

# SAE 5.01

Mise en place d'un réseau 5G

21/02/2025

## Sommaire

Sommaire.....	2
Glossaire.....	3
Liste des figures .....	6
Cahier des charges.....	7
Chapitre I : Introduction .....	8
Chapitre II : Outils et matériel utilisés .....	9
Open 5GS.....	9
SRS RAN Project .....	9
USRP B210 .....	9
Omni Key 6121.....	9
Carte Sim .....	9
Téléphone .....	9
Chapitre III : Système de transmission .....	10
Configuration Requisite.....	10
Open5gs.....	10
srsRAN Project.....	13
PySim.....	15
Création de notre réseau 5G .....	16
Handover .....	20
Chapitre IV : Expérimentations et résultats.....	22
Chapitre V : Gestion de projet.....	25
Description de la méthode.....	25
Organisation des sprints.....	25
Chapitre VI : Conclusion et retour d'expérience .....	27
Annexe / Bibliographie .....	28

## Glossaire

Terme	Définition
AMF (Access and Mobility Management Function)	Composant du cœur de réseau, il gère les connexions et la mobilité des utilisateurs. Il est responsable de l'authentification et de l'autorisation des abonnés, ainsi que de la gestion des handovers entre les stations de base 5G (gNB), garantissant une continuité de service sans interruption.
ADM1 (Authentication Data Management 1)	Code d'authentification supplémentaire utilisé pour renforcer la sécurité des communications et des transactions sur le réseau mobile. Il est souvent utilisé en complément des autres clés de sécurité pour assurer une protection accrue des données de l'abonné.
APN (Access Point Name)	Nom du point d'accès.
AUSF (Authentication Server Function)	Composant du cœur de réseau, il vérifie l'identité des utilisateurs et des dispositifs qui tentent de se connecter au réseau.
BSF (Binding Support Function)	Composant du cœur de réseau, il gère les liaisons entre les différentes fonctions réseau, assurant une communication fluide et efficace entre elles.
gNB (gNodeB)	C'est un noeud qui sert de lien entre les équipements utilisateurs (UE) et le reste du réseau. C'est équivalent d'une station de base.
ICCID (Integrated Circuit Card Identifier)	Numéro unique gravé sur la carte SIM qui l'identifie physiquement. Il est utilisé par les opérateurs pour gérer les abonnements.
IMSI (International Mobile Subscriber Identity)	Identifiant unique de la carte SIM, il est composé du mnc, mcc et du numéro de client.
Ki (Key Identifier)	Clé secrète de 128 bits (16 octets) attribuée à chaque carte SIM par l'opérateur. Elle est utilisée pour l'authentification de l'abonné sur le réseau mobile et ne doit jamais être transmise en clair.
MCC (Mobile Country Code)	Code à 3 chiffres identifiant le pays d'un réseau mobile.

## Glossaire

MNC (Mobile Network Code)	Code à 2 ou 3 chiffres identifiant l'opérateur mobile dans un pays.
NRF (Network Repository Function)	Il fonctionne comme un référentiel centralisé pour toutes les fonctions réseau 5G dans le réseau de l'opérateur.
NSSF (Network Slice Selection Function)	Composant du cœur de réseau, il sélectionne les tranches de réseau appropriées pour les utilisateurs en fonction des exigences de service et des politiques de l'opérateur.
OPC (Operator Variant Algorithm Configuration Field)	Version dérivée de la clé OP (Operator Key), combinée avec Ki. Elle est utilisée pour sécuriser l'authentification et le chiffrement des communications en 3G, 4G et 5G, évitant ainsi l'envoi direct de Ki.
PCF (Policy Control Function)	Fonction de politique et de tarification : Le PCF gère les politiques et la tarification des abonnés.
PLMN ID	Un identifiant unique utilisé pour distinguer les réseaux mobiles publics composé de MCC et de MNC
RRC (Radio Resource Control)	Protocole gérant les connexions.
SCP (Service Communication Proxy)	Un composant crucial de l'architecture 5G qui agit comme un intermédiaire pour améliorer la communication entre les différentes fonctions réseau. Il optimise la communication, pour une efficacité et des performances optimales du réseau.
SEPP (Security Edge Protection Proxy)	Composant du cœur de réseau, assure la sécurité des communications entre les réseaux 5G domestiques et visités. Il fournit une communication sécurisée et authentifiée entre les réseaux.
SMF (Session Management Function)	Fonction de gestion de session : Le SMF gère les sessions utilisateur, y compris l'établissement, la modification et la libération des sessions.
TAC (Tracking Area Code)	Un identifiant utilisé dans les réseaux mobiles pour identifier une zone de suivi spécifique.

## Glossaire

UDM (Unified Data Management)	Composant du cœur de réseau qui gère les données des abonnés, y compris les informations d'authentification et les profils des utilisateurs.
UDR (Unified Data Repository)	Composant du cœur de réseau qui stocke les informations sur les abonnés, telles que les services disponibles, les paramètres de QoS et les profils des utilisateurs.
UPF (User Plane Function)	Composant clé du cœur de réseau 5G, responsable du traitement et du transport des données utilisateur entre le réseau et les utilisateurs finaux. Il gère le routage des paquets, l'application des politiques de qualité de service (QoS) et l'encapsulation des données pour leur transmission via le réseau.

## Liste des figures

Figure 1 - logo d'open 5GS.....	10
Figure 2 - schéma de notre architecture .....	11
Figure 3 - interface web d'open5GS.....	13
Figure 4 - logo srsRAN project.....	14
Figure 5 - interface de création d'un nouvel utilisateur.....	17
Figure 6 - fichier amf.yml.....	17
Figure 7 - extrait fichier amf.yml.....	18
Figure 8 - extrait du fichier upf.yml .....	18
Figure 9 - extrait du fichier smf.yml .....	19
Figure 10 - extrait du fichier nrf.yml.....	19
Figure 11 - extrait du fichier gnb_rf_b200_tdd_n78_20mhz.yml.....	20
Figure 12 - schéma du handover .....	21
Figure 13 – paramètres du point d'accès sur le téléphone .....	22
Figure 14 - configuration réseau du téléphone .....	23
Figure 15 - rubrique ru_sdr dans le fichier gnb_rf_b200_tdd_n78_20mhz.yml .....	23
Figure 16 – lancement de l'antenne et vérification du fonctionnement du handover ...	24

## Cahier des charges

La 5G est une technologie de communication mobile plus performante que les précédentes. Nous avons pour but de déployer un réseau 5G entièrement fonctionnel entre une antenne et un téléphone.

