

RAPPORT

“TP APPLIQUE AU RESEAU 5G AUTOUR DE FREE5GC/DOCKER”

CONTENU DU DOCUMENT

Rapport du projet d'étude d'une mise en place d'une Architecture 5g pour l'entreprise «NextGenTelco »

SUIVI DES VERSIONS

| DATE | VERSION | ÉVOLUTION | RÉDACTEUR | APPROBATEUR | TYPE DE DIFFUSION |
|------------|---------|--------------------------|---------------------|---------------------|-------------------|
| 19/01/2024 | 1.0 | Plan | Rayane NAIT MERZOUG | Sylia BENBACHIR | Discord |
| 21/01/2024 | 1.1 | Installation | Sylia BENBACHIR | Rayane NAIT MERZOUG | Discord |
| 22/01/2024 | 1.2 | Déploiement & Conception | Rayane NAIT MERZOUG | Sylia BENBACHIR | Discord |
| 25/01/2024 | 1.3 | Déploiement & Conception | Sylia BENBACHIR | Rayane NAIT MERZOUG | Discord |
| 26/01/2024 | 2.0 | Depot | Rayane NAIT MERZOUG | Mme Suzanne Chan | Mail |

Sommaire

| | | |
|----------|---|-----------|
| 1 | Introduction | 2 |
| 2 | Installation : | 3 |
| 2.1 | Environnements | 3 |
| 2.2 | Mise en place de l'installation | 4 |
| 2.3 | Manipulation de l'environnement avec Docker – Prise en Main..... | 6 |
| 3 | Déploiement..... | 8 |
| 3.1 | Création du réseau..... | 8 |
| 3.2 | Multiplication UE & gNB | 9 |
| 3.3 | Identification des cas de panne et résolution..... | 13 |
| 4 | Conception..... | 15 |
| 4.1 | Situation Hand Over..... | 15 |
| 4.2 | Schéma applicatif décrivant les routes pour différents scénarios..... | 16 |
| 4.3 | Etude de faisabilité | 17 |
| 5 | Conclusion..... | 19 |
| 6 | Bibliographie..... | 20 |

1 Introduction

NextGenTelco, un nouvel acteur ambitieux dans le secteur des télécommunications, a de grandes aspirations en envisageant d'entrer sur le marché en exploitant la technologie émergente 5G. Pour concrétiser cette vision audacieuse, l'entreprise mène actuellement une étude approfondie pour évaluer la faisabilité de mettre en place la solution open source "free5gc" en mode Docker, reconnue pour sa légèreté, sa rapidité et sa simplicité.

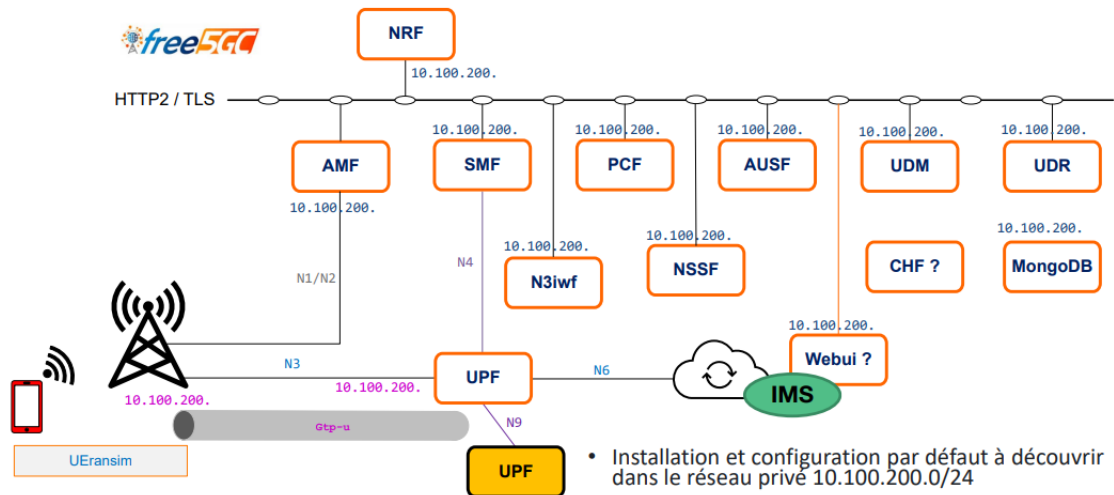
Dans le cadre de cette démarche stratégique, la société a mis en ligne une page détaillant ses tarifications et ses offres d'abonnements.

Cette approche stratégique vise à accélérer considérablement le déploiement de l'infrastructure 5G, permettant ainsi à NextGenTelco de se démarquer par une mise en service rapide et efficace. L'étude en cours est principalement axée sur l'adaptation de free5gc au réseau privé de NextGenTelco, ouvrant ainsi la voie à une nouvelle ère de connectivité et de services de télécommunications de pointe.

2 Installation :

Tout d'abord nous avons commencé par installer Core 5G disponible en open source sur GitHub nommé free5GC qui sera à mettre en place sur Docker.

Ayant comme architecture la suivante :



2.1 Environnements

2.1.1 Environnements

Free5GC crée un environnement spécifique permettant le déploiement complet d'un réseau 5G, en intégrant divers éléments pour offrir une connectivité avancée.

Système

Free5GC s'intègre harmonieusement dans l'écosystème Linux, exploitant ses fonctionnalités pour fournir une infrastructure réseau 5G complète.

2.1.2 Prérequis

GIT

GIT, en tant que système de gestion de version décentralisé, joue un rôle crucial dans le développement, la collaboration et le suivi des modifications du code source de Free5GC.

