

# 5G SIM: Simulation de réseau 5G Core pour évaluation de performance dans le cadre uRLLC

Amina KHADIR, Amira HADDAD, Douaa El-Ikhlas HENNANE, Ikram Zineb DEBBIH,  
Maha Fatima Zohra REMIL, Nada KOUADRI, Yahia Abdeldjalil BENYAHIA

4ème année Option Systèmes Informatiques (SIQ), École Nationale Supérieure d'Informatique  
Alger, Algérie

{la\_khadir, la\_haddad, ld\_hennane, li\_debbih, lm\_remil, ln\_kouadri, ly\_benyahia}@esi.dz

**Abstract**—Les services critiques de la 5G, notamment l'uRLLC, exigent une latence minimale et une haute fiabilité. Dans ce contexte, le placement de la fonction UPF (User Plane Function) a un impact déterminant sur les performances du réseau. Cet article présente une simulation du cœur de réseau 5G visant à évaluer ces effets selon différentes topologies. Après une étude des simulateurs existants, un environnement extensible est mis en œuvre pour tester plusieurs scénarios. Les résultats montrent l'influence du placement des UPF sur les performances du réseau, notamment la latence et la fiabilité.

**Index Terms**—5G core, URLLC, simulation, UPF, placement, latence, fiabilité

## I. INTRODUCTION

### A. Contexte et motivation

L'évolution des réseaux mobiles a conduit à la conception du réseau 5G, une technologie clé qui répond aux besoins croissants en termes de débit, de latence et de fiabilité. Parmi les trois principaux services de la 5G, les communications ultra-fiables à faible latence (uRLLC- Ultra-Reliable and Low Latency Communications) sont essentielles pour les applications nécessitant une communication instantanée et sécurisée, comme l'automatisation industrielle, les services d'urgence et la gestion des véhicules autonomes.

Le réseau cœur 5G (5G Core) joue un rôle central dans la garantie de ces exigences strictes. L'un des défis majeurs est le placement stratégique des User Plane Functions (UPF), qui sont chargées du traitement du trafic utilisateur. Leur positionnement impacte directement la latence et la fiabilité du réseau.

Pour mieux comprendre et optimiser ce placement, une simulation du réseau 5G Core est nécessaire. Elle permet d'évaluer différents scénarios et configurations avant un déploiement réel, réduisant ainsi les risques et les coûts. Ce projet s'inscrit dans une démarche de mesure et l'analyse des performances du 5G Core, avec un accent particulier sur la simulation du nombre d'UPF, leur placement et la distance entre eux, afin d'explorer des configurations optimisées permettant de minimiser la latence et de maximiser la fiabilité des services uRLLC.

### B. Problématique

L'amélioration des performances des services uRLLC dans un réseau 5G Core repose sur plusieurs facteurs, notamment la

gestion efficace du traitement du trafic utilisateur, la répartition des charges réseau et la réduction des délais de transmission. Cependant, l'adéquation entre l'emplacement des UPF, la distance qui les sépare, et la capacité du simulateur à modéliser ces contraintes demeure un défi non trivial.

La problématique centrale est donc la suivante : Comment simuler efficacement un réseau 5G Core afin d'analyser et d'optimiser le nombre et le placement des UPF, tout en respectant les contraintes de latence et de fiabilité, spécifiquement pour les services uRLLC dans des environnements critiques ?

### C. Objectifs

Ce projet a pour objectif principal d'évaluer l'impact du placement des UPF sur les performances des services uRLLC dans un réseau 5G privé simulé. Les objectifs spécifiques sont les suivants :

- Sélectionner et utiliser un simulateur 5G adapté à la prise en compte des contraintes spécifiques aux services uRLLC.
- Identifier un outil capable de mesurer et d'analyser les indicateurs de performance clés, tels que la latence et la fiabilité.
- Modéliser différentes topologies de placement des UPF, en faisant varier leur nombre et la distance entre eux.
- Explorer divers scénarios afin d'identifier les configurations optimales de positionnement des UPF.
- Assurer une simulation réaliste du trafic réseau, en prenant en compte la charge utilisateur et le comportement des protocoles de signalisation et de transport.

### D. Contribution de ce travail

Notre contribution s'articule à la fois sur le plan théorique et sur le plan pratique :

- **Sur le plan théorique** : Une revue de littérature a été menée afin d'identifier les simulateurs et outils existants, suivie d'une analyse comparative de leurs capacités. Notre étude met en évidence l'absence d'outils permettant de simuler de manière précise l'effet de la distance entre les UPF sur la latence et la bande passante, ce qui justifie le besoin d'une solution spécialisée.

- **Sur le plan pratique** : Nous proposons “5G-SIM”, un outil de simulation modulaire qui permet de :

- Créer des topologies personnalisées du 5G Core.
- Simuler différents scénarios de placement des UPF.
- Générer et acheminer dynamiquement le trafic des UE.
- Mesurer les performances du réseau en termes de latence et de fiabilité à l’aide d’OWAMP.

L’outil s’appuie sur des composants existants comme Free5GC et UERANSIM, en y ajoutant une couche d’automatisation permettant l’adaptation dynamique des chemins UPF selon les paramètres du réseau (nombre d’UE, distance, charge).

Il permet de valider empiriquement des hypothèses sur l’optimisation du placement des UPF grâce à des expérimentations contrôlées. Enfin, 5G-SIM contribue à enrichir l’écosystème autour de Free5GC, encore peu doté en outils de simulation réaliste de la distance, en s’appuyant sur des fondements théoriques solides.

## II. FONDEMENTS THÉORIQUES

### A. Architecture de réseau 5G core

1) *Composants et découplage du réseau*: L’architecture du réseau 5G Core (5GC), définie par le 3GPP, repose sur une approche modulaire avec un découplage explicite entre le Plan de Contrôle (Control Plane - CP) et le Plan Utilisateur (User Plane - UP). Ce découplage améliore la flexibilité, la scalabilité et facilite le déploiement des fonctions réseau dans des environnements virtualisés (NFV) ou en cloud natif.

Le plan de contrôle gère la signalisation, la mobilité, l’authentification, la session, tandis que le plan utilisateur traite les données utilisateur (trafic IP). Ce découplage permet de :

- Déployer les fonctions CP et UP indépendamment.
- Optimiser les performances selon les besoins (edge computing, latence...).
- Réduire les coûts d’exploitation.

#### 2) Plan de contrôle:

- **AMF (Access and Mobility Management Function)** : Gère l’enregistrement, la connectivité et la mobilité des terminaux.
- **SMF (Session Management Function)** : Responsable de la gestion des sessions et de l’allocation des adresses IP.
- **AUSF (Authentication Server Function)** : Fournit les services d’authentification.
- **UDM (Unified Data Management)** : Stocke et gère les données d’abonnement.
- **PCF (Policy Control Function)** : Définit les politiques de réseau et de QoS.
- **NRF (Network Repository Function)** : Agit comme un registre central pour que les NF puissent découvrir d’autres NF.
- **NSSF (Network Slice Selection Function)** : Sélectionne les instances de network slicing appropriées.
- **NEF (Network Exposure Function)** : Expose les capacités du réseau aux applications externes.

