

Exploring the ns-3 mmWave module

Report

Artur André Oliveira, Daniel Batista, Roberto Hirata Jr.

April 2019

1 Introduction

In this report we described several experiments conducted using the ns-3 simulator and its module mmWave. These experiments were motivated by some research questions, namely how to determine the best positioning of 5g radio base stations. Due to some limitations over the millimeter wave technology with relation to distances between the radio base and an user device great (spatially) scale simulations were not considered. Nevertheless by reading this report one could derive new research questions more trimmed to the capabilities of this module and even improve some of the methods herein shown.

The Network Simulator ns-3[1] is a discrete event simulator that allows one to simulate mobile devices, radio base stations, routers, computers and even obstacles (e.g. buildings) for wireless communications. Although some of its citations in the literature are related to comparisons between network simulators [2, 3], there are papers that directly validates some particular functionality of the ns-3 simulator using real measurements [4], some using mathematical models [5, 6, 7] and in some papers new modules and algorithms, implemented over ns-3, are proposed [8, 9, 10]. In these latter papers the ns-3 is used for evaluation. The mmWave is an enabling technology for 5g networks and also a module (with the same name) for the ns-3 software. The module is built and validated through multiple real world experiments, such as waving a hand between a cellphone and a radio base station in a controlled environment to measure this kind of interference caused by a transient obstacle [1]. Although this controlled environment is limited in the sense that perhaps it could not capture all the possible elements that could affect a 5g communication the scenarios considered in this paper are reasonably simple and thus major discrepancies between them and observations in real similar scenarios are not expected.

The experiments will be conducted over simulations created using the network simulator software ns-3[1]. In order to have more accurate simulations, the real position of avenues and the real size of buildings (as its positions) are used in some of the scenarios.

Our main interest along these experiments is to measure the Signal-to-interference-plus-noise ratio (SINR), as an indirect measure for the Quality of Service, in multiple real case scenarios and also in hypothetical ones. We try to answer the research question

aforementioned by first deriving a setup that allows one to simulate a 5g network and then through some measures determine the optimal positioning of the radio base stations.

Moreover this report also has the purpose of being a starting guide to simulate 5g networks in mmWave that complements its official guides [11, 12, 13].

2 Bibliographic review

According to Bega et al. (2017) [14] and Shancang et al. (2018) [15] the current state of networks based upon the 4g technology will not attend to near future demands like the ones required by Internet of Things (IoT) devices, smart devices, low-latency controlled devices, vehicular networks, etc. The main challenges for such kinds of communications relies on a mixture of low-latency, high bandwidth and high availability. Although the millimeter wave technology (mmWave) enables the 5g technology [16, 11, 14, 17] it still have some difficulties due to high attenuation [17]. Some works like [16] describes the dual connectivity paradigm where a device can handover between the 4g LTE and the 5g network, in order to benefit from 4g stability and 5g advantages.

There are quite a few tools for simulation 5g networks nowadays [18, 19, 20, 11]. Most of them are products targeted at the telecommunication industry as simulation tools. Even though any of them could be used for research, due to its usual high price it becomes unfeasible to compare all of them in order to determine strengths and weaknesses of each one. Since the goal of this report is to provide a guide for implementing and analyse simulations of the 5g networks we choose the ns-3 with the mmWave module to conduct the experiments.

One of the most critical issues when dealing with 5g networks is its low distance of operation (when compared to 4g networks) even with a clear line of sight[11]. In our experiments we noticed that the SINR starts to degenerate quickly as soon as the distance between the 5g radio base station and the user device have a distance of approximately 200 meters between them. So most of the scenarios created in this report assume that the user equipment (i.e. its cellphone) is at most at 200 meters away from the connected 5g radio station.

3 Components of the simulation

In this section details about the simulation implementation and methodology will be described. The simulation will be focused in two actors, the 5g Radio Base Stations, namely evolved Node-B (eNB) radio bases and mobile devices called as User Equipment (UE).

3.1 mmWave

The mmWave module is built upon the LTE one [21, 22] and is meant for creating simulations concerning 5g networks. Due to the nature of how ns-3 modules are constructed and even due to its working philosophy there are some incompatibilities between some other

modules (that may or may not be compatible with the LTE module) and the mmWave one. In order to overcome such incompatibilities and also to make it easy to deploy a working copy of the ns-3 with the mmWave module already set it is recommended to install directly from the mmWave's repository instead of firstly getting a copy of the ns-3 simulator and then installing the module. In the appendix section (at the end of the report) we'll describe one way of installing the mmWave together with the ns-3 simulator and how to run the studied scenarios.

3.2 User Equipment

User Equipment will be modeled as simple systems running networks applications that generates traffic. By constantly monitoring their signal strength, uplink and downlink transmission rates a measure of Quality of Service will be derived such that it'll be used as part of the cost function at a determined position and time in the simulation.

3.3 Radio Base Stations

Radio Base Stations will be modeled using the module mmWave[11] developed over the ns-3 LTE module[21] to incorporate 5g technology characteristics and simulate multiple real life aspects such as interference between radio bases.

3.4 Metrics

In order to measure the Quality of Service we use the Signal-to-interference-plus-noise ratio (SINR) as an indirect measure, and from the application layer we simulate in some scenarios an TCP application sending packages as well as how does the PING application behaves under certain conditions.

4 Scenarios

In this section the scenarios will be grouped according to the research question that originated it. The order of the scenarios reflects nearly the order in which limitations of the platform were being found.

4.1 Where to deploy 5g eNB base stations in order to maximize SINR considering buses trajectories?

In this set of experiments we tried to create scenarios with 5g eNB radio stations, UE devices grouped in containers (simulating users in a bus).

4.1.1 Scenario 1

In this scenario we created a container with 40 UE devices (a full bus) stopped near 5 eNBs (all of them 5g radio stations) located at the same spot. The rationale behind this arrangement is that perhaps with more eNBs the UEs could connect with distinct eNBs and the overall SINR would be better. It turns out that by using the function *AttachToClosestEnb* from the *MmWaveHelper* class all the UEs will connect to the same eNB even when more are available if all eNBs are at the same position. In figure 1 we plot the SINR from all the RNTIs in this scenario.

It's important to notice that the graph in 1 shows the *Radio Network Temporary Identifier* (RNTI) from each UE rather than some unique identifier of the UE. A RNTI is assigned to an UE when it connects to the radio station, when the same UE reconnects (even to the same eNB) a new RNTI is assigned, and each RNTI is assigned in an incremental way, that is, each connection generates a new RNTI, so in this graph it's hard to tell which points belongs to the same UE or even which points represents the same RNTI. In this graph there are 147 distinct RNTIs (which implies that some of the 40 UEs has been reconnected more than once). Our last observation about this scenario is that although 5 eNB radio stations were available (all at the same location) only the first eNB, (referred in data by the *cellId* column with value 1), got connections, none UE connected to any of the other eNBs during the simulation.

