Hidden in Plain Sight: Communicating using Interference

Ashton Palacios, Daniel Harman, Christopher Kitras, Elle Kelsey, Mitchell C. Burnett, Willie K. Harrison, Philip Lundrigan
Department of Electrical and Computer Engineering
Brigham Young University
Provo, Utah, USA
{apal6981, jdharman, chkitras, eh448, mitch.burnett, willie.harrison, lundrigan}@byu.edu

***Abstract*—Radio astronomy observatories examine the universe by capturing faint RF signals from space. The equipment to capture these signals is extremely sensitive, and nearby transmitters, such as LEO satellites, cause destructive interference. We propose the Spectra watermarking protocol to add an identifying fingerprint to these interfering transmitters. Spectra changes the timing of key transmissions to encode identifying information that can be decoded by examining transmission timestamps and without having to demodulate the actual transmission. We implement multiple forward error correction codes for Spectra to account for dynamic LEO satellite channels. We evaluate Spectra and show that it can operate over highly variable channels with low BER while preserving the network performance of the interfering device.**

***Index Terms*—wireless subprotocol, spectrum sharing, radio astronomy**

I. Introduction

The radio spectrum is a scarce resource. As more wireless devices and applications are developed, the demand for spectrum allocation is increasing [1], [2]. The explosion of low-earth orbit (LEO) satellite constellations highlights how fast we are using the spectrum on a global scale. LEO satellites provide tremendous benefits for connecting anywhere in the world without the need of terrestrial infrastructure. Starlink plans to deploy 40,000 satellites as part of its LEO constellation [3]. Other providers, such as Eutelsat's OneWeb [4] and Amazon's Project Kuiper [5], are planning deployments of their own. These LEO networks have the potential to change the way everyone in the world connects with each other. However, such ubiquitous connectivity comes at the cost of disruptive interference to other spectrum users. Previously, harmful interference was limited to sources on Earth. Now, wireless signals are coming from all directions. When such disruptive interference occurs, spectrum victims are left with little actionable information on where the interference came from. We can no longer develop and deploy wireless services without considering how it will affect other spectrum users. New techniques for spectrum cooperation must be developed [6]. Nowhere is the problem of spectrum crowding felt more than with radio astronomy observatories. Radio astronomy (RA) is the science of exploring the universe through studying the distant RF signals the universe generates. RA observatories have been key in discovering distant astronomical phenomena, such as cosmic microwave background radiation, pulsars, and distant galaxies [7]. RA observatories are *passive* users, meaning they only receive, making them especially vulnerable to harmful interference because RA observatories are impossible to detect by active users. LEO satellites are particularly problematic for RA observatories because of their position in the sky and their powerful transmissions relative to other signals in the sky [8]. RA observatories have dedicated spectrum bands, but also listen to spectrum active users occupy. Active LEO satellites bands are relevant to pulsar, fast-radio burst, and molecular line studies by RA observatories [9]. Efforts have been taken by LEO satellite providers to mitigate interference with RA observatories such as not transmitting in specific geographic regions where observatories are located, called radio quiet zones [10]. However, these efforts are not enough because LEO satellite constellations are large and always have satellites visible above the horizon [9]. RA observatories are extremely sensitive because they are designed to receive distant signals [11]. As a result, multipath propagation, satellite antenna side lobes, and imprecise satellite

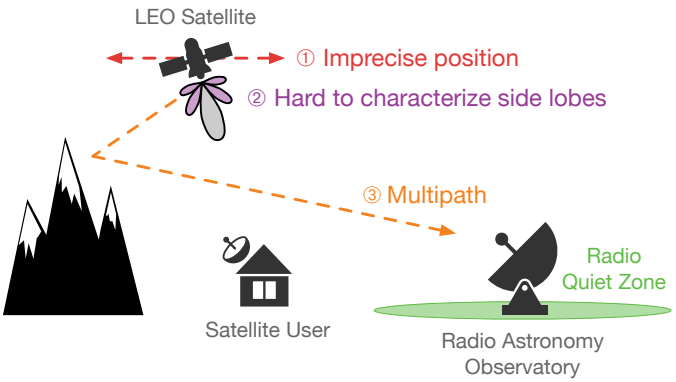

Fig. 1. An overview of potential causes of interference from LEO satellites on RA observatories.

This material is based on work supported by the National Science Foundation under Grant No. 2030165 and No. 2153317.

locations make it challenging to predict where interference will occur, and as a result RA observatories are still experiencing interference [12]–[15]. Fig. 1 shows a representation of the factors that lead to interference at RA observatories from LEO constellation deployments.

When interference occurs, what actionable information does the victim have? Besides knowing the time of when it happened, the frequency range, and possibly the direction, there is little *actionable* information about the source of the interference. If interference cannot be mitigated, the collected data are flagged and deleted [16], which is a waste of time and money. One solution to this problem would be to require all wireless devices to have a well-defined preamble with information about the device transmitting. This would allow any device to decode the preamble and receive information about the interfering device, regardless of modulation of the payload. While possible in theory, there are many practical limitations, such as slowing down the throughput of a device, that make this impossible. A universal preamble becomes especially challenging when dealing with a diverse set of heterogeneous devices, with their own capabilities and needs. The question becomes, *how can heterogeneous transmitters encode actionable information about themselves without negatively affecting their data rate*? This is where our work innovates. In this paper, we present a way of communicating **through interference**. We call this system **Spectra**. Spectra encodes secondary data in the primary flow of data by making slight changes to the timing of packets/transmissions. This allows a transmitter to send **two parallel streams of data at once**, the primary data, which is sent using the device's normal modulation techniques, and a secondary spectrum coordination data stream that is encoded in the **timing of transmissions**. Although timing changes are imperceptible to the end user and have no impact on throughput, a device that is being interfered with can decode the secondary data **without having to demodulate the primary signal** by looking at the timing of transmissions. With Spectra, when interference occurs, the victim can receive secondary data from the interference source, giving it actionable information to identify the source of the interference and mitigate it. The novelty of the idea is that it encodes data in already transmitted data. If a device does not transmit, then no secondary data is sent, but no secondary data is needed because the device is not causing interference on the spectrum. If a device is transmitting a lot of data, then it will encode the coordination data faster. *Our system requires no changes to the hardware and can be completely implemented in the software*. In other words, this protocol provides a **low impact** way of communicating spectrum coordination data between heterogeneous devices by controlling **only the software** of the sender and receiver devices.

In this paper, we focus on the problem of interference from LEO satellites on RA observatories because of the importance to allow the RA community to conduct future observations [17]. However, the techniques that we develop can be applied to any wireless system. Our system is a general solution to the problem of identifying sources of interference. The major contributions of this paper are as follows.

