is preceded by a preamble to help with the detection of packet start time and subsequent decoding. Let the preamble signal be denoted by $p(t)$. Then the sequence of preambles $s(t)$ transmitted by the satellite over the course of the N data packets is:

$$s(t) = \sum_{k=0}^{N-1} h_k\ p(t-k\tau) + n(t) \qquad (1)$$

where τ represents the inter-packet interval at which the satellite transmits its data packets. The wireless channel experienced by the kth data packet, denoted as h_k, and the received signal's noise, $n(t)$, comprise both thermal noise and external interference originating from terrestrial source or other satellite transmissions.

Traditional packet detection methods would attempt to find the packet start times by correlating the received signal $s(t)$ with the known preamble $p(t)$. However, in our real-world experiments, due to the overwhelming amount of noise, we could not detect any packet start times reliably. To address this challenge, Spectrumize's insight is that we can leverage the known periodicity of the transmitted signal and construct a longer virtual preamble train to correlate with the received signal. This, in turn, would allow us to boost the correlation score and would raise the peaks corresponding to the packet start times above the noise floor.

Specifically, the simulated preamble train $\tilde{p}(t)$ that we construct to correlate with the received signal $s(t)$ is:

$$\tilde{p}(t) = \sum_{k=0}^{N-1} p(t-k\tau) \qquad (2)$$

where $p(t)$ is the known preamble structure. So given the virtual preamble train, our packet detection method works by correlating $s(t)$ with $\tilde{p}(t)$. That is, we want to find

$$arg\ max_{t'}\ |s(t)\ \tilde{p}^*(t-t')| \qquad (3)$$

where $\tilde{p}^*$ is the complex conjugate of $\tilde{p}$. The time t' that

maximizes the above correlation would be when the preamble trains in the two signals $s(t)$ and $\tilde{p}(t)$ perfectly align. So when the two trains align, the value of the correlation score is

$$Corr\_Score = \sum_{k=0}^{N-1} h_k \left|\left|p(t)\right|\right|^2 \qquad (4)$$

As observed in Eq. 4, the utilization of a preamble train for correlation results in the aggregation of correlation terms from each preamble in the train. This aggregation allows the peak corresponding to the packet start times to surpass the noise floor. It's important to note, however, that the terms in the summation do not add up constructively due to the presence of the channel term h_k. The channel h_k introduces a random phase and gain variation to each term in the summation. While coherent addition of the terms in the summation would be the ideal scenario for packet detection, we still get significant gains from non-coherent combination alone. In Appendix B, we present the proof for the following Lemma.

Lemma 3.1 *Consider $a_1, a_2, \ldots, a_n$ to be complex numbers on the unit circle with random phase. That is, $a_i = e^{j\theta_i}$ where θ_i is uniformly sampled from $[0, 2\pi]$. Let s_n be defined as $s_n = \sum_{i=1}^{n} a_i$. Then we have $\mathbb{E}\left[\|s_n\|^2\right] = O(n)$.*

The above Lemma 3.1 demonstrates that the gains from non-coherent combining grows monotonically with the number of terms in the non-coherent summation. Hence, leveraging a preamble train allows us to detect the packets buried under the noise floor, and this was not possible with single preamble-based correlation techniques. This observation also aligns with the rich body of literature on non-coherent combining and beamforming of wireless signals [32, 37, 50, 51, 58].

Up to this point, we have discussed how leveraging a virtual preamble train can enhance the SNR and enable the detection of packet start times. However, this alone is not sufficient. In addition to the wireless channel's contribution represented by h_k, it is crucial to consider the presence of a Doppler signature component. As previously discussed, the orbital motion of the satellite results in a unique Doppler signature for each satellite-ground station link, which overlays the data transmissions from the satellite. This time-varying Doppler signature is denoted as $f_d(t)$. As illustrated in Fig. 5, the Doppler signatures for two satellites are depicted in green and blue. This signature essentially introduces frequency modulation on top of the sequence of data packets transmitted by each satellite. Given that the Doppler signature differs for each satellite-ground station link, we can liken them to CDMA codes. We can leverage these Doppler signatures to further enhance the SNR of the desired signal.

Specifically, we can modify our simulated preamble train $\tilde{p}(t)$ as follows

$$\tilde{p}(t) = \sum_{k=0}^{N-1} p(t - k\tau)\, e^{j2\pi f_d(t)t} \qquad (5)$$

Note that the actual received signal $s(t)$ from the satellite already has the Doppler shift encoded in it, so in the correla-

tion equation shown in Eq. 4, when we plug in our modified $\tilde{p}(t)$ from Eq. 5, the Doppler signature terms will cancel and our correlation score is going to remain the same as in Eq. 4. Hence, correcting for this Doppler signature $f_d(t)$ is essential for the correlation across preambles from different packets to add up constructively.

Packet separation: We would like to emphasize that the term $e^{j2\pi f_d(t)t}$ effectively functions as a time-varying code in this context, aiding in the separation of packets between transmissions from terrestrial sources or other satellites operating in the same frequency band. To illustrate this, let's consider the reception of transmissions from two satellites, as depicted in Fig. 5, with their respective Doppler signatures denoted as $f_d^1(t)$ and $f_d^2(t)$. Without loss of generality, let's focus on detecting the start times of packets from satellite 1, while considering packets from satellite 2 as interference. For satellite 1, employing the modified preamble train shown in Eq. 5, our correlation score achieves its highest value when the simulated preamble train with the Doppler signature aligns precisely with the train of packets present in the transmitted signal from satellite 1.

