Performance Measurement of CRTS

Kalyani Gadgil

March 5, 2019

1 Hardware

The software radios platform used in CRTS(v.2) is the USRP X310. The hardware architecture of X310 combines two extended-bandwidth daughterboard slots covering DC – 6 GHz with up to 160 MHz of baseband bandwidth, multiple high-speed interface options (PCIe, dual 10 GigE, dual 1 GigE), and a large user-programmable Kintex-7 FPGA in a convenient desktop or rack-mountable half-wide 1U form factor. [2]

Figure 1 shows a general USRP architecture. The bandwidth of the USRP device varies at each point in this signal chain. [1]

USRPs can be connected in the configurations shown in Figure. 3. In the current setup, they are connected via the 10 Gigabit Ethernet Interface to the equivalent ports of a Juniper's QFX5100-96S Device as shown in Fig.2. There are 16 USRPs connected.

QFX5100-96S acts as the switch between the X310s on one hand and the computers on the other hand which are also connected to the switch's 10 Gigabit Ethernet ports. These devices provide us with the hardware setup necessary to run our software frameworks via virtual machines as explained below.

1.1 Drivers

USRP X310s come equipped with cross-platform driver support through the UHD (USRP Hardware Driver). The UHD architecture provides compatibility to GNURadio and CRTS(v.2).

UHD comes with its own benchmarking tool that let us benchmark transmission rate and reception rate. This can help us separate timing information