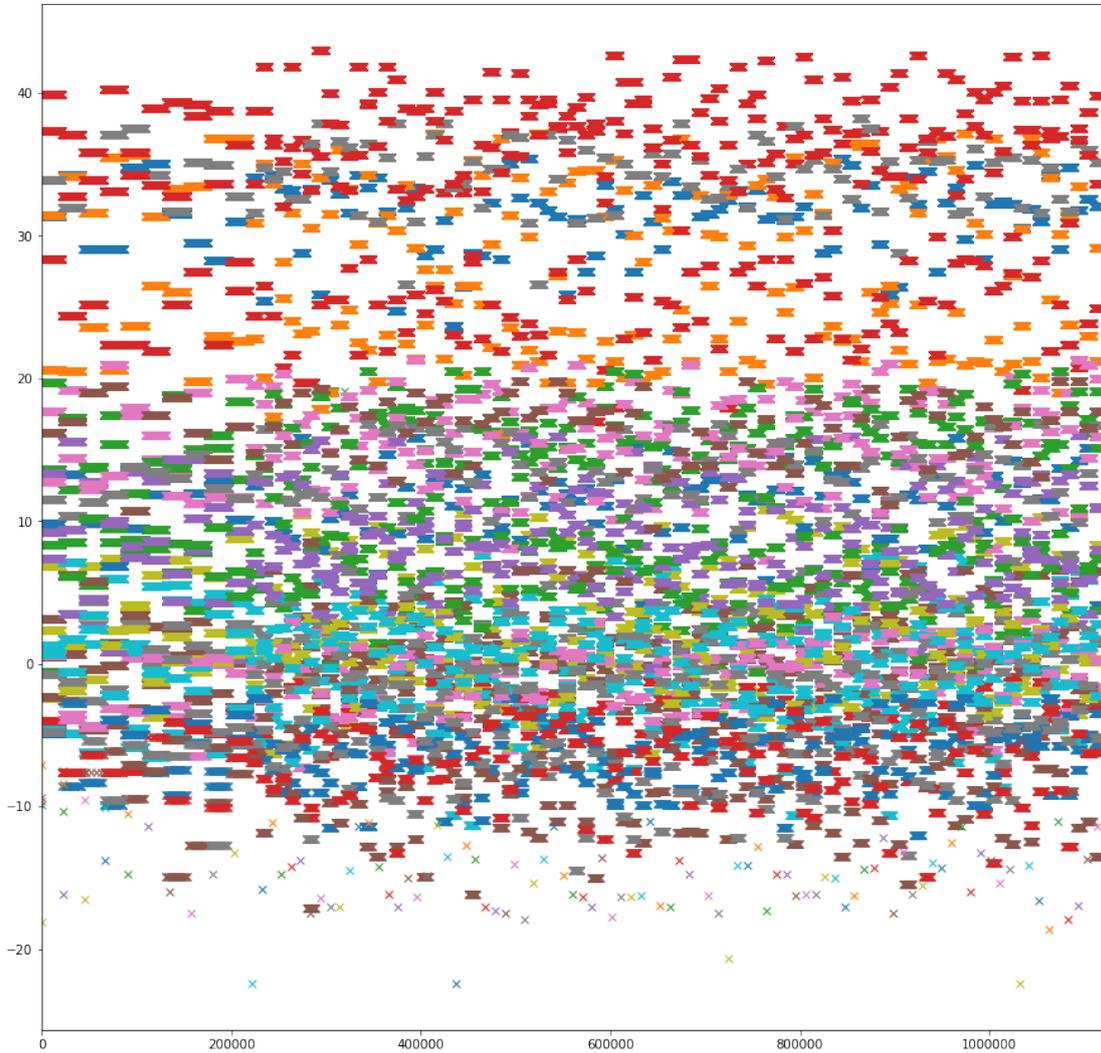


Figure 1: SIRS of 40 devices connecting to the same eNB radio station. Here we have each datapoint as a 'x' symbol, we can notice that due to the granularity of the mmWave (nanoseconds) the SINR is the same for a number of steps. Moreover notice that the x-axis (time here) is relabeled so that each timestep has a unique identifier ranging from 0 to 1119116, that is, although the simulation has a resolution of nanoseconds, implying in possibly 10 billions distinct time points, only nearly 1 million of them are really distinct so that the number of generated events are not so big as the resolution suggests.

4.1.2 Scenario 2

This scenario is exactly like the last one except that instead of having 5 eNBs at the same location now there is only one.

With this we can conclude if the eNBs interfere with each other or not, since in the scenario one we observed that when multiple eNBs are at the same spot the UEs (also in one single location) connects to just one eNB.

We noticed that the SINR from each RNTI is stable around a certain range but, each RNTI had its SINR oscillating in a different range. The expected behaviour was that given an UE its SINR could sometimes be higher and sometimes lower when compared to the others. What we observed is that some RNTIs only exist in a single point in time, that is, a connection is created between an UE and the eNB, generating a RNTI, if the SINR for this connection is too low (lower than 7 dB) then the UE disconnects and tries to reconnect, generating then a new RNTI possibly with a better SINR. By observing this behavior we concluded that the SINR is directly related to the number of RNTIs generated, but once the connection created between the UE and the eNB has a good SINR it is kept until the end of the simulation. Thus to better understand how the connections are created and how the locations of the UEs and the eNBs are related to the created connections new experiments are required.

4.1.3 Scenario 3

Following yet the investigations from the last scenarios now we tried to keep 40 UEs at a fixed location and deployed two 5g eNBs each one at 150 meters away from the UEs but in antipodal locations around the UEs. Still the conclusions and observations are quite the same as the last two scenarios. So even in distinct locations, if the distance between an UE and two eNBs the first eNB considered by the simulator will be the only one used.

4.2 How to deal with buildings and moving UEs?

In this set of experiments we considered eNBs and UEs with obstacles (i.e. buildings) between them. Moreover the UEs are allowed to move, so that they can have a direct line of sight to the eNB in some parts and an obstruction in others during the same simulation.

4.2.1 Scenario 4

In this scenario two real world buildings¹ and a real world avenue are simulated together with one UE crossing the avenue. There is one single eNB located behind one of the buildings (from the point of view of the UE) so that during part of the simulation the UE has direct line of sight to the eNB and in some parts it is blocked by the buildings. Notice that the distance between the UE and the eNB changes very little during the whole simulation. So the minimum distance is 50 meters and the maximum distance is nearly 80 meters. Besides in this simulation in order to capture the RTT and the CWND one remote host (i.e. a web-server over the internet) is inserted in the simulation, so that this remote host sends data packages to each UE.

In figure 2 the SINR of the UE (in blue) is plotted over its RTT graph (orange). The time axis (horizontal one) is converted from nanoseconds to timesteps, the whole simulation lasts 10 seconds or equivalently 297328 distinct timesteps. Notice that the RTT is scaled so that its order of magnitude (10^{-2}) becomes comparable to the one of the SINR. In figure 3 it is plotted the RTT graph and from this graph we can notice that the RTT (except at the first moments) is quite stable during the simulation. One possible reason for that is that although the simulation contains some obstacles the eNB is relatively near to the UE during the whole simulation. So the most noticeable characteristic is that the SINR's mean decreases slightly as the UE moves away from the eNB during the simulation.

