

Robotique/V2X UTAC Développement d'un nouveau mode conduite dans un convoi autonome

Joseph LOUVILLE

2^{eme} rapport bimensuel de stage.

Tuteurs :

Mme Bouchemal

M. Jun Kim

Fait à Paris, le 06/05/2022

0.1 Installation de OMNeT++

Pour faire fonctionner OMNeT++, il faut déjà installer le JDK de Java : **sudo apt-get install default-jdk** Autres packages à installer :

- flex
- build-essential
- gcc
- g++
- perl
- tcl-dev
- tk-dev
- blt
- libxml2-dev
- zlib1g-dev
- default-jre
- doxygen
- graphviz
- libwebkitgtk-1.0-0
- openmpi-bin
- libopenmpi-dev
- libpcap-dev
- autoconf
- automake
- libtool
- libproj-dev
- libgdal-dev
- libfox-1.6-dev
- libgdal-dev
- libxerces-c-dev
- qt4-dev-tools
- bison
- openscenegraph-plugin-osgearth
- libosgearth-dev

Choisi la version 5.6.2 de OMNeT++ car compatible avec Veins, INET. Pour faire fonctionner Simu5G, il faut pour cela installer la version 6.0, qui est la dernière en date à l'écriture de ce rapport.

Quand on télécharge les fichiers d'OMNeT++, il faut faire exécuter le fichier **configure** puis fonctionner la makefile du dossier.

0.2 Installation de SUMO

Il faut ajouter à OMNeT++ SUMO pour la simulation de trafic de véhicule. Pour l'utilisation de Veins, il faut installer la version 1.8.0 afin que le système de réseau corresponde. dans le dossier SUMO créé, il faut faire un **export SUMO_HOME="\$PWD"** puis créer un nouveau dossier cmake-build dans build puis dans ce dossier appeler la fonction : **cmake ../..** puis **make -j\$(nproc)**.

0.3 Veins

Pour faire fonctionner le package de Veins, il faut avant de le lancer sur OMNeT++, lancer le fichier **sumo-launchd.py** avec l'exécution de SUMO. Pour faire appel à SUMO en même temps, il faut écrire la commande :

```
./sumo-launchd.py -vv -c ~/Documents/sumo-1_8_0/bin/sumo
```

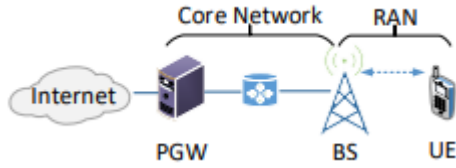
0.4 Explications sur l'article Simu5G : A System-level Simulator for 5G Networks

La cinquième génération des réseaux cellulaires, la 5G est attendu pour une transformation radical du paysage des réseaux. De cette technologie viens se rajouter le Multi-access Edge Computing (MEC), qui vient ajouter les systèmes de clouds en bordure des réseaux de communications.

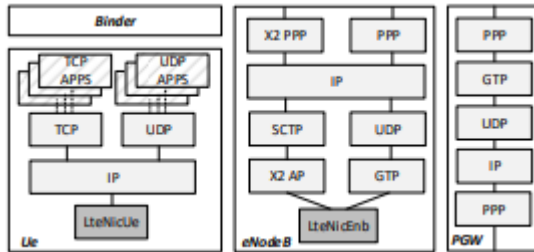
La technologie de Simu5G propose une simulation end-to-end d'application de communications complexes, des scénarios hétérogènes, et qui permet la coexistence de la 4G et 5G.

Simu5G est basé sur le simulateur simuLTE, qui permet l'évaluation de performance des réseaux LTE et LTE Advanced pour le système OMNeT++. SimuLTE est écrit en C++ et est entièrement customisable. L'évaluation des performances des réseaux 5G est d'une importance primordiale, car d'un côté il y a un grand besoin de concevoir et de valider des projets de gestion des ressources pour la 5G RAN (Radio Access Network). Par exemple, la répartition des ressources pour la connection double sur à la fois la 4G et la 5G des UE (User Equipments), ou les algorithmes de planification des gNBs. Aussi la 5G devient important pour le développement de systèmes dépendant de latences critiques, les systèmes temps réel, on besoin de vérifier la faisabilité et les performances de l'utilisation de services basés sur la 5G. Il existe d'autres simulateurs de 5G, 5G-LENA, 5GK-Simulator, Vienna 5G SL Simulator et WiSE, mais ils ne proposent pas de simuler des paquets d'application qui circulent dans le réseau.