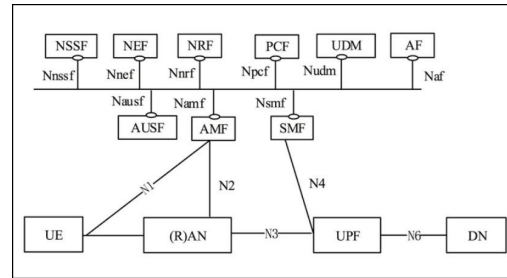


Fig. 1. 5GC-Architecture

### 3) Plan utilisateur:

- **UPF (User Plane Function)** : Traite le trafic utilisateur et applique les règles de QoS.
- **DN (Data Network)** : Connexion aux réseaux externes (Internet, IMS, services cloud).

La figure 1 résume l’architecture d’un réseau 5G core.

### B. User Plane Function (UPF)

Le User Plane Function (UPF) est un composant essentiel du plan utilisateur dans l’architecture 5G Core (5GC). Contrairement aux fonctions du plan de contrôle, le UPF gère exclusivement le traitement et le transport des paquets de données utilisateur.

1) *Interfaces et protocoles du UPF*: Le UPF est connecté à plusieurs entités via des interfaces clés :

- **N3 (avec le gNB)** : transporte le trafic utilisateur encapsulé via GTP-U.
- **N4 (avec le SMF)** : utilise le protocole PFCP pour configurer dynamiquement le traitement des flux (routage, QoS, filtrage).
- **N6 (vers les réseaux de données)** : permet la sortie du trafic vers Internet, le cloud ou le MEC, en utilisant IP ou SRv6.
- **N9 (entre UPF)** : permet un acheminement hiérarchisé ou distribué, utile pour les services à faible latence.

2) *Importance de placement du UPF*: L’impact du placement du UPF dans l’architecture réseau est déterminant pour la latence, le débit et la gestion du trafic. Un placement en périphérie (edge) permet de réduire la latence et d’alléger le trafic traversant le cœur de réseau, ce qui est essentiel pour les services uRLLC. En revanche, un placement plus centralisé peut être préférable pour simplifier la gestion et l’orchestration du réseau et réduire les coûts d’infrastructure.

Des implémentations légères et distribuées (basées sur Docker, DPDK, P4...) permettent de répondre aux besoins de flexibilité et de performance, mais rendent la gestion dynamique plus complexe. Le placement optimal doit ainsi concilier latence minimale, coûts maîtrisés, QoS assurée et continuité de service.

## III. ÉTAT DE L’ART

### A. Vue d’ensemble des simulateurs 5G core

Récemment, plusieurs simulateurs ont été développés pour faire de la recherche et de l’expérimentation dans le domaine

des réseaux 5G. Cependant, peu d'entre eux offrent un support complet des fonctionnalités 5G Core, notamment le placement des upfs (User Plane Function). La majorité de ces outils se concentrent soit sur la partie Radio ou implémentent une partie limitée du Core 5G. Cette limitation constitue un frein pour les chercheurs qui veulent réaliser des scénarios complets qui simulent le fonctionnement des UPFs, le network slicing ou encore l'optimisation de performances. Cette section propose une vue d'ensemble des principaux simulateurs du cœur 5G retenus dans notre étude. Ces derniers ont été choisis en se basant sur plusieurs critères comme : la disponibilité de la documentation, simulation de la partie cœur et la facilité d'installation. Les simulateurs choisis sont :

- OMNeT++/Simu5G
- ns-3
- Free5GC
- Open5GS

Chaque outil est brièvement présenté dans la suite de cette section. Une évaluation comparative sera ensuite proposée, basée sur des critères techniques et fonctionnels.

### B. Simulateurs étudiés

1) *OMNeT++/ Simu5G*: OMNeT++ est un framework de simulation d'événements discrets pour la modélisation réseau, doté d'un IDE intégré. La bibliothèque INET enrichit ses capacités avec des modèles de protocoles et d'interfaces variés.

Simu5G, une extension d'OMNeT++, est dédiée à l'évaluation des performances des réseaux 5G NR et LTE/LTE-A. Elle simule les entités comme les gNodeB et UEs, intégrant une pile protocolaire 5G complète. Simu5G permet une simulation approfondie de la pile 5G NG et automatise les scénarios. Elle supporte le calcul multi-cœurs et l'émulation temps réel, bénéficiant d'une communauté active.

Cependant, Simu5G présente des limites : un support incomplet du cœur de réseau 5G (seule l'UPF est simulée), une scalabilité limitée due à la complexité de la simulation de pile protocolaire sur chaque nœud, et des évaluations de mobilité, latence et fiabilité qui manquent de réalisme à cause de l'absence de handovers complets ou de modèles radio détaillés.

2) *ns-3*: NS-3 est un simulateur de réseau discret open-source, principalement développé en C++, qui offre une interface optionnelle en Python. Il est fréquemment utilisé dans le milieu universitaire pour modéliser et analyser les protocoles réseau. NS-3 se concentre essentiellement sur la modélisation de l'accès radio (RAN) pour la 5G, en particulier par le biais de l'intégration du module 5G-LENA, conforme aux spécifications de la version 3GPP Release-15. Ce module permet de modéliser les transmissions 5G NR (New Radio) avec une précision minutieuse sur les couches PHY et MAC.

#### Avantages :

NS-3 propose une structure modulaire qui simplifie l'intégration et l'implémentation de nouveaux protocoles. Il offre une simulation précise des technologies d'accès comme le Wi-Fi, LTE et NR, tout en conservant un bon degré de réalisme radio. Il est aussi compatible avec des instruments

de suivi et de représentation (Wireshark, NetAnim), et jouit d'une communauté dynamique proposant une multitude de ressources accessibles.

#### Limites :

NS-3 ne fournit pas de mise en œuvre des fonctions centrales du réseau 5G (5GC), telles que l'AMF, le SMF ou l'UPF, restreignant ainsi son utilisation dans des scénarios de bout en bout. L'absence de ces éléments empêche une évaluation exhaustive des services 5G comme l'URLLC dans le cadre de leur réseau global. Par ailleurs, NS-3 n'est pas compatible avec les architectures cloud-native ou virtualisées (NFV, microservices), et sa sophistication technique peut constituer un obstacle pour les utilisateurs novices.

3) *Free5GC*: Free5GC est une implémentation open source du cœur de réseau 5G, conforme aux spécifications 3GPP Release 15 et développée à Taiwan par le Network Convergence Laboratory. Destinée aux chercheurs et développeurs, elle offre une alternative aux équipements propriétaires pour tester et déployer des solutions 5G.

Basée sur une architecture orientée services (SBA) et écrite en Go, Free5GC utilise des microservices interconnectés via des APIs REST, généralement déployés avec Docker/Kubernetes pour plus de flexibilité.

