

## Systèmes ubiquitaires (Edge, Vehicular et UAV)

### **Rapport de projet :**

*« Application du réseau 5G avec Free5gc/Docker »*

**Réalisée par :**

**Ibrahima DRAME**

**Idriss KOURBANHOUSSEN**

**Encadrant : Suzanne CHAN**

# **SOMMAIRE**

## **I. Introduction**

## **II. Installation Free5gc**

## **III. Déploiement de Free5gc**

1. Mise en route du réseau avec les configurations par défaut
2. Multiplier les UE
3. Multiplier des GnodeB
4. Multiplier le Network
5. Analyses des traces et protocoles
6. Routage

## **IV. Conception**

1. Schéma de Hand Over
2. Schéma de parcours d'un paquet

## **V. Conclusion**

## **I. Introduction**

Bienvenue chez **“Idriss et Ibrahima Télécom”**, votre nouveau partenaire innovant dans le domaine des télécommunications. En tant que nouvel acteur sur le marché, notre objectif est d'explorer les opportunités offertes par la solution open source avant-gardiste "free5gc" en mode Docker. La légèreté, la rapidité et la simplicité de cette infrastructure constituent des atouts cruciaux pour répondre aux besoins dynamiques du secteur des télécommunications.

Notre démarche consiste à évaluer la faisabilité de l'intégration de la solution free5gc au sein de notre réseau privé. Nous avons choisi de déployer cette infrastructure sur notre propre réseau, en l'adaptant à notre environnement spécifique. De plus, nous effectuerons des ajustements essentiels, tels que le changement du code MNC opérateur, pour refléter notre identité distinctive sur le marché.

Au cours de cette exploration, nous examinerons de près les différentes étapes du processus, en mettant en avant la rapidité de déploiement de l'infrastructure free5gc. À travers des captures d'écran et des exemples concrets, nous montrerons comment ces adaptations personnalisées s'inscrivent dans notre quête d'efficacité opérationnelle et de performance accrue.

Rejoignez-nous dans cette aventure novatrice où **“Idriss et Ibrahima Télécom”** aspire à redéfinir les normes de l'industrie des télécommunications en s'appuyant sur la puissance et la flexibilité de la solution free5gc.

## **II. Installation de free 5gc**

Free5GC est un projet open source issu de la 3GPP, une collaboration entre plusieurs organismes de normalisation œuvrant à définir les spécifications des technologies mobiles, dont la 5G.



Conçu pour être léger, rapide et flexible, "free5GC" offre une solution alternative open source pour les opérateurs de télécommunications et d'autres acteurs du secteur souhaitant déployer une infrastructure 5G. Sa facilité d'intégration résulte de sa conception, permettant une personnalisation aisée des composants du réseau 5G selon les besoins spécifiques.

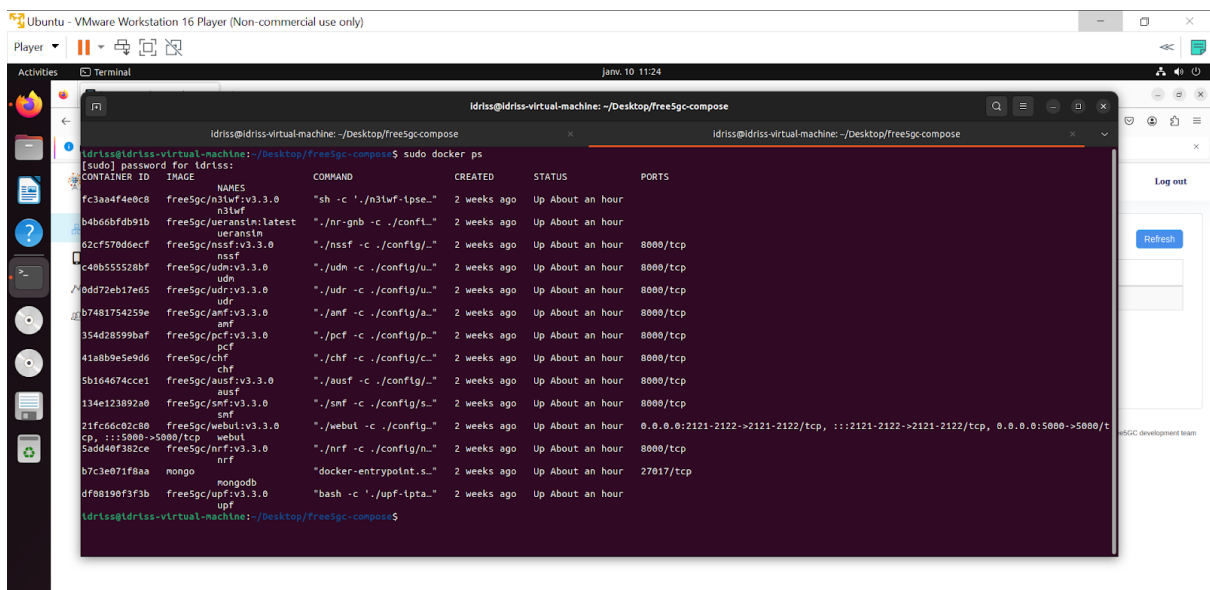
L'utilisation de Docker, un système de conteneurisation, simplifie le déploiement en isolant les différents composants de l'infrastructure. Cela contribue à rendre le processus plus efficace.