Objectif :

- Mettre en place un réseau 5G fonctionnel
- Mettre en place un cœur de réseau avec Open5GS
- Mettre en place une station de base gNB avec srsRAN
- Tester le réseau 5G avec un équipement utilisateur

Architecture du système :

- Cœur de réseau 5G : Open 5GS exécuté depuis un serveur linux
- Station de base simulée avec srs RAN
- Simuler des UE grâce à un téléphone équipé d'une carte SIM

Contraintes :

- Compatibilité avec les normes propre à la 5G
- Disponibilités du matériel, module USRP B210 et surtout carte SIM

## Chapitre I : Introduction

L'évolution des réseaux mobiles a permis d'améliorer la connectivité et d'introduire de nouveaux usages à chaque génération, en s'appuyant sur des protocoles spécifiques.

La **1G** (années 1980) était basée sur des communications analogiques et permettait uniquement les appels vocaux via le protocole AMPS (Advanced Mobile Phone System). Avec une vitesse théorique limitée à 2,4 Kbps, cette technologie souffrait d'une faible qualité audio et d'une sécurité limitée.

La **2G** (années 1990) a marqué la transition vers le numérique, introduisant les SMS et MMS, tout en s'appuyant sur les protocoles GSM, GPRS et EDGE. Les vitesses théoriques de la 2G variaient de 9,6 Kbps pour GSM à 384 Kbps pour EDGE, permettant des échanges de données simples et une meilleure gestion du spectre radio, tout en améliorant la sécurité.

La **3G** (années 2000) a apporté une avancée majeure en permettant l'accès à Internet mobile, le streaming vidéo et l'envoi d'emails, grâce aux protocoles UMTS et HSPA. Avec des vitesses théoriques atteignant 384 Kbps en mobilité et jusqu'à 14 Mbps avec HSPA, la 3G a ouvert la voie à une connectivité plus stable et plus rapide.

La **4G** (années 2010), reposant sur le protocole LTE, a considérablement augmenté les débits, permettant des usages plus gourmands en données, comme la visioconférence, le cloud computing et le téléchargement rapide, avec des vitesses théoriques pouvant atteindre jusqu'à 1 Gbps en stationnaire.

Enfin, la **5G** (années 2020) a révolutionné les télécommunications avec une latence réduite et des débits beaucoup plus élevés, allant jusqu'à 10 Gbps en téléchargement, grâce aux protocoles NR (New Radio) et 5GC (5G Core). Cette nouvelle génération a permis l'émergence de technologies avancées comme les objets connectés, la médecine à distance et les villes intelligentes.

Chaque génération a apporté des améliorations clés en termes de vitesse, de capacité et de nouvelles fonctionnalités, rendant les communications mobiles toujours plus performantes et polyvalentes.

Ainsi, dans le cadre de notre troisième année de BUT réseaux et télécommunications, nous avons pour mission, de mettre en place un réseau 5G, grâce aux outils, SRS RAN et Open 5GS.

Ce projet a pour but de comprendre l'architecture d'un réseau 5G et ses différents composants et de maîtriser les concepts de base des réseaux mobiles (EPC, gNB, UE).



## Chapitre II : Outils et matériel utilisés

Pour mener à bien ce projet nous utilisons deux outils Open 5GS, SRS RAN, PySim :

### Open 5GS

Open 5GS est un projet open source, permettant de simuler un réseau mobile, cet outil est disponible sur Ubuntu et Debian linux. C'est grâce à cela que nous simulons notre cœur de réseau 5G.

Pour l'installer, il faut suivre les instructions et utilise les commandes décrites sur la page quick start (lien en annexe).

### SRS RAN Project

Software Radio Systems Radio Access Network, est également un projet open source, qui sert à la configuration des réseaux 5G. Ici, cela nous servira à programmer l'antenne pour qu'elle émette à une puissance et une fréquence définie.

### USRP B210

L'USRP B210 est un émetteur-récepteur logiciel (SDR - Software Defined Radio) qui permet d'interfacer le cœur de réseau avec le signal radio. Cet équipement est essentiel pour simuler un réseau mobile en mode expérimental.

### Omni Key 6121

L'OmniKey 6121 est un lecteur de carte SIM qui nous permet d'accéder aux paramètres de la carte SIM, nécessaires à la configuration de notre réseau.

### Carte Sim

La carte SIM est une puce électronique qui stocke des informations nécessaires à la connexion et permet de recevoir le signal du réseau 5G. Pour configurer correctement notre srsRAN.

### Téléphone

Pour tester notre réseau 5G nous avons besoin d'un téléphone dans lequel insérer la carte SIM. Le réseau sera configuré en fonction de notre carte SIM. Ce téléphone mobile nous permettra de vérifier l'authentification ainsi que l'accès aux services du réseau.

## Chapitre III : Système de transmission

### Configuration Requise

Pour garantir le bon fonctionnement de Open5gs et de srsRAN Project, notre PC doit répondre aux spécifications suivantes :

- **Système d'exploitation (OS) :** Ubuntu 22.04
- **Mémoire vive (RAM) :** Minimum 16 Go

Il est crucial d'utiliser Ubuntu 22.04 car les applications requièrent cette version spécifique du système d'exploitation. Les dépendances et bibliothèques utilisées par Open5gs et srsRAN Project sont testées et validées pour cette version d'Ubuntu comme préciser dans les documentations des applications.

L'importance des 16 Go de RAM réside dans le fait que les informations traitées sont en temps réel et nécessitent une grande quantité de mémoire vive, pour assurer un fonctionnement fluide et efficace.

### Open5gs

Open5GS est une implémentation open-source pour le cœur de réseau 5G, c'est-à-dire le réseau central des réseaux LTE/NR. Il est conçu pour permettre la création et la gestion de réseau mobile NR/LTE, que ce soit pour des tests, de la recherche ou des déploiements.



