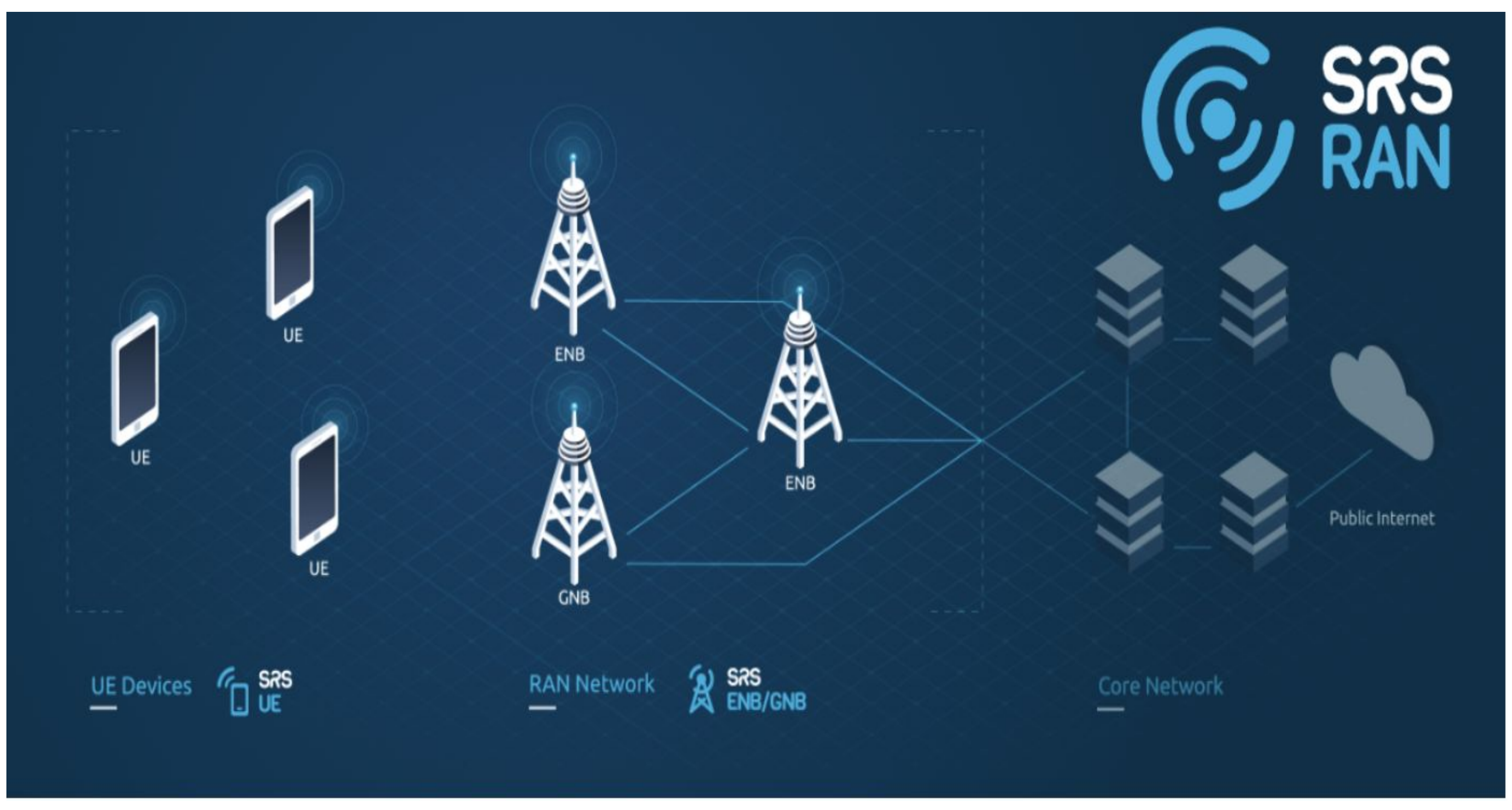
Research and Testing of 5G wireless Scheduler Algorithms

Team: sddec21-15
Members: Anh To, Bradley Norman, Haan Zilmer, Elias Zougmore
Faculty Advisor: Hongwei Zhang
Client: Hongwei Zhang



Intro/Motivation

Advancements in 5G technology have led to an increase in demand for qualified engineers with the ability to develop and prototype advanced wireless solutions. 5G wireless networks are expected to enable not only Gbps mobile connectivity but also machine-type communications for smart agriculture, connected and automated vehicles, smart grid, Industry 4.0, and AR/VR. Our project is in a research capacity, so while we will not be solving any specific problem, we will be looking into ways to improve the scheduling algorithm for 5G Systems.