L'objectif principal de free5GC est de fournir une solution 5G complète, englobant des composants tels que l'UE, la GnodeB, UPF, AUSF, et autres. La nature open source du projet permet une configuration, modification et personnalisation totale du réseau 5G en fonction des exigences spécifiques, un aspect que nous explorerons tout au long de ce rapport.

L'installation de free5GC s'est effectuée via le référentiel GitHub <https://github.com/free5gc/free5gc-compose>. Les prérequis comprenaient l'installation de git, Docker, ainsi que gtp5. Nous avons opté pour une VM Ubuntu 22.4.2, privilégiée pour sa compatibilité et sa performance supérieure par rapport à d'autres systèmes d'exploitation lors du fonctionnement de free5GC.

Le seul souci rencontré était qu'il fallait que la machine virtuelle sur laquelle nous voulions installer Free5gc est qu'il fallait la présence du **package linux-headers-'uname -r'**, c'est pour cela que nous avons choisis de déployer sur Ubuntu.

```
ldriss@ldriss-virtual-machine:~/Desktop/free5gc-compose$ sudo docker images
[sudo] password for ldriss:
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
free5gc/smf-base    latest          dfd183167ada   About a minute ago   25.4MB
free5gc/pcf-base    latest          d3afbfbba94    12 minutes ago      29.2MB
free5gc/nssf-base   latest          a460db8c66e5   16 minutes ago      22.7MB
free5gc/nrf-base    latest          ad9a36141844   18 minutes ago      28.8MB
free5gc/ausf-base   latest          55a1ece078c9   21 minutes ago      24.7MB
free5gc/amf-base    latest          2e2ba9726d07   24 minutes ago      28.6MB
free5gc/base        latest          67328d860083   27 minutes ago      1.28GB
mongo               latest          2e123a0ccb4b   44 hours ago        757MB
free5gc/ueransim    latest          9e5933cc53fd   7 weeks ago         155MB
free5gc/chf         <none>          4555a3c2427a   2 months ago        46.1MB
free5gc/smf         v3.3.0         f03dc13fa8aa   6 months ago        24.6MB
free5gc/udm         v3.3.0         9b6215bf0a15   6 months ago        22.6MB
free5gc/nrf         v3.3.0         b4b1ecf1ac94   6 months ago        28MB
free5gc/ausf        v3.3.0         7c1952789afd   6 months ago        23.9MB
free5gc/nssf        v3.3.0         efaae85d547d   6 months ago        21.8MB
free5gc/webui       v3.3.0         724f77370f79   6 months ago        115MB
free5gc/amf         v3.3.0         527021e3cbea   6 months ago        27.7MB
free5gc/upf         v3.3.0         6abd9349af50   6 months ago        123MB
free5gc/pcf         v3.3.0         a84fca3ed266   6 months ago        28.4MB
free5gc/udr         v3.3.0         854121fab907   6 months ago        28.3MB
free5gc/n3iwf       v3.3.0         4f964e0a9aff   6 months ago        27MB
ldriss@ldriss-virtual-machine:~/Desktop/free5gc-compose$
```



### III. Déploiement

#### 1. Mise en route du réseau avec les configurations par défaut

- Pour lancer free5gc via Docker, on utilise la commande : **sudo docker-compose -f docker-compose.yaml up -d && sudo docker compose up**
- Le GnodeB se lance automatiquement lors du lancement de Free5gc, on a précisé les configurations du GnodeB dans le fichier **docker-compose.yaml**
- Nous avons une interface utilisateur qui nous permet de créer et d'enregistrer des subscribers dans la base de données du réseau. Nous avons ainsi le numéro IMSI de chaque l'utilisateur. Dans l'onglet Realtime Status, on peut suivre quelle subscriber est actif et s'il est lancé ou pas. (Voir la capture d'écran ci-après)
- Pour la gestion du GnodeB et des UE : **sudo docker exec -it ueransim bash**
  - Pour activer un UE (subscriber) : **./nr-ue -c config/uecfg.yaml**

The screenshot shows the free5gc web interface in a browser window. The address bar displays '127.0.0.1:5000/#/'. The left sidebar contains navigation links: REALTIME STATUS (selected), SUBSCRIBERS, ANALYTICS, and TENANT AND USER. The main content area is titled 'Registered UEs' and includes a 'Refresh' button. Below the title is a table with columns 'SUPI', 'Status', and 'Details'. One entry is visible with SUPI 'imsi-2089300000000001' and Status 'CONNECTED'. A 'Show Info' button is located next to this entry.

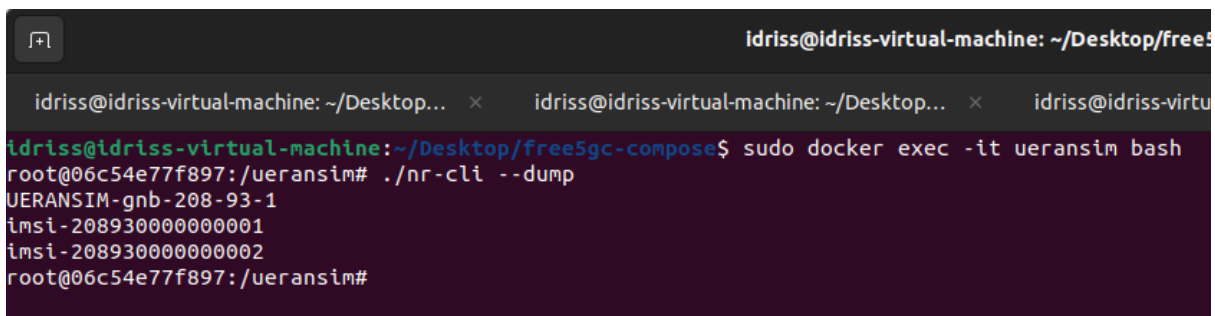
SUPI	Status	Details
imsi-2089300000000001	CONNECTED	Show Info