*Figure 1 - logo d'open 5GS*

Open5gs est composé de plusieurs services que nous utilisons, pour notre noyau 5G SA. La 5G SA (Standalone) représente une nouvelle architecture de réseau mobile qui utilise exclusivement un noyau 5G, sans dépendance aux fonctions de contrôle du réseau 4G LTE. Contrairement à la 5G NSA (Non-Standalone), qui s'appuie sur l'infrastructure 4G existante, la 5G SA est construite à partir de zéro avec des technologies 5G pour le réseau central.

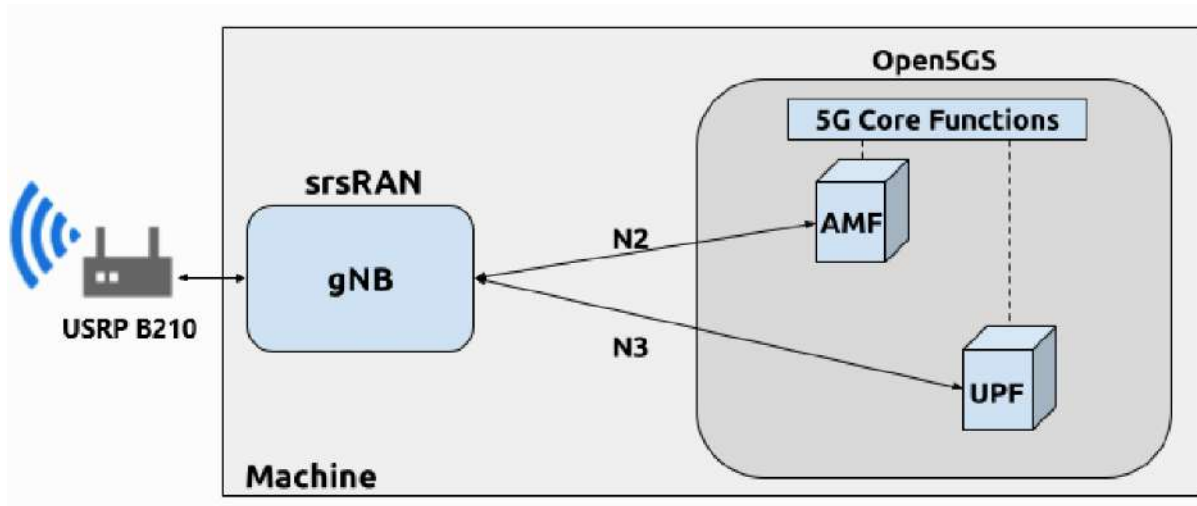


Figure 2 - schéma de notre architecture

La 5G SA utilise les services suivants :

- **NRF - Fonction de référentiel NF** : Le NRF fonctionne comme un référentiel centralisé pour toutes les fonctions réseau 5G dans le réseau de l'opérateur.
- **SCP - Proxy de communication de service** : Le SCP est un composant crucial de l'architecture 5G qui agit comme un intermédiaire pour améliorer la communication entre les différentes fonctions réseau. Il optimise la communication, pour une meilleure efficacité et des performances optimales du réseau.
- **SEPP - Proxy de protection de bord de sécurité** : Le SEPP assure la sécurité des communications entre les réseaux 5G domestiques et visités. Il fournit une communication sécurisée et authentifiée entre les réseaux.
- **AMF - Fonction de gestion des accès et de la mobilité** : L'AMF gère les connexions et la mobilité des utilisateurs. Il est responsable de l'authentification et de l'autorisation des abonnés, ainsi que de la gestion des handovers entre les stations de base 5G (gNB), garantissant une continuité de service sans interruption.
- **SMF - Fonction de gestion de session** : Le SMF gère les sessions utilisateur, y compris l'établissement, la modification et la libération des sessions.
- **UPF - Fonction de plan utilisateur** : L'UPF est responsable du traitement et du transport des données utilisateur entre le réseau et les utilisateurs finaux. Il gère le routage des paquets, l'application des politiques de qualité de service (QoS) et l'encapsulation des données pour leur transmission via le réseau.
- **AUSF - Fonction de serveur d'authentification** : L'AUSF vérifie l'identité des utilisateurs et des dispositifs qui tentent de se connecter au réseau.
- **UDM - Gestion unifiée des données** : Le UDM gère les données des abonnés, y compris les informations d'authentification et les profils des utilisateurs.
- **UDR - Référentiel de données unifié** : L'UDR stocke les informations sur les abonnés, telles que les services disponibles, les paramètres de QoS et les profils des utilisateurs.
- **PCF - Fonction de politique et de tarification** : Le PCF gère les politiques et la tarification des abonnés.

- **NSSF - Fonction de sélection de tranche de réseau :** Le NSSF sélectionne les tranches de réseau appropriées pour les utilisateurs en fonction des exigences de service et des politiques de l'opérateur.
- **BSF - Fonction de support de liaison :** Le BSF gère les liaisons entre les différentes fonctions réseau, assurant une communication fluide et efficace entre elles.

Nous allons maintenant installer Open5GS en suivant les étapes.

On importe la clé publique utilisée par le système de gestion des packages :

```
sudo apt update
sudo apt install gnupg
curl -fsSL https://pgp.mongodb.com/server-6.0.asc | sudo gpg -o
/usr/share/keyrings/mongodb-server-6.0.gpg --dearmor
```

On crée le fichier de liste `/etc/apt/sources.list.d/mongodb-org-6.0.list` pour notre version d'Ubuntu.

Installation de MongoDB pour la base de données.

```
sudo apt install -y mongodb-org
sudo systemctl start mongod
sudo systemctl enable mongod
```

Installation de Open5gs sur Ubuntu 22.04.

```
sudo add-apt-repository ppa:open5gs/latest
sudo apt update
sudo apt install open5gs
```

Il faut installer l'interface Web de Open5gs avec le package « Node.js », afin de modifier des clients de manière interactive et facile.

```
sudo apt update
sudo apt install -y ca-certificates curl gnupg
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo
gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg

NODE_MAJOR=20
```

## Chapitre III : Système de transmission

```
echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg]
https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | sudo tee
/etc/apt/sources.list.d/nodesource.list

sudo apt update
sudo apt install nodejs -y

$ curl -fsSL https://open5gs.org/open5gs/assets/webui/install | sudo -E bash
_
```

Dès à présent, on peut accéder à la visualisation de l'interface WEB de l'Open5gs, en entrant l'url suivant « <http://localhost:9999> ».