Simu5G prend en compte la bibliothèque INET, qui permet de simuler des réseaux TCP/IP génériques incluant des interfaces 2-couches NR (New Radio) 5G. En particulier, Simu5G simule le plan de données du 5G RAN et du réseau central. Il permet la simulation des communications 5G à la fois dans la division de fréquence Duplexage (FDD) et duplexage par répartition dans le temps (TDD), avec des gNB hétérogènes (macro, micro, pico, etc.), communiquant éventuellement via l'interface X2 pour prendre en charge le transfert et les interférences intercellulaires coordonnées. Double connectivité entre un eNB (LTE station de base) et une gNB (station de base 5G NR) est également disponible. Les couches de protocole conformes au 3GPP sont fournies, tandis que la couche physique est modélisée via des modèles de canaux réalistes et personnalisables. La planification des ressources dans les sens montants et descendants sont prises en charge, avec prise en charge de l'agrégation de porteuses et de plusieurs numéologies, comme spécifié par la norme 3GPP (3GPP TR 38.300, TR 38.211). Simu5G supporte un grande variété de modèles de mobilité des UE, y compris mobilité véhiculaire. De plus, il permet d'instancier des scénarios où une application utilisateur, en cours d'exécution sur l'UE, communique avec une application MEC résidant sur un hôte MEC, pour évaluer (par exemple) la latence aller-retour d'un service de nouvelle génération, y compris le temps de calcul au MEC hôte. Plus précisément, Simu5G peut fonctionner en mode d'émulation temps réel, permettant l'interaction avec de vrais appareils. En fait, d'une part OMNeT++ permet la programmation d'événements en temps réel ; d'autre part, la librairie INET permet d'être paramétrée pour échanger des paquets IP entre les applications locales ou les interfaces réseau et le simulateur. Un réseau cellulaire se compose d'un RAN et d'un réseau central. Le RAN est composé de cellules, sous le contrôle d'une seule station de base (BS). Les UE sont attachés à une BS et peuvent changer la BS de desserte via une procédure de transfert. Les BS communiquent entre elles via l'interface X2, une connexion logique qui fonctionne normalement sur un réseau câblé. Le réseau central est constitué de routeurs IP reliant un point d'entrée, le Packet Gateway (PGW) aux BS. La transmission dans le réseau central s'effectue à l'aide du protocole de tunnellation GPRS (GTP). Dans le RAN, les com-



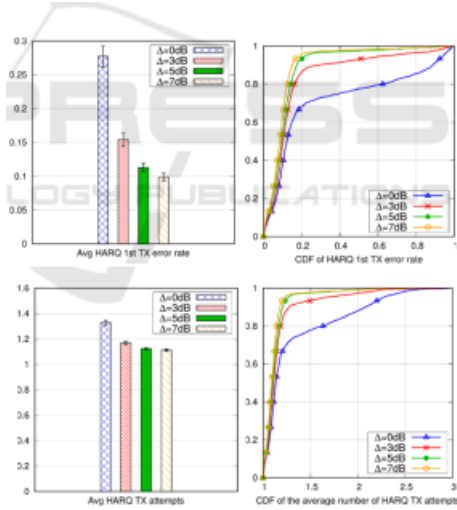
munications entre la BS et l'UE se produisent au niveau de la couche 2 du modèle de référence OSI. Les couches 1 et 2 sont mises en œuvre à l'aide d'une pile de quatre protocoles, à la fois sur la BS et sur l'UE. De haut en bas, nous trouvons d'abord le protocole PDCP (Packet Data Convergence Protocol), qui reçoit les datagrammes IP, effectue le chiffrement et la numérotation, et les envoie à la couche RLC (Radio Link Control). Les unités de données de service RLC (SDU) sont stockées dans la mémoire tampon RLC et sont récupérées par la couche MAC (Media Access Control) sous-jacente lorsque cette dernière doit composer une transmission. Le MAC assemble les unités de données de protocole RLC (PDU) en blocs de transport (TB), ajoute un en-tête MAC et envoie tout via la couche physique (PHY) pour transmission.