- We design a fully fledged wireless protocol on top of another wireless protocol, with its own modulation scheme, coding, and packet structure. We call this sub-protocol **Spectra**. This protocol allows spectrum-sharing coordination to occur with little to no overhead on the transmitter between heterogeneous devices.
- We present the signal and channel models for Spectra, including the major error modes: packet drops and network delays. We implement short block length error control codes to improve message fidelity and present bit error rate (BER) results for simulated and captured data.
- We implement Spectra in software on commodity hardware and characterize its performance. We characterize its effect on the primary stream of data, showing that it has no impact on throughput and only minimal impact on jitter under specific conditions. We develop a modulation scheme that can encode multiple bits of data into the jitter of a packet. Using this technique, we are able to achieve 200 bps. We demonstrate our system over an actual LEO satellite link to show that it works in real world scenarios.

II. Related Work

1) Ghost Modulation: The work presented in this paper is a continuation of a protocol named Ghost Modulation (GM) [18]. GM selects specific transmissions to perturb the transmission time of to encode information. GM has a few major drawbacks that make it impractical to implement on real LEO satellite networks.

The first major flaw is the way that GM synchronizes. GM uses differentially encoded packet times to synchronize. This is brittle because any packet that arrives in the synchronizing window can cause synchronization errors, whereas Spectra uses correlation that allows noise up to a user-defined threshold. GM is also limited in throughput speeds. GM can only encode a single bit per symbol period. Spectra can encode multiple bits in the same symbol period using multiple packets within the symbol period. GM is insecure, which allows GM sequences of packets to be captured and replayed. This could lead to a host of problems for radio astronomy observatories depending on how the embedded identifying information is used. Spectra tackles this by utilizing a time-based one-time password to allow Spectra receivers to detect a replayed Spectra packet. Finally, GM packets are susceptible to noise because no channel coding is applied to the bits. Spectra has multiple designed forward error correction codes to lower bit error rates. Using channel coding and the ability to send multiple bits per symbol period allow multiple modulation and coding schemes (MCS) to be devised and implemented, enabling Spectra to function over a wide range of channel conditions.

2) Interference Mitigation in Radio Astronomy: The field of radio astronomy has employed various techniques to help mitigate radio frequency interference. Three approaches commonly used today are centralized spectrum sharing servers, radio quiet zones, and signal processing techniques.

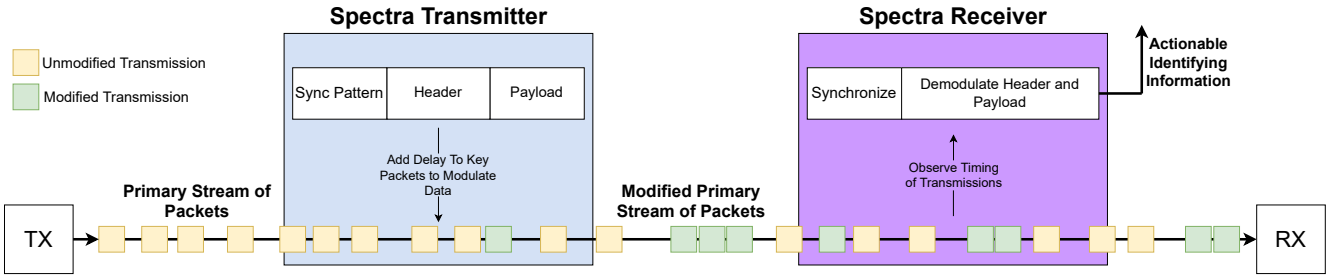

Fig. 2. Spectra transmitter and receiver modulating and decoding actionable information on top of a stream of packets (shown in yellow). The transmitter encodes data by adding small delays to specific packets (shown in green), while the receiver examines the transmission timestamps to decode the actionable data. The receiver passes the encoded data to spectrum sharing or interference mitigation techniques while the packet stream continues to the intended receiver.

One method of coordinating spectrum usage with observatories is to use a centralized spectrum calendar [19], similar to how the Citizens Broadband Radio Service Spectrum Access Service operates [20]. Observatories post what bands of spectrum are being observed and for how long. All other spectrum users in the area check the posted schedule daily and use the spectrum accordingly. Although users should check the calendar before using the spectrum, interference is still experienced at the RA observatory as a result of devices not strictly adhering to the spectrum coordination plan. Spectra can work congruently with the spectrum calendar to help identify devices that violate the coordination plan.

RA observatories are located within geographical regions denoted as RF quiet zones where radio transmitter regulations are enforced and aggressively monitored. [10]. This reduces the amount of interference by reducing the number of transmitters in the area. However, LEO satellite constellations pose a problem because they still fly over these areas [15], [21]. Although these satellites are programmed not to transmit to these quiet zones locations, satellite transmissions still propagate to these locations via the antenna side lobe [12], multipath propagation [13], or the satellite position is uncertain and the satellite transmits unknowingly into the region [14]. Geographical restrictions have worked well in reducing interference from terrestrial networks, but with more satellite transmissions occurring, additional methods, such as Spectra, are needed to help mitigate interference.

Signal processing techniques are also used to identify and remove interference [22]–[24]. Our work does not supplant these methods but provides complementary information about the interference that can be used to improve those techniques. This additional information can include, but is not limited to, satellite position, trajectory, and frequency plan.

3) Other Inter-Packet Modulation Techniques: Others have encoded data in the timing of packets [25]. Many of these timing solutions are able to use the underlying multiple access schemes or similar mechanisms of the wireless technology they are targeting. This type of solution allows the authors to have fine control over when packets are transmitted and received. These types of solution have higher bit rates but require greater control over the transmitter and receiver. Our solution avoids manipulating lower layer control schemes and rather embraces the added jitter these mechanisms add to the system. Other inter-packet techniques look at the inter-packet delay of consecutive packets. These techniques struggle when there are additional packets on the channel that are not part of the IPD technique, leading to bit errors. In addition, most of these solutions are only binary channels and have bit errors when multiple packets arrive within the decoding time. Our solution accounts for and embraces the possibility having multiple packets arrive within the decoding window by using the multiple packet arrivals to create a multibit encoding that enables higher throughput and is less susceptible to noise.

III. Spectra Design

Spectra is a protocol that encodes information on top of another stream of data. Encoding information on top of another stream is challenging because we do not have full control of the stream, such as when packets are generated. However, devices that are passing along or transmitting this stream of data generally have the ability to modify the timing of the packets. Spectra takes advantage of this by encoding information by adding only small delays to the transmissions of that stream of data. The primary stream of data does not need to be demodulated by the Spectra receiver, but the Spectra receiver merely needs the ability to observe when transmissions are on the channel. We discuss in detail both the transmitter and receiver and the functionalities needed to enable Spectra to work.

