Boston University
Electrical & Computer Engineering
EC 463 Senior Design Project

First Prototype Test Report

Over-the-Air C-V2X Communications on Software Defined Radios

by

Team 13
C-V2X Misbehavior Detection System

Team Members

Michael Aliberti mjali@bu.edu
Max Ellsworth maxell@bu.edu
Jason Inirio jasonini@bu.edu
Sam Krasnoff krasnoff@bu.edu
Julia Zeng zjulia@bu.edu
Yixiu Zhu zhuyixiu@bu.edu

Required Materials

Hardware:

- 3 x DragonOS (Lubuntu) PCs
    - 8 Cores
    - 16 gb RAM
- 3 x NI 2901 USRP
- 2 x Wall power adapters
- 5 x Vert 2450 Radio Antena
- 2 x Keysight 33500B Waveform Generator
    - 1 x 1 MHz, Pulse Wave
    - 1 x 10 MHz, Square Wave

Software:

- SrsRAN Library
    - SrsEPC
    - SrsENB
    - SrsUE
- GQRX
- Bash commands
    - ping
    - iperf

Set Up

At the core of this project are three NI 2901 USRPs, which are connected to a variety of devices for efficient, stable communication. Two of the USRPs will model end-to-end communication, and the third will monitor network traffic. Each of the communicating USRPs is plugged into the wall to get a consistent 1A, as propagating a signal in the 5.9 GHz range requires a high, consistent flow of power. All of the USRPs are connected to our desktop computers running DragonOS, an Lubuntu variant that specializes in radio software. This USB connection enables us to use the SrsRAN library to configure the radios through software. From here, everything from the frequency band to the envelope itself can be digitally sent to the SDR. The base station and user end USRPs will be connected to both a signal generator at 10 MHz and a signal generator at 1 Hz to synchronize communications. The base station SDR will run SrsENB and SrsEPC to serve as the virtual network base for the other USRP devices to connect to. The user end will run SrsUE, functioning as a communicating node to the base station. The third USRP will currently function as our data collecting node, monitoring the frequency on which the downlink and uplink are situated.

Pre-testing Setup Procedure

1. Connect the Base Station and User End USRPs to power sources and separate computers.
2. Ensure these USRPs have been properly connected and are visible to the computer by running `uhd_find_devices` on the command-line. Look for the device name “b200.”
3. Configure signal generators with the following settings:
    a. A “Pulse, Off, 50 Ohm” wave that is an “AM modulated by sine” wave, frequency of 1 Hertz, amplitude of 10 dBm.
    b. A “Square, Off, 50 Ohm” wave that is an “AM modulated by sine” wave, frequency of 10 Megahertz, amplitude of 100 millivolts (this should probably be 10 dBm).
4. Connect the base station and user end USRPs to both signal generators. Ensure that each signal generator is connected to the same relative input on each USRP.
5. Open up a terminal on the base station; run `srsepc` and `srsenb`.
6. Open a terminal on the user end and run `srsue`.
7. Activate the trace on the User End and Base Station.

8. Connect the monitor USRP to the computer hosting the Base Station.
9. Open Wireshark on the PC attracted to the monitor.
10. Open GQRX on the PC attached to the monitor. Configure it to read data from the monitor by checking “Ettus B200” in “File => I/O Devices” and typing “ctrl+D.”
11. Flag uplink frequency 2.560.000.000 and downlink frequency 2.680.000.000 for ease of access during testing.

