Voice Over LoRa™

Francis Kagai*, Philip Branch†, Jason But‡ and Rebecca Allen§
*†‡ Department of Engineering Technologies, § Department of Physics and Astronomy
Swinburne University of Technology, Melbourne, Australia
Email: *fkagai@swin.edu.au, †pbranch@swin.edu.au, ‡jbut@swin.edu.au, §rebeccaallen@swin.edu.au

***Abstract*—In this paper, we describe our work on transmitting voice over LoRa. Our goal is to develop a LoRa / Satellite voice gateway for emergency services. This paper describes the LoRa component of that gateway. Our approach involves developing a prototype that utilizes our novel voice streaming protocol and assessing how different parameters affect voice transmission over the LoRa physical layer. Our experiments confirm that low-bit-rate voice can be transmitted over LoRa at distances exceeding one kilometer with acceptable levels of packet loss and bit error rates. Voice quality is positively correlated with larger packet sizes, albeit with increased packet loss. Smaller packets introduce some latency but reduce packet loss, while larger packets, especially those exceeding 124 bytes, pose a risk of packet fragmentation and a higher frequency of lost packets and bit errors.**

***Index Terms*—LoRa, Codec 2, Voice Over LoRa, Low bitrate**

I. Introduction

In this paper, we report on the first part of our research in developing a low-cost, easily deployed Voice-over LoRa (VLoRA) system. VLoRa is part of the Emergency Buddy System (EBS) intended for facilitating emergency voice communication in situations where existing infrastructure has been compromised or rendered inoperative. EBS architecture, as depicted in Figure 1, comprises a WiFi to LoRa gateway and a LoRa to satellite gateway. People living in disaster-prone areas would be provided with a low-cost WiFi-LoRa gateway designated for emergency use. In the event of a disaster, such as floods or bushfires, they would connect to the gateway and use press-to-transmit communication to talk to an emergency services center. The gateway then uses LoRa to carry their voice communications to a local hub which then uses satellite communication to transmit voice to the emergency response centre. The LoRa-Satellite hub would be either permanently located within the community or shipped in at the first opportunity following the disaster. In this paper, we report on our work in transmitting voice over LoRa using press-to-transmit.

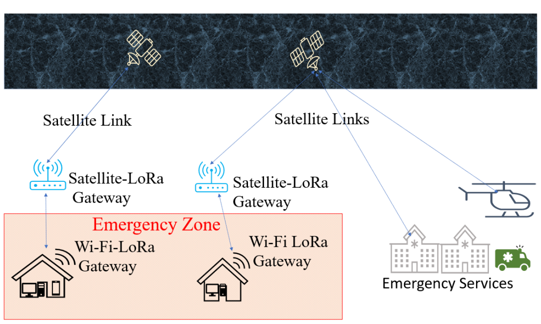

Fig. 1. The Emergency Buddy system

In most countries, radio frequencies are regulated by government authorities to prevent interference and ensure efficient use of the spectrum [1]. Certain portions of the radio frequency spectrum are designated for unlicensed use (e.g. 433MHz, 868MHz, 915MHz, 2.4 GHz, and 5 GHz bands). However, these unlicensed frequencies are not entirely ”free” in the sense that they are subject to standards, regulations, and restrictions that limit their applications [1]. Specific bands are earmarked for radio and television broadcasting, distinct ones for mobile communication, and yet others for Wi-Fi and other unlicensed applications [1]. These allocations help reduce interference among diverse services.

LoRa is a low-cost technology that operates in the unlicensed frequency bands. We are interested in LoRa because it serves as a cost-effective option, allowing individuals isolated by a disaster to connect to a satellite hub. By enabling multiple people to connect via the hub, expensive satellite infrastructure can be shared. Past experiments show that LoRa signals can propagate up to approximately two kilometers in dense urban environments and more than 500 meters inside a building [2]. This versatility places LoRa as a good candidate for linking short- and long-range networks [3].

LoRa uses a modulation technique known as Chirp Spread Spectrum(CSS) to transmit data over long distances with low power consumption [4]. CSS modulation involves linearly varying the frequency of the transmitted signal over time, creating a chirp signal [4]. This technique allows LoRa to achieve a long-range while remaining resilient to interference. The frequency range for LoRa varies by country or region. In Africa, it operates at 433MHz, in Australia, at 915 MHz, in Europe, the range spans from 863 MHz to 870 MHz, and in North America, it covers the 902-928 MHz band [5].

