
RAPPORT PROJET DE FIN D'ÉTUDES :

Etude de performances des réseaux LTE

Rédigé par :

AYOUB ESSABER TAIHI

RACIM HAFFAF

Encadré par :

M. MARWEN ABDENNEBI

Table des matières

I	Introduction	1
II	Évolution des générations des réseaux mobiles	2
1	2G	2
2	2.5G et 2.75G	2
3	3G	3
III	Étude de la LTE	4
1	Nœuds, architectures et technologies	4
2	Caractéristiques de la LTE et LTE-A	5
IV	Simulation LTE	7
1	Outils logiciels	7
2	Installations	9
3	Simulation LTE-A - Réseau complet	11
4	Résultats et étude de performances	12
V	Conclusion	15
	Annexes	16

I. Introduction

Les réseaux de communications mobiles sont devenus en quelques décennies une composante majeure du développement des technologies de l'information au niveau mondial. En particulier, les réseaux cellulaires. Ces derniers ont complètement transformé la notion de connectivité ainsi que nos modes de communications. Les réseaux mobiles et sans fil ont connu un essor sans précédent ces dernières années. Il s'agit d'une part du déploiement de plusieurs générations successives de réseaux de télécommunications essentiellement dédiés à la téléphonie (2G), puis plus orientés vers le multimédia (3G), la 4G de réseaux sans fil apporte un véritable tournant dans le foisonnement et la disparité des solutions existantes. L'objectif cette fois sera certes d'augmenter les débits et les applications prises en charge par ces réseaux mais encore de construire un cadre permettant leur interopérabilité. Des premières solutions sont disponibles auprès de la plupart des opérateurs de télécommunications mais la plupart d'entre elles sont des solutions spécifiques à un type d'interconnexion et à un opérateur.

Par ailleurs, il existe deux grandes catégories de technologie de réseaux mobiles 4G ; La première s'appelle LTE (Long-Term Evolution) et la deuxième est la LTE-A (Long-Term Evolution), la version "évoluée" de la première. La norme LTE, définie par le consortium 3GPP (3rd Generation Partnership Project), a d'abord été considérée comme une norme de troisième génération « 3.9G » (car proche de la 4G), spécifiée dans le cadre des technologies IMT-2000, car dans les « versions (ou rel pour releases en anglais) 8 et 9 » de la norme, elle ne satisfaisait pas toutes les spécifications techniques imposées pour les normes 4G par l'Union Internationale des Télécommunications (UIT). La norme LTE n'est pas figée. Par conséquent, le consortium 3GPP l'a fait évoluer en permanence (en général, une nouvelle version tous les 12 à 18 mois).

L'évolution technique de la norme 4G, apparue sur le marché en 2014, a permis la création de la technologie LTE Advanced, qui est aussi appelée parfois la « vraie 4G », considérée comme une norme de quatrième génération. Elle intègre une technique de multiplexage appelée MIMO, en standard 2x2, puis 4x4 et expérimentalement avec 8x8 niveaux ; c'est le cœur de la 4G / LTE Advanced : l'adoption du MIMO 2x2 intégré dans chaque catégorie de terminal valable à partir de la "rel 6" LTE, le plus souvent en 4x4 et jusqu'à 8x8 pour la 4G+. La normalisation de la première version (version ou rel 10) utilise des fréquences identiques et les codages radio (OFDMA et SC-FDMA) déjà utilisés dans les réseaux LTE (réseau radio EUTRAN). Le LTE-Advanced est capable de fournir des débits pics descendants très élevés atteignant 1,2 Gb/s à l'arrêt et à plus de 100 Mb/s pour un terminal en mouvement, grâce aux technologies réseaux intelligentes qui permettent de maintenir des débits plus élevés en tout point de la cellule radio, alors qu'ils baissent fortement en bordure des cellules 3G et 3.9G, de la LTE.

Dans notre projet, l'objectif principal est de mesurer les performances qu'un réseau LTE peut apporter en se basant sur les aspects techniques et théoriques fournis dans le paragraphe précédent, après l'avoir simulé grâce à un outil logiciel à événements discrets. Cet outil s'appelle le NS3. Après avoir effectué l'étude bibliographique d'un réseau LTE, on doit d'abord définir une architecture pertinente. Ensuite, on doit mettre en œuvre un scénario de mobilité pour les terminaux pour tester le fonctionnement d'un processus appelé le handover. Enfin, on effectue les mesures de performances de handover pour vérifier à quel point les réseaux LTE sont puissants et qu'ils répondent parfaitement aux spécifications des standards ainsi qu'aux besoins des usagers.

II. Évolution des générations des réseaux mobiles

1. 2G

La deuxième génération des téléphones mobiles (2G) repose sur une technologie numérique. Elle a été développée à la fin des années 1980. Ces systèmes cellulaires utilisent une technologie numérique pour la liaison (ou la signalisation), ainsi que pour le signal vocal, ce qui constitue le principal changement par rapport à la 1G. De plus, ce système apporté, par rapport à ses prédécesseurs, les changements suivants :

- Une meilleure qualité ainsi qu'une plus grande capacité à moindre coût pour l'utilisateur.
- Conversations téléphoniques cryptées numériquement, au moins entre le téléphone mobile et la station de base cellulaire (mais pas nécessairement dans le reste du réseau).
- Plus de services de données pour mobile (tels que les SMS, messages texte et photo).
- Meilleure gestion de la localisation de l'utilisateur ainsi que du handover (changement d'une cellule à une autre pour l'utilisateur en pleine session téléphonique d'appel).

2. 2.5G et 2.75G

Le General Packet Radio Service ou GPRS est une norme pour la téléphonie mobile dérivée du GSM et complémentaire de celui-ci. Le GPRS, à la différence de la GSM, ajoute la transmission par paquets IP. Cette méthode est plus adaptée à la transmission des données (en plus de la voix, on peut transmettre l'image et un document). En effet, les ressources ne sont allouées que lorsque des données sont échangées (en GSM, un circuit est établi et les ressources sont associées en même temps). Grâce à ce concept, le GPRS a permis un débit de transmission beaucoup plus élevé (171 Kbps contre 9.6 Kbps).

Par ailleurs, le GPRS comporte, en plus des nœuds qui sont présents dans la GSM, les éléments suivants :

1. SGSN

- Interface avec le réseau de paquets externe (réseau IP) via une autre passerelle, le GGSN (Gateway GPRS Support Node) pour les appels sortants, et vers la station mobile pour les appels entrants.
- Maintien des informations identifiant l'abonné, les services utilisés et contrôle de la localisation du mobile.

2. GGSN

- Le GGSN (Gateway GPRS Support Node) est un nœud d'interconnexion entre le réseau paquet mobile (GPRS ou UMTS) et les réseaux IP externes.

Ensuite, une autre évolution du GPRS est venue, appelée EDGE acronyme anglais signifiant aussi « tranchant (d'une lame) ». Elle a comme principale nouveauté la modification du type de modulation. Tout comme la norme GPRS, le standard EDGE est utilisé comme transition vers la troisième génération (3G) (on parle ainsi de 2.75G). EDGE utilise une modulation différente de la modulation utilisée par GSM (EDGE utilise la modulation 8-PSK), appelée modulation adaptative (modulation adaptée en fonction des conditions radio), ce qui permet d'atteindre des débits allant jusqu'à 384 kbit/s pour les stations fixes (piétons et véhicules lents) et jusqu'à 144 kbit/s pour les stations mobiles (véhicules rapides).

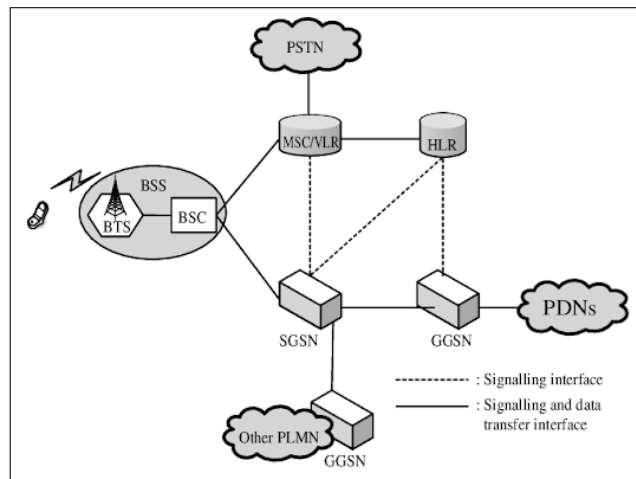


FIGURE 1 – Architecture d'un réseau mobile GSM (2G).

3. 3G

La 3G est représentée principalement par les normes Universal Mobile Telecommunications System (UMTS) et CDMA2000, permettant des débits (de 2 à 42 Mb/s définis par la dernière génération des réseaux UMTS : l'HSPA+ DC) qui sont bien plus rapides que les générations précédentes. Ce système permet des services de communications plus rapides notamment pour la télécopie, l'Internet et l'intégration de la voix (qui n'existait pas en EDGE), de n'importe quel endroit et à tout moment. L'UIT IMT-2000 est la norme internationale de la 3G. Elle a ouvert la voie à de nouvelles applications et services comme par exemple le divertissement multimedia, la visiophonie, la localisation des services.

Outre l'augmentation de débit, un point complexe à résoudre était de passer d'un service de téléphonie (connexion circuit) vers un service DATA (connexion paquets), passer vers une technologie basée sur le W-CDMA (technique de codage qui utilise un accès multiple par division de code). Du fait de cette contrainte, il fallait ajouter d'autres étages aux antennes de l'ancienne BTS de la GSM (ajout d'amplificateurs de puissance, de DSP, de batteries, de climatiseurs Back-up...). L'ancienne BSC a également subi des changements technologiques. De même pour les stations mobiles (exigences en termes de batterie).

Les nouveaux nœuds qui les ont remplacés sont : UE (User Equipment), Node B et RNC (Radio Network Controller).