A. Spectra Overview

Spectra has three entities, the Spectra transmitter and receiver, and a stream of packets, as shown in Fig. 2. The Spectra transmitter, the device causing interference, can be implemented in any layer of the Internet protocol stack that has the ability to modify the timing of packets. However, the closer the Spectra transmitter is implemented to the transmitting hardware, the more control Spectra has over modifying the timing of packets. By modifying the timing of the primary stream, the primary stream of data becomes the medium into which the transmitter adds delays to encode information. To encode a Spectra packet within a stream of packets, the transmitter begins by transmitting a synchronization pattern that the Spectra receiver synchronizes to in time. Synchronization

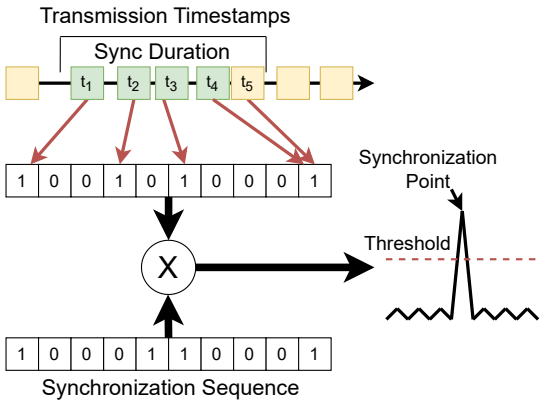

Fig. 3. Time synchronization for Spectra. Transmission timestamps and the synchronization time sequence are quantized into time bins and then correlated to determine the time synchronization point. Green boxes represent actual Spectra modified transmissions while yellow are other transmissions

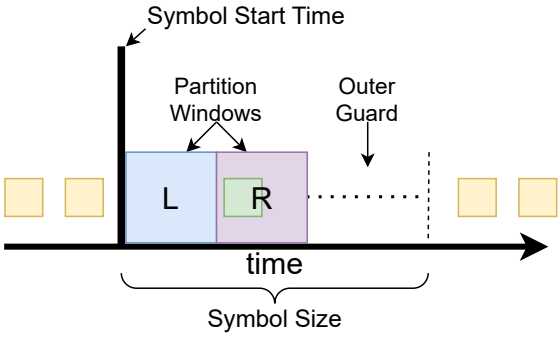

Fig. 4. Spectra symbol window. A Spectra symbol window contains two partition windows where Spectra modulates information by placing transmission within the partition (green box). The outer guard interval pads between Spectra symbols. Yellow box are other transmissions that were not used for Spectra.

is further discussed in Section III-B. After sending the synchronization pattern, the transmitter sends the Spectra header and payload. The header and payload are transmitted using different modulation and coding schemes (MCS), depending on the number of packets in the primary stream of data and the current channel metrics. Due to lack of space, the selection of the MCS is briefly discussed in Section III-D. After the header is transmitted, the Spectra transmitter securely sends the payload, which contains identifying information about the transmitter of the primary stream of data that the Spectra receiver can use to inform spectrum sharing or interference mitigation techniques and algorithms. While the Spectra transmitter needs the ability to modify the timing of packets, the Spectra receiver only needs the ability to timestamp packets or transmissions. Adding timestamps to received astronomical data at the output of the digitizer is already standard practice in RA receivers. The individual pieces of the Spectra transmitter and receiver are now discussed.

B. Synchronization

Spectra synchronizes across time using a correlation technique. The Spectra receiver records transmission detection times until enough time elapsed between the first and most recent transmission to contain a synchronization bit sequence. The transmission timestamps and the synchronization bit sequence are then quantized in time slice bins shown in Fig. 3 where a 1 in a bin denotes *at least one* transmission occurring within that time slice.

After the detected transmission timings are quantized, they are cross-correlated with the quantized synchronization bit sequence. If a returned correlation value is above a set threshold, synchronization is achieved. However, some additional processing is performed before the Spectra data bits are decoded. We calculate the average error quantization error and use it to adjust the timing of decoding Spectra bits to account for the timing error.

C. Data Modulation and Demodulation

Spectra operates around the idea of purposefully delaying a few packets of a primary stream of data by a small amount to encode secondary information. Fig. 4 shows the basic structure of a Spectra symbol. The start of the symbol is denoted as the symbol start time. A sequence of symbol start times is created from a pseudorandom sequence generated from a preshared key between the Spectra transmitter and receiver. The pseudorandom sequence of symbol start times is used to spread the impact of Spectra on the primary stream of data. If every packet of the primary stream of data is modulated, the entire stream of data experiences increased latency and jitter. We choose to distribute the increased latency and jitter over a larger portion of time to lower the average latency and jitter at the cost of Spectra throughput to preserve the transmitting device's network performance.

Within the Spectra symbol window we have two different types of time windows: partition windows and a guard interval shown in Fig. 4. The partition windows are where Spectra aims to place a transmission to encode information. There are partition windows for both zero and one bits. The guard interval pads between symbol windows to reduce the likelihood of transmissions meant for one symbol window landing in another symbol window.

Decoding symbol windows is accomplished by examining transmissions in both partition windows and comparing the ratio between the two. A ratio is used because Spectra does not control all the transmissions on the channel and other transmissions may fall into a partition window. Decoding using a ratio dampens the effects of these spurious transmissions. If the ratio of packets of one partition window is larger than the other than the other by a defined threshold, the bit denoted by that window is returned. If the ratios are inconclusive, then a bit erasure is returned which is defined and discussed in Section III-F.

The discussion thus far has been about 2-ary Spectra modulation. Spectra is easily expanded to an M-ary modulation.

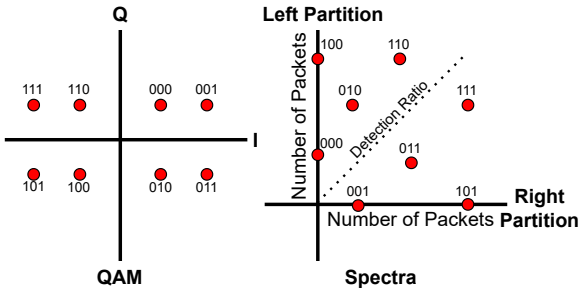

Fig. 5. M-ary Spectra Constellation. M-ary Spectra is similar to QAM. Instead of phase and amplitude M-ary Spectra uses left to right partition ratio and number of transmissions, respectively. Detection is done by examining the ratio of packets between the left and right partition windows.

TABLE I
M-ARY SPECTRA MODULATED BITS TO TRANSMISSION RATIO IN PARTITION WINDOWS.

