

KPM Semester project - 10

Tomáš Hadwiger, Tomáš Calábek

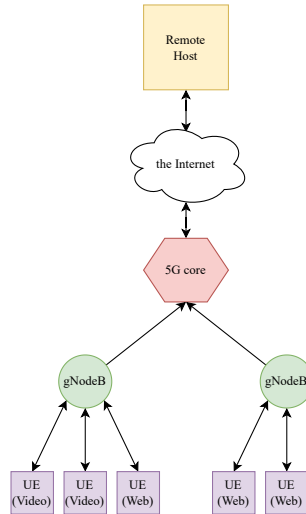
1 Our task & objectives

The objective of this project was to create a comprehensive network scenario using the 5G-LENA module of the NS3 simulator, configure the parameters of the network, and analyze the network performance.

2 Network Scenario

Our network scenario (see Figure 1) was made of 5 UEs and 2 gNodeBs. Two of the UEs were streaming video from a remote host over the Internet, while the other three UEs were browsing the web. Our network was further comprised of 2 gNodeBs, with maximum of three UE connections per single gNodeB.

Figure 1: A diagram of the network scenario.



3 Configuration

Parameters

A short list of the most relevant parameters for our simulation, as well as their default values, can be seen in Listing 1. These values can be modified with command line arguments when running the script using NS3 syntax.

Listing 1: Relevant default parameters.

```
uint32_t udpPacketSizeBrowsing = 25; // bytes
uint32_t udpPacketSizeVideo = 50;    // bytes
uint32_t lambdaBrowsing = 10000;     // packets per sec
uint32_t lambdaVideo = 10000;        // packets per sec
double totalTxPower = 35.0;          // dBm
```

Our script utilizes a sub-6GHz frequency band, as per the assignment. The sub-6GHz band is said to typically ranges from 410 MHz to 7125 MHz. This frequency band offers wide coverage and better penetration through obstacles when compared to the mmWave band and, while it delivers slower transfer speeds compared to the mmWave, its speed is still a significant improvement over 4G. We use central frequency of 2.8 GHz for video streaming and 2.82 GHz for web browsing.

Further, we also take advantage of two numerologies, one for each type of communication (e.g. video streaming and web browsing). For video streaming we utilize numerology 4, while for web browsing the numerology 2 is used. This corresponds to subcarrier spacing of 240 kHz and 60 kHz respectively.

Lastly, we set up appropriate IPv4 addresses and ranges for all the devices in the network. Our PGW is connected to the remote host using a Point-to-Point link and is assigned the address 8.0.0.0 with the subnet mask of 255.0.0.0; the UEs are assigned address from a 7.0.0.0 IP range.

Traffic generation

According to our assignment, the traffic on the network should be comprised of web browsing and video streaming. To this end, we create two dedicated bearers with the appropriate traffic flow templates (e.g. GBR.CONV_VIDEO for video streaming and NGBR.LOW_LAT_EMBB for web browsing) assigned to each. The traffic flow templates ensure appropriate traffic flow and QoS (Quality of Service) are used to each type of communication.

The communication itself is facilitated by an UDP server installed onto the remote host and a UDP client installed onto the UEs.

Listing 2: Traffic bearer definition.

```
NrEpsBearer bearerBrowsing(NrEpsBearer::NGBR_LOW_LAT_EMBB);
NrEpsBearer bearerVideo(NrEpsBearer::GBR_CONV_VIDEO);
```

4 Flow analysis results

To quantify our results, we use the FlowMonitor tool, which is an integral part of the NS3 simulator. We capture key metrics, like data speed, packet loss and throughput, into logs which we then use to analyze the impact of various parameters onto the network performance.

The FlowMonitor tool collects vast amounts of data for each simulation run, so for clarity we only include some of the most interesting results of our many simulation runs. In addition, we collected the mean data from only one of the flows shown by the FlowMonitor and investigated how it behaved when we changed the parameters. One example of an output from FlowMonitor is on Listing 3.

Listing 3: Example output from NS3 FlowMonitor

```
Flow 1 (8.0.0.2:49153 -> 7.0.0.2:1234) proto UDP
Tx Packets: 9900
Tx Bytes: 524700
TxOffered: 4.240000 Mbps
Rx Bytes: 521096
Lost Packets: 68
Packet loss: 0.686869%
Throughput: 4.210877 Mbps
Mean delay: 0.460727 ms
Mean jitter: 0.107409 ms
Rx Packets: 9832
```

In our analysis we focused on throughput, packet loss, delay and jitter of our network connection. From Table 1, we can see that increasing the total transmitting power beyond 35 dBm doesn't increase the chances of a succesful transmission, as the packet loss remained the same for 50 dBm.

We have also tested the bandwidth optimization on the system and found that bandwidth of $50 \cdot 10^6$ Hz is ideal. We saw some improvement in packet loss with $100 \cdot 10^6$ Hz, but the gains were minimal.