Design Requirements

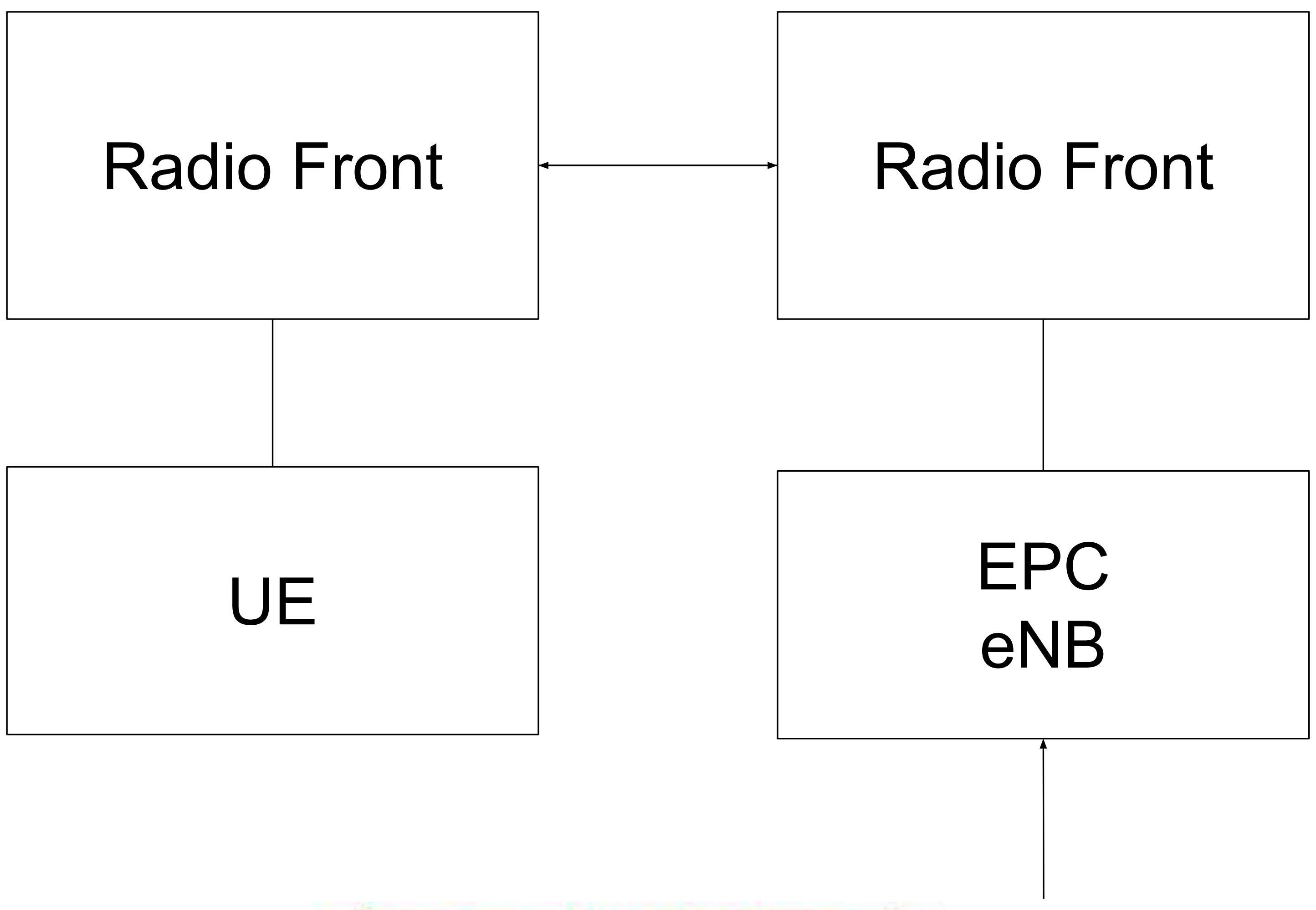
Functional requirements are to ensure schedule and time allocation efficiency, ensure communication between base stations and the mobile core. The non-functional requirements are the research on 5G wireless Systems, srsRAN base code and the the documentation on srsRAN code.
We have an open-source software platform (srsRAN) and test are run on testbed.