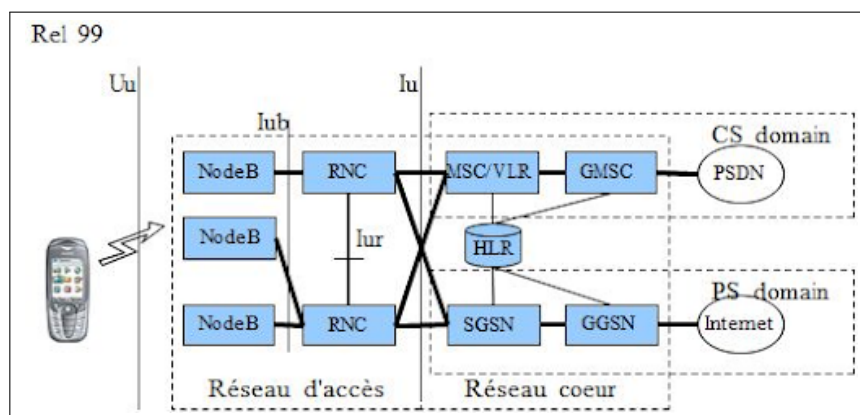


FIGURE 2 – Architecture d'un réseau mobile UMTS (3G).

III. Étude de la LTE

Le réseau LTE a connu une grande révolution comparant à ses prédécesseurs de UMTS (3G), GPRS/EDGE (2.5G), GSM (2G) et 1G, en termes d'architecture, de technologie et de vitesse, après avoir été développé par le 3GPP (3rd Generation Partnership Project), qui a contribué à son développement.

En effet, les changements offerts en LTE ont été sans précédent ; débit de transmission allant jusqu'à 300 Mégabits par seconde (en UMTS, on était limité à 2 Mégabits par seconde, soit une centaine de fois de moins que la LTE), meilleure gestion de mobilité des appareils mobiles, interruptions des services (transfert d'appels, de voix, d'images...) quasi inexistantes...etc. Pour parvenir à ses fins, le principe adopté par la LTE est de mettre toute son infrastructure basée sur une communication sans fil, grâce à l'IP (en UMTS, les services temps réels dédiés aux conversations téléphoniques, telles que la vidéo-téléphonie, les jeux vidéo, les applications multimédia fonctionnaient en mode circuit filaire). De plus, l'interface X2 d'interconnexion entre les stations de bases permet l'échange de données du plan utilisateur et informations de contrôle pour de meilleures performances, notamment lors du handover. D'autre part, de nouveaux équipements sont arrivés avec cette technologie et une architecture beaucoup plus simplifiée, le tout en étant interopérable et compatible avec les technologies des générations précédentes.

Par ailleurs, la 4ème génération donne des avantages suivants :

- Garanti de la continuité de la session en cours.
- Réduction des délais d'attente.
- Fournissement d'une meilleure qualité de service.
- Optimisation de l'utilisation des ressources.
- Réduction des pertes de paquets.
- Minimisation du coût de signalisation.

En outre, d'un point de vue technologique, la LTE dispose de l'architecture et des nœuds suivants :

1. Nœuds, architectures et technologies

Les réseaux LTE sont des réseaux cellulaires constitués de milliers de cellules radio qui utilisent les mêmes fréquences hertziennes, grâce aux codages radio OFDMA et SC-FDMA. De leurs côtés, les nœuds Node B, ne sont plus appelés ainsi, mais plutôt des ENode B (Evolved Node B), notamment du fait qu'il n'y a plus d'élément de contrôleur séparé RNC (RNC étant supprimé). L'ENode B s'occupe lui-même d'effectuer les tâches du RNC de l'UMTS (contrôle des transmissions radio, allocation des ressources radio, chiffrement des données, gestion du handover...) et est donc plus intelligent que le NodeB. De plus, du fait de la généralisation de la technologie IP à travers tout le réseau LTE, les nœuds GGSN-SGSN ne sont plus existants. Ils ont été remplacés par d'autres nœuds appelés MME (Mobility Management Entity), Serving Gateway et Packet Gateway. La figure 3 présente l'architecture du réseau LTE. Typiquement, une architecture LTE est subdivisée en trois sous-systèmes :

1. Réseau EPC

Le cœur de réseau appelé "EPC" (Evolved Packet Core) utilise des technologies "full IP" (ou "tout

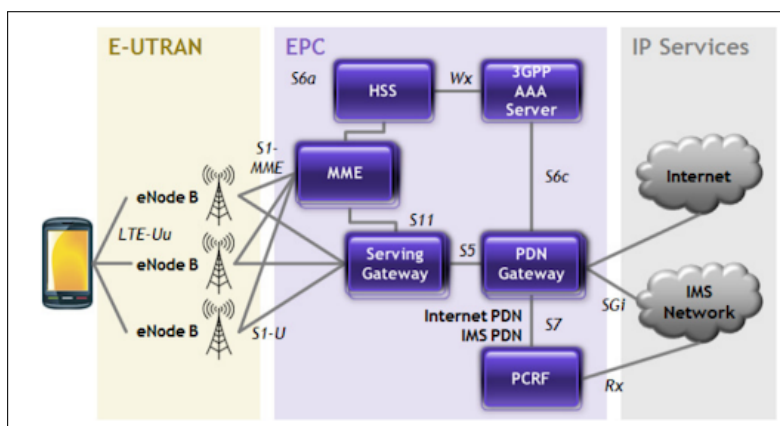


FIGURE 3 – Architecture d'un réseau mobile 4G.

IP”), c’est-à-dire basées sur les protocoles Internet qui permettent des temps de latence réduits, le transport de la voix et le transport des données. Ce cœur de réseau assure les fonctions suivantes :

- Interconnexion via des routeurs avec les autres ENode B distants, les réseaux des autres opérateurs mobiles, les réseaux de téléphonie fixe et le réseau Internet.
- Simplification du réseau d’architecture “tout IP”.
- Garanti de la mobilité et de la fonction du handover entre les réseaux fonctionnant sous la technologie LTE, tout comme ceux d’autres technologies sans fil, à savoir WIMAX et CDMA2000.

2. Réseau E-UTRAN

La partie radio du réseau, appelée “E-UTRAN” est simplifiée par rapport à celles des réseaux 2G (BSS) et 3G (UTRAN) par l’intégration dans les stations de base “ENode B” avec des liaisons en fibres optiques et des liens IP reliant les eNodeB entre eux (liens X2). Ainsi que des fonctions de contrôle qui étaient auparavant implémentées dans les RNC (Radio Network Controller) des réseaux 3G UMTS. Cette partie est responsable sur le management des ressources radio, la porteuse, la compression, la sécurité, et la connectivité vers le réseau cœur évolué.

3. Réseau IMS

L’IMS (IP Multimedia Sub-system) est une architecture standardisée NGN (Next Generation Network) pour les opérateurs de téléphonie. Elle permet de fournir des services multimédias fixes et mobiles. Cette architecture n’a pas seulement comme objectif de permettre de nouveaux services, existants ou futurs, proposés sur Internet, les utilisateurs doivent aussi être capables d’utiliser ces services aussi bien en déplacement (situation de roaming) que depuis chez eux. Pour cela, l’IMS utilise les protocoles standards IP. Ainsi, une session multimédia, qui s’effectue entre deux utilisateurs IMS, entre un utilisateur IMS et un internaute, ou bien encore entre deux internautes, est établie en utilisant exactement le même protocole. De plus, les interfaces de développement de services sont également basées sur les protocoles IP. C’est pour cela qu’IMS fait véritablement converger l’Internet et le monde de la téléphonie cellulaire ; Il utilise les technologies cellulaires pour fournir un accès en tout lieu, et les technologies Internet pour fournir les services.

2. Caractéristiques de la LTE et LTE-A

Un réseau LTE dispose des caractéristiques suivantes :

1. Débit et bande passante

L'évolution des débits suit une progression semblable à celle de la capacité de calcul. Chaque nouvelle technologie de réseaux mobiles augmente les débits et suscite une attente de débits supérieurs. Les objectifs de débit maximal définis pour le LTE sont les suivants :

- 300 Mbit/s en voie descendante (liaison ENode B - UE) pour une largeur de bande allouée de 20 MHz.
- 75 Mbit/s en voie montante (liaison UE - ENode B) pour une largeur de bande allouée de 20 MHz.

2. Délai d'attente

L'objectif fixé pour le LTE est d'améliorer la latence par rapport à l'UMTS, via un temps de transition inférieur à 10 ms.

3. Technique d'accès OFDMA

L'utilisation du codage OFDMA est une technologie de codage radio de type Accès multiple par répartition en fréquence (AMRF ou en anglais FDMA) pour la liaison descendante et du SC-FDMA (Le single-carrier FDMA est une technologie de codage radio de type accès multiple par répartition en fréquence pour la liaison montante au lieu du W-CDMA en UMTS). L'OFDMA et sa variante SC-FDMA sont dérivés du codage OFDM, mais contrairement à l'OFDM, l'OFDMA est optimisé pour l'accès multiple, c'est-à-dire le partage simultané de la ressource spectrale (bande de fréquence) entre plusieurs utilisateurs distants les uns des autres (figure 4). L'OFDMA est compatible avec la technique des antennes MIMO.

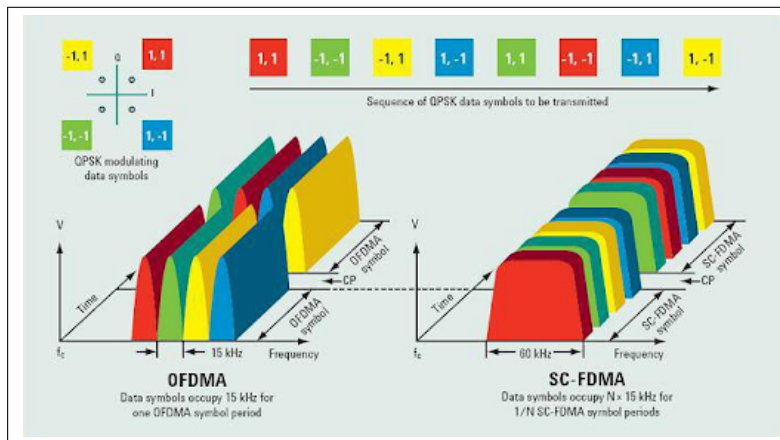


FIGURE 4 – Différence entre la modulation SC-FDMA et la modulation OFDMA.