Bits	000	001	010	011	100	101	110	111
L/R Ratio	2/0	0/2	3/1	1/3	5/0	0/5	5/3	3/5

M-ary modulation is accomplished by changing the number of purposeful transmissions and how they are distributed within a symbol window. The number of transmissions and how they are distributed are similar to the amplitude and phase of QAM, shown in Fig. 5. Using the number of packets in each window, we are able to form arbitrary constellations, similar to QAM. Demodulation of M-ary Spectra is similar to 2-ary Spectra with the only difference being that the transmissions can be distributed differently throughout the symbol window. For example, Table I shows different combinations of the number of transmissions and their distributions to the corresponding Spectra bits.

Employing M-ary Spectra modulation has one main advantage; it increases the throughput of Spectra while keeping the impact of Spectra on the primary network low. Spectra throughput in the 2-ary case could be increased by lowering the symbol window time and the space in between symbol windows, but this places an undue burden on the primary flow of data if there are more transmissions to be transmitted than there are symbol periods within the same time period. This causes increased latency, jitter, and packet drops for the primary flow of data. A key insight into the design of Spectra is that by allowing multiple transmissions to be used to encode Spectra bits within a symbol window, more transmissions may be left unperturbed or perturbed less while increasing Spectra throughput.

D. Header

Spectra packets are designed to be as short as possible to give spectrum victims multiple opportunities to decode actionable information in a Spectra packet. As such, the Spectra header is 3 bytes as shown in Fig. 6.

The Spectra header includes three fields: a modulation and coding scheme (MCS), a checksum, and security bytes. The

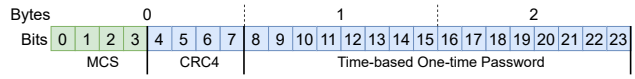

Fig. 6. Spectra header structure. MCS bits are transmitted at the Default MCS rate while the rest of the header is sent at the selected MCS rate.

MCS field allows the Spectra transmitter and receiver to adapt to the current conditions the satellite is experiencing both in terms of throughput and jitter. The MCS controls the partition window size, outer guard window size, how often within time a Spectra symbol is sent, which FEC code set to use, and the modulation style the Spectra packet is using. With the MCS field being 4 bits long, 16 different MCS sets can be used for any given Spectra packet. This enables the Spectra transmitter to monitor the current channel conditions and dynamically choose the MCS set that provides high Spectra throughput while maintaining a low bit error rate for every transmitted Spectra packet. Spectra's checksum is a CRC4 [26] checksum that is adequate to check the small number of bytes sent in a given Spectra message. The security bits are discussed in Section III-G.

To ensure that the Spectra receiver can decode the Spectra packet, the MCS field is first transmitted. In addition, it must be transmitted with a default MCS that works under all channel conditions. This enables the receiver to decode the MCS field without additional outside information. This is analogous to WiFi transmitting its MCS field at the lowest supported data rate. The transmitter then transmits the rest of the header and payload bits using the chosen MCS parameters. The receiver will then decode the remaining header and payload bits using the received MCS.

E. Payload

To provide actionable information to the device experiencing interference, the payload of our Spectra message is the satellite's North American Aerospace Defense (NORAD) ID [27]. This will ensure that each satellite's message is unique, identifies, and integrates seamlessly with the established cataloging system. The NORAD ID can then be used by the RA observatory to find additional information such as current operating frequency, mean motion, motion plan, or eccentricity about the interfering satellite from databases such as Space-Track [28].

The NORAD ID ranges from seventeen to thirty bits, depending on whether a five- or nine-digit ID is assigned. Due to this small size, the Spectra payload is 16 bytes long. This enables other actionable information to be included if desired, while still being brief. Due to the brevity of the Spectra payload, the Spectra packet will be transmitted multiple times within a short time period, which provides the spectrum victim multiple opportunities to discover who the interferer is.

F. Channel Coding

Spectra experiences errors due to loss and jitter of network packets. Packet loss and jitter cause Spectra transmissions to

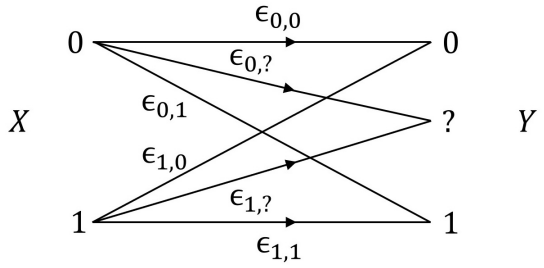

Fig. 7. Spectra channel model with generally asymmetric probabilities.

not land within the wanted Spectra partition window. These errors can result in ambiguities for the Spectra receiver in addition to potential bit flips. These ambiguities are referred to as erasures, as the symbol information is erased rather than flipped. In this section, we discuss the need for forward error correction (FEC) and discuss our choices of error correction codes. We also introduce the signal and channel models for Spectra. A more comprehensive theoretical analysis of Spectra is found at [29].

1) Signal Model: A binary[1] Spectra signal of length N symbols, can be represented as

$$s(t) = \sum_{k=0}^{N} \delta(t - kT_\mathrm{s} - \theta_k T_\mathrm{w}) \quad \theta_k \in \{0, 1\} \tag{1}$$

where T_s and T_w are the symbol and bit window times respectively. θ_k represents a Spectra symbol from the constellation $\{0, 1\}$. The Dirac delta function, $\delta(t)$, represents a transmitted Spectra symbol because all the Spectra symbol information is contained in the time stamp of the packet. The received signal can be represented as

$$r(t) = \sum_{k=0}^{N} z_k \delta(t - kT_\mathrm{s} - \theta_k T_\mathrm{w} - \tau_k) \tag{2}$$

where $z_k \sim \text{Bern}(1-\rho)$, ρ is the packet drop rate, the network delay is modeled as $\tau_k \sim e(\lambda)$, and $\frac{1}{\lambda}$ is the mean network delay.

2) Channel Model: It is clear from (2) that there are two sources of error, network jitter and packet loss. The network jitter implies a channel with memory. To simplify the model, however, we use an equivalent discrete memoryless channel depicted in Fig. 7. We use the random variable X to denote a channel input and the random variable Y to represent a channel output. We model the binary inputs as $X \sim \text{Bern}(\gamma)$ where γ is the probability of a 0 and $1 - \gamma$ is the probability of a 1. The ternary channel output is modeled as random variable Y. The transition probability, $P(Y|X = x)$, from state n to m is denoted as $\epsilon_{n,m}$.