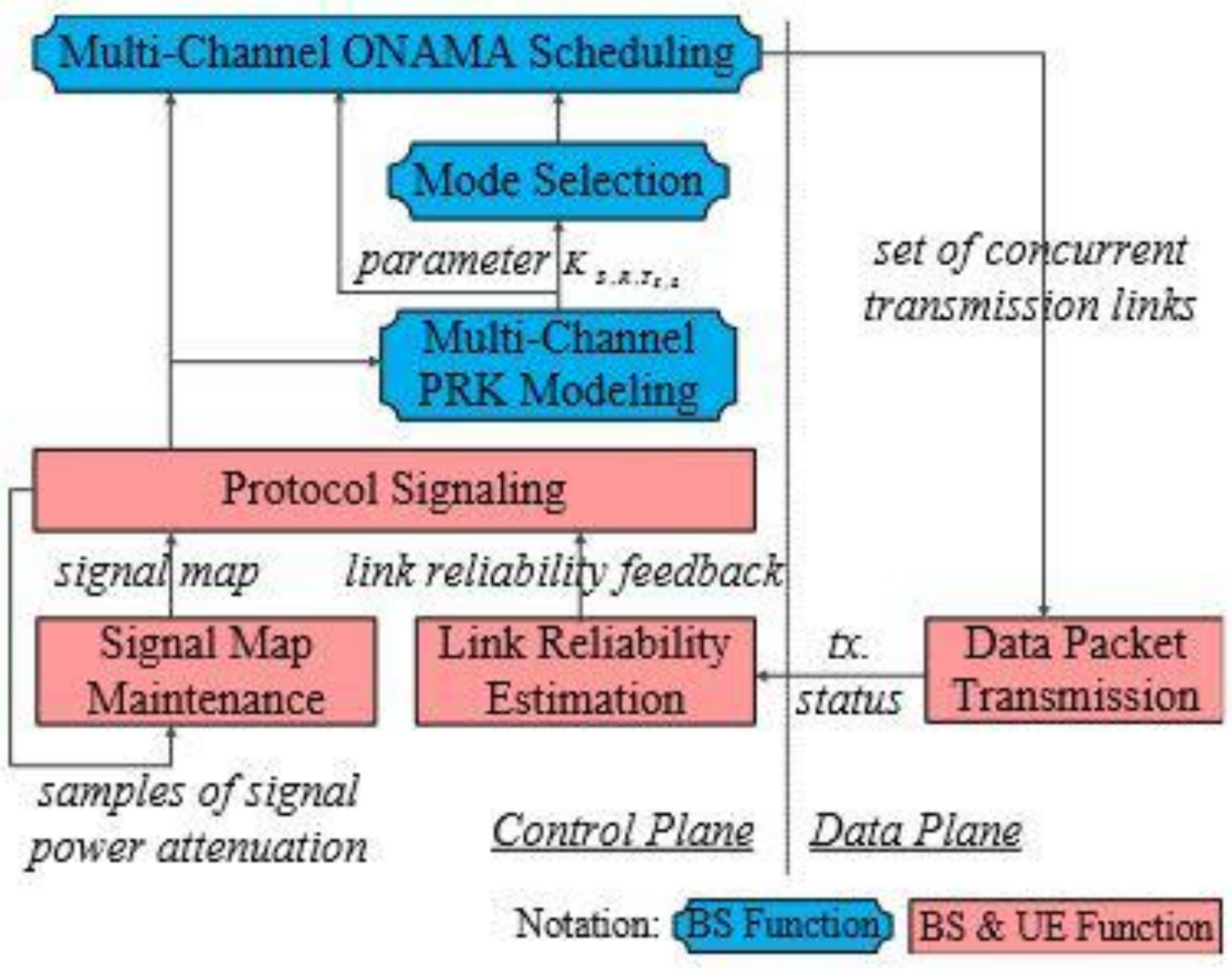
Intended users: Undergrad Students, Graduate Students, and Professors
Intended uses: The goal was to obtain research regarding the UCS algorithm's relative function to base scheduler functions for other researchers to use in more commercial research, however, upon not being able to reach that goal we shifted to creating documentation for future senior design groups to help them avoid the massive time sink we had go through.