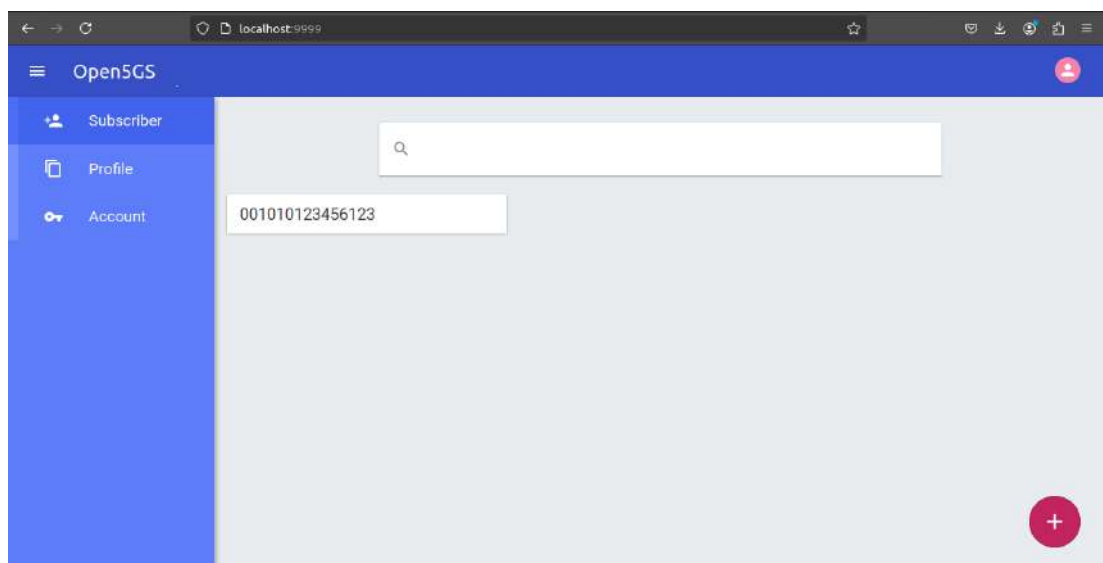


Figure 3 - interface web d'open5GS

### srsRAN Project

Le projet srsRAN Project est une solution open-source complète pour le réseau d'accès radio (RAN) 5G, développée par Software Radio Systems (SRS). SrsRAN offre une implémentation complète des couches L1, L2 et L3 avec des dépendances externes minimales.

Il inclut une unité centralisée et une unité distribuée natives O-RAN, permettant une flexibilité et une évolutivité accrues dans le déploiement des réseaux 5G. Portable à travers différentes architectures de processeurs, srsRAN est optimisé pour les architectures x86 et ARM, et peut être déployé sur des systèmes embarqués à faible consommation d'énergie jusqu'aux environnements cloudRAN.

Ce projet est idéal pour la recherche et le développement dans le domaine des communications mobiles, ainsi que pour les déploiements de réseaux privés 5G.



Figure 4 - logo srsRAN project

Tous d'abord, l'installation des différents dépendances nécessaires pour le bon fonctionnement :

```
sudo apt-get install cmake make gcc g++ pkg-config libfftw3-dev libmbedtls-dev libscrt-dev libyaml-cpp-dev libgtest-dev
```

De plus, l'installation du driver UHD pour la carte USB B210 est nécessaire.

```
sudo apt-get install uhd-hosts
```

Il faut ensuite télécharger les fichiers srsRAN Project à partir de github et l'installation du système et du gnb avec la commande « make » de la dépendance cmake.

```
sudo git clone https://github.com/srsRAN/srsRAN Project.git

cd srsRAN_Project
mkdir build
cd build
cmake ../
make -j $(nproc)
make test -j $(nproc)

cd /apps/gnb/
sudo make install
```

Pour finir, nous installons des derniers packages pour l'utilisation du système srsRAN Project.

```
sudo add-apt-repository ppa:softwareradiosystems/srsran-project
sudo apt-get update
sudo apt-get install srsran-project -y
```

### PySim

PySim est une implémentation en Python de divers logiciels destinés à la gestion des cartes SIM d'abonnés pour les réseaux cellulaires. Développé par Osmocom, PySim est utilisé pour provisionner des cartes SIM dans le cadre de réseaux cellulaires privés ou personnalisés. Le projet comprend plusieurs composants, notamment une bibliothèque Python, un programme interactif en ligne de commande (pySim-shell), un décodeur de traces APDU (pySim-trace), ainsi que les outils pySim-prog et pySim-read. PySim est une solution flexible et collaborative pour la gestion des cartes SIM, facilitant l'exploitation de réseaux mobiles privés.

Dans notre projet, nous utilisons pySim-prog pour écrire et configurer les paramètres de la carte SIM, et pySim-read pour extraire ces informations. Ces outils nous permettent, par la suite, d'enregistrer les abonnés dans le cœur de réseau et d'assurer la bonne intégration avec notre environnement 5G.

Un certain nombre d'informations caractérise une carte SIM :

- **MCC** : Code à 3 chiffres identifiant le pays d'un réseau mobile. Exemple : 208 pour la France, 310 pour les États-Unis.
- **MNC** : Code à 2 ou 3 chiffres identifiant l'opérateur mobile dans un pays. Exemple : 01 pour Orange France, 02 pour SFR.
- **Country Code** : Code du pays utilisé pour certaines configurations de la carte SIM, similaire au MCC mais utilisé dans un autre contexte.
- **IMSI** : Identifiant unique d'un abonné mobile, composé du MCC + MNC + un identifiant d'abonné (MSIN).
- **ICCID** : Numéro unique gravé sur la carte SIM qui l'identifie physiquement. Il est utilisé par les opérateurs pour gérer les abonnements.
- **Ki** : Clé secrète de 128 bits (16 octets) attribuée à chaque carte SIM par l'opérateur. Elle est utilisée pour l'authentification de l'abonné sur le réseau mobile et ne doit jamais être transmise en clair.
- **OPC** : Version dérivée de la clé OP (Operator Key), combinée avec Ki. Elle est utilisée pour sécuriser l'authentification et le chiffrement des communications en 3G, 4G et 5G, évitant ainsi l'envoi direct de Ki.
- **ADM1** : Code d'authentification supplémentaire utilisé pour renforcer la sécurité des communications et des transactions sur le réseau mobile. Il est souvent utilisé en complément des autres clés de sécurité pour assurer une protection accrue des données de l'abonné.