As discussed in [29], the channel is generally asymmetric. Additionally, certain errors produce ambiguous outputs (e.g. a dropped packet gives no information, even if the receiver is certain that a packet was supposed to arrive during a certain symbol period), and so a ternary output is required. These ambiguities are labeled as erasures (denoted by ? in Fig. 7). The erasures are resolved with FEC coding.

3) Error Correction Coding for the Asymmetric Channel: Due to the simplex nature of Spectra, Spectra cannot use the typical network error-correction scheme of a checksum or robust error detection codes (e.g., cyclic redundancy checks) in tandem with automatic repeat requests (ARQs) to correct corrupted packets. Instead, a robust FEC code must be used to correct errors in the receiver without additional aid from the transmitter. This proves to be an interesting coding and information-theoretic challenge, as no capacity-achieving codes are known for asymmetric channels of this type.

Previous works address coding for standard asymmetric channels [30], although common approaches in the literature use large block length capacity-achieving codes such as polar codes. These schemes are ill-suited to the Spectra signal and channel models due to the inherently low data rate and short block length of Spectra packets. For this reason, we explore common, short block length codes with erasure decoding algorithms, namely Hamming codes and Reed-Muller codes. Hamming codes are used because of their ubiquity within error control coding. In addition, we use Reed-Muller codes as they have recently been found to achieve capacity over the binary erasure channel (BEC) [31] and have short block-length constructions at coding rates suitable for Spectra.

Specifically, the codes employed and tested are Hamming codes of order 3 and 4 as well as first-order Reed Muller codes with block lengths of 2^3 and 2^4. These code constructions are commonly denoted as Hamm(3), Hamm(4), RM(1,3) and RM(1,4). We instead refer to them by their block length n and dimension k i.e., Hamm(n,k) or RM(n,k) to emphasize their different code rates and the trade-off between error-correcting capability and added overhead.

G. Security

We design Spectra to be secure against three potential security vulnerabilities: packet alteration, spoofing, and replay attacks. In the security design of Spectra, we assume an adversary capable of listening to the primary stream of communication as well as Spectra communication (eavesdropping). The adversary can transmit energy on the channel to spoof or modify a Spectra transmission. The ability of an adversary to jam communication, both the primary and the Spectra communication, is beyond the scope of this work. To analyze these security threats in the context of Spectra, we introduce Alice, Bob, Eve, and Ollie to represent the actors and their roles in potential attack scenarios.

Alice is a LEO satellite network that transmits data to Bob. Alice also uses Spectra to encode secondary data into the primary data stream. Eve is a malicious eavesdropper who wishes to replay or modify and spoof the Spectra transmission. Ollie is a passive observer (in our case, a RA observatory) who is the target of Eve's attacks. Each entity is shown in Fig. 8. Attacks on Alice and Bob's communication by Eve

[1]Analysis here is performed for 2-ary Spectra for simplicity, but it is trivial to extend to the M-ary case.

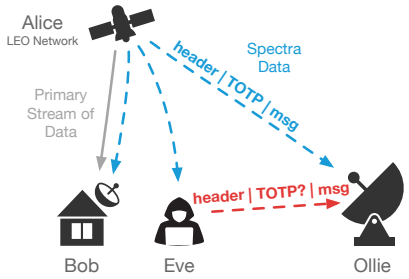

Fig. 8. Spectra protects against an adversary that modifies or spoofs other Spectra transmissions.

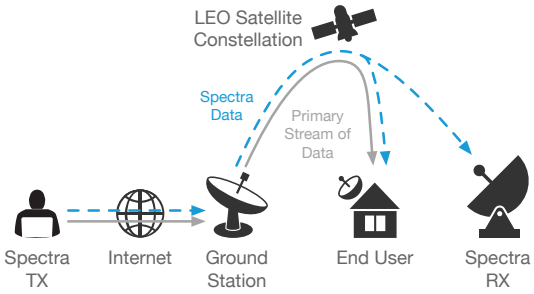

Fig. 9. Our implemented Spectra system. A Spectra transmitter sends secondary data through the Internet and a LEO network to a device listening to the RF spectrum.

are outside the scope of this work, since they depend on the security of the primary communication stream, which our system does not determine. Our security considerations focus on protecting the secondary communication stream. The secondary data Spectra sends is a public NORAD ID, no confidential information is transmitted. As a result, our security does not include encryption to ensure confidentiality. If confidentiality is necessary, a user can encrypt the payload of the Spectra packet.

Since Eve can transmit, she might alter the Spectra data to attribute the interference source (i.e. the LEO satellite to be identified) to a different entity. We must also protect against Eve spoofing a Spectra message or transmitting a previously recorded set of transmissions. Since Ollie is not demodulating the transmissions and only looking at the time at which energy is the channel, Eve only needs to replay the energy on the channel. Eve's motivation for replaying a previously recorded Spectra transmission or modifying a legitimate Spectra transmission is to fool Ollie, who makes decisions based on the presence of these transmissions. For example, Eve might want to trick Ollie into thinking that the source of interference is from a LEO satellite, even if one is not present.

To avoid the possibility of Eve modifying the data in flight, Alice and Ollie use the pseudorandom key that determines where the symbol windows are, as discussed in Section III-C. Since Eve cannot determine the pseudorandom sequence where Spectra happens, it is extremely difficult for her to introduce noise in an intentional way that will modify the meaning of a Spectra payload. If she introduces too much noise to the point where the original message is lost, it becomes a jamming attack and is out of the scope of our protocol.

To avoid Eve sending a spoofed message or replaying an old transmission, we include a time-based one-time password (TOTP) [32] in the header of a transmitted Spectra packet. The TOTP is derived using a seed shared by Ollie and Alice. The TOTP changes periodically as defined by its time slice, which we set to one minute. Since Eve does not know the shared seed, she cannot derive the TOTP to spoof a message. Since the TOTP is time-based, replaying an old message is detectable by Ollie. Eve's only option is to flood the network with messages while attempting to guess the TOTP field. To mitigate this, we derive a TOTP with an adequate length and a time slice of sufficient duration to avoid TOPT replays.

Given Spectra's relatively slow data rate, we choose a smaller TOTP length to reduce the overhead of each packet. However, too small of a length leaves us vulnerable to Eve randomly guessing the TOTP. We create a TOTP with a length of 16 bits, which gives her a pool of 65,536 possibilities. Although this may initially seem too small, we also consider that the shortest duration of a Spectra transmission is approximately four seconds (using the highest MCS value). At that rate, it would take Eve on average 32,768 attempts or 1.5 days of continuous transmission to guess the correct TOTP. We also assume there is no possibility for parallel attacks for this MCS value, since the highest MCS leaves no space between symbol windows, leaving no room for another valid stream of Spectra transmission.