#### Avantages :

L'architecture 5G Core présente plusieurs avantages. Sa conception en microservices permet une modularité et une évolutivité adaptées aux besoins réseau. Elle est conforme aux standards 3GPP, ce qui garantit l'interopérabilité. Le caractère open source la rend personnalisable, et elle est compatible avec différents RAN, comme UERANSIM.

#### Limites :

En revanche, elle a aussi ses limites. Ses performances peuvent être inférieures à celles des solutions matérielles. La topologie réseau est statique, sans adaptation dynamique. Enfin, la dépendance à la communauté peut poser problème pour le support et les mises à jour.

4) *Open5GS*: Open5GS est une implémentation open-source du cœur de réseau mobile 4G/5G conforme aux spécifications 3GPP. Conçu en langage C avec une architecture modulaire, il permet de simuler un réseau 5G complet en mode Standalone (SA), notamment lorsqu'il est couplé à UERANSIM pour l'émulation des UEs et gNBs. Il adopte l'approche CUPS et la Service-Based Architecture, où les fonctions réseau interagissent via HTTP/2 et JSON.

#### Avantages :

Open5GS prend en charge l'ensemble des fonctions 5GC (AMF, SMF, UPF, etc.) et offre une grande flexibilité de déploiement (mono-instance ou distribué). Il est hautement personnalisable et compatible avec des outils de mesure comme Wireshark, Prometheus ou iperf3, facilitant l'évaluation fine des performances réseau.

#### Limites :

Open5GS ne propose pas de mécanisme automatique de tolérance aux pannes et effectue un placement statique des UPFs, ce qui limite l'évaluation de la résilience et des stratégies dynamiques d'optimisation réseau.

### C. Évaluation comparative des simulateurs

1) *Critères d'évaluation*: Choisir un simulateur 5G implique d'évaluer des critères techniques et pratiques essentiels pour la recherche, les tests et l'optimisation réseau avant déploiement.

#### Critères Techniques

- Support des protocoles 5G Core et 5G NR : Le simulateur doit prendre en charge la simulation du cœur de réseau et de l'accès radio 5G NR pour une analyse complète des performances.
- Capacité à simuler différentes topologies et scénarios : Il doit permettre de modéliser diverses configurations réseau (positionnement UPF, densité d'UEs, pannes de nœuds) et des scénarios réalistes.
- Génération de métriques de QoS (latence, fiabilité) : Essentiel pour évaluer la qualité de service, notamment pour les communications URLLC, en mesurant la latence et le taux de perte de paquets.
- Scalabilité : La capacité à modéliser l'ajout progressif d'utilisateurs (UEs) et de fonctions UPF, tout en observant l'impact sur la latence et la fiabilité, est cruciale pour les réseaux 5G privés.

#### Critères Pratiques

- Facilité d'installation et configuration : Le simulateur doit être simple à installer et à configurer pour optimiser le temps dédié à l'expérimentation.
- Documentation et ressources d'apprentissage : Une documentation claire, des tutoriels, des exemples et une communauté active sont indispensables pour une bonne prise en main et un support efficace.
- Possibilités d'extension et personnalisation : Un simulateur modulaire et extensible permet d'ajouter de nouvelles fonctionnalités, modules ou scénarios selon les besoins spécifiques du projet.
- Interface utilisateur et visualisation des résultats : Une interface graphique intuitive et des outils de visualisation des résultats améliorent l'expérience utilisateur et facilitent l'interprétation des données.

2) *Résultats de comparaison*: Le tableau III-C2 compare les simulateurs OMNeT++/Simu5G, ns-3, Free5GC et Open5GS, en se basant sur les critères techniques et pratiques définis précédemment. Les évaluations sont issues des fonctionnalités documentées, des limites identifiées et de leur pertinence pour les objectifs du projet.

## IV. MÉTHODOLOGIE

### A. Choix du simulateur retenu

En se basant sur le tableau comparatif de la section précédente, on constate que :

- OMNeT++ / Simu5G offre un support limité du cœur 5G. Il se concentre principalement sur la couche radio (5G NR), ce qui le rend insuffisant pour des scénarios complets incluant les fonctions du Core.

- ns-3 permet une simulation fine de la couche radio grâce à l'extension 5G LENA, mais ne couvre pas entièrement le cœur 5G.
- Open5GS propose un cœur 5G complet, mais il est principalement conçu pour des environnements de déploiement réel, ce qui le rend moins adapté à la simulation académique.
- Free5GC, en revanche, fournit une implémentation complète du cœur 5G, avec une architecture modulaire permettant la personnalisation. Il s'intègre facilement avec UERANSIM pour simuler l'accès radio, et permet de mesurer des métriques telles que la latence et la fiabilité, essentielles pour les services uRLLC.

Sa forte modularité et son architecture basée sur des microservices peuvent entraîner une certaine complexité de configuration et une baisse de performance, mais ces caractéristiques représentent un atout en milieu académique, car elles permettent de tester indépendamment chaque fonction réseau. → Grâce à sa modularité et à sa complétude fonctionnelle, Free5GC est retenu comme base pour les simulations menées dans ce projet.

### B. Vue d'ensemble de notre solution "5G SIM"

Notre solution "5G SIM" est un simulateur de réseau 5G complet basé sur l'architecture free5GC déployée via Docker Compose. Cette approche conteneurisée garantit une installation simplifiée et une portabilité optimale sur différents environnements de développement et de test.

a) *Architecture du système*: Le simulateur adopte une architecture entièrement conteneurisée où chaque composant fonctionne de manière isolée :

- **Free5GC Core Network** : Chaque fonction réseau (NF) - AMF, SMF, UPF, PCF, UDR, UDM, AUSF, NRF - est déployée dans un conteneur Docker dédié, garantissant l'isolation et la scalabilité.
- **UERANSIM** : Les composants radio (gNB et UE) sont également conteneurisés, permettant une simulation réaliste de la partie radio 5G.
- **Interface terminal** : Console de commande native fonctionnant sous Linux Ubuntu.
- **Interface web** : Interface utilisateur web intuitive pour la gestion et le monitoring.