The screenshot shows the free5gc web interface in a browser window. The address bar displays '127.0.0.1:5000/#/subscriber'. The left sidebar contains navigation links: REALTIME STATUS, SUBSCRIBERS (selected), ANALYTICS, and TENANT AND USER. The main content area is titled 'Subscribers' and includes a 'New Subscriber' button. Below the title is a table with columns 'PLMN', 'UE ID', 'MSISDN', and two buttons: 'Delete' and 'Modify'. There are six entries in the table, all with PLMN '20893' and UE IDs ranging from 'imsi-2089300000000000' to 'imsi-2089300000000005'.

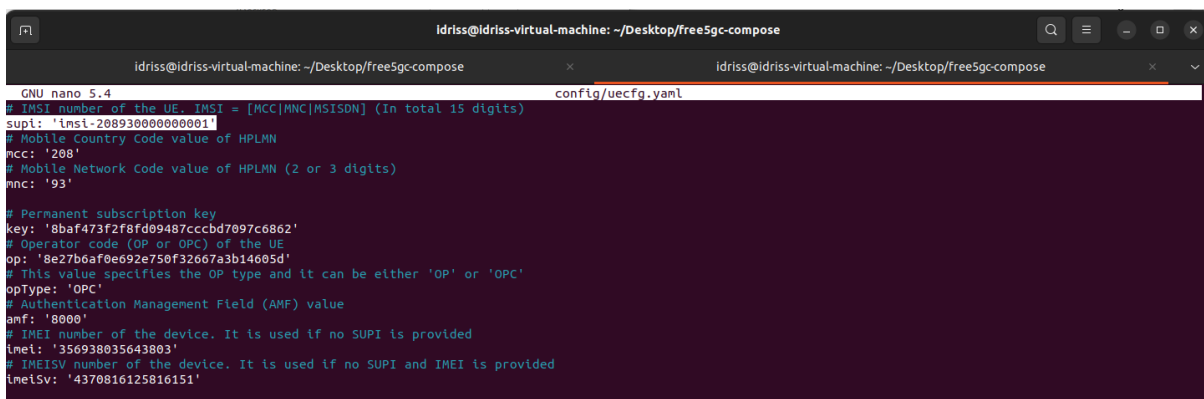
PLMN	UE ID	MSISDN	Delete	Modify
20893	imsi-2089300000000000		Delete	Modify
20893	imsi-2089300000000001		Delete	Modify
20893	imsi-2089300000000002		Delete	Modify
20893	imsi-2089300000000003		Delete	Modify
20893	imsi-2089300000000004		Delete	Modify
20893	imsi-2089300000000005		Delete	Modify

## 2. Multiplier les UE

- Pour la gestion du GnodeB et des UE : **sudo docker exec -it ueransim bash**
  - Pour activer un UE (subscriber) : **./nr-ue -c config/uecfg.yaml**
  - On doit modifier le fichier **uecfg.yaml** qui se trouve dans le répertoire **config** pour lancer l'UE que nous voulons. On modifie la première ligne du fichier en précisant le numéro IMSI de l'UE que nous voulons activer. (Voir la capture d'écran ci-après)
  - Ensuite pour lancer un nouveau UE, on lance la commande dans un nouvel onglet en n'oubliant pas de changer le fichier **uecfg.yaml**
- Comme on peut le voir sur la capture d'écran, nous avons un gnodeB qui est actif ainsi que 2 UE. (Commande : **./nr-cli --dump**)
- Ajouter une capture d'écran sur un ping entre UE 1 et UE 2



```
idriiss@idriiss-virtual-machine: ~/Desktop/free5gc-compose$ sudo docker exec -it ueransim bash
root@06c54e77f897:/ueransim# ./nr-cli --dump
UERANSIM-gnb-208-93-1
imsi-208930000000001
imsi-208930000000002
root@06c54e77f897:/ueransim#
```

