Latency Measurements for USRP B210 and X300 for GPP based SDR Applications

Sumit KUMAR, Florian KALTENBERGER

Eurecom, France

Abstract

This measurement is conducted in order to estimate the round trip and receive latency for USRP B210 and USRP X300. The measured latency is used to investigate GPP based bidirectional 802.11g SDR implementation feasibility.

1. Critical SIFS requirement for bidirectional 802.11g

- SIFS is the time from the reception of the last PPDU symbol to the transmission of the first symbol of the response PPDU.
- SIFS for 802.11a (5GHz band) is 16 microsecs and 802.11g (2.4 GHz) is 10 microsecs.
- Additionally in order to transmit any type of frame in the air, 802.11 follows CSMA/CA with DCF among the transmitting stations. Once the channel is sensed idle, the transmitter has to respond instantaneously. Delay in response may result in false channel information and hence collision of packets in air.

2. Test Methods

We conduct two different tests. One for measuring the round trip latency and another for receive latency for both the aforementioned USRPs. We use UHD (Universal Hardware Driver) API in order to stream samples from USRP to CPU and vice versa. Following hardware were used in the tests :

- LeCroy 6050 : 500 MHz Oscilloscope

- Rohde&Schwarz VSG SMB 100
- USRP B210
- USRP X300
- Dell Precision 5510 (For USRP B210)
- Intel(R) Xeon(R) CPU E5-2687W v3 @ 3.10GHz (For USRP X300)

Sampling rate of 20 MHz with a center frequency of 400 MHz was used.

3. Round Trip Latency Test method

For round trip latency test, we use the script latency_test.cpp which is provided by UHD. It can be found in /uhd/host/examples. This script receives a packet at time t and tries to send the same packet at time t + rtt, where rtt is the requested round trip time(round trip time sample time from device to host and back to the device). rtt can be inputted to the script as command line parameter. The script takes rtt as input and tries to perform the mentioned routine and outputs if there were underruns or overruns in form of ACK. For a success, the number of ACK received has to equal to number of runs. The test was conducted with 100 samples per run and 1000 runs in total.

We changed the default values of following parameters in order to minimize the latency as much as possible. We tuned the parameters till there were no overflows and underflows. Finally following values were used.

USRP B210	USRP X300
("sc16", "sc16")	("sc16", "sc16")
"spp" = "80"	"spp" = "80"
"num_recv_frames" = "256"	
"recv_frame_size" = "20"	
"num_send_frames" = "256"	
"send_frame_size" = "20"	

Table 1: UHD stream args used for latency_test.cpp

Params mentioned in bold in Table 1 are valid only for USB based transfer(USRP B210) and not for PCIe(USRP X300) based transfer between USRP and CPU.

The test setup is shown in Figure 1

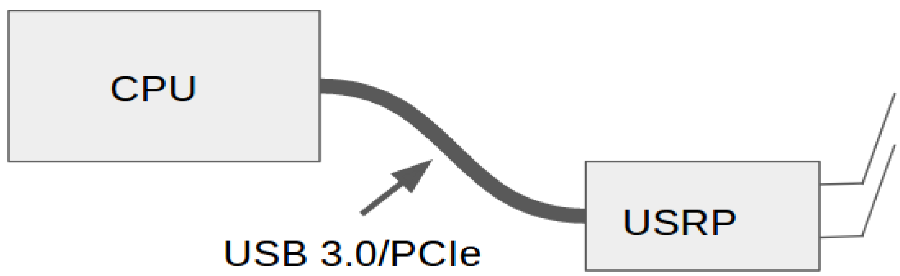

Figure 1: Round Trip Latency Test Setup

3.1. Observations

Minimum rtt supported by USRP B210 was 160 micro secs while 75 micro secs for USRP X300 for the configurations mentioned in Table 1.

4. Receive Latency Test Method

For receiver latency test, we made a custom script in C++ using UHD C++ API. A signal generator(VSG) was taken and configured to continuously transmit 802.11g frames at an inter-frame interval of 100ms. The 802.11g signal was splitted and fed to oscilloscope and USRP simultaneously. USRP was set into receiver mode at a sampling rate of 20 MHz. The sampled data from USRP was fed to CPU. The script continuously compares the average energy over a window of samples and compares it against a threshold. Size of window was equal to size of the buffer(mentioned below as spp). If window energy exceeds the threshold, the script triggers all the GPIO pins on the USRP to go HIGH, which were otherwise LOW. The GPIO pins of the USRP were connected to the other channel of oscilloscope and the delay between the transmit signal from signal generator was being continuously compared.

We used following parameters to configure the USRPs thru UHD. Measured time highly depends on transport parameters (for USRP B210) and samples per packet parameter (spp for both USRPs).

Similar to round trip latency test, we tried to put spp parameter as less as possible till there were no overruns. However receive latency tests (for both USRP B210 and X300), did not comply with spp value of 80 which was used in round trip latency tests. One of the possible reasons could be continuous streaming of samples in receiver latency tests vs non-continuous streaming of samples in round trip latency test.

