
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
École nationale Supérieure d'Informatique
ex. INI (Institut National de formation en Informatique)

Mémoire

Pour l'obtention du diplôme d'Ingénieur d'État en Informatique

Option : Système Informatique (SIQ)

Conception et développement d'une API abstraite pour l'orchestration de vols de Drones sur un réseau 5G

PAR : Samir Si-Mohammed

ENCADRÉ PAR :

Promoteur : Mr. Adlen KSENTINI, Professeur, EURECOM

Co-encadrants :

- Mr. Yacine CHALLAL, Professeur, ESI Alger
- Mr. Amar BALLA, Professeur, ESI Alger

Septembre 2020

Remerciements

Je tiens avant tout à adresser mes remerciements les plus sincères et les plus vifs à mes encadrants, les Professeurs Yacine CHALLAL, Amar BALLA et Karima BENATCHBA pour leurs conseils, le temps qu'ils m'ont consacré et pour la confiance qu'ils m'ont témoignée. J'espère avoir été à la hauteur de leurs attentes.

J'aimerais ensuite remercier chaleureusement mon promoteur le Professeur Adlen KSENTINI, d'abord pour m'avoir offert cette opportunité, ensuite pour la confiance totale qu'il m'a accordée tout au long de ce stage, ainsi que pour son aide, son soutien, sa gentillesse, et son hospitalité notamment durant tout un mois de Ramadan en plein confinement que j'ai eu à passer loin de chez moi, mais qui grâce à lui, et à Mme KSENTINI que je remercie par la même occasion, ne me rendit pas si dépaysé que ça...

Je tiens aussi à remercier mes deux camarades Karim BOUTIBA et Mohamed MEKKI qui ont effectué leur stage de fin d'études à EURECOM aussi, et avec qui j'ai passé beaucoup de très bons moments.

Je remercie toute l'équipe d'EURECOM : Maha BOUAZIZ, Sagar ARORA, Bouziane BRIK, Siham BAKRI et Sabra BEN SAAD pour tous les bons moments que nous avons passés ainsi que pour la convivialité qui régnait dans l'équipe.

Je remercie l'ensemble du Staff de l'Ecole Nationale Supérieure d'Informatique, qui font un travail formidable et qui chaque jour contribuent à nous rendre fiers de notre école. Particulièrement Mme Dahbia AIT ALI YAHIA pour sa gentillesse, sa patience et pour avoir toujours été à notre écoute.

Je tiens aussi à remercier les membres du Jury, pour l'intérêt qu'ils ont témoigné à mes travaux et pour le temps qu'ils passeront à me lire.

Je remercie l'ensemble de mes amis à l'ESI que je ne prendrai pas le risque de citer exhaustivement, pour avoir été une seconde famille pour moi, et avec qui nous avons tout partagé durant ces années.

J'aimerais réservé une place ici au Club Artistique et Culturel de l'ESI, pour m'avoir fait passer tellement de moments inoubliables.

Je remercie ensuite toutes celles et ceux, que je n'ai pas pu citer mais qui ont contribué à la réalisation de ce modeste travail.

Enfin je remercie les personnes les plus chères à mes yeux, mon grand frère, qui ne cesse de me rendre fier et surtout de me placer la barre de plus en plus haut..., ma mère, qui n'a cessé de me soutenir durant ces longs mois, et enfin, celui qui fut en même temps mon plus grand supporter, et surtout mon plus grand modèle, mon père.

Abstract

5G networks represent the new evolution of mobile networks. Unlike previous evolutions, which only saw an increase in physical throughput, 5G aims to create a new ecosystem ; by opening up mobile networks to vertical industries, such as connected cars, IOT...etc. To do so, 5G must guarantee not only high physical throughput, but also very low latency and high communication reliability. Now, after several years of research, the specifications and architecture of 5G are coming to maturity. The European Commission through the 5G PPP framework has funded several 5G research projects and 5G test platforms in several European countries. EURECOM hosts one of these 5G test sites. The European 5G !Drones project, of which EURECOM is a partner, aims to test and validate the use of 5G by flying drones. The challenge in this context is to fly UAVs in a secure way, ensuring not only a high throughput for the use of video (in the case of surveillance), but above all very low delays in order to precisely control flying UAVs. As part of the 5G !Drones project, EURECOM will develop a management engine (definition, launch/stop, data collection) for the deployment of flying UAVs using EURECOM's 5G !Drones platform. This flight management engine should make it possible to abstract the complexity of the deployment on the platform by offering a high-level test description language. The test management engine must interpret the test descriptor in order to deploy the "Network Slice" that will host the test and retrieve the results. This will be done by using the APIs offered by the EURECOM platform, which represent the objective of this Final Year internship. This API should allow the definition of test scenarios as well as the metrics and results to be collected during the test, using a high-level architecture such as REST. One other parallel objective is to propose algorithms and solutions to optimize the use of 5G resources for flying drones in an efficient way.

Keywords : 5G, Drones, API, Abstraction, REST.

Résumé

Les réseaux 5G représentent la nouvelle évolution des réseaux mobiles. A la différence des précédentes évolutions, qui ont vu uniquement une augmentation du débit physique, la 5G a pour objectif de créer un nouvel écosystème ; en ouvrant les réseaux mobiles à l'industrie verticale, tel que les voitures connectés, IOT, Industrie 4.0...etc. Pour cela la 5G doit garantir, non seulement un débit physique élevé, mais des temps de latence très faibles et une grande fiabilité des communications. A présent et après plusieurs années de recherche, les spécifications et l'architecture de la 5G arrivent à maturité. La commission Européenne à travers le framework 5GPPP a financé plusieurs projets de recherche sur la 5G, et des plateformes de test 5G dans plusieurs pays Européen. EURECOM abrite un de ces sites de test de la 5G. Le projet Européen 5G !Drones , dont EURECOM est partenaire, a comme objectif de tester et valider l'usage de la 5G par les drones volants. Le défi dans ce contexte est de faire voler les drones de manière sécurisée, en assurant non seulement un débit important pour l'usage de la vidéo (dans le cas de surveillance), mais surtout des délais très faibles afin de contrôler de façon précise les drones volants. Dans le cadre du projet 5G !Drones, EURECOM va développer un moteur de gestion (définition, lancement/arrêt, collecte de données) de déploiement de drones volants utilisant la plateforme 5G d'EURECOM. Ce moteur de gestion de vols doit permettre d'abstraire la complexité du déploiement sur la plateforme, en offrant un langage de description de tests de haut niveau. Le moteur de gestion de vols doit interpréter le descripteur du test afin de déployer le « slice réseau » qui hébergera le test et récupérer les résultats des tests. Ceci se fera en utilisant les APIs proposées par la plate-forme d'EURECOM, ce qui représente l'objectif principal de ce stage de fin d'études. Cette API doit permettre de définir les scénarios de tests ainsi que les métriques et résultats à collecter durant le test, en utilisant une architecture de haut niveau, telle que REST. L'autre objectif parallèle de ce stage est la proposition d'algorithmes et de solutions permettant l'optimisation de l'utilisation des ressources 5G pour permettre les vols de drones de manière efficace.