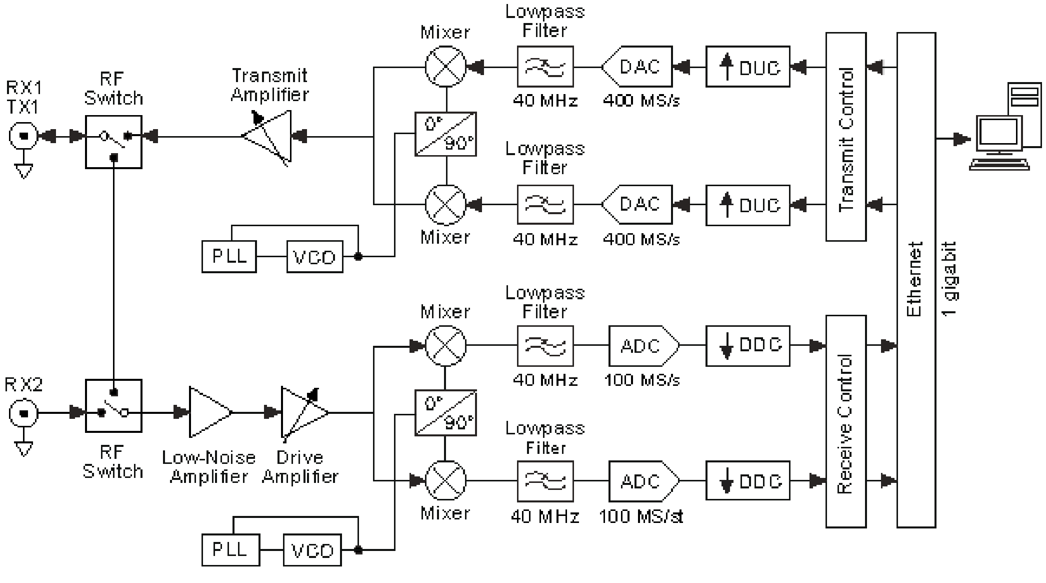


Figure 1: USRP Architecture

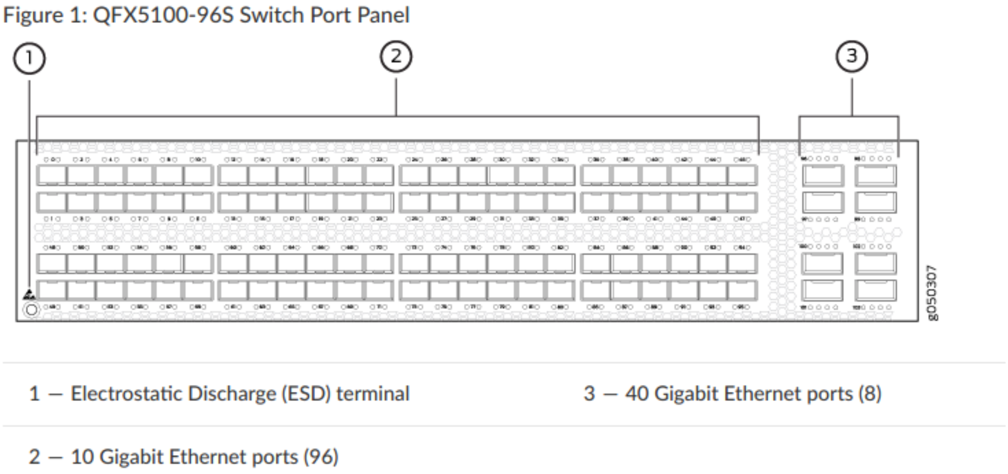


Figure 1: QFX5100-96S Switch Port Panel

1 — Electrostatic Discharge (ESD) terminal

3 — 40 Gigabit Ethernet ports (8)

2 — 10 Gigabit Ethernet ports (96)

Figure 2: Juniper's QFX5100-96S switch port panel [3]

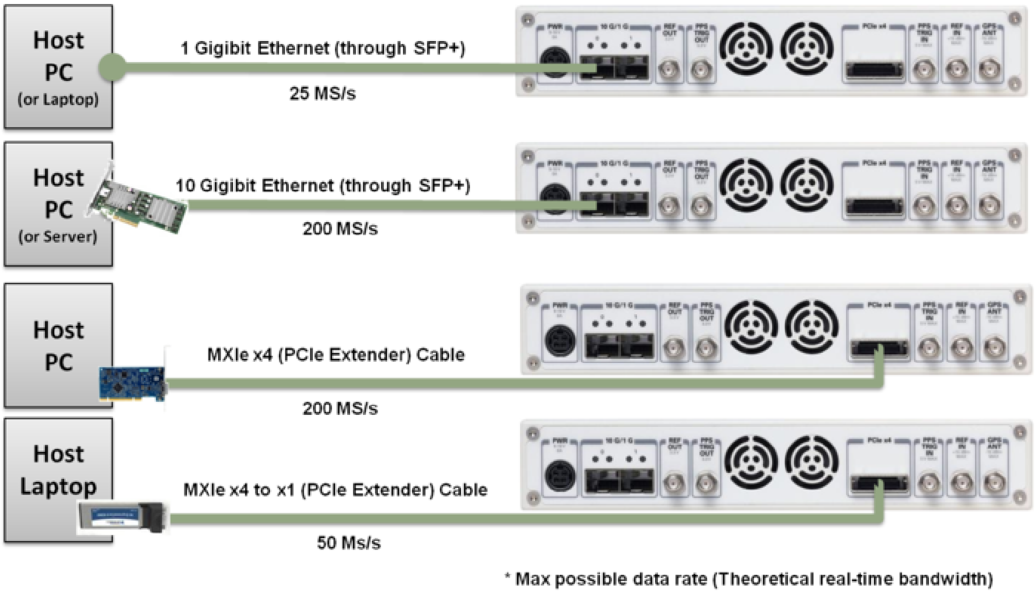

Host PC (or Laptop)
1 Gigibit Ethernet (through SFP+)
25 MS/s
Host PC (or Server)
10 Gigibit Ethernet (through SFP+)
200 MS/s
Host PC
MXIe x4 (PCIe Extender) Cable
200 MS/s
Host Laptop
MXIe x4 to x1 (PCIe Extender) Cable
50 Ms/s
* Max possible data rate (Theoretical real-time bandwidth)

Figure 3: USRP X310 Connectivity Options [2]

of software from hardware and drivers. It serves the purpose of better software performance measurement.

2 Software

The computers connected to the switch have access to the USRPs directly. Virtual machines can be hosted on these computers. The ESXi hypervisor is used to setup the virtual machines.