IV. Implementation

Spectra focuses on perturbing preexisting traffic. This enables Spectra to be implemented in many places along the flow of traffic. The flow of traffic within a LEO satellite network has four main entities that generate or pass traffic: devices on the Internet, ground stations, LEO satellite constellations, and the end users, as shown in Fig. 9. The closer Spectra is implemented to the actual source of interference, the satellite, the less likely the Spectra message will be corrupted by noise. Due to the lack of hardware and software access to LEO satellite constellations and ground stations, we implement and evaluate Spectra from the perspective of devices on the Internet, which introduces additional jitter to the traffic within which Spectra operates. Doing so is more challenging and shows the resiliency of Spectra to noise.

To demonstrate the adaptability of Spectra, we implement Spectra on top of Starlink. We create an application that generates traffic and allows Spectra packets to be modulated on top of it. This traffic is then transmitted over a real Starlink network between a cloud server and a local machine in our lab. The application on the local machine receives the traffic and simultaneously measures network performance and decodes Spectra messages for evaluation from packet reception times.

In addition to implementing Spectra on Starlink, we implement a Spectra receiver using a software-defined radio (SDR). Using an SDR allows us to more closely approximate interference detection and Spectra decoding from the perspective of an RA observatory. We do not have sensitive enough equipment to properly detect beamformed transmissions from LEO satellites. In this work, we approximate this scenario using an SDR and commercial off-the-shelf WiFi hardware. We use an SDR to timestamp transmissions on a WiFi channel while a WiFi device is generating and transmitting traffic that has embedded Spectra packets. This setup is a harsher scenario because there are dozens of devices transmitting that may interfere with the Spectra transmission. In future work, an SDR with a satellite downlink RF receiver would be sensitive enough to properly detect beamformed transmissions and better approximate a RA receiver.

V. Evaluation

To ensure that Spectra can help mitigate interference for RA observatories while preserving LEO satellite network performance, we evaluate Spectra in three cases: measuring the impact of LEO satellite jitter on Spectra, the ability of passive devices to detect and decode Spectra in the spectrum, and finally the impact that Spectra has on the native transmission.

A. Worst Case Jitter on Starlink Network

LEO satellite constellations operate with a dynamic RF channel. One of the largest impediments for these channels is weather. Weather disrupts satellite communication by absorbing and scattering transmissions [33]. These effects cause a variable amount of jitter for the packets on the channel. High amounts of jitter on these packets could lead to bit errors within the Spectra message that we want to characterize.

1) Actual Starlink Jitter: We measure and characterize the jitter of our Starlink link using iPerf, a common network performance measurement application. We run the iPerf application every five minutes for a minute to collect approximately three weeks worth of data with varying weather conditions ranging from sunshine to rain to snow. The average jitter of the worst 0.05% of our data is 12 ms. Another research group has recorded slightly worse jitter metrics at around 20 ms [34]. Spectra needs to have the capability to operate with a low bit error rate (BER) when there is a large amount of jitter on the network.

2) Spectra BER with Simulated Starlink Jitter: We simulate different levels of jitter by sweeping the average jitter in an exponential distribution and evaluate its effect on Spectra bits. While sweeping jitter values, we also apply a 2% packet drop rate [35] to the simulated packets. The results are shown in Fig. 10. The results show that by using our FEC codes, Spectra can operate over a wide range of jitter values and still have a low BER. As the simulated jitter values decrease, the BER plateaus for each code. This is due to the packet drop rate of 2% creating a performance limit that adjusts as the packet drop rate is adjusted.

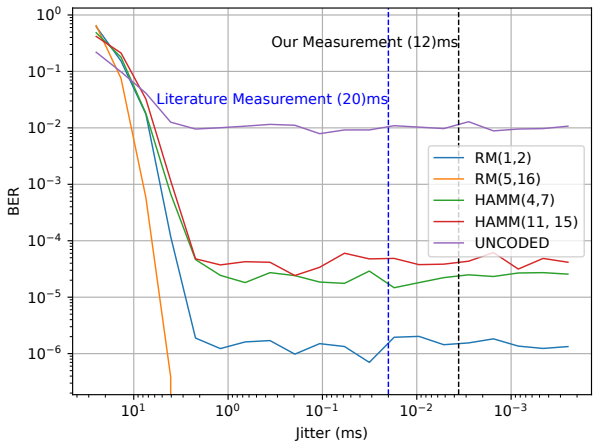

Fig. 10. Evaluation of Spectra channel codes across different jitter values. Selected Spectra channel codes provide adequate gain to have low BER for channels experiencing levels of jitter measured on real LEO satellite constellation links.

The RM(5,16) code performs extremely well in that we could not simulate enough bits to accurately plot the BER as the simulated jitter value decreased. We choose to use the RM(5,16) channel code for the remaining evaluations.

B. Spectra Detection in RF

We evaluate the resiliency of Spectra on a real RF channel with many sources of interference. RA observatories do not usually have the capability to demodulate transmissions but merely listen and record samples of the RF channel looking for signals of interest. We evaluate using a similar topology by sampling a WiFi channel in our lab. This is a noisy channel filled with lots of transmissions, whereas RA observatories are usually located within radio quiet zones that shield RA equipment from many sources of interference. Our evaluation setup represents a worst-case scenario for RA observatories in that there are many sources of interference at the same time. We use an Ettus USRP B210 SDR to monitor our WiFi channel for transmissions that have embedded Spectra packets, and we control the load on the WiFi channel by using the iPerf application to transmit packets over the air at different throughput rates.

We evaluate both the Spectra bit error rate (BER) and the packet error rate (PER). The results of our evaluation are found in Fig. 11. Spectra is affected by how many transmissions are on the channel and not necessarily by the duration of the transmissions. As such, we measure the BER and PER of Spectra as a function of average inter-transmission spacing on the channel normalized to the symbol window size. We normalize to the symbol window size because the closer packet transmissions occur on the channel, the more likely spurious transmissions land within the Spectra symbol window which can lead to increases in the BER and PER.