LoRa operates at the physical layer which is the lowest layer of the Open Systems Interconnection (OSI) model. It is commonly used in association with LoRaWAN although it is increasingly used without it. LoRaWAN is a higher-layer protocol that operates on top of the LoRa physical layer. It defines the networking protocols, data rates, and communication infrastructure for LoRa-based devices using the ALOHA protocol while relying on a star topology [6]. LoRaWAN reduces the raw capacity and speed of LoRa for individual devices significantly but this also brings significant benefits in terms of creating large-scale, low-power, and wide-area networks. The trade-offs are intentional and designed to enable efficient, long-range communication while ensuring fair and balanced access to the network resources.

In areas where there may be no other LoRa networks or the need to connect to other LoRaWAN networks, a simplified LoRa solution may suffice eliminating the need for the LoRaWAN protocol's features and complexity. This approach relieves the network of the added overhead and complexity of LoRaWAN's network management but allows more straightforward, point-to-point, or linear network topologies [3]. In our work, we do not use LoRaWAN.

Our approach involves the development of both a prototype and protocol for processing and streaming low-bit-rate voice. To address the challenges of voice communication over bandwidth-constrained LoRa links, we utilize Codec 2, an open-source codec developed by David Rowe, renowned for its efficiency in converting analog voice signals into a digital format at very low bitrates while maintaining reasonable voice quality [7].

Voice over LoRa as an option for infrastuctureless or low infrastructure communications is increasingly gaining attention. The most notable attempt to create a protocol for streaming voice-over LoRa using Codec 2 was undertaken by Beartooth as part of their proprietary communication infrastructure for use in the ITU Region 2 [8]. Their protocol combines multiple LoRa channels to facilitate voice streaming. In contrast, our protocol focuses on a single LoRa channel and can be applied in any region for emergency communications. Our experiments involve independently evaluating each of the 7 LoRa channels to understand their specific challenges. Subsequently, we develop mechanisms for real-time voice transmission over LoRa to address these challenges, resulting in a prototype and a protocol for sending voice over LoRa. Our research contributions are as follows:

a) We present a prototype for encoding, streaming, and decoding low-bit-rate voice, facilitating smooth voice transmission over low-bit-rate links such as LoRa, with adaptability to other similar technologies.
b) Our research introduces a comprehensive protocol for managing voice streaming over low-bit-rate links. This protocol has been specifically evaluated on LoRa technology and includes key functionalities such as stream initiation, voice data packetizing, streaming, warnings and error reporting, and voice termination.
c) Our research highlights the critical aspects of transmitting voice over LoRa, focusing on data packet size and latency per LoRa channel. This unique emphasis on addressing these issues in low-bit-rate communications represents a significant contribution to the broader understanding and development of low-bit-rate communication systems.

The rest of this paper describes the VLoRa system, which can be seamlessly integrated into low-bitrate networks, offering a dependable solution for the establishment of robust emergency communication networks. We begin with background information in Section 2. In Section 3, we describe the architecture of the VLoRa system developed for our experiments. Section 4 presents the results obtained from our system and their analysis. Finally, in Section 5, we present our conclusions.

II. Background Information

Despite LoRa technology being in existence since 2009 [9], most research on LoRa focuses on MAC layer messaging while relying on the LoRaWAN protocol [8]. The limited studies that focus on the LoRa physical layer do not provide critical details that could help in determining the technical feasibility of streaming voice over a single LoRa channel [8], [10], [11]. Additionally, past studies on voice-over LoRa mostly focused on evaluating the quality of a voice file transmitted over LoRa and on splitting voice data packets over multiple LoRa channels [8], [10], [11]

Storing of voice in a file and transmitting it over LoRa has been described in at least three past studies [11] [10], [12]. One of them is the ReSoNate system which has a phase for recording voice before transmission [11]. The ReSoNate system is evaluated based on the quality of the voice file transmitted over LoRa. In another study, voice data sent over LoRa is encoded using A-law, a well-established telecommunication standard for voice compression and decompression [10]. This approach provides a predictable level of performance and is highly compatible with legacy systems. However, the approach results in a large file not ideal for low-bitrate communications.