A large size of spp will leads to higher latency as the USRP buffer wait for the buffer to be filled up before flushing it to the CPU.

USRP B210	**USRP X300**
stream_args("sc16", "sc16")	stream_args("sc16", "sc16")
stream_args.args["spp"] = "500"	stream_args.args["spp"] = "200"
stream_args.args["num_recv_frames"] = "256"	
stream_args.args["recv_frame_size"] = "20"	
stream_args.args["num_send_frames"] = "256"	
stream_args.args["send_frame_size"] = "20"	

Table 2: UHD Params used for Receive Latency Test

Params mentioned in bold in Table 2 are valid only for USB based transfer(USRP B210) and not for PCIe(USRP X300) based transfer between USRP and CPU.

The test setup and hardware are shown in Figure 2 and 3 respectively

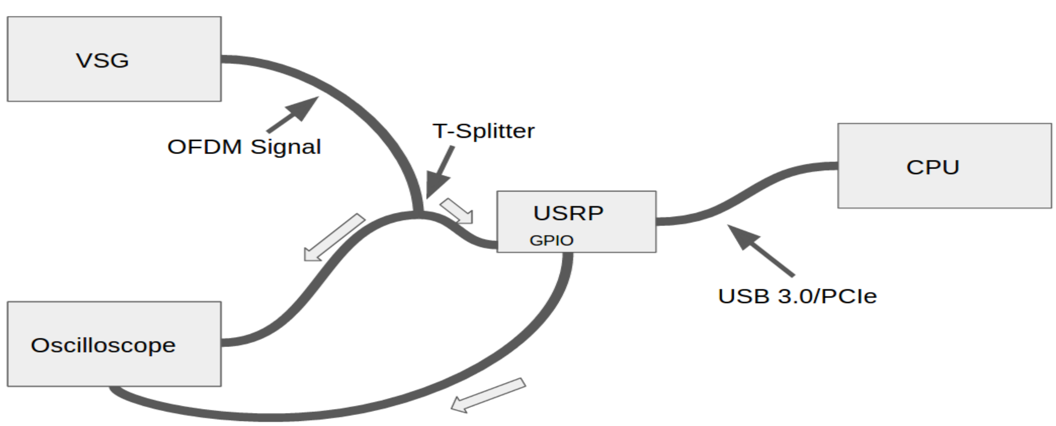

Figure 2: Receive Latency Test Setup

Figure 3: Hardware Setup

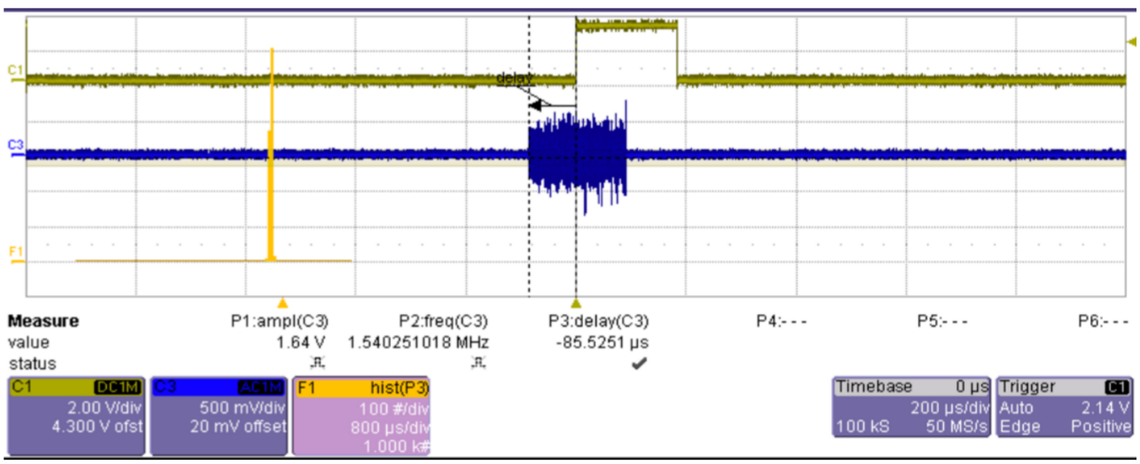

Figure 4: Oscilloscope View

Figure 4 shows the latency measurement from LeCroy Oscilloscope

In Figure 4 the blue line is from signal generator while the yellow line is from GPIO. Delay is continuously been tracked (arrow mark).

Figure 5 shows the calculation of receive latency as "Measured Time"

4.1. Observations

The measured time was highly fluctuating, hence we took statistical measurements by obtaining the histogram of the delay. Results from the histogram are shown in Table 3

Histogram of the receive latency for B210 and X300 are shown in Figure 6 and 7 respectively.