As the spacing of transmissions on the channel approaches the Spectra symbol window size, both the BER and PER

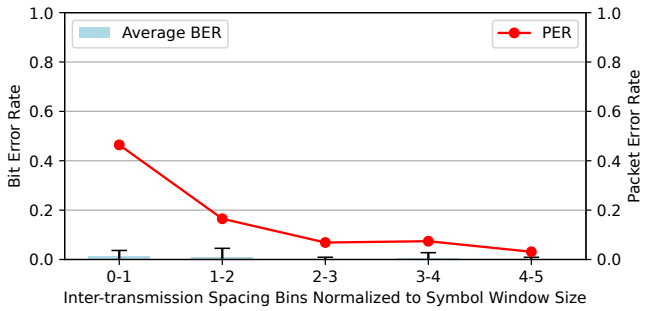

Fig. 11. Spectra performance over a WiFi RF channel. BER and PER increase as the average spacing between transmissions approaches the symbol window size.

increases. Depending on where the Spectra transmitter is implemented, different amounts of traffic can be handled. Additionally, Spectra packets are designed to be transmitted multiple times over a single geographical area, providing multiple opportunities to correctly decode the entire Spectra message. RA observatories can detect and decode Spectra messages in the radio spectrum that have low BER and PER depending on the symbol window size, which is controllable through the selection of the MCS and the amount of traffic on the channel. Our BER is higher compared to our simulated results in Section V-A2 because of the complexities of real networks.

C. Spectra on Starlink

We evaluate the impact of transmitting over a Starlink network has on Spectra and the impact Spectra has on a Starlink network to ensure that Spectra can watermark LEO satellite transmissions while having minimal impact on the LEO satellite network. We use the configuration shown in Fig. 9. We create four different MCS sets that have both 2-ary and M-ary Spectra modulations. Our primary data stream, which has embedded Spectra packets, is transmitted from a server in the cloud through a Starlink connection to a computer in our lab.

1) Impact of Starlink of Spectra: Spectra is designed to be versatile and have the capability to operate on a real LEO satellite network. We measure the BER for the four selected MCS sets and report the results in Table II. The results show that each MCS set has a low BER over the Starlink network. This shows that LEO satellite transmissions can be watermarked with Spectra messages that spectrum victims can decode to identify the source of interference.

2) Spectra Impact on Primary Stream: Spectra aims to have minimal impact on the throughput and jitter of the primary stream of data. Our primary data stream is generated by a custom implementation of the iPerf application. This allows us to easily choose different throughput rates to transmit at while also embedding Spectra. We measure the throughput and jitter of this link for the different MCS sets, shown in Table II, and compare against network metrics when Spectra packets are not embedded. Our results are shown in Fig. 12. The results show

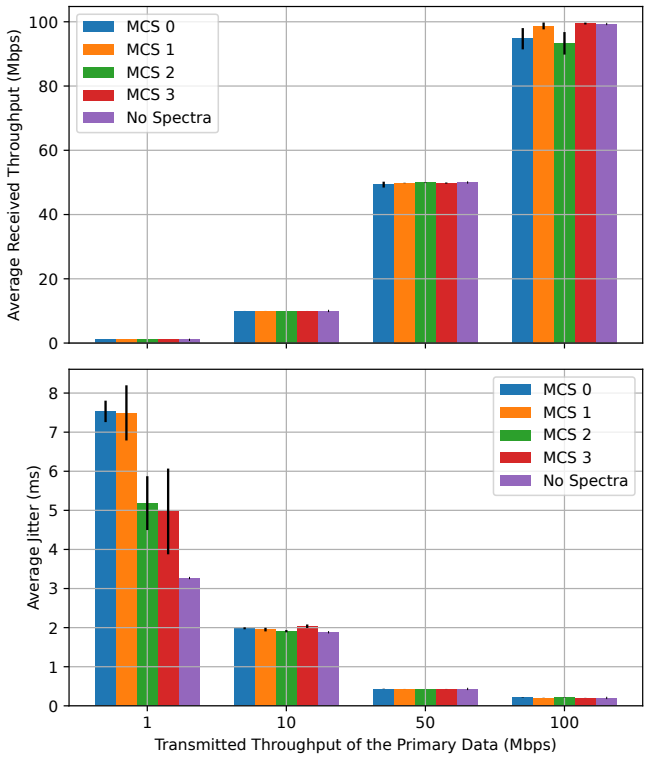

Fig. 12. Impact of Spectra packets on the primary channel's network performance. Spectra has little to no impact on the throughput of the channel with minimal impact on the jitter. The amount of jitter added to the network is dependent on throughput and Spectra MCS selection.

TABLE II
MODULATION AND CODING SETS FOR DIFFERENT PARTITION SIZES AND M-ARY SPECTRA MODULATIONS

MCS	Partition Size (ms)	Outer Guard Size (ms)	Bits per Symbol	Packets per Symbol	BER
0	40	5	1	1	0.1%
1	40	5	2	2,4	0.1%
2	2.5	5	1	1	3.1%
3	2.5	5	2	2,4	3.2%

that the average received throughput rate is approximately equal to the throughput rate sent. As the transmit throughput increases, the spread of the received throughput also starts to increase. This is due to the inefficiencies of our custom iPerf application and the actual Starlink connection.

The results also show the spread of the jitter values across the different MCS sets for the various transmit throughput rates. Ideally, the jitter values across the MCS sets would be equal to the jitter experienced by the data stream without Spectra. The results show that the added amount of jitter is dependent on the throughput rate and the MCS set chosen. This comes in part because if there are fewer transmissions, then more of the transmissions will be modulated by Spectra. M-ary Spectra MCS sets also show that they added reduced amounts of jitter compared to 2-ary MCS sets in some cases by over 30%. M-ary Spectra is designed to efficiently use more

transmissions to encode information. Because M-ary Spectra uses more transmissions, fewer transmissions are perturbed by large delays, which leads to less jitter in the system while simultaneously increasing Spectra throughput.

Although the throughput rate affects the amount of jitter experienced by the receiver, the added jitter is still far below the constraint of having less than 30ms for demanding VoIP [36] applications. Spectra is able to be added on top of primary data streams to create a secondary data stream with minimal impact to the primary stream. The source generating and embedding Spectra messages can monitor channel conditions and throughput and select the appropriate MCS set to meet performance constraints of the primary and Spectra data streams.

VI. Conclusion

Spectra is a software-implemented protocol for watermarking transmissions from interfering LEO satellites, enabling radio astronomy observatories to identify interfering LEO satellites. The Spectra watermark is embedded within the timing of the interfering transmissions. We find that Spectra has the ability to work within the dynamic conditions of a satellite communication channel. We also find that embedded Spectra packets have little to no impact on the primary data stream, thus preserving the timing and other network constraints of the interfering device. Spectra enhances interference management with minimal impact on satellite communication.

References

[1] W. J. Hastyo, M. Suryanegara, Fardan, and K. Anwar, "Spectrum demand for mobile broadband: A review and simulations for case of indonesia," in *2023 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD)*, 2023, pp. 203–208.

[2] I. Shayea, T. Abd. Rahman, M. Hadri Azmi, C. Tien Han, and A. Arsad, "Predicting required licensed spectrum for the future considering big data growth," *ETRI Journal*, vol. 41, no. 2, pp. 224–234, 2019.