Tous d'abord, on procède à l'installation des différents dépendances nécessaires pour le bon fonctionnement :

```
sudo apt-get install python3-pip python3-serial python3-pyscard
```

Ensuite, nous installons PySim, en clonant le dépôt depuis GitHub avec les commandes ci-dessous.

```
sudo git clone https://gitea.osmocom.org/sim-card/pysim.git
cd pysim
pip3 install -r requirements.txt
```

Ci-dessous la signification de chaque paramètre :

- **-p** : Sélectionne le lecteur de carte SIM utilisé.
- **-n** : Définit le nom de l'opérateur.
- **-t** : Spécifie le type de carte SIM.
- **-i** : Définit l'IMSI.
- **-c** : Définit le code du pays.
- **-x** : Définit le Mobile Country Code.
- **-y** : Définit le Mobile Network Code.
- **-s** : Définit l'ICCID.
- **-k** : Définit le Ki.
- **-o** : Définit l'OPC.
- **-a** : Le code ADM1.

### Création de notre réseau 5G

Pour créer notre réseau 5G, nous commençons par programmer notre carte SIM afin d'y inscrire les informations nécessaires, qui seront ensuite utilisées dans notre cœur 5G. Pour ce faire, nous avons utilisé la commande «./pySim-prog.py » avec nos informations, permettant ainsi à la carte SIM de récupérer ces données.

Les informations sont :

- **IMSI** : 001010123456123
- **MCC** : 001
- **MNC** : 01
- **K** : 465B5CE8 B199B49F AA5F0A2E E238A6BC
- **AMF**: 8000
- **OPc/OP**: E8ED289D EBA952E4 283B54E8 8E6183CA

```
./pySim-prog.py -p 0 -s 8988211000001037962 -i 001010123456123 -k 465B5CE8
B199B49F AA5F0A2E E238A6BC -o E8ED289D EBA952E4 283B54E8 8E6183CA -a 94625567
-x 001 -y 01
```

Ensuite, il est nécessaire de créer un client dans l'interface Open5GS pour notre nouvelle carte SIM en utilisant les informations précédemment spécifiées. De plus, il faudra définir l'APN, que nous nommerons « FLNETWORK » dans notre cas.



## Chapitre III : Système de transmission

Subscriber Configuration

IMSI\*

001010123456123

+

Subscriber Key (K)\*

465B5CE8 8199B49F AA5FOAZE E238A6BG

Authentication Management Field (AMF)\*

8000

USIM Type

OPc

Operator Key (OPc/OP)\*

E8ED289D EBA952E4 283B54E8 8E6183CA

UE-AMBR Downlink\*

1

Unit

Gbps

UE-AMBR Uplink\*

1

Unit

Gbps

Subscriber Status (TS 29.272 7.3.29)

CANCEL SAVE

Figure 5 - interface de création d'un nouvel utilisateur

**PLMN ID** est un identifiant unique utilisé pour distinguer les réseaux mobiles publics composé de MCC et de MNC.

**TAC** est un identifiant utilisé dans les réseaux mobiles pour identifier une zone de suivi spécifique.

Dans notre configuration nous devons changer dans le fichier `amf.yml` le PLMN ID ainsi que le TAC :

```
sudo nano /etc/open5gs/amf.yml
```

```
GNU nano 6.2
server:
  - address: 127.0.0.5
    port: 9090
guami:
  - plmn_id:
      mcc: 001
      mnc: 01
    amf_id:
      region: 2
      set: 1
tai:
  - plmn_id:
      mcc: 001
      mnc: 01
    tac: 7
plmn_support:
  - plmn_id:
      mcc: 001
      mnc: 01
    s_nssai:
      - sst: 1
security:
  integrity_order: [ NIA2, NIA1, NIA0 ]
  ciphering_order: [ NEA0, NEA1, NEA2 ]
network_name:
  full: FL_NETWORK
  short: Next
amf_name: open5gs-amf0
time:
  # t3502:
  #   value: 720 # 12 minutes * 60 = 720 seconds
  t3512:
```

Figure 6 - fichier `amf.yml`

## Chapitre III : Système de transmission

```
GNU nano 6.2 amf.yaml
#
- uri: http://127.0.0.10:7777
scp:
- uri: http://127.0.0.200:7777
ngap:
  server:
    - address: 127.0.0.5
metrics:
  server:
    - address: 127.0.0.5
    port: 9090
guami:
  - plmn_id:
      mcc: 001
      mnc: 01
    amf_id:
      region: 2
      set: 1
  tai:
    - plmn_id:
        mcc: 001
        mnc: 01
      tac: 7
  plmn_support:
    - plmn_id:
        mcc: 001
        mnc: 01
      s_nssai:
        - sst: 1
security:
  integrity_order : [ NIA2, NIA1, NIA0 ]
  cipher_order : [ NEA0, NEA1, NEA2 ]
```

Figure 7 - extrait fichier amf.yaml

Puis, nous modifions le fichier upf.yaml en renseignant l'adresse IP et l'APN de notre réseau. Pour cela, il faut renseigner les informations suivantes :

