KPM Project - 11

Matěj Baranyk
Jakub Lepík
Slávek Rylich
Martin Dolák

5G NR Simulation in NS-3 – Assignment 11

This project is for Assignment 11 in the MPA-KPM course at the FEEC BUT University. It focuses on simulating a 5G NR (New Radio) network using the 5G-LENA module in NS-3. The goal is to explore the behavior and performance of a 5G network.

Network Scenario

The simulated network structure is shown in Figure 1; it includes 5 UEs, 2 of which are engaged in a phone call with each other, each attached to a different gNodeB, while the remaining 3 are browsing the web and connected to the remote host server over the internet. Two gNodeBs are each connected to 2 UEs, while the third gNodeB is connected to only 1 UE. This setup is created using the GridScenarioHelper, a requirement for the assignment, and defines the stationary mobility of all UEs.

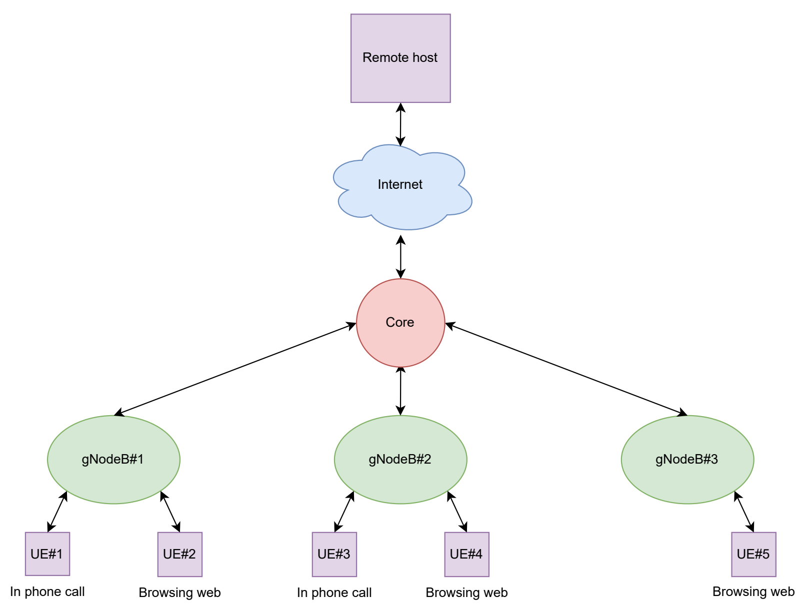

Figure 1: Network structure

Configuration Settings

The simulation script allows flexible adjustment of key parameters through the command-line arguments illustrated in Figure 2. By default, the script is configured to meet the assignment requirements, utilizing the mmWave frequency band, incorporating at least two distinct numerologies for the bandwidth parts (BWPs), and specifying appropriate transmission power settings for the gNodeBs. These parameters can be easily tailored to suit diverse simulation scenarios.

```
    // Command line argument parsing
    std::string direction = "DL";  // Default is "DL"
    std::string mode = "COVERAGE_AREA";  // Default is "COVERAGE_AREA"
    uint32_t udpPacketSizeBrowsing = 25;  // Default UDP packet size for browsing
    uint32_t udpPacketSizeVoiceCall = 50;  // Default UDP packet size for voice calls
    uint32_t lambdaBrowsing = 10000;  // Default lambda for browsing traffic
    uint32_t lambdaVoiceCall = 10000;  // Default lambda for voice traffic
    double totalTxPower = 35.0;  // Default total TX power

    CommandLine cmd(__FILE__);
    cmd.AddValue("direction", "Direction of the REM: 'UL' or 'DL'", direction);
    cmd.AddValue("mode", "Mode for the REM: 'BEAM_SHAPE', 'COVERAGE_AREA', or 'UE_COVERAGE'", mode);
    cmd.AddValue("udpPacketSizeBrowsing", "UDP packet size for browsing traffic in bytes", udpPacketSizeBrowsing);
    cmd.AddValue("udpPacketSizeVoiceCall", "UDP packet size for voice call traffic in bytes", udpPacketSizeVoiceCall);
    cmd.AddValue("lambdaBrowsing", "Packet generation rate (packets/sec) for browsing traffic", lambdaBrowsing);
    cmd.AddValue("lambdaVoiceCall", "Packet generation rate (packets/sec) for voice call traffic", lambdaVoiceCall);
    cmd.AddValue("totalTxPower", "Total transmission power in dBm", totalTxPower);

    // If --PrintHelp is provided, display the help message and exit
    cmd.Parse(argc, argv);
```