Docker

Docker, une plateforme de conteneurisation, est utilisé pour isoler l'environnement d'exécution de Free5GC, garantissant ainsi la portabilité et la cohérence sur différentes plateformes.

2.2 Mise en place de l'installation

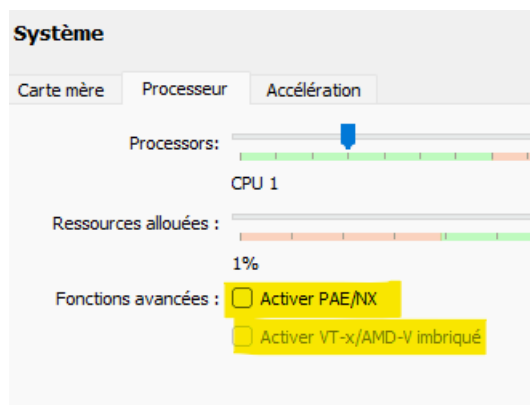
2.2.1 Contexte

L'installation de Free5GC, une suite de composants pour le déploiement d'un réseau 5G, a présente des défis. Les principaux problèmes ont été identifiés et résolus pour assurer le bon fonctionnement de Free5GC dans l'environnement spécifique.

2.2.2 Problèmes rencontrés

Virtualisation sur une machine virtuelle

L'utilisation de docker compose n'as pas été possible car dans les paramètres de boot les paramètre de virtualisation n'était pas activés.



Dépendances manquantes

L'installation initiale a échoué en raison de dépendances manquantes, notamment des bibliothèques et des paquets logiciels essentiels. Certains composants de Free5GC requièrent des versions spécifiques de ces dépendances.

Incompatibilité de version du noyau

L'installation initiale a échoué en raison de l'incompatibilité entre la version actuelle du noyau (5.10.0-13-amd64) et les dépendances du module GTP5G.

Dépendance manquante - tunnel UDP

Le module GTP5G dépend d'un autre module noyau appelé "tunnel UDP". L'absence de ce module a empêché l'installation réussie du module GTP5G.

2.2.3 Résolution des problèmes

Activer la virtualisation

Accéder au cmd Windows en tant qu'administrateur au dossier VirtualBox et exécuter la commande suivante :

```
C:\Program Files\Oracle\VirtualBox>VBoxManage modifyvm MonProjet --nested-hw-virt on
```

Mise à jour du noyau

La solution proposée pour résoudre l'incompatibilité de version a été d'effectuer une mise à jour du noyau vers la version compatible (5.10.0-26-amd64).

```
1 apt install linux-image-5.10.0-26-amd64
2 reboot
```

Compilation et Installation du Module GTP5G

Suite à la mise à jour, le module GTP5G a été compilé et installé avec succès.

```
1 make -C /lib/modules/5.10.0-26-amd64/build M=/[projet_PATH]/gtp5g modules
2 make install
3 modprobe udp_tunnel
4 /sbin/depmod -a
5 modprobe gtp5g
6 echo "gtp5g" >> /etc/modules
```

Vérification et Informations Finales

Après la résolution des problèmes et l'installation réussie, les informations finales des modules ont été vérifiées.

| | | | | |
|----|--|--|-------------------------------|---------|
| 1 | sylia@ubuntu:~/projets/free5gc-compose\$ | docker ps | | |
| 2 | CONTAINER ID | IMAGE | COMMAND | CREATED |
| | STATUS | PORTS | NAMES | |
| 3 | dff8559368e9 | free5gc/upf:v3.3.0 | "bash -c './upf-i p t a '" | 2 weeks |
| | ago Up 2 weeks | | upf | |
| 4 | fb0c47240bbf | free5gc/ueransim:latest | "./nr-gnb -c ./ c o n f i '" | 2 weeks |
| | ago Up 2 weeks | | ueransim_host | |
| 5 | bae1124969cc | free5gc/n3iwf:v3.3.0 | "sh -c './n3iwf-i p s e '" | 2 weeks |
| | ago Up 2 weeks | | n3iwf | |
| 6 | b571e8a80677 | free5gc/ueransim:latest | "./nr-gnb -c ./ c o n f i '" | 2 weeks |
| | ago Up 2 weeks | | ueransim | |
| 7 | ca4d41493825 | free5gc/chf | "./chf -c ./ config/ c '" | 2 weeks |
| | ago Up 2 weeks | 8000/tcp | chf | |
| 8 | 6c3744a14df3 | free5gc/pcf:v3.3.0 | "./pcf -c ./ config/ p '" | 2 weeks |
| | ago Up 2 weeks | 8000/tcp | pcf | |
| 9 | 77b79588503e | free5gc/ausf:v3.3.0 | "./ausf -c ./ config/ '" | 2 weeks |
| | ago Up 2 weeks | 8000/tcp | ausf | |
| 10 | 079a0d4b404a | free5gc/nssf:v3.3.0 | "./nssf -c ./ config/ '" | 2 weeks |
| | ago Up 2 weeks | 8000/tcp | nssf | |
| 11 | 4a4af8f7fa71 | free5gc/amf:v3.3.0 | "./amf -c ./ config/ a '" | 2 weeks |
| | ago Up 2 weeks | 8000/tcp | amf | |
| 12 | 5683cd95a17e | free5gc/udr:v3.3.0 | "./udr -c ./ config/ u '" | 2 weeks |
| | ago Up 2 weeks | 8000/tcp | udr | |
| 13 | 355d932e7fa9 | free5gc/udm:v3.3.0 | "./udm -c ./ config/ u '" | 2 weeks |
| | ago Up 2 weeks | 8000/tcp | udm | |
| 14 | al15e3482d1b | free5gc/smf:v3.3.0 | "./smf -c ./ config/ s '" | 2 weeks |
| | ago Up 2 weeks | 8000/tcp | smf | |
| 15 | aa4711012c35 | free5gc/webui:v3.3.0 | "./webui -c ./ c o n f i g '" | 2 weeks |
| | ago Up 2 weeks | 0.0.0.0:2121-2122->2121-2122/tcp, :::2121-2122->2121-2122/tcp, 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp | webui | |
| 16 | c49e9f0aeea9 | free5gc/nrf:v3.3.0 | "./nrf -c ./ config/ n '" | 2 weeks |
| | ago Up 2 weeks | 8000/tcp | nrf | |
| 17 | 2e0c896d858d | mongo | "docker-entrypoint. s '" | 2 weeks |
| | ago Up 2 weeks | 27017/tcp | mongodb | |

