Using FPGAs for Spectrum Sensing and Modulation Recognition Project

Group Members:
Ryan Davis
Zhuohuan Li
Sid Mandayam
Jacob Morin
Advisor: Richard Martin

Date: 07/30/2020

Ryan Davis

Class of 2021
Rutgers University
Computer Engineering
and Computer Science

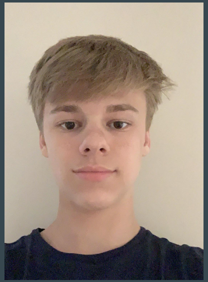
Zhuohuan Li

Class of 2020
Rutgers University

Computer Engineering

Sid Mandayam

Class of 2022
Rutgers University
Computer Science and
Mathematics



Jacob Morin

Class of 2021
Pingry High School

A Little Background

- Modulation Scheme
  - The “language” a device speaks
  - Will tell us how a device communicates
- IQ Data
  - The “noises” coming out of the device’s mouth
  - Putting these “noises” together in a controlled fashion makes a “language”
  - Putting IQ samples together in a controlled fashion makes a modulation scheme
- SDRs (Software Defined Radios)
  - The “ears” that hear the “noises”
  - Will listen to the radio environment and jot down what it hears as IQ Data
- FPGA (Field Programmable Gate Array)
  - Most wireless devices have very specific hardware that allows them to communicate
  - Regular computers can approximate the performance of this hardware
  - FPGAs can do this better

Project Overview

- Project seeks to use machine learning to recognize different wireless devices
- Use software defined radios (SDRs) to record IQ samples from various wireless devices
- Use IQ samples as training data for neural networks
- Train neural networks to distinguish between modulation schemes
- Classify device based on its modulation scheme

Data Collection

Goal: Mimic wireless transmissions in a (mostly) controlled environment to be used later as training data for a modulation recognition neural network.

Purpose	Resource
Physical Transmission / Reception	USRP X310, B210
WiFi 802.11n IQ Sample Generation	MATLAB Wireless Toolbox
Transmit / Receive IQ Samples	UHD Sample Scripts
Experiment Management	OEDL

USRP = Universal Software Radio Peripheral (a type of SDR)
UHD = USRP Hardware Driver
OEDL = Orbit Experiment Description Language

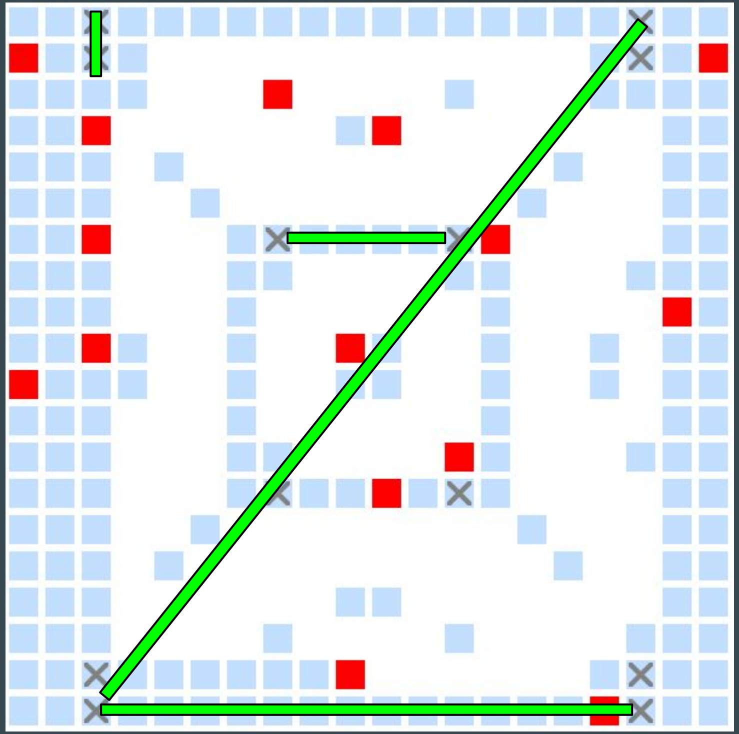
The Experiment

Constants

Transmitter Bandwidth	20 Mhz
Receiver Sampling Rate	40 MSps
Input/Output Binary File Format	int16
Packet Payload Size	1500 Bytes