In figure 4 we plot the SINR and the TCP's congestion window size (referred from now as CWND) over the same graph. Again due to fact that both values have very distinct ranges the TCP values were re-scaled so that its behavior could be compared with that of the SINR. In figure 5 the CWND graph is shown alone. As will be see later, for this kind of scenario with the eNB always close to the UE the CWND and the RTT won't change as a result of the variations of the SINR.

¹Located at the Paulista Avenue, 119 and 149, they are the Sesc Avenida Paulista and Instituto Itaú Cultural respectively.

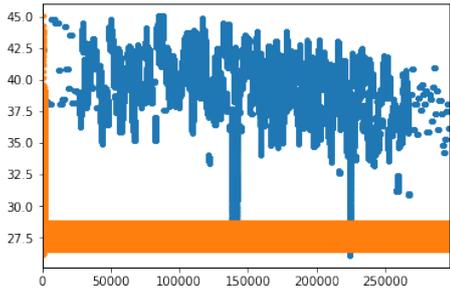


Figure 2: SINR and RTT plotted over the same graph.

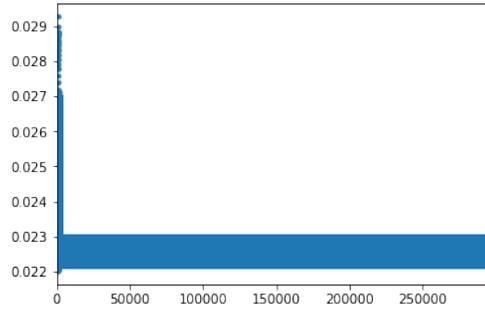


Figure 3: RTT graph.

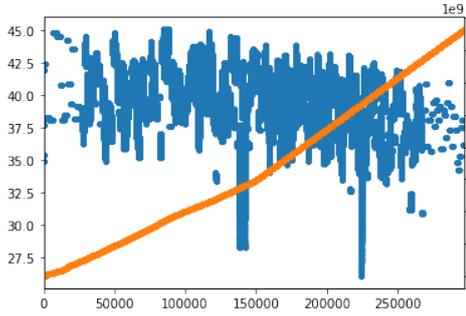


Figure 4: SINR and TCP's congestion window size plotted over the same graph.

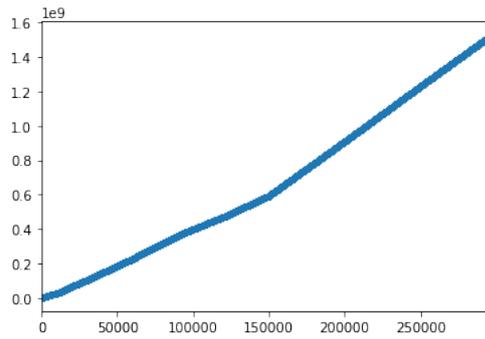


Figure 5: TCP's congestion window size graph.

4.2.2 Scenario 5

This scenario follows the same setting as the last one, except that the UE moves now backwards (starting far away from the eNB and ending near it).

As mentioned before, the graphs in the figures 6, 7, 8 and 9 have a behavior quite similar (for the RTT and the CWND metrics) as those of the last scenario, since the SINR is good for the whole simulation and this last fact derives from the proximity from the eNB to the UE. Once again the most noticeable characteristic now is that the mean of the SINR in the figures 6 and 8 improves as the UE comes closer to the eNB.

4.2.3 Scenario 6

Since the behavior of the SINR, RTT and of the CWND, has proven to be quite consistent for a simple scenario with one moving UE, one single eNB and two buildings, in this scenario we tried to combine the two last ones. Now we have two UEs moving one towards the other.

In figure 10 the SINR is analysed for two types of events, the Download events (DL), that is, packets sent from the eNB to the UE, and the Upload events (UL), those events

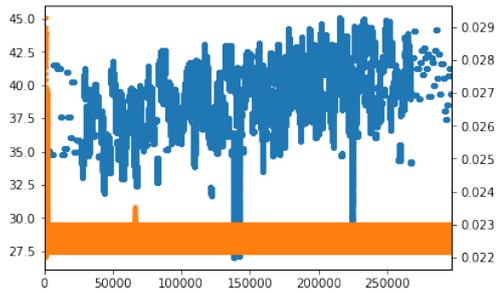


Figure 6: SINR and RTT plotted over the same graph.

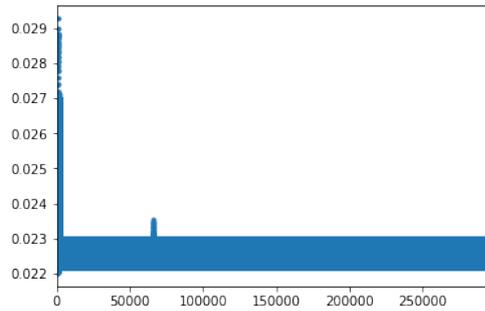


Figure 7: RTT graph.

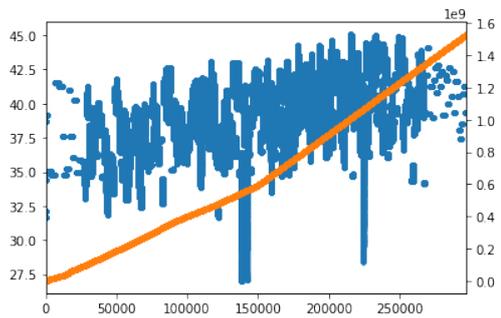


Figure 8: SINR and TCP's congestion window size plotted over the same graph.

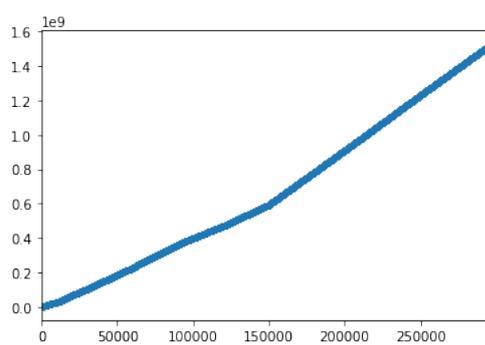


Figure 9: TCP's congestion window size graph.

happens when a packet is sent from the UE to the eNB. Notice that this nomenclature (DL and UL) only specifies the direction of the event, they are not related to the application that generates the RTT and CWND logs. Similarly in figure 11 we can see the SINR for the UE 2 during the simulation, both for the UL and DL directions.

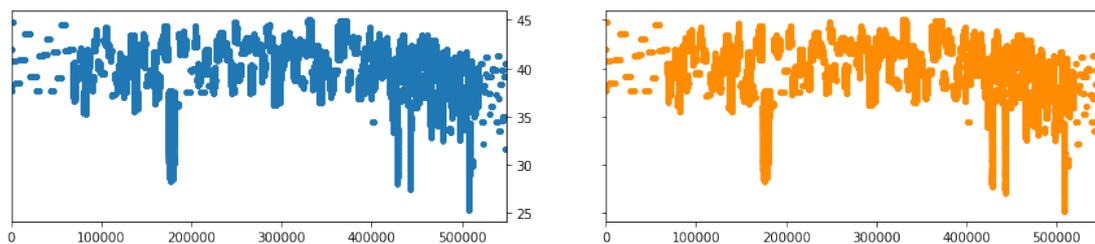


Figure 10: SINR for the connection between the UE 1. The left side shows the SINR capture during DL and the right side the SINR during UL.