L'ensemble de ces conteneurs communique via des réseaux Docker personnalisés, et nous utilisons Linux Traffic Control (tc) pour gérer les interfaces réseau entre les différents conteneurs, permettant ainsi un contrôle précis des conditions de transmission.

b) *Fonctionnalités principales*: Notre solution offre des capacités avancées couvrant tous les aspects de la simulation 5G :

**Simulation des NFs et protocoles 5G** : En s'appuyant sur l'implémentation open-source free5GC, notre simulateur reproduit fidèlement le comportement de toutes les Network Functions (NFs) essentielles, y compris les protocoles d'interface entre les différents éléments du core network. Cette base nous permet d'émuler avec précision les procédures 5G

TABLE I  
COMPARAISON DES SIMULATEURS ET CŒURS 5G

Critères	OMNeT++ / Simu5G	ns-3	Free5GC	Open5GS
Support des protocoles 5G NR et 5G Core	Limité. Modules 5G NR et partiellement 5G Core (UPF)	Non. Support via 5G-LENA sans core	Oui. Intégration avec UERANSIM pour la radio	Oui. Support complet 5GC et LTE via UERANSIM
Capacité à simuler des topologies variées	Oui. Support multi-cœurs et scénarios paramétriques	Oui. Support multi-cœurs et scénarios paramétriques	Oui. Focalisé sur le cœur	Oui. Scénarios variés
Génération de métriques de performance	Oui	Oui	Oui	Oui
Scalabilité	Limité. Coût élevé pour dizaines de nœuds	Oui	Oui	Oui. Adapté aux réseaux privés 5G
Facilité d'installation	Oui	Oui	Oui	Oui
Documentation et communauté	Oui. Documentation détaillée + OMNeT++ Summit	Oui. Large communauté	Limité. Docs techniques, communauté réduite	Limité. Bonne doc, moins de support communautaire
Extensibilité et personnalisation	Oui. Code modulaire, modules ajoutables	Oui. Architecture modulaire avec API	Oui. Conteneurisation supportée	Oui. Adaptation du cœur réseau possible
Interface utilisateur et visualisation	Oui. Interface graphique Eclipse avec visualisation	Non. Ligne de commande uniquement	Non. Pas d'interface graphique native	Non. Similaire à Free5GC

standards telles que l'attachement des UEs, l'établissement des sessions PDU et les handovers.

**Automatisation complète :** Nous avons développé des scripts Python sophistiqués permettant la génération automatique complète des topologies réseau. Ces scripts contrôlent de manière dynamique :

- Le nombre et la configuration des UPFs (User Plane Functions) dans le réseau.
- Le nombre d'UEs (User Equipment) à simuler et leurs caractéristiques.
- Les chemins réseau empruntés par les UEs depuis l'accès radio jusqu'au Data Network (DN).
- L'établissement dynamique des sessions PDU pour chaque UE.

**Génération et mesure de trafic :** Le système intègre des outils avancés de génération de trafic permettant de simuler différents types de flux (ICMP et UDP) avec des profils variables. Les mesures de performance sont réalisées via OWAMP (One-Way Active Measurement Protocol) pour capturer avec précision la latence unidirectionnelle et le débit effectif.

**Simulation de la distance :** Grâce à l'utilisation de Linux Traffic Control (tc), nous pouvons appliquer des règles précises de gestion du trafic pour émuler des conditions de propagation réalistes, incluant :

- La latence variable selon la distance géographique simulée.
- La limitation de bande passante.
- Les pertes de paquets aléatoires.

*c) Interfaces utilisateur:* Le simulateur propose deux interfaces complémentaires :

**Interface terminal :** Une interface en ligne de commande robuste optimisée pour Linux Ubuntu, offrant un contrôle granulaire et des fonctionnalités avancées pour les utilisateurs expérimentés. avec des logs détaillés de chaque étape de la simulation

**Interface web :** Une interface graphique moderne et intuitive permettant une gestion simplifiée des scénarios, la visualisation en temps réel des métriques de performance, et la configuration des paramètres de simulation.

Cette architecture modulaire et ces fonctionnalités avancées font de "5G SIM" un outil puissant pour l'évaluation et l'optimisation des réseaux 5G dans des environnements contrôlés.

### C. Implémentation de la solution

*1) Simulation de la topologie réseau:* La présente section détaille la mise en œuvre d'une simulation intégrale de la couche radio et du cœur de réseau, visant à évaluer les performances des services URLLC.

La couche radio est modélisée au sein d'un environnement conteneurisé UERANSIM, permettant la simulation des User Equipments (UEs) et du gNodeB.

Le cœur de réseau est constitué de diverses fonctions, également implémentées via conteneurisation, notamment les User Plane Functions (UPFs). Les UPFs sont catégorisées en trois types distincts :

- UPF edge : Directement connecté au gNB.

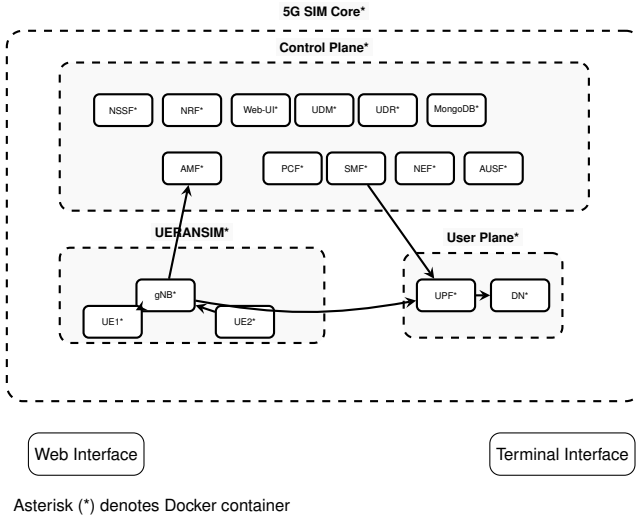


Fig. 2. l'architecture de 5G SIM

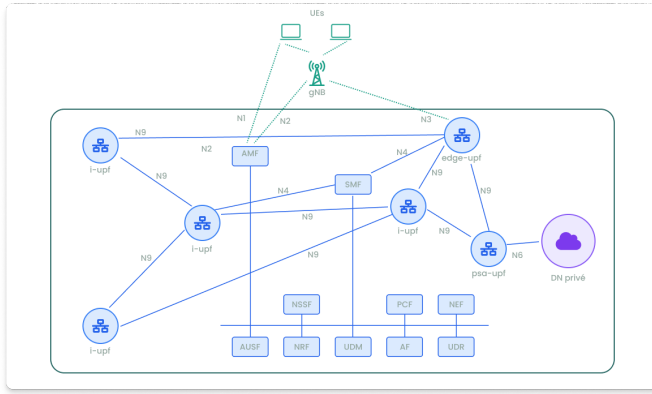


Fig. 3. Architecture du Réseau 5G modélisée

- UPFs intermédiaires : Assurant le rôle de relais dans le chemin des données.
- UPF PSA (PDU Session Anchor) : Relié au Data Network Name (DNN).

Une fonction d'auto-crédation d'UEs est intégrée pour simuler la charge réseau. Avant l'instanciation d'un nouvel UE, une vérification de son existence est effectuée. Si l'UE est nouveau, son fichier de configuration (uecfg\*.yaml) est généré dans le conteneur UERANSIM, puis l'UE est enregistré dans la base de données MongoDB de Free5GC. Après leur création, tous les UEs sont activés simultanément au début de la simulation. La figure 3 montre l'architecture du réseau 5G modélisée.