Figure 2: Key command-line arguments for the 5G NR ns-3 simulation include REM direction and mode, UDP packet sizes, traffic generation rates, and total transmission power, enabling flexible configuration for diverse scenarios.

Specifically, the NR setup consists of two BWPs operating within the mmWave range requirement for the assignment. The mmWave frequency band, typically ranging from 24 GHz to 100 GHz, is chosen due to its high bandwidth potential, which supports the demanding data rates and low-latency requirements of 5G networks. One BWP uses a numerology of 4 with a central frequency of 28 GHz and a bandwidth of 50 MHz, while the second BWP operates with a numerology of 2 at 28.2 GHz with a similar bandwidth. The total transmission power for both BWPs is set to 35 dBm. The scenario is designed to simulate voice call traffic and web browsing traffic, with the different BWPs allocated for each type. Parameters such as UDP packet sizes and traffic rates for the different traffic types are also adjustable, allowing further customization and analysis.

These modifiable parameters allow us to experiment with different settings and observe how the simulation behaves under varying conditions. This is essential for further analysis and performance evaluation throughout the assignment.

We also set up appropriate IP addresses for all network devices for the assignment. The remote host, simulating the internet connection, is connected to the PGW via a high-speed Point-to-Point link, and the network devices are assigned IPv4 addresses within the range 1.0.0.0 for internet access. The UEs, web-browsing and voice-call devices, are assigned IP addresses on the 7.0.0.0 network. Static routing is configured to ensure proper communication between the UEs and the remote host via the SGW/PGW. This setup allows proper routing and communication between devices in the simulated network.

Traffic Generation

In the traffic generation part of the simulation, we configured two types of traffic: voice call traffic and web browsing traffic. Each type is identified by a specific UDP port and handled separately to simulate realistic network behavior.

The simulation applies different Traffic Flow Templates (TFTs) for both types of traffic. A dedicated bearer for web browsing (with `NrEpsBearer::NGBR_LOW_LAT_EMBB`) and another for voice calls (with `NrEpsBearer::GBR_CONV_VOICE`) are used to manage the traffic flow and ensure proper QoS. The UDP clients for both web browsing and voice calls are configured to interact with the remote host, and the traffic flow starts at the designated start time and stops at the end of the simulation.

These traffic generation parameters allow us to simulate different types of network usage, such as voice calls and web browsing, and observe how the network performs under these different conditions.

Network Simulation Results Analysis

In our simulation, we utilize the FlowMonitor tool to capture detailed metrics about network performance. We then log and output the results of the flows, which represent the captured traffic between nodes in the network. This includes information such as transmitted (Tx) and received (Rx) packets, total bytes transferred, throughput (in Mbps), packet loss, and various delay metrics like mean delay and jitter. While we captured many flows representing traffic between different nodes in the network, for clarity, we focus on a single flow: a node on one eNodeB engaged in a call with another node connected to a different eNodeB. This specific flow allows us to compare the effects of different parameter settings consistently and meaningfully. The complete logs for all flows are in the project files for further review.

The simulation results are presented in Listings 1, 2, and 3, each showcasing the impact of different parameter settings on network performance. In Listing 1, the throughput is 6.22 Mbps, and packet loss is relatively low at 0.24 %, with a delay of around 1.88 ms. These results were obtained using a transmission power of 35 dBm and low traffic levels.