[3] C. Henry. (2019, Oct) Spacex submits paperwork for 30,000 more starlink satellites. Accessed: 2024-08-23.

[4] Eutelsat. (2024) Oneweb. Accessed: 2024-08-23. [Online]. Available: https://oneweb.net

[5] Amazon. (2024) Project kuiper. Accessed: 2024-08-23. [Online]. Available: https://www.aboutamazon.com/what-we-do/devices-services/project-kuiper

[6] F. H. Sanders, "A 53-year history of spectrum efficiency studies and recommended future directions," Institute for Telecommunication Sciences, Tech. Rep., 2018.

[7] M. Panda and Y. Chandra, "Unveiling the mysteries of the cosmos: An overview of radio astronomy and its profound insights," 2023. [Online]. Available: https://arxiv.org/abs/2308.09415

[8] Q. Liu, Y. Liu, N. Wang, and M.-Z. Chen, "Quantified interference level limits for qtt key areas," in *2016 Radio Frequency Interference (RFI)*, 2016, pp. 55–58.

[9] C. Walker, S. Di Pippo, M. Aubé, J. Barentine, Z. Benkhaldoun, P. Benvenuti, C. Bouroussis, R. Green, J. Hearnshaw, H. Liszt *et al.*, "Dark and quiet skies for science and society," *International Astronomical Union*, 2020.

[10] C. Beaudet, P. Woody, W. Sizemore, R. McCullough, J. Ford, F. Ghigo, C. Niday, and C. Clark, "The green bank interference protection group: Policies for rfi management," Jan 2007.

[11] B. F. Burke, F. Graham-Smith, and P. N. Wilkinson, *An introduction to radio astronomy*. Cambridge University Press, 2019.

[12] A. Hills, J. M. Peha, J. Munk, and S. Pogorelc, "Controlling antenna sidelobe radiation to mitigate ku-band leo-to-geo satellite interference," *IEEE Access*, vol. 11, pp. 71 154–71 163, 2023.

[13] N. Heydarishahreza, T. Han, and N. Ansari, "Spectrum sharing and interference management for 6g leo satellite-terrestrial network integration," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024.

[14] C. Foster, H. Hallam, and J. Mason, "Orbit determination and differential-drag control of planet labs cubesat constellations," 2015. [Online]. Available: https://arxiv.org/abs/1509.03270

[15] P. P. Sitompul, T. Manik, M. Batubara, and B. Suhandi, "Radio frequency interference measurements for a radio astronomy observatory site in indonesia," *Aerospace*, vol. 8, no. 2, p. 51, 2021.

[16] R. Series, "Techniques for mitigation of radio frequency interference in radio astronomy," *Rep. ITU-R RA*, pp. 12–13, 2013.

[17] S. Harper and C. Dickinson, "Interference from global navigation satellites in future hi intensity mapping surveys," in *2016 Radio Frequency Interference (RFI)*, 2016, pp. 31–36.

[18] A. Palacios, C. Bledsoe, E. Kelsey, L. Landon, J. Backman, and P. Lundrigan, "Stealthy signals: Using ghost modulation to watermark interference," in *Proceedings of the 1st ACM Workshop on LEO Networking and Communication*, ser. LEO-NET '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 13–18.

[19] G. B. A. Observatory, "Gbt schedule," 2022.

[20] FCC, "Citizens broadband radio service rules," 2015. [Online]. Available: https://www.ecfr.gov/current/title-47/chapter-I/subchapter-D/part-96

[21] Z. Fang, G. Li, J. Zheng, and T. Feng, "Interference analysis for mobile cellular and leo satellites co-existence," in *2022 IEEE 22nd International Conference on Communication Technology (ICCT)*, 2022, pp. 434–439.

[22] C. Barnbaum and R. F. Bradley, "A new approach to interference excision in radio astronomy: Real-time adaptive cancellation," *The Astronomical Journal*, vol. 116, no. 5, p. 2598–2614, 1998.

[23] A. Leshem and A.-J. van der Veen, "Radio-astronomical imaging in the presence of strong radio interference," *IEEE Transactions on Information Theory*, vol. 46, no. 5, p. 1730–1747, 2000.

[24] A. Palacios, D. Ward, D. Bronson, J. Backman, D. Heo, K. F. Warnick, and P. Lundrigan, "Network layer spectral coordination integrated with hadamard projection for multilayer interference mitigation," in *2024 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2024, pp. 402–411.

[25] A. K. Biswas, D. Ghosal, and S. Nagaraja, "A survey of timing channels and countermeasures," *ACM Comput. Surv.*, vol. 50, no. 1, Mar. 2017.

[26] M. O. Ezea, H. O. Osuagwu, and M. A. Ahaneku, "Performance analysis of cyclic redundancy check (crc) error detection technique in the wireless sensor network," *Int. Res. J. Eng. Technol*, vol. 7, no. 6, pp. 4104–4110, 2020.

[27] R. T. Savely. National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Division, 1991.

[28] S. admin@space track.org, "Spacetrack.org." [Online]. Available: https://www.space-track.org/auth/login

[29] D. Harman, A. Palacios, P. Lundrigan, and W. K. Harrison, "An information theoretic analysis of ghost modulation," 2024. [Online]. Available: https://arxiv.org/abs/2412.05249

[30] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "How to achieve the capacity of asymmetric channels," *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3371–3393, 2018.

[31] S. Kudekar, S. Kumar, M. Mondelli, H. D. Pfister, E. Şaşoğlu, and R. Urbanke, "Reed-muller codes achieve capacity on erasure channels," in *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 658–669.

[32] D. M'Raihi, J. Rydell, M. Pei, and S. Machani, "TOTP: Time-Based One-Time Password Algorithm," RFC 6238, May 2011.

[33] Y. Wang, R. Li, J. Ma, Y. Song, L. Liu, and X. Wang, "Effect of rain attenuation on the availability of leo satellite communication system," in *2023 5th International Conference on Electronics and Communication, Network and Computer Technology (ECNCT)*, 2023, pp. 11–17.

[34] H. Fang, H. Zhao, F. Wang, Y. C. Chou, L. Chen, J. Shi, and J. Liu, "Streaming media over leo satellite networking: A measurement-based analysis and optimization," *ACM Trans. Multimedia Comput. Commun. Appl.*, Sep. 2024, just Accepted.

[35] F. Michel, M. Trevisan, D. Giordano, and O. Bonaventure, "A first look at starlink performance," in *Proceedings of the 22nd ACM Internet Measurement Conference*, ser. IMC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 130–136.

[36] T. Szigeti and C. Hattingh, *End-to-end QoS network design: Quality of service in lans, wans, and vpns*. Cisco Press, 2010.