In figures 12 13 we can see that the RTT, for the UE 1 and UE 2 respectively, is almost not affected by the SINR. Note that the graphs in the figures are scatter plots

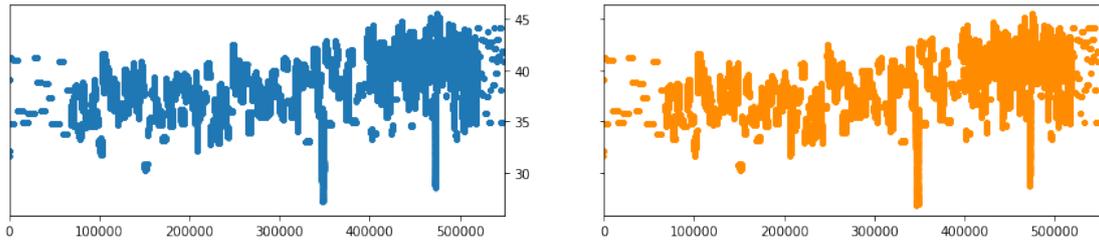


Figure 11: SINR for the connection between the UE 2. The left side shows the SINR capture during DL and the right side the SINR during UL.

(with dots) for both the SINR and the RTT. The thick horizontal orange line is just a large set of points that oscillates in the same small range for almost the whole duration of the simulation.

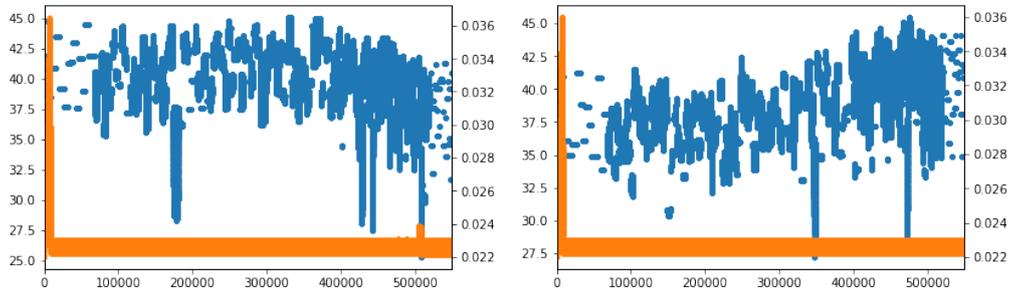


Figure 12: SINR (blue) and RTT (orange) for the UE 1. Figure 13: SINR (blue) and RTT (orange) for the UE 2.

In figures 14 and 15 the congestion window of the TCP application over the UEs are plotted together with their SINR. From these graphs it seems that the SINR doesn't affect none of the congestion windows standard growth behavior.

4.2.4 Scenario 7

Now in addition to the last scenario we also inserted a second eNB (at the same distance from the second UE as the first one with relation to the first UE) so that now each UE will be getting close to an eNB at the same time that it is getting away from another. With this scenario we want to check how the UEs interfere with each other and how is the interaction of the eNBs in the presence of the obstacles.

In figure 16 it is shown the SINR for the connection between the first UE and the first eNB. We notice that although the signal gets quite degraded by the end of the simulation still the first UE keeps its connection with the first eNB instead of connecting to the second one (now closer to it than the first eNB). Moreover the SINR in this scenario varies more than in the previous scenarios in which it was oscillating only in the range between 30 and 50 decibels. In the figure 17 we can see the SINR between the second

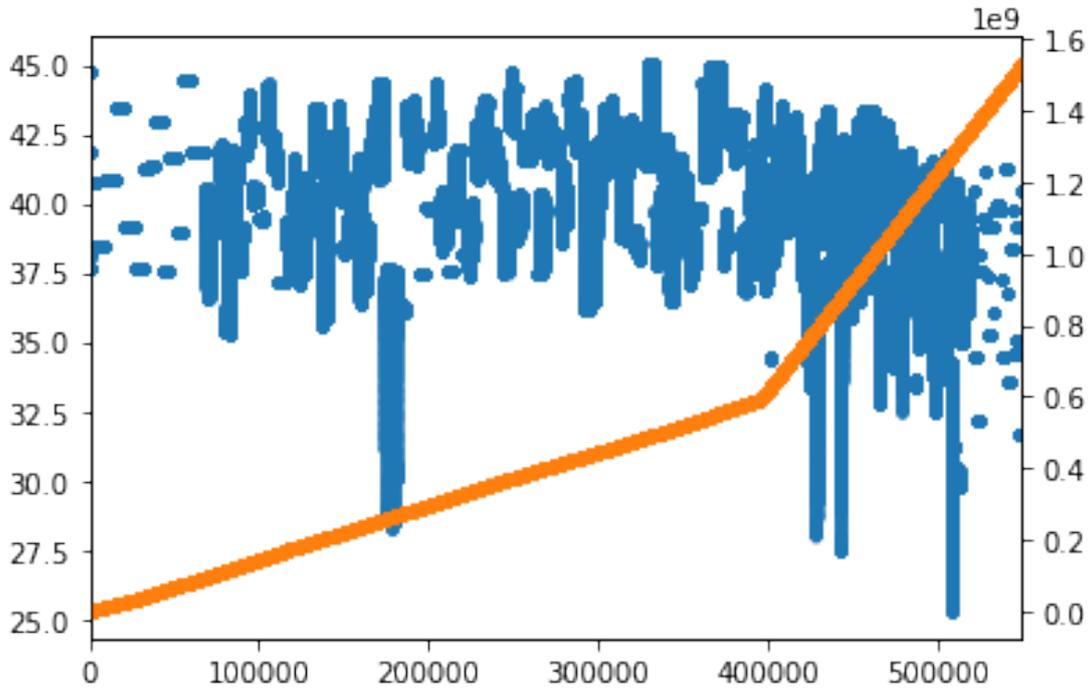


Figure 14: SINR (blue) and CWND (orange) for the UE 1.

UE and the second eNB during the simulation. It is similar to the SINR of the figure 16, which was expected given the symmetry of the scenario.

In figure 18 it is shown the SINR (in blue) and the RTT (in orange), over the same graph, for the connection between the UE 1 and the eNB 1. In this figure it's not clear what is the relationship between the SINR and the RTT, but it is clear that as the UE get further away from the eNB the RTT tends to get larger. In the figure 19 it is shown the RTT in its original scale (seconds).

Now in the figures 20 and 21 it is shown the SINR and the RTT between the second UE and the second eNB. The relationship between the SINR and the RTT still not clear. This time the RTT oscillates more than the RTT between the UE 1 and the eNB 1, but the maximum value now is 0.12 seconds, while before it was 0.25 seconds.