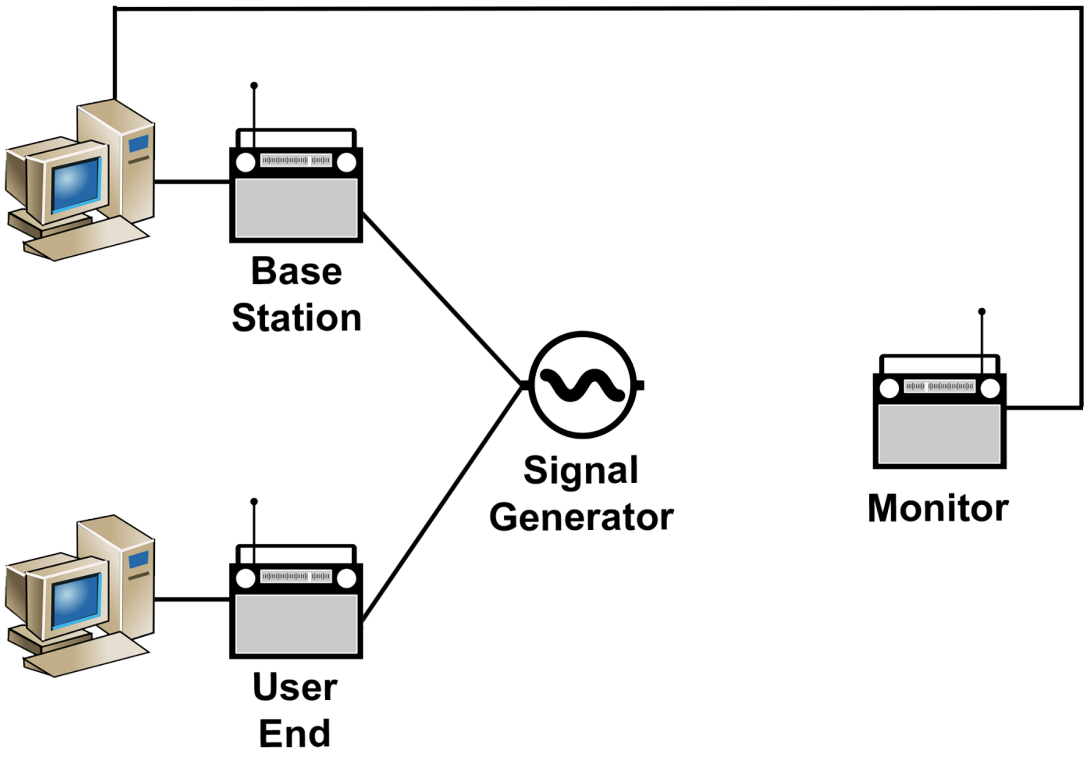

Figure 1: Illustration of Setup

Testing Procedure

1. Demonstrate uplink communication between USRPS using the SRS radio suite.
    a. Run `ifconfig` on the Base Station and record the ENB IP.
    b. Run `ping {ENB IP}` on the User End.
    c. Run on the `iperf -s` Base Station and `iperf -c {ENB IP} -t 20s` on the User End
    d. Repeat Step 1c with `iperf -c {ENB IP} -P 20 -t 20s` on the User End.

2. Demonstrate downlink communication between USRPS using the SRS radio suite.
    a. Repeat steps 1a-1d, but reverse the roles of the User End and Base Station

Troubleshooting Procedure

There are errors that can arise with connectivity when running the demo. A few are highlighted below.

1. "/usr/src/srsRAN/lib/src/phy/rf/rf_uhd_imp.cc:1335: USRP reported the following error: EnvironmentError: IOError: usb rx6 transfer status: LIBUSB_TRANSFER_ERROR" when running the base station command.
2. "Connect failed: Operation now in progress" when running iperf.
3. The iperf/ping statistics are returning 0 for the bitrate (`brate`). This indicates that the UE (user equipment) and ENB (base station) are not connected. This can occur if the system is left idle for a period of time, e.g. no iperf/ping program running.

To resolve these issues, try the following:

1. Teardown: Kill the terminal processes in the order of `srsue`, `srsenb`, `srsepc` and restart them in reverse order.
2. If step (a) does not work, power cycle the USRPs (turn them off and unplug the USB and power cables).

Measurable Criteria

1. The Base Station and User End USRPs should be able to communicate both uplink and downlink when running ping and iperf.
    a. The ENB and UE should output that a connection has been established.
    b. There should be incoming and outgoing ICMP/TCP packets on Wireshark addressed to and from the ENB and UE.
    c. There should be nonzero bitrates in the ENB and UE trace output.
    d. Uplink bitrate should be higher for uplink communication and vice versa.
    e. Both the red and green LEDs should be on for both USRPs.
2. The Monitor USRP should be able to see traffic between the ENB and UE.
    a. GQRX should see heightened activity on the uplink frequency (2.560.000.000 Hz) and downlink frequency (2.680.000.000 Hz) during ping and iperf execution.

Score Sheet

Test	Connected	Wireshark Packets	Nonzero Trace	Bitrate Order	Red LED	Green LED	Monitor (UL)	Monitor (DL)
Ping (UL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Iperf, 1 connection (UL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Iperf, 20 connections (UL)	Temp	Temp	Temp	Temp	Yes	Yes	Temp	Temp
Ping (DL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Iperf, 1 connection (DL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Iperf, 20 connections (DL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Conclusion

The majority of tests passed with no issue. When executing ping and singly-connected iperf, both the uplink and downlink tests exhibited the expected behaviors, with the UE and ENB issuing “connected” status messages, the USRP transmission and reception LEDs turning on, and both ends recording nonzero bitrates aligned in the direction of traffic. The monitor was able to see communications between the two radios in both packet form via Wireshark and as radio activity on dedicated frequencies via GQRX. The same can be said for multiply-connected iperf in the downlink connection, though the activity on GQRX exceeded the range of the expected frequency band.

The only test that deviated from expectations was multiply-connected iperf in the uplink direction, in which a disconnection during transmission occurred due to the flood of data from the UE. The error was informative for visualizing what a waterfall graph on GQRX looks like during a DoS attack: the entire frequency band was occupied (yellow) and hijacks resources from other users. This can be classified as a basic transport layer DoS attack that also targets the limitations of the USRP hardware. We suspect this is due to a buffer overflow on the base station end, because the error occurs when the TCP window size is set to 2.0 MBytes, but is fine when the window is 85 KByte. One such explanation for this is that due to the large data transfer, a buffer somewhere in the USB to DMA FIFO to FPGA overflowed.

Our setup also changed post-testing in reaction to feedback from Professor Hirsch. The signal generators used to synchronize the UE and ENB previously used a square wave and a pulse with sine modulation. Following the test battery, the square wave was changed to a sine wave, and modulation was removed from both signal generators. After further testing, this has not appeared to have an impact on reliability or clarity of transmission. The disconnection error described above persists, but appears to be relatively rare and is not exclusive to the case of downlink communication. Fortunately, the volume of transmission demanded in multiply-connected iperf exceeds the demands of C-V2X, so this error should not occur in the final design.