CENTER FOR CONNECTED AUTONOMY AND ARTIFICIAL INTELLIGENCE
College of Engineering and Computer Science
Florida Atlantic University
ca-ai.fau.edu

Implementation and Analysis of ExpressLRS Under Interference Using GNU Radio

Gabriel Garcia
MS Student, Center for Connected Autonomy and AI (CA-AI)
Florida Atlantic University
Department of Electrical Engineering and Computer Science
Boca Raton, FL, 33431, USA
garciag2022@fau.edu

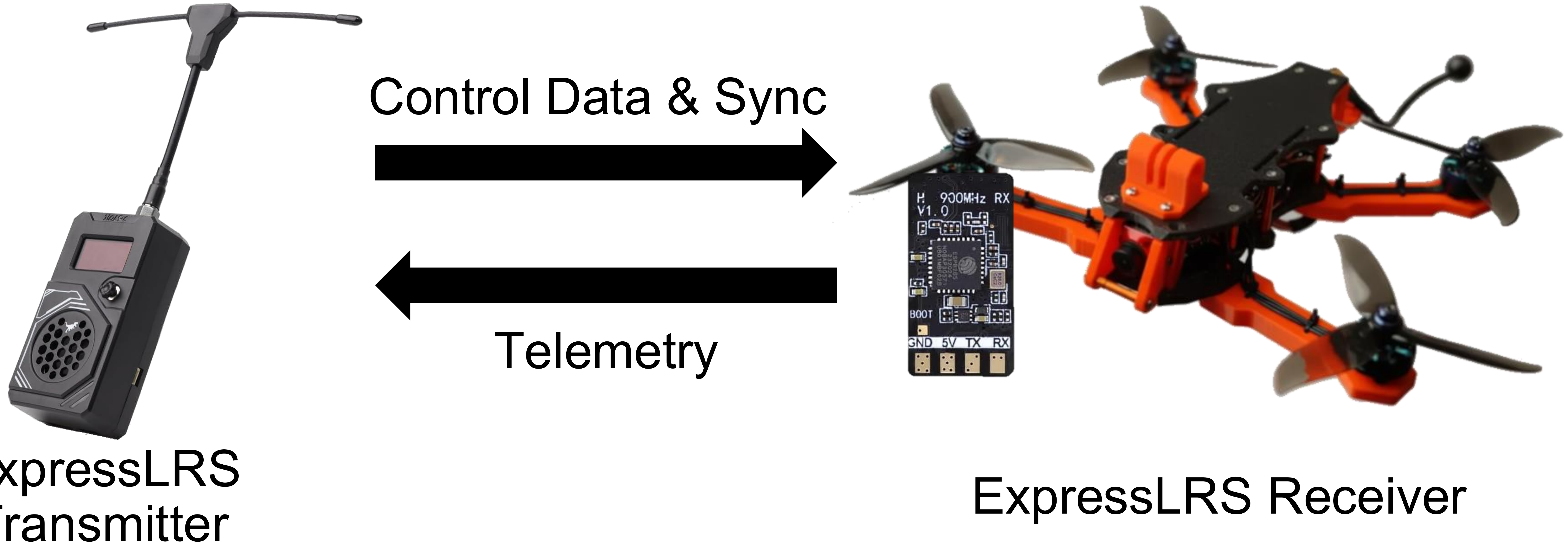
in collaboration with George Sklivanitis (FAU) and Dimitris A. Pados (FAU)

GNU Radio Conference 2025
Session: Applications of GNU Radio
September 8-12, 2025 | Everett, WA

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

Introduction to ExpressLRS

- What is ExpressLRS?
  - An open source, high packet frequency radio control link for FPV (First Person View) applications based on LoRa.
  - Designed for low latency, maximum range, and high resilience to interference.
  - Combines LoRa modulation with a frequency-hopping spread spectrum (FHSS).



Introduction to ExpressLRS (cont.)