Mots clés : 5G, Drones, API, Abstraction, REST.

Publications issues du PFE

Samir Si-Mohammed, Adlen Ksentini, Maha Bouaziz, Yacine Challal and Amar Balla.
“UAV mission optimization in 5G : On reducing MEC service relocation”, IEEE Global Communications Conference, 2020.

Table des matières

| | |
|--|-----------|
| Abstract | 5 |
| Résumé | 7 |
| Publications issues du PFE | 9 |
| Introduction Générale | 19 |
| | |
| I Synthèse bibliographique | 21 |
| | |
| 1 Vers un moteur d'orchestration optimisée de réseau de drones 5G | 23 |
| 1.1 Historique | 23 |
| 1.2 Architecture | 24 |
| 1.3 Network Slicing | 28 |
| 1.3.1 Définition | 28 |
| 1.3.2 Historique | 29 |
| 1.3.3 Concepts | 29 |
| 1.3.4 Types | 30 |
| 1.3.5 Cas d'utilisation | 32 |
| 1.3.6 Management | 34 |
| 1.4 Software Defined Networking | 35 |
| 1.4.1 Définition | 35 |
| 1.4.2 Architecture | 36 |
| 1.4.3 API Nord/Sud | 37 |
| 1.4.4 API Nord | 38 |
| 1.5 Orchestration réseau | 41 |
| 1.5.1 Solutions Open Source | 42 |
| 1.5.2 Solutions commerciales | 44 |
| 1.6 Conclusion du premier chapitre | 45 |
| | |
| 2 Etat de l'art sur les réseaux de Drones 5G | 47 |
| 2.1 Définition | 47 |
| 2.2 Domaines d'application | 48 |
| 2.2.1 Recherche et Rescousse | 49 |

| | | |
|-----------|--|-----------|
| 2.2.2 | Couverture | 49 |
| 2.2.3 | Transport de biens | 50 |
| 2.2.4 | Construction | 51 |
| 2.3 | Besoins | 52 |
| 2.3.1 | Connectivité | 52 |
| 2.3.2 | Infrastructures | 55 |
| 2.4 | UAV Traffic Management | 56 |
| 2.4.1 | Définition | 56 |
| 2.4.2 | Rôle | 57 |
| 2.4.3 | Architecture | 57 |
| 2.5 | Optimisation des vols de drones | 61 |
| 2.5.1 | Architecture U-space | 61 |
| 2.5.2 | Optimisation du Plan de vol | 62 |
| 2.6 | Conclusion du deuxième chapitre | 65 |
| II | Travail effectué | 67 |
| 3 | Conception | 69 |
| 3.1 | Présentation du projet | 69 |
| 3.1.1 | Objectifs | 70 |
| 3.1.2 | Systèmes UAV Trial | 70 |
| 3.1.3 | Architecture globale du système | 73 |
| 3.1.4 | Slice Orchestrator | 75 |
| 3.1.5 | Portail Web | 76 |
| 3.2 | Analyse des besoins | 78 |
| 3.2.1 | Spécifications fonctionnelles | 78 |
| 3.2.2 | Spécifications techniques | 79 |
| 3.3 | Optimisation de vols | 79 |
| 3.3.1 | Architecture U-space | 79 |
| 3.3.2 | Plan de vol | 85 |
| 3.4 | Conclusion du troisième chapitre | 91 |
| 4 | Tests et résultats | 93 |
| 4.1 | Slice Orchestrator | 93 |
| 4.2 | Portail Web | 94 |
| 4.3 | Scénario d'exécution | 95 |
| 4.4 | Optimisation de vols | 101 |
| 4.4.1 | Evaluation de performances | 102 |

TABLE DES MATIÈRES

| | |
|---|------------|
| | 13 |
| Conclusion Générale | 107 |
| III Annexes | 121 |
| 5 Outils de manipulation de drones | 123 |

Table des figures

| | | |
|------|---|----|
| 1.1 | Architecture 5G en couches (Iwamura, 2015) | 24 |
| 1.2 | Architecture détaillée 5G (Viavi Solution, 2016) | 26 |
| 1.3 | Catégories de Services 5G (Ji et al., 2018) | 32 |
| 1.4 | Gestion de Service et contrôle de Slices (Afolabi et al., 2018) | 34 |
| 1.5 | Architecture SDN (Tijare et Vasudevan, 2016) | 36 |
| 1.6 | Architecture détaillée SDN (Kreutz et al., 2014) | 37 |
| 1.7 | Architecture REST (Fiverr) | 40 |
| 1.8 | Architecture Procera (Kim et Feamster, 2013) | 41 |
| 2.1 | Domaines d'application UAV (Hayat et al., 2016) | 49 |
| 2.2 | Drone UAV sans hélices (Designboom, 2017) | 51 |
| 2.3 | Algorithme de détection de collisions (Alejo et al., 2014) | 52 |
| 2.4 | Architecture UTM (Federal Aviation Agency, 2018) | 58 |
| 2.5 | Acteurs de planification de vol UAV | 64 |
| 3.1 | Architecture 5G !Drones UAV Trial | 71 |
| 3.2 | Architecture 5GEVE | 73 |
| 3.3 | Architecture 5GDrones-5GEVE | 74 |
| 3.4 | Network Slice Template | 76 |
| 3.5 | Architecture Portail Web | 78 |
| 3.6 | Architecture USP-5G | 80 |
| 3.7 | Topologie MEC du réseau | 87 |
| 3.8 | Structure du graphe | 88 |
| 3.9 | Poids des arêtes | 88 |
| 4.1 | Slice Orchestrator, API Nord | 93 |
| 4.2 | Page d'accueil Portail Web | 94 |
| 4.3 | Création d'un Slice | 94 |
| 4.4 | Arrêt d'un Slice | 95 |
| 4.5 | Suppression d'un Slice | 95 |
| 4.6 | Architecture détaillée 5GEVE-SA | 96 |
| 4.7 | Formulaire Métadonnées | 97 |
| 4.8 | Formulaire RAN | 98 |
| 4.9 | Formulaire NFV | 99 |
| 4.10 | Socket Host | 99 |

| | |
|--|-----|
| 4.11 Logs Serveur GPS. | 100 |
| 4.12 Fonctionnement Portail Web. | 101 |
| 4.13 Evolution des métriques du Scenario 1 | 104 |
| 4.14 Evolution des métriques du Scénario 2 | 105 |
| 5.1 Drone utilisé. | 123 |
| 5.2 Télécommande de Drone. | 123 |
| 5.3 Contrôleur Pixhawk4 | 124 |
| 5.4 ARDU Pilot Mission Planner. | 125 |
| 5.5 Mini-Ordinateur Drone. | 125 |
| 5.6 Configuration Mini-Ordinateur. | 126 |
| 5.7 Intégration Mini-Ordinateur au Drone. | 126 |
| 5.8 Dronekit pour la Télémétrie. | 128 |