SimuLTE simule le plan de données du RAN et du réseau central LTE/LTE-A. Ses principaux nœuds de composants sont illustrés à la figure 3. Les UE et les BS (appelés eNB dans LTE) sont implémentés sous forme de modules composés, qui peuvent être connectés les uns aux autres et à d'autres nœuds (par exemple, des routeurs, des applications, etc.) pour composer des réseaux. Les deux ont une carte d'interface réseau LTE (NIC), qui implémente la pile de protocoles LTE. Le module Ue intègre également les protocoles IP et TCP/UDP, ainsi que des vecteurs d'applications TCP/UDP. Le module eNodeB comprend deux interfaces PPP : le module X2PPP permet la connexion directe avec les eNB voisins utilisant le protocole Stream Control Transmission Protocol (SCTP) tel que spécifié par la norme, tandis que le module PPP est connecté au module PGW. SimuLTE simule le plan de données du RAN et du réseau central LTE/LTE-A. Les UE et les BS (appelés eNB dans LTE) sont implémentés sous forme de modules composés, qui peuvent être connectés les uns aux autres et à d'autres nœuds (par exemple, des routeurs, des applications, etc.) pour composer des réseaux. Les deux ont une carte d'interface réseau LTE (NIC), qui implémente la pile de protocoles LTE. Le module Ue intègre également les protocoles IP et TCP/UDP, ainsi que des vecteurs d'applications TCP/UDP. Le module eNodeB comprend deux interfaces PPP : le module X2PPP permet la connexion directe avec les eNB voisins utilisant le protocole Stream Control Transmission Protocol (SCTP) tel que spécifié par la norme, tandis que le module PPP est connecté au module PGW.

Dans une configuration de réseau typique, le gNB est connecté au PGW pour communiquer avec Internet. C'est ce qu'on appelle le déploiement autonome (SA). Cependant, la norme 3GPP définit un déploiement E-UTRA/NR Dual Connectivity (ENDC), où LTE et 5G coexistent (3GPP – TR 38.801). Cette option est particulièrement intéressante dans les premières phases de

développement de la 5G. Dans cette configuration, le gNB fonctionne comme un nœud secondaire (SN) pour un eNB LTE, qui agit comme nœud maître (MN) et est connecté au réseau central. L'eNB et le gNB sont connectés via l'interface X2 et tout le trafic NR doit passer par l'eNB. Selon (3GPP - TR 37.340). En référence à ce dernier, les données destinées à un UE desservi par l'eNB (Master Cell Group - MCG - porteur) suivent le Pile de protocole LTE, tandis que les données destinées à un UE desservi par le gNB (Secondary Cell Group - SCG - porteur) pénètrent dans l'entité NR PDCP au niveau de l'eNB et sont transférées à son entité d'appairage NR RLC dans le gNB, via l'interface X2. La norme 3GPP prend également en charge les supports fractionnés (SB). Avec cette fonctionnalité, les données appartenant à la même connexion peut traverser soit l'eNB soit le gNB. La couche PDCP côté UE réorganisera alors les PDU provenant des couches LTE/NR RLC. Les communications R peuvent s'effectuer sur plusieurs composantes porteuses de fréquence (CC), c'est-à-dire sur des portions disjointes de fréquence. Chaque CC a un certain nombre de RB. Chaque e/gNB peut implémenter plusieurs CC, chacun caractérisé par sa propre fréquence porteuse, dans le mécanisme dit d'agrégation de porteuses (CA). Pour prendre en charge CA, nous avons modélisé un module carrierAggregation pour stocker toutes les informations relatives aux CC employés dans le réseau. Comme le Binder, il est modélisé comme un module global visible par tous les e/gNBs et UEs dans la simulation. Il comprend un vecteur de sous-modules componentCarrier, dont la fréquence porteuse peut être configurée via NED/INI. Cependant, nous devons tenir compte du fait que les e/gNB et les UE peuvent avoir des capacités limitées en termes de gamme de fréquences prises en charge, par conséquent, un gNB/UE peut ne pouvoir utiliser qu'un sous-ensemble des CC disponibles. Par conséquent, un UE ne peut être attaché à un e/gNB que si ce dernier prend en charge au moins un des CC pris en charge par l'UE lui-même. L'e/gNB, à son tour, peut programmer un UE dans un CC donné uniquement si l'UE prend en charge ce CC. La validation du modèle démontre que

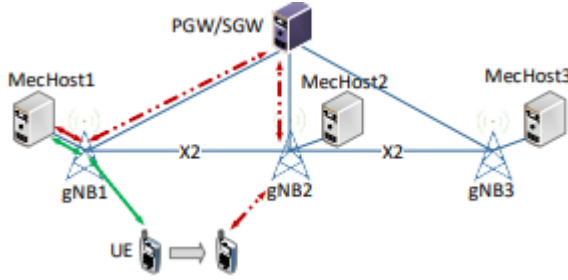


les pertes de messages sont inversement proportionnelles aux décibels lors de la calibration de SimuLTE BLER. Pour évaluer les performances du réseau, on recrée un déploiement en FDD (Frequency-division duplexing) avec seulement du DL pour 30 UEs dans un trafic de Constant Bit Rate (CBR) sur OMNeT++. En augmentant le taux d'envoi, de 80 à 240 kbps, on observe que le trafic devient saturé, et le chargement de certains UEs et le nombre d'UEs obtiennent moins de chargement pour des indices de numéologie inférieurs μ .