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

ExpressLRS Transmitter

Rapidly Sends Sync Packets Over All Frequency Channels

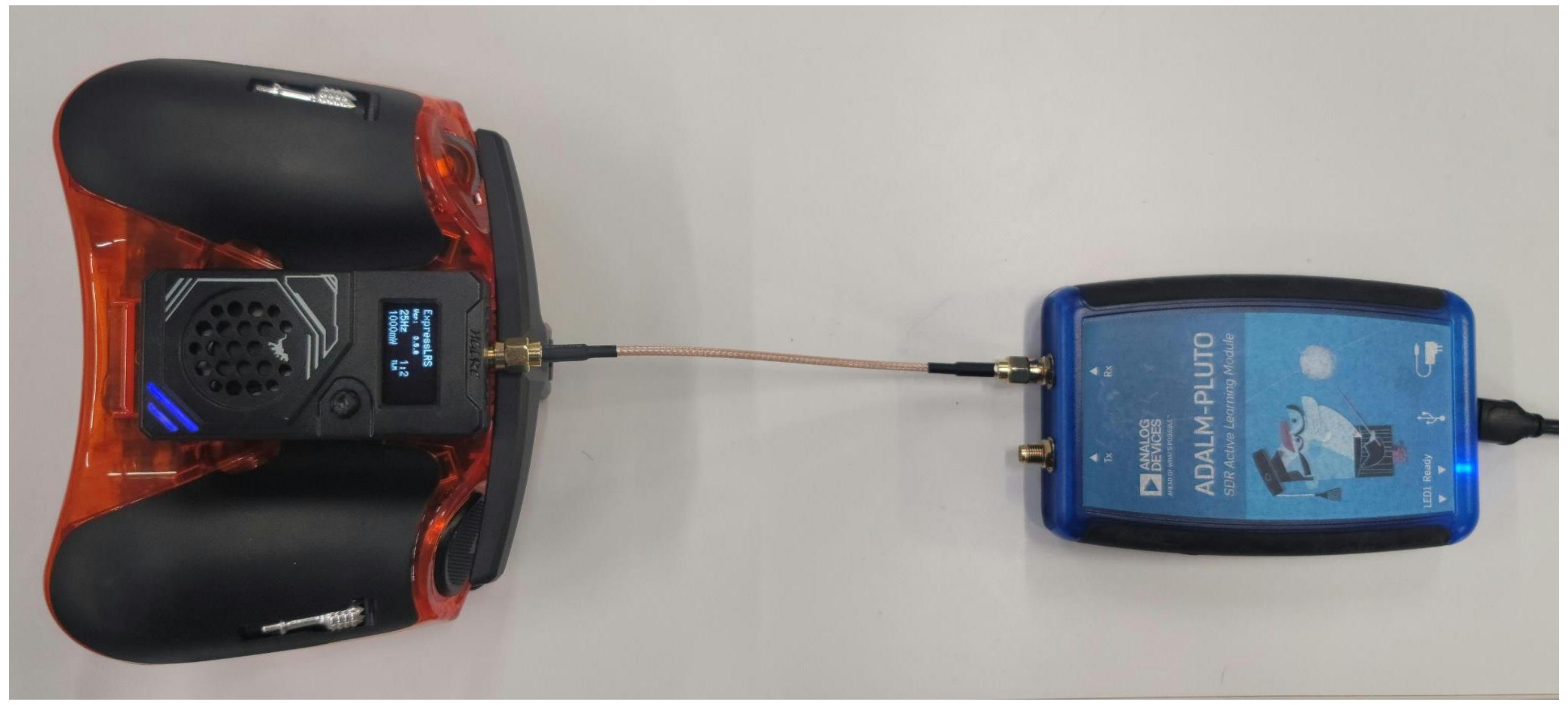
1. RX Receives Sync Packet
2. Locks Onto TX Timings
3. Sends Link Confirm Packet