Technical Details

Our project was primarily software with some hardware support towards the end. For our software resources we were using srsRAN's code base downloaded from github in the language C++. For hardware we were given 2 SDRs and 1 monitor by our faculty advisor in order to try to test our algorithm in a physical environment.

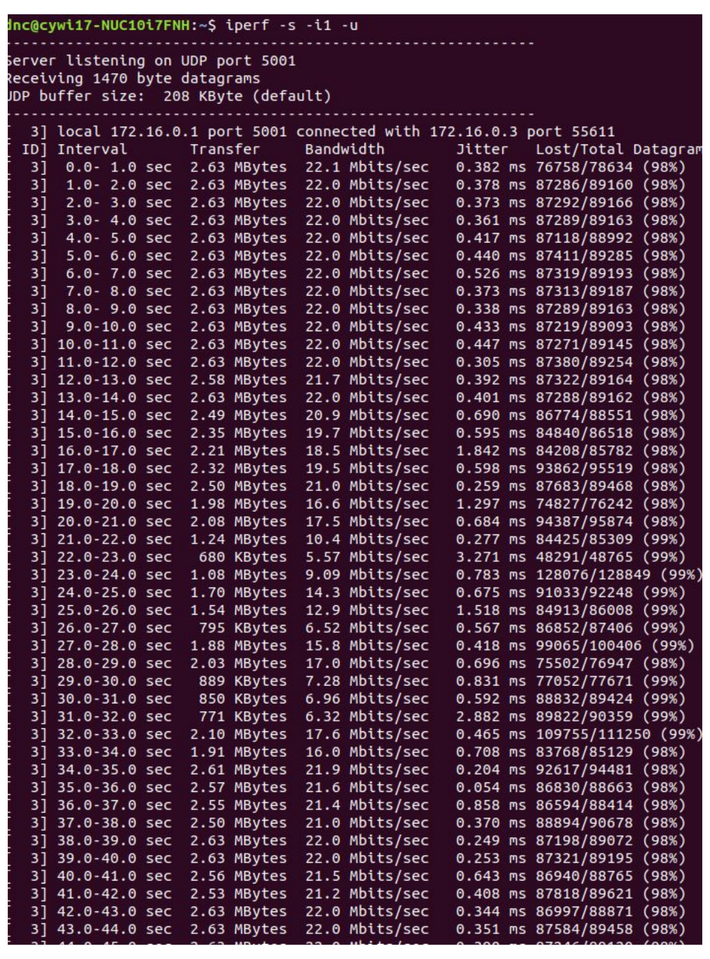
Design Approach




Multi-Channel ONAMA Scheduling
Mode Selection
parameter K
Multi-Channel PRK Modeling
set of concurrent transmission links
Protocol Signaling
signal map
link reliability feedback
Signal Map Maintenance
Link Reliability Estimation
tx. status
Data Packet Transmission
samples of signal power attenuation
Control Plane
Data Plane
Notation:
BS Function
BS & UE Function

This diagram shows the architecture of the Unified Cellular Scheduling framework based on the status of uplink transmission as well as downlink and D2D link transmissions, the Bses and UEs estimate the communication reliability respectively