3 Performance Metrics

Cognitive Radios Test System (CRTS) has gone through various versions but a conclusive performance analysis hasn't been performed. To this end, I plan to perform various tests on the current CRTS(v.2) platform with the some performance metrics in mind. The following flowgraphs can be simulated and tested :

1. transmitter, receiver

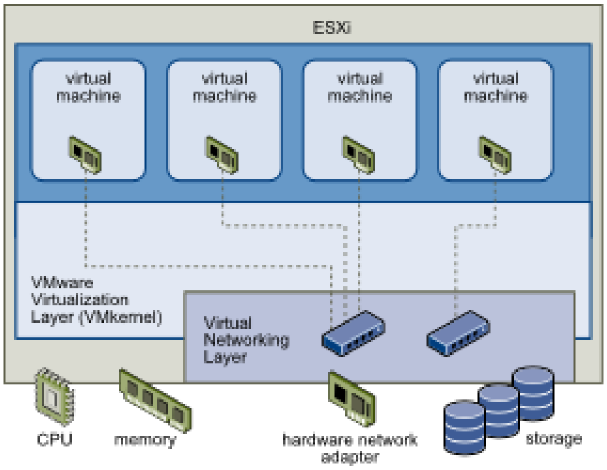


Figure 4: General VM architecture

2. transceiver
3. FM receiver
4. AM receiver

This analysis will lead to an understanding of the performance metrics listed below.

1. Latency. Universal Software Radio Peripherals (USRPs) are commonly used hardware platforms for software radios testbeds. USRPs, devised by Ettus Research, comes with USRP Hardware Driver libraries that act as an interface between the host computer and the hardware. The latency incurred while running any radio application can therefore, be broken down into three key areas of interest; software latency incurred due to complexity of signal processing tasks, bus latency while communicating between host computer and hardware platform and hardware latency due to buffering of samples in the USRP. [4–7]

2. Throughput comparison with GNURadio. The throughput measurements can be performed with number of samples received in a certain period of time for common flowgraphs in GNURadio and CRTS.

3. % CPU usage vs. workload

4. Memory usage: disk and RAM over a duration of time
5. Memory bandwidth
6. Response time and wait times across software and buses.

GNURadio flowgraphs are structured such that there is a buffer in between consecutive blocks. Data is copied from one block to the next whereas in CRTS, the data is simply passed through. The above tests can demonstrate what the benefits of the new approach are. Based on these metrics, we can evaluate CRTS to provide a comparative in-depth analysis of the the CRTS software. It can also inform design designs now and in the future.

Software performance varies with the processor architecture as well. It would benefit us to know how the software performs on x86, ARM in the general-purpose computing area and GPUs or DSPs.

Owing to the networked nature of the software, different radios are connected over a network to each other and the computers. This complexity in the architecture needs to be well-understood to inform further software and hardware development.

3.1 GNURadio Flowgraphs

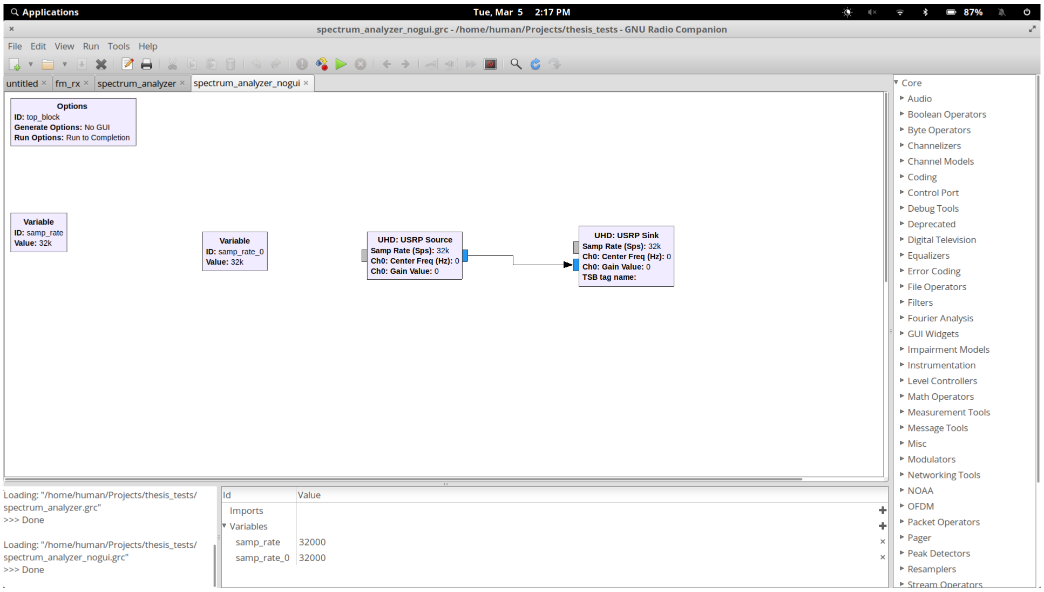
This section shows diagrams of GNURadio flowgraphs to be implemented for performance testing. Fig.5 shows a simple spectrum analyzer which can display the signals being transmitted over a fixed frequency. This can be adapted to search over a range of frequencies. Fig.6 shows the flowgraph for a wideband FM receiver. Fig.7 shows the flowgraph for a UHD transmitter transmitting random signals over a certain frequency.