Begins Sending Control Data Packets

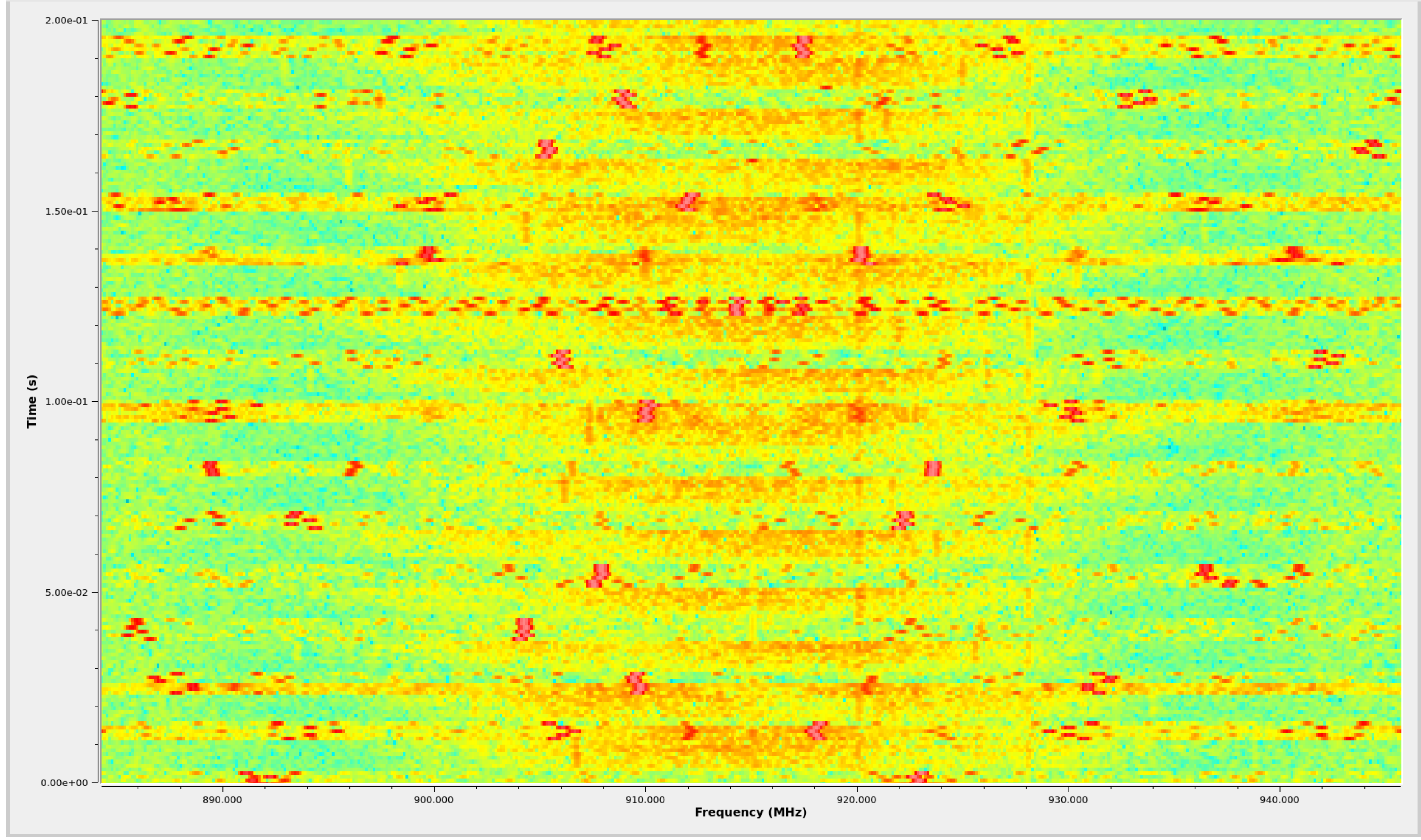
ExpressLRS Receiver

Introduction to ExpressLRS (cont.)

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science



SDR Setup Capturing a COTS 915Mhz ELRS Transmitter in its “Search Phase”

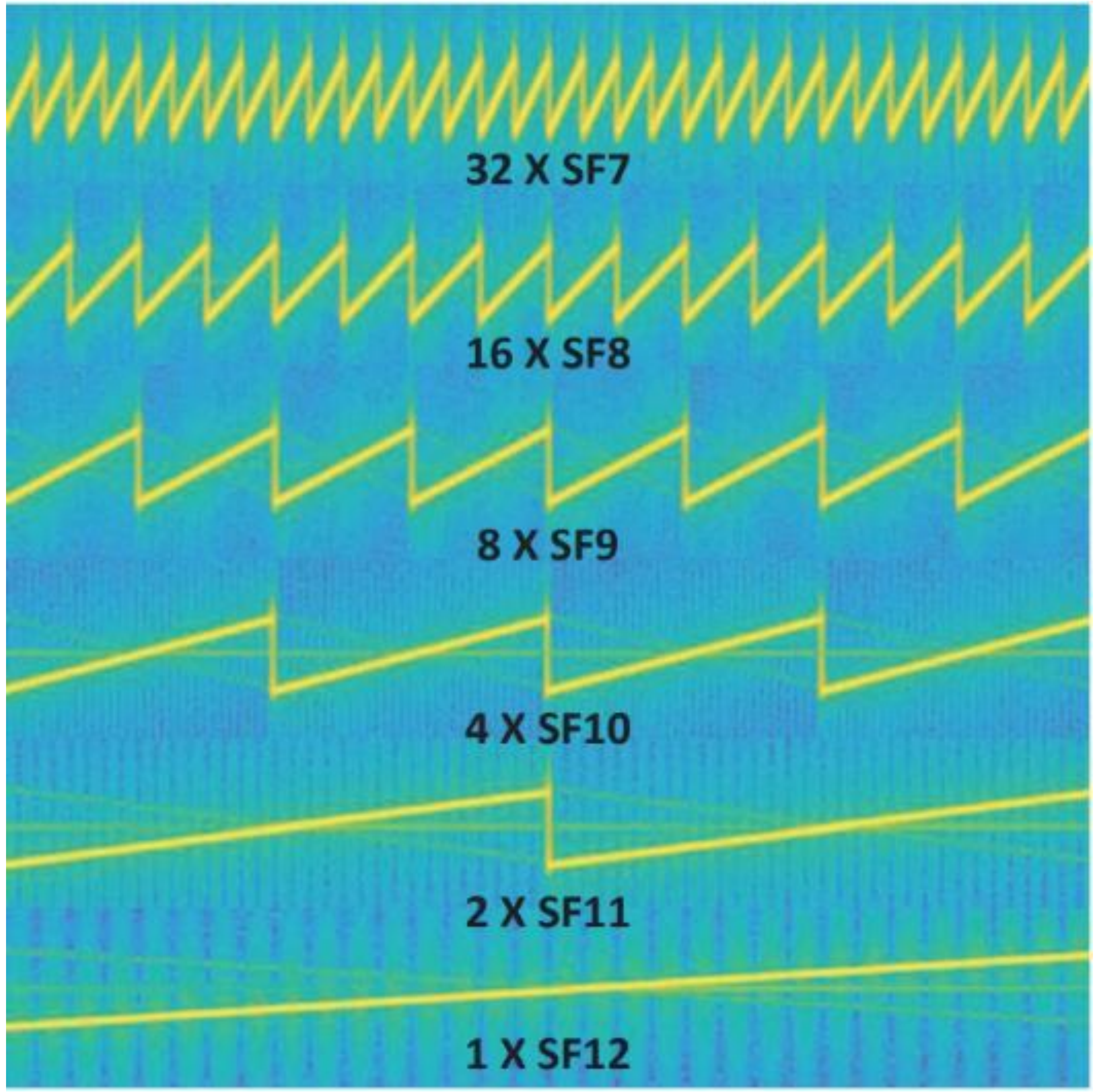


Resulting Spectrogram

Core Concepts of LoRa & FHSS

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

- LoRa (Long Range) Physical Layer
  - Developed by Semtech.
  - Utilizes **Chirp Spread Spectrum (CSS)**.
  - **Spreading Factor (SF)** is the key parameter that trades data rate for range and latency.
  - A higher SF increases each symbol's air time, making it easier to distinguish from noise.



from Borges et al., "On the Simulation of LoRaWAN Networks: A Focus on Reproducible Parameter Configuration," 2024.