Liste des tableaux

| | | |
|-----|--|----|
| 1.1 | Cas d'utilisation et Besoin 5G (Nakao et al., 2017) | 34 |
| 1.2 | Solutions Open Source d'orchestration réseau (De Sousa et al., 2019) | 44 |
| 2.1 | Couverture de drones (Hayat et al., 2016) | 50 |
| 3.1 | Méthodes de l'API Nord 5GEVE | 77 |
| 3.2 | Liste des KPI à mesurer 5GEVE | 77 |

Introduction Générale

Parmi les industries qui ont récemment connu un développement considérable se trouve l'industrie des véhicules aériens sans pilotes (Unmanned Aerial Vehicle UAV), ou plus communément appelés drones. En effet, ces dispositifs sont de plus en plus utilisés aujourd'hui, que ce soit à des fins de divertissement par des utilisateurs civils, ou à des fins avec plus d'enjeux pour d'autres types d'utilisateurs (entreprises, gouvernements...). Les drones peuvent être utilisés pour assister les humains dans de nombreuses tâches, comme le transport de marchandises, la surveillance, la prise de photos, la gestion de situations de crise en cas de catastrophe naturelle... etc. Dans tous les cas, il serait préférable de pouvoir contrôler le drone de façon sécurisée mais surtout en temps réel, car dans certaines situations chaque seconde peut être vitale, par exemple dans le cas de fourniture de premiers soins à des victimes d'accidents dans des zones accidentées et difficiles d'accès.

Or, l'avènement des technologies numériques a plongé le monde dans une ère d'évolution fulgurante. Durant les dernières décennies l'ère industrielle a laissé place à ce que l'on appelle « la révolution numérique ». Le domaine des TIC (technologies de l'information et de la communication) connaît donc un essor considérable à travers le monde. Le monde entre désormais dans l'ère de la 5G, qui est une technologie on ne peut plus prometteuse, avec des prédictions de performances remarquables : En particulier, les latences escomptées dans la 5G dépassent tout ce qui a déjà été vu : D'après les spécifications de 3GPP, les réseaux 5G devront offrir des latences de 0.5 ms (3GPP, 2017).

Ainsi est née l'idée d'utiliser la 5G comme technologie de communication pour les drones. Cela permettrait d'avoir un contrôle permanent, fiable et rapide sur les drones pendant les opérations critiques. Le projet 5G!Drones est une initiative de l'union Européenne afin d'effectuer les tests d'une telle intégration de la 5G avec les drones au niveau Européen, et c'est dans le cadre de ce projet que EURECOM, centre de recherche Européen, partenaire important du projet 5G!Drones et détenteur d'une plateforme 5G (5G-EVE) propose un stage de fin d'études pour l'automatisation du déploiement de drones sur sa plateforme.

Plus précisément, le travail consiste dans un premier temps à développer un orchestrateur de réseaux virtuels (Network Slices) permettant d'effectuer les tests de déploiement de drones sur les réseaux 5G offerts par la plateforme 5G-EVE, et dans un second temps de proposer des algorithmes et des solutions afin d'optimiser l'utilisation des réseaux 5G

pour les vols de drones. Ces contributions ont permis de développer un système d'orchestration de réseaux virtuels sur lesquels les drones pourront voler, à travers une plateforme Web, mais aussi à des solutions offrant des solutions prometteuses quant à l'optimisation de vols de drones : en termes d'architecture, mais aussi en termes de l'optimisation du chemin emprunté par les drones durant le vol.

C'est donc avec un grand engouement et une grande motivation que nous avons saisi l'opportunité d'effectuer ce stage auprès d'un centre de recherche de premier plan, afin d'une part de découvrir le travail dans le monde de la recherche, et d'autre part de nous familiariser avec la pratique professionnelle au sein d'un établissement d'envergure mondiale, dans un domaine extrêmement motivant et passionnant pour nous.

Dans ce rapport, nous commençons par effectuer une étude bibliographique sur la 5G, ainsi que sur le domaine des drones. Nous présentons ensuite notre contribution en termes de développement, et en termes de recherche. Nous finissons par quelques activités supplémentaires que nous eûmes la chance d'effectuer durant le stage.

Nous avons pu à l'issue de ce stage, développer un Orchestrateur de réseaux 5G permettant d'effectuer la gestion des vols de drones, ainsi qu'un Portail Web à travers lequel cet Orchestrateur peut être utilisé. Par ailleurs, nous avons proposé une nouvelle architecture optimisée pour l'utilisation de la 5G pour le vol de drones, en assurant la sécurité nécessaire durant les missions de drones. Enfin, nous avons développé une solution permettant d'optimiser le chemin de vols emprunté par les drones durant une mission, ce qui a donné lieu à une publication dans la conférence internationale IEEE Globecom 2020 dont voici la référence : Samir Si-Mohammed, Adlen Ksentini, Maha Bouaziz, Yacine Challal and Amar Balla. “**UAV mission optimization in 5G : On reducing MEC service relocation**”, IEEE Global Communications Conference, 2020.

PREMIÈRE PARTIE

Synthèse bibliographique

Vers un moteur d'orchestration optimisée de réseau de drones 5G

La 5ème génération de réseaux mobiles, ou plus communément appelée 5G, a pour but d'accroître le plus possible la qualité de l'expérience utilisateur, et ce, en assurant un meilleur débit de données, une réduction de la latence... etc. Pour ce faire, des améliorations drastiques doivent être réalisées dans l'architecture des réseaux cellulaires. Nous présentons dans ce chapitre une vue globale de 5G ainsi que certaines des principales technologies émergentes qui permettent d'optimiser l'utilisation de l'architecture et répondre aux demandes des utilisateurs.