Testing

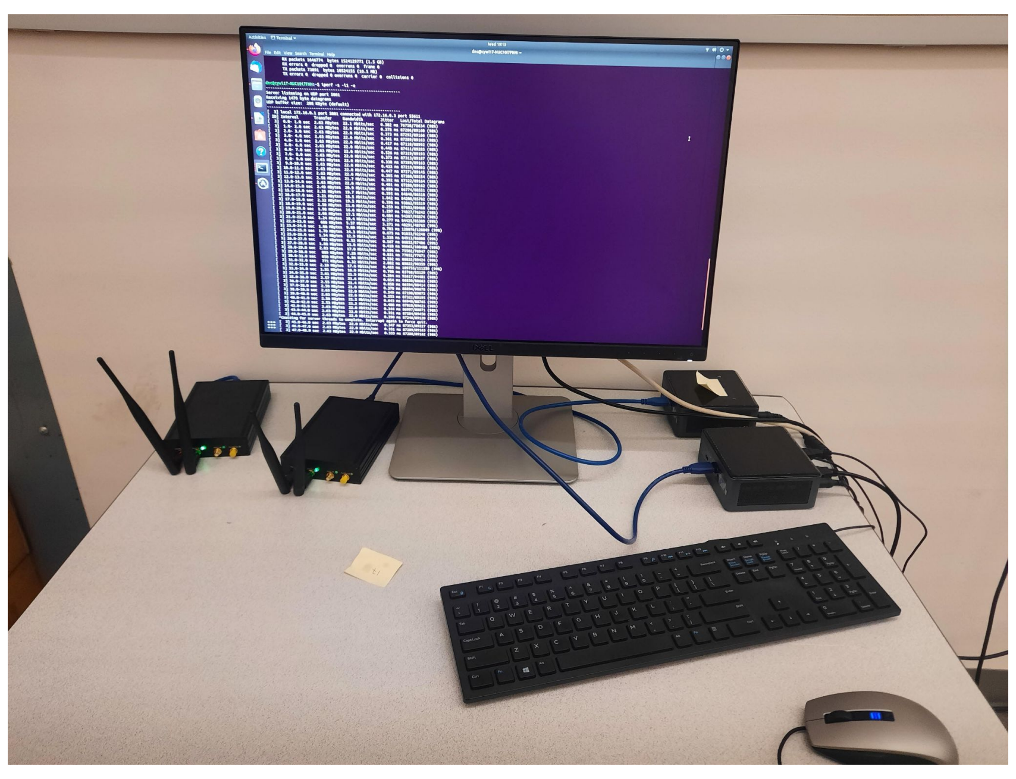


```
dnc@cywi17-NUC10i7FNH:~$ iperf -s -i1 -u
------------------------------------------------------------
erver listening on UDP port 5001
eceiving 1470 byte datagrams
DP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 172.16.0.1 port 5001 connected with 172.16.0.3 port 55611
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagram
[  3]  0.0- 1.0 sec  2.63 MBytes  22.1 Mbits/sec   0.382 ms 76758/78634 (98%)
[  3]  1.0- 2.0 sec  2.63 MBytes  22.0 Mbits/sec   0.378 ms 87286/89160 (98%)
[  3]  2.0- 3.0 sec  2.63 MBytes  22.0 Mbits/sec   0.373 ms 87292/89166 (98%)
[  3]  3.0- 4.0 sec  2.63 MBytes  22.0 Mbits/sec   0.361 ms 87289/89163 (98%)
[  3]  4.0- 5.0 sec  2.63 MBytes  22.0 Mbits/sec   0.417 ms 87118/88992 (98%)
[  3]  5.0- 6.0 sec  2.63 MBytes  22.0 Mbits/sec   0.440 ms 87411/89285 (98%)
[  3]  6.0- 7.0 sec  2.63 MBytes  22.0 Mbits/sec   0.526 ms 87319/89193 (98%)
[  3]  7.0- 8.0 sec  2.63 MBytes  22.0 Mbits/sec   0.373 ms 87313/89187 (98%)
[  3]  8.0- 9.0 sec  2.63 MBytes  22.0 Mbits/sec   0.338 ms 87289/89163 (98%)
[  3]  9.0-10.0 sec  2.63 MBytes  22.0 Mbits/sec   0.433 ms 87219/89093 (98%)
[  3] 10.0-11.0 sec  2.63 MBytes  22.0 Mbits/sec   0.447 ms 87271/89145 (98%)
[  3] 11.0-12.0 sec  2.63 MBytes  22.0 Mbits/sec   0.305 ms 87380/89254 (98%)
[  3] 12.0-13.0 sec  2.58 MBytes  21.7 Mbits/sec   0.392 ms 87322/89164 (98%)
[  3] 13.0-14.0 sec  2.63 MBytes  22.0 Mbits/sec   0.401 ms 87288/89162 (98%)
[  3] 14.0-15.0 sec  2.49 MBytes  20.9 Mbits/sec   0.690 ms 86774/88551 (98%)
[  3] 15.0-16.0 sec  2.35 MBytes  19.7 Mbits/sec   0.595 ms 84840/86518 (98%)
[  3] 16.0-17.0 sec  2.21 MBytes  18.5 Mbits/sec   1.842 ms 84208/85782 (98%)
[  3] 17.0-18.0 sec  2.32 MBytes  19.5 Mbits/sec   0.598 ms 93862/95519 (98%)
[  3] 18.0-19.0 sec  2.50 MBytes  21.0 Mbits/sec   0.259 ms 87683/89468 (98%)
[  3] 19.0-20.0 sec  1.98 MBytes  16.6 Mbits/sec   1.297 ms 74827/76242 (98%)
[  3] 20.0-21.0 sec  2.08 MBytes  17.5 Mbits/sec   0.684 ms 94387/95874 (98%)
[  3] 21.0-22.0 sec  1.24 MBytes  10.4 Mbits/sec   0.277 ms 84425/85309 (99%)
[  3] 22.0-23.0 sec   680 KBytes  5.57 Mbits/sec   3.271 ms 48291/48765 (99%)