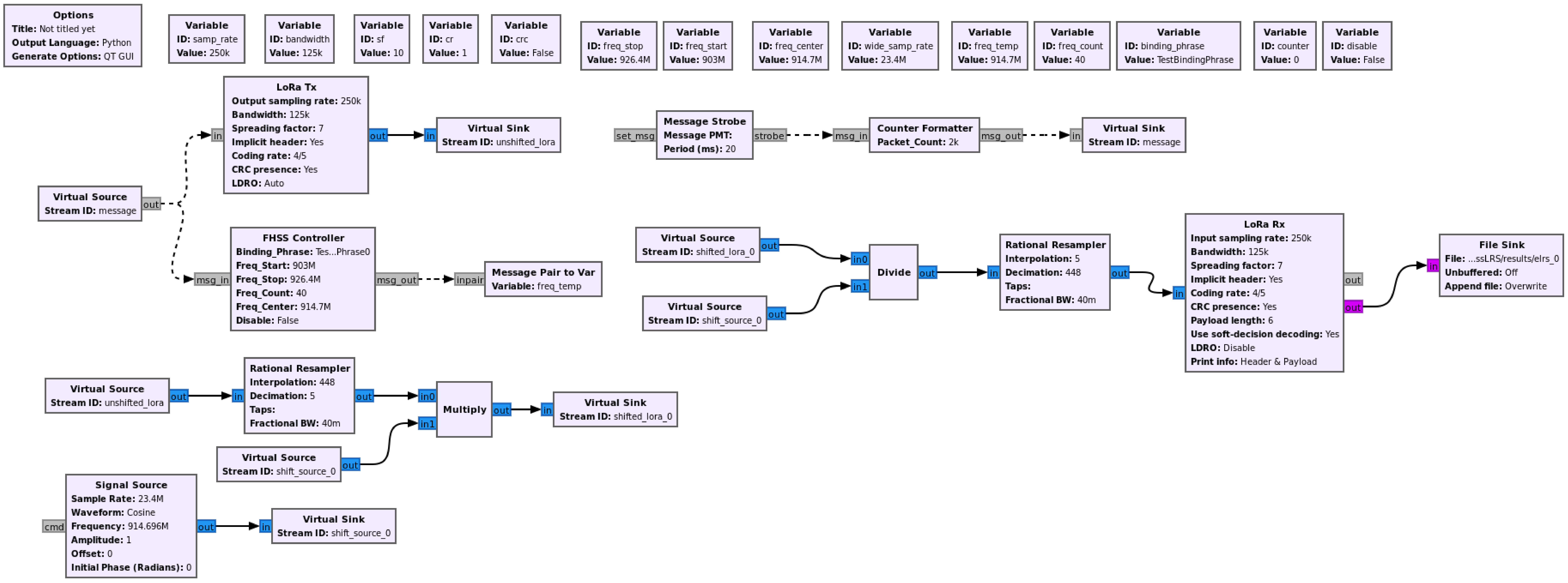
Core Concepts of LoRa & FHSS (cont.)

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

- ExpressLRS Frequency-Hopping (FHSS)
  - Employs an FHSS scheme where the transceiver's center frequency is updated at a configurable, packet-based interval.
  - The hopping sequence is generated pseudo-randomly on the TX & RX.
  - The binding phrase on TX & RX seeds the generation.
  - Sync packets from TX maintain RX's position in the sequence to counter hardware drift.

GNU Radio Implementation

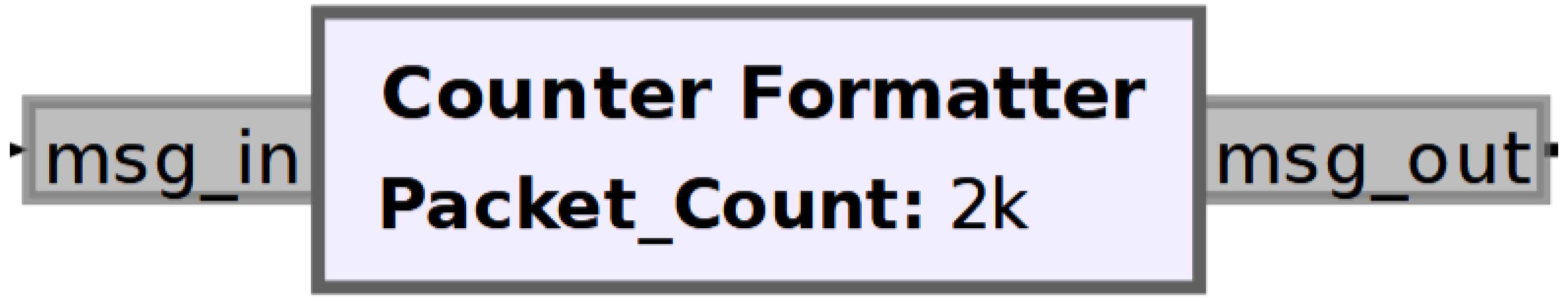
FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science