Nous commençons par donner un bref historique des réseaux mobiles, puis nous détaillons son architecture, ensuite nous décrivons deux des concepts les plus importants utilisés en 5G : Le Network Slicing et le Software Defined Networking. Pour chacun de ces concepts, nous donnons sa définition ainsi que les travaux actuels qui sont faits dans ces domaines. Enfin, nous concluons ce chapitre par un bref état de l'art sur l'orchestration des réseaux, concept sur lequel porte notre stage de fin d'études.

1.1 Historique

Au cours des deux dernières décennies, le monde a connu l'évolution des technologies de communication cellulaire Long Term Evolution (LTE) du système 2G vers la 4G. La principale motivation a été les exigences de largeur de bande toujours plus importante et offrant de faibles latences. Parallèlement à l'amélioration des performances liées au débit, certains paramètres connexes comme la gigue, l'inter-canal l'interférence et la connectivité, l'évolutivité et la compatibilité avec les réseaux existants ont également été pris en considération (Iwamura, 2015).

Lorsque nous sommes passés du système 2G au système 3G, une vitesse de réseau plus élevée et une vitesse de téléchargement plus rapide ont été remarquées, ce qui a permis de passer des appels vidéo en temps réel. La LTE et la LTE-A (LTE-Avancé) qui a suivi ont offert une capacité du réseau, une réduction des délais d'accès aux serveurs d'applications

et l'accès au trafic triple play (données, voix et vidéo) possible, sans fil, à tout moment et en tout lieu. La 4G peut donc être considérée un réseau haut débit. Bien que la 3G ait été la première à large bande, elle a été conçue à l'origine pour la voix avec une certaine prise en compte du multimédia et des données, tandis que La 2G a été conçue comme la première voix mobile numérique et une norme de communication pour une couverture améliorée par rapport au 1G les services vocaux analogiques de base. Donc, dans le domaine de la téléphonie mobile, nous avons constaté une amélioration constante du débit de données, qui est passé de 64 kbps en 2G à 2 Mbps en 3G et 50 à 100 Mbps en 4G. La 5G devrait non seulement améliorer le transfert de données la vitesse du réseau mobile, mais aussi pour améliorer l'évolutivité, la connectivité et l'efficacité énergétique du réseau. Une véritable "société en réseau" est en train d'émerger par l'internet des objets (IoT), ce qui exige le contrôle des machines par nos appareils mobiles en temps réel. Ainsi, on peut décrire la 5G comme étant un système à haut débit et à faible latence, très fiable, plus évolutif et plus économique en énergie (Agrawal et al., 2016).

1.2 Architecture

(Iwamura, 2015) propose de modéliser l'architecture 5G comme étant une architecture en couches, comme l'indique la figure 1.1. Les principales couches qui y figurent sont :

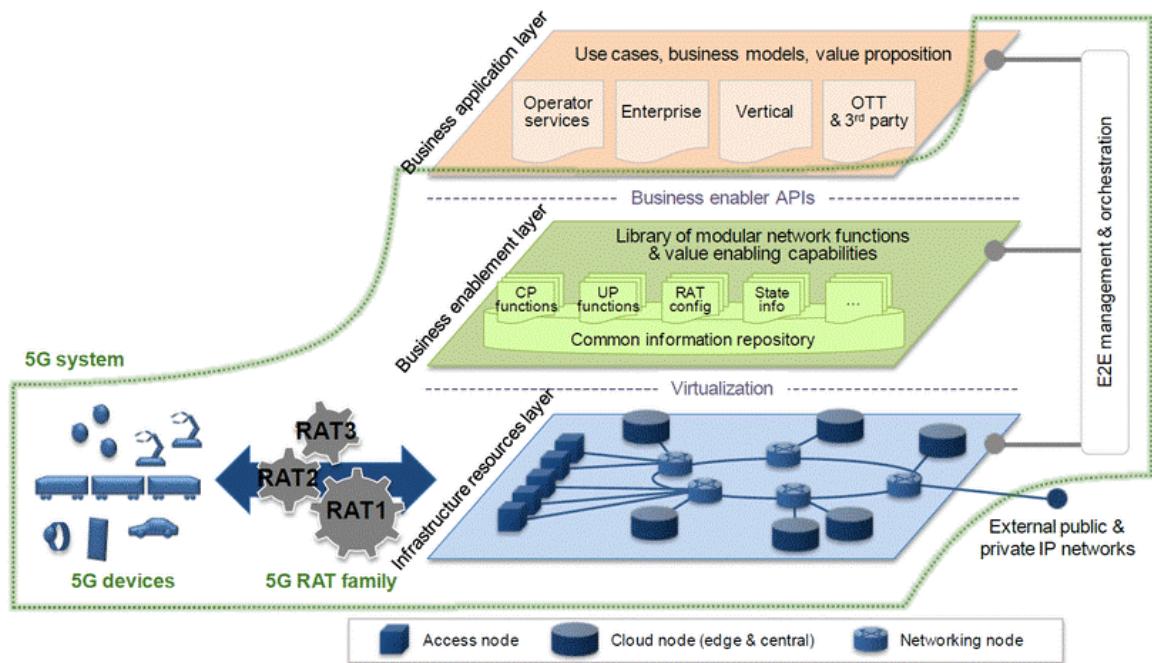


FIGURE 1.1 – Architecture 5G en couches (Iwamura, 2015).

- **Couche Infrastructure** : Les ressources physiques du réseau, comprenant les nœuds d'accès, les nœuds du Cloud (qui peuvent être des ressources permettant le

traitement ou le stockage) (Feng et al., 2014), les appareils 5G (Capteurs, objets connectés, véhicules autonomes...), les nœuds de réseau et les liens associés. Les appareils 5G peuvent avoir de multiples capacités configurables et agir comme un relais ou une ressource informatique de stockage, selon le contexte (Mukherjee et al., 2015). Les appareils 5G sont donc également considérés comme faisant partie de la ressource infrastructurelle configurable. Les ressources sont exposées aux couches supérieures et à l'entité d'orchestration grâce aux principes de virtualisation, c'est-à-dire que la plupart des fonctions offertes dans cette couche sont sous formes de machines virtuelles. Nous définissons plus en détail ce concept par la suite. L'acronyme RAT signifie la technologie d'accès radio (Radio Access Technology), permettant de connecter les appareils 5G aux ressources physiques du réseau principal.