4. MIMO

MIMO (Multiple Input, Multiple Output) désigne une technologie d'antenne destinée aux communications sans fil, qui consiste à utiliser plusieurs antennes tant au niveau de la source (émetteur) qu'à celui de la destination (récepteur). Les antennes présentes à chaque extrémité du circuit de communication sont combinées pour minimiser les erreurs et optimiser le débit des données. L'implémentation de la MIMO 8x8, dans la technologie LTE-A, a pu augmenter 8 fois le débit théorique, et atteindre jusqu'à 1 Gbps.

5. Gestion de la mobilité et le handover

Dans des conditions où l'utilisateur se déplace à grande vitesse, le handover pourra s'effectuer dans des conditions où l'utilisateur se déplace à une vitesse qui peut même aller jusqu'à 350 km/h et ce sans interruption du service. L'interface X2 joue un rôle clé dans l'opération de handover. L'ENode B source utilisera l'interface X2 pour envoyer le message Handover Request au cible ENode B. Si l'interface X2 n'existe pas entre

les deux ENode B dans question, alors des procédures doivent être lancées pour en établir une avant que le transfert puisse être atteint.

Toutes les étapes du handover sont décrites dans la figure suivante :

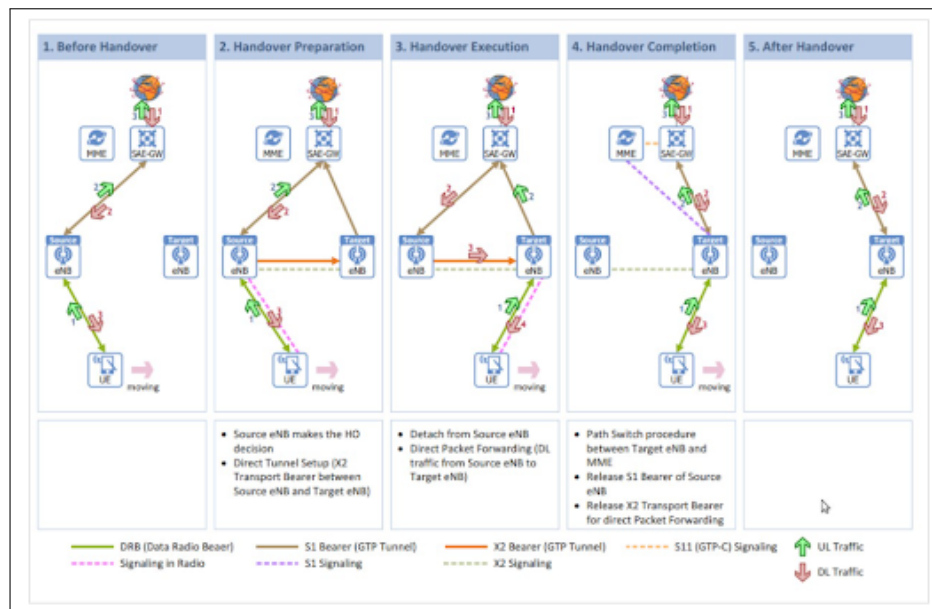


FIGURE 5 – Schéma décrivant le handover dans la 4G, avec l'intervention de l'interface X2.

IV. Simulation LTE

La simulation est un outil d'aide à la décision très utilisé par les concepteurs et les gestionnaires des systèmes complexes. Elle consiste à construire un modèle d'un système réel (physique, économique, humain ... etc.) et à conduire des expériences sur ce modèle afin de bien comprendre le comportement de ce système et d'en améliorer les performances.

Le but principal de cette étude est de montrer l'efficacité du mécanisme de handover pour du trafic en temps-réel dans le cas où les stations mobiles (eNode B) font un déplacement.

Nous présenterons dans ce qui suit le déroulement des étapes de simulation que nous avons mené dans ce travail (études, installation, prise en main...), qui consiste à simuler un réseau 4G. Ensuite, nous étudierons les performances du handover et nous analyserons les résultats obtenus par des graphes.

1. Outils logiciels

1. NS3

NS3 (Network Simulator 3) est un logiciel libre de simulation à événements discrets très utilisé dans la recherche académique et dans l'industrie. La version 3.35 (qu'on a utilisé) est entièrement écrite en C++ avec des bindings en Python ce qui permet d'écrire des scripts dans les deux langages C++ et Python.

Son développement a d'abord commencé en juillet 2006, et devait durer quatre ans, il est financé par les instituts comme l'université de Washington, Georgia Institute of Technology et le Centre de l'ICSI pour la recherche sur Internet, la première version majeure publique et stable a été publiée en juin 2008.

2. Environnement de simulation

NS-3 utilise des termes largement employés dans le domaine des réseaux, mais qui peuvent avoir une signification particulière au sein du simulateur. Voici les principaux :

- Un nœud "Node" : Représente tout élément de réseau. La composition d'un nœud peut être gérée (ajout de composants, d'applications et de protocoles).
- Une application "Application" : Représente un code exécuté par un utilisateur. Ce code peut être nécessaire au déroulement d'une simulation. L'échange de paquets durant une simulation nécessite par exemple la description d'une application au sein des nœuds participants. Les applications peuvent être attachées à un Node.
- Un canal de communication "Channel" : Channel est le lien qui relie les NetDevices installés dans les nœuds. Des spécialisations de cette classe sont définies, comme par exemple LTEChannel pour modéliser un réseau LTE.
- Une interface de communication : Appelée NetDevice, qui modélise à la fois les équipements et les pilotes de communication. Des spécialisations sont fournies comme par exemple LTENetDevice qui peut être reliée à un LTEChannel.

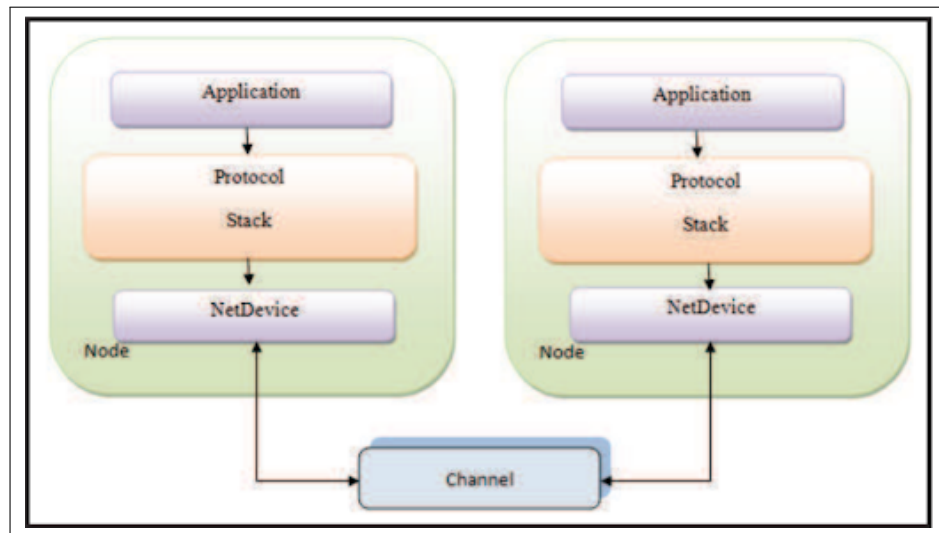


FIGURE 6 – Architecture logicielle du NS3.

Par ailleurs, nous avons utilisé une machine virtuelle (VMware Player pro 16), un émulateur qui permet de virtualiser notre réseau 4G. Pour simuler notre réseau, nous avons installé sur la machine virtuelle les outils environnants suivants :

- Système Linux (Ubuntu 21.04) : création et la configuration des accès au réseau 4G et prise en charge de ses connexions, ses liens ainsi que ses protocoles.
- Éditeur de texte (Visual Studio Code) : Prise en charge du débogage, de la mise en évidence de la syntaxe et de la complétion intelligente du code source C++.
- Packages requis pour NS3 (C++, Python, NetAnim) : Fonctionnement des applications par des bibliothèques, des distributions et des composantes logicielles qui permettent de les faire tourner.

3. NetAnim

NetAnim est l'interface qui affiche graphiquement le scénario d'une simulation. Il effectue une animation à l'aide d'un fichier de trace XML. Le trace XML stocke et décrit les résultats des événements au cours de la simulation, qui seront lus par NetAnim. Enfin, NetAnim se charge de traduire ce trace XML en une animation décrivant le scénario de notre réseau 4G.

2. Installations

1. Installation NS3

NS3 est un logiciel open-source, disponible en licence libre. On peut le télécharger gratuitement à partir du lien suivant : <https://www.nsnam.org/releases/ns-3-35/download/>, puis après avoir cliqué sur "ns-allinone-3.35". Une fois téléchargé, on obtient un dossier ns-allinone-3.35 compressé. On le décompresse. Ensuite, on réalise les étapes suivantes :

i. Dans le répertoire : ns-allinone-3.35

- **En ligne de commande, écrire :** `./waf configure --build-profile=optimized --enable-examples --enable-tests`

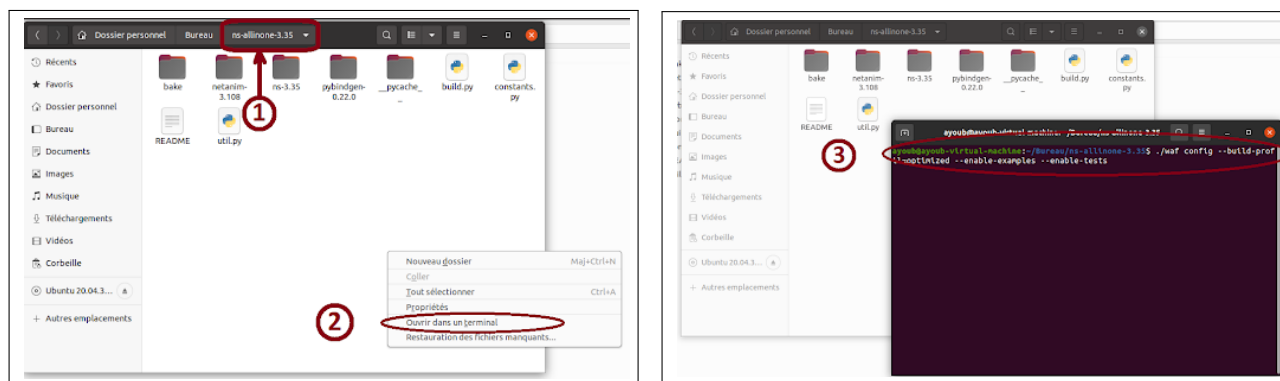


FIGURE 7 – Configuration du WAF (outil du build basé sur Python).