The Counter Formatter Block

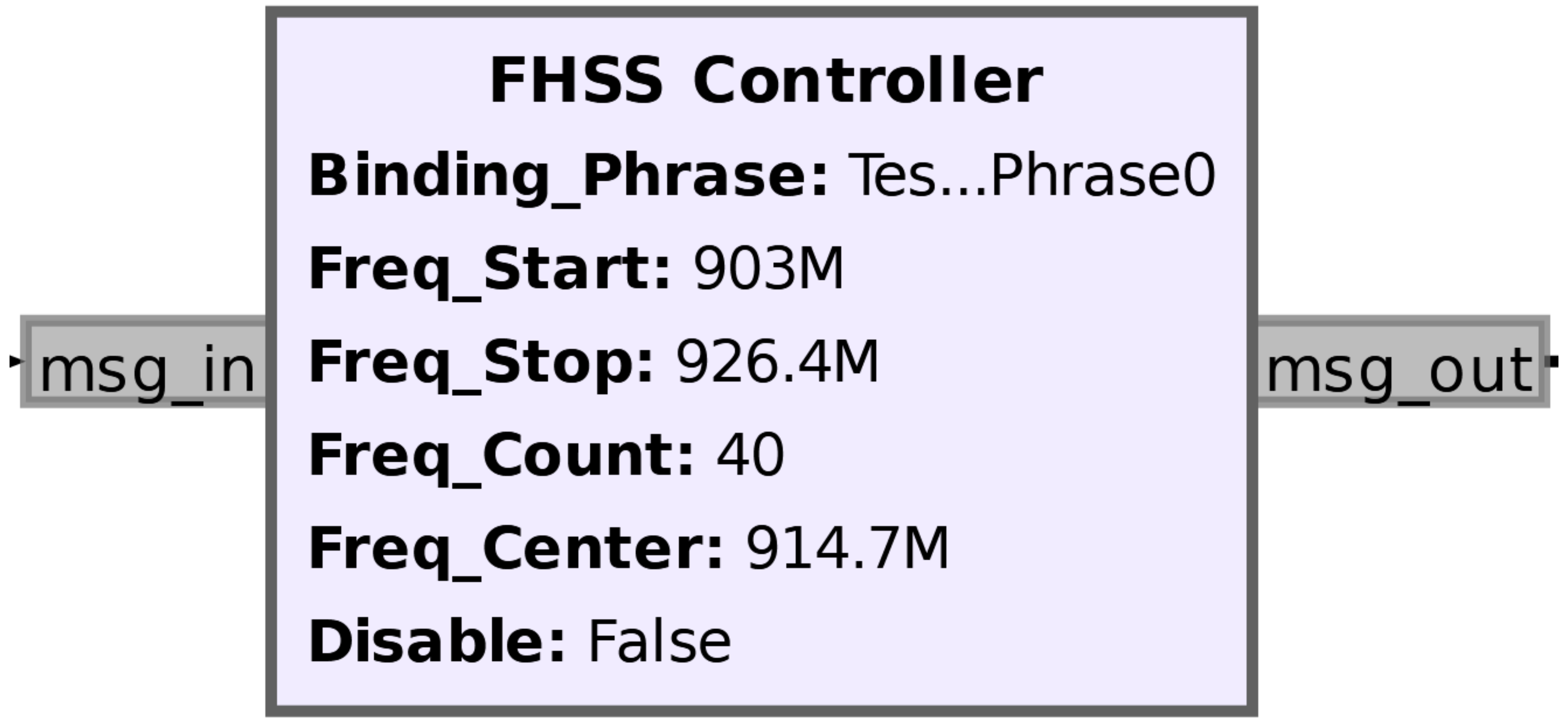
FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

- Generates sequenced 6-byte UTF-8 strings.
- Emulates the payload portion of an ELRS packet that is also 6 bytes.
  - ELRS packets can be 8 or 13 bytes each, we are emulating the 8 bytes packets.
  - The CRC portion of the ELRS packet is emulated by allowing the LoRa TX block to produce 16-bit CRCs.
- Can be configured by the user to produce up to 99,999 sequenced packets.
- Is triggered to activate by a message strobe block.



The FHSS Controller Block

- Implements the FHSS logic.
  - Accepts a binding phrase that will be used to seed the generation of the FHSS sequence.
  - Allows the user to set values to emulate existing ELRS domains or define their own.
    - Domains in ELRS are regional specific settings to comply with local regulations.
    - Settings in photo depict “FCC915”.
  - Can be disabled entirely if needed, reverting to standard LoRa at the specified center frequency.



The FHSS Controller Block (cont.)