One point worth of attention is that between the timestep 100000 and the timestep 150000 there is a gap. This gap could in principle be a moment where the UE and the eNB lost connection, but in this case the reconnection would imply in a new RNTI being assigned to the second UE, but only one RNTI is present in the eNB 2 during the whole simulation, thus the UE and the eNB keeps its connection throughout the simulation.

In figures 22 and 23 it is shown the SINR and the CWND between the UE 1 and the eNB 1. In this graphs once again there seems to be no relationship between the UE and the congestion windows size. Now when it comes to the same measurements between the UE 2 and the eNB 2 the behavior after the same gap observed in the RTT graph (for

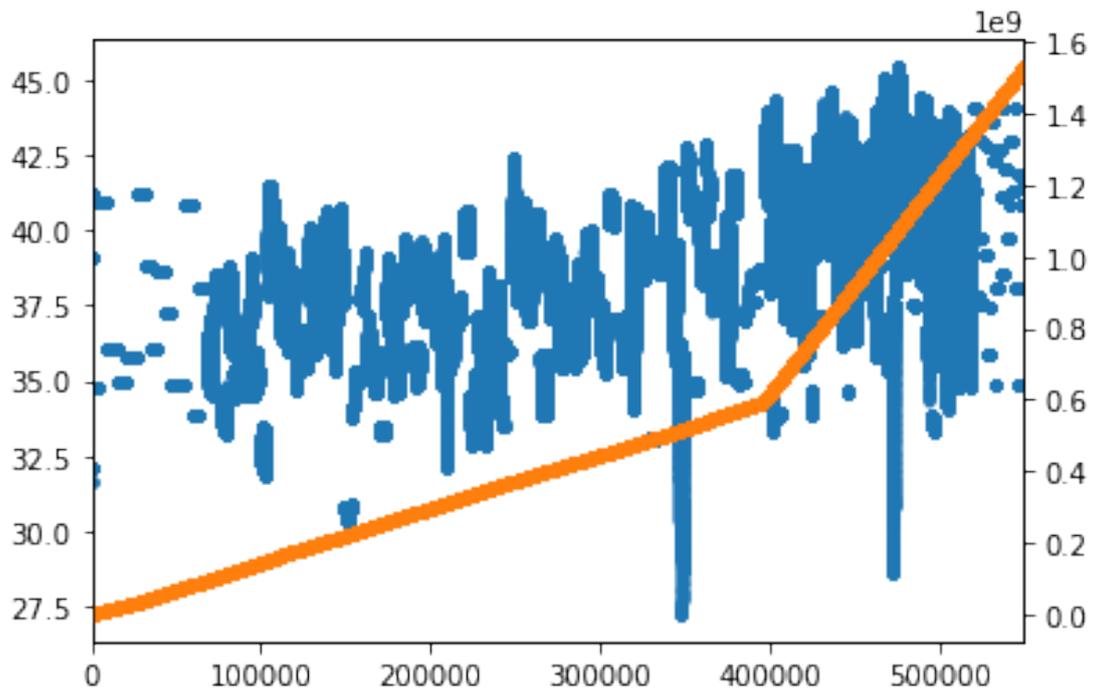


Figure 15: SINR (blue) and CWND (orange) for the UE 2.

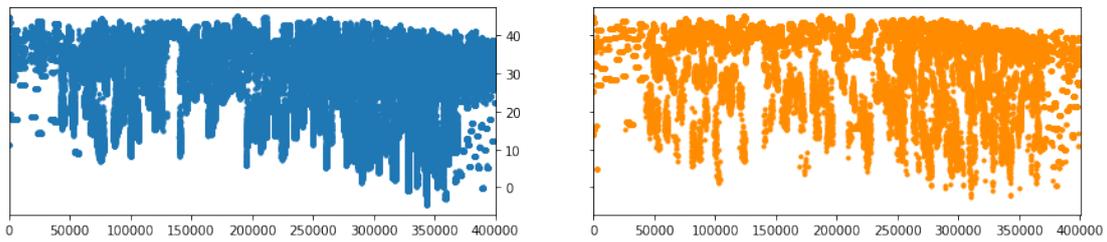


Figure 16: SINR for the connection between the UE 1 and the eNB 1.

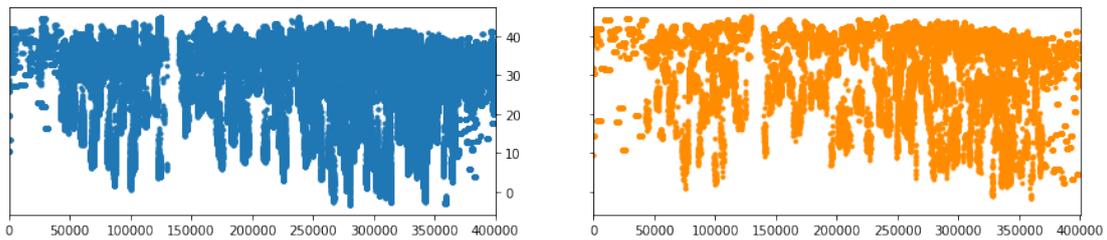


Figure 17: SINR for the connection between the UE 2 and the eNB 2.

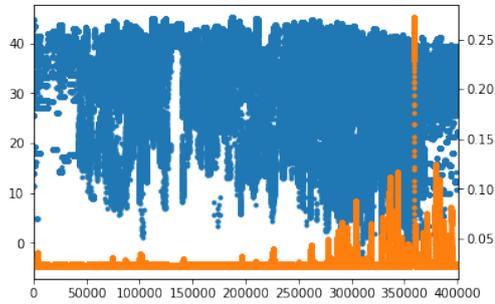


Figure 18: SINR and RTT graphs for the UE 1 and the eNB 1. Notice that the RTT graph (the orange one) was re-scaled.

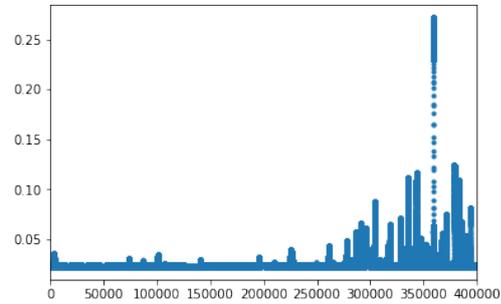


Figure 19: RTT graph for the UE 1 and the eNB 1.

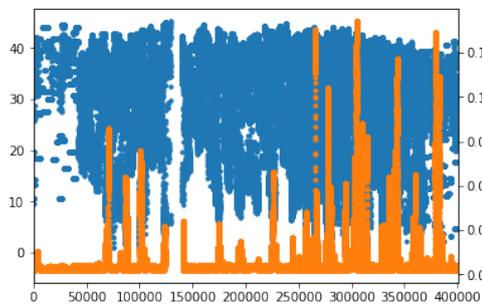


Figure 20: SINR and RTT graphs for the UE 2 and the eNB 2. Notice that the RTT graph (the orange one) was re-scaled.