- **En ligne de commande, écrire :** `./waf configure --build-profile=debug --enable-examples --enable-tests`

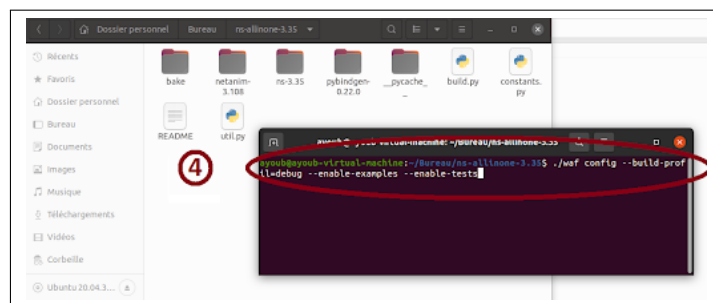


FIGURE 8 – Exécution du WAF et établissement des tests.

ii. Dans le répertoire : ns-3.35 (sous-répertoire de ns-allinone-3.35)

- **Écrire en ligne de commande :** `./waf`

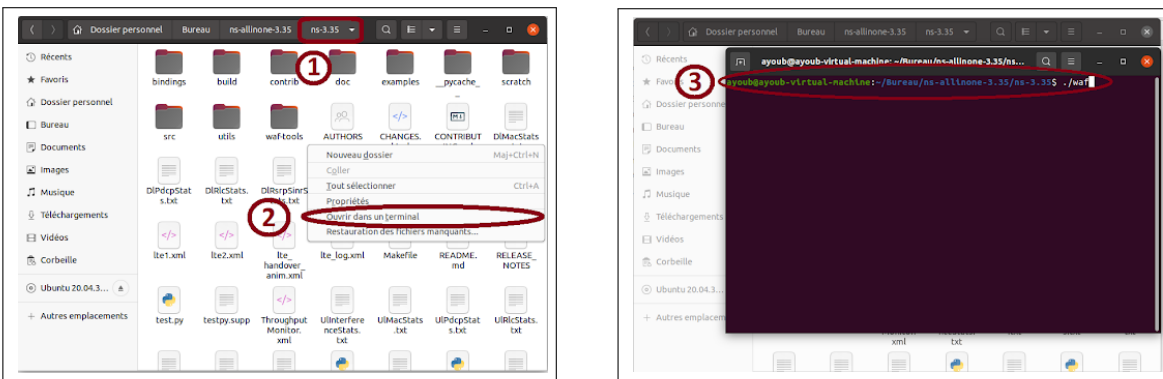


FIGURE 9 – “Build” du WAF.

- Écrire en ligne de commande : `./test.py`

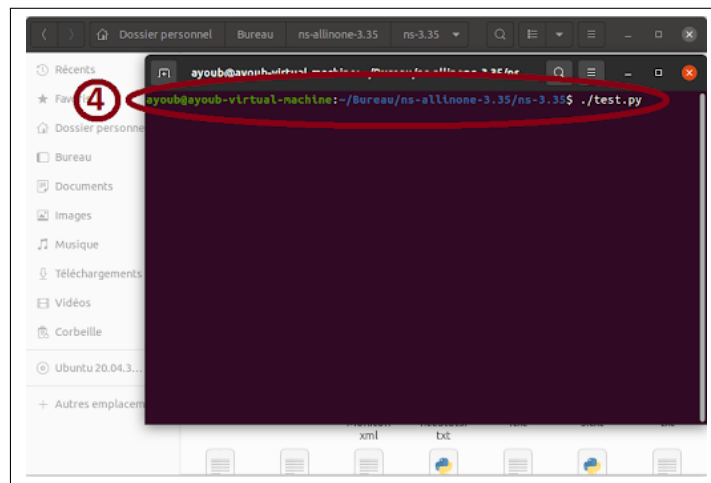


FIGURE 10 – Écriture et lancement de tests Python.

2. Installation NetAnim

On réalise les étapes suivantes :

- Entrer dans le répertoire : `/ns-allinone-3.35/netanim-3.108` (sous-répertoire de `ns-allinone-3.35`)
- Écrire en ligne de commande : `make clean`

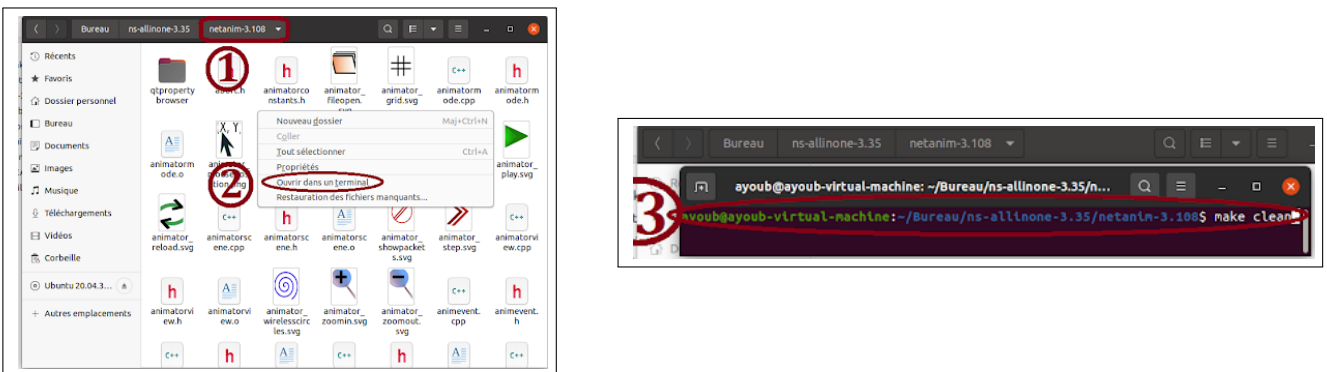


FIGURE 11 – Écriture de `make clean` dans la ligne de commande.

- Créer NetAnim pour exécuter l'animation de notre réseau : qmake NetAnim.pro

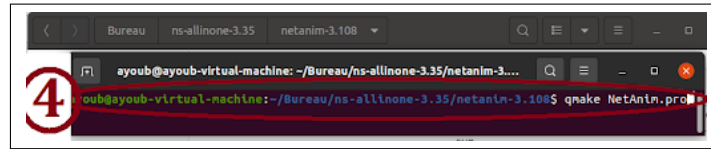


FIGURE 12 – Écriture de qmake NetAnim.pro dans la ligne de commande.

- Automatiser la compilation de notre projet par : make

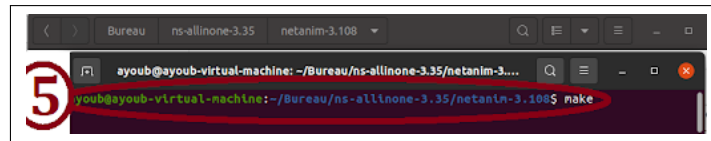


FIGURE 13 – Écriture de make dans la ligne de commande.

- Exécuter NetAnim par : ./NetAnim

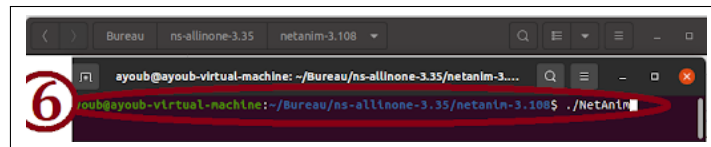


FIGURE 14 – Écriture de ./NetAnim dans la ligne de commande.

- Au final, la structure logicielle pour la virtualisation de notre réseau LTE est décrite dans la figure suivante :

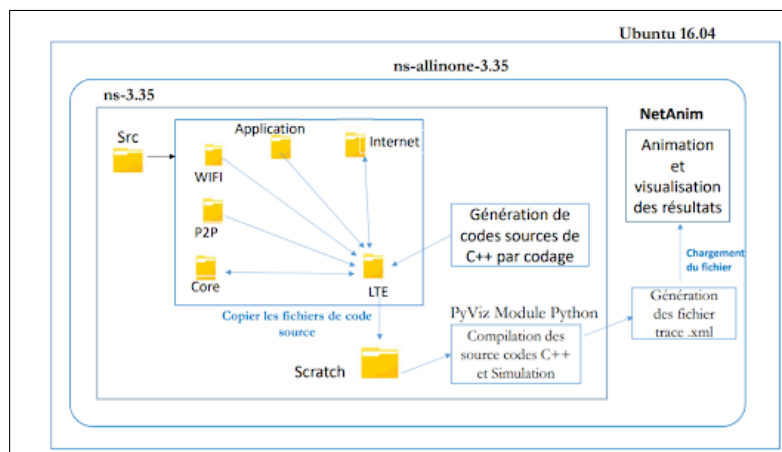


FIGURE 15 – Structure logicielle pour la virtualisation d'un réseau 4G.

3. Simulation LTE-A - Réseau complet

Place maintenant à la mise en œuvre du réseau LTE. Le scénario qu'on a implémenté pour notre projet consiste en deux cellules (ou plus précisément 2 ENodes B), ainsi que 3 UEs pour chaque cellule. Chaque triplé de cellule se déplace en contre sens de l'autre. Ainsi, chaque ENode B réalise un handover pour que les deux triplets ne perdent pas la communication en pleine session.

