|  |
| --- |
| **Cell Selection**  BLIER Lukas  CARTERET Thomas  SURATTEAU-HONIGMANN Maxime |



Summary

Table of Contents

Summary 3

I – Introduction 4

II – Conception 5

**1)** **Simulation of a User Equipment in UMTS** 5

**2)** **Selection and soft/hard handover procedure** 5

III – Implementation 7

**1)** **Moving User Equipment** 7

**2)** **Implementation of the selection process** 9

**3)** **Implementation of the re-selection process** 10

IV – User Manual 10

V – Conclusion 14

**1)** **??** 14

I – Introduction

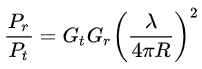
In the context of the RE56 project, we produce a cell selection project mainly consisting of a communication simulation system in UMTS standard. The main objective is to focus on the selection and handover procedures. The system studied will be simplified and should consist of a single User Equipment travelling the map with a set of antennas.

Our simulation should be based on “Radio Mobile Network Tool” which is a Web application simulating radio mobile communication, the project should be a plugin of this app. We should simulate a User Equipment travelling the map with its current connection parameter displayed. These parameters will be produced according to antenna and UE settings. In a second part, we should simulate the antenna selection and reselection procedures.

II – Conception

1. Simulation of a User Equipment in UMTS

As far as we are not taking into account, the relief and city infrastructure, we will use the Friis formula. Indeed, this equation represent radio propagation in free space. That’s why we will only consider three parameters to simulate the communication, the antenna gain, the UE gain and the distance between those two. Those three losses should be enough to have a satisfying simulation for our project.



With the two linear gains at transmitter and receiver and the distance between antenna and UE. Finally, is the carrier frequency.

As we are not working in linear scale but with , we will use another version of this equation:

Avec

On a , (, et .

1. Selection and soft/hard handover procedure

**Selection**

UMTS: based on Pilot CPICH received signal level (RSCP) and SINR (Ec/Io) of cells belong to the BA List

Ec/Io = CPICHattachedcell – ( watt2db ( ∑ db2watt(CPICHothercell)

On activation, the UE ranks the neighboring cells according to the SQual and SRxLev criteria with both values > 0

· SQual = Ec/Io measure – (Ec/Io min + Ec/Io\_min\_offset) > 0

· SRxLev = RSCP measure – (RSCP min + RSCP \_min\_offset) – Pcompensation > 0

**Reselection**

The cell from the BA List with higher SQual and SRxLev criteria will be re-selected by

the UE. BA cells list is the neighboring cells of the currently attached cell of the UE

**Handover**

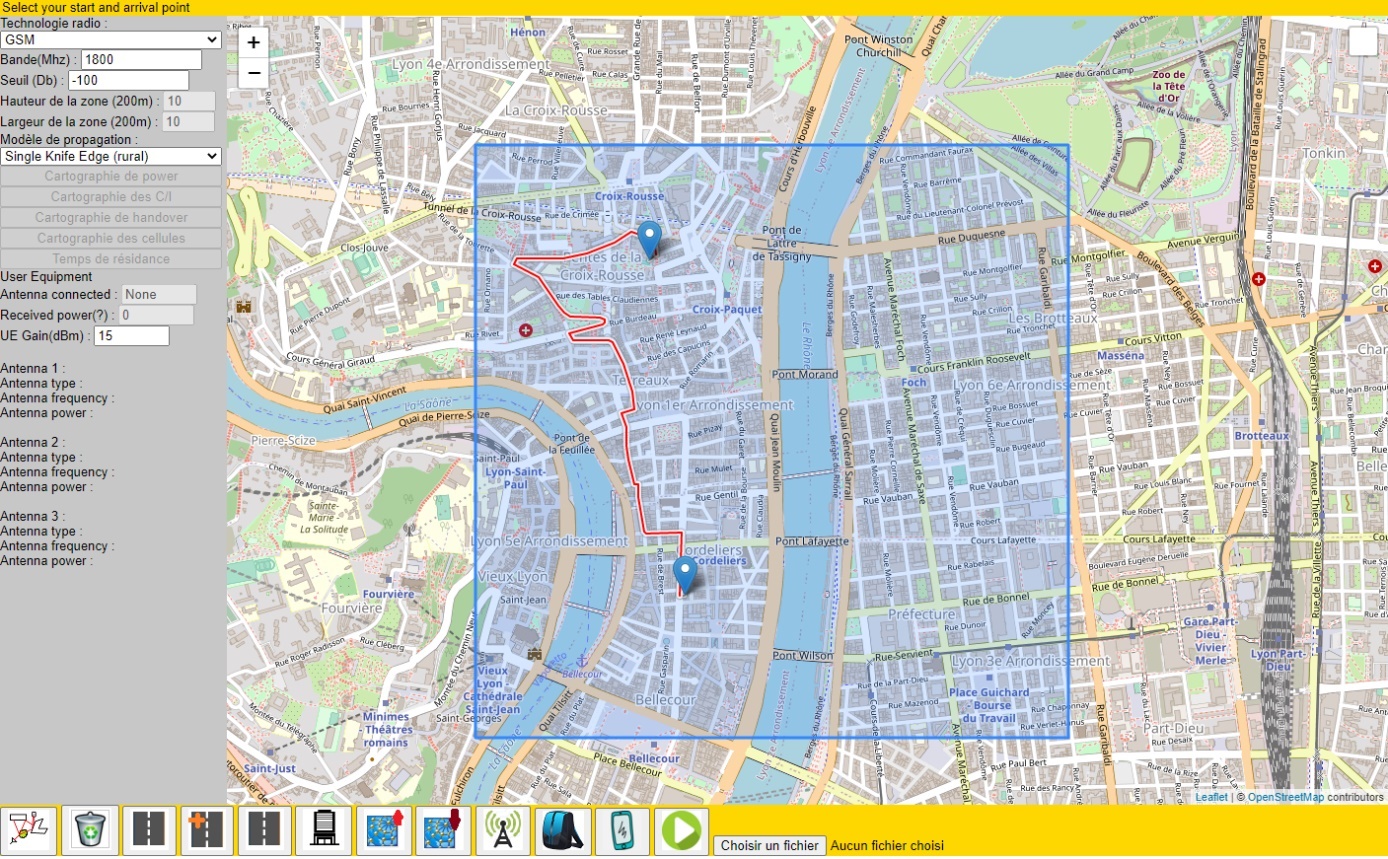
We have 3 cases :