## 2) Simulation de la distance entre UPF et gNB/UE:

a) *Limites des simulateurs actuels:* Dans ce cadre, la simulation correcte de la distance entre les nœuds du cœur 5G est une condition sine qua non pour évaluer le comportement effectif du réseau dans des circonstances distribuées. Or, les simulateurs open source comme Free5GC actuellement développés ne comportent pas de mécanisme natif permettant

de prendre en compte l'influence de la distance physique. Ceci complique singulièrement l'analyse des performances dans des situations uRLLC, si bien qu'il devient nécessaire de concevoir un mécanisme explicite de simulation de la distance.

b) *Lien entre distance physique et bande passante:* Il est impératif de considérer le lien étroit entre distance et bande passante, puisque plus la distance augmente, plus la bande passante disponible diminue, en raison de la physique du milieu de transmission — notamment en vertu de l'atténuation du signal, de la dispersion ou du bruit thermique — qui sont tous omniprésents et dépendent de la distance. Conscients de cela, nous avons choisi de modéliser la distance non pas comme l'ajout artificiel d'un délai (comme cela se fait le plus souvent), mais par la variation dynamique de la bande passante des liens du réseau.

c) *Pourquoi modéliser la distance par la bande passante:* Certains arguments techniques et physiques qui augmentent la vraisemblance de la simulation font de ce raisonnement notre choix privilégié.

- **Modèle physique véridique :** La bande passante décroissante en fonction de la distance est liée par les lois physiques en transmission optique. Implanter cette dégradation dans le simulateur permet de se rapprocher du comportement d'un réseau déployé.
- **Sensible à la taille des paquets :** Une bande passante dégradée limite l'efficacité de la transmission de gros volumes de données, dans du URLLC où chaque milliseconde compte.
- **Simulation fidèle de la congestion :** Le fait d'agir sur la bande passante au lieu de la latence éveille naturellement les phénomènes de congestion attendant et saturant, qui sont autant de critères nous permettant d'évaluer la résilience et les performances du cœur de réseau en situation réelle.

## d) Modèle d'atténuation utilisé:

$$B(d) = B_0 \cdot 10^{-\frac{\alpha d}{10}}$$

avec :

- $B(d)$  : bande passante restante après une distance  $d$  (en Mbps)
- $B_0$  : bande passante initiale (par exemple, 10 000 Mbps)
- $\alpha$  : coefficient d'atténuation (de l'ordre de 0,2 dB/km pour la fibre optique)
- $d$  : distance (km)

## e) Méthodologie de simulation:

- 1) **Récupération des coordonnées géographiques :** Latitude et longitude des extrémités des liens réseau à simuler.
- 2) **Calcul de la distance :** La distance entre deux points est calculée grâce à la formule de Haversine :

$$d = 2r \cdot \arcsin \left( \sqrt{\sin^2 \left( \frac{\Delta \varphi}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left( \frac{\Delta \lambda}{2} \right)} \right)$$

avec  $r$  le rayon de la Terre,  $\Delta \varphi$  la différence de latitude,  $\Delta \lambda$  celle de longitude.

- 3) **Identification des interfaces concernées** : Seules les interfaces utilisant le protocole GTP sont conservées.
- 4) **Calcul de la bande passante effective** : À l'aide du modèle d'atténuation :

$$B(d) = B_0 \cdot 10^{-\frac{\alpha d}{10}}$$

- 5) **Application de l'algorithme de seuil à jetons** :

$$burst = \frac{B(d)}{8} \times \Delta t$$

avec  $\Delta t$  une fenêtre temporelle (par exemple 0,1 s).

- 6) **Application des paramètres via Traffic Control (tc)** : Configuration via l'outil `tc qdisc` dans les conteneurs Docker représentant les UPFs.

f) *Implémentation dans les conteneurs Docker*: Ces configurations sont exécutées directement dans les conteneurs Docker représentant les UPFs aux niveaux de leurs interfaces réseau. De cette façon, les contraintes de bande passante, de burst et de latence s'appliquent bien à l'ensemble du trafic sortant de façon isolée et contrôlable par UPFs, aussi précise que possible visant à reproduire physiquement les effets topologiques, dans un cœur 5G SIMul.

g) *Simulation du temps de traitement dans les UPFs*: Étant donné que Free5GC est une implémentation open source du cœur 5G, disposant ainsi d'un accès direct à son code source, il est aisé de réaliser les modifications requises pour des expérimentations particulières..

Le fichier **gtp5g.go** du projet Free5GC revêt une grande importance, car il fait fonction d'interface en Go avec le module noyau gtp5g sous Linux, indispensable pour la gestion du trafic GTP-U (GPRS Tunneling Protocol - User Plane) sur un réseau 5G. Il assure l'échange d'informations entre les composants en espace utilisateur de Free5GC et le module noyau **gtp5g** pour créer, gérer et détruire les tunnels GTP selon les règles du Packet Forwarding Control Protocol (PFCP).

A l'aune de la simulation du time-to-handle interne des paquets au sein des UPFs, la fonction **WritePacket()** présente dans **gtp5g.go**, qui construit les paquets GTP-U à partir des données utilisateurs et applique les règles FAR (Forwarding Action Rule) et QER (QoS Enforcement Rule), les encode, puis les envoient par socket UDP vers l'adresse de destination, a été modifiée.

Pour qu'un traitement subisse une latence artificielle, nous avons intégré un bloc de code induisant un délai proportionnel à la taille du paquet. Cette latence est définie de manière suivante :

```
packetSize := len(pkt)
baseDelay := 5 * time.Millisecond
extraDelay := time.Duration(packetSize/100)
* time.Millisecond
totalDelay := baseDelay + extraDelay
time.Sleep(totalDelay)
```

Grâce à ce mécanisme que l'on utilise, on peut modéliser le comportement d'un UPF en charge variable, en tenant compte du fait qu'un paquet peut avoir ses temps de traitement qui ne

sont pas uniformes, en fonction de sa taille. Ce modèle est plus en phase avec le fonctionnement des équipements réseaux, qui plus on monte en taille des paquets à traiter, plus le temps de traitement s'accroît, notamment dans les cas de saturation ou forte sollicitation des équipements.

Cette modification permet donc une évaluation plus précise de l'effet du traitement local dans les UPFs sur la qualité de service globale, en particulier dans les scénarios sensibles tels que les communications **uRLLC**.

### 3) Détermination des chemins privilégiés pour les UE:

Le chemin que chaque UE emprunte pour atteindre le PSA-UPF est déterminé dynamiquement en fonction de la topologie réseau et du nombre d'UPF intermédiaires à traverser. L'objectif est d'optimiser la latence et la charge en définissant un chemin privilégié entre le gNB et le PSA-UPF via des UPF intermédiaires. Le processus repose sur deux étapes principales:

#### 1. Calcul du chemin optimal:

L'utilisateur spécifie :

- Le nombre d'UPFs avec leurs coordonnées.
- Le nombre d'UPFs que chaque UE doit traverser (m).

Le script `find_best_path()` prend ces informations et calcule automatiquement le chemin optimal pour les UEs. Il suit cette logique :

- Identifie le UPF le plus proche du gNB pour le rôle de Edge UPF.
- Utilise un algorithme de Dijkstra contraint pour trouver le chemin de m UPFs menant jusqu'au PSA-UPF, en tenant compte de la distance (alpha) et de la charge réseau (beta).
- Met à jour la charge des UPFs impliqués pour refléter le trafic UE.
- Cette méthode permet de sélectionner un chemin adapté en fonction de critères de performance et d'équilibrage de charge.