Pour mener à bien notre réalisation, on se dirige dans le répertoire scratch (qui contient les codes

source C++ et simulation, figure 15). Puis on écrit le code source, en langage C++, qui permet de virtualiser notre réseau LTE (cf annexe). La compilation de ce code source permet de générer un fichier trace .xml. Le fichier trace sera ouvert par NetAnim. Enfin, le chargement de ce fichier dans NetAnim permet d'animer graphiquement la structure LTE qu'on a créé ainsi que le scénario de mobilité qu'on a choisi. Les étapes de compilation puis l'animation de notre scénario sont résumées dans les points suivants :

- **Entrer dans le répertoire : /ns-allinone-3.35/ns-3.35/scratch (sous-répertoire de ns-3.35).**
- **Créer le fichier .cc (extension C++) de simulation de notre réseau LTE.**
- **Ouvrir le terminal.**
- **Écrire dans le terminal, pour la compilation du code source, la commande suivante : ./waf --runnom_fichier_cc**
- **Entrer dans le répertoire : /ns-allinone-3.35/netanim-3.108/ (sous-répertoire de ns-allinone-3.35).**
- **Ouvrir le terminal.**
- **Exécuter NetAnim par : ./NetAnim**
- **Cliquer sur la touche d'ouverture du fichier trace .xml.**
- **Ouvrir le fichier trace .xml correspondant à notre simulation.**

4. Résultats et étude de performances

Au final, on aura une structure du réseau LTE, avec différentes caractéristiques des nœuds suivants :

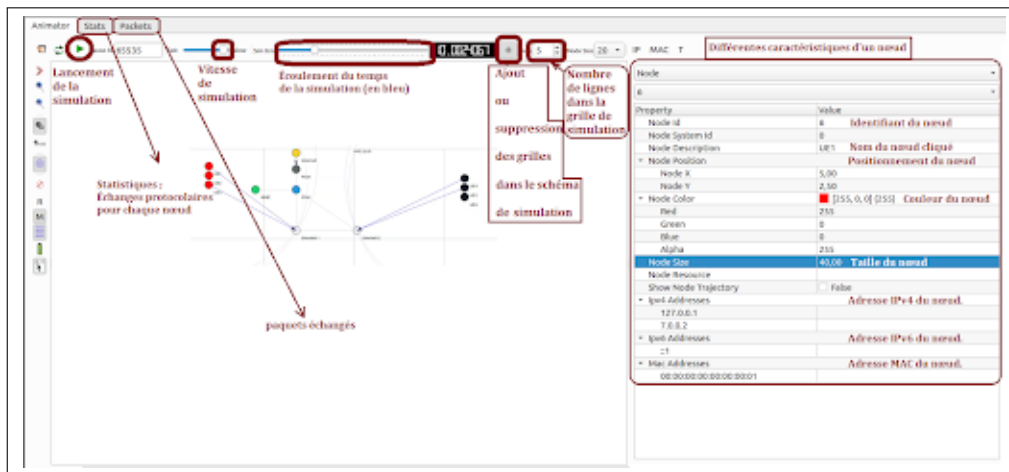


FIGURE 16 – Simulation de notre réseau LTE, avec les différents outils du logiciel NetAnim.

Lorsqu'on fait tourner la simulation, on aura bel et bien le handover qui s'exécute, où la session reste active lors du changement de chaque triplet d'UE d'une cellule à une autre. Comme le démontre la figure suivante :

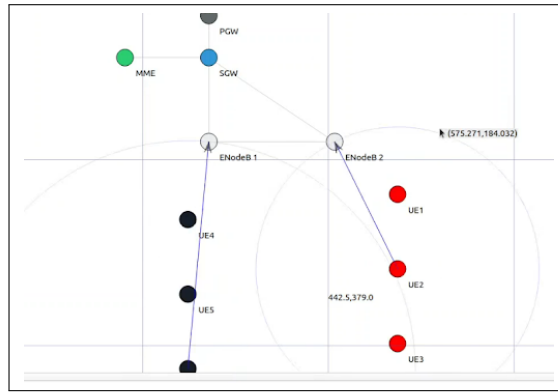


FIGURE 17 – Simulation montrant l'exécution du handover pour chaque triplet d'UEs.

En ce qui concerne les performances, on les déduit à partir des fichiers textes générés après exécution du code source et déroulement de la simulation dans NetAnim. Ces fichiers stockent différentes informations correspondant au déroulement de la simulation (couches MAC, RLC, physiques...). Parmi ces fichiers, on trouve *DlPdcStats.txt*. Il contient des champs clés où sont stockées les informations suivantes : Début et fin d'envoi des paquets, numéro de cellule courante, nombre de paquets émis et reçus, temps de latence... etc, et *SNRVSThroughput.plt* qui contient deux champs, le rapport signal sur bruit et le débit de transmission.

A partir de Matlab, on lit ces fichiers et on extrait les différents champs qui nous seront utiles pour la mesure des performances.

1. Débit de transmission

On écrit un code Matlab qui permet de lire le fichier *SNRVSThroughput.plt* puis extraire ses deux champs. Enfin, on dessine la courbe Débit en fonction du rapport signal sur bruit, et on aura les résultats suivants :

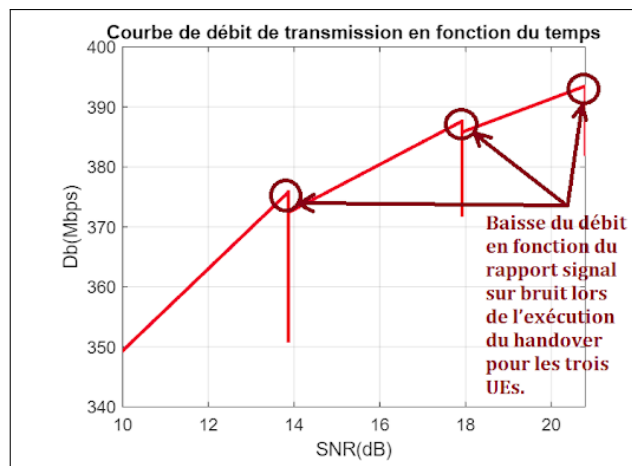


FIGURE 18 – Courbe de débit de transmission en fonction du rapport signal sur bruit de notre réseau LTE.

Dans cette courbe, on déduit que le débit moyen est très élevé (il dépasse les 390 Mbps). D'autre part, on note la dégradation de ce dernier en trois lieux (figure 18). Ces dégradations sont dues à l'exécution du handover pour les trois UEs. Tout de même, ces dégradations ne s'avèrent pas trop contraignantes ni ressenties. Résultat, notre réseau LTE est très performant en termes de débit de transmission.

2. Temps de latence

On écrit un code Matlab qui permet de lire le fichier DIPdcpStats.txt puis extraire ses champs de temps d'envoi et delay (délai d'attente). Enfin, on dessine la courbe Temps d'envoi en fonction du délai d'attente, et on aura les résultats suivants :

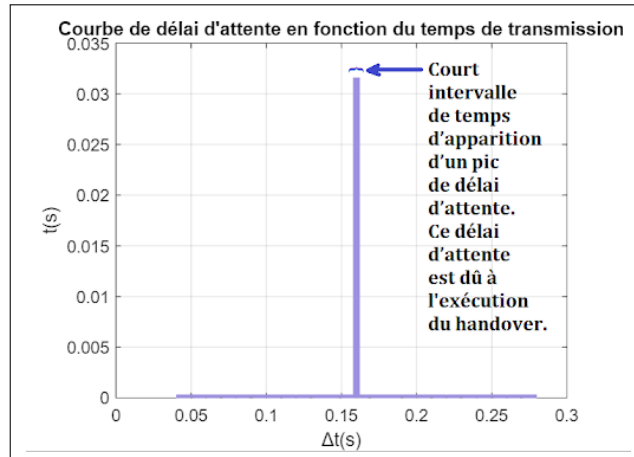


FIGURE 19 – Courbe de délai de transmission de notre réseau LTE.

On déduit de la courbe que le délai d'attente est très faible. On observe lors d'un court intervalle de temps l'apparition d'un pic (figure 19). Ce pic survient en raison de l'exécution des handovers. Néanmoins, il n'est qu'à l'ordre de 32 petites millisecondes, et est très court. Par conséquent, cela aura des répercussions négligeables sur les performances. Partout ailleurs, on remarque l'absence de délai d'attente. Ces résultats-là montrent que notre réseau LTE est très performant, ce qui est très utile notamment pour des applications exigeantes en termes de délai d'attente (applications de streaming, de visio-conférence...).

3. Perte de paquets

On écrit un code Matlab qui permet de lire le fichier DIRlcStats.txt puis extraire ses champs de temps d'envoi et RxBytes (nombre de paquets reçus) et TxBytes (nombre de paquets transmis). Ensuite, on fait la soustraction entre RxBytes et TxBytes, qui désigne le nombre de paquets perdus à chaque instant d'envoi. Enfin, on dessine la courbe Paquets perdus en fonction du temps, et on aura les résultats suivants :

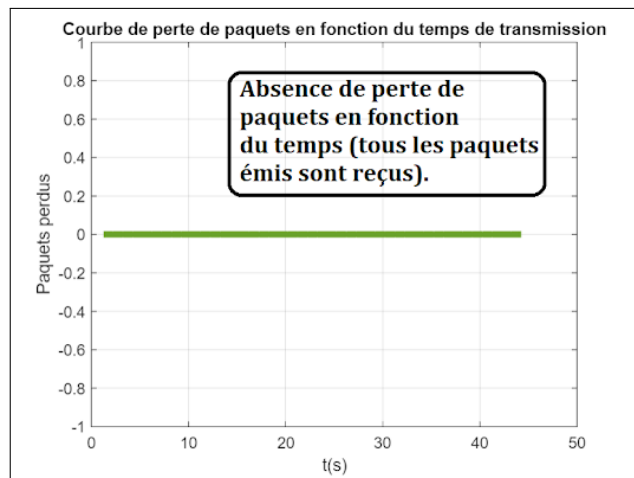


FIGURE 20 – Courbe correspondant au paquets perdus en fonction du temps de transmission de notre réseau LTE.

On déduit que tout au long de la simulation, il n'existe pas de perte de paquets ; tous les paquets émis

de tous les nœuds émetteurs sont complètement reçus par les nœuds récepteurs destinés, que ce soit lors des instants des handovers ou en dehors de ces derniers. De ce fait, notre réseau 4G est ultra performant.

V. Conclusion

Au cours de notre réalisation, on a pu constater que les générations de téléphonie mobile ont été développées pour garantir de meilleure performance, en particulier les Qualités de Services (QoS), tout en apportant de nouvelles applications, de nouveaux services, ainsi qu'une meilleure gestion de mobilité. De nos jours, la génération la plus déployée est la 4G. Elle est connue comme la meilleure génération existante permettant le très haut débit ainsi qu'une interopérabilité avec les autres réseaux d'anciennes générations (2G, 3G).

Le réseau 4G est fondé sur une technologie ultra résistante aux coupures causées par le déplacement d'un terminal mobile entre des cellules couvertes. Cela se passe de façon qu'un utilisateur en communication ne perde pas sa connexion en cours, cette technologie s'appelle le handover.