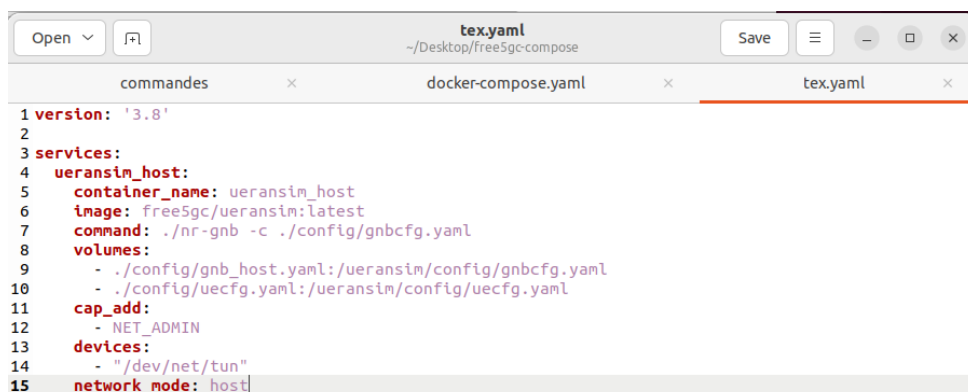


```
GNU nano 5.4 config/uecfg.yaml
# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (in total 15 digits)
supi: 'imsi-208930000000001'
# Mobile Country Code value of HPLMN
mcc: '208'
# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '93'

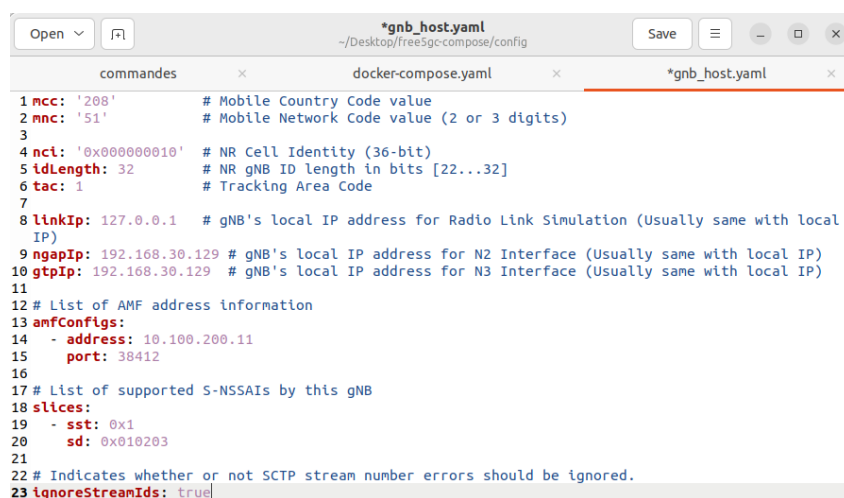
# Permanent subscription key
key: '8baf473f2f8fd09487ccbd7097c6862'
# Operator code (OP or OPC) of the UE
op: '8e27b6af0e692e750f32667a3b14605d'
# This value specifies the OP type and it can be either 'OP' or 'OPC'
opType: 'OPC'
# Authentication Management Field (AMF) value
amf: '8000'
# IMEI number 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'
```

### 3. Multiplier les GnodeB

- Pour multiplier le GnodeB, il faut s'intéresser au fichier **docker-compose.yaml**. Afin de lancer un nouveau GnodeB nous avons créé un nouveau fichier **tex.yaml** dans lequel nous avons changé le nom du conteneur en **ueransim\_host**, changer le fichier de configuration du GnodeB par un nouveau (**gnb\_host.yaml**) et enfin le mode du réseau par host. (Voir la capture d'écran ci-après)
- Et enfin, c'est dans le nouveau fichier **gnb\_host.yaml** qui se trouve dans le répertoire config que nous avons précisé les configurations nécessaires au nouveau GnodeB. Nous avons juste changé les adresses ip du nouveau GnodeB par rapport au fichier d'origine par défaut. (Voir la capture d'écran ci-après)
- Commande pour lancer le nouveau Ueransim : **sudo docker-compose -f tex.yaml up -d && sudo docker compose up**
- Pour précision, dans le fichier par défaut, le code MNC (Mobile Network Code value) était de 93, mais nous avons choisi de le modifier par 51. Le MNC est une partie du PLMN qui identifie de manière unique un opérateur de réseau mobile au sein d'un pays donné.



```
1 version: '3.8'
2
3 services:
4   ueransim_host:
5     container_name: ueransim_host
6     image: free5gc/ueransim:latest
7     command: ./nr-gnb -c ./config/gnbcfg.yaml
8     volumes:
9       - ./config/gnb_host.yaml:/ueransim/config/gnbcfg.yaml
10      - ./config/uecfg.yaml:/ueransim/config/uecfg.yaml
11     cap_add:
12       - NET_ADMIN
13     devices:
14       - "/dev/net/tun"
15     network_mode: host
```



```
1 mnc: '208' # Mobile Country Code value
2 mnc: '51' # Mobile Network Code value (2 or 3 digits)
3
4 nci: '0x000000010' # NR Cell Identity (36-bit)
5 idLength: 32 # NR gNB ID length in bits [22...32]
6 tac: 1 # Tracking Area Code
7
8 linkIp: 127.0.0.1 # gNB's local IP address for Radio Link Simulation (Usually same with local IP)
9 ngapIp: 192.168.30.129 # gNB's local IP address for N2 Interface (Usually same with local IP)
10 gtpIp: 192.168.30.129 # gNB's local IP address for N3 Interface (Usually same with local IP)
11
12 # List of AMF address information
13 amfConfigs:
14   - address: 10.100.200.11
15     port: 38412
16
17 # List of supported S-NSSAIs by this gNB
18 slices:
19   - sst: 0x1
20     sd: 0x010203
21
22 # Indicates whether or not SCTP stream number errors should be ignored.
23 ignoreStreamIds: true
```