2.3 Manipulation de l'environnement avec Docker – Prise en Main

L'utilisation de Docker pour déployer Free5GC offre une approche flexible et isolée. Ce Rapport détaille les étapes pour manipuler l'environnement Free5GC à l'aide de conteneurs Docker, couvrant le démarrage, l'arrêt, et la gestion des services.

2.3.1 Manipulation des conteneur Docker

Démarrage des Conteneurs

Démarrer les conteneurs Free5GC en utilisant la commande

```
1 sylvia@ubuntu:~/projet/free5gc-compose$ sudo docker-compose -f docker-  
compose.yaml up && sudo docker-compose up
```

Arrêt des conteneurs

Arrêter les conteneurs Free5GC en utilisant la commande

```
1 sylvia@ubuntu:~/projet/free5gc-compose$ docker stop [container_id]
```

2.3.2 Gestion des Services Free6GC

Vérification des services

Utiliser la commande docker ps pour vérifier l'état des conteneurs Free5GC

```
1 sylvia@ubuntu:~/projet/free5gc-compose$ docker ps
```

Accès aux Journaux des Services

Pour consulter les journaux des services spécifiques, utilisez la commande

```
1 sylvia@ubuntu:~/projet/free5gc-compose$ docker logs upf
2 2024-01-05T14:39:49.403317593Z [INFO][UPF][Main] UPF version:
3 free5GC version: v3.3.0
4 build time: 2023-06-08T16:41:36Z
5 commit hash: 4474dc86
6 commit time: 2023-06-08T03:37:39Z
7 go version: go1.17.13 linux/amd64
8 2024-01-05T14:39:49.406710262Z [INFO][UPF][CFG] Read config from [./config/  
upfcfg.yaml]
9 2024-01-05T14:39:49.418872201Z [INFO][UPF][CFG]
10 2024-01-05T14:39:49.418918825Z [INFO][UPF][CFG] (*factory.Config)(0  
xc0003e27d0){
11   Version: (string) (len=5) "1.0.3",
12   Description: (string) (len=31) "UPF initial local configuration",
13   Pfcfg: (*factory.Pfcfg)(0xc0004695f0){
14     Addr: (string) (len=15) "upf.free5gc.org",
15     NodeID: (string) (len=15) "upf.free5gc.org",
16     RetransTimeout: (time.Duration) 1s,
17     MaxRetrans: (uint8) 3
18   },
19   Gtpu: (*factory.Gtpu)(0xc0004696e0){
20     Forwarder: (string) (len=5) "gtp5g",
21     IfList: ([]factory.IfInfo) (len=1 cap=1) {
22       (factory.IfInfo) {
23         Addr: (string) (len=15) "upf.free5gc.org",
24         Type: (string) (len=2) "N3",
25         Name: (string) "",
26         IfName: (string) "",
27         MTU: (uint32) 0
28       }
29     }
30   }
31   .
32   .
```


3 Déploiement

3.1 Création du réseau

Dans la phase de création de notre réseau 5G, nous avons commencé par examiner le réseau par défaut, tel qu'illustré dans la capture d'écran initiale.

```
networks:
  privnet:
    ipam:
      driver: default
      config:
        - subnet: 10.100.200.0/24
    driver_opts:
      com.docker.network.bridge.name: br-free5gc
  n6net:
    ipam:
      driver: default
      config:
        - subnet: 10.100.201.0/24
    driver_opts:
      com.docker.network.bridge.name: 2-br-free5gc
volumes:
  dbdata:
```

Ensuite, nous avons procédé à des modifications importantes dans le fichier de configuration YAML. Notre objectif principal était de mettre en place de nouvelles adresses réseau, et pour ce faire, nous avons configuré la plage d'adresses IP en utilisant l'adresse 127.5.10.0/24, comme le montre clairement la deuxième image que nous avons jointe. Ces ajustements ont été cruciaux pour mettre en place un réseau 5G personnalisé et adapté à nos besoins spécifiques.

```
networks:
  privnet:
    ipam:
      driver: default
      config:
        - subnet: 127.5.10.0/24
    driver_opts:
      com.docker.network.bridge.name: br-free5gc
  n6net:
    ipam:
      driver: default
      config:
        - subnet: 127.5.10.1/24
    driver_opts:
      com.docker.network.bridge.name: 2-br-free5gc
volumes:
  dbdata:
```

3.2 Multiplication UE & gNB

Dans ce document, nous décrivons les procédures que nous avons suivies pour multiplier les équipements utilisateur (UE) et les nœuds B de nouvelle génération (gNB) dans le core 5G de Free5GC. En utilisant UERANSIM et Docker, nous avons pu tester la capacité et les performances du réseau 5G de Free5GC dans des scénarios à grande échelle.

