ECE430.
Laura and Joshua
Radio Wars Ep 2

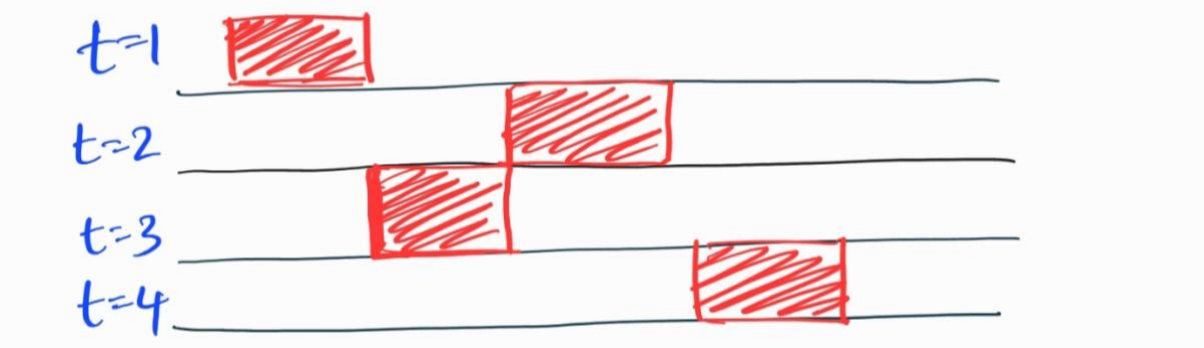

Figure 1: frequency hopping procedure explained

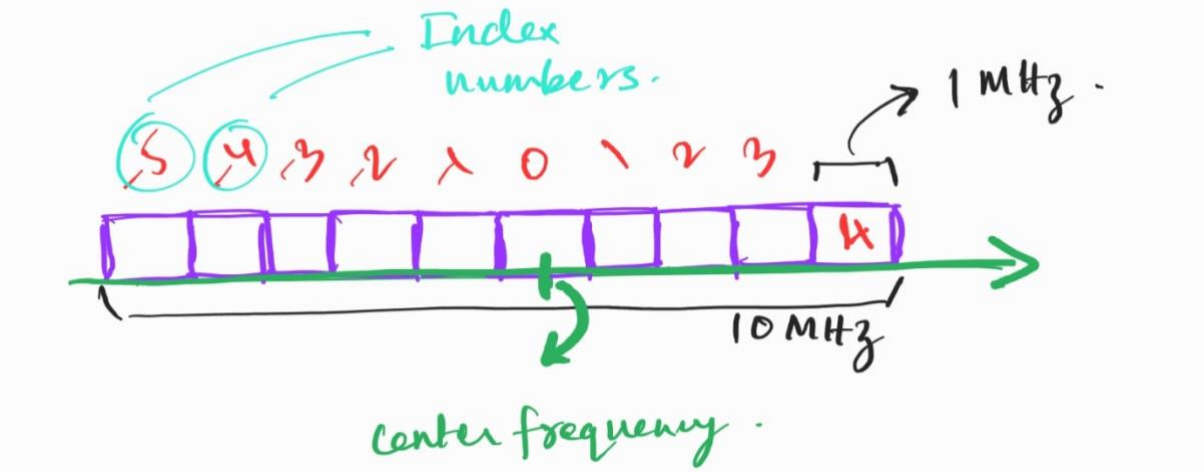

Figure 2: frequency hopping design used for Radio Wars

Our Implementation:

i. Initialize center frequency
ii. Create 10 slots (-5 to 4)
iii. Make the center frequency adjust itself using the generated slots
iv. Select one random carrier frequency each time and then repeat this procedure to get a unique carrier frequency each time (after a 1 second wait period).

This producer allows for our carrier frequency to be constantly changing; therefore, reducing the probability of our transmission being blocked by a jammer.