## 4. Multiplier le network

- Pour multiplier le réseau, nous devons nous concentrer sur le fichier **docker-compose.yaml**. Dans ce dernier, on peut rajouter un sous-réseau dans la section **networks** mais en n'oubliant pas de changer le nom du bridge. L'autre modification se fait au début du fichier, dans la section **free5gc-upf**, il ne faut pas oublier de rajouter le nouveau sous-réseau.
- La modification de la configuration réseau dans la section **free5gc-upf** assure la cohérence entre les configurations de réseau au niveau de l'UPF et celles définies au niveau global dans la section **networks**. Cela garantit que l'UPF fonctionnera correctement dans le nouveau sous-réseau que nous avons ajouté, permettant ainsi l'extension et la multiplication du réseau selon vos besoins spécifiques.

```
252   - gnb.free5gc.org
253   depends_on:
254   - free5gc-amf
255   - free5gc-upf
256
257 networks:
258   privnet:
259     ipam:
260       driver: default
261       config:
262         - subnet: 20.200.200.0/24
263       driver_opts:
264         com.docker.network.bridge.name: br-free5gc
265   n6net:
266     ipam:
267       driver: default
268       config:
269         - subnet: 20.200.201.0/24
270       driver_opts:
271         com.docker.network.bridge.name: br-free5gc-2
272
273 volumes:
274   dbdata:
```

```
1 version: '3.8'
2
3 services:
4   free5gc-upf:
5     container_name: upf
6     image: free5gc/upf:v3.3.0
7     command: bash -c "./upf-iptables.sh && ./upf -c ./config/upfcfg.yaml"
8     volumes:
9       - ./config/upfcfg.yaml:/free5gc/config/upfcfg.yaml
10      - ./config/upf-iptables.sh:/free5gc/upf-iptables.sh
11     cap_add:
12       - NET_ADMIN
13     networks:
14       privnet:
15         aliases:
16           - upf.free5gc.org
17       n6net:
18         aliases:
19           - upf.free5gc.org
20
21   db:
22     container_name: mongodb
23     image: mongo
24     command: mongod --port 27017
25     expose:
26       - "27017"
```

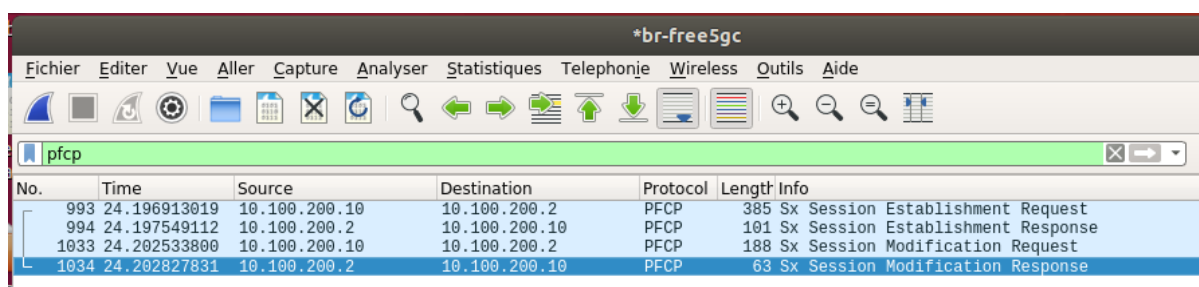


## 5. Manipulation traces et protocoles

Nous allons essentiellement nous concentrer ici sur les traces des échanges qui sont réalisés lors de l'authentification et l'enregistrement de l'UE dans le réseau.

Le protocole NGAP est utilisé sur le lien N2 entre l'AMF et le GNB il permet à l'AMF de réaliser son rôle de gestion de la mobilité et des services.

Lors de l'activation de l'UE nous n'avons trouvé aucun paquet HTTP2 cependant ce que l'on peut remarquer avec Wireshark c'est l'envoi et la réception de paquet PFCP (Packet Forwarding Control Protocol) qui est un protocole 3GPP qui est utilisé au niveau du lien N4 qui permettent de faire la communication entre le control plane et l'User Plane Function (UPF).

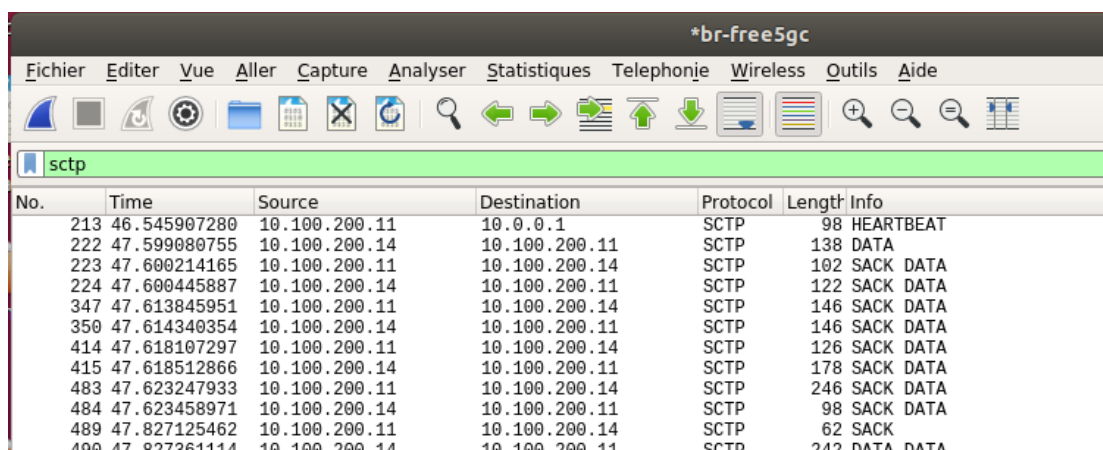


The image shows a Wireshark capture window titled '\*br-free5gc'. The filter bar is set to 'pfcip'. The packet list shows four packets:

No.	Time	Source	Destination	Protocol	Length	Info
993	24.196913019	10.100.200.10	10.100.200.2	PFCP	385	Sx Session Establishment Request
994	24.197549112	10.100.200.2	10.100.200.10	PFCP	101	Sx Session Establishment Response
1033	24.202533800	10.100.200.10	10.100.200.2	PFCP	188	Sx Session Modification Request
1034	24.202827831	10.100.200.2	10.100.200.10	PFCP	63	Sx Session Modification Response

On peut voir la création d'une session à l'initiative du SMF (10.100.200.10) vers l'UPF (10.100.200.2). Cela peut être vérifié en utilisant la commande **sudo docker inspect** (UPF ou SMF).