3.2.1 Multiplication UE

Étape 1 : Accès à UERANSIM

Nous avons ouvert un terminal et exécuté la commande : `docker exec -it ueransim bash`

Étape 2 : Modification de la Configuration de l'UE

Nous nous sommes dirigés vers le dossier de configuration : `cd config`

Nous avons ouvert le fichier de configuration de l'UE : `nano uecfg.yaml`

Nous avons modifié l'IMSI et d'autres paramètres nécessaires.

```
GNU nano 5.4                                uecfg.yaml *
# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits)
supi: 'imsi-208230000000003'
# Mobile Country Code value of HPLMN
mcc: '208'
# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '23'

# Permanent subscription key
key: '8baf473f2f8fd09487cccbd7097c6862'
# Operator code (OP or OPC) of the UE
op: '8e27b6af0e692e750f32667a3b14605d'
# This value specifies the OP type and it can be either 'OP' or 'OPC'
opType: 'OPC'
# Authentication Management Field (AMF) value
amf: '8000'
# IMEI number of the device. It is used if no SUPI is provided
imei: '356938035643803'
# IMEISV number of the device. It is used if no SUPI and IMEI is provided
imeiSv: '4370816125816151'

# List of gNB IP addresses for Radio Link Simulation
gnbSearchList:
  - 127.0.0.1

# UAC Access Identities Configuration
uacAic:
  mps: false
  mcs: false

# UAC Access Control Class
```

Étape 3 : Exécution de l'UE

Nous avons lancé l'UE avec la commande : `./nr-ue -c config/uecfg.yaml`

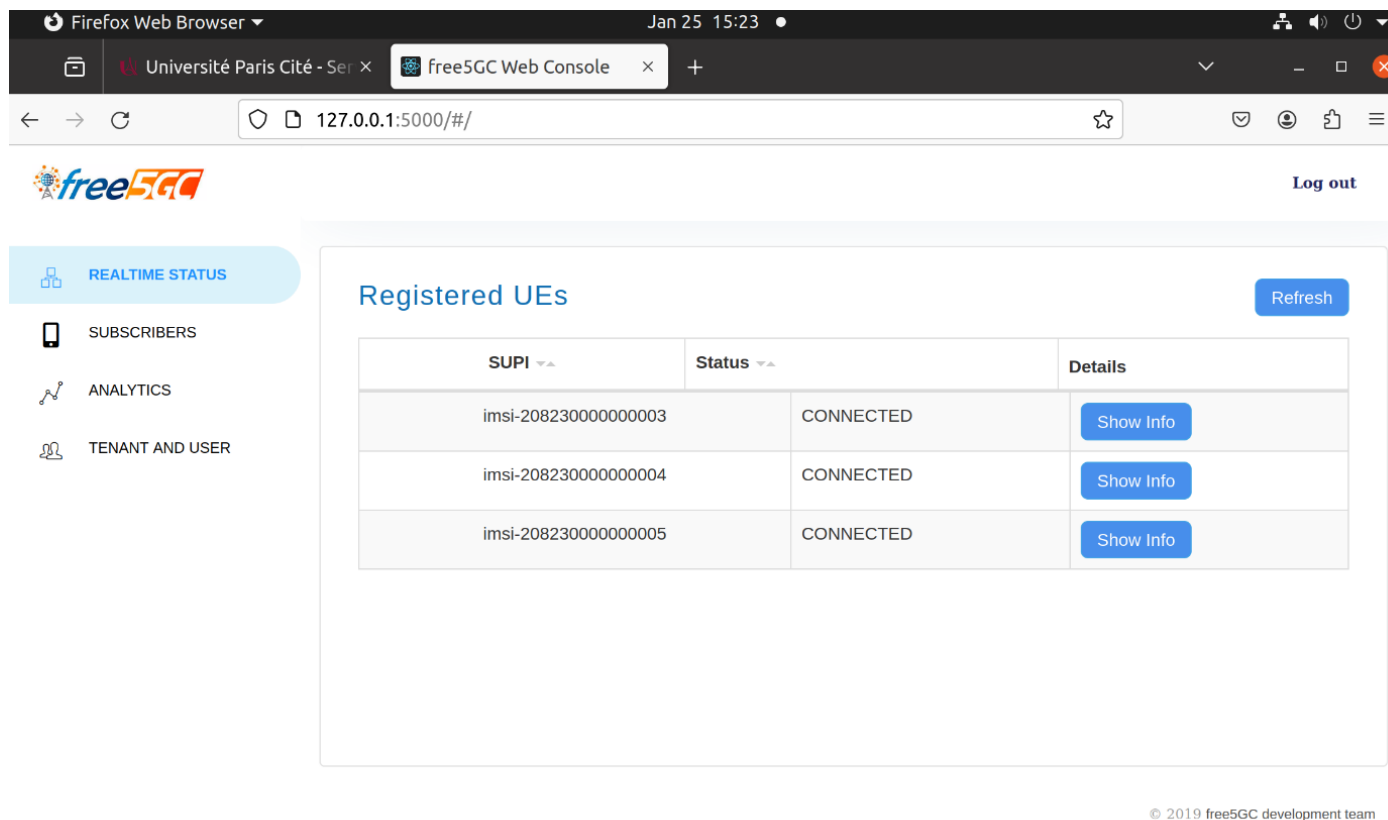
Étape 4 : Dépannage

En cas de problème, nous avons utilisé la commande : `./nr-cli --dump` pour le diagnostic.