However, when considering the transmissions from satellite 2 that cause interference at the ground station, the correlation between our simulated preamble train and the signal from satellite 2 diminishes. This is because the Doppler signature encoded in satellite 2's transmissions differs from the Doppler signature we are seeking in the received signal. Specifically, the correlation score for satellite 2's transmission will be:

$$Corr\_Score \sim \sum_{k=0}^{N-1} h_k \left|\left|p(t)\right|\right|^2 e^{j2\pi(f_d^2(t) - f_d^1(t))t} \qquad (6)$$

Hence, as evident from the above equation, the correlation terms from different packets in the train no longer add up constructively. This occurs because the value of the term $(f_d^2(t) - f_d^1(t))$ varies with time, contributing different phase offsets to each term in the summation. Consequently, by harnessing the unique Doppler signature for each satellite, we can further effectively suppress transmissions from other sources.

3.2 Hardware Imperfections

However, it is important to note that we have not yet accounted for hardware imperfections in our technique. Due to these hardware imperfections and the lack of perfect synchronization, there will be disparities between the real signal $s(t)$ and our simulated preamble train $\tilde{p}(t)$. More specifically, the received signal $s(t)$ will exhibit a Carrier Frequency Offset (CFO) component, causing it to deviate from the simulated preamble train $\tilde{p}(t)$. Additionally, our assumption of a uniform periodic train of data packets every τ seconds may not hold true in practical scenarios. Processing delays through different layers of the network stack often introduce some jitter, causing the packets in the train not to be transmitted at precisely τ-second intervals.

It is imperative that we consider the impact of these hardware imperfections when creating the virtual preamble train used for detecting the presence and start times of packets. However, standard channel-based techniques are inadequate for estimating these offsets, as we must first locate the packet to measure these channels. Consequently, we begin with a set of initial values and iterate on these values until we observe the peaks from the data packets rising above the noise floor. Specifically, we construct the virtual preamble train with a CFO value of Δf and jitter denoted as ε. In this scenario, our modified virtual preamble takes the following form –

$$\tilde{p}(t) = \sum_{k=0}^{N-1} p(t - k(\tau + \varepsilon)) \, e^{j2\pi f_d(t)t} \, e^{j2\pi\Delta ft} \qquad (7)$$

The above equation for the preamble train takes into account the unique Doppler signature $f_d(t)$, the Carrier Frequency Offset (CFO) Δf, and the jitter ε in the periodic train. It is worth noting that the jitter may result in non-uniformly spaced packets, rather than just a uniformly perturbed periodicity of ε. However, for the sake of maintaining manageable computational complexity, we assume only a jitter in the periodicity of the train. Our experiments have demonstrated that this approach is sufficient to elevate the correlation score above the noise floor.

The algorithm operates as follows: We start by selecting a set of candidate CFO values $\{\Delta f_1, \ldots, \Delta f_x\}$ and a set of candidate jitter values $\{\varepsilon_1, \ldots, \varepsilon_y\}$. It is important to note that the Doppler signature is known, thanks to the TLE information from the satellite. For each pair of CFO and jitter values, we create a virtual preamble train to correlate with the received signal $s(t)$. From the collection of candidate preamble trains, we identify the CFO value Δf^* and jitter value ε^* that maximize the correlation score with $s(t)$. These chosen values of CFO and jitter may not be perfectly accurate, but our objective is to select values that are *close enough* to elevate the correlation score above the noise floor. This approach enables us to determine the position of the first packet in the signal. However, what about subsequent packets?

According to Eq.4, the correlation peaks for subsequent packets decrease as we decode and discard packets from the signal. It causes the peaks of trailing packets in the train to become increasingly obscured by noise. This makes detection as challenging as single-preamble-based detection for these trailing packets, as Eq.4 loses its summation advantage. To tackle this issue, we must restore the summation advantage by reconstructing the preamble train. To achieve this, following the initial packet detection and decoding, we remove the corresponding portion in the signal and append it to the end of the signal. We apply the same process to the virtual preamble train, with the second packet now becoming the head of the train. This procedure continues until we successfully decode all the packets in a circular queue fashion. This approach ensures the train's length is maintained, allowing us to preserve the summation advantage in Eq. 4.