References

[1] About usrp bandwidths and sampling rates. https://kb.ettus.com/About_USRP_Bandwidths_and_Sampling_Rates. Accessed: 2010-02-20.

[2] Ettus Research. *USRP X300 and X310 Product Overview*, 2 2014.

[3] Juniper Networks. *Port Panel of a QFX5100-96S Device*, 3 2018.



Options
ID: top_block
Generate Options: No GUI
Run Options: Run to Completion

Variable
ID: samp_rate
Value: 32k

Variable
ID: samp_rate_0
Value: 32k

UHD: USRP Source
Samp Rate (Sps): 32k
Ch0: Center Freq (Hz): 0
Ch0: Gain Value: 0

UHD: USRP Sink
Samp Rate (Sps): 32k
Ch0: Center Freq (Hz): 0
Ch0: Gain Value: 0
TSB tag name:

```
Loading: "/home/human/Projects/thesis_tests/
spectrum_analyzer.grc"
>>> Done

Loading: "/home/human/Projects/thesis_tests/
spectrum_analyzer_nogui.grc"
>>> Done
```

Id	Value
Imports	
Variables	
samp_rate	32000
samp_rate_0	32000

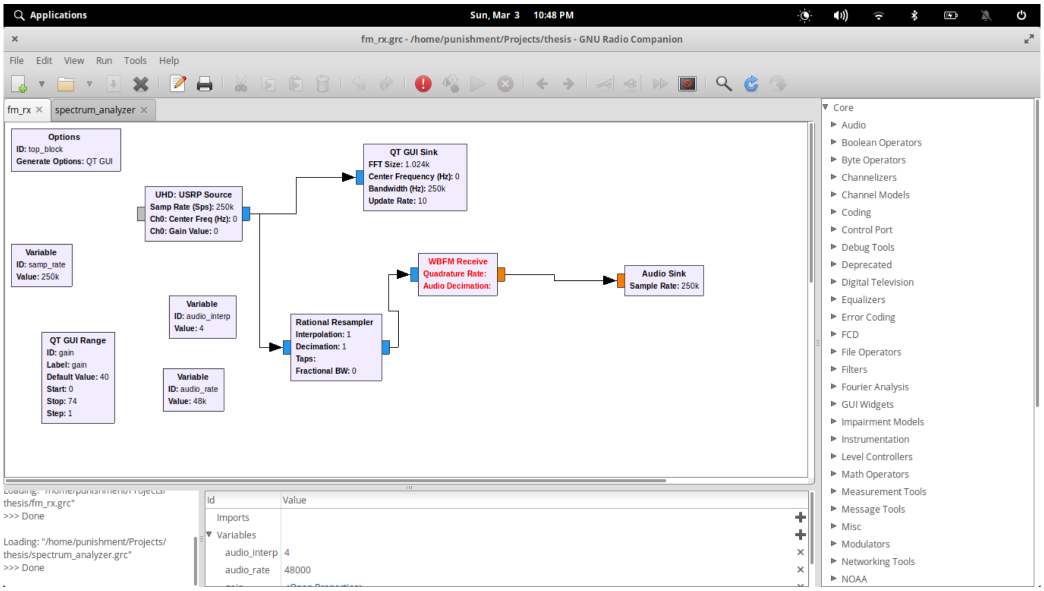
Figure 5: Spectrum Analyzer using UHD

[4] André Puschmann, Mohamed A Kalil, and Andreas Mitschele-Thiel. Implementation and evaluation of a practical sdr testbed. In *Proceedings of the 4th International Conference on Cognitive Radio and Advanced Spectrum Management*, page 15. ACM, 2011.

[5] Thomas Schmid, Oussama Sekkat, and Mani B Srivastava. An experimental study of network performance impact of increased latency in software defined radios. In *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pages 59–66. ACM, 2007.

[6] N. B. Truong and C. Yu. Investigating latency in gnu software radio with usrp embedded series sdr platform. In *2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications*, pages 9–14, Oct 2013.