Étape 5 : Vérification

Sur l'interface web 127.0.0.1 :5000

Nous vérifions que les UE ont bien été enregistrés sur notre réseau avec comme MNC : 23



The screenshot shows the free5GC Web Console interface in a Firefox browser. The address bar shows the URL `127.0.0.1:5000/#/`. The interface includes a sidebar with navigation options: REALTIME STATUS (selected), SUBSCRIBERS, ANALYTICS, and TENANT AND USER. The main content area displays the 'Registered UEs' section, which contains a table with the following data:

| SUPI | Status | Details |
|----------------------|-----------|---------------------------|
| imsi-208230000000003 | CONNECTED | Show Info |
| imsi-208230000000004 | CONNECTED | Show Info |
| imsi-208230000000005 | CONNECTED | Show Info |

A 'Refresh' button is located in the top right corner of the 'Registered UEs' section. The footer of the interface indicates '© 2019 free5GC development team'.



3.2.2 Multiplication gNB

Étape 1 : Création du fichier de configuration gNB

Nous avons préparé un fichier de configuration YAML pour chaque gNB comme on peut le voir la capture ci-dessous.

```
GNU nano 4.8                                     tex.yaml
version: '3'
services:
  ueransim_host:
    container_name: ueransim_host
    image: free5gc/ueransim:latest
    command: ./nr-gnb -c ./config/gnbcfg.yaml
    volumes:
      - ./config/gnb_host.yaml:/ueransim/config/gnbcfg.yaml
      - ./config/uecfg.yaml:/ueransim/config/uecfg.yaml
    cap_add:
      - NET_ADMIN
    devices:
      - "/dev/net/tun"
    network_mode: host
```

Étape 2 : Modification de docker-compose.yaml

Dans le fichier docker-compose.yaml, sous la section networks, nous avons ajouté le sous-réseau 127.5.10.1/24 (l'image jointe fournit les détails).

Étape 3 : Attribution des Adresses IP

Nous avons attribué des adresses IP uniques à chaque gNB dans le sous-réseau spécifié.

Étape 4 : Configuration du Réseau

Pour prendre en charge les gNB supplémentaires, nous avons configuré notre réseau en ajoutant les routes nécessaires. (Nous spécifions les routes à ajouter selon notre configuration réseau).



Étape 5 : Redémarrage de Free5GC

Après la configuration, nous avons redémarré le core 5G de Free5GC pour appliquer les modifications.

Étape 6 : Test de Connexion

Nous avons testé la connexion des gNB et des UE multipliés pour vérifier leur fonctionnement correct dans le réseau.

Après de tests de ping, malheureusement notre gNB est Unreachable !!

Nous avons essayé d'activer le Forwarding pour autoriser le transit des paquets IPv4 avec la commande :

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Mais cela n'a pas fonctionner.

3.3 Identification des cas de panne et résolution

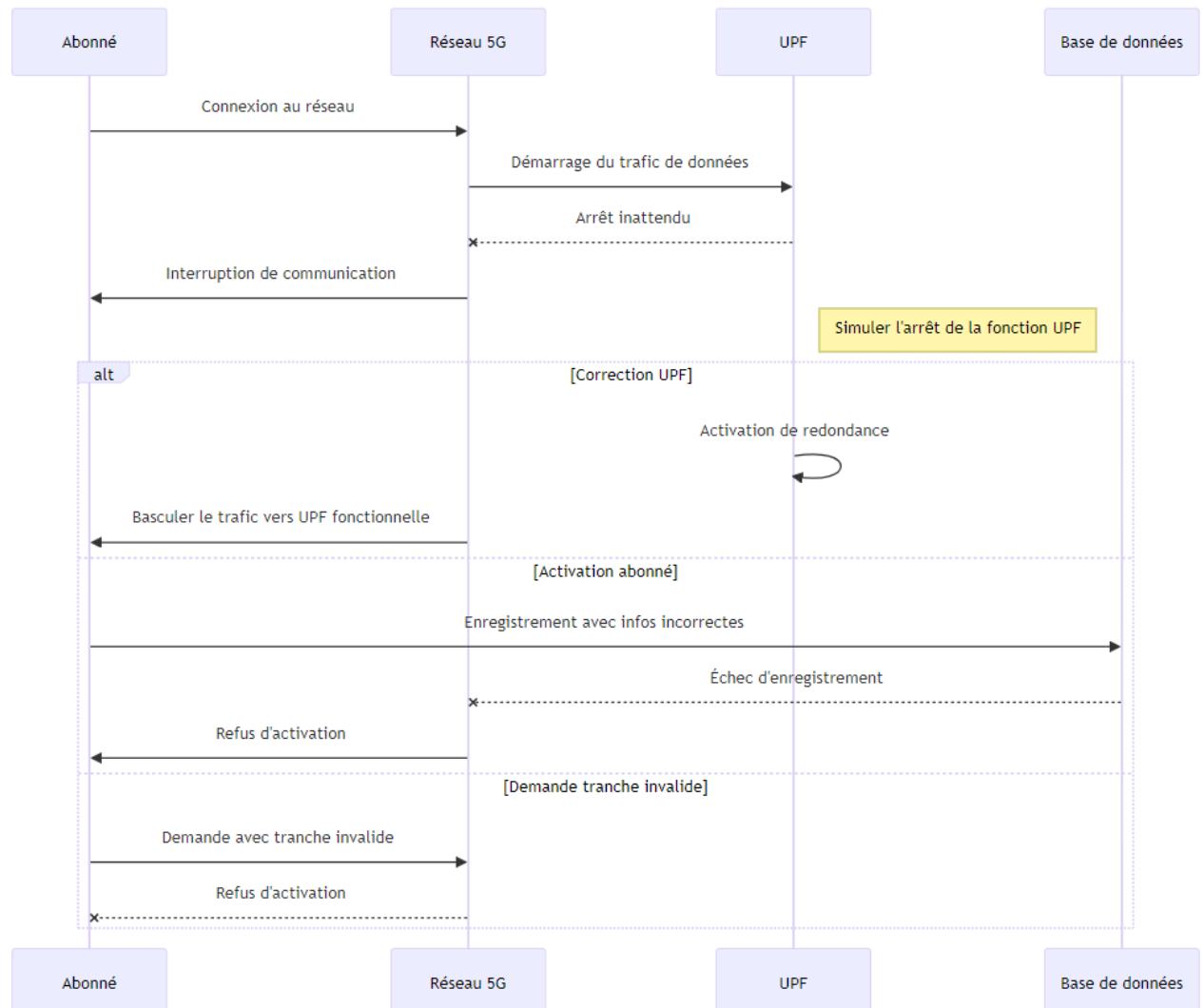
Scénario 1 : Arrêt de la fonction UPF (User Plane Function)