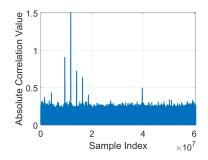## 3.3 Collision Resolution for Packet Decoding

As previously mentioned, ground stations frequently find themselves within the overlapping footprints of multiple satellites due to the large coverage areas of these satellites. Although they have the potential to communicate with several satellites, current scheduling practices dictate that only one satellite can be active at a time. This precaution is taken to prevent collisions between transmissions from satellites within the same constellation, but it comes at the cost of poor spectral efficiency. As a remedy, Spectrumize facilitates simultaneous transmissions in the same frequency band from multiple satellites within the same constellation.
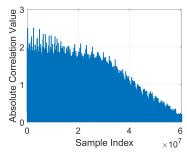
Spectrumize efficiently separates and decodes data symbols from the intended satellite while filtering out symbols from collided packets of other satellites. This capability is made possible by the distinct Doppler signature unique to each satellite-ground station link. Each packet, within its duration, carries a Doppler signature that differs significantly among satellites. In essence, each packet can be seen as encoded with a unique physical layer code defined by its Doppler signature, as demonstrated in our results. Multiplying the combined signals of collided packets by the complex conjugate of the relevant Doppler signature ($f_d^*(t)$) allows us to enhance the SNR of the desired packet's data symbols while attenuating data symbols from interfering packets. This approach leverages Doppler signatures in a manner akin to CDMA codes. Following this SNR enhancement, we employ a standard LoRa collision resolution algorithm to further isolate the symbols corresponding to the target packet, effectively eliminating symbols from interfering packets.
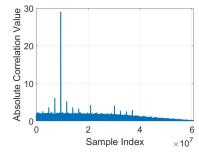
We focus on LoRa modulation for our packet decoding use case, which is prevalent in the satellite IoT industry [6, 7, 43]. Our satellite employs LoRa modulation along with 2-FSK, and the techniques we employ can also be extended to 2-FSK. Numerous studies have delved into decoding collided LoRa packets [24, 45, 46, 52, 53]. In our work, we employ the CIC algorithm [38] for decoding collided packets. It is important to note that merely applying the CIC algorithm [38] to the received signal $s(t)$ without prior Doppler signature correction, as demonstrated in Section 4, is ineffective, especially for long packets. The correction of the Doppler signature is a prerequisite for successfully decoding collided packets and extracting the desired data symbols.

4 Evaluation

We evaluate Spectrumize's performance in both simulation and real-world environments. In simulation and benchtop emulator setup, we assess the micro-benchmarks and scalability of Spectrumize, while in the real-world testbed, we evaluate its performance in actual operation settings.

(a) *Single preamble with Doppler signature*   (b) *Preamble train without Doppler*   (c) *Preamble train with Doppler signature*

Figure 6: *Micro-benchmark of packet detection techniques: a single preamble or a train without the Doppler signature suffering from large noise. A preamble train with the Doppler signature applied exhibits a 13dB gain in correlation detection power.*

4.1 Real-world Testbed

Our real-world testbed consists of our own satellite that is launched in collaboration with FOSSA Systems, along with two ground stations operated by the same collaborator, as described in Section 2. While most of the experiments are conducted using our own satellite, we also use other satellites in the constellation for certain experiments, such as those involving packet collisions. At the ground stations, we collect raw SDR recordings of satellite passes, as well as the SNR and RSSI values of the packets that are decoded by the Semtech SX126x radio. This radio is connected to the receive chain of the ground station through an RF splitter. Additionally, we utilize the Microsoft Azure Orbital ground stations to receive signals from our satellite [29].

4.2 Simulator and Benchtop Emulator Setup

For simulations, we employ MATLAB, leveraging the Communications Toolbox for 2-FSK modulation and LoRaMatlab for LoRa modulation [12]. To bring the evaluation closer to real-world conditions, we also establish a benchtop emulator setup. In this benchtop setup, we utilize three SDRs, specifically USRP B200 [11], integrated with GNU Radio and tuned to the satellite frequency plan. Two of these SDRs replicate the transmitters of two distinct satellites, while the third serves as the ground station. To introduce the Doppler effect, we calculate the Doppler signature for a satellite pass over our actual ground station location using the corresponding satellite's TLE. Subsequently, we multiply this Doppler signature with the transmission signal. It's important to note that, similar to the real-world testbed, there is no clock synchronization across the SDRs.

4.3 Packet Detection

In order to assess the effectiveness of Spectrumize's packet detection technique, we initially conduct a series of micro-benchmark tests to determine how different parameters impact our approach, as detailed in Section 3. Subsequently, we apply our method to the data collected from the real-world testbed, and compare its performance with the conventional technique.

4.3.1 Micro-benchmark

We first conduct the benchtop emulation to examine the impact of different correlation setups. We take a scenario where a satellite transmits 2-GFSK packets at a 9600 baud rate, every 30 seconds, with Doppler variation and Additive White Gaussian Noise (AWGN) based on real-world measurements. Our baseline approach employs classic preamble-based correlation for packet detection in the received signal. However, we encounter a lack of prominent correlation peaks at the start of packets, as illustrated in Fig. 4. To improve the correlation results, we correlate the signal with a single preamble multiplied by the Doppler signature. This results in marginally improved correlation outcomes where the peak remains relatively subtle in terms of correlation magnitude as shown in Fig. 6a. Subsequently, we experiment with correlating the signal using a preamble train aligned with the satellite's packet transmission sequence but without Doppler correction. This approach yields subpar results, with indistinct peaks that are challenging to distinguish, as demonstrated in Fig. 6b. Finally, we implement our proposed method, which involves correlating the signal with a preamble train multiplied by the Doppler signature. This approach demonstrates a significant improvement, showcasing a gain of approximately 13dB in correlation detection power, as depicted in Fig. 6c. In certain scenarios, we achieve an even higher gain of up to 25dB demonstrating the effectiveness of our technique in detecting packets despite heavy noise.