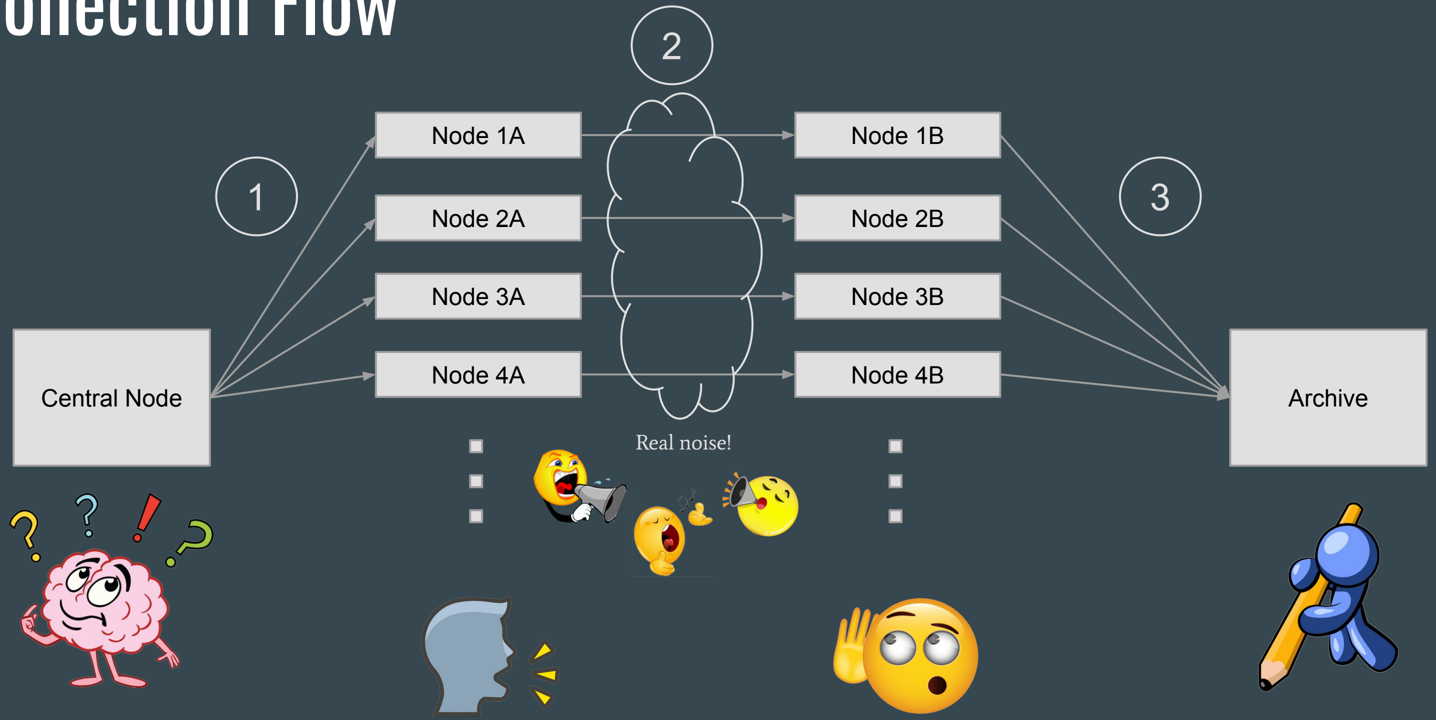
Variables

Distance (ft)	3, 15, 45, 72
Frequency (Mhz)	2412, 2437, 2462, 5180, 5240, 5745, 5825
Modulation and Coding Scheme (MCS)	0, 1, 2, 3, 4, 5, 6, 7



Data Collection Flow

Node = Computer



1 Copy WiFi IQ sample files to set “A” nodes (transmitters)

2 Transmit WiFi IQ sample files via USRP to set “B” nodes (receivers)

3 Copy received WiFi IQ sample files to archival storage

* Each A-B node pair represents a topology defined by the physical distance between the nodes

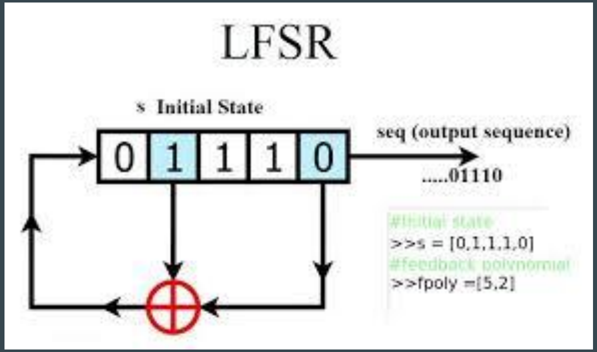
Matched Filter

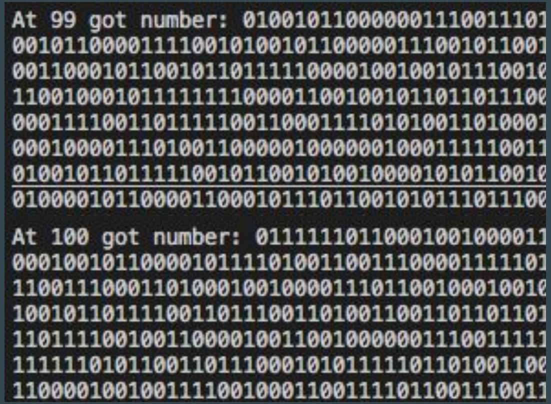
- What it is:
  - Detect if a “template” signal is present in a noisy, unknown signal (Do they *match?*)
  - Used to determine if the unknown signal follows a particular modulation scheme
- What it is to us:
  - Neural Network Based Modulation Recognition
    - Requires very little a priori information
    - Used for signals we know very little about
  - Matched Filter
    - Requires a priori information (cannot work without it)
    - Used for signals we have decent knowledge of
  - Performance Comparison

*Neural Network: A system used to cluster and classify the data that you store and manage

Linear Feedback Shift Register

- Pseudo-random number generator
  - Generates sequences of 0’s and 1’s based off inputs
- Creates the random data
  - Each signal has its own sequence
- Can use matched filter to detect patterns in the sequences and compare signals





Future Work

- Using Matched Filter to ID Modulation Schemes for Random Sequences
  - Get random data using number generator (i.e. 0100011101010.....)
  - Create 8-bit sequence + modulation to use for matched filter (i.e. 01110101)
  - Use matched filter to find hits with the random data

0100011101010..... → 01110101 → 0100 01110101 0.....

01110101 !

Random data using number generator

8-bit sequence used as matched filter

8-bit sequence used as matched filter

Future Work (cont.)

- Use Matched Filter in Conjunction with Neural Net Data and FPGA to Classify Devices
  - Compare Effectiveness of Matched Filters vs Machine Learning
  - Transition to Making Matched Filter in Go (different programming language)
  - Compile Go Filter into Verilog using argo2verilog compiler to be used with FPGA
- Matlab
  - Access to many built-in signal processing functions, easier to compare results of MF versus ML in Matlab
- Go
  - Higher level language, use of goroutines allow programs to run in parallel and faster
- Verilog
  - Can be used with FPGA to detect RF signals of various devices

Questions?