* When the RSCPmeasure of another cell exceeds a given threshold (>-12dB for e.g.)
* When the RSCPmeasure of the attached cell falls below a given threshold (< -100db e.g.)
* When the RSCPmeasure of another cell is better than the RSCPmeasure of the attached cell by a given threshold (10db e.g.)

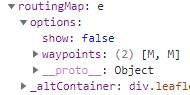
III – Implementation

1. Moving User Equipment

First, we added a button to select the User Equipment menu, once clicked the button is designed to select the departure and arrival points. Thus, we created events that collect the user click on the map. Once the user clicked the map, we collect both departure and arrival waypoint with longitude and latitude coordinates and we wanted to generate the shortest path between them thanks to a GPS features.

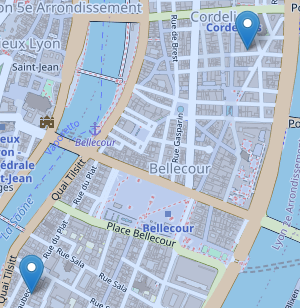
The main problem we faced at this time is that thanks to leaflet API we have a GPS that provide us the shortest path between two waypoints with various parameters like for instance the type of vehicle used, the kind of road we want etc. So, we get this kind of display:

The main problem is that there is no way to get the polyline or the coordinates generated by the *Routing* object. Indeed, to build a path, we create a *Routing.control* object containing all our presets and departure, arrival waypoints.



Once it is correctly instantiated, we call the API that generate itself a request to <https://router.project-osrm.org/route/v1/driving/> which responds inside the API the list of coordinates and metadata corresponding to the GPS path. These information’s are the ones we are targeting but the problem is that they are not accessible because they are inside the API. Right after, from this information, the API generates an HTML polyline that we also can’t access because if we inspect the HTML generated, we do not have any name or ID to collect the polyline and so move our UE along the path.

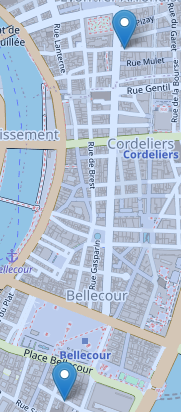
Thus, we decided to move our UE throughout every item on the map from Starting to Arrival point. That’s why we have some math back there, as far as we want a linear movement along a straight line reaching those point, we need some calculation to transform the affine function of this line to a linear movement.



With et

And *x* and *y* both respectively longitude and latitude.

We obtain the affine function connecting the two waypoints.

The problem we now face is that with a path looking for instance to a straight vertical line if we increase *x* of 1, in *y* and so in latitude axis we will increase of maybe ten which would not represent a 1 step long but a 10,05 step that’s why we need some more calculations.

We want that the distance between two points to be equal to 1 (for the example in fact it will correspond with the loop rate to the speed).