After examining our technique's performance under ideal conditions, we then investigate its robustness against real-world factors such as hardware imperfections. We begin by introducing a CFO of 100 Hz to the transmitted packets. When we use our correlation technique to detect these packets, we observe a decrease in the correlation peak (Fig. 7a), but it is still very prominent compared to the baseline results shown in Fig. 4. Next, we consider an extreme case of Doppler esti-
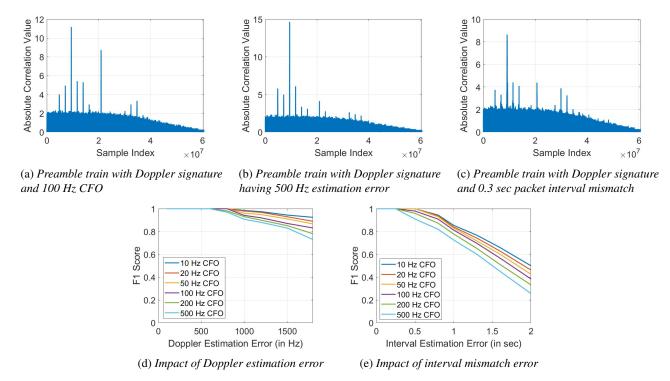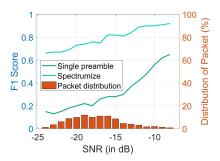
(a) *Preamble train with Doppler signature and 100 Hz CFO*

(b) *Preamble train with Doppler signature having 500 Hz estimation error*

(c) *Preamble train with Doppler signature and 0.3 sec packet interval mismatch*

(d) *Impact of Doppler estimation error*

(e) *Impact of interval mismatch error*

Figure 7: *Robustness of packet detection to errors: our packet detection is robust to timing or frequency mismatch.*

mation error, 500 Hz. Although the resulting correlation peak is shorter (Fig. 7b) than the ideal case, it is still very prominent. We then study the impact of packet interval mismatch by adding a 0.3 sec delay to each preamble in our preamble train. As shown in Fig. 7c, this factor has a negative impact on our performance. However, we still obtain a distinguishable correlation peak that is much higher than the baseline. For further investigation of the impact of these factors, we try out a range of values. The results are presented in Fig. 7d and 7e. In Fig. 7d, we demonstrate the effect of Doppler estimation error and CFO on the F1 score for packet detection. We observe that our technique can tolerate a Doppler estimation error of up to 1 kHz, while the CFO can reach several hundred Hz without significantly affecting the F1 score. Fig. 7e shows the impact of packet interval estimation error on the F1 score, indicating that our technique can perform well with an interval estimation error of up to one second. However, we have observed an average interval estimation error of around 0.18 sec in real-world scenarios, and our algorithm makes iterative adjustments to the preamble train to further reduce this error. Additionally, we found that when the F1 score decreases, it is mainly due to a decrease in precision rather than recall. This indicates that our technique can detect actual packets even under very adverse conditions.
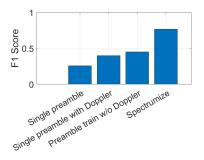
4.3.2 Real-world evaluation

In addition to emulating our technique, we also test it in real-world scenarios using our testbed. We apply our technique on

SDR recording traces gathered from ground station setups, which contain packets transmitted from our satellite. We collect traces from 67 satellite passes at various times over a one-year period to ensure that we capture a range of factors related to satellite orbital motion and noise at the ground station, including weather conditions. These traces contain over 700 packets of varying lengths transmitted using 2-GFSK and LoRa modulation schemes with intervals of a few seconds to a minute between transmissions. To carry out the evaluation, we implement our technique using MATLAB and execute it in a virtual machine having 32 GiB RAM and 4 vCPUs.

The results of the real-world evaluation and comparisons among different approaches are presented in Fig. 8. Fig. 8a displays the F1 score for packet detection using Spectrumize and a classic approach of single preamble-based correlation at different SNR values (rounded off to the closest integer number). Additionally, the distribution of packets across SNR values is shown. As we only have the SNR values for the packets decoded by the SX126x radio at the ground station during a pass, we use this as a reference and calculate relative SNR values for the packets detected by our approach but not by the SX126x radio. Fig. 8a demonstrates that Spectrumize outperforms the classic approach by 5× in low SNR scenarios, where we have a higher distribution of packets since a significant amount of packets reaching the ground station suffer from low SNR. In Fig. 8b, the overall performance of Spectrumize and other approaches in detecting packets is shown. Spectrumize exhibits 3× better performance than the classic approach of single preamble-based correlation and

(a) *F1 score for packet detection using Spectrumize and classic approach along with distribution of packet by SNR*

(b) *Overall F1 score for packet detection using Spectrumize and other approaches*

Figure 8: *Real-world evaluation: performance of Spectrumize in packet detection and comparison against classic approach (single preamble correlation without Doppler correction) and other intermediate approaches.*
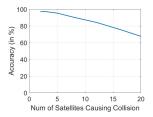
$2\times$ better performance even when we incorporate Doppler correction with a single preamble.
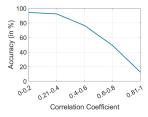
4.4 Collision Resolution

As explained in Section 2 and 3, packet collision is a significant problem when downloading data from a satellite, affecting the spectral efficiency. We now evaluate how effectively Spectrumize mitigates this issue.