#### 2. Mise à jour des fichiers de configuration:

Une fois le chemin déterminé, le script `update_upf_path()` met automatiquement à jour les fichiers YAML nécessaires à Free5GC:

- Fichier SMF (`smfcfg.yaml`) : Met à jour la section links en connectant le gNB au premier UPF, puis chaque UPF entre eux jusqu'au PSA.
- Fichier de routage des UEs (`uerouting.yaml`) :
  - Met à jour la topology pour chaque UE en suivant le chemin optimal calculé.
  - Met à jour `specificPath` si défini, pour forcer l'utilisation du chemin privilégié.

Cette mise à jour assure que chaque UE emprunte effectivement le chemin réseau défini par l'algorithme.

4) *Génération du trafic réseau*: Pendant la simulation, l'ensemble des équipements utilisateurs (UE) procède à un enregistrement simultané auprès du cœur de réseau. Cette phase permet de solliciter intensément les fonctions d'attachement et de gestion de sessions, reproduisant ainsi un contexte réaliste de surcharge. Une fois l'enregistrement achevé, chaque UE génère du trafic URLLC, en personnalisant la taille des paquets, leur nombre, et la fréquence d'envoi, via son interface `uesimtun` (représentant l'interfaçage UERANSIM/Linux). Cela

en utilisant OWAMP, un outil de mesure active implémentant le protocole One-Way Active Measurement Protocol reposant sur UDP.

##### 5) Mesure des performances et collecte des résultats:

Les communications URLLC imposent des contraintes strictes en matière de performances réseau, notamment en termes de latence extrêmement faible, de fiabilité élevée et de gigue minimale. Pour évaluer avec précision la conformité de l'infrastructure simulée à ces exigences, trois indicateurs clés sont mesurés en aller simple entre chaque UE et le réseau de données (DN) :

- la latence
- le taux de perte de paquets
- la gigue

Contrairement aux outils conventionnels tels que ping, qui se basent sur des mesures aller-retour (RTT), OWAMP (One-Way Active Measurement Protocol) permet d'obtenir des mesures unidirectionnelles précises, indispensables dans un contexte URLLC où chaque milliseconde compte. Son intégration dans notre solution permet une observation fine du comportement réseau.

6) *Automatisation du processus de simulation*: Pour assurer un déploiement efficace et personnalisé des topologies 5G, un script de python a été développé. Celui-ci coordonne plusieurs sous-scripts afin de gérer l'ensemble des processus depuis la génération des fichiers de configuration jusqu'au déploiement des conteneurs Docker, l'initialisation des UEs et la mesure du trafic sur la plateforme Free5GC. Le processus débute par la saisie de paramètres tels que: le nombre total des upfs et la désignation des UPFs en position edge. À partir de ces informations, le script automatiquement génère :

- Les fichiers de configuration pour les upfs intermédiaires et les UPF-PSA.
- Le fichier du SMF(Session Management Function).
- le fichier Docker Compose regroupant les services réseau principaux (NRF, UDR, PCF, etc.).
- et les règles de routage UE.

Ce script supporte aussi la génération dynamique des fichiers de configuration des UEs. En effet, un identifiant et un fichier Yaml sont associés à chaque UE, et ils sont insérés dans la base de données MongoDB.

Une fois les fichiers générés, l'environnement Docker est initialisé. Le script vérifie le statut opérationnel des conteneurs des upfs avant de poursuivre. Par la suite, il récupère les coordonnées géographiques de chaque upfs (latitude et longitude) afin de permettre la simulation de la distance comme indiqué précédemment. UEs sont ensuite connectés via UERANSIM en exécutant le processus 'nr-ue' dans les conteneurs correspondants. Un délai entre les lancements est introduit pour assurer le bon établissement de la session. Le routage des paquets est basé sur un algorithme de sélection de chemin optimal.

Enfin, un script de mesure est appelé pour évaluer les indicateurs de performance réseau tels que la latence, la gigue (jitter) et la perte de paquets.

Cette solution assure un déploiement rapide, reproductible, et fidèle aux contraintes réelles des topologies 5G expérimentales.

## V. RÉSULTATS

Cette section présente les résultats expérimentaux obtenus avec notre simulateur 5G SIM. Étant donné que notre projet se concentre sur le placement optimal des UPFs, nous évaluons les performances en fonction du nombre d'UPFs déployés, de la distance géographique et du nombre d'UEs supportés. Pour simuler un cas d'utilisation réel de trafic URLLC, chaque UE envoie un flux UDP avec les caractéristiques suivantes :

- Type de trafic: UDP
- Taille des paquets: 64 octets
- Intervalle entre paquets: 0.001s
- Nombre de paquets envoyés: 10 par UE

### A. Environnement de Test

Les expérimentations ont été menées dans l'environnement suivant (machine virtuelle):

- **OS** : Ubuntu 20.04 LTS
- **CPU** : Intel i5 (8 cœurs, 1.6 GHz)
- **RAM** : 8 GB
- **Stockage** : SSD NVMe 40 GB
- **Docker** : 20.10.12
- **Free5GC** : v3.2.0
- **UERANSIM** : v3.2.6

### B. Méthodologie d'Évaluation

Trois paramètres principaux :

- **Nb UPFs** : 1 à 5
- **Distance** : 1 km à 100 km
- **UEs** : 1 à 50

Métriques : latence, débit, perte de paquets.

### C. Impact du Nombre d'UPFs

1) *Configuration de Test*: Test avec 20 UEs et distance fixe (10 km) :

- **1-6 UPFs** : UEs répartis automatiquement

TABLE II  
PERFORMANCES VS Nb UPFs (20 UEs, 10 km)

UPFs	Latence (ms)	Débit (Mbps)
1	2.80	1206
2	3.63	2061
3	4.92	2694
4	6.27	3260
5	6.85	3554
6	7.97	3734

### 2) Résultats de Performance:

### D. Impact de la Distance

Tests avec 3 UPFs et 15 UEs. Voir la table III et la figure 5.



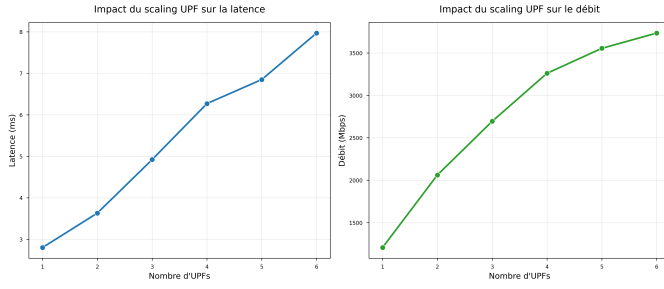


Fig. 4. Évolution des performances avec le nombre d'UPFs

TABLE III  
IMPACT DE LA DISTANCE (3 UPFs, 15 UEs)

Distance (km)	Latence (ms)	Débit (Mbps)	Perte de paquets (%)	Jitter (ms)
1	4.12	1875	1.184	0.52
5	3.36	1686	1.142	1.27
20	3.81	1535	1.817	1.60
50	5.03	957	2.396	2.59
100	6.09	577	3.980	4.80
200	9.13	103	7.350	7.86

## VI. DISCUSSION

### A. Synthèse des résultats

Cette section présente une synthèse des résultats expérimentaux obtenus avec notre simulateur 5G SIM pour l'optimisation du placement des UPFs dans les réseaux 5G.