So we have

With whose whate we are looking for (corresponding to the longitude step that we must proceed to advance of Z along the affine function)

That can be transformed into

We develop

And we resolve the second degree equation, to have :

and

With , and

Thus, we obtain the longitude needed to execute a 1 step in arrival direction.

While the User Equipment is walking, we have a display on the left side bar of the connection parameters of both User Equipment and closest antennas.

1. Implementation of the selection and Reselection process

Our first selection process was based on the closest antenna, this choice has been made because it represents a first interesting step to gather all the data we need and check the correct functioning of the overall web application.

Then we started implementing the selection process. As we are working in a free-space propagation model we had to implement it first. That’s why we created this model with the associated function in the code called PropagationFreeSpace().



In this function we calculate for each cell of the designated area (the blue square drown at the beginning of the simulation) the power received by a cell from each antenna with the Friis formula (because we are working on a free-space simulation) and put it in the tab called power.

About the first cell selection process, a user equipment will start on a random cel

3) Implementation of handover

Once the inner workings of handover were understood, implementing it in MobileNetworks was not as complex as it might have seemed. I expended on the functions provided by my two co-workers, whom had already implemented UE mouvement and calculation of the power recieved by the UE.

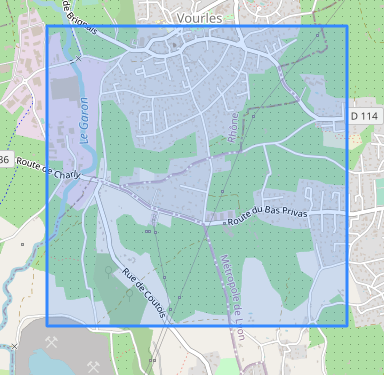
I started with a way to display the three closest antennas in the interface. For this purpose, I had to \*\*sort\*\* the list of antennas \*\*by the power recieved\*\*, without breaking the rest of the application. This was the bulk of the work, since when this sort was done, all that was left for handover was to compare the powers of the antenna the UE is connected to, to the closest non-connected antenna, and if the thresholds defined in \*\*##When to switch\*\* were met, to \*\*initiate the handover\*\*.

To make the handover visible on the application, \*\*new interface elements\*\* were added to configure the thresholds, to display the type of handover, and in the case of soft/softer handover, to show the connection to the new antenna being made \*\*before\*\* the old one is broken, as per the specifications.

IV – User Manual

In order to use properly the simulation, you have to follow these instructions

1) Define the study zone (by clicking one time on the map)



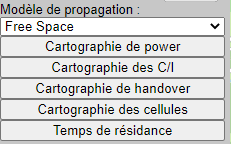
2) Define the antenna's parameters and position them on the map (by clicking)

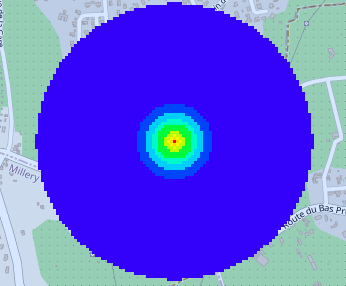




3) After positionning your antennas, activate the propagation model, then you can choose your cartography (power,cells,…)

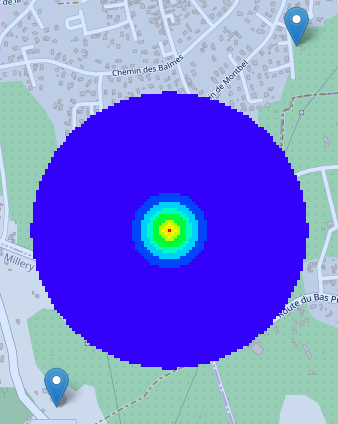






4) Create a user equipment and define the path it will follows by double-clicking on the map





5) Start the simulation by clicking on the Play button



V – Conclusion

1. Result comments
2. Personal feedbacks

This project was a good opportunity to work on UMTS, apply Friis propagation model and discover how handover works. Indeed, as we had not any knowledge about handover functioning, we had to work with a research methodology in an area which is pretty unfamiliar to me.

*Lukas :*

*Maxime :*

Sharing the work between us three enabled us to focus on a clearly defined part, without needing for us to rely on each other's work too much, as it is harder to communicate with most of us being off-site. This focus helped us work efficiently.

Aside for how the work was performed, this project helped me get a clear and profound understanding of certain parts of UMTS, and a better comprehension of mobile networks inner workings in general. Working on this project with the rest of my team was, in my opinion, an important and enjoyable part of the course.

*Thomas :*

I personally worked on researches, theoretical calculation application and global webapp architecture. The main difficulties I faced were about finding the information’s I needed in tough documentations. As we are first informaticians, the implementation part was not a real difficulty except the fact that we had to incorporate our code in an already existing and running environment.