Nous avons essayé de simuler des scénarios de transmission de données dans un réseau LTE tout en appliquant le handover, puis on a comparé les résultats obtenus avec les performances théoriques de ce réseau. Les résultats théoriques s'avèrent très satisfaisants et reflètent les grandes performances que la 4G a apportées.

Par la suite de notre réalisation, nous ambitionnons à améliorer davantage notre réseau (en créant plus de cellules, augmentant plus de stations mobiles et simulant d'autres scénarios de mobilité) pour confirmer encore plus la puissance du réseau 4G, tout en montrant ses limites. Nous aspirons aussi, dans l'avenir proche, à faire la simulation de la 5G afin de voir à quel point les transitions technologiques de réseaux mobiles améliorent de plus en plus les performances.

Enfin, malgré toutes les difficultés qu'on a rencontrées au cours de notre réalisation, en termes d'installations logicielles, de réalisation des codes, de déduction des performances et de courts créneaux horaires proposés pour faire le projet, on a réussi à atteindre notre objectif principal. En somme, ce projet de fin d'études a été un accomplissement riche en expérience et en acquis scientifiques. Il nous a permis de les consolider et nous permettra dans l'avenir de les mettre en pratique aussi bien dans notre stage de fin d'études que dans notre vie d'ingénieurs.

Annexes

1- Code source C++ :

- lte.cc

```

/***** Projet de fin d'etudes *****/
*
* Nom du projet :
*
* - Etude de performances des reseaux LTE
*
*
* Realise par :
*
* - Ayoub Taihi
*
* - Racim Haffaf
*
*
* Encadre par :
*
* - M. Abdennebi
*
*****/

/** I- Librairies **/

#include "ns3/flow-monitor.h"
#include "ns3/flow-monitor-helper.h"
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/mobility-module.h"
#include "ns3/lte-module.h"
#include "ns3/applications-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/config-store-module.h"
#include "ns3/netanim-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("LenaX2HandoverMeasures");

```

```
/**
 *
 */
```

```
/** II – Methodes */
```

```
void NotifyConnectionEstablishedUe (std::string context ,
                                     uint64_t imsi ,
                                     uint16_t cellid ,
                                     uint16_t rnti)
```

```
{
    std::cout << context
               << " UE IMSI " << imsi
               << ": connected to CellId " << cellid
               << " with RNTI " << rnti
               << std::endl;
}
```

```
void NotifyHandoverStartUe (std::string context ,
                             uint64_t imsi ,
                             uint16_t cellid ,
                             uint16_t rnti ,
                             uint16_t targetCellId)
```

```
{
    std::cout << context
               << " UE IMSI " << imsi
               << ": previously connected to CellId " << cellid
               << " with RNTI " << rnti
               << ", doing handover to CellId " << targetCellId
               << std::endl;
}
```

```
void NotifyHandoverEndOkUe (std::string context ,
                             uint64_t imsi ,
                             uint16_t cellid ,
                             uint16_t rnti)
```

```
{
    std::cout << context
               << " UE IMSI " << imsi
               << ": successful handover to CellId " << cellid
               << " with RNTI " << rnti
               << std::endl;
}
```

```

void NotifyConnectionEstablishedEnb (std::string context ,
                                     uint64_t imsi ,
                                     uint16_t cellid ,
                                     uint16_t rnti)
{
    std::cout << context
               << " eNB CellId " << cellid
               << ": successful connection of UE with IMSI " << imsi
               << " RNTI " << rnti
               << std::endl;
}

void NotifyHandoverStartEnb (std::string context ,
                             uint64_t imsi ,
                             uint16_t cellid ,
                             uint16_t rnti ,
                             uint16_t targetCellId)
{
    std::cout << context
               << " eNB CellId " << cellid
               << ": start handover of UE with IMSI " << imsi
               << " RNTI " << rnti
               << " to CellId " << targetCellId
               << std::endl;
}

void NotifyHandoverEndOkEnb (std::string context ,
                             uint64_t imsi ,
                             uint16_t cellid ,
                             uint16_t rnti)
{
    std::cout << context
               << " eNB CellId " << cellid
               << ": completed handover of UE with IMSI " << imsi
               << " RNTI " << rnti
               << std::endl;
}

```

```

/*****

```

```

/** III – Programme principal */

```

```

int main (int argc , char *argv[])

{

CommandLine cmd;

Ptr<LteHelper> lteHelper = CreateObject<LteHelper> ();

// Instanciation de certains objets communs (par exemple, l'objet Channel).
// Methodes de configuration et d'ajout des ENodes B et des UEs.

Config::SetDefault ("ns3::UdpClient::Interval", TimeValue (Milliseconds (1000)));
Config::SetDefault ("ns3::UdpClient::MaxPackets", UIntegerValue (100000));
Config::SetDefault ("ns3::LteHelper::UseIdealRrc", BooleanValue (false));

// Mise en oeuvre du handover.

Ptr<PointToPointEpcHelper> epcHelper = CreateObject<PointToPointEpcHelper> ();

lteHelper->SetEpcHelper (epcHelper);
lteHelper->SetSchedulerType ("ns3::RrFfMacScheduler");

lteHelper->SetHandoverAlgorithmType ("ns3::A2A4RsrqHandoverAlgorithm");
lteHelper->SetHandoverAlgorithmAttribute ("ServingCellThreshold",
                                         UIntegerValue (30));
lteHelper->SetHandoverAlgorithmAttribute ("NeighbourCellOffset",
                                         UIntegerValue (1));

// Creation du Packet Data Network Gateway (PGW).

Ptr<Node> pgw = epcHelper->GetPgwNode ();

// Configuration , installation et creation d'une machine distante .

NodeContainer remoteHostContainer;
remoteHostContainer.Create (1);
Ptr<Node> remoteHost = remoteHostContainer.Get (0);
InternetStackHelper internet;
internet.Install (remoteHostContainer);

// Creation d'un reseau Internet.

```

```

PointToPointHelper p2ph;
p2ph.SetDeviceAttribute ("DataRate", DataRateValue (DataRate ("100Gb/s")));
p2ph.SetDeviceAttribute ("Mtu", UIntegerValue (1500));
p2ph.SetChannelAttribute ("Delay", TimeValue (Seconds (0.010)));
NetDeviceContainer internetDevices = p2ph.Install (pgw, remoteHost);
Ipv4AddressHelper ipv4h;
ipv4h.SetBase ("1.0.0.0", "255.0.0.0");
Ipv4InterfaceContainer internetIpIfaces = ipv4h.Assign (internetDevices);
Ipv4Address remoteHostAddr = internetIpIfaces.GetAddress (1);

// Routage Internet (vers le reseau LTE).

Ipv4StaticRoutingHelper ipv4RoutingHelper;
Ptr<Ipv4StaticRouting> remoteHostStaticRouting = ipv4RoutingHelper.GetStaticRouting

// Creation des interfaces :
    // - 0 signifie localhost.
    // - 1 signifie le peripherique p2p.

remoteHostStaticRouting->AddNetworkRouteTo (Ipv4Address ("7.0.0.0"), Ipv4Mask ("255

// Creation des objets "Node" pour les ENodes B et les UEs.

NodeContainer enbNodes;
enbNodes.Create (2);
NodeContainer ueNodes;
ueNodes.Create (6);

// Installation d'un scenario de mobilite dans les ENodes B (scenario que nous avon

Ptr<ListPositionAllocator> enbPositionAlloc = CreateObject<ListPositionAllocator>

Vector enbPosition (300 , 200, 0);
enbPositionAlloc->Add (enbPosition);

Vector enbPosition1 (600 , 200, 0);
enbPositionAlloc->Add (enbPosition1);

MobilityHelper enbMobility;

```

```

enbMobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
enbMobility.SetPositionAllocator (enbPositionAlloc);
enbMobility.Install (enbNodes);

// Installation d'un scenario de mobilite dans les UEs (scenario que nous avons ch

MobilityHelper ueMobility;
ueMobility.SetMobilityModel ("ns3::ConstantVelocityMobilityModel");
ueMobility.Install (ueNodes);

for (uint16_t i = 0; i < 3; i++)
{
    ueNodes.Get (i)->GetObject<MobilityModel> ()->SetPosition (Vector (0, 10 * (i),
    ueNodes.Get (i)->GetObject<ConstantVelocityMobilityModel> ()->SetVelocity (Vect
}

for (uint16_t i = 3; i < 6; i++)
{
    ueNodes.Get (i)->GetObject<MobilityModel> ()->SetPosition (Vector (800, 10 * (i
    ueNodes.Get (i)->GetObject<ConstantVelocityMobilityModel> ()->SetVelocity (Vect
}

// Installation d'une pile de protocoles LTE sur les ENodes B.

NetDeviceContainer enbDevs;
enbDevs = lteHelper->InstallEnbDevice (enbNodes);

// Installation d'une pile de protocoles LTE sur les UEs.

NetDeviceContainer ueDevs;
ueDevs = lteHelper->InstallUeDevice (ueNodes);

// Installation d'une pile IP sur les UEs.

internet.Install (ueNodes);
Ipv4InterfaceContainer ueIpIfaces;
ueIpIfaces = epcHelper->AssignUeIpv4Address (NetDeviceContainer (ueDevs));

// Liaison de chaque triplet d'UEs+FFFDleur ENode B, pour les raisons suivantes :
// - Configuration de chaque UE en fonction de la configuration ENode B.
// - Creation d'une connexion RRC entre eux.