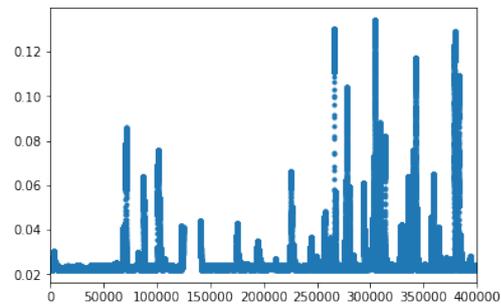


Figure 21: RTT graph for the UE 2 and the eNB 2.

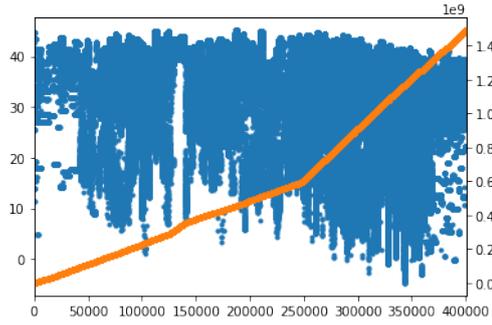


Figure 22: SINR and CWND graphs for the UE 1 and the eNB 1. Notice that the CWND graph (the orange one) was re-scaled.

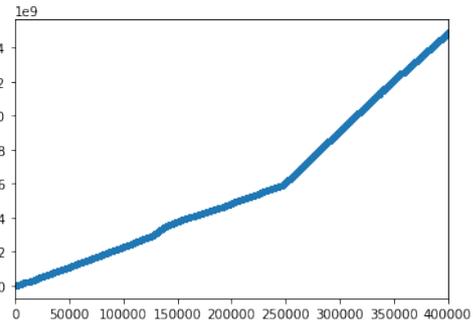


Figure 23: CWND graph for the UE 1 and the eNB 1.

the UE 2 and eNB 2) is different. In Figure 25 the SINR and CWND are plot together and firstly the CWND grows quite fast (when compared to the CWND of the UE 1 and eNB 1) reaching it's peak just before the gap, then it keeps low values for the rest of the simulation.

In figure 24 it is shown the packet size received by the UE 2 by the remote host during the simulation. Although the size of the package received is always the same, since the CWND is at least 1400 at all moments, during the gap (observed in figures 17, 21 and 26) no packages were received.

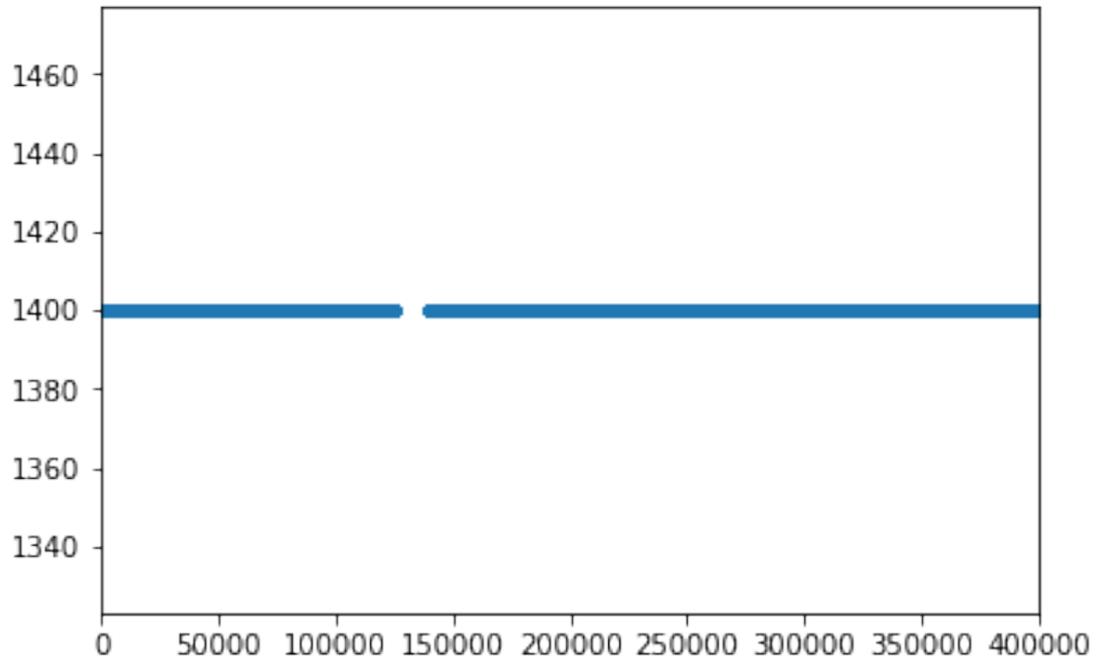


Figure 24: Packet sizes received by the UE 2.

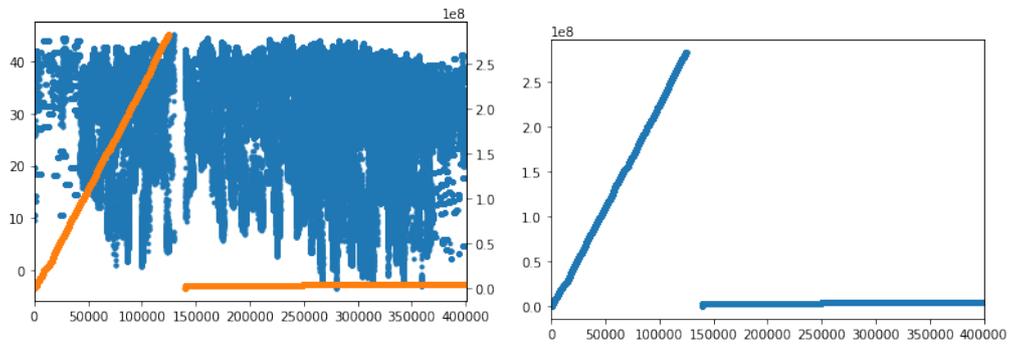


Figure 25: SINR (in blue) and CWND (in orange) graphs for the UE 2 and the eNB 2.

Figure 26: CWND graph for the UE 2 and the eNB 2.

5 Conclusion

Our primary objective in these series of experiments is to derive a feasible and accurate way to estimate the optimal positioning of 5g eNBs in the urban context. Through the experiments done it was possible to delineate limitations and good practices when using the ns-3 simulator with its mmWave module.

Since the range of options and configurations to create a scenario as well as the complexity of the interaction between the elements of the simulation is very high more investigations are needed in order to reach a better understanding of the causes of degradation of the several measures one can take when inspecting the connection between an UE, an eNB and even a remote host. Consequently by reaching to a better comprehension about the causes and effects of each element of the simulation over those measures one could start investigations about how this measures and the Quality of Service are related.

5.1 Validity of the simulation

The mmWave module itself is built taking into account multiple real life aspects (i.e. the interference caused by the hand of the user). Even though in this report we tried to keep it close to a real scenario by exploiting available real buildings and avenues relative locations as well as real 4g radio base stations locations.

