**CHANGES TO THE MAC DIRECTORY:**

* **Added: network\_components/client.cc**
* **Modified: mac.cc => new include header line and added lines 909-912**
* **Modified: CMakeLists.txt => added set(…, network\_components/client.cc)**

1. **UHD install**

Dependencies

sudo apt-get install autoconf automake build-essential ccache cmake cpufrequtils doxygen ethtool g++ git inetutils-tools libboost-all-dev libncurses5 libncurses5-dev libusb-1.0-0 libusb-1.0-0-dev libusb-dev python3-dev python3-mako python3-numpy python3-requests python3-scipy python3-setuptools python3-ruamel.yaml

Installation

git clone [https://github.com/EttusResearch/uhd](about:blank) && cd uhd/host &&

git checkout [UHD-3.15.LTS](about:blank) && mkdir build && cd build && cmake .. &&

make -j8 && sudo make install && sudo ldconfig

1. **Install srsRAN: (with our Solution)**

Before installation: Check your you UHD driver finds the UHD devices using the following commands

uhd\_find\_devices

uhd\_usrp\_probe

Install pre-requsite libraries

sudo apt-get install build-essential cmake libfftw3-dev libmbedtls-dev libboost-program-options-dev libconfig++-dev libsctp-dev

Then install our adaptive resource allocation solution using the following commands:

git clone [https://gitlab.ilabt.imec.be/mgirmay/adaptive-mbsfn.git](about:blank) && cd adaptive-mbsfn && mkdir build && cd build && cmake ../ && make -j8 && sudo make install && sudo ldconfig

Install in atleast two host PCs .

1. **Running the test: eNB PC**

Pre-requisites to run the test: set your machine in performance mode

echo "performance" | sudo tee /sys/devices/system/cpu/cpu\*/cpufreq/scaling\_governor

**On eNB PC**

Run EPC

cd adaptive-mbsfn/srsepc

sudo srsepc epc.conf.example

MAKE SURE THAT db\_file is set to “user\_db.csv.example” (line 43 in conf file)

Run MBMS gateway

cd adaptive-mbsfn/srsepc

sudo srsmbms mbms.conf.example

Run eNodeB (eNB)

cd adaptive-mbsfn/srsenb

sudo srsenb enb.conf

1. **Running the test: eNB PC**

**On UE PC**

Run UE

cd adaptive-mbsfn/srsue

sudo srsue ue.conf

1. **Running iperf test**

**Set unicast Iperf**

On UE PC – new terminal

iperf -s -u

On eNB PC – new terminal

iperf -u -c 172.16.0.2 -b 30M -i 1 -t 6000

**Set multicast Iperf**

On UE PC – new terminal

sudo route add -net 239.255.1.0 netmask 255.255.255.0 dev tun\_srsue

iperf -s -u -B 239.255.1.1 -i 1

On eNB PC – new terminal

sudo route add -net 239.255.1.0 netmask 255.255.255.0 dev sgi\_mb

iperf -u -c 239.255.1.1 -b 20M -T 64 -t 60

1. **Running video streaming test**

**Set unicast video**

On eNB PC – new terminal

ffmpeg -re -i unicast1 -v 0 -f mpegts udp://172.16.0.2:1234 | (sleep 5 && ffplay unicast1)

On UE PC – new terminal

ffplay udp://172.16.0.1:1234

**Set multicast video**

On eNB PC – new terminal

sudo route add -net 239.255.1.0 netmask 255.255.255.0 dev sgi\_mb

ffmpeg -re -i multicast -v 0 -vcodec h264 -f mpegts udp://sgi\_mb@239.255.1.1:23000 | (sleep 5 && ffplay multicast)

On UE PC – new terminal

sudo route add -net 239.255.1.0 netmask 255.255.255.0 dev tun\_srsue

ffplay udp://tun\_srsue@239.255.1.1:23000

**MULTICAST VIDEO WITH VLC**

* **https://docs.videolan.me/vlc-user/3.0/en/advanced/streaming/stream\_over\_udp.html**

1. **Add routes just as before**
2. **On eNode side: (in directory videos)**

vlc sample.mp4 --sout="#std{access=udp, mux=ts, dst=239.255.1.0:23000}"

1. **On UE side:**

vlc udp://@ 239.255.1.0:23000