** picture containing logo

Description automatically generated**

**Université de Franche Comté**

**Radio Networks - RI53**

**Project-CS: Cell selection**

**Report 3– Team 3**

**Done by**

MD AFTAB Uddin, Massilva Boutiab, Mandy-Ann Owusu Manu, Hovhannes Stepanyan

**Supervisor:** Mr. Mabed and Mrs. Baala

May 31, 2022

In our previous report, we worked on figuring the algorithm for the handover and for how the mobility of UE works in UMTS, fixing the interface and to also understand the code that comes along.

In this phase, we will be working on our handover algorithm along with a color function to show the handover when it is more likely to do a handover and also the mobility simulation algorithm to show how we can select, reselect and handover a mobile to a cell in a path that a user chose.

**The Algorithm**

This is the first model of the algorithm, and we are not taking into consideration soft handover and hard handover and the details of the the architecture of UMTS (ie the authentication server and so on)

Algorithm:

While each row and column of the matrix power

We check for each antenna the power received from the point

( after doing the power cartography on the interface)

And then we compare the one with the highest power is the antenna we decide to chose the one with the more power

Than we draw the colors on the cartography we do not use the colors the teacher did but we will put

0= blue no handover

Hand= a number= best place to do handover

Between 0 and hand will be the difference between the old power and the current power we have got from the antenna we will use that to define how likely it is to do a handover or now

We draw on mymap[i][j]

This is a first version of the algorithm while coding we will optimize it more

The handover approximative code /

Function mobilitySimulation() {

Int oldmobilecell=-1;

Int currentmobilecell=-1;

Int oldpower=-1000;

Int currentpower=-1000;

For( i in range(areaheight)){

For( j in range(areawidth)){

For( a in range(nbantenna)){

If(power[a][i][j]> currentpower){

Currentmobilecell=a;

Currentpower= power[a][i][j];

}

}

ColorHandover(oldpower-currentpower);

If(oldMobilecell != currentmobilecell){

Colorhandover(hand)

}

}

}

}

Simulation algorithm:

for this one we did not provide an example of code but the algorithme shows well how we are willing to do the simulation coding

Draw a path on the screen and then with get the pixel related to the path in a list

While iterating through the list of the path pixels

For first point of the path (we do a selection and) we chose the highest signal we receive and draw and arrow to the cell we belong to

If the start button is pressed then

each 10m we go through the path and compare the signal received like in handover ( we can also use a matrix of the handover result of where to do handover ) and draw arrow from where we are to the cell we are attached to

if the zone has no power( or not normal power) so idle\_mode = true also draw no arrow

if after we had no power we have again a power (then it is a reselection) then chose a cell and draw arrow

wait 1s and then go to second path

if pause button is pressed then

we can select whatever we are on call or transferring data, or we are not in communication or we switch off the phone or turn on the phone

if we are on call we do a handover draw arrow

else if we are transferring data or we are not in communication then we do reselection draw -------arrow

else if we are switched on we do a selection chose a cell and draw arrow

else we are off

we save the situation for than next point on the path

Conclusion: Next week we will try implement this algorithm into code.