In contrast, Listings 2 and 3 explore scenarios with high traffic rates that significantly increase the offered load to 422.4 Mbps. Although Listing 2 uses a transmission power of 35 dBm and Listing 3 reduces it to 25 dBm, the outcomes are nearly identical, with a throughput of 51.5 Mbps, packet loss at 87.8 %, and delay at 396.5 ms in both cases. This suggests that network performance is dominated by congestion and traffic volume under such high traffic loads, rendering transmission power adjustments ineffective in mitigating these issues.

Listing 1: Simulation with parameters following parameters set: `udpPacketSizeVoiceCall = 50` and `totalTxPower = 35`.

```
Flow 8 (1.0.0.2:49156 -> 7.0.0.5:1235) proto UDP
  Tx Packets: 900
  Tx Bytes:   70200
  TxOffered:  6.240000 Mbps
  Rx Bytes:   70284
  Lost Packets: 2
  Packet loss: 0.18%
  Throughput: 6.22 Mbps
  Mean delay:  0.74 ms
  Mean jitter:  0.016 ms
  Rx Packets: 831
```

Listing 2: Simulation with parameters following parameters set: `udpPacketSizeVoiceCall = 50` and `totalTxPower = 35`.

```
Flow 8 (1.0.0.2:49156 -> 7.0.0.5:1235) proto UDP
  Tx Packets: 9000
  Tx Bytes:   702000
  TxOffered:  62.400000 Mbps
  Rx Bytes:   700308
  Lost Packets: 22
  Packet loss: 0.24%
  Throughput: 62.21 Mbps
  Mean delay:  0.85 ms
  Mean jitter:  0.018 ms
  Rx Packets: 8978
```

Listing 3: Simulation with parameters following parameters set: `udpPacketSizeVoiceCall = 50` and `totalTxPower = 25`.

```
Flow 8 (1.0.0.2:49156 -> 7.0.0.5:1235) proto UDP
  Tx Packets: 9000
  Tx Bytes:   702000
  TxOffered:  62.400000 Mbps
  Rx Bytes:   349794
  Lost Packets: 4527
  Packet loss: 50.3%
  Throughput: 31.0 Mbps
  Mean delay:  22.5 ms
  Mean jitter:  0.03 ms
  Rx Packets: 4473
```

The parameters used in the simulation, such as `udpPacketSizeVoiceCall = 5000`, were deliberately set to unrealistic values to fulfill the assignment's requirement to modify parameters and observe the resulting effects. The results would have been more typical of real-world conditions without these adjustments. For instance, typical voice call traffic operates at much lower data rates than the 51 Mbps observed in Listings 2 and 3. While these changes help assess network performance under different conditions, they lead to unrealistic throughput and packet loss for voice calls, highlighting the need for careful parameter selection to achieve more accurate simulations.

Radio Environment Map Generation

To further facilitate the analysis per the assignment, we utilized the `NrRadioEnvironmentMapHelper` class in NS-3 to generate radio environment maps (REM). These maps are instrumental in visualizing signal propagation, coverage, and interference patterns within a 5G network by modeling different REM modes. The simulation focused on three high-level REM modes, each capturing distinct aspects of signal behavior:

- **Beam Shape**: This mode represents the beam configurations specified in the script, assuming the receiver has a quasi-omni pattern. Those beams are directed toward the user equipment (UE) of interest. The aim is to visualize the REM based on the specific beam configuration in the scenario.
- **Coverage Area**: This mode produces two REMs worst-case SINR (Signal-to-Interference-

plus-Noise Ratio) and best-case SNR (Signal-to-Noise Ratio) for each REM point. The worst-case SINR assumes all interfering devices use the beam directed towards the REM point, while the best-case SNR selects the best directional beam-pair for each device and REM point, optimizing signal quality.

- **UE Coverage**: In this mode, the transmitting device is the UE (uplink direction), and the receiving device is the gNB (next-generation Node B). The beams from other gNBs are directed towards the Rx gNB. In the TDD (Time Division Duplex) case, this mode also displays the interference caused by the downlink transmissions from the other gNBs.