Pour ce scénario, nous devons simuler l'arrêt inattendu de la fonction UPF, ce qui peut entraîner une interruption de la communication pour nos abonnés. Voici comment nous pouvons le mettre en place :

Nous configurons des abonnés pour qu'ils se connectent au réseau.

Nous démarrons le trafic de données entre les abonnés.

Nous simulons l'arrêt de la fonction UPF, peut-être en éteignant le serveur qui l'exécute.



Correction du scénario 1 :

Pour corriger ce problème, nous devons mettre en place des mécanismes de tolérance aux pannes dans notre infrastructure 5G, tels que la redondance des fonctions UPF. En cas de défaillance d'une UPF, le trafic doit être basculé vers une autre UPF fonctionnelle pour éviter une interruption de service.

Scénario 2 : Activation d'un abonné avec une base non provisionnée

Pour ce scénario, nous devons simuler l'activation d'un abonné dont les informations ne sont pas correctement provisionnées dans notre système. Voici comment nous pouvons le mettre en place :

Nous configurons un abonné avec des informations incorrectes ou manquantes dans notre base de données d'abonnés.

Nous tentons de faire enregistrer cet abonné sur le réseau 5G.

Correction du scénario 2 :

Pour corriger ce problème, nous devons mettre en place des contrôles de validation et de provisionnement robustes dans notre système. Nous devons nous assurer que toutes les informations requises sont correctement renseignées avant d'autoriser l'activation d'un abonné. Si des informations sont incorrectes ou manquantes, l'activation doit être refusée, et un message d'erreur approprié doit être renvoyé à l'abonné.

Scénario 3 : Activation d'un abonné avec une tranche demandée invalide

Pour ce scénario, nous devons simuler une demande d'activation d'un abonné avec une tranche de réseau 5G invalide. Voici comment nous pouvons le mettre en place :

Nous configurons un abonné pour demander l'utilisation d'une tranche de réseau qui n'existe pas ou qui est mal configurée dans notre infrastructure.

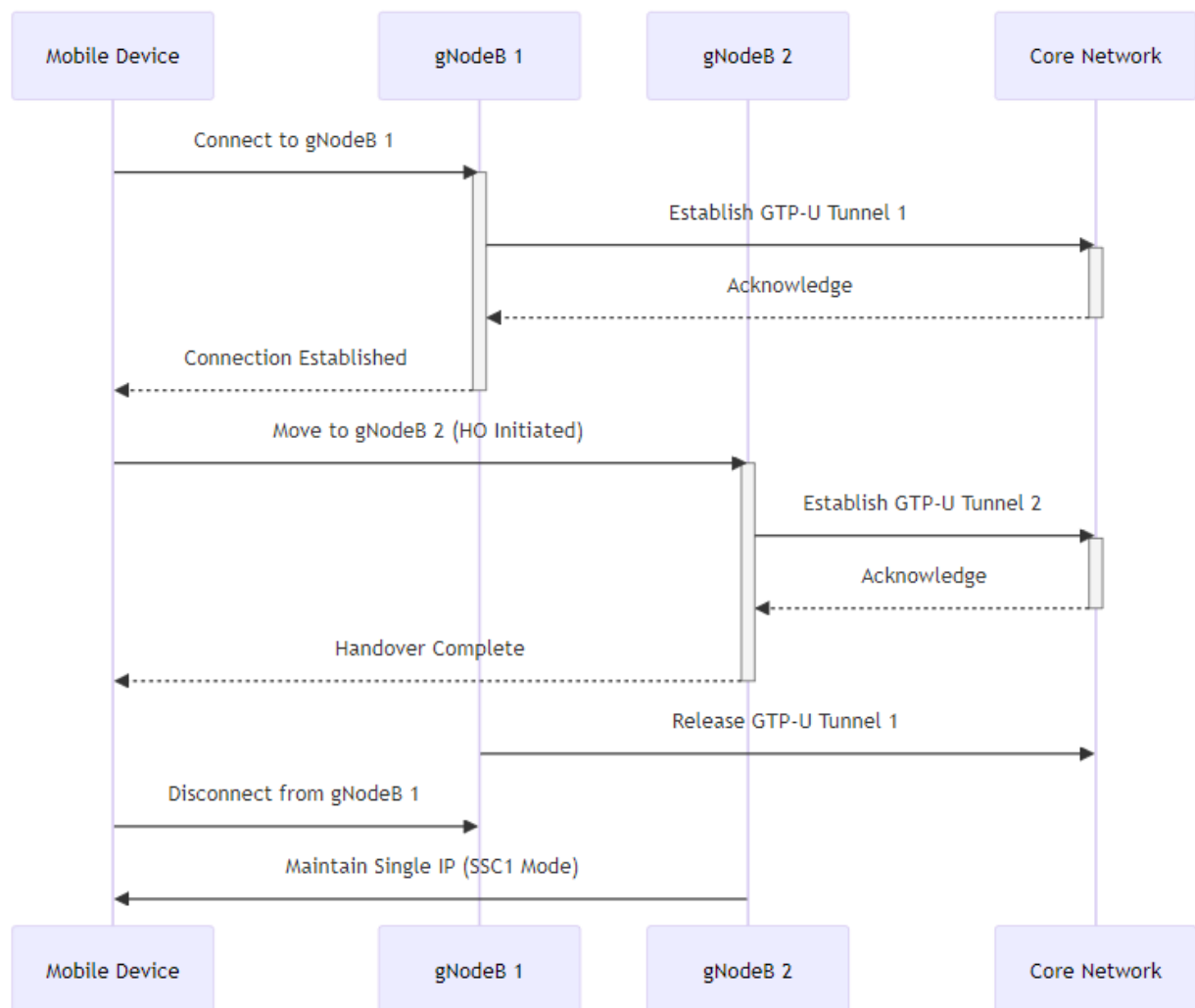
Correction du scénario 3 :