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

- The seed generated from the MD5 hashed binding phrase seeds the RNG function.
  - The RNG function is a 16-bit Linear Congruential Generator (LCG).
- The seeded LCG is then used to perform a Fisher-Yates shuffle on a list of channel indices.
- The logic for generating the FHSS sequence is simplified compared to the original firmware due to not incorporating synchronization channels.

```
    uid_bytes = md5(str(f'-DMY_BINDING_PHRASE="{binding_phrase}"') \
        .encode('utf-8')).digest()[:6]
    self.seed = (uid_bytes[2] << 24) | (uid_bytes[3] << 16) \
        | (uid_bytes[4] << 8) | (uid_bytes[5] ^ self.OTA_VERSION_ID)
    self.freq_sequence = [0] * self.num_primary_bands
    self.fhss_index = 0

    self.build_random_fhss_sequence()

def elrs_rng(self):
    self.seed = (0x343FD * self.seed + 0x269EC3) % 0x80000000
    return self.seed >> 16

def build_random_fhss_sequence(self):
    # Only build if freq_count is valid
    if self.freq_count < 2: return

    for i in range(self.num_primary_bands):
        self.freq_sequence[i] = i % self.freq_count

    for i in range(1, self.num_primary_bands):
        if (i % self.freq_count) != 0:
            offset = (i // self.freq_count) * self.freq_count
            rand = (self.elrs_rng() % (self.freq_count - 1)) + 1
            self.freq_sequence[i], self.freq_sequence[offset + rand] = \
                self.freq_sequence[offset + rand], self.freq_sequence[i]
```

Software-In-The-Loop (SITL) Testing

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

- All tests were conducted around measuring the Packet Error Rate (PER) that a single ELRS transmitter and receiver pair experiences from interference caused by up to four other ELRS transmitters.

- A baseline measurement was taken with the individual ELRS transmitter and receiver pair before another ELRS transmitter was enabled.

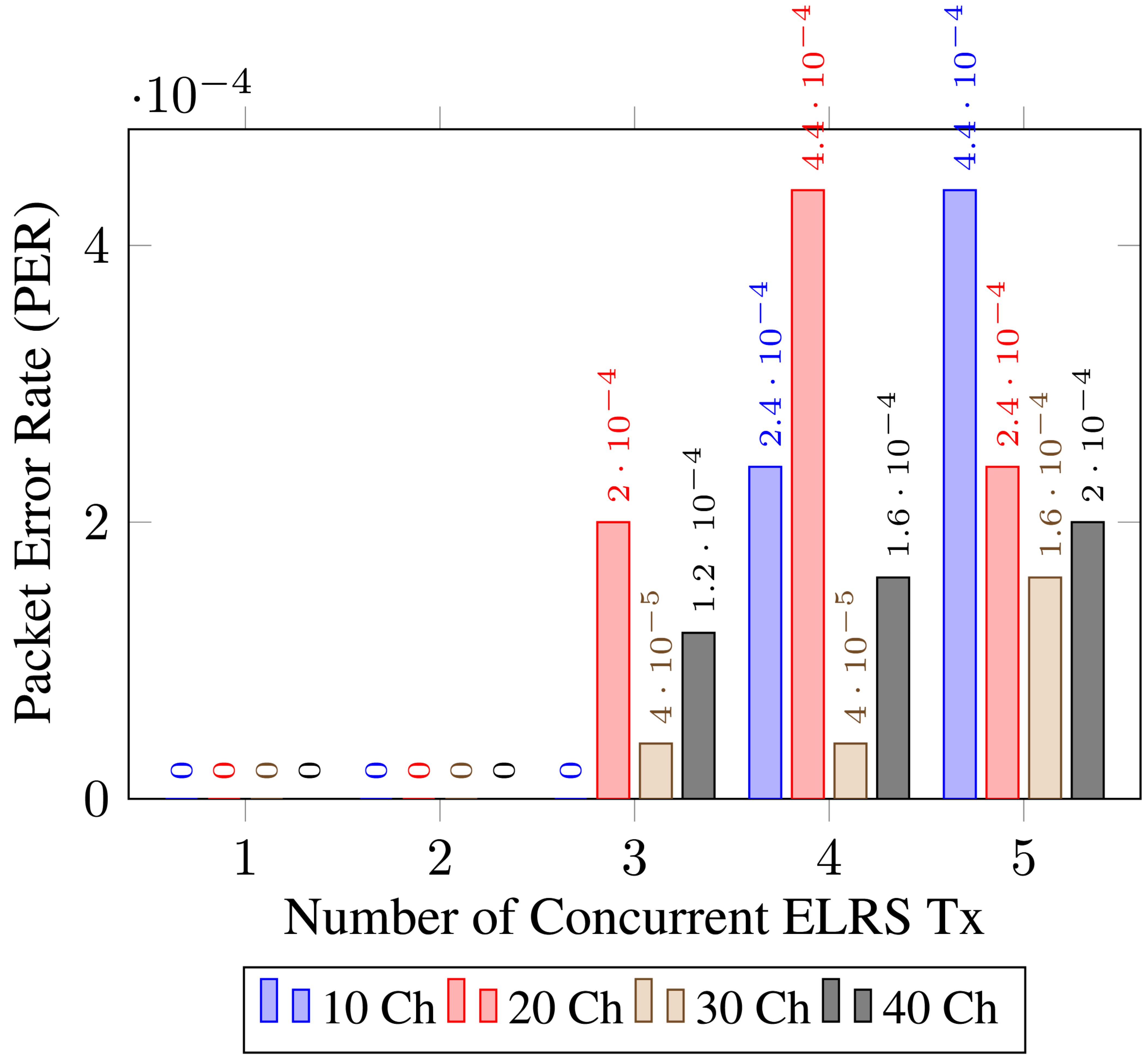
- LoRa Transceiver was configurated with a bandwidth of 125KHz, a sample rate of 250KSps, a coding rate of 4/5, and utilizes an implicit header with CRC enabled.