μ	0	1	2	3	4
TTI (ms)	1	0.5	0.25	0.125	0.0625

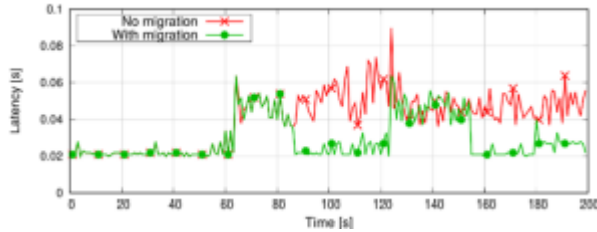
μ est un indice qui permet de définir la valeur du TTI (Transmission Time Interval), interval qui permet de donner le temps entre chaque envoi sur le réseau pour chaque CC (Carrier Component).

Pour prouver l'efficacité du concept, on crée un environnement utilisant un MEC (Multi-access Edge Computing) prenant en compte 3 serveurs représentant des hôtes ME (Mobile Edge) qui sont disposés avec trois gNBs.



Un UE se déplace du gNB1 au gNB2 et gNB3 directement, à une vitesse de 30km/h. L'UE décharge périodiquement les tâches sur l'hôte ME, avec une période = 200 ms. L'UE décharge périodiquement les tâches sur l'hôte ME, avec une période = 200 ms. Pour chaque tâche déchargée, l'UE transfère le contexte à une application ME s'exécutant dans l'hôte ME, qui effectue des calculs et renvoie le contexte à l'UE.

Nous supposons qu'au début de la simulation, l'UE décharge ses tâches sur l'hôte ME 1. Lorsque l'UE effectue le transfert vers les hôtes ME 2 et 3, nous considérons les deux scénarios suivants : a) l'UE continue de décharger ses tâches sur l'hôte ME 1, et b) l'UE décharge ses tâches sur l'hôte ME colocalisé avec le gNB de desserte, c'est-à-dire que le service ME migre en fonction de la mobilité de l'UE. Dans le premier scénario, les données doivent être acheminées via la PGW, la latence supplémentaire est donc uniformément répartie entre 15 et 35 ms (Emara et al., 2018). Dans le deuxième scénario, nous supposons que la migration nécessite un temps compris entre (20, 30), ce qui est compatible avec les résultats de (Taleb et al., 2019). Des algorithmes et des modèles de migration plus avancés peuvent être facilement implémentés et testés dans Simu5G. Les résultats montrent que la migration permet de réduire la latence et stabilise les échanges avec le réseau alors que sans la migration le fait de rediriger le message vers le PGW augmente la latence.



Bibliographie

- [1] Daniel Iancu John Glossner, Samantha Murphy. *An Overview of the Drone Open-Source Ecosystem*. PhD thesis, University of Science and Technology Beijing, 2021.
- [2] F. Launay. *NG-RAN et 5G-NR : L'accès radio 5G et l'interface radioélectrique*. Collection réseaux et télécommunications. ISTE editions, 2021.
- [3] Giovanni Nardini, Dario Sabella, Giovanni Stea, Purvi Thakkar, and Antonio Virdis. Simu5g—an omnet++ library for end-to-end performance evaluation of 5g networks. *IEEE Access*, 8 :181176–181191, 2020.
- [4] Giovanni Nardini., Giovanni Stea., Antonio Virdis., and Dario Sabella. Simu5g : A system-level simulator for 5g networks. In *Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications - SIMULTECH*., pages 68–80. INSTICC, SciTePress, 2020.
- [5] Tiago Pinto. *Towards Wireless ADAS - Retrofitting IoT for increased Safety*. PhD thesis, Polytechnic Institute of Porto, 2019.
- [6] Mukunda Bharathesa Yogesh Simmhan Srikrishna Acharya, Bharadwaj Amrutur. *CORNET 2.0 : A Co-Simulation Middleware for Robot Networks*. phdthesis, Indian Institute of Science, 2021.
- [7] Berk Sunar Vincenzo DiLuoffo, William R.Michalson. *Credential Masquerading and OpenSSL Spy : Exploring ROS 2 using DDS security*. PhD thesis, Worcester Polytechnic Institute (WPI), 2019.