On peut voir des paquets SCTP (Stream Control Transmission protocol) qui implique des conteneurs comme AMF (10.100.200.11), ueransim(10.100.200.14) et d'autres.



The image shows a Wireshark capture window titled '\*br-free5gc'. The filter bar is set to 'sctp'. The packet list shows multiple SCTP packets:

No.	Time	Source	Destination	Protocol	Length	Info
213	46.545907280	10.100.200.11	10.0.0.1	SCTP	98	HEARTBEAT
222	47.599080755	10.100.200.14	10.100.200.11	SCTP	138	DATA
223	47.600214165	10.100.200.11	10.100.200.14	SCTP	102	SACK DATA
224	47.600445887	10.100.200.14	10.100.200.11	SCTP	122	SACK DATA
347	47.613845951	10.100.200.11	10.100.200.14	SCTP	146	SACK DATA
350	47.614340354	10.100.200.14	10.100.200.11	SCTP	146	SACK DATA
414	47.618107297	10.100.200.11	10.100.200.14	SCTP	126	SACK DATA
415	47.618512866	10.100.200.14	10.100.200.11	SCTP	178	SACK DATA
483	47.623247933	10.100.200.11	10.100.200.14	SCTP	246	SACK DATA
484	47.623458971	10.100.200.14	10.100.200.11	SCTP	98	SACK DATA
489	47.827125462	10.100.200.11	10.100.200.14	SCTP	62	SACK
490	47.827361114	10.100.200.14	10.100.200.11	SCTP	242	DATA DATA

Le reste de nos captures sont en majorité des paquets TCP ou sont en train de communiquer SMF, UDM, PCF, NRF et d'autres conteneurs.

*br-free5gc									
Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide									
Apply a display filter ... <Ctrl-/> Expression...									
No.	Time	Source	Destination	Protocol	Length	Info			
626	47.845551516	10.100.200.12	10.100.200.10	TCP	74	8000 → 45200	[SYN, ACK]	Seq=0 Ack=1 Win=65160 Len=0	MS
627	47.845557421	10.100.200.10	10.100.200.12	TCP	66	45200 → 8000	[ACK]	Seq=1 Ack=1 Win=64256 Len=0	TSval=3
628	47.845588752	10.100.200.10	10.100.200.12	TCP	130	45200 → 8000	[PSH, ACK]	Seq=1 Ack=1 Win=64256 Len=64	T
629	47.845596928	10.100.200.12	10.100.200.10	TCP	66	8000 → 45200	[ACK]	Seq=1 Ack=65 Win=65152 Len=0	TSval=1
630	47.845629153	10.100.200.10	10.100.200.12	TCP	269	45200 → 8000	[PSH, ACK]	Seq=65 Ack=1 Win=64256 Len=203	
631	47.845632090	10.100.200.12	10.100.200.10	TCP	66	8000 → 45200	[ACK]	Seq=1 Ack=268 Win=65024 Len=0	TSval=1
632	47.845783533	10.100.200.12	10.100.200.10	TCP	105	8000 → 45200	[PSH, ACK]	Seq=1 Ack=268 Win=65024 Len=35	
633	47.845788178	10.100.200.10	10.100.200.12	TCP	66	45200 → 8000	[ACK]	Seq=268 Ack=40 Win=64256 Len=0	TSva
634	47.845821068	10.100.200.12	10.100.200.10	TCP	88	8000 → 45200	[PSH, ACK]	Seq=40 Ack=268 Win=65024 Len=2	
635	47.845827593	10.100.200.10	10.100.200.12	TCP	66	45200 → 8000	[ACK]	Seq=268 Ack=62 Win=64256 Len=0	TSva
636	47.845893871	10.100.200.10	10.100.200.12	TCP	75	45200 → 8000	[PSH, ACK]	Seq=268 Ack=62 Win=64256 Len=5	
637	47.845901994	10.100.200.12	10.100.200.10	TCP	66	8000 → 45200	[ACK]	Seq=62 Ack=277 Win=65024 Len=0	TSva

## 6. Routage

Le résultat de la commande `Ip route` :