Message Queuing Telemetry Transport (MQTT) is another protocol for LPWAN communications that has attracted a lot of research [12]. In one such case, voice-based messaging is integrated with the MQTT system for users who cannot read and write but as is in [10], the system sends voice as a file as opposed to a real-time stream. Additionally, the system is inefficient and not well adapted for low bandwidth communications. The results of the study show that it takes over 100 seconds to send voice files on LoRa spreading factor 7. This is due to the system relying on the MP3 file format which generates larger data sizes, resulting in increased transfer times over LoRa [12].

The most advanced Voice over LoRa system was developed by Beartooth [8]. Beartooth's system uses Codec 2 to stream voice while employing a frequency hopping mechanism that depends on semi-orthogonal hopping sequences. This approach makes the system compliant with a US regulation that limits the maximum channel transmission duration to 500 milliseconds [8]. As such, Beartooth's voice Over LoRa is designed to use multiple channels to stream voice with each channel transmitting voice data for less than 500 milliseconds. The received data is then rearranged in real time on the receiving side for playback via Codec 2. The system uses a configurable gateway protocol that relies on a configurable file placed in all participating nodes, and employs Raspberry Pi in tandem with an embedded SX1276 LoRa chipset shield to enhance its compatibility with mobile phones.

While Beartooth's approach is innovative, it differs from our work in many ways, most notably in its use of multiple channels for voice communication. By contrast, our work uses a single channel per user. While that may result in poorer voice quality than Beartooth, it permits more concurrent users which is an important consideration for an emergency communications system.

The scope of this paper differs from previous work on Voice over LoRa in that our VLoRa system is designed to be adaptable for integration into a larger satellite-based emergency communication system. In essence, VLoRa serves as a backup system, remaining idle until there is a need for emergency communication in areas where satellite signals cannot reach, or where other communication systems are unreliable, nonexistent, or have failed.

III. System Architecture

VLoRa is a peer-to-peer (P2P) communication system that utilises LoRa to transmit an audio stream between two devices. As illustrated in Figure 2, the network comprises two end devices for data processing and two intermediary microprocessors with embedded LoRa shields for physical layer communications. The end devices both capture/playback audio and generate the packet stream. The microprocessors act as a bridge between LoRa and the application, enabling modification of stream parameters without requiring re-compilation for each experiment. We expect that ultimately all processing could be performed on a dedicated device.

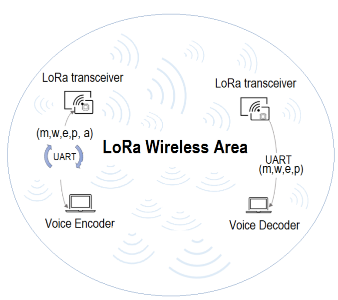

Fig. 2. The Physical Design of the VLoRa system

As the channel capacity offered by LoRa is limited, LoRa is unable to transmit an uncompressed voice stream. A codec is essential for voice encoding and compression on the transmitting device, and to decode and playback at the receiver.

Voice and music codecs are distinct in design and tailored to the unique characteristics of the audio they manage. They prioritize efficient compression within a narrow frequency range, emphasizing intelligibility even at low bitrates. In contrast, music codecs address the broader and complex frequency spectrum of music, aiming to support richness and detail. Voice codecs commonly employ Adaptive Differential Pulse Code Modulation (ADPCM) and low-bitrate techniques, while music codecs use advanced methods operating at higher bit rates to preserve audio quality.

In our case, we are further constrained by the limited bitrate offered by LoRa. Table I summarises the CoDecs we reviewed that are capable of achieving toll-quality voice using advanced low-bitrate techniques.

Offering the highest compression rate, we selected Codec 2 [7] at the lowest available bit rate of 700bps for our experiments. Our goal is to evaluate all possible combinations of LoRa signal bandwidth and spreading factors to determine the suitability of using LoRa for real-time voice transmission.

A. *Voice Generation and Playback*

Audio signals are analog, to transmit over LoRa, it is necessary to convert the audio to a digital stream before encoding and transmission. The encoded bitstream is then aggregated into bytes which are segmented to form a packet. We aim to evaluate the best range of packet sizes that can be efficiently and reliably transmitted over LoRa.

B. *VLoRa Protocol*

Like most modern networks, LoRa is a packet-based network. Data is required to be segmented for transmission. To emulate an environment where the network can be used for multiple applications, we need to develop a VoIP-like protocol to manage the data stream between the two end nodes.