The same approaches used here could be extended in order to simulate larger scenarios with more eNBs, UEs, buildings and remote hosts. The scenarios (as well as Jupyter Notebooks with the analyses) can be found at [23].

6 Appendix

Here we detail how to install and run simulations using the mmWave module from the ns-3 simulator. Furthermore we also describe two source code scenarios that are variations from the base sources codes used to develop all the scenarios in this report.

6.0.1 Installing

In order to install the mmWave with the ns-3 simulator one can go to the official repository [24] clone it locally and then run the following commands from the root directory (i.e. .../ns3-mmwave/):

```
./waf configure --enable-examples
./waf --run mmwave-example
```

If everything worked correctly it'll be printed to the output a sequence of lines like ******* UPDATING CHANNEL MATRIX (instance 79) ******* and some output files (i.e. RxPacketTrace.txt) will be generated at the end of the simulation.

6.0.2 Running a source file code

Once the mmWave repository[24] has been cloned one can simply copy a simulation source code (i.e. **mycode.cc** into the scratch folder and then execute the following commands:

```
./waf
./waf --run mycode
```

The first line will build the scenario and the second one effectively execute it.

The scenarios described in this reported are available for download at [23].

Understanding the code from the scenarios

There are two files that were used to create all our scenarios, the **mmwave-tcp-building-example.cc** and the **mmwave-example.cc**. The former one considers user equipment, radio base stations, buildings and remote hosts (i.e. a web-server over the internet), the latter considers just user equipment and radio base stations. In scenario 3 (section 4.1.3) it is presented a scenario with 40 user devices and 2 radio base stations. This scenario is based on the file **mmwave-example.cc** and we'll use it to describe how to modify an already existing simulation:

```
int ueNodesNum = 40;
int enbNodesNum = 2;
```

After the boilerplate part used to set details concerning the mmWave physical and link layers this two variables are set in order to determine how many user devices (`ueNodesNum`) and how many radio base stations (`enbNodesNum`) will be created in this scenario.

```
NodeContainer enbNodes;
NodeContainer ueNodes;
```

In the ns-3 simulator one can define "containers" of objects. Applying a property to a container has the same effect of applying it to all its internal objects.

```
enbNodes.Create(enbNodesNum);
ueNodes.Create(ueNodesNum);
```

Here we actually initialize 2 objects in the **enbNodes** container and 40 in the in the **ueNodes** container.

```
Ptr<ListPositionAllocator> enbPositionAlloc =
CreateObject<ListPositionAllocator> ();
```

This line instantiates a list of positions, that is, an array that will receive coordinates that will define the relative position of objects in the simulation.

```
enbPositionAlloc->Add (Vector (0,0,3));
enbPositionAlloc->Add (Vector (0,300,3));
```

Since in this scenario there are two radio base stations (referred also as eNBs) with distinct positions we need to insert in the **enbPositionAlloc** list two coordinates, here (by convention) the coordinate system will consider the first position of the **Vector**

object as a longitudinal coordinate, the second one as a latitudinal and the third one as height. The ns-3 simulator doesn't take into account the geometry of the space, so one could switch the coordinate system without changing the outputs (as long as it is kept consistent).

```
Ptr<ListPositionAllocator> uePositionAlloc =  
CreateObject<ListPositionAllocator> ();
```

```
uePositionAlloc->Add (Vector (0,150,1));
```

Similarly to what is done to define the location of the radio base stations, in this part the position of the (ueNodes) container will be set, notice that since all 40 user devices are initialized inside this container then all of them will be located at the same position.

Now the next explored file is the one of the scenario 7 (section 4.2.4):

This file starts defining an application (a simulated application to run over the UEs and on a remote host) and some functions used to generate output logs.

```
int scenario = 2;
```

The base file of this scenario (called mmwave-tcp-building-example.cc) contains inside it another 3 other scenarios regarding the positioning of the buildings simulated for our purposes only the scenario 2 is modified and the other internal scenarios are kept untouched.

```
double stopTime = 25;  
double simStopTime = 25;
```

These two variables controls when the simulated application (the one running in the UEs) will be halted and the latter variable defines for how long the simulation (the scenario 7) will run. Here both variables are defined in seconds and the former should be smaller than or equal to the latter.

```
bool tcp = true;
```

This scenario also allows one to simulate an UDP application sending and receiving packages, but since we're interested in the round-trip-time (RTT) of a package sent we'll keep this flag as true.

```
// Create a single RemoteHost  
NodeContainer remoteHostContainer;  
remoteHostContainer.Create (1);  
Ptr<Node> remoteHost = remoteHostContainer.Get (0);
```

In this session the remote host responsible for sending packages to the UEs is created. Firstly it is initialized as a node from a node container and then the pointer to the single node created is assigned as the variable **remoteHost**.

```
InternetStackHelper internet;  
internet.Install (remoteHostContainer);
```

Here the **InternetStackHelper** class is used to simulate an internet stack so that this computer could be accessed by the UEs through a 5g connection.

```
Ptr < Building > building1;
building1 = Create<Building> ();
//Sesc building
building1->SetBoundaries(Box(0.0, 0.0, 0.0, 15.66, 34.87, 58.85));

//Itau Cultural building
Ptr < Building > building2;
building2 = Create<Building>();
building2->SetBoundaries(Box(39.03, 0.0, 0.0, 13.93, 33.68, 58.9));
```

This block (starting at line 381) defines the two buildings used in the scenario 7. The buildings are defined after they are instantiated by the function **SetBoundaries** that receives an object **Box** as its parameter. The constructor of the **Box** object is called with the parameters:

1. longitudinal position,
2. latitudinal position,
3. altitude,
4. width,
5. depth,
6. height

After defining the buildings the radio base stations and the user devices are instantiated and positioned.

```
ueNodes.Get (0) ->GetObject<MobilityModel> () ->
SetPosition (Vector (-20, 40, 1));
ueNodes.Get (1) ->GetObject<MobilityModel> () ->
SetPosition (Vector (60, 40, 1));
```

Here the positions of each UE is defined manually (rather than using the container).

```
ueNodes.Get(0) ->GetObject<ConstantVelocityMobilityModel> () ->
SetVelocity (Vector (0, 0, 0));

ueNodes.Get (1) ->GetObject<ConstantVelocityMobilityModel> () ->
SetVelocity (Vector (0, 0, 0));
```

They are initialized stopped at first.

```

for (int i = 2; i < 22; i++)
{
    Simulator::Schedule (Seconds (i),
        &ChangeSpeed,
        ueNodes.Get (0), Vector (4, 0, 0));

    Simulator::Schedule (Seconds (i),
        &ChangeSpeed,
        ueNodes.Get (1), Vector (-4, 0, 0));
}

```

This loop is used as a trick to trigger the ChangeSpeed function and consequently output to the file **mobility-trace-example.mob** the current position of each UE at every second. This is useful for debugging the scenario. In order to just move an UE only a single call to the Schedule function, indicating when the speed should be changed and what is the velocity, is enough.

```

Simulator::Schedule (Seconds (22),
    &ChangeSpeed,
    ueNodes.Get (0), Vector (0, 0, 0));