4.4.1 Simulation

We first evaluate Spectrumize's ability in resolving packet collision through simulation. It further aims to assess the scalability of Spectrumize in terms of the number of satellites visible from a ground station as illustrated in Fig. 2. We select a polar region as the location for the ground station, as this is where satellites most frequently pass overhead, resulting in higher overlaps of their footprints. We simulate over 170 IoT satellites from the Swarm constellation by using their TLEs. Satellites are set to transmit packets at 2-second intervals when they come into view of the ground station. The packet length is set to 3.5 seconds to cause frequent collisions. The rest of the simulation setup is similar to that described in

(a) *Collision resolution accuracy by number of satellites in view of ground station*

(b) *Collision resolution accuracy by the correlation coefficient of Doppler signatures*

Figure 9: *Performance of Spectrumize in collision resolution and its scalability in simulation.*
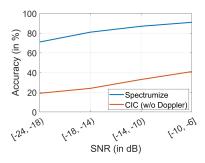

Figure 10: *Performance of Spectrumize's collision resolution in real-world. Spectrumize achieves over 80% accuracy on average across different SNR conditions.*

Section 4.2. As discussed in Section 3.3, we here use Spectrumize for packet detection and boosting the SNR of a packet of interest w.r.t. colliding packets using Doppler signature. We later employ CIC [38] to extract the symbols in a packet. Since the opensource source code of CIC supports only LoRa modulation, we use LoRa packets in our simulation. However, a similar technique can be translated to 2-FSK.

The performance of Spectrumize in resolving packet collisions is depicted in Fig. 9a, where we illustrate how its accuracy varies by the number of satellites that cause collisions. We begin with a few satellites from the Swarm constellation and gradually increase the number to the maximum in the simulation. We achieve an accuracy of over 90% in resolving collisions caused by up to eight satellites. The accuracy goes down as the number of satellites causing collision increases. This occurs since more satellites lead to a higher probability of similarity in Doppler signatures among the satellites. In Fig. 9b, we show the impact of similarity in Doppler signature on the accuracy, where it is measured in terms of the correlation coefficient between Doppler signatures from two satellites. The accuracy is observed to decline exponentially with the correlation coefficient.

4.4.2 Real-world evaluation

In addition to the simulation, we conduct the real-world evaluation of Spectrumize's ability to resolve packet collision using our testbed. To perform this evaluation, we use multiple satel-

lites from the same constellation launched by our collaborator. We configure two satellites to cause packet collision at the ground stations using LoRa modulation with a spreading factor of 11, a bandwidth of 125 kHz, and a code rate of 8. Given the limited number of satellites in orbit, naturally occurring collision scenarios are low. To address this, we employ our benchtop emulator to create such collisions and enrich the dataset of SDR traces. We then employ Spectrumize on the SDR traces to detect the packets and resolve the collision. We further compare its performance with the standalone CIC [38] approach proposed for terrestrial networks as a baseline. It is important to note that CIC does not use Doppler correction. The results in Fig. 10 show that Spectrumize achieves over 80% accuracy on average, with more than 70% accuracy in low SNR conditions and 90% accuracy in good SNR conditions. On the other hand, CIC performs poorly in low SNR conditions as it relies on the classic approach of single preamble correlation for packet detection. Although the performance slightly improves with the SNR, the absence of Doppler adjustment hampers its performance. We also observe that CIC struggles in decoding longer packets compared to shorter ones given the Doppler change within a packet.

5 Related Work

The quest to enhance spectral efficiency in satellite communication has garnered considerable attention in the state-of-the-art literature, with numerous techniques proposed, including MIMO and beamforming. [20, 44, 54], cognitive radio communication [39, 40], and interference mitigation and cancellation [16, 22, 22, 22, 23, 41], among others. As the coexistence of space and terrestrial networks becomes important in 5G and 6G, this field is gaining significant interest [18, 26, 34]. However, all of these techniques have been developed for broadband communication in space. This paper focuses on solving the problem of spectral efficiency in narrowband IoT satellite communication, which has unique challenges such as power constraints, lack of beamforming capability, lower frequency and bandwidth, and the use of low-cost and low-complexity connectivity hardware. As a remedy, we leverage the unique Doppler signature of satellites in packet detection and collision resolution.

5.1 Packet detection

In wireless communication, a preamble is added to the start of a packet to aid in synchronization. Various correlation-based methods have been proposed and implemented to detect the preamble in the received signal [27, 30, 56, 57]. However, in low SNR situations encountered in satellite IoT networks, single-preamble correlation techniques are not effective, as we have previously shown. To counteract the Doppler shift caused by the high relative motion of LEO satellites,

Doppler compensation techniques are typically used in satellite communication, both at the hardware and software levels [13, 17, 42, 49, 55]. To the best of our knowledge, we are the initial proponents of utilizing a series of preambles multiplied with the Doppler shift of a satellite to identify packets in situations where the signal-to-noise ratio is low.