Beyond these two parameters, we have also confirmed that increasing the size of the UDP packets that we send over the network also increases the mean delay and packet loss.

Table 1: Comparisons of network parameters when changing total transmitting power.

T_x [dBm]	Throughput [Mbps]	Packet Loss [%]	Mean Delay [ms]	Mean Jitter [ms]
20	4.208	0.747	0.442	0.108
35	4.210	0.707	0.271	0.030
50	4.210	0.707	0.271	0.030

5 REM analysis

The REM (Radio Environment Map) is a tool in NS3 which generates maps of the radio environment, that help with visualizing signal propagation, interference patterns and coverage with the network.

While many different modes, and thus different data points, of the REM are possible, we only focus on two:

- **Beam Shape:** this mode allows us to visualize the shape of the beam in terms of SNR/SINR map;
- **Coverage Area:** this mode produces the best-case SNR (visualizes best possible beams) and worst-case SINR scenario (assuming all devices are pointing towards the REM point) for each REM point;
- **UE Coverage:** this mode works in the uplink direction (e.g. UE is the transmitter) and assumes the beams of the other gNodeBs are pointed towards the receiving gNodeB.

The results of our simulation are shown on Figure 2 and Figure 3.

Unfortunately, when creating the REM map for UL UE Coverage scenario, the simulation persistently crashed with error show in Listing 4. We tried to run the simulation on multiple devices and it seemed presistent, so we assume we have either made an error in our configuration of the REM module, or we have found a bug in the NS3 module itself.

Listing 4: Error produced by NS3 when generating UL UE Coverage REM

```
NS_ASSERT failed , cond="m_spectrumModel == x.m_spectrumModel" ,
+0.100000000s -1
file=/Vut/kpm/ns-3-dev/src/spectrum/model/spectrum-value.cc ,
line=107
NS_FATAL, terminating
```

Figure 2: REM map results in downlink direction.

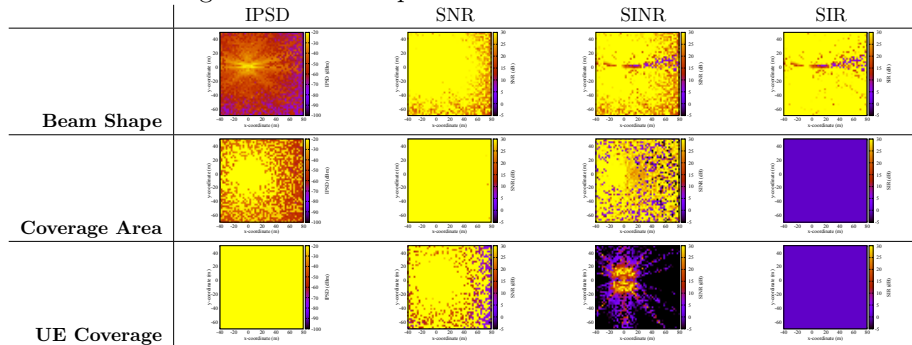
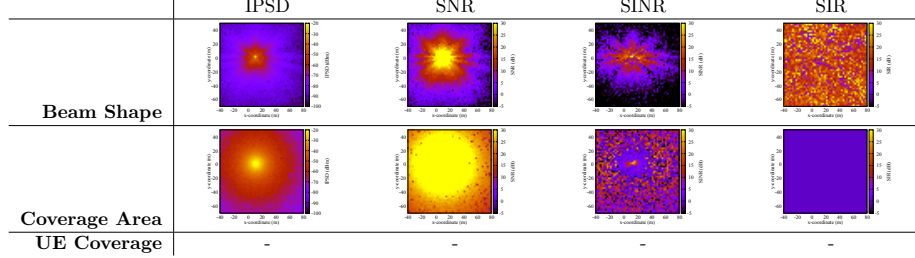


Figure 3: REM map results in uplinkdirection.



6 Challenges in working with NS3

Overall, working with the NS3 simulator and the 5G-LENA plugin wasn't extremely difficult, but it did provide a challenge, especially due to some lacking documentation on the parts of the 5G-LENA plugin.

We were also unable to complete some parts of the REM maps, as we have either encountered a bug in a NS3 module, or created our script in such a way that it worked fine for all the other configurations of the REM module, except for UL UE Coverage.

7 Final insights

While our setup in NS3 does much to simulate the real world, it still cannot account for all the complex real variations of wireless transmission, especially when it comes to the complexity of the network (our scenario only includes 5 UEs and two gNodeBs, while in reality, you could have hundreds of UE devices connected to a single gNodeB). Further, our scenario assumes the UEs are in a fixed position over the duration of the communication which isn't always true in real world scenarios. Our script also uses idealized models for propagation and interference, which do not fully model real factors like weather patterns, obstacles and device variability in terms of antennas.

Both of us were somewhat familiar with the C++ language and NS3-specific syntax thanks to the laboratory exercises and this project only helped to further our understanding of both.