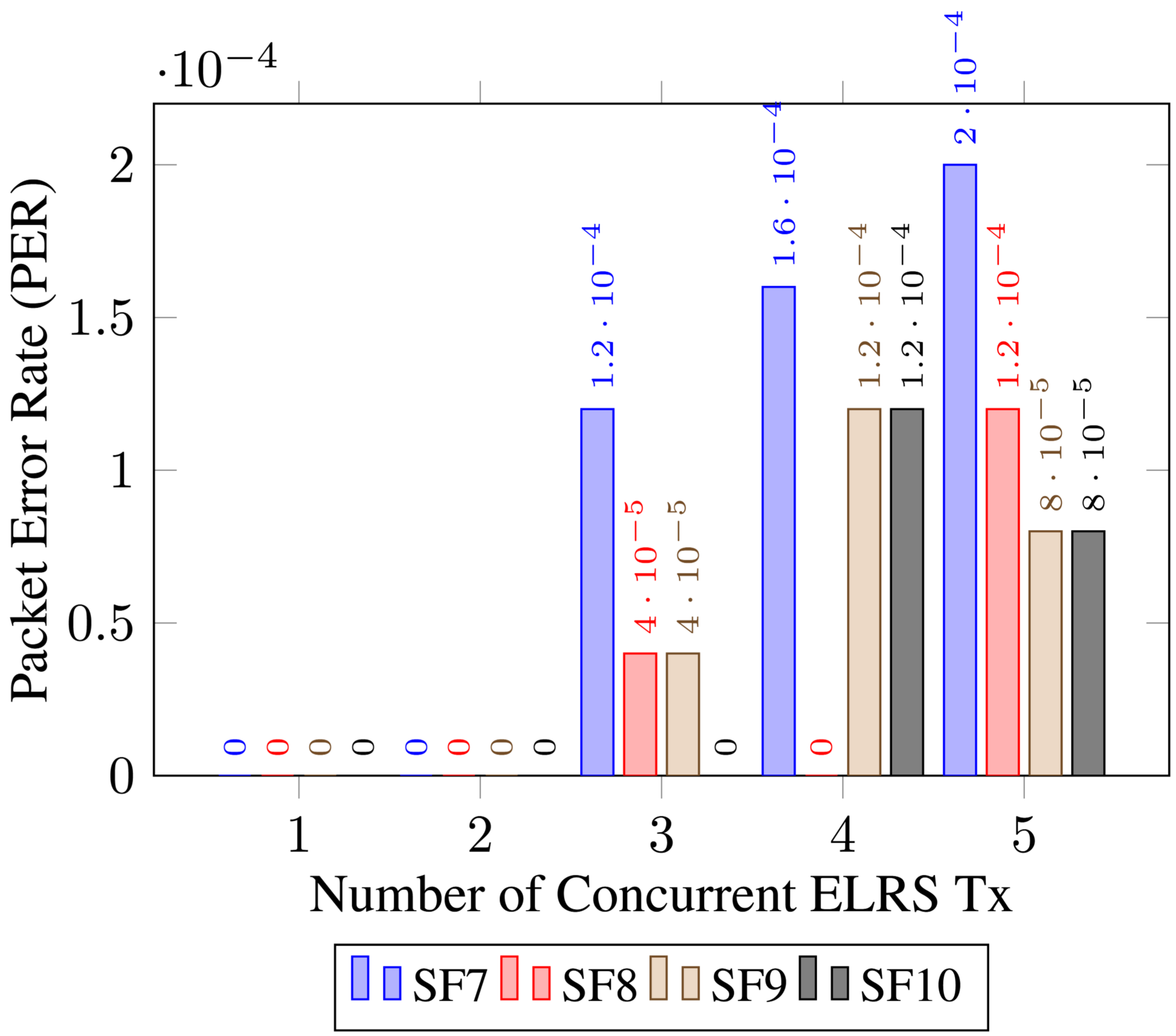
- The variables that were tested were:
    - Number of frequency channels testes: 10 through 40.
    - Spreading Factor (SF): SF7 through SF10.
    - Number of simultaneous transmitters.

- We tested each configuration over 5000 packet transmissions.