- dnn: FLNETWORK

```
sudo nano /etc/open5gs/upf.yaml
```

```
GNU nano 6.2 upf.yaml
logger:
  file:
    path: /var/log/open5gs/upf.log
  # level: info # fatal/error/warn/info(default)/debug/trace
global:
  max:
    ue: 1024 # The number of UE can be increased depending on memory size.
    peer: 64
upf:
  pfcps:
    server:
      - address: 127.0.0.7
    client:
      - smf: # UPF PFCP Client try to associate SMF PFCP Server
        - address: 127.0.0.4
  gtpu:
    server:
      - address: 127.0.0.7
  session:
    - subnet: 10.45.0.0/16
      gateway: 10.45.0.1
      dnn: FLNETWORK
    - subnet: 2001:db8:cafe::/48
      gateway: 2001:db8:cafe::1
      dnn: FLNETWORK
  metrics:
    server:
      - address: 127.0.0.7
        port: 9090
```

Figure 8 - extrait du fichier upf.yaml

Le fichier smf.yaml nous permet de gérer l'attribution des DNS et du sous réseau de l'utilisateur.

```
GNU nano 6.2 smf.yml
upf:
  - address: 127.0.0.7
gtpc:
  server:
    - address: 127.0.0.4
gtpu:
  server:
    - address: 127.0.0.4
metrics:
  server:
    - address: 127.0.0.4
    port: 9090
session:
  - subnet: 10.45.0.0/16
    gateway: 10.45.0.1
  - subnet: 2001:db8:cafe::/48
    gateway: 2001:db8:cafe::1
dns:
  - 8.8.8.8
  - 8.8.4.4
  - 2001:4860:4860::8888
  - 2001:4860:4860::8844
mtu: 1400
# p-cscf:
#   - 127.0.0.1
#   - ::1
# ctcf:
#   enabled: auto # auto(default)/yes/no
freeDiameter: /etc/freeDiameter/smf.conf
#####
# SMF Info
#####
# <SMF Selection - 5G Core only>
```

Figure 9 - extrait du fichier smf.yml

Dans le fichier nrf.yml nous devons changer le PLMN ID de la même façon que dans le fichier amf.yml.

```
GNU nano 6.2 nrf.yml
logger:
  file:
    path: /var/log/opensgs/nrf.log
  # level: info # fatal/error/warn/info(default)/debug/trace
global:
  max:
    ue: 1024 # The number of UE can be increased depending on memory size.
    peer: 64
nrf:
  serving: # 5G roaming requires PLMN in NRF
    - plmn_id:
        mcc: 001
        mnc: 01
    sbi:
      server:
        - address: 127.0.0.10
          port: 7777
#####
# SBI Server
#####
# o Override SBI address to be advertised to NRF
# sbi:
#   server:
#     - dev: eth0
#       advertise: opensgs-nrf.svc.local
# sbi:
#   server:
#     - address: localhost
#       advertise:
#         - 127.0.0.99
```

Figure 10 - extrait du fichier nrf.yml

Le fichier gNB permet de configurer l'antenne et se connecte à l'AMF via le port 38412. Dans ce fichier, nous spécifions l'adresse et le port de l'AMF, ainsi que le type de liaison utilisé, qui est ici l'USRP B210.

Dans la rubrique cell\_cfg nous devons spécifier le PLMN ID, le TAC ainsi que la bande de fréquence utilisée.

```
GNU nano 6.2 gnb_rf_b200_tdd_n78_20mhz.yml
cu_cp:
  amf:
    addr: 127.0.0.5
    port: 38412
    bind_addr: 127.0.0.1
    supported_tracking_areas:
      - tac: 7
        plmn_list:
          - plmn: "00101"
            tal_slice_support_list:
              - sst: 1
ru_sdr:
  device_driver: uhd
  device_args: type=b200,num_recv_frames=64,num_send_frames=64
  srates: 23.04
  otw_format: sc12
  tx_gain: 90
  rx_gain: 70
cell_cfg:
  dl_arfcn: 026976
  band: 78
  channel_bandwidth_MHz: 20
  common_scs: 30
  plmn: "00101"
  tac: 7
  pci: 1
```

Figure 111 - extrait du fichier `gnb_rf_b200_tdd_n78_20mhz.yml`

Pour terminer, nous configurons le routage dans notre cœur 5G afin de le connecter à Internet, permettant ainsi à notre utilisateur d'accéder à Internet via une règle NAT pour son réseau.

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -A POSTROUTING -s 10.45.0.0/16 ! -o ogstun -j MASQUERADE
```

### Handover

En réseaux 5G, le "handover" (ou transfert) est un processus crucial qui permet de maintenir la continuité de la connexion lorsque l'utilisateur se déplace d'une carte à une autre.

Voici quelques points clés, du transfert entre deux gNB différents. Le gNB source initie le transfert et envoie une demande de handover au gNB cible. Le gNB cible effectue un contrôle d'admission et, si accepté, envoie la configuration RRC nécessaire au gNB source, qui la transmet ensuite à l'utilisateur pour accéder à la nouvelle cellule.

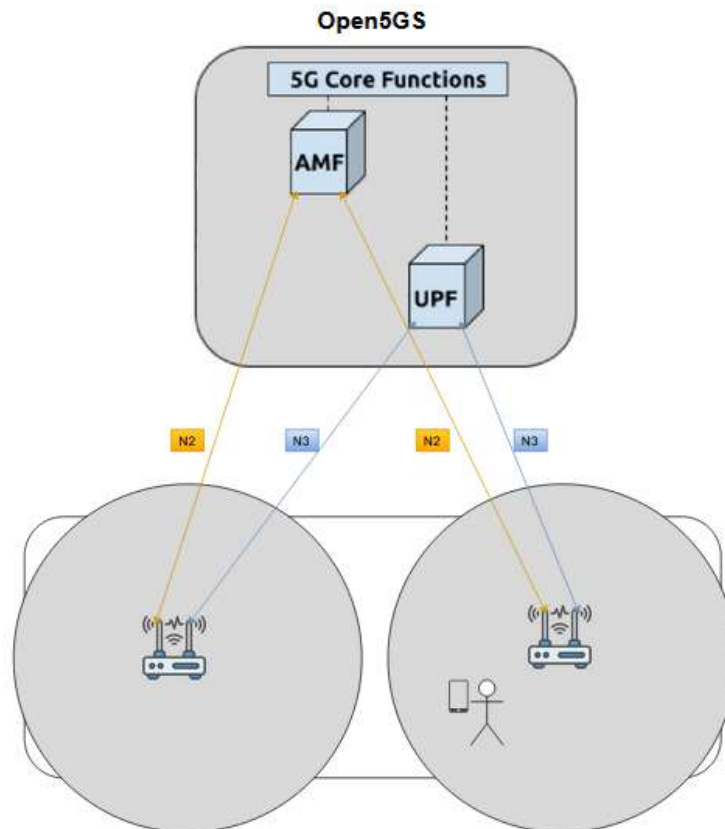


Figure 122 - schéma du handover

Dans notre cas, nous avons ajouté un gNB à notre cœur 5G, en utilisant la même configuration que le premier, mais avec une adresse IP différente spécifiée dans la rubrique :