- La table de routage affiche une route par défaut via l'interface `ens33`, des entrées pour les réseaux `10.100.200.0/24` et `10.100.201.0/24` associées à des interfaces virtuelles inactives.
- Une route pour le réseau local `169.254.0.0/16` via `ens33`, une entrée pour le réseau Docker `172.17.0.0/16` via `docker0` inactive.
- Une route pour le réseau local `192.168.30.0/24` via `ens33` avec une métrique de 100.

```

idriss@idriss-virtual-machine: ~/Desktop/free5gc-compose
idriss@idriss-virtual-machine: ~/Desktop/free5gc-compose
idriss@idriss-virtual-machine: ~/Desktop/free5gc-compose$ ip route
bash: /tools/dev/bash_profile: No such file or directory
idriss@idriss-virtual-machine:~/Desktop/free5gc-compose$ ip route
default via 192.168.30.2 dev ens33 proto dhcp metric 100
10.100.200.0/24 dev br-free5gc proto kernel scope link src 10.100.200.1
10.100.201.0/24 dev br-free5gc-2 proto kernel scope link src 10.100.201.1
169.254.0.0/16 dev ens33 scope link metric 1000
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
192.168.30.0/24 dev ens33 proto kernel scope link src 192.168.30.129 metric 100
idriss@idriss-virtual-machine:~/Desktop/free5gc-compose$

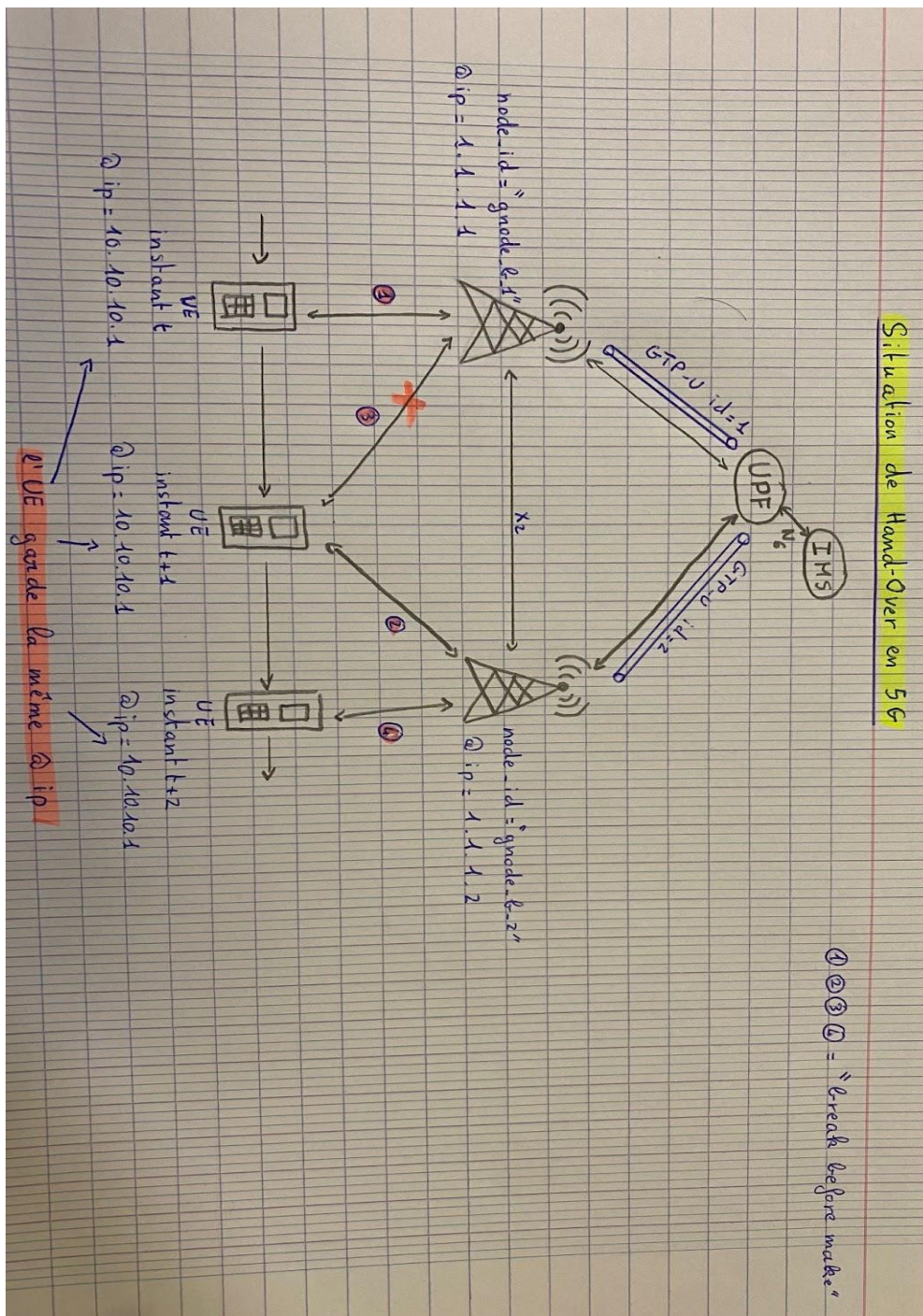
```

## IV. Conception

### 1. Schéma de Hand Over

Nous avons réalisé ci-après un schéma applicatif décrivant une situation de HO (Hand Over) dans lequel le mobile se déplace d'un GnodeB à un autre.

- Nous avons mis en évidence les deux tunnels GTP-U (entre les GnodeB et l'UPF).
- Le mode concerné est SSC1 : On peut voir que l'adresse IP de l'UE reste la même lors du processus de Hand Over. On peut aussi voir le « Break Before Make ».