1) *Observations Principales*: Les expérimentations menées révèlent plusieurs tendances importantes concernant l'impact du nombre d'UPFs et de la distance géographique sur les performances réseau.

**Scalabilité des UPFs** : L'augmentation du nombre d'UPFs de 1 à 6 améliore significativement le débit agrégé du système, passant de 1206 Mbps avec un seul UPF à 3734 Mbps avec six UPFs. Cette amélioration s'explique par la répartition de la

charge entre plusieurs instances UPF, permettant un traitement parallèle des flux de données. Cependant, cette scalabilité s'accompagne d'une augmentation de la latence, qui passe de 2.80 ms à 7.97 ms. Cette dégradation résulte de la complexité accrue du routage et de la coordination entre multiples UPFs.

**Impact critique de la distance** : Les résultats démontrent que la distance géographique constitue le facteur le plus critique pour les performances. Pour des distances inférieures à 20 km, le système maintient des performances acceptables avec une latence inférieure à 4 ms et un débit supérieur à 1500 Mbps. Au-delà de 50 km, une dégradation notable apparaît avec une latence dépassant 5 ms et un débit chutant sous la barre des 1000 Mbps. À 200 km, les performances deviennent critiques avec une latence de 9.13 ms, un débit de seulement 103 Mbps et un taux de perte de paquets atteignant 7.35%.

2) *Implications pour le Déploiement*: Ces résultats mettent en évidence l'importance cruciale du placement géographique des UPFs pour les applications URLLC. Le maintien des UPFs à proximité des utilisateurs finaux (distance inférieure à 20 km) s'avère plus déterminant que l'augmentation du nombre d'instances UPF pour respecter les exigences de latence ultra-faible.

**Point d'équilibre optimal** : L'analyse suggère qu'un déploiement avec 3 à 4 UPFs représente un compromis optimal entre amélioration du débit et maîtrise de la latence pour la configuration testée. Au-delà de ce seuil, les gains de débit ne compensent plus l'augmentation de latence pour les applications critiques.

3) *Limitations*: Il convient de noter que les résultats obtenus ne correspondent pas encore aux exigences strictes du trafic URLLC, qui nécessite une latence inférieure à 1 ms et une fiabilité de 99.999%. Les performances observées dans nos tests, bien qu'encourageantes, nécessitent des optimisations supplémentaires au niveau de la configuration réseau, des algorithmes de routage et de la gestion des ressources pour atteindre les standards URLLC.

4) *Validation du Simulateur*: Les résultats obtenus valident la capacité de notre simulateur 5G SIM à reproduire de manière réaliste le comportement des réseaux 5G et à fournir des métriques de performance cohérentes. L'outil développé constitue une plateforme solide pour l'expérimentation et l'optimisation des déploiements 5G, offrant aux chercheurs et ingénieurs la flexibilité nécessaire pour tester diverses configurations sans les contraintes et coûts d'un déploiement physique.

### B. Limites de la solution actuelle

Malgré les avancées significatives apportées par la plateforme 5G SIM, certaines limites techniques demeurent, en particulier en lien avec la complexité croissante des scénarios 5G et les fonctionnalités encore non prises en charge.

La gestion de la mobilité, notamment le support de plusieurs gNodeB et les transitions intercellulaires, n'est pas encore intégrée.

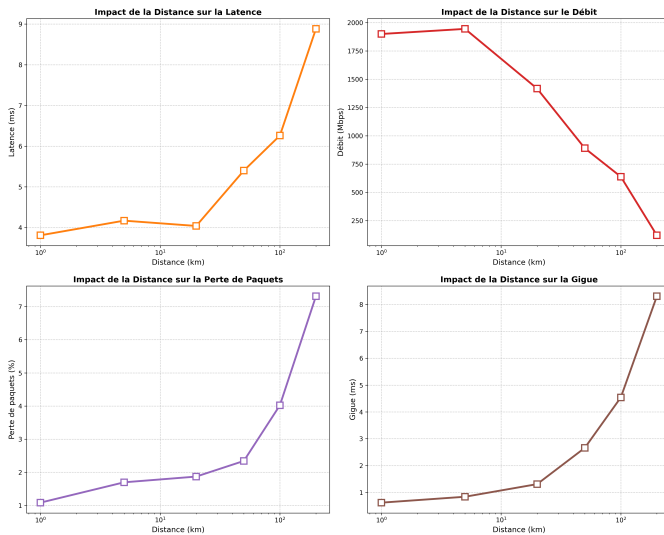


Fig. 5. Performances selon la distance

Par ailleurs, la tolérance aux pannes — comme la résilience en cas de défaillance d'un UPF — n'est pas encore simulée, réduisant la représentativité des environnements critiques.

Ces éléments ne remettent pas en cause l'efficacité de la solution proposée, mais soulignent des axes de renforcement qui permettraient à 5G SIM d'évoluer vers un simulateur encore plus complet et polyvalent, adapté à une plus grande diversité de scénarios 5G.

### C. Difficultés rencontrées lors de l'implémentation

La mise en place de la solution 5G SIM s'est accompagnée de plusieurs défis techniques, liés à l'intégration entre les différents composants du cœur 5G, la configuration dynamique des éléments du réseau, et l'automatisation complète du déploiement.

Une première difficulté notable était l'automatisation de la génération des fichiers de configuration. Le format des fichiers imposé par le Free5GC et les adresses IP nécessitait une grande coordination entre les UPFs, SMF, UE. La moindre incohérence empêchait le lancement correct des services réseaux.

Un autre défi était la création de plusieurs UPFs par la modification des fichiers de configuration. Ces changements avaient besoin d'une compréhension approfondie de l'architecture basique du cœur 5G dans le Free5GC notamment l'interaction SMF-UPF.

La gestion de la communication entre conteneurs Docker représentait également un point sensible. Les UPFs intermédiaires et les autres composants du plan utilisateurs devaient échanger des paquets via des interfaces spécifiques et la configuration des réseaux virtuels entre conteneurs devait être minutieusement orchestrée pour éviter les conflits d'adressage.

Malgré ces défis, la mise en œuvre réussie de l'outil 5G SIM démontre la faisabilité de simuler le placement dynamique d'UPF dans un réseau 5G privé. Les connaissances acquises en surmontant ces difficultés ont été essentielles pour affiner notre architecture de simulation et améliorer la robustesse et la flexibilité de la solution proposée.