Pour corriger ce problème, nous devons mettre en place une vérification des tranches de réseau lors de l'activation des abonnés. Si la tranche demandée est invalide, notre système doit refuser la demande d'activation et renvoyer un message d'erreur indiquant que la tranche spécifiée n'est pas disponible ou qu'elle est incorrecte. Nous devons également nous assurer de disposer de mécanismes de gestion de configuration pour gérer correctement les tranches de réseau disponibles.

4 Conception

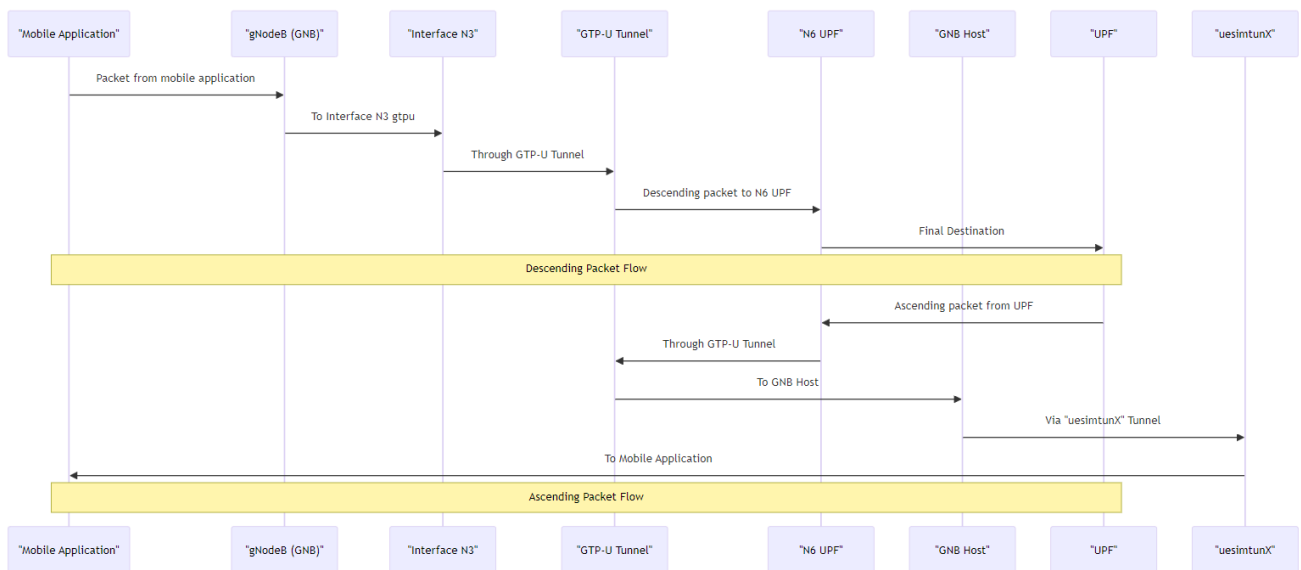
4.1 Situation Hand Over

Voici le diagramme de séquence illustrant le processus de transfert dans un réseau mobile, comprenant la création et la libération de tunnels GTP-U, ainsi que la maintenance d'une seule adresse IP en mode SSC1 :



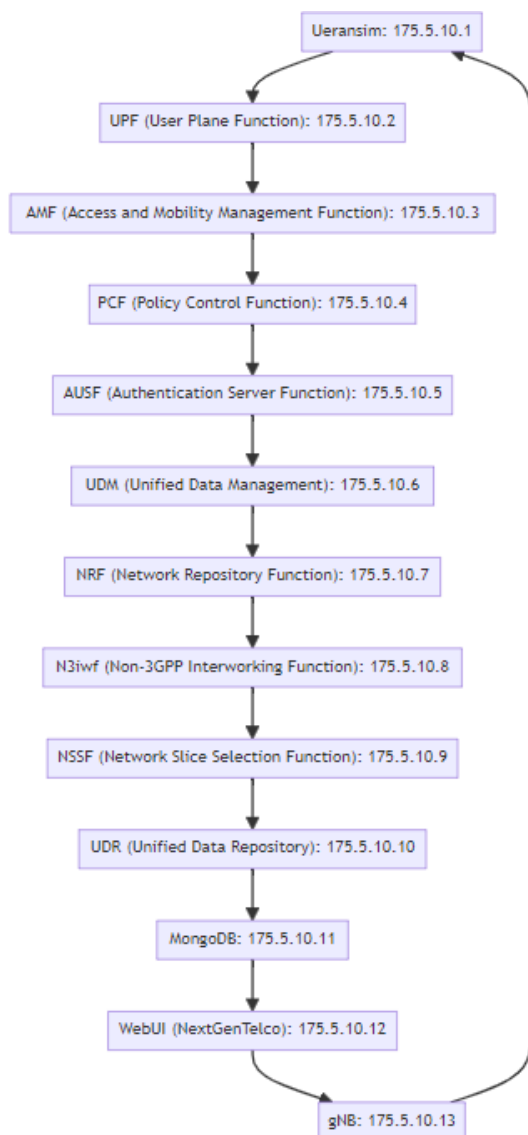
4.2 Schéma applicatif décrivant les routes pour différents scénarios