5.2 Collision resolution

The process of collision resolution in satellite communication is well researched, but not specifically in the context of narrowband IoT satellite communication. On the other hand, in terrestrial IoT networks, such as LoRa, various techniques have been proposed to address packet collision. For example, Choir [24] groups LoRa symbols based on the CFO of the transmitter hardware, while FTrack [53] generates time-frequency tracks of the symbols using sliding window Short Term Fourier Transforms (STFT) on the de-chirped signal. Similarly, mLoRa [52] and CoLoRa [46] group LoRa symbols based on the power of the received signal. NScale [45] translates the packet time offset into frequency features for improved robustness, while CIC [38] cancels out interference by combining spectra obtained from different parts of each symbol. However, the major challenges in applying these techniques to satellite IoT networks are very low SNR and RSSI, and Doppler frequency offset.

6 Conclusion

The IoT satellite industry is growing rapidly due to the low cost, low-complexity, and ease of deployment of satellite constellations in orbit. However, our experience with launching a satellite shows that the downlink from the IoT satellite to the ground station suffers from large packet losses due to their low link budget, large footprint and packet collisions, and satellite motion. This paper introduces Spectrumize, a novel approach to improve the spectral efficiency of IoT satellite downlink. Spectrumize leverages the predictable motion of the satellite and uses the temporal variation of the Doppler shift as a signature to boost signal SNR and de-multiplex simultaneous transmissions. Our evaluation involving our launched satellite and collaborator's ground stations shows that Spectrumize improves downlink communication by $3\times$ over state-of-the-art.

7 Acknowledgement

References

[1] SatNOGS – Open Source global network of satellite ground-stations. https://satnogs.org/.

[2] TinyGS – The Open Source Global Satellite Network. https://tinygs.com/.

[3] Australian IoT startup Myriota raises USD 19 million. https://tinyurl.com/mw77mzh3, 2020.

[4] Satellite launches fueling race to connect out of reach devices. https://tinyurl.com/3ehc65r8, 2021.

[5] SpaceX buys out satellite IOT startup Swarm technologies. https://tinyurl.com/5y58ar4u, 2021.

[6] Cash-rich EchoStar to take on global IoT market next year. https://tinyurl.com/2rmw8kdw, 2023.

[7] FOSSA Systems - Global Cost-Effective IoT Connectivity. https://fossa.systems/, Feb 2023.

[8] Nanosatellite Launch Forecasts - Track Record and Latest Prediction. https://tinyurl.com/2p948kwv, 2023.

[9] Swarm - Low cost global satellite connectivity for IoT. https://swarm.space/, 2023.

[10] SX1302: LoRa Core Digital Baseband Chip for LoRaWAN network gateways. https://tinyurl.com/5d8xsb7k, 2023.

[11] USRP B200. https://www.ettus.com/all-products/ub200-kit/, 2023.

[12] Bassel Al Homssi, Kosta Dakic, Simon Maselli, Hans Wolf, Sithamparanathan Kandeepan, and Akram Al-Hourani. IoT network design using open-source LoRa coverage emulator. *IEEE access*, 9:53636–53646, 2021.

[13] Irfan Ali, Naofal Al-Dhahir, and John E Hershey. Doppler characterization for LEO satellites. *IEEE transactions on communications*, 46(3):309–313, 1998.

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

[15] Anritsu. Resolving Interference Issues at Satellite Ground Stations. https://tinyurl.com/2fbdzpek.

[16] Bassel F Beidas, Hesham El Gamal, and Stan Kay. Iterative interference cancellation for high spectral efficiency satellite communications. *IEEE transactions on communications*, 50(1):31–36, 2002.

[17] HD Black and A Eisner. Correcting satellite Doppler data for tropospheric effects. *Journal of Geophysical Research: Atmospheres*, 89(D2):2616–2626, 1984.

[18] Christophe Braun, Andra M Voicu, Ljiljana Simić, and Petri Mähönen. Should we worry about interference in emerging dense NGSO satellite constellations? In *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–10. IEEE, 2019.

[19] Tusher Chakraborty, Heping Shi, Zerina Kapetanovic, Bodhi Priyantha, Deepak Vasisht, Binh Vu, Parag Pandit, Prasad Pillai, Yaswant Chabria, Andrew Nelson, et al. Whisper: IoT in the TV White Space Spectrum. In *19th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2022*, pages 401–417. USENIX Association, 2022.

[20] Dimitrios Christopoulos, Symeon Chatzinotas, Gan Zheng, Joël Grotz, and Björn Ottersten. Linear and nonlinear techniques for multibeam joint processing in satellite communications. *EURASIP journal on wireless communications and networking*, 2012:1–13, 2012.

[21] Semtech Corporation. Semtech SX1262. https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1262, 2020.

[22] Cottatellucci, Laura and Debbah, Merouane and Gallinaro, Gennaro and Mueller, Ralf and Neri, Massimo and Rinaldo, Rita. Interference mitigation techniques for broadband satellite systems. In *24th AIAA International Communications Satellite Systems Conference*, page 5348, 2006.

[23] Marcos Alvarez Díaz, Nicolas Courville, Carlos Mosquera, Gianluigi Liva, and Giovanni E Corazza. Nonlinear interference mitigation for broadband multimedia satellite systems. In *2007 International Workshop on Satellite and Space Communications*, pages 61–65. IEEE, 2007.

[24] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. Empowering low-power wide area networks in urban settings. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 309–321, 2017.