Inspired by the Real Time Protocol (RTP) for streaming over the Internet, we developed a cut-down version to facilitate the management of the VLoRa stream. Like RTP, we use a connection-less approach, and acknowledgments and retransmissions are not used. The receiver can identify a lost packet via a missing sequence number in the stream.

All VLoRa packets contain a 3-byte header followed by the packet payload. The packet format is displayed in Figure 3.

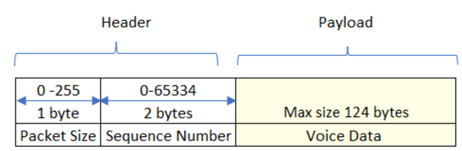

Fig. 3. Data Packet with a header and payload

The header consists of an 8-bit value indicating the VLoRa payload size in bytes, followed by a 16-bit value containing the stream sequence number encoded in network byte order. Experimentation showed that transmission of LoRa packets greater than 124 bytes was unreliable, typically resulting in fragmentation by the hardware. As such the maximum payload size is capped at 124 bytes, and 8 bits is ample to contain the packet size. The 16-bit sequence number allows the VoIP stream to contain a maximum of 65,534 voice packets.

To allow the receiver to properly manage received data, we cannot just transmit the compressed audio data. The protocol needs a mechanism to inform the receiver when an audio stream is about to start and when it is terminated. The protocol sequence is illustrated in Figure 4.

VLoRa uses a Stream Initialisation packet to start the streaming process. Following this, a series of stream data packets are sent until the audio stream is terminated. Upon conclusion, a stream termination packet is sent to inform the receiver that the stream has ended.

TABLE I
LOW BITRATE VOICE CODECS

Voice Codec	Sample Rate per sec	Minimum Bit Rate	Best Uses
G.711	8000 Khz	64 kbps	Communication between VoIP & PSTN
G.729	8000 Khz	8 kbps	Low Bandwidth Channels
Opus	up to 48,000 Khz	6 kbps	HD Voice and clear sound
Lyra V2	8000 Khz	3.2Kbps	Low Bandwidth Channels
Codec 2	8000 Khz	0.7 kbps	Low Bandwidth Channels

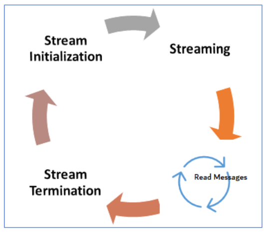

Fig. 4. Voice Streaming Cycle

1) Stream Initialisation Packet: Streaming cannot commence until the **Stream Initialisation** packet is successfully received. The format of this packet is shown in Figure 5.

8 bits	16 bits	16 bits	16 bits
0 -255	0-65334	0-65334	0-255
Packet Size	Sequence Number	Packet Number	Codec

Fig. 5. Stream Initiation Header Packet Format

All **Stream Initialisation** packets contain a payload size of 3 and a sequence number of 0. The payload consists of a 16-bit value in network-byte order nominating the total number of audio packets in the stream, or zero if the number is not known. This is followed by an 8-bit value indicating the CoDec used to encode the data. The current supported CoDecs are:

- **1** – G.711
- **2** – Codec 2
- **9** – G.729

Upon receiving a **Stream Initialisation** packet, the receiver should initialise the selected CoDec and playback buffers, and prepare to receive a stream of VLoRa **Stream Data** packets.

2) Stream Data Packet: The encoded audio stream is sent in a series of **Stream Data** packets. All **Stream Data** packets are encoded with incrementing sequence numbers starting at 1. Sequence numbers 0 and 65,535 are reserved for **Stream Initialisation** and **Stream Termination** packets respectively.

The **Stream Data** packet payload consists solely of encoded audio data. Our implementation uses a fixed payload size, with the possible exception of the final packet. As most voice CoDecs generate fixed bitrate data, each packet will contain a fixed duration of encoded audio.

3) Stream Termination Packet: The **Stream Termination** packet is sent to conclude a stream and has a sequence number of 65,535 and no payload. This packet is used by the receiver to terminate playback after draining any remaining buffered data and subsequently release playback resources.

C. System Implementation

At the application layer, we use Python for voice processing at the sending and receiving edges of the system. The Python Pyaudio library is used to access the audio platform. Audio is recorded live and passed to the Codec 2 module for compression. Compressed data is packaged on the fly and sent over the serial link to the Arduino, which in turn bridges received packets to the LoRa physical layer for transmission.