```

```

Simulator::Schedule (Seconds (22),
    &ChangeSpeed,
    ueNodes.Get (1), Vector (0, 0, 0));

```

Here both UEs are set to stop after 22 seconds of simulation, since the simulation lasts 25 seconds, in the last 3 seconds the UEs are stopped.

```

// Install LTE Devices to the nodes
NetDeviceContainer enbDevs =
mmwaveHelper->InstallEnbDevice (enbNodes);

```

```

NetDeviceContainer ueDevs =
mmwaveHelper->InstallUeDevice (ueNodes);

```

Here the eNBs container **enbNodes** will be set to be radio base stations and the UEs container **ueNodes** will be set to be LTE devices with the mmWave communication capability.

```

Ptr<OutputStreamWrapper> stream10 =
asciiTraceHelper.
CreateFileStream ("mmWave-tcp-window-newreno0.txt ");

```

```

Ptr<OutputStreamWrapper> stream11 =
asciiTraceHelper.
CreateFileStream ("mmWave-tcp-window-newreno1.txt ");

```

Here we can define the name of the output files containing the size of the congestion window of the connection of the first UE (the file ending in newreno0.txt) and of the second UE (...newreno1.txt).

```
Ptr<OutputStreamWrapper> stream40 =  
asciiTraceHelper.CreateFileStream  
("mmWave-tcp-rtt-newreno0.txt");
```

```
Ptr<OutputStreamWrapper> stream41 =  
asciiTraceHelper.CreateFileStream  
("mmWave-tcp-rtt-newreno1.txt");
```

Similarly, here one can define the name of the output files for the logs of the RTT times for both UEs.

```
Ptr<OutputStreamWrapper> stream20 =  
asciiTraceHelper.CreateFileStream  
("mmWave-tcp-data-newreno0.txt");
```

```
Ptr<OutputStreamWrapper> stream21 =  
asciiTraceHelper.CreateFileStream  
("mmWave-tcp-data-newreno1.txt");
```

At last here we can define the names of the output files for the size of the data packets received by each UE from the remote host.

References

- [1] Thomas R Henderson, Mathieu Lacage, George F Riley, Craig Dowell, and Joseph Kopena. Network simulations with the ns-3 simulator. *SIGCOMM demonstration*, 14(14):527, 2008.
- [2] Stig Bosmans, Siegfried Mercelis, Joachim Denil, and Peter Hellinckx. Testing iot systems using a hybrid simulation based testing approach. *Computing*, 101(7):857–872, 2019.
- [3] Hang Zhang, Xi Wang, Parisa Memarmoshrefi, and Dieter Hogrefe. A survey of ant colony optimization based routing protocols for mobile ad hoc networks. *IEEE Access*, 5:24139–24161, 2017.
- [4] Paul Fuxjaeger and Stefan Ruehrup. Validation of the ns-3 interference model for ieee802.11 networks. In *2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)*, pages 216–222. IEEE, 2015.
- [5] Rohan Patidar, Sumit Roy, Thomas R Henderson, and Morteza Mehrnoush. Validation of wi-fi network simulation on ns-3. 2017.

- [6] Le Tian, Sébastien Deronne, Steven Latré, and Jeroen Famaey. Implementation and validation of an ieee 802.11 ah module for ns-3. In *Proceedings of the Workshop on ns-3*, pages 49–56. ACM, 2016.
- [7] Richard Rouil, Fernando J Cintrón, Aziza Ben Mosbah, and Samantha Gamboa. Implementation and validation of an lte d2d model for ns-3. In *Proceedings of the Workshop on ns-3*, pages 55–62. ACM, 2017.
- [8] Jeroen van der Hooft, Stefano Petrangeli, Maxim Claeys, Jeroen Famaey, and Filip De Turck. A learning-based algorithm for improved bandwidth-awareness of adaptive streaming clients. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 131–138. IEEE, 2015.
- [9] Janaka Wijekoon, Rajitha Tennekoon, Erwin Harahap, and Hiroaki Nishi. Introducing a distance vector routing protocol for ns-3 simulator. In *SimuTools*, pages 38–46, 2015.
- [10] Christopher Thomas. Ns-3 simulation of wimax networks. 2011.
- [11] Marco Mezzavilla, Menglei Zhang, Michele Polese, Russell Ford, Sourjya Dutta, Sundeep Rangan, and Michele Zorzi. End-to-end simulation of 5g mmwave networks. *IEEE Communications Surveys & Tutorials*, 20(3):2237–2263, 2018.
- [12] ns-3-mmwave github repository welcome page. <https://github.com/nyuwireless-unipd/ns3-mmwave/wiki/Welcome>. Accessed: 2019-06-11.
- [13] ns-3 documentation page. <https://www.nsnam.org/documentation/>. Accessed: 2019-06-11.
- [14] Dario Bega, Marco Gramaglia, Carlos Jesus Bernardos Cano, Albert Banchs, and Xavier Costa-Perez. Toward the network of the future: From enabling technologies to 5g concepts. *Transactions on Emerging Telecommunications Technologies*, 28(8):e3205, 2017.
- [15] Shancang Li, Li Da Xu, and Shanshan Zhao. 5g internet of things: A survey. *Journal of Industrial Information Integration*, 10:1–9, 2018.
- [16] Michele Polese, Marco Mezzavilla, and Michele Zorzi. Performance comparison of dual connectivity and hard handover for lte-5g tight integration. In *Proceedings of the 9th EAI International Conference on Simulation Tools and Techniques*, pages 118–123. ICST (Institute for Computer Sciences, Social-Informatics and . . .), 2016.
- [17] Imtiaz Parvez, Ali Rahmati, Ismail Guvenc, Arif I Sarwat, and Huaiyu Dai. A survey on low latency towards 5g: Ran, core network and caching solutions. *IEEE Communications Surveys & Tutorials*, 20(4):3098–3130, 2018.
- [18] 5g toolbox: Model, simulate, design and test 5g systems with matlab video - matlab. <https://www.mathworks.com/videos/>

5g-model-simulate-design-and-test-5g-systems-with-matlab-1538401305240.html. Accessed: 2019-06-11.

- [19] 5g and mimo simulation software. <https://www.remcom.com/5g-urban-small-cell-analysis>. Accessed: 2019-06-11.
- [20] End-to-end 5g network simulator - polaris networks. <http://www.polarisnetworks.net/5g-network-emulators.html>. Accessed: 2019-06-11.
- [21] Overview of the lte-epc simulation model. <https://www.nsnam.org/docs/models/html/lte-design.html#fig-epc-topology>. Accessed: 2019-04-10.
- [22] ns-3: Lte models. https://www.nsnam.org/docs/release/3.29/doxygen/group__lte.html. Accessed: 2019-06-11.
- [23] ns-3-mmwave-scenarios github repository. <https://github.com/arturandre/ns3-mmwave-scenarios>. Accessed: 2019-06-11.
- [24] ns-3-mmwave github repository. <https://github.com/nyuwireless-unipd/ns3-mmwave.git>. Accessed: 2019-06-11.