Figures 4 and 5 represent a 12-plot matrix showing performance metrics for downlink (DL) and uplink (UL) directions, respectively. The plots are organized into the three REM modes—Beam Shape (top row), Coverage Area (middle row), and UE Coverage (bottom row) – and include key metrics such as Signal-to-Interference Ratio (SIR), Signal-to-Noise Ratio (SNR), Signal-to-Interference-plus-Noise Ratio (SINR), and Interference Power Spectral Density (IPSD), offering a comprehensive view of the radio environment.

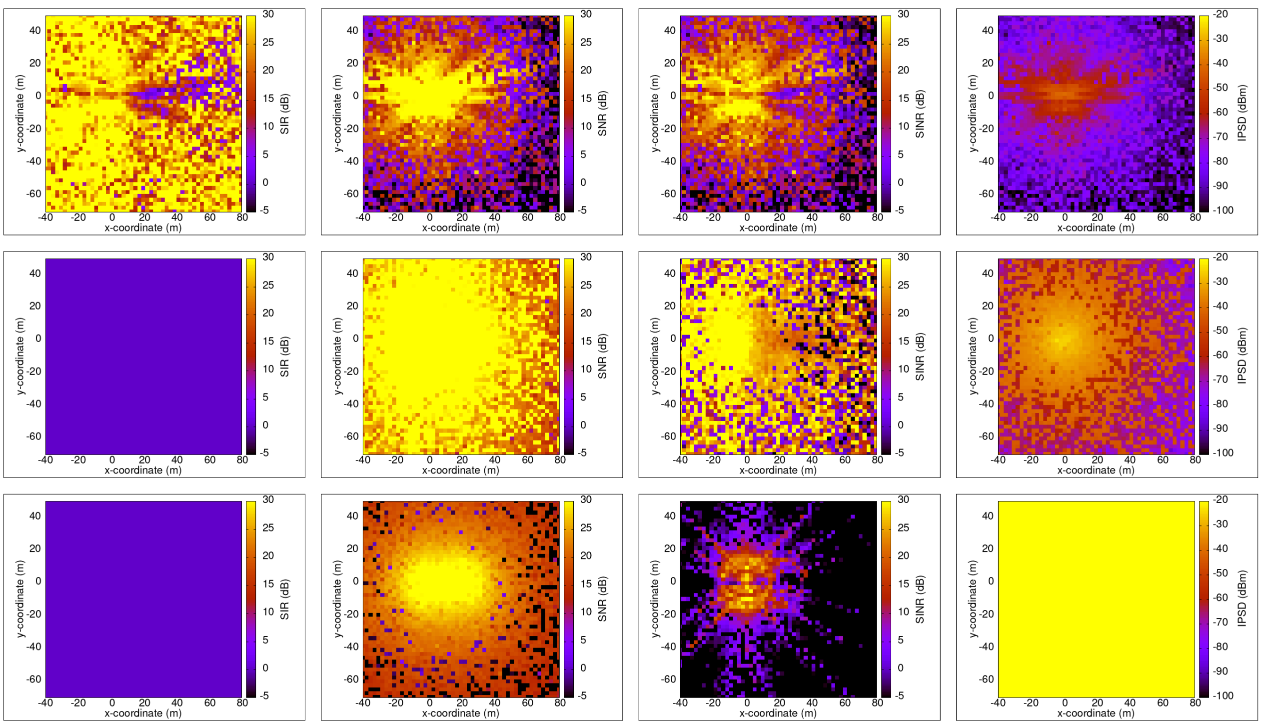

Figure 4: This 12-plot matrix shows key metrics for **downlink** performance across three REM modes: Beam Shape, Coverage Area, and UE Coverage, with SIR, SNR, SINR, and IPSD for each configuration.