- **Couche Business enabler :** Cette couche regroupe l'ensemble de toutes les fonctions requises au sein d'un réseau convergent sous la forme de blocs, dans une architecture modulaire, et un ensemble de paramètres de configuration pour certaines parties du réseau comme par exemple l'accès radio. Ces fonctions et capacités sont appelées à la demande par l'entité d'orchestration, par le biais des API (Interface d'application). Cette couche peut s'avérer très utile, car différentes mises en œuvre de la même fonctionnalité et de la même ressource physique peuvent donner des performances ou des caractéristiques différentes, ce qui permet à cette couche d'affecter positivement les performances du réseau. Les différents niveaux de performance et les capacités offertes pourraient être utilisés pour différencier les fonctionnalités de réseau de façon beaucoup plus significative que dans les réseaux actuels (Liu et al., 2017).
- **Couche Business Application :** La couche d'application métier contient des applications spécifiques et les services de l'opérateur, de l'entreprise, des verticaux ou des tiers qui utilisent le réseau 5G. L'interface avec l'entité de gestion et d'orchestration de bout en bout permet, par exemple, de construire des Network Slices dédiés à une application, ou d'associer une application aux Network Slices existants (Iwamura, 2015). Un Network Slice est une sorte de réseau virtuel dans un réseau physique, avec certains besoins et fonctionnalités. Le concept de Network Slicing est abordé en détail dans la suite de ce rapport.

Ces trois couches sont articulées par une entité d'orchestration, qui joue un rôle central dans cette architecture. Elle a la capacité de gérer un réseau virtualisé de bout en bout, en ajout à l'automatisation traditionnelle des OSS (Operation Support System), qui désigne l'ensemble des composants opérationnels ou les systèmes informatiques utilisés par un opérateur de télécommunications afin de gérer le réseau. L'entité sert de point

de contact pour traduire les cas d'utilisation et les modèles commerciaux d'abord en services réels, et enfin en Network Slices. Elle définit ces Slices pour une application donnée, lie les fonctions de réseau modulaire adéquates et utilisées pour ce service, leur attribue les configurations de performance nécessaires, et enfin fait le lien entre ces Slices et les ressources dans l'infrastructure. Elle gère la mise à l'échelle de la capacité de ces fonctions ainsi que leur répartition géographique. Dans certains modèles d'entreprise, il pourrait également posséder des capacités pour des tiers (par exemple les verticaux) pour créer et gérer leurs propres Slices, par le biais des API et des principes XaaS (Everything as a Service), qui est une appellation regroupant les principes Cloud IaaS (Infrastructure as a Service), Saas (Software as a Service) et PaaS (Platform as a Service) (Iwamura, 2015).

D'une autre part, une autre architecture plus détaillée a été proposée par (Viavi Solution, 2016), qui est une entreprise offrant des services 5G, et celle-ci sépare le réseau 5G en différentes parties, comme l'indique la figure 1.2 :

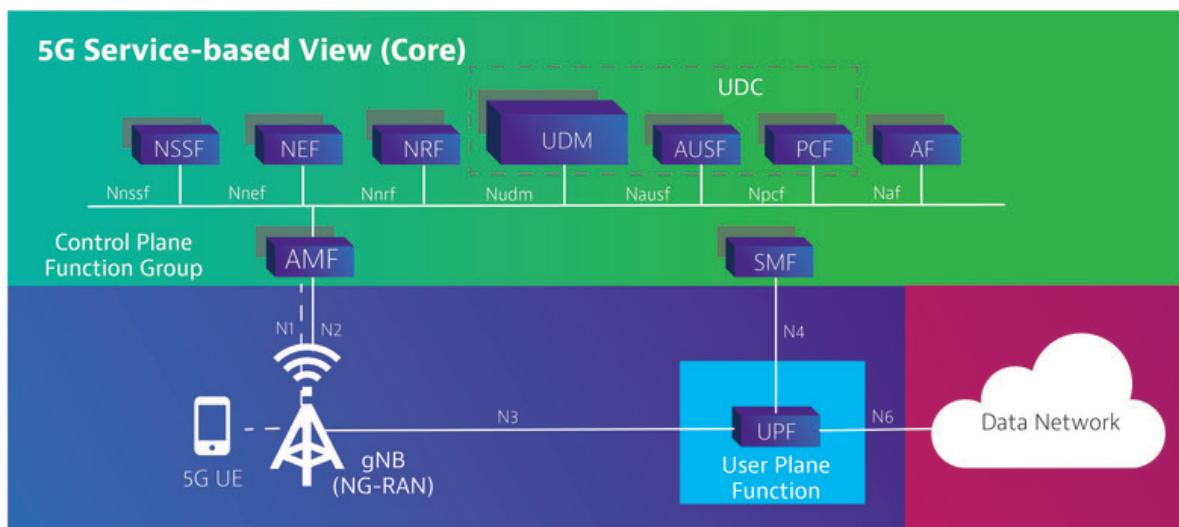


FIGURE 1.2 – Architecture détaillée 5G (Viavi Solution, 2016).

- **Radio Access Network :** Un élément primordial de l'infrastructure du réseau cellulaire est le réseau d'accès radio (RAN), dont le rôle est de fournir une connectivité sans fil à grande échelle aux appareils mobiles. Le problème fondamental que le RAN résout est de trouver la meilleure façon d'utiliser et de gérer un spectre limité pour obtenir cette connectivité. Dans un déploiement sans fil dense avec des noeuds mobiles et un spectre limité, il devient difficile d'allouer les ressources radio, de mettre en œuvre les transferts, de gérer les interférences, d'équilibrer la charge entre les cellules, etc. (Gudipati et al., 2013). Dans la 5G, le concept de virtualisation s'étend au RAN grâce à, par exemple, la désagrégation des réseaux promue par des alliances telles que O-RAN (o-ran.org). Cela crée une flexibilité des

réseaux et crée aussi de nouvelles possibilités de concurrence, fournit des interfaces ouvertes et le développement de logiciels libres, pour finalement faciliter le déploiement de nouvelles fonctionnalités et technologies à l'échelle. Les avantages de la virtualisation des composants du RAN permettent d'être plus rentables du point de vue matériel et logiciel, en particulier pour les applications IoT où le nombre de dispositifs se compte en millions (Viavi Solution, 2016).