```

```

for (uint16_t i = 0; i < 3; i++)
{
    lteHelper->Attach (ueDevs.Get (i), enbDevs.Get (0));
}

for (uint16_t i = 3; i < 6; i++)
{
    lteHelper->Attach (ueDevs.Get (i), enbDevs.Get (1));
}

NS_LOG_LOGIC ("setting up applications");
uint16_t dlPort = 10000;
uint16_t ulPort = 20000;
Ptr<UniformRandomVariable> startTimeSeconds = CreateObject<UniformRandomVariable> (

    startTimeSeconds->SetAttribute ("Min", DoubleValue (0));
    startTimeSeconds->SetAttribute ("Max", DoubleValue (0.010));

    // Activation du canal radio "bearer" de flux de donnees entre chaque UE et son ENB

for (uint32_t u = 0; u < 6; ++u)
{
    Ptr<Node> ue = ueNodes.Get (u);

    // Definition de la passerelle (gateway) par default pour chaque UE.

    Ptr<Ipv4StaticRouting> ueStaticRouting = ipv4RoutingHelper.GetStaticRouting (ue);
    ueStaticRouting->SetDefaultRoute (epcHelper->GetUeDefaultGatewayAddress (), 1);

    for (uint32_t b = 0; b < 1; ++b)
    {
        ++dlPort;
        ++ulPort;

        ApplicationContainer clientApps;
        ApplicationContainer serverApps;

        NS_LOG_LOGIC ("installing UDP DL app for UE " << u);
        UdpClientHelper dlClientHelper (ueIpIfaces.GetAddress (u), dlPort);
        clientApps.Add (dlClientHelper.Install (remoteHost));
        PacketSinkHelper dlPacketSinkHelper ("ns3::UdpSocketFactory",

```



```

                                InetSocketAddress (Ipv4Address::GetAn
serverApps.Add (dlPacketSinkHelper.Install (ue));

NS_LOG_LOGIC (" installing UDP UL app for UE " << u);
UdpClientHelper ulClientHelper (remoteHostAddr, ulPort);
clientApps.Add (ulClientHelper.Install (ue));
PacketSinkHelper ulPacketSinkHelper ("ns3::UdpSocketFactory",
                                InetSocketAddress (Ipv4Address::GetAn
serverApps.Add (ulPacketSinkHelper.Install (remoteHost));

Ptr<EpcTft> tft = Create<EpcTft> ();
EpcTft::PacketFilter dlpf;
dlpf.localPortStart = dlPort;
dlpf.localPortEnd = dlPort;
tft->Add (dlpf);
EpcTft::PacketFilter ulpf;
ulpf.remotePortStart = ulPort;
ulpf.remotePortEnd = ulPort;
tft->Add (ulpf);
EpsBearer bearer (EpsBearer::NGBR_VIDEO_TCP_DEFAULT);
lteHelper->ActivateDedicatedEpsBearer (ueDevs.Get (u), bearer, tft);

Time startTime = Seconds (startTimeSeconds->GetValue ());
serverApps.Start (startTime);
clientApps.Start (startTime);

    }
}

// Ajout des interfaces X2 pour l'activation du handover.

lteHelper->AddX2Interface (enbNodes);

// Connexion des usagers personnalisés pour l'établissement de la connexion RRC et

Config::Connect ("/NodeList/*/DeviceList/*/LteEnbRrc/ConnectionEstablished",
                MakeCallback (&NotifyConnectionEstablishedEnb));
Config::Connect ("/NodeList/*/DeviceList/*/LteUeRrc/ConnectionEstablished",
                MakeCallback (&NotifyConnectionEstablishedUe));
Config::Connect ("/NodeList/*/DeviceList/*/LteEnbRrc/HandoverStart",

```

```

        MakeCallback (&NotifyHandoverStartEnb));
Config::Connect ("/NodeList/*/DeviceList/*/LteUeRrc/HandoverStart",
        MakeCallback (&NotifyHandoverStartUe));
Config::Connect ("/NodeList/*/DeviceList/*/LteEnbRrc/HandoverEndOk",
        MakeCallback (&NotifyHandoverEndOkEnb));
Config::Connect ("/NodeList/*/DeviceList/*/LteUeRrc/HandoverEndOk",
        MakeCallback (&NotifyHandoverEndOkUe));

Ptr<FlowMonitor> flowMonitor;
FlowMonitorHelper flowHelper;
flowMonitor = flowHelper.InstallAll();

// Precision du temps d'arret de la simulation.

Simulator::Stop (Seconds (45));

/* Ceci est necessaire pour l'arret de la simulation , car :
    - L'evenement de debut de sous-trame est planifie a plusieurs reprises ,
    - Le planificateur du simulateur NS3 ne manquera donc jamais d'evenements
*/

lteHelper->EnableMacTraces ();
lteHelper->EnableRlcTraces ();

// Execution de la simulation.

AnimationInterface anim ("lte_handover_anim.xml");
anim.SetMobilityPollInterval (Seconds (1));
anim.SetMaxPktsPerTraceFile (1000000000);

Simulator::Run ();
flowMonitor->SerializeToXmlFile ("lte_log.xml", true , true);
Simulator::Destroy ();
return 0;

}

```

2- Code Matlab :

```

clc;
clear all;
close all;

```

```
%% Performances de notre reseau 4G simule par NS3. %%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    % 1. Debit en fonction du SNR. %
```

```
data_throughput = importdata('SNRVSThroughput_.plt');
SNR = data_throughput([1:end],1);
throughput = data_throughput([1:end],2);

figure;
plot(SNR, throughput, 'LineWidth', 2, 'Color', [0.9414,0.014,0.1614]);
title("Courbe de debit de transmission en fonction du temps")
xlabel("SNR(dB)")
ylabel("Db(Mbps)")
grid on;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    % 2. Delai d'attente %
```

```
data_delay = importdata('DlPdcPStats.txt');
time = data_delay([1:end],2);
delay = data_delay([1:end],12);

figure;

plot(time, delay, 'LineWidth', 4, 'Color', [0.614,0.564,0.874]);

title("Courbe de delai d'attente en fonction du temps de transmission")
xlabel("deltat(s)");      ylabel("t(s)")
grid on;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    % 3. Perte de paquets et taux de perte %
```

```
data_packet_loss = importdata('DIRlcStats.txt');
nTx = data_packet_loss([1:end], 8);
nRx = data_packet_loss([1:end], 10);
```

```
figure;  
plot(time, abs(nRx - nTx), 'LineWidth', 4, 'Color', [0.414, 0.6344, 0.174]);  
title("Courbe de perte de paquets en fonction du temps de transmission")  
xlabel("t(s)");          ylabel("Paquets perdus")  
grid on;
```

```
rate_loss = abs(nRx - nTx) ./ nTx;
```

% time	cellId	IMSI	RNTI	layer	mcs	size	rv	ndi	ccId
1013	2	4	1	0	6	325	0	1	0
1014	2	4	1	0	6	153	0	1	0
1014	2	5	3	0	6	153	0	1	0
1015	1	1	2	0	6	153	0	1	0
1015	1	3	1	0	6	153	0	1	0
1015	2	4	1	0	6	153	0	1	0
1015	2	5	3	0	6	153	0	1	0
1016	1	1	2	0	6	153	0	1	0
1016	1	3	1	0	6	153	0	1	0
1016	2	4	1	0	6	153	0	1	0
1016	2	5	3	0	6	153	0	1	0
1017	1	1	2	0	6	153	0	1	0
1017	1	3	1	0	6	153	0	1	0
1017	2	4	1	0	6	153	0	1	0
1017	2	5	3	0	6	153	0	1	0
1018	1	1	2	0	6	153	0	1	0
1018	1	3	1	0	6	153	0	1	0
1018	2	4	1	0	6	153	0	1	0
1018	2	5	3	0	6	153	0	1	0
1019	1	1	2	0	6	153	0	1	0
1019	1	3	1	0	6	153	0	1	0

% time	cellid	IMSI	RNTI	layer	mcs	size	rv	ndi	correct	ccld
1013	2	4	1	0	6	325	0	1	1	0
1014	2	4	1	0	6	153	0	1	1	0
1014	2	5	3	0	6	153	0	1	1	0
1015	1	3	1	0	6	153	0	1	1	0
1015	1	1	2	0	6	153	0	1	1	0
1015	2	4	1	0	6	153	0	1	1	0
1015	2	5	3	0	6	153	0	1	1	0
1016	1	3	1	0	6	153	0	1	1	0
1016	1	1	2	0	6	153	0	1	1	0
1016	2	4	1	0	6	153	0	1	1	0
1016	2	5	3	0	6	153	0	1	1	0
1017	1	3	1	0	6	153	0	1	1	0
1017	1	1	2	0	6	153	0	1	1	0
1017	2	4	1	0	6	153	0	1	1	0
1017	2	5	3	0	6	153	0	1	1	0
1018	1	3	1	0	6	153	0	1	1	0
1018	1	1	2	0	6	153	0	1	1	0
1018	2	4	1	0	6	153	0	1	1	0
1018	2	5	3	0	6	153	0	1	1	0
1019	1	3	1	0	6	153	0	1	1	0
1019	1	1	2	0	6	153	0	1	1	0