[25] Tim Fernholz. Silicon Valley satellite startup installs ground station in Antarctica. https://tinyurl.com/4na6w7v6, Feb 2020.

[26] Ghaith Hattab, Prakash Moorut, Eugene Visotsky, Mark Cudak, and Amitava Ghosh. Interference analysis of the coexistence of 5G cellular networks with satellite earth stations in 3.7-4.2 GHz. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE, 2018.

[27] David L Herrick and Paul K Lee. CHESS a new reliable high speed HF radio. In *Proceedings of MILCOM'96*

IEEE Military Communications Conference, volume 3, pages 684–690. IEEE, 1996.

[28] Nozhan Hosseini and David W Matolak. Software defined radios as cognitive relays for satellite ground stations incurring terrestrial interference. In *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*, pages 1–4. IEEE, 2017.

[29] Microsoft. Azure Orbital. https://azure.microsoft.com/en-us/services/orbital/.

[30] Sumeeth Nagaraj, Sheehan Khan, Christian Schlegel, and Marat V Burnashev. Differential preamble detection in packet-based wireless networks. *IEEE Transactions on Wireless Communications*, 8(2):599–607, 2009.

[31] NASA. State-of-the-Art of Small Spacecraft Technology. https://www.nasa.gov/smallsat-institute/sst-soa/communications, 2022.

[32] Cunhua Pan, Hong Ren, Maged Elkashlan, Arumugam Nallanathan, and Lajos Hanzo. The non-coherent ultradense C-RAN is capable of outperforming its coherent counterpart at a limited fronthaul capacity. *IEEE Journal on Selected Areas in Communications*, 36(11):2549–2560, 2018.

[33] Eugenio Pasqua. Satellite IoT connectivity: Three key developments to drive the market size beyond USD 1 billion. https://iot-analytics.com/satellite-iot-connectivity/, Aug 2022.

[34] Deyi Peng, Dongxuan He, Yun Li, and Zhaocheng Wang. Integrating terrestrial and satellite multibeam systems toward 6G: Techniques and challenges for interference mitigation. *IEEE Wireless Communications*, 29(1):24–31, 2022.

[35] Jason Rainbow. Spanish startup to upgrade tiny satellites to take on Global IOT Market. https://tinyurl.com/4vkm6r6z, Feb 2023.

[36] Microsoft Research. CosmicBeats-Simulator: A space simulation platform that caters to individuals with diverse research interests, including networking, AI, computing, and more. Unlike traditional simulators tied to specific research applications, our design allows for seamless integration of various space-related research verticals. https://github.com/microsoft/CosmicBeats-Simulator, 2023.

[37] François Rottenberg, Ming-Chun Lee, Thomas Choi, Jianzhong Zhang, and Andreas F Molisch. Robust non-coherent beamforming for FDD downlink massive MIMO. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5. IEEE, 2020.

[38] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. Concurrent interference cancellation: Decoding multi-packet collisions in LoRa. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 503–515, 2021.

[39] Shree Krishna Sharma, Symeon Chatzinotas, and Björn Ottersten. Satellite cognitive communications: Interference modeling and techniques selection. In *2012 6th Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC)*, pages 111–118. IEEE, 2012.

[40] Shree Krishna Sharma, SYMEON Chatzinotas, and BJORN Ottersten. Cognitive radio techniques for satellite communication systems. In *2013 IEEE 78th vehicular technology conference (VTC Fall)*, pages 1–5. IEEE, 2013.

[41] Shree Krishna Sharma, Symeon Chatzinotas, and Björn Ottersten. In-line interference mitigation techniques for spectral coexistence of GEO and NGEO satellites. *International Journal of Satellite Communications and Networking*, 34(1):11–39, 2016.

[42] Marvin K Simon and Dariush Divsalar. Doppler-corrected differential detection of MPSK. *IEEE transactions on communications*, 37(2):99–109, 1989.

[43] Swarm Technologies Inc. Application for Mobile Satellite Service. https://fcc.report/IBFS/SAT-LOA-20181221-00094, 2018.

[44] Shigenori Tani, Katsuyuki Motoyoshi, Hiroyasu Sano, Atsushi Okamura, Hiroki Nishiyama, and Nei Kato. An adaptive beam control technique for Q band satellite to maximize diversity gain and mitigate interference to terrestrial networks. *IEEE Transactions on Emerging Topics in Computing*, 7(1):115–122, 2016.

[45] Shuai Tong, Jiliang Wang, and Yunhao Liu. Combating packet collisions using non-stationary signal scaling in LPWANs. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 234–246, 2020.

[46] Shuai Tong, Zhenqiang Xu, and Jiliang Wang. Colora: Enabling multi-packet reception in lora. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 2303–2311. IEEE, 2020.

[47] David A Vallado and Paul J Cefola. Two-line element sets-practice and use. In *63rd International Astronautical Congress, Naples, Italy*, pages 1–14, 2012.

[48] Deepak Vasisht, Jayanth Shenoy, and Ranveer Chandra. L2D2: Low Latency Distributed Downlink for LEO Satellites. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, SIGCOMM '21, page 151–164, New York, NY, USA, 2021. Association for Computing Machinery.