At the receiver, the process occurs in reverse. Packets arriving at the Arduino are forwarded to the application via the serial link. This data is then decoded and buffered for playback via Pyaudio. The bit rate is managed by the CoDec, and the total size of data generated is dependent on the recording duration.

All VLoRa packet generation and decoding is handled at the application layer. Fully formed packets are sent to the Arduino for transmission over LoRa. Similarly, fully formed packets are received from the Arduino. These packets are transmitted using a simple serial protocol. A single character signifies the type of message being transferred followed by the message. Valid message types are:

- **p**: A binary encoded, formatted packet. The receiver uses the encoded packet length to locate the end of the packet
- **m, w, e**: A NULL terminated ASCII string sent from the Arduino to the application for display. The message is alternatively **m** Informational; **w** Warning; or **e** Error
- **a**: A VLoRa packet header indicating acknowledgment that the nominated packet was transmitted. Sent to the application by the sending Arduino to indicate readiness to accept another packet for transmission

Some applications were written to facilitate repeated testing of the VLoRa protocol under controlled conditions. These applications allowed the specification of both CoDec and packet size to automate data collection.

IV. RESULTS AND ANALYSIS

Past studies on the performance of the LoRa physical layer demonstrated a higher throughput than that needed for real-time streaming of voice encoded by low-bitrate codecs such as G.729, Opus, and Codec 2 [13]. We aggregated data (see figure 6) from such studies to form a baseline for this research. This illustrates which signal bandwidth and spreading factor combinations can support the target Codec 2 bitrate of 700 bits per second. Our hypothesis is: “We can stream data on LoRa channels that have a bandwidth of around 700 bits per second”. To test this, we also evaluated channels below 700 bits per second to better understand the behavior of voice data in bandwidth-limited environments.

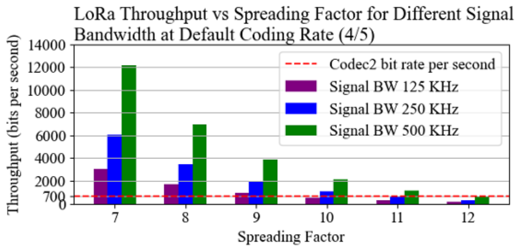

Fig. 6. The documented LoRa Bandwidth

A. The Impact of Packet Size on LoRa Latency

Time is an important factor in the computation of throughput and goodput of a communication channel. In this study, we investigate LoRa latency to evaluate if there are any factors affecting the time on air of the voice data during the voice stream. We recorded three types of latencies namely: sender latency, LoRa latency, and receiver latency.

The baud rate of the UART interface is set at a high speed (115200 baud) to minimise time spent transferring data between the Arduino and application layer devices so that it doesn't interfere with data transfers between the microprocessor and sending/ receiving devices. Experiments were run using packets of different sizes to evaluate the impact of packet size on various LoRa channels.