% start	end	CellId	IMSI	RNTI	LCID	nTxPDUs	TxBytes	nRxPDUs	RxBytes	delay	stdDev	min	max	PduSize	stdDev
1	1.25	1	1	2	4	9	1072	9	1072	0.00471429	0	0	0.00471429	0.004714	0.004714
1	1.25	1	2	3	4	7	1068	7	1068	0.00471429	0	0	0.00471429	0.004714	0.004714
1	1.25	1	3	1	4	9	1072	9	1072	0.00471429	0	0	0.00471429	0.004714	0.004714
1	1.25	2	4	1	4	6	1066	6	1066	0.00471429	0	0	0.00471429	0.004714	0.004714
1	1.25	2	5	3	4	7	1068	7	1068	0.00471429	0	0	0.00471429	0.004714	0.004714
1	1.25	2	6	2	4	5	1064	5	1064	0.00471429	0	0	0.00471429	0.004714	0.004714
2	2.25	1	1	2	4	6	1066	6	1066	0.00471429	0	0	0.00471429	0.004714	0.004714
2	2.25	1	2	3	4	4	1062	4	1062	0.00471429	0	0	0.00471429	0.004714	0.004714
2	2.25	1	3	1	4	6	1066	6	1066	0.00471429	0	0	0.00471429	0.004714	0.004714
2	2.25	2	4	1	4	6	1066	6	1066	0.00471429	0	0	0.00471429	0.004714	0.004714
2	2.25	2	5	3	4	7	1068	7	1068	0.00471429	0	0	0.00471429	0.004714	0.004714
2	2.25	2	6	2	4	5	1064	5	1064	0.00471429	0	0	0.00471429	0.004714	0.004714
3	3.25	1	1	2	4	6	1066	6	1066	0.00471429	0	0	0.00471429	0.004714	0.004714
3	3.25	1	2	3	4	4	1062	4	1062	0.00471429	0	0	0.00471429	0.004714	0.004714
3	3.25	1	3	1	4	6	1066	6	1066	0.00471429	0	0	0.00471429	0.004714	0.004714
3	3.25	2	4	1	4	6	1066	6	1066	0.00471429	0	0	0.00471429	0.004714	0.004714
3	3.25	2	5	3	4	7	1068	7	1068	0.00471429	0	0	0.00471429	0.004714	0.004714
3	3.25	2	6	2	4	5	1064	5	1064	0.00471429	0	0	0.00471429	0.004714	0.004714
4	4.25	1	1	2	4	5	1064	5	1064	0.00471429	0	0	0.00471429	0.004714	0.004714
4	4.25	1	2	3	4	3	1060	3	1060	0.00471429	0	0	0.00471429	0.004714	0.004714
4	4.25	1	3	1	4	6	1066	6	1066	0.00471429	0	0	0.00471429	0.004714	0.004714
4	4.25	2	4	1	4	6	1066	6	1066	0.00471429	0	0	0.00471429	0.004714	0.004714
4	4.25	2	5	3	4	7	1068	7	1068	0.00471429	0	0	0.00471429	0.004714	0.004714
4	4.25	2	6	2	4	5	1064	5	1064	0.00471429	0	0	0.00471429	0.004714	0.004714
5	5.25	1	1	2	4	4	1062	4	1062	0.00471429	0	0	0.00471429	0.004714	0.004714
5	5.25	1	2	3	4	2	1058	2	1058	0.00471429	0	0	0.00471429	0.004714	0.004714
5	5.25	1	3	1	4	5	1064	5	1064	0.00471429	0	0	0.00471429	0.004714	0.004714
5	5.25	2	4	1	4	4	1062	4	1062	0.00471429	0	0	0.00471429	0.004714	0.004714
5	5.25	2	5	3	4	5	1064	5	1064	0.00471429	0	0	0.00471429	0.004714	0.004714
5	5.25	2	6	2	4	4	1062	4	1062	0.00471429	0	0	0.00471429	0.004714	0.004714
6	6.25	1	1	2	4	3	1060	3	1060	0.00471429	0	0	0.00471429	0.004714	0.004714
6	6.25	1	2	3	4	2	1058	2	1058	0.00471429	0	0	0.00471429	0.004714	0.004714
6	6.25	1	3	1	4	4	1062	4	1062	0.00471429	0	0	0.00471429	0.004714	0.004714

FIGURE 23 –

% time	cellId	IMSI	RNTI	sinrLinear	componentCarrierId
0.041	1	3	1	3.36995 0	
0.041	2	4	1	3.86895 0	
0.042	1	1	2	3.41937 0	
0.042	2	6	2	3.9448 0	
0.043	1	2	3	3.39708 0	
0.043	2	5	3	3.90543 0	
0.081	1	3	1	3.4008 0	
0.081	2	4	1	3.85722 0	
0.082	1	1	2	3.45212 0	
0.082	2	6	2	3.93376 0	
0.083	1	2	3	3.42894 0	
0.083	2	5	3	3.89402 0	
0.121	1	3	1	3.43203 0	
0.121	2	4	1	3.84552 0	
0.122	1	1	2	3.48532 0	
0.122	2	6	2	3.92274 0	
0.123	1	2	3	3.46122 0	
0.123	2	5	3	3.88263 0	
0.161	1	3	1	3.46367 0	
0.161	2	4	1	3.83383 0	
0.162	1	1	2	3.51899 0	
0.162	2	6	2	3.91173 0	
0.163	1	2	3	3.49393 0	
0.163	2	5	3	3.87126 0	
0.201	1	3	1	3.49571 0	
0.201	2	4	1	3.82216 0	
0.202	1	1	2	3.55313 0	
0.202	2	6	2	3.90074 0	
0.203	1	2	3	3.52709 0	
0.203	2	5	3	3.85991 0	
0.241	1	3	1	3.52816 0	
0.241	2	4	1	3.81051 0	
0.242	1	1	2	3.58774 0	

FIGURE 24 –

[illegible]

FIGURE 25 –

% time	cellId	IMSI	frame	sframe	RNTI	mcs	size	ccId
1.007	2	4	101	8	1	6	325	0
1.008	2	4	101	9	1	6	153	0
1.008	2	5	101	9	3	6	153	0
1.009	1	3	101	10	1	6	153	0
1.009	1	1	101	10	2	6	153	0
1.009	2	4	101	10	1	6	153	0
1.009	2	5	101	10	3	6	153	0
1.01	1	3	102	1	1	6	153	0
1.01	1	1	102	1	2	6	153	0
1.01	2	4	102	1	1	6	153	0
1.01	2	5	102	1	3	6	153	0
1.011	1	3	102	2	1	6	153	0
1.011	1	1	102	2	2	6	153	0
1.011	2	4	102	2	1	6	153	0
1.011	2	5	102	2	3	6	153	0
1.012	1	3	102	3	1	6	153	0
1.012	1	1	102	3	2	6	153	0
1.012	2	4	102	3	1	6	153	0
1.012	2	5	102	3	3	6	153	0
1.013	1	3	102	4	1	6	153	0
1.013	1	1	102	4	2	6	153	0
1.013	2	4	102	4	1	6	153	0
1.013	2	5	102	4	3	6	153	0
1.014	1	3	102	5	1	6	101	0
1.014	1	1	102	5	2	6	101	0
1.014	1	2	102	5	3	6	101	0
1.014	2	6	102	5	2	6	153	0
1.014	2	5	102	5	3	6	153	0
1.015	1	3	102	6	1	6	101	0
1.015	1	1	102	6	2	6	101	0
1.015	1	2	102	6	3	6	101	0
1.015	2	6	102	6	2	6	153	0
1.015	2	5	102	6	3	6	153	0

FIGURE 26 –

Références

<https://www.echosdunet.net/dossiers/4g-lte#9/48.8/2.3>

https://fr.wikipedia.org/wiki/LTE_Advanced
<https://en.wikipedia.org/wiki/2G>
marche-public.fr/Terminologie/Entrees/2G.htm
<https://fr.wikipedia.org/wiki/3G>
https://fr.wikipedia.org/wiki/General_Packet_Radio_Service
https://fr.wikipedia.org/wiki/Gateway_GPRS_Support_Node
https://fr.wikipedia.org/wiki/Serving_GPRS_Support_Node
<http://www.marche-public.fr/Terminologie/Entrees/2G.htm>
<http://www.mcours.net/cours/pdf/leilclic3/leilclic949.pdf>
<https://blogs.univ-poitiers.fr/f-launay/2011/08/22/evolutiondelaigversla4g/#:~:text=L'avantage%,20de%20cette%20connexion,entre%20elles%20par%20plusieurs%20voies.>
<https://www.digitalcorner-wavestone.com/2020/01/de-la-2g-a-la-4g/>
<https://www.jilnetwork.com/2019/07/definition-des-reseaux-mobiles-1g-2g-3g-4g-5g.html>
https://fr.wikipedia.org/wiki/Visitor_Location_Register
https://fr.wikipedia.org/wiki/Home_Location_Register
<https://web.maths.unsw.edu.au/~lafaye/CCM/telephonie-mobile/edge.htm>
<https://fr.wikipedia.org/wiki/3G>
<https://fr.sawakinome.com/articles/access-network-communication/difference-between-bts-and-node-b.html>
<https://boowiki.info/art/telephonie-mobile/noeud-b.html>
https://fr.wikipedia.org/wiki/ENode_B
<https://parsons-technology.com/fr/wcdma-ou-gsm-quelle-difference/>
<https://www.memoireonline.com/09/13/7373/Image-8-Outil-d-analyse-cartographique-d-aide--l-ameliora.html>
[https://fr.wikipedia.org/wiki/LTE_\(r%C3%A9seaux_mobiles\)](https://fr.wikipedia.org/wiki/LTE_(r%C3%A9seaux_mobiles))
<https://fr.wikipedia.org/wiki/OFDMA>
<https://definir-tech.com/noyau-de-paquet-evolue-epc/>
https://fr.wikipedia.org/wiki/Radio_Network_Controller
https://fr.wikipedia.org/wiki/IP_multimedia_subsystem
https://www.juniper.net/documentation/en_US/junos-mobility11.2/topics/concept/gateways-mobility-gg.html
<https://fr.wikipedia.org/wiki/EUTRAN#:~:text=Le%20terme%20EUTRAN%20d%C3%A9signe%20la,normalis%C3%A9%20par%20l'organisme%203GPP.>
<https://www.microwavejournal.com/ext/resources/whitepapers/July2012/IMS-Architecture-White-Paper.pdf?1612492910>
https://www.echosdunet.net/dossiers/dossier_13225_de%204g-lte%20lte-advanced%20quels%20debits%20.html
<https://blogs.univ-poitiers.fr/f-launay/2014/02/28/gestion-de-litinerance-part-2-la-mobilite-des-u>
<https://upcommons.upc.edu/bitstream/handle/2099.1/21093/LTE+Handover+Performance+Evaluation.pdf?sequence=4>
https://polytechnique.cm/ENSP/Cours/TP%20de%20Simulation%20de%20r%C3%A9seaux%20de%20transmission%20avec%20NS3%20et%20Agent%20conversationnel_4GTEL2020.pdf

<https://www.choisir.com/forfait-mobile/articles/132535/4g-vs-5g-queelles-sont-les-differences#:~:text=Une%20vitesse%20de%20t%C3%A9chargement%20plus%20rapide&text=Avec%20la%204G%20il%20est,de%20t%C3%A9chargement%20des%20fichiers%20multim%C3%9dia>.
<http://homepages.rpi.edu/~wangy52/PersonalWebsite/build/html/Network/NS/NS3/NS3-tutorial/01-NetAnim/NS3-NetAnim.html>
<https://www.nsnam.org/docs/release/3.28/models/singlehtml/index.html>
<https://www.nsnam.org/docs/tutorial/html/getting-started.html>
<https://blogs.univ-poitiers.fr/f-launay/tag/gnb/#:~:text=L%27interface%20X2%20peut%20%C3%Aatre,les%20deux%20stations%20de%20base>.
<https://www.3glteinfo.com/intra-lte-handover-using-x2-interface/>
<https://doc.ubuntu-fr.org/network-manager>
<https://openclassrooms.com/forum/sujet/difference-entre-make-et-make-clean-64315>