[49] Enric Vilar and John Austin. Analysis and correction techniques of doppler shift for non-geosynchronous communication satellites. *International journal of satellite communications*, 9(2):123–136, 1991.

[50] Quang-Doanh Vu, Le-Nam Tran, and Markku Juntti. Distributed noncoherent transmit beamforming for dense small cell networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4599–4603. IEEE, 2019.

[51] Quang-Doanh Vu, Le-Nam Tran, and Markku Juntti. Noncoherent joint transmission beamforming for dense small cell networks: Global optimality, efficient solution and distributed implementation. *IEEE Transactions on Wireless Communications*, 19(9):5891–5907, 2020.

[52] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. mlora: A multi-packet reception protocol in lora networks. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, pages 1–11. IEEE, 2019.

[53] Xianjin Xia, Yuanqing Zheng, and Tao Gu. FTrack: Parallel decoding for LoRa transmissions. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 192–204, 2019.

[54] Liang Yin, Ruonan Yang, Yuanzhou Yang, Li Deng, and Shufang Li. Beam pointing optimization based downlink interference mitigation technique between NGSO satellite systems. *IEEE wireless communications letters*, 10(11):2388–2392, 2021.

[55] Moon-Hee You, Seong-Pal Lee, and Youngyearl Han. Adaptive compensation method using the prediction algorithm for the Doppler frequency shift in the LEO mobile satellite communication system. *ETRI journal*, 22(4):32–39, 2000.

[56] Li Zhen, Hao Qin, Bin Song, Rui Ding, Xiaojiang Du, and Mohsen Guizani. Random access preamble design and detection for mobile satellite communication systems. *IEEE Journal on Selected Areas in Communications*, 36(2):280–291, 2018.

[57] Li Zhen, Yukun Zhang, Keping Yu, Neeraj Kumar, Ahmed Barnawi, and Yongbin Xie. Early collision detection for massive random access in satellite-based internet of things. *IEEE Transactions on Vehicular Technology*, 70(5):5184–5189, 2021.

[58] Lizhong Zheng and David N. C. Tse. Communication on the Grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel. *IEEE transactions on Information Theory*, 48(2):359–383, 2002.

[59] Xiangming Zhu, Chunxiao Jiang, Linling Kuang, Ning Ge, and Jianhua Lu. Non-orthogonal multiple access based integrated terrestrial-satellite networks. *IEEE Journal on Selected Areas in Communications*, 35(10):2253–2267, 2017.

A Simulation of Orbital Dynamics

Throughout the paper, particularly in Section 2, we incorporate various statistical data related to the orbital dynamics of a LEO IoT satellite constellation. This data includes information on the satellite's footprint coverage, elevation angles, and more. To generate these statistics, we utilize the CosmicBeats Simulator [36] provided by Microsoft Research. We configure the simulator with the locations of our ground stations and the TLEs for the complete Swarm constellation.

B Proof of Lemma 3.1

Lemma: *Consider $a_1, a_2, \ldots, a_n$ to be complex numbers on the unit circle with random phase. That is, $a_i = e^{j\theta_i}$ where θ_i is uniformly sampled from $[0, 2\pi]$. Let s_n be defined as $s_n = \sum_{i=1}^{n} a_i$. Then we have $\mathbb{E}\left[\|s_n\|^2\right] = O(n)$.*

Proof: s_n is the sum of n random phasors on the unit circle with phase sampled uniformly from $[0, 2\pi]$. By leveraging the commutative property of addition of complex numbers we have

$$s_n = s_{n-1} + a_n \tag{8}$$

We know that a_n has magnitude 1 and has uniform random phase in $[0, 2\pi]$. Hence, the phase difference between the complex numbers a_n and s_{n-1} can also be treated as uniformly distributed in $[0, 2\pi]$. Let the phase difference between a_n and s_{n-1} be denoted by ϕ_n. So from a simple inner product, we have

$$\|s_n\|^2 = \|s_{n-1}\|^2 + \|a_n\|^2 + 2\cos(\phi_n)\|a_n\|\|s_{n-1}\| \tag{9}$$

$$\|s_n\|^2 = \|s_{n-1}\|^2 + 1 + 2\cos(\phi_n)\|s_{n-1}\| \tag{10}$$

$$\mathbb{E}\left[\|s_n\|^2\right] = \frac{1}{2\pi}\int_0^{2\pi} \left(\mathbb{E}\left[\|s_{n-1}\|^2\right] + 1 + 2\cos(\phi_n)\,\mathbb{E}\left[\|s_{n-1}\|\right] \right) d\phi \tag{11}$$

$$\mathbb{E}\left[\|s_n\|^2\right] = \mathbb{E}\left[\|s_{n-1}\|^2\right] + 1 \tag{12}$$

We know that $s_1 = 1$. Hence, by induction we can see that

$$\mathbb{E}\left[\|s_n\|^2\right] = O(n) \tag{13}$$

The key takeaway from the above lemma is that even though we are adding up unit phasors with random phase values in $[0, 2\pi]$, the expected magnitude of their summation grows monotonically with number of terms rather than going to 0. Hence, by leveraging a virtual preamble train in Spectrumize, although we do not see the complete benefits of coherent combination, we still see enough gains that allows our system to detect satellite packets even in very low signal strength conditions.