- **Core Network :** L'architecture Core network est au cœur de la nouvelle spécification 5G et permet de répondre à la demande accrue de débit que la 5G doit supporter. Le nouveau Core 5G, tel que défini par le 3GPP, utilise une architecture basée sur les services (Service Based Architecture), basée sur le cloud. Le Core Network couvre toutes les fonctions et interactions, y compris l'authentification, la sécurité, la gestion des sessions et l'agrégation du trafic provenant des équipements utilisateurs. Il met en outre l'accent sur le NFV (Network Function Virtualized, qui représente la virtualisation des fonctions réseau) en tant que principe de conception intégrale avec des fonctions logicielles virtualisées pouvant être déployées en utilisant l'infrastructure MEC (Mobile Edge Computing) qui est au cœur des principes architecturaux 5G (Viavi Solution, 2016). Il est à noter que la 5G se divise en deux types : la 5G Standalone, avec des fonctions réseau et des équipements Radio propres à la 5G, et la 5G non-standalone, qui utilise les équipements et les fonctions réseau de la 4G pour fournir de la 5G, c'est-à-dire que le Core Network est celui de la 4G, et la partie Radio est 5G, s'appuyant sur des stations de base 4G (eNodeB). Ce dernier type de 5G est aussi appelé 4.5G.
- **Multi-Access Edge Computing (Data Network) :** Le Multi-Access ou Mobile Edge Computing (MEC) est récemment apparu comme une technique prometteuse. L'idée centrale du MEC est de rapprocher le calcul des utilisateurs, en répartissant de petits serveurs ou des micro-centres de données pouvant héberger des applications en Cloud sur le réseau et connectées directement à des entités, telles que les stations de base cellulaires, à la périphérie du réseau mobile (Taleb et al., 2018). Le MEC devrait également être plus robuste que les systèmes de cloud computing centralisés traditionnels (Satyanarayanan et al., 2013), car elle est distribuée et donc moins affectée par les défaillances en un point centralisé. L'idée de distribuer des serveurs en nuage à la périphérie du réseau mobile est également connue sous le nom de "cloudlets" (Satyanarayanan et al., 2013), "edge computing" (Davy et al., 2014) et "fog computing" (Bonomi et al., 2012). Dans toutes ces techniques, chaque ensemble de serveurs ou chaque centre de micro-données est responsable d'une zone géographique bien définie, bien que certains serveurs/micro-centres de données puissent ne pas être directement connectés à la station de base dans la zone

en question. Le paradigme MEC permet d'offrir des environnements caractérisés par une faible latence, une bande passante large et une connaissance de l'emplacement qui peuvent être exploités par les applications (Aissiou et al., 2018).

Après avoir vu l'architecture de la 5G, nous allons maintenant aborder deux des concepts Software les plus importants en 5G.

1.3 Network Slicing

La 5G étant une technologie qui, à l'instar de la 4G, a été faite pour prendre en charge diverses demandes de services, le Network Slicing s'est vite avéré incontournable quand on parle de 5G (Afolabi et al., 2018). Nous détaillons dans cette section le concept de Network Slicing :

1.3.1 Définition

Il consiste à virtualiser une infrastructure réseau en plusieurs réseaux logiques (Appelés Slices) où les ressources peuvent être flexibles et configurées en fonction des demandes des applications déployées sur cette infrastructure, pour satisfaire au mieux leur demande de qualité de service. Un exemple d'utilisation du Network Slicing est le déploiement sur le même réseau physique, d'un Slice assurant une communication fiable et à faible latence pour des applications de communication en temps réel, et un autre Slice offrant un haut débit pour des applications d'IoT massif (Objets connectés) (An et al., 2019).

Selon (Zhang et al., 2017), répartir un réseau en plusieurs Slices peut permettre de supporter des services à la demande pour des applications distinctes, et ce sur le même réseau physique. Grâce au Network Slicing, les ressources réseau peuvent donc être allouées de manière dynamique et efficace à des Slices logiques, selon les demandes de qualités de service des applications.

Selon ces définitions, le Network Slicing revient à utiliser la virtualisation dans des réseaux physiques. Les mêmes principes que nous connaissons dans le concept de Machine Virtuelles reviennent ici comme la répartition des ressources. (Afolabi et al., 2018) définissent même un Slice comme étant une unification de ressources virtuelles (ex. Machines Virtuelles) où un ensemble de VNF (Fonctions Réseau Virtuelles) sont instanciées et connectées à travers un réseau virtuel. Une fonction réseau virtuelle étant une machine virtuelle ayant pour objectif l'exécution d'une fonction réseau commune (DNS, DHCP...). L'intérêt des VNF est que l'on peut faire exécuter sur un même serveur physique plusieurs fonctions réseau, et ce en optimisant l'utilisation des ressources physiques (CPU,

RAM...).

1.3.2 Historique

Le Network Slicing est basé sur la virtualisation, dont les origines remontent aux années 60, quand IBM a tenté un accès interactif et concurrent à une même machine physique par plusieurs machines virtuelles, où chacune était la copie exacte de la machine physique (Nanda et Chiueh, 2005). Le succès de cette expérience a lancé une importance vague de recherche dans ce domaine (Goldberg, 1975). C'est durant les années 80 que la notion de réseaux virtuels a vu le jour, où l'on a essayé de connecter des nœuds à travers des connexions logiques sur une infrastructure physique.

En 2008, la création de l'initiative GENI (Global Environment for Networking Innovation) (Berman et al., 2014) a donné un essor au développement de la virtualisation réseau. Toutefois et comme nous pouvons le constater, les recherches sur le Network Slicing tel que nous le connaissons aujourd'hui n'en sont qu'à leur début, cela reste un domaine encore à explorer.

1.3.3 Concepts

(Afolabi et al., 2018) définissent les principes suivants comme étant inhérents et dont le respect est primordial dans le Network Slicing :

- **Automatisation** : Faire de la configuration des Slices une configuration automatisée. (Kafle et al., 2018) proposent une méthode de configuration automatisée basée sur l'apprentissage automatique (Machine Learning) pour la conception de l'allocation de Machines Virtuelles pour les Slices.
- **Isolation** : Garantir la performance et la sécurité en assurant une isolation entre les Slices. Ceux-ci peuvent être isolés de manière physique (avec des ressources physiques distinctes), par virtualisation sur la même ressource physique, ou par le partage d'une ressource en appliquant une politique de partage d'accès pour chaque Slice. (Kotulski et al., 2018) traitent en détail de l'isolation dans le Network Slicing.
- **Personnalisation** : Assurer que les ressources allouées à une partie tenant (Utilisateur ou bénéficiaire du Slice) sont efficacement utilisées afin de respecter de la meilleure façon les demandes du service en question.
- **Elasticité** : Permet la réallocation dynamique des ressources initialement allouées à une partie tenant, afin de rendre l'affectation des ressources la plus efficace pos-

sible. Cela peut être utile lorsque l'on fait face à des variations de conditions Réseau, de nombre d'utilisateurs ou de changement de zone géographique due au déplacement de l'utilisateur. L'élasticité peut par exemple prendre la forme d'allocation d'une nouvelle VNF. (Zheng et De Veciana, 2019) proposent un critère d'allocation de ressources qui permet d'apporter une élasticité dans le réseau tout en y offrant plus de protection.