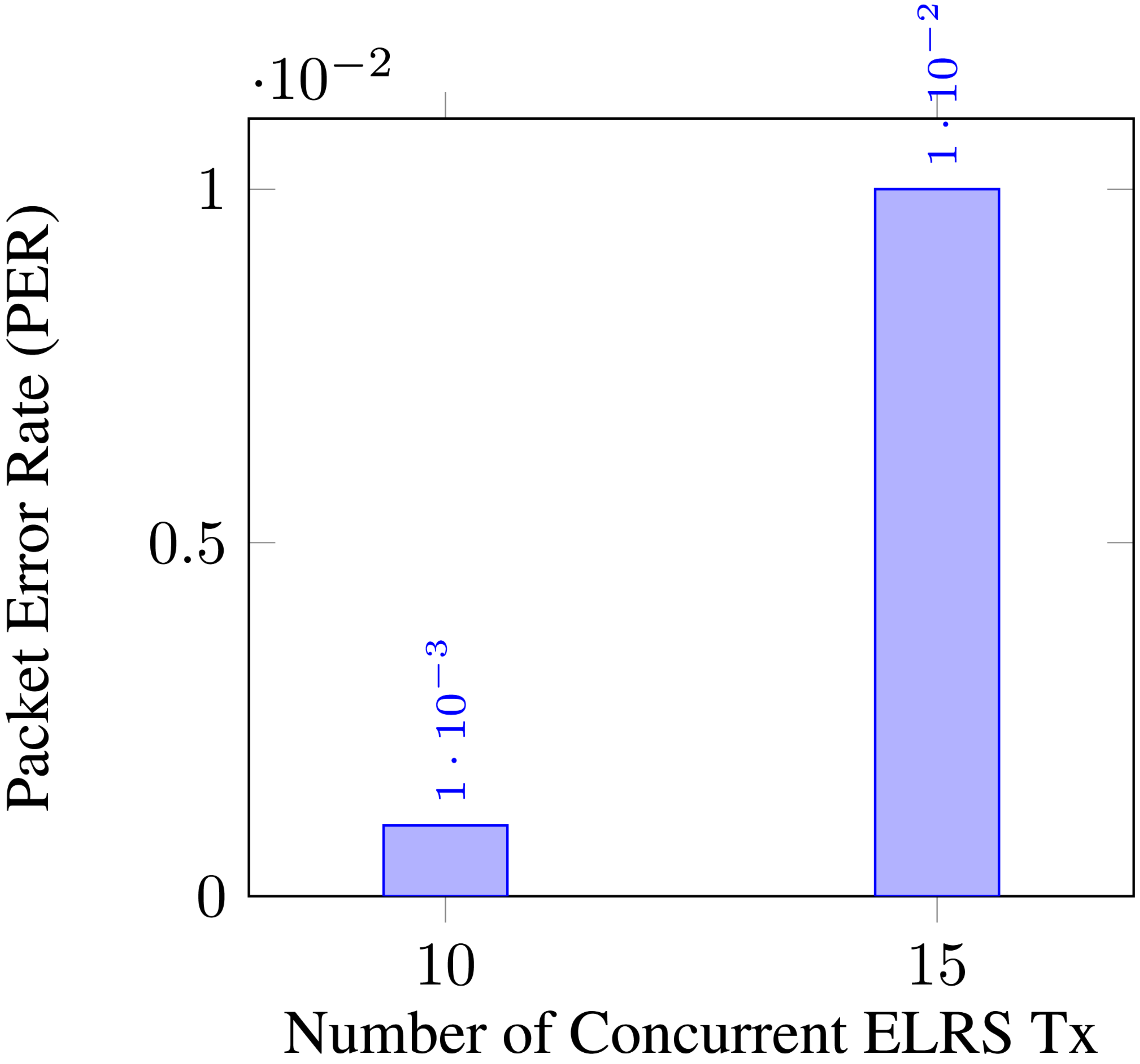
Software-In-The-Loop (SITL) Results

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science





Software-In-The-Loop (SITL) Results (cont.)



FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

Challenges and Limitations

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

- SITL Simulation Scaling Limit
  - The simulation was limited to approximately 15 transmitters.
  - High computational load led to timing jitter.
- Hardware-in-the-Loop (HITL)
  - Planned tests on ADALM-Pluto could not be completed successfully.
  - Unpredictable latency broke the precise time-alignment required to correctly de-hop the signal.

Future Work

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

- Full, more robust, implementation of ExpressLRS with advanced features such as dual diversity transmission.
- Overcome timing limitations by moving the high-level protocol simulation to a discrete-event simulation platform (e.g., MATLAB).
- Move the entire implementation to an FPGA platform to eliminate software-based timing issues and enable true real-time performance.

Demo

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

- https://youtu.be/D0aBrkvM-GE

Thank you! Questions

FLORIDA ATLANTIC UNIVERSITY
Center for Connected Autonomy and Artificial Intelligence
College of Engineering & Computer Science

- Data and source code is available under:

https://github.com/Diamond-D0gs/GNU_Radio_ExpressLRS

References

[1] Borges, M. V., Souza, R. D., Damasceno, H. G., & Dias, U. S. (2024). *On the Simulation of LoRaWAN Networks: A Focus on Reproducible Parameter Configuration*. ResearchGate. Technical Report.

[2] Joachim, T., & Burg, A. (2024). Design and implementation of lora physical layer in gnu radio. *Proceedings of the GNU Radio Conference*, *9*(1). Retrieved from https://pubs.gnuradio.org/index.php/grcon/article/view/145