Seed used: random.seed(round(time.time(),-5))
This is because, as seeds both at the transmitter and receiver needs to be same for synchronization between the frequency hopping bands, we have rounded the time.time() so that we get the date, which is bound to be same at both the transmitter and receiver.

```
    return sched

async def frequencyHop(self):
    try:
        while True:
            # Uncomment to set frequency to 1.31 GHz every second. Please do
            # something more intelligent!
            #self.frequency = 1.34e9

            center_freq =1.34e9
            slots = np.arange(-5,5)
            freq_slots = (slots*1e6) + center_freq
            self.frequency = freq_slots[random.randint(0,9)]
            print(self.frequency)

            # Sleep for 1 second
            await asyncio.sleep(1)
    except CancelledError:
        pass

main():
```

Figure 3: frequency hopping function

Figure 4: Tx and Rx commands

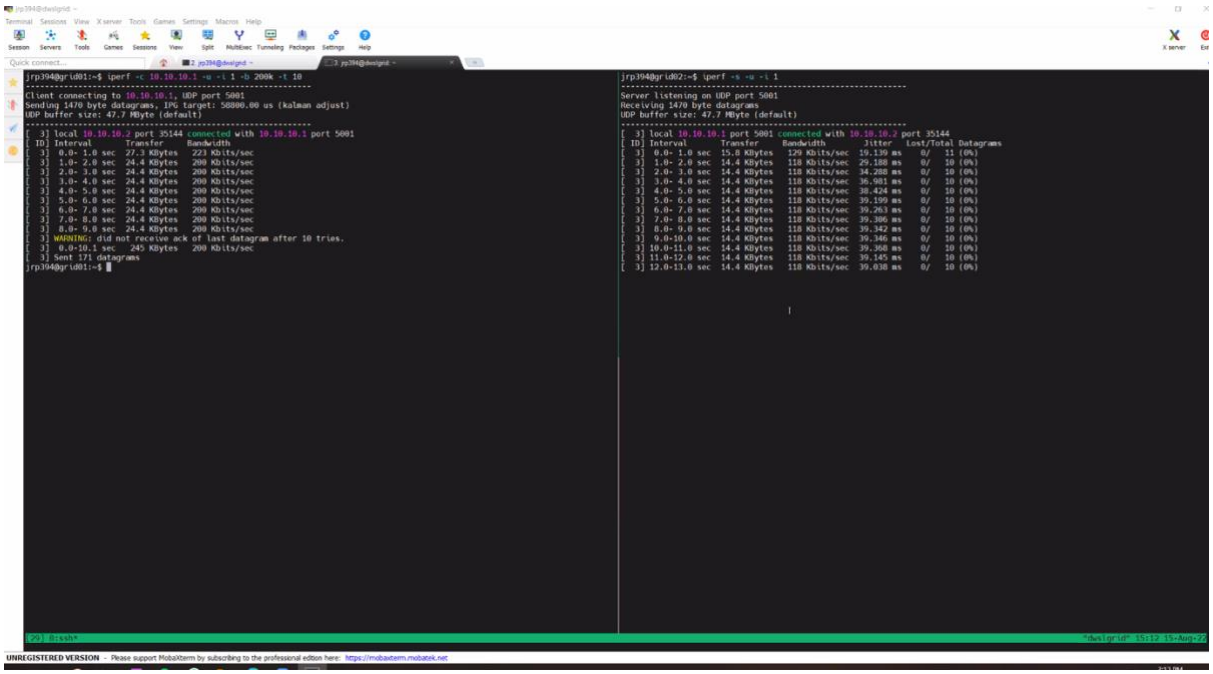

Figure 5: Tx and Rx are communicating

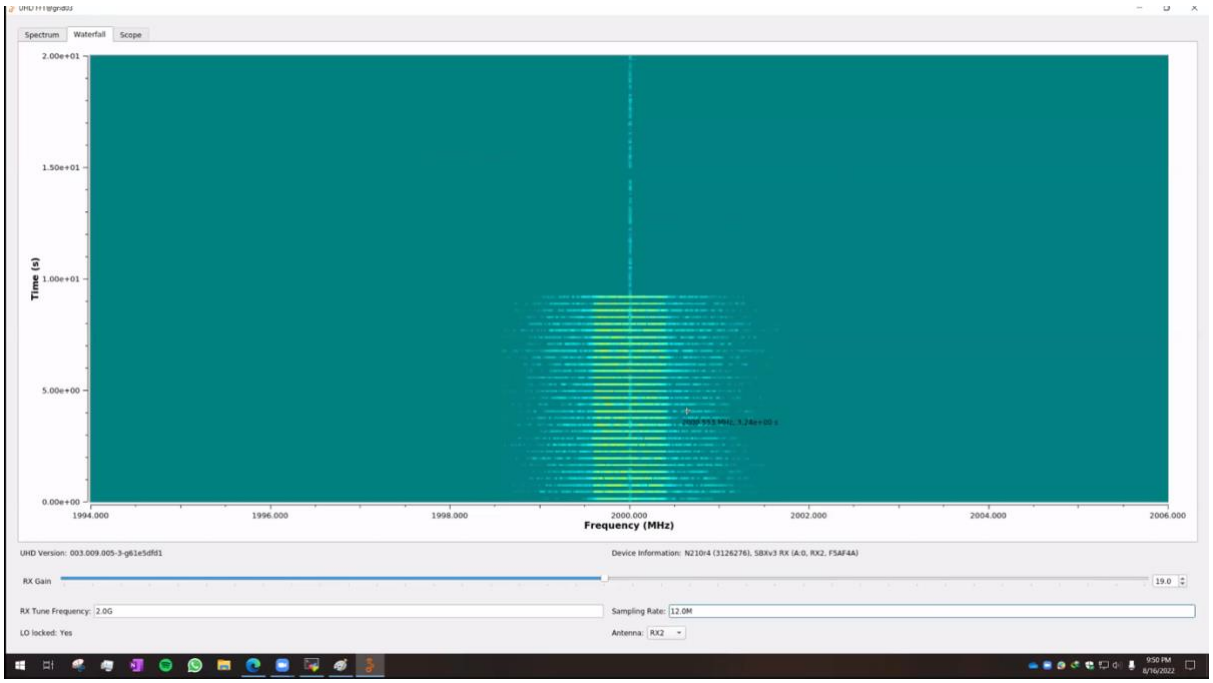

Figure 6: Transmission without frequency hopping enabled