- **Progammabilité :** Permet à des parties tierces de contrôler l'allocation des ressources via des APIs offertes exposant les fonctionnalités réseau d'un Slice. (Zhou et al., 2016) vont même jusqu'à parler de Network Slicing As A Service. Ils proposent une architecture qui permettrait à des opérateurs d'offrir des Slices à la demande en leur assurant une programmabilité quasi-totale.
- **Bout en bout :** Permet à un Slice de combiner des ressources appartenant à différents providers d'infrastructure, ou d'unifier des technologies réseau hétérogènes, dans le but d'améliorer la qualité du réseau ainsi que la qualité de service.
- **Abstraction hiérarchique :** Permet de déléguer à une partie bénéficiant d'un Slice offert par un provider, d'offrir elle-même de manière complète ou partielle les ressources qui lui ont été allouées pour le Slice en question.

1.3.4 Types

L'architecture de la 5G pouvant se diviser en deux grandes parties : Radio Access Network (RAN) et Core Network, le Slicing lui aussi peut être réparti en deux types comme suit :

RAN Slicing :

Le Network Slicing dans un réseau mobile est fortement lié au partage du réseau, en particulier au partage du RAN dans le cas des réseaux mobiles. Il existe différents modèles à envisager pour le mettre en œuvre. Selon leur niveau d'isolation, on peut citer deux types de ressources : les ressources dédiées et les ressources partagées.

Pour les ressources dédiées, le Slice RAN est établi en séparant et en isolant les Slices en termes de trafic, mais surtout en termes de ressources physiques, c'est-à-dire les ressources physiques sont strictement consacrées à un Slice spécifique, par exemple, un pourcentage de la ressource physique PRB (Physical Resource Blocks) est consacré à chaque Slice, ou un sous-ensemble de la chaîne radio est consacré à chaque Slice. Un PRB est une unité de mesure de l'ensemble de ressources Radio attribuées à un Slice, ou un utilisateur. Bien

que le modèle de ressources dédiées accorde des ressources élémentaires pour le Slice, il réduit l'élasticité du Slice ainsi que l'extensibilité, et limite le gain de multiplexage. En effet, l'utilisation du modèle de ressources dédiées ne permet pas de découpage propriétaire pour modifier facilement le niveau de la ressource (c'est-à-dire le PRB) accordé à un Slice pendant son cycle de vie. En outre, le modèle de ressources dédiées peut conduire à un gaspillage de ressources, car les PRB sont strictement consacrés à un Slice, même s'ils ne sont pas utilisés (Ksentini et Nikaien, 2017).

La deuxième approche, qui est le modèle de ressources partagées, elle permet au Slice de partager les mêmes trafic de contrôle et ressources physiques. Dans cette solution, les PRB sont gérés par un planificateur commun qui distribue les PRB aux utilisateurs des Slices selon différents critères, y compris l'accord sur le niveau de service (Service Level Agreement SLA), la priorité, et ainsi de suite. Alors que cette solution exploite les statistiques de l'ordonnancement des ressources physiques, qui garantit plus d'évolutivité et d'élasticité par rapport au modèle de ressources dédiées, elle peut manquer le support d'une garantie stricte de qualité de service (QoS) pour les Slice et l'isolation du trafic (Ksentini et Nikaien, 2017).

Core Slicing :

Les entités du réseau dans le EPC (Evolved Packet Core ; Nom donné au Core Network de la norme 4G) actuel peuvent être classées en deux types (3GPP, 2015) : (1) les entités comme le MME (qui est l'équipement qui gère la signalisation entre les équipements utilisateur et le Core Network) qui ont des tâches de contrôle et ne s'occupent donc que du trafic du flux de contrôle, et (2) les entités qui doivent également s'occuper du trafic du flux utilisateur, comme le SGW et le PGW (qui sont deux équipements d'un réseau mobile 4G LTE qui participent à l'acheminement des données).

Afin de bien comprendre comment décomposer et optimiser l'EPC, les sous-fonctions de MME, SGW et PGW peuvent être classées en groupes de métä-fonction. (Sama et al., 2016) proposent d'effectuer la décomposition des fonctions de l'EPC de deux façons : décomposition horizontale et décomposition verticale. La décomposition "horizontale" fait référence au découplage des fonctions des flux de Contrôle et Utilisateur. La "décomposition verticale" se réfère à l'identification des fonctions individuelles englobées dans une entité de réseau. En utilisant l'approche NFV, ces fonctions décomposées peuvent être mises en œuvre sous forme de logiciel et une liste de fonctions réseau peut être sélectionnée et compilée pour des groupes d'utilisateurs dédiés. La virtualisation des entités responsables du flux de Contrôle est simple, car ils peuvent être instanciés comme n'importe quel autre logiciel informatique (An et al., 2012).

En revanche, certaines des fonctions doivent rester sur du matériel physique dans quelques cas spécifiques où les exigences de performance ne peuvent pas être satisfaites efficacement par des logiciels. Par exemple, les SGW et PGW sont les éléments de réseau qui gèrent à la fois le contrôle et surtout le trafic utilisateur, ce qui entraîne une augmentation massive et la quantité de données qu'ils doivent traiter. Les fonctions du réseau (que ça soit les fonctions utilisateur ou de contrôle) doivent se déplacer sur le réseau en fonction des exigences des cas d'utilisation. En conclusion, le Slicing sur le Core Network consiste en la virtualisation des fonctions réseau composant l'architecture EPC.