[7] Stefan Valentin, Holger von Malm, and Holger Karl. Evaluating the gnu software radio platform for wireless testbeds. *University of Paderborn, Tech. Rep*, 2006.



fm_rx.grc - /home/punishment/Projects/thesis - GNU Radio Companion

Options
ID: top_block
Generate Options: QT GUI

UHD: USRP Source
Samp Rate (Sps): 250k
Ch0: Center Freq (Hz): 0
Ch0: Gain Value: 0

QT GUI Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 250k
Update Rate: 10

Variable
ID: samp_rate
Value: 250k

WBFM Receive
Quadrature Rate:
Audio Decimation:

Audio Sink
Sample Rate: 250k

Variable
ID: audio_interp
Value: 4

QT GUI Range
ID: gain
Label: gain
Default Value: 40
Start: 0
Stop: 74
Step: 1

Rational Resampler
Interpolation: 1
Decimation: 1
Taps:
Fractional BW: 0

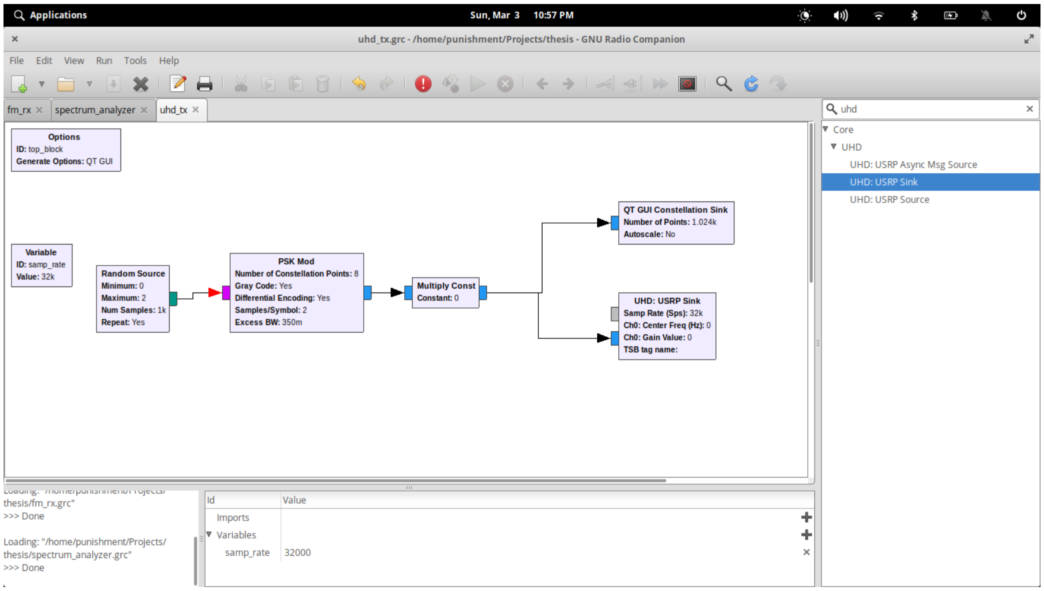
Variable
ID: audio_rate
Value: 48k

```
Loading: "/home/punishment/Projects/
thesis/fm_rx.grc"
>>> Done

Loading: "/home/punishment/Projects/
thesis/spectrum_analyzer.grc"
>>> Done
```

Id	Value
Imports	
Variables	
audio_interp	4
audio_rate	48000

Figure 6: FM Radio Receiver



uhd_tx.grc - /home/punishment/Projects/thesis - GNU Radio Companion

Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 32k

Random Source
Minimum: 0
Maximum: 2
Num Samples: 1k
Repeat: Yes

PSK Mod
Number of Constellation Points: 8
Gray Code: Yes
Differential Encoding: Yes
Samples/Symbol: 2
Excess BW: 350m

Multiply Const
Constant: 0

QT GUI Constellation Sink
Number of Points: 1.024k
Autoscale: No

UHD: USRP Sink
Samp Rate (Sps): 32k
Ch0: Center Freq (Hz): 0
Ch0: Gain Value: 0
TSB tag name:

```
Loading: "/home/punishment/Projects/
thesis/fm_rx.grc"
>>> Done

Loading: "/home/punishment/Projects/
thesis/spectrum_analyzer.grc"
>>> Done
```

Id	Value
Imports	
Variables	
samp_rate	32000

Figure 7: Transmitter