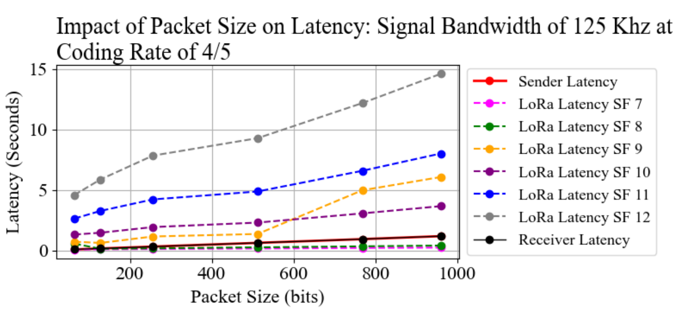

Fig. 7. Performance of Voice over LoRa on 125 Khz channel

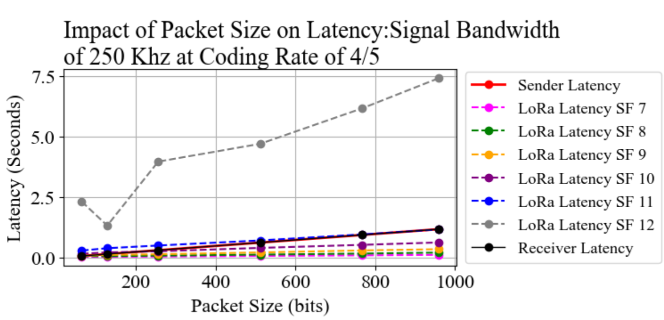

Fig. 8. Performance of Voice over LoRa on 250 Khz channel

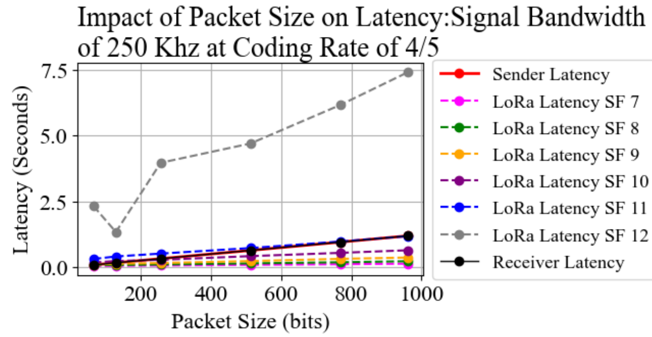

Fig. 9. Performance of Voice over LoRa on 500 Khz channel

Results are plotted in Figures 7, 8, and 9. we note a peculiar trend where the latency exhibits distinct behavior based on packet size despite employing a consistent bandwidth, spreading factor, and coding rate throughout. Smaller packets, such as 8, 16, and sometimes 32 bytes, display remarkably high non-consistent latency. In contrast, larger packets, starting from 32 bytes and beyond, demonstrated reduced and stabilized latency.

The observed variations in latency behavior can be attributed to a fundamental issue related to the data generation rate. Specifically, Codec 2 generates data at a fixed rate of 700 bits per second. When data is transmitted at a pace faster than Codec's data generation capacity, it leads to data unavailability for immediate transmission. Consequently, smaller packets experience longer delays as they must be queued leading to the observed higher latency. In contrast, larger packets contain more data to transmit, which means data is always available for transmission leading to consistent latency.

The high latency of smaller packets can be addressed by creating a buffer either on the sending side or receiving side to align the data generation process with the transmission rate. Queuing and Buffering is an ideal solution that would stabilize voice stream over LoRa at the expense of latency. It is worthwhile to note that, these approaches would give the wrong evaluation as some channels that couldn't stream will consequently stream.

B. Packet loss and Bit errors

To assess packet loss rates and bit error occurrences in VLoRa, we conducted experiments using voice streams with varying packet sizes, involving up to 100 packets. The results are presented in Figures 11 and 12. These tests took place in an environment where the transmitter and receiver were positioned approximately 1.2 kilometers apart in the Melbourne metropolitan area. Both the transmitter and receiver were situated at an elevation of about 100 meters above the ground.

From the graphs, it is evident that packets smaller than 16 bytes experienced minimal or no packet loss. The rate of packet loss was found to be correlated with packet size, with larger packets showing a higher incidence of packet loss. However, the trend is not entirely predictable due to the influ-

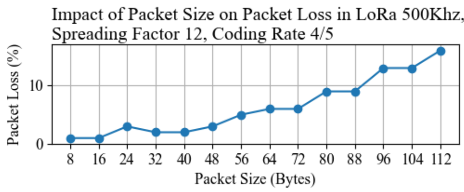

Fig. 10. Packet size vs Packet Loss

ence of various factors. For instance, in areas with significant obstructions, the plotted results were only achievable when the transmitter and receiver were positioned within a maximum distance of 500 meters from each other.

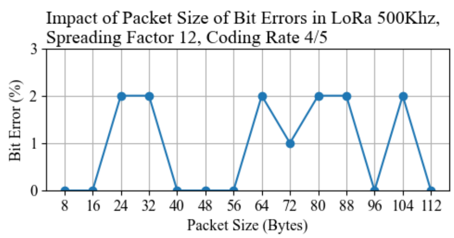

Fig. 11. Packet Size vs Bit Errors

Interestingly, the bit error rate remained unaffected by packet size throughout our experiments. In all test cases involving the transmission of 100 voice packets, a maximum of two packets exhibited bit errors, regardless of their size. Even in these instances, not all bits within a packet were affected. Larger packets did have a higher count of bit errors, primarily because they contained more bits. To validate our findings, we initiated a continuous stream, generating 1400 packets. On average, we observed that while packet loss exhibited an unpredictable pattern, bit errors were contingent on the stream's duration, with an average occurrence of approximately 2 packets with bit errors for every 300 packets transmitted.