1.3.5 Cas d'utilisation

Afin de permettre l'émergence de plusieurs services et applications, l'Union Internationale de la Télécommunications (ITU) a classé les services 5G en trois catégories (ITU-R, 2015), et comme l'indique la figure 1.3 :

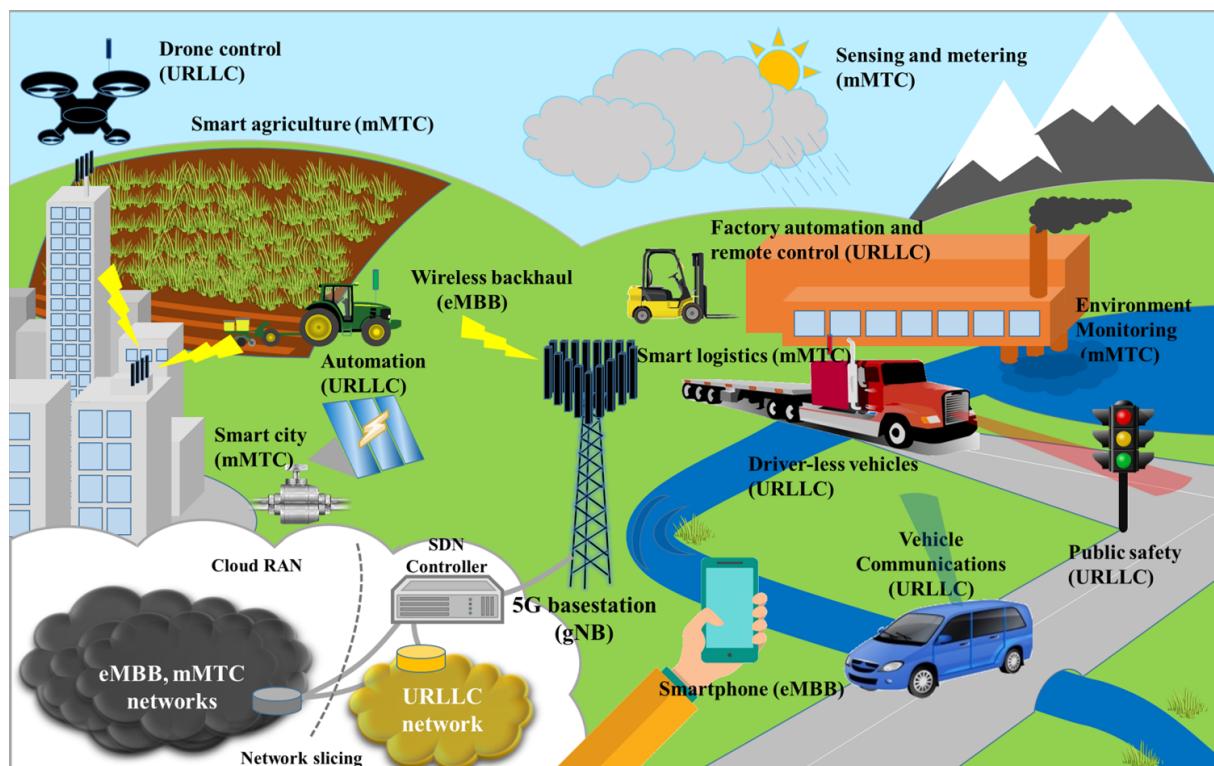


FIGURE 1.3 – Catégories de Services 5G (Ji et al., 2018).

URLLC (Ultra Reliable Low Latency)

Les communications ultra fiables et à faible latence (URLLC) sont une nouvelle catégorie de services en 5G à prendre en charge les services et applications émergents ayant des exigences de latence et de fiabilité strictes (Ji et al., 2018). Il est souvent difficile d'atteindre les besoins escomptés dans ce type de slices, i.e. une latence faible et une grande

fiabilité du réseau, et ceci car quand on tente d'améliorer la fiabilité du réseau, on doit souvent utiliser plus de ressources pour la signalisation, la retransmission, la redondance et la parité, ce qui engendre des latences supplémentaires (Soret et al., 2014). Il est toutefois possible dans certains cas d'atteindre de faibles latences quand la taille des paquets est petite, et au coût d'une réduction du débit (Polyanskiy et al., 2010).

Cette catégorie a pour but de supporter les applications sensibles à la latence, comme le contrôle à distance de dispositifs, les voitures autonomes ou l'internet tactile (Schulz et al., 2017). Le Network Slicing peut être utilisé pour construire un lien dédié entre l'application et le service URLLC, ce qui peut réduire les latences (Schulz et al., 2017).

MMTC (Massive Machine Type Communication)

Les MMTC forment une catégorie de services dont l'objectif est de supporter l'accès simultané d'un grand nombre d'équipements au réseau. Les services qu'offrent les Slices requièrent d'importantes densités de connexion, ainsi qu'une utilisation efficace de l'énergie. Un exemple typique du MMTC est la collection de mesures depuis un nombre massif de capteurs (Bockelmann et al., 2016). La définition des MMTC englobe plusieurs variétés de concepts émergents tels que l'Internet des Objets (IoT), les villes intelligentes (Smart City) ... etc.

EMBB (Enhanced Mobile Broadband) :

eMBB est une catégorie de service liée aux exigences élevées de bande passante, telles que la haute résolution streaming vidéo, réalité virtuelle et réalité augmentée. Le principal défi des systèmes 4G est d'améliorer le débit du système.

Les technologies de couche physique introduites à cette fin incluent la transmission de modulation d'ordre élevé, agrégation de porteuses, densification cellulaire via un réseau hétérogène et transmission à entrées multiples et sorties multiples (Massive Input Massive Output MIMO).

Afin de multiplier par 100 la capacité par rapport aux systèmes 4G, des changements drastiques dans les capacités physiques améliorant l'efficacité spectrale et exploitant le spectre inexploré sont nécessaires. Les technologies envisagées comprennent le MIMO (Multiple Input Multiple Output) massif (Ji et al., 2017), communication utilisant des ondes millimétriques (Han et al., 2015), et forme d'onde localisée spectralement (3GPP, 2017).

Le tableau 1.1 (Nakao et al., 2017) résume ces cas d'utilisation ainsi que leurs besoins

en termes de Qualité de Service (QoS) :

| 5G Cas d'utilisation | Exemple | Besoins | Mobilité |
|----------------------|---|---|----------|
| eMBB/Xmbb | Vidéos 4K/8K haute définition, Réalité Augmentée/-Virtuelle | Grande capacité, Cache vidéo | Oui |
| mMTC | Réseaux de capteurs (Smart city... etc.) | Connexion massive couvrant une grande zone de dispositifs peu mobiles | Non |
| URLLC/uMTC | Voitures autonomes, téléchirurgie | Latence faible et grande fiabilité | Oui |

TABLE 1.1 – Cas d'utilisation et Besoin 5G (Nakao et al., 2017)

1.3.6 Management

Le Network Slicing offre une connectivité de bout-en-bout (Nakao et al., 2017), permettant la cohabitation de différentes technologies réseau sur la même infrastructure (NGMN Alliance, 2016), et ce en répétant un processus de façon continue afin d'analyser les besoins de service et assurer les performances escomptées (Sharma et al., 2017). Ce processus est fait à travers deux couches, comme illustré dans la figure 1.4, et comme suit (Afolabi et al., 2018) :