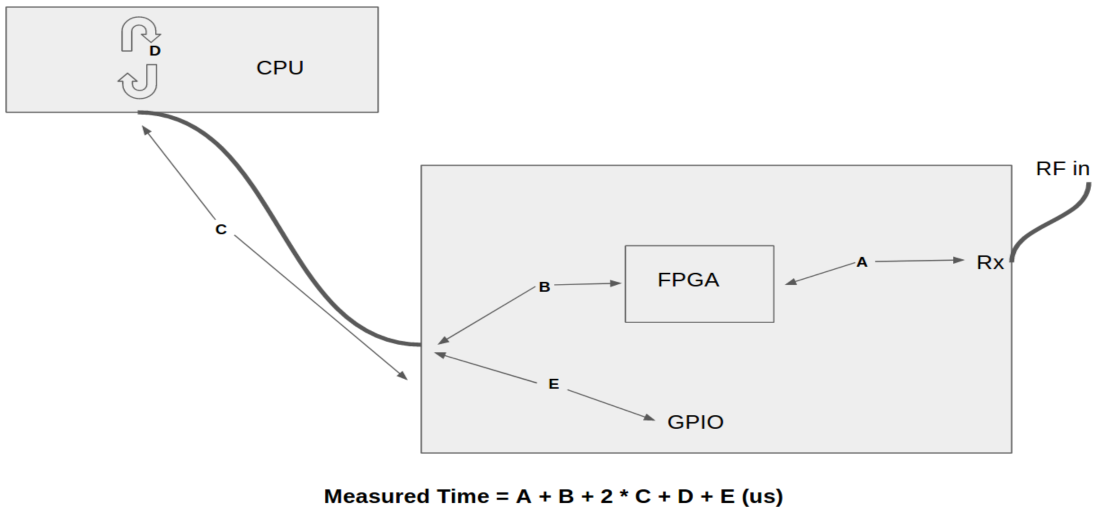

Figure 5: Measured Time

	USRP B210	USRP X300
Maxm Latency (μs)	126	105
Minm Latency (μs)	68.8	57.6
Mode (μs)	88.8	83.6

Table 3: Receive latency test results

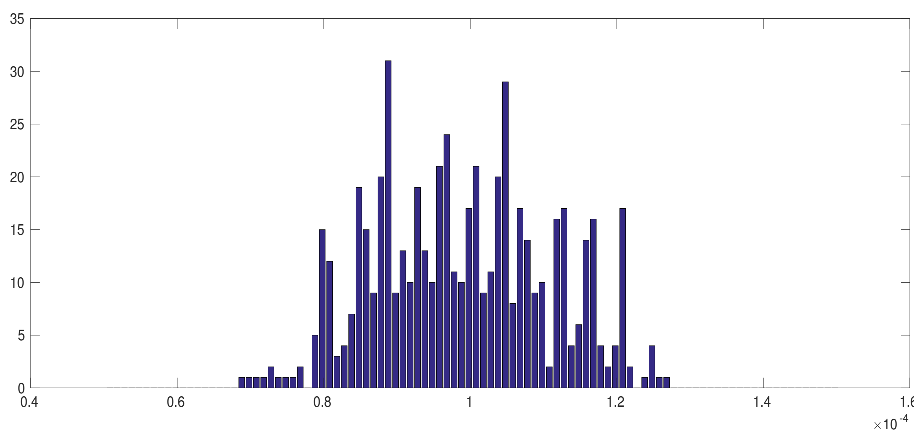

Figure 6: Receive Latency for B210

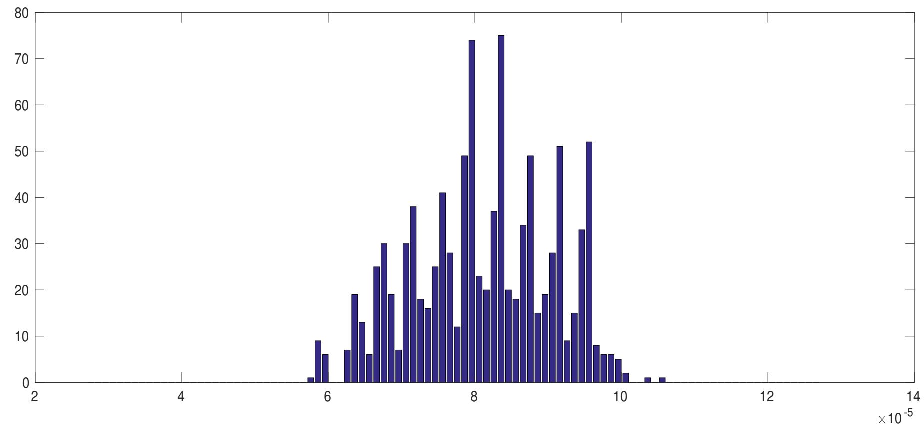

Figure 7: Receive Latency for X300

5. Conclusion

Half of the measured time gives a rough indication of latency. We observe that with the current GPU based architecture to implement bidirectional WiFi (802.11g – SIFS 10 microsecs) is not feasible with such delays using USRP X300 (PCIe) and USRP B210 (USB 3.0).