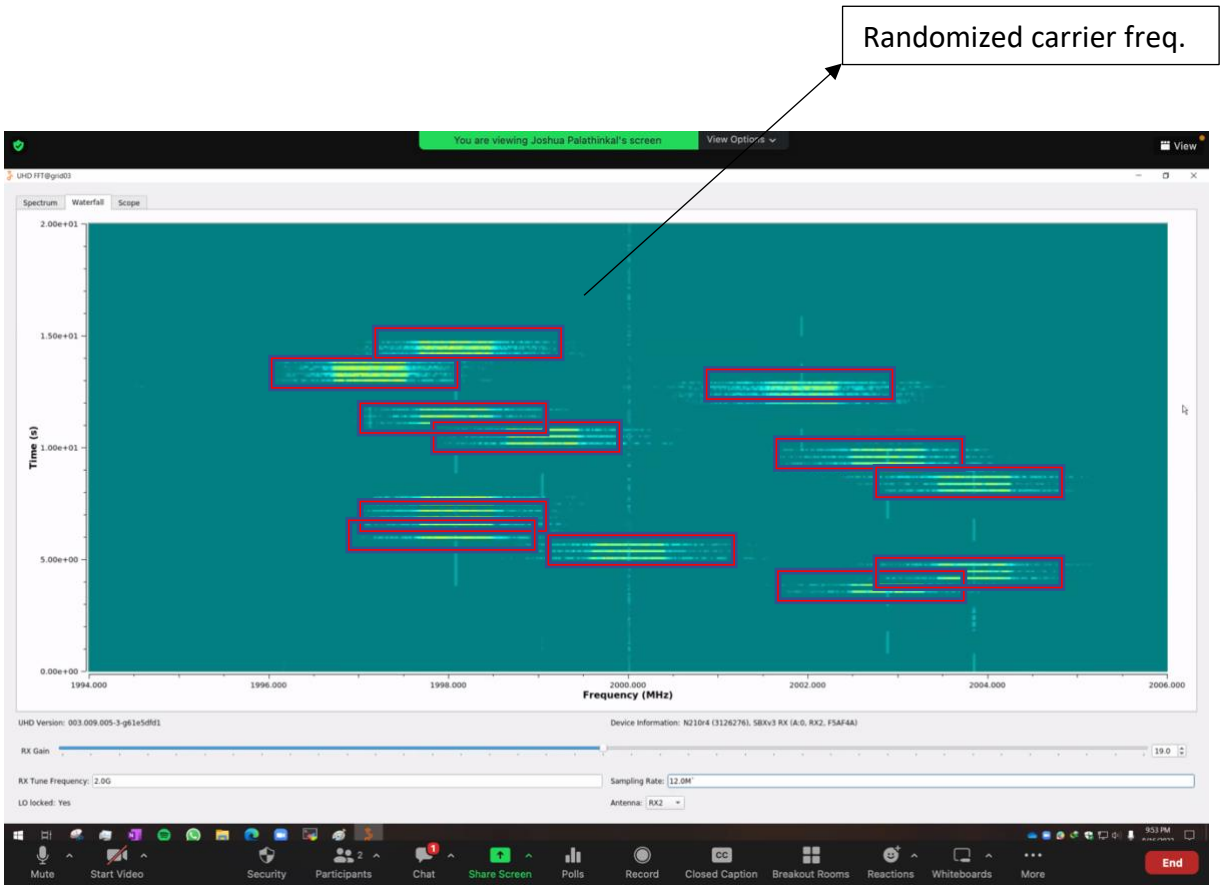

Figure 7: transmission with frequency hopping enabled

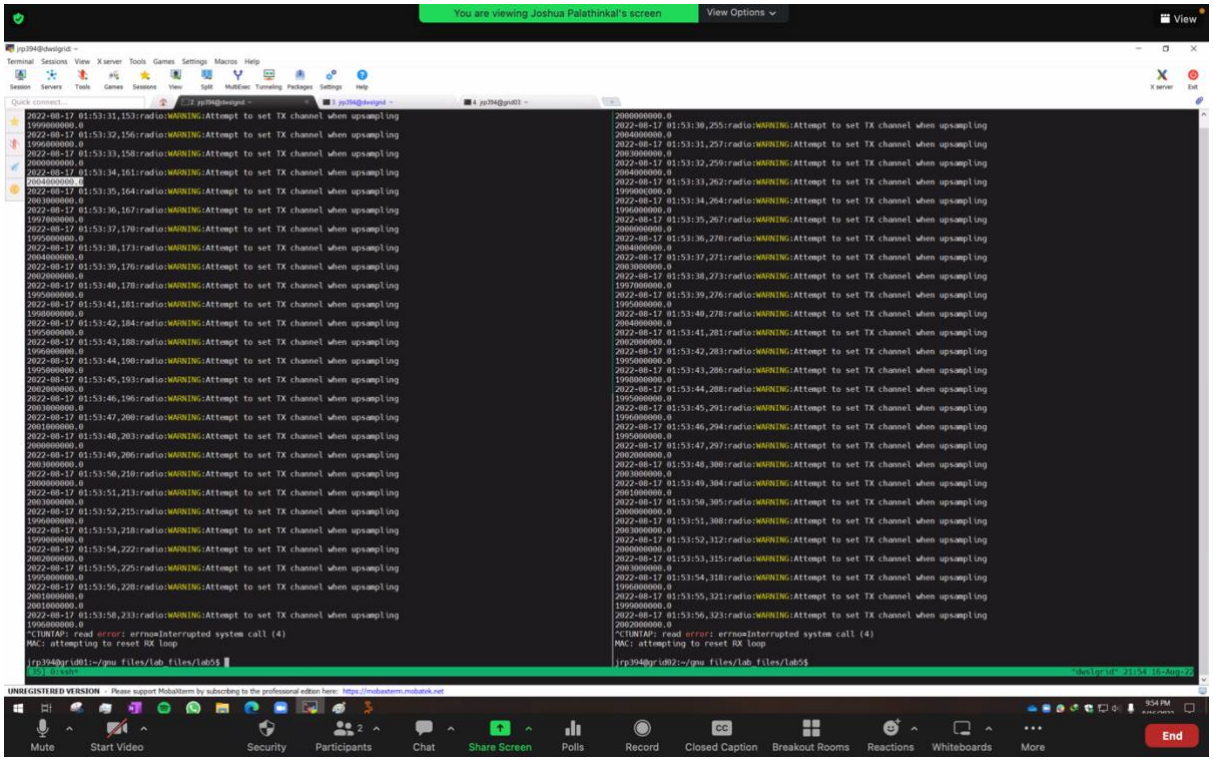

Figure 8: Frequencies taken at Tx and Rx

It can be observed that the frequencies assigned at both Tx and Rx are not in sync despite using the seed intelligently. One of the ways in which this can be mitigated is to start the WHILE loop for the FrequencyHop function after both Tx and Rx reached the next minute (seconds = 00) from when it was started. This'll be implemented for the next Radio Wars.