[  3] 23.0-24.0 sec  1.08 MBytes  9.09 Mbits/sec   0.783 ms 128076/128849 (99%)
[  3] 24.0-25.0 sec  1.70 MBytes  14.3 Mbits/sec   0.675 ms 91033/92248 (99%)
[  3] 25.0-26.0 sec  1.54 MBytes  12.9 Mbits/sec   1.518 ms 84913/86008 (99%)
[  3] 26.0-27.0 sec   795 KBytes  6.52 Mbits/sec   0.567 ms 86852/87406 (99%)
[  3] 27.0-28.0 sec  1.88 MBytes  15.8 Mbits/sec   0.418 ms 99065/100406 (99%)
[  3] 28.0-29.0 sec  2.03 MBytes  17.0 Mbits/sec   0.696 ms 75502/76947 (98%)
[  3] 29.0-30.0 sec   889 KBytes  7.28 Mbits/sec   0.831 ms 77052/77671 (99%)
[  3] 30.0-31.0 sec   850 KBytes  6.96 Mbits/sec   0.592 ms 88832/89424 (99%)
[  3] 31.0-32.0 sec   771 KBytes  6.32 Mbits/sec   2.882 ms 89822/90359 (99%)
[  3] 32.0-33.0 sec  2.10 MBytes  17.6 Mbits/sec   0.465 ms 109755/111250 (99%)
[  3] 33.0-34.0 sec  1.91 MBytes  16.0 Mbits/sec   0.708 ms 83768/85129 (98%)
[  3] 34.0-35.0 sec  2.61 MBytes  21.9 Mbits/sec   0.204 ms 92617/94481 (98%)
[  3] 35.0-36.0 sec  2.57 MBytes  21.6 Mbits/sec   0.054 ms 86830/88663 (98%)
[  3] 36.0-37.0 sec  2.55 MBytes  21.4 Mbits/sec   0.858 ms 86594/88414 (98%)
[  3] 37.0-38.0 sec  2.50 MBytes  21.0 Mbits/sec   0.370 ms 88894/90678 (98%)
[  3] 38.0-39.0 sec  2.63 MBytes  22.0 Mbits/sec   0.249 ms 87198/89072 (98%)
[  3] 39.0-40.0 sec  2.63 MBytes  22.0 Mbits/sec   0.253 ms 87321/89195 (98%)
[  3] 40.0-41.0 sec  2.56 MBytes  21.5 Mbits/sec   0.643 ms 86940/88765 (98%)
[  3] 41.0-42.0 sec  2.53 MBytes  21.2 Mbits/sec   0.408 ms 87818/89621 (98%)
[  3] 42.0-43.0 sec  2.63 MBytes  22.0 Mbits/sec   0.344 ms 86997/88871 (98%)
[  3] 43.0-44.0 sec  2.63 MBytes  22.0 Mbits/sec   0.351 ms 87584/89458 (98%)
```

For the testing we needed to test for packet loss and bandwidth of both the default srsRAN with RR and PF scheduling as well as the newly implemented scheduling algorithm. To do this we installed srsRAN onto two compute nodes with two software defined radios, we then had to determine the transport layer protocol utilized by srsRAN and run an iperf command which is depicted above.

Resources

- 2 Radio fronts
- 2 PCs
- srsRAN code base



Standards

- IEEE and SESC software development standard
- Transparent on using open-source resources