Simulation Insights and Challenges

Our simulation models a 5G NR network with two UEs on voice calls and three UEs browsing the web. While this setup is adequate for our analysis, it doesn't reflect the complexity of real-world networks, which support many more UEs and face greater challenges. The simulation uses static mobility models, whereas real-world mobility—especially in 5G—is dynamic and unpredictable. Additionally, idealized models for propagation, interference, and simplified traffic patterns do not

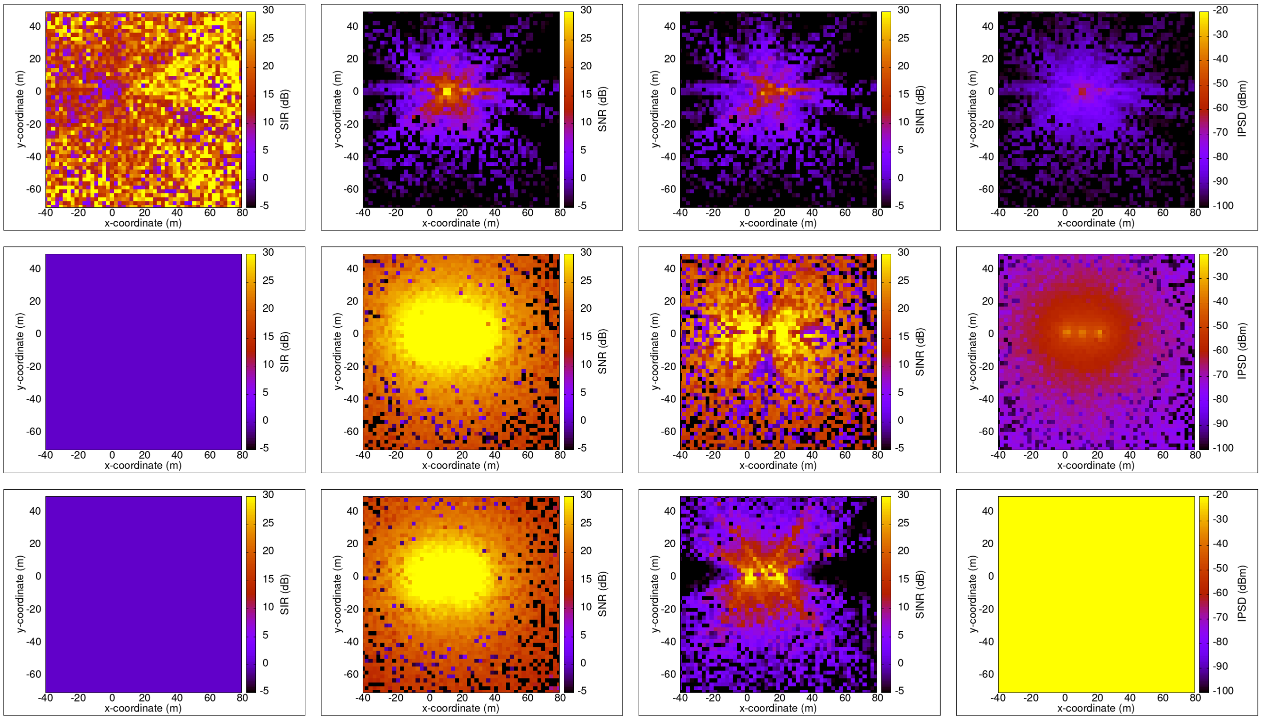

Figure 5: This 12-plot matrix shows key metrics for **uplink** performance across three REM modes: Beam Shape, Coverage Area, and UE Coverage, with SIR, SNR, SINR, and IPSD for each configuration.

capture real-world factors like obstacles, weather, and device variability. Hardware limitations, such as antenna imperfections and delays, are also abstracted in the simulation.
We also encountered challenges, particularly with debugging the ns-3 code. The lack of line-by-line debugging tools made troubleshooting a complex and time-consuming process. Despite these difficulties, this project was an engaging and insightful experience. More complex simulations in the future could incorporate real-world factors, yielding even more accurate and meaningful results.