V. Conclusion

Our comprehensive examination of voice transmission over LoRa networks has shed light on a critical determinant in optimizing voice quality over low-bitrate links exposed to interference: the packet size. Throughout our research, we observed that packet size plays a pivotal role in striking a balance between voice quality, latency, and susceptibility to packet loss. While larger packet sizes have the potential to enhance voice quality, they come with a significant drawback, as extended transmission times increase the likelihood of packet loss due to interference. Conversely, smaller packets introduce latency into the transmission process, primarily because voice codecs maintain a fixed bit rate per second. Consequently, it is imperative to maintain an equilibrium between packet size and frame size to ensure seamless audio playback. In situations where network delays result in throughput falling below the codec bit rate, the guarantee of a smooth audio experience becomes uncertain. To address this challenge, creating a buffer equivalent to the time differential becomes necessary.

In the context of our study, it is crucial to acknowledge that, while latency can be controlled and adjusted to some extent, mitigating packet loss and bit errors presents a more complex challenge. Additionally, relying on a network that discourages packet retransmission further emphasizes the importance of optimizing packet transmission. As a result, our findings underscore the significance of reducing the duration of packets in the air to minimize packet fragmentation, packet loss, and bit errors. We also emphasize the importance of ensuring an appropriate distance between the sender and receiver to minimize the potential for interference. This distance varies based on the use of the technology. Considering that these experiments were conducted in an urban, densely built-up area with anticipated substantial interference, we anticipate improved results when replicating similar experiments in less developed areas with minimal interference.

Acknowledgment

We gratefully acknowledge support from the SmartSatCRC in funding this research. We also gratefully acknowledge guidance from Dr Mark Rice of the Safety from Space project.

References

[1] M. Matinmikko-Blue, S. Yrjölä, and P. Ahokangas, "Spectrum management in the 6g era: The role of regulation and spectrum sharing," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*, pp. 1–5, 2020.

[2] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of LoRa: Experiences from a large-scale measurement study," *Association for Computing Machinery*, vol. 15, feb 2019.

[3] P. Branch, B. Li, and K. Zhao, "A LoRa-Based linear sensor network for location data in underground mining," *Telecom*, vol. 1, no. 2, pp. 68–79, 2020.

[4] B. Tuch, "Development of WaveLAN®, an ISM band wireless LAN," *ATandT Technical Journal*, vol. 72, no. 4, pp. 27–37, 1993.

[5] N. Lethaby, "Wireless connectivity for the internet of things: One size does not fit all," *Texas Instruments*, vol. 16, 2017.

[6] R. M. Liaqat, P. Branch, and J. But, "LoRa based linear network applications, design considerations and open challenges: A review," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, SenSys '22, (New York, NY, USA), p. 913–917, Association for Computing Machinery, 2023.

[7] D. Rowe, "Codec 2." https://www.rowetel.com/wordpress/.

[8] B. Mekiker, M. P. Wittie, J. Jones, and M. Monaghan, "Beartooth relay protocol: Supporting real-time application streams with dynamically allocated data reservations over LoRa," in *2021 International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–9, 2021.

[9] Semtech, "What is LoRa?." https://www.semtech.com/lora.

[10] R. Kirichek, V.-D. Pham, A. Kolechkin, M. Al-Bahri, and A. Paramonov, "Transfer of multimedia data via LoRa," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems* (O. Galinina, S. Andreev, S. Balandin, and Y. Koucheryavy, eds.), (Cham), pp. 708–720, Springer International Publishing, 2017.

[11] S. Adam, Y. Liu, A. Ahmar, S. Michiels, and D. Hughes, "Resonate - a protocol for audio transmission over low power wide area networks," in *International Journal of Design, Analysis and Tools for Integrated Circuits and Systems*, vol. 11, pp. 6 – 11, International Journal of Design, Analysis, and Tools for Integrated Circuits and Systems, 10 2022.

[12] K. Nakamura, P. Manzoni, M. Zennaro, J.-C. Cano, and C. T. Calafate, "Adding voice messages to a low-cost long-range data messaging system," in *Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good*, GoodTechs '20, (New York, NY, USA), p. 42–47, Association for Computing Machinery, 2020.

[13] J. Lyu, D. Yu, and L. Fu, "Analysis and optimization for large-scale LoRa networks: Throughput fairness and scalability," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9574–9590, 2022.