Voici le diagramme de séquence illustrant le chemin des paquets dans un réseau mobile, comprenant les paquets descendants vers l'UPF N6 et les paquets ascendants de l'application mobile vers l'interface N3, mettant en évidence les tunnels GTP-U et "uesimtunX" ainsi que les routes au niveau de l'hôte GNB et de l'UPF :

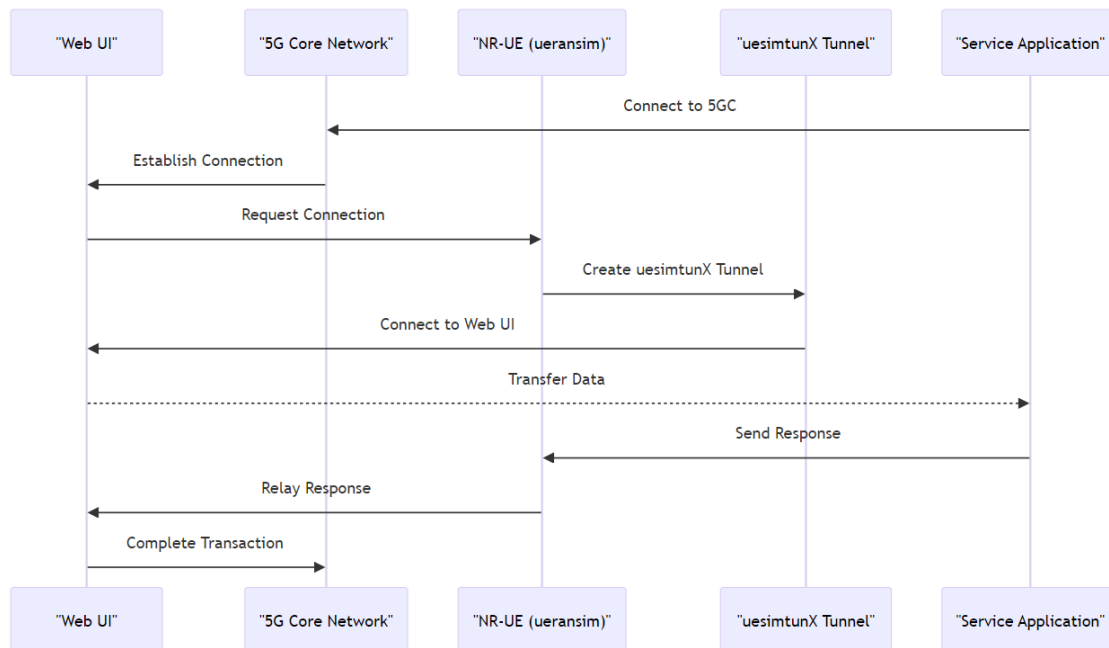


4.3 Etude de faisabilité

Tout d'abord voici la logique de routage pour l'établissement de la connexion afin que l'utilisateur puisse accéder au site Web NextGenTelCo.



Une fois donc que la route est ajoutée voici diagramme de séquence mettant en œuvre la faisabilité de la connexion entre les deux extrémités.



5 Conclusion

Notre projet sur l'intégration de Free5GC et Docker dans un réseau 5G est à la fois un défi complexe et une innovation. Il implique l'installation et la configuration de Free5GC sur Docker, la mise en place et la gestion d'un réseau 5G, l'incorporation de multiples équipements utilisateurs (UE) et de nœuds B de nouvelle génération (gNB), ainsi que l'identification et la résolution de divers scénarios de défaillance. Ce travail met en avant nos compétences techniques, notamment dans l'adaptation des environnements de virtualisation, la gestion des conteneurs Docker, et la configuration de réseau spécifique. Nous avons soigneusement documenté chaque étape, élaboré des diagrammes de séquence, et mené des études de faisabilité, démontrant ainsi notre approche méthodique et approfondie. Ce projet témoigne de notre compréhension approfondie des technologies 5G et des outils de virtualisation, ainsi que de notre aptitude à résoudre des problèmes complexes dans des environnements de réseau avancés. Il contribue de manière significative à la mise en œuvre et à l'optimisation des réseaux 5G.

6 Bibliographie

| RÉFÉRENCE DOCUMENTAIRE | DESCRIPTION | DOCUMENT (FORMAT ÉLECTRONIQUE) |
|-------------------------|--|---|
| INSTALLATION | | |
| Fichier PDF | Fichier explicatif mis à disposition par l'encadrant | M2rsa.pdf |
| Fichier PDF | Fichier d'installation mis à disposition par l'encadrant | Mode_OP_v0.1.pdf |
| RESOLUTION DES PROBLEMS | | |
| Site web | Forum de résolution des problèmes Free5GC | https://free5gc.org/ |

Table 1 - Références documentaires