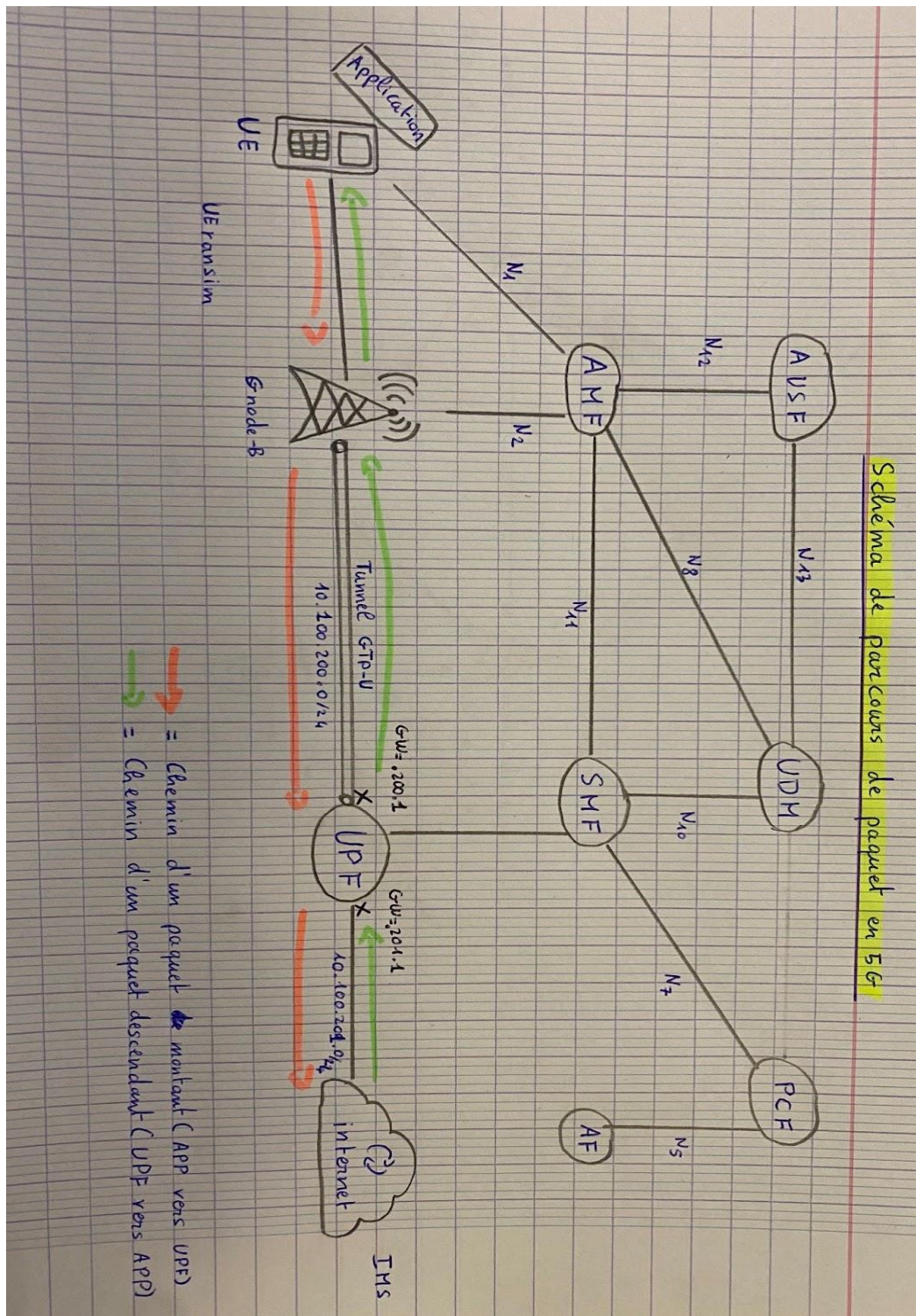




## 2. Schéma de parcours d'un paquet

Nous avons réalisé ici un schéma applicatif décrivant la route empruntée par :

- Un paquet descendant (UPF vers l'Application mobile)
- Un paquet montant (Application du mobile vers UPF)
- Nous avons mis en évidence le tunnel GTP-U, ainsi que les Gateway et adresses IP des 2 interfaces avec lesquels l'UPF est connecté.



## V. Conclusion

En conclusion de ce rapport sur Free5GC, nous avons pu voir que l'installation et la configuration de Free5GC sont relativement simples, offrant une solution accessible pour déployer un réseau 5G. Grâce à sa structure modulaire et à ses guides détaillés, même les utilisateurs avec une expertise limitée peuvent réussir à mettre en place cet élément crucial de l'infrastructure 5G.

**Nous n'avons pas pu réaliser toutes les tâches demandées, telles que la partie routage ainsi que la partie conception avec la mise en place d'un service applicatif. Mais grâce à ce projet et les différents TP réalisés au cours de ce module autour de Free5gc, nous avons pu savoir en profondeur comment fonctionnent les différentes parties du réseau 5G, leurs configurations mais aussi leurs rôles au sein du réseau.**

La 5G, en tant que technologie de communication de nouvelle génération, représente une révolution majeure dans notre vie quotidienne. Sa capacité à fournir des débits de données beaucoup plus rapides, une latence réduite et une connectivité plus fiable ouvre la voie à des applications et des services innovants.

La 5G facilite la croissance de l'Internet des objets (IoT), permettant une connectivité plus étroite et plus rapide entre les appareils. Les applications telles que la réalité virtuelle, la réalité augmentée, et les voitures autonomes bénéficient également de la puissance de la 5G.

En somme, la 5G transforme notre manière de communiquer, d'accéder à l'information, et offre des opportunités nouvelles dans de nombreux secteurs, contribuant ainsi à une société plus connectée et plus intelligente.