## VII. CONCLUSION ET TRAVAIL FUTUR

### A. Contribution du travail

Ce travail apporte plusieurs contributions majeures, tant sur le plan méthodologique que technique, pour l'étude et l'optimisation du placement des UPF dans les réseaux 5G privés à contraintes uRLLC :

#### 1) Revue et analyse comparative des simulateurs 5G Core:

- Dresser un état de l'art exhaustif des outils existants (OM-NeT++/Simu5G, ns-3, Free5GC, Open5GS), en évaluant à la fois leurs capacités à modéliser le cœur 5G et leurs fonctionnalités pratiques (installation, documentation, extensibilité) .
- Identifier l'absence d'un simulateur capable d'intégrer de façon native l'influence de la distance physique entre UPF sur la latence et la bande passante, justifiant le développement d'une solution ciblée .

#### 2) Conception et implémentation de "5G-SIM" :

- Développement d'un environnement de simulation modulaire basé sur Free5GC et UERANSIM, conteneurisé via Docker Compose pour garantir la portabilité et la reproductibilité .
- Mise en œuvre d'un modèle de distance réaliste, fondé sur l'atténuation de bande passante, appliqué dynamiquement aux liens UPF à l'aide de Linux tc .
- Ajout d'une latence de traitement interne proportionnelle à la taille des paquets dans le code Go de Free5GC (gtp5g.go), afin de simuler l'impact du dimensionnement et de la charge des UPF .

#### 3) Automatisation complète de la génération de topologies et de configurations :

- Scripts Python pour :
  - Générer automatiquement les fichiers SMF et UE-routing YAML selon le chemin optimal déterminé .
  - Déployer et vérifier l'état des conteneurs réseau (NF, gNB, UE) avant lancement des expérimentations .
- Intégration d'un algorithme de Dijkstra contraint prenant en compte distance () et charge () pour sélectionner dynamiquement les chemins UPF privilégiés .s

#### 4) Mesure et validation expérimentale des performances uRLLC:

- Utilisation d'OWAMP pour des mesures unidirectionnelles précises de latence, gigue et perte de paquets, adaptées aux exigences uRLLC .
- Expérimentations variées (nombre d'UPF, distance, nombre d'UE) démontrant :
  - Le compromis optimal entre débit agrégé et latence pour 3–4 UPF.
  - L'effet critique de la distance (> 20 km) sur la qualité de service.
- Validation de l'aptitude de 5G-SIM à reproduire fidèlement les comportements d'un déploiement réel et à guider le dimensionnement des infrastructures.

En synthèse, 5G-SIM enrichit l'écosystème de simulation 5G en offrant un outil complet et reproductible pour l'optimisation du placement des UPF, combinant modélisation fine de la distance, automatisation dynamique des chemins et mesures précises uRLLC.

### B. Améliorations futures

Les futures améliorations de la plateforme de simulation visent à renforcer sa pertinence et ses capacités :

- Gestion de la Mobilité et Topologies Multi-gNB: L'intégration de la simulation de multiples gNodeB (gNB) permettra d'implémenter des scénarios de mobilité complexes, incluant la gestion des handovers entre cellules et la possibilité de définir des chemins de trafic différenciés en fonction du gNB d'attachement.
- Tolérance aux Pannes et Mécanismes de Sauvegarde des UPFs : Le développement de mécanismes de gestion

des pannes au niveau des User Plane Functions (UPFs) et l'implémentation de solutions de sauvegarde (backup) renforceront la robustesse de la simulation. Cela permettra d'évaluer l'impact des défaillances de composants sur la continuité de service.

- Fonctionnalités UPF Avancées : L'ajout de fonctionnalités avancées au sein des UPFs, telles que la gestion de la Qualité de Service (QoS), les fonctions de pare-feu (firewall) ou d'autres services de traitement de paquets.
- Optimisation du Routage des Paquets : L'intégration d'algorithmes d'optimisation pour la sélection du chemin de données (path selection).
- Modélisation des Flux de Trafic et des Piles Protocolaires : Une amélioration de la fidélité des modèles de trafic, intégrant des schémas de génération plus réalistes, ainsi qu'une simulation plus détaillée et complète des piles protocolaires, contribueront à une représentation plus fidèle des performances réelles du réseau.

## RÉFÉRENCES

- [1] G. Nardini *et al.*, Simu5G—An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks , *IEEE Access*, 2020. [En ligne]. Disponible : <https://ieeexplore.ieee.org/document/9211504>
- [2] Simu5G, Simu5G - Official GitHub Repository , [En ligne]. Disponible : <https://github.com/Unipisa/Simu5G>
- [3] Open5GS, Documentation , [En ligne]. Disponible : <https://open5gs.org/open5gs/docs/>
- [4] Free5GC, Documentation officielle , [En ligne]. Disponible : <https://free5gc.org/guide/>
- [5] S. Hosseinishamoushaki, Comparative Performance Analysis of free5GC and Open5GS Core Networks Using UERANSIM Integration , Mémoire de Master, Université de Padoue, 2023. [En ligne]. Disponible : <https://thesis.unipd.it/handle/20.500.12608/82357>
- [6] T. H. de O. Magalhães, Implementation of a virtualized 5G network , Université du Minho, 2023. [En ligne]. Disponible : <https://repositorium.sdum.uminho.pt/bitstream/1822/89317/1/Tiago%20Henrique%20de%20Oliveira%20Magalhaes.pdf>
- [7] GeeksforGeeks, 5G Network Architecture , [En ligne]. Disponible : <https://www.geeksforgeeks.org/5g-network-architecture/>
- [8] Orange 5G Lab, FAQ Architecture 5G – Réponses à vos questions , [En ligne]. Disponible : <https://5glab.orange.com/fr/faq-architecture-5g-reponses-a-vos-questions/>
- [9] Viavi Solutions, Qu'est-ce que l'architecture 5G ? , [En ligne]. Disponible : <https://www.viavisolutions.com/fr-fr/quest-ce-que-larchitecture-5g>
- [10] TEC India, 5G Core Network – Study Paper , [En ligne]. Disponible : [https://www.tec.gov.in/public/pdf/Studypaper/5G%20Core%20Network\\_Study%20Paper\\_v8.pdf](https://www.tec.gov.in/public/pdf/Studypaper/5G%20Core%20Network_Study%20Paper_v8.pdf)
- [11] Intel, Migrer vers un cœur de réseau 5G , [En ligne]. Disponible : <https://www.intel.fr/content/www/fr/fr/wireless-network/core-network.html>
- [12] NS-3 Project, Documentation officielle , [En ligne]. Disponible : <https://www.nsnam.org/>
- [13] perfSONAR, OWAMP – One-Way Active Measurement Protocol , GitHub repository, [En ligne]. Disponible : <https://github.com/perfsonar/owamp>
- [14] Lobna, Blog personnel sur la technologie et l'ingénierie , [En ligne]. Disponible : [https://lobna.me/?source=top\\_nav\\_blog\\_home](https://lobna.me/?source=top_nav_blog_home)
- [15] Docker Inc., Docker Hub – Cloud-based repository for container images , [En ligne]. Disponible : <https://hub.docker.com/>