```
Bind_addr = $Adresse IP différente
```

Ensuite, une fois le handover terminé, nous devons basculer entre les différentes cartes.

## Chapitre IV : Expérimentations et résultats

Pour notre expérimentation, nous avons utilisé un téléphone que nous avons connecté au cœur de notre réseau 5G. Cette connexion nous a permis de tester les capacités et la performance de notre infrastructure réseau dans des conditions réelles. Afin d'intégrer ce téléphone au réseau, nous avons configuré un nouveau point d'accès, que nous avons nommé "FLNETWORK". Cette configuration a été réalisée en utilisant les informations spécifiques du MCC et du MNC de notre carte SIM programmée.

- **Préparation du téléphone** : Nous avons d'abord vérifié que le téléphone avec la possibilité d'utiliser la 5G et qu'il disposait des dernières mises à jour logicielles.
- **Configuration du point d'accès** : Nous avons ajouté un nouveau point d'accès aux paramètres du téléphone. Ce point d'accès a été nommé "FLNETWORK".
- **Entrée des informations MCC et MNC** : Nous avons ensuite saisi les informations du MCC et du MNC, qui sont respectivement "001" et "01". Ces codes sont essentiels pour que le téléphone puisse se connecter correctement au réseau.
- **Vérification de la connexion** : Une fois le point d'accès configuré, nous avons redémarré les données mobiles 5G du téléphone, pour nous assurer de la bonne prise en compte des changements. Nous avons ensuite vérifié que le téléphone pouvait accéder à Internet via notre réseau 5G.

Les informations de configuration sont les suivantes :

- APN : "FLNETWORK"
- MCC : "001"
- MNC : "01"



Figure 133 – paramètres du point d'accès sur le téléphone

## Chapitre IV : Expérimentations et résultats

Ensuite, sur notre téléphone, nous avons utilisé une application appelée PingTools pour récupérer diverses informations réseau fournies par le cœur de notre réseau. Grâce à cette application, nous avons pu analyser et surveiller les performances et la connectivité de notre réseau 5G.

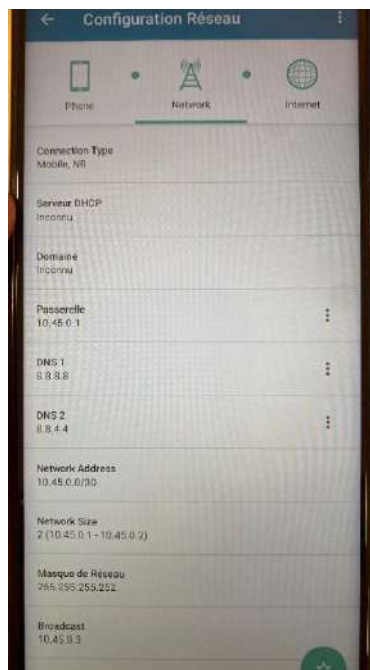


Figure 144 - configuration réseau du téléphone

Nous avons mis en place un handover pour tester le fonctionnement du transfert de l'utilisateur vers une autre antenne. Pour ce faire, nous avons réduit la puissance en dB de la réception et de l'émission afin de diminuer la portée. Les paramètres utilisés pour baisser le signal sont **tx\_gain** et **rx\_gain**.

```
ru_sdr:
  device_driver: uhd
  device_args: type=b200,num_recv_frames=64,num_send_frames=64
  srate: 23.04
  otw_format: sc12
  tx_gain: 90
  rx_gain: 70
```

Figure 155 - rubrique ru\_sdr dans le fichier gnb\_rf\_b200\_tdd\_n78\_20mhz.yml

Nos tests ont été effectués dans une salle de classe, nous nous sommes déplacés dans la salle afin de déterminer la portée de chaque antenne.

Ensuite, grâce à la visualisation des paquets (touche "T" sur le gNB), nous pouvons voir que lorsque nous passons d'une antenne à une autre, des paquets apparaissent sur l'antenne correspondant à l'UE.

## Chapitre IV : Expérimentations et résultats

```

etudiant@RT106-Master:~/srsRAN_Project/build/apps/gnb$ sudo ./gnb -c gnb_rf_b200_tdd_n78_20mhz.yml

--== srsRAN gNB (commit a041e3162) ==--

Lower PHY in quad executor mode.
Available radio types: uhd.
[INFO] [UHD] linux; GNU C++ version 11.4.0; Boost 107400; UHD_4.6.0.HEAD-0-g50fa3baa
[INFO] [LOGGING] Fastpath logging disabled at runtime.
Making USRP object with args 'type=b200,num_recv_frames=64,num_send_frames=64'
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Setting master clock rate selection to 'automatic'.
[INFO] [B200] Asking for clock rate 15.000000 MHz...
[INFO] [B200] Actually got clock rate 16.000000 MHz.
[INFO] [MULTI USRP] Setting master clock rate selection to 'manual'.
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
Cell pci=1, bw=20 MHz, IT1R, dl_arfcn=632628 (n78), dl_freq=3489.42 MHz, dl_ssb_arfcn=632256, ul_freq=3489.42 MHz

N2: Connection to AMF on 192.168.5.254:38412 completed
==== gNB started ====
Type <h> to view help
t

pci rnti |-----DL-----|-----UL-----|
 1 4601 | cqi ri mcs brate ok nok (%) dl_bs | pusch rsrp mcs brate ok nok (%) bsr ta phr
 1 4601 | 15 1.0 0 0 0 0 0% 0 | n/a n/a 0 0 0 0 0% 0 301n 25
 1 4601 | 15 1.0 0 0 0 0 0% 0 | n/a n/a 0 0 0 0 0% 0 323n 25
 1 4601 | 15 1.0 0 0 0 0 0% 0 | 34.9 -15.8 27 4.9k 1 0 0% 0 343n 18
 1 4601 | 15 1.0 0 0 0 0 0% 0 | n/a n/a 0 0 0 0 0% 0 310n 18
 1 4601 | 15 1.0 12 2.3k 5 0 0% 0 | 29.9 -21.0 27 9.1k 2 0 0% 0 338n 20
 1 4601 | 15 1.0 0 0 0 0 0% 0 | 32.1 -20.5 27 4.9k 1 0 0% 0 303n 16

```

Figure 166 – lancement de l'antenne et vérification du fonctionnement du handover



## Chapitre V : Gestion de projet

Pour l'organisation de notre projet, nous avons fait le choix d'utiliser la méthode agile Scrum, car celle-ci correspondait le mieux aux spécificités et aux exigences de notre projet.

Le matériel spécifique à ce projet n'étant accessible que sur place nous ne pouvions travailler que depuis l'IUT, et donc en flux continu, nous avons donc naturellement opté pour la méthode Scrum.

Cette méthode nous permettait en effet de sprinter efficacement lorsque nous nous retrouvions ensemble dans les salles de classe, avec tout le matériel nécessaire à notre disposition.

### Description de la méthode

Dans la méthode Scrum le travail est organisé en plusieurs sprints. Les sprints sont définis comme des périodes de travail courtes (1 à 4 semaines) au cours desquels l'équipe planifie, développe, teste et livre les éléments fonctionnels du produit. Chaque sprint sert à planifier, développer et tester des parties spécifiques du site, en intégrant des ajustements basés sur les retours obtenus. Cette méthodologie nous permettra de rester à jour face aux éventuels changements, tout en maintenant une vision claire de nos objectifs et en assurant un suivi constant de l'avancement du projet.

La méthode Scrum possède plusieurs rôles :

- Product Owner : Représente les parties prenantes et s'assure que l'équipe travaille sur les fonctionnalités apportant le plus de valeur.
- Scrum Master : Facilite le processus Scrum, élimine les obstacles et aide l'équipe à suivre les pratiques de Scrum.
- Équipe de développement : Les membres qui travaillent ensemble pour concevoir, coder, tester et livrer l'incrément du produit.

### Organisation des sprints

#### **Sprint 1 (Semaines 1 et 2)**

- Product Owner : Filipe
- Scrum Master : Aadil
- Équipe de développement : Coralie, Luc, Mehdi

#### **Objectif :**

L'objectif pour les deux prochaines semaines est de se documenter sur les informations nécessaires à la réalisation d'un réseau 5G. En parallèle, nous déployons les équipements, configurons les services d'Open5GS et effectuons des tests pour assurer une connectivité stable.

- **Fin de la première semaine :**
  - Documentation.
- **Fin de la deuxième semaine :**
  - Réalisation physique.
  - Tests.

### **Sprint 2 (Semaines 3 et 4)**

- Product Owner : Aadil
- Scrum Master : Filipe
- Équipe de développement : Coralie, Luc, Mehdi

### **Objectif :**

L'objectif pour les deux prochaines semaines est de réaliser le rapport et le poster du projet, ainsi que d'améliorer le projet en intégrant le Handover avec une autre équipe. Cela inclut la rédaction et la mise en forme du rapport, la création d'un poster, et la collaboration avec une autre équipe pour implémenter et tester le Handover afin d'améliorer notre projet.

- **Fin de la troisième semaine :**
  - Réalisation du rapport.
  - Réalisation du poster.
  - Réalisation du résumer en anglais.
- **Fin de la quatrième semaine :**
  - Amélioration du projet.

## Chapitre VI : Conclusion et retour d'expérience

Pour conclure, notre projet a permis de développer un cœur 5G capable de fournir une connectivité réseau à notre téléphone. Nous avons établi une communication entre notre antenne USRP B210 et notre téléphone en utilisant les différents services d'Open5GS, tels que l'AMF et l'UPF. Grâce à notre configuration, notre téléphone a pu obtenir une adresse IP et accéder à Internet via notre cœur 5G.

Ce projet a démontré la faisabilité technique de la mise en place d'un réseau 5G privé, ainsi que l'acquisition des compétences nécessaires pour configurer et gérer les composants réseau.

Durant ce projet, nous avons approfondi notre compréhension du fonctionnement d'un réseau 5G cœur en utilisant diverses applications. Cette expérience nous a permis d'explorer un domaine complexe, enrichissant ainsi notre savoir-faire technique et notre connaissance des réseaux de téléphonie mobile que nous utilisons quotidiennement.

Cette expérience a renforcé notre capacité à collaborer efficacement en équipe pour résoudre des problèmes complexes. Les connaissances et compétences acquises au cours de ce projet seront précieuses pour nos futurs projets et carrières dans le domaine des réseaux et des télécommunications.

De plus, cette réalisation ouvre la voie à des améliorations futures, notamment l'augmentation de la couverture réseau, l'amélioration de la qualité de service et l'intégration de nouvelles fonctionnalités comme le Handover sur des modèles d'antenne compatibles. En somme, ce projet représente une avancée significative vers la maîtrise des technologies 5G et leur application pratique dans des environnements réels.

En travaillant sur ce projet, nous avons acquis des compétences précieuses en matière de configuration et de gestion des composants réseau, ainsi qu'une meilleure compréhension des défis liés à la mise en œuvre de la 5G. Nous avons également abordé des aspects cruciaux tels que la sécurité des données, la gestion de la bande passante et la réduction de la latence.

## Annexe / Bibliographie

Installation et configuration d'open5GS :

<https://open5gs.org/open5gs/docs/guide/01-quickstart/>

Installation et configuration du GNB :

[https://docs.srsran.com/projects/project/en/latest/user\\_manuals/source/installation.html](https://docs.srsran.com/projects/project/en/latest/user_manuals/source/installation.html)

Installation Pysim :

<https://github.com/osmocom/pysim>

Documentation Pysim :

<https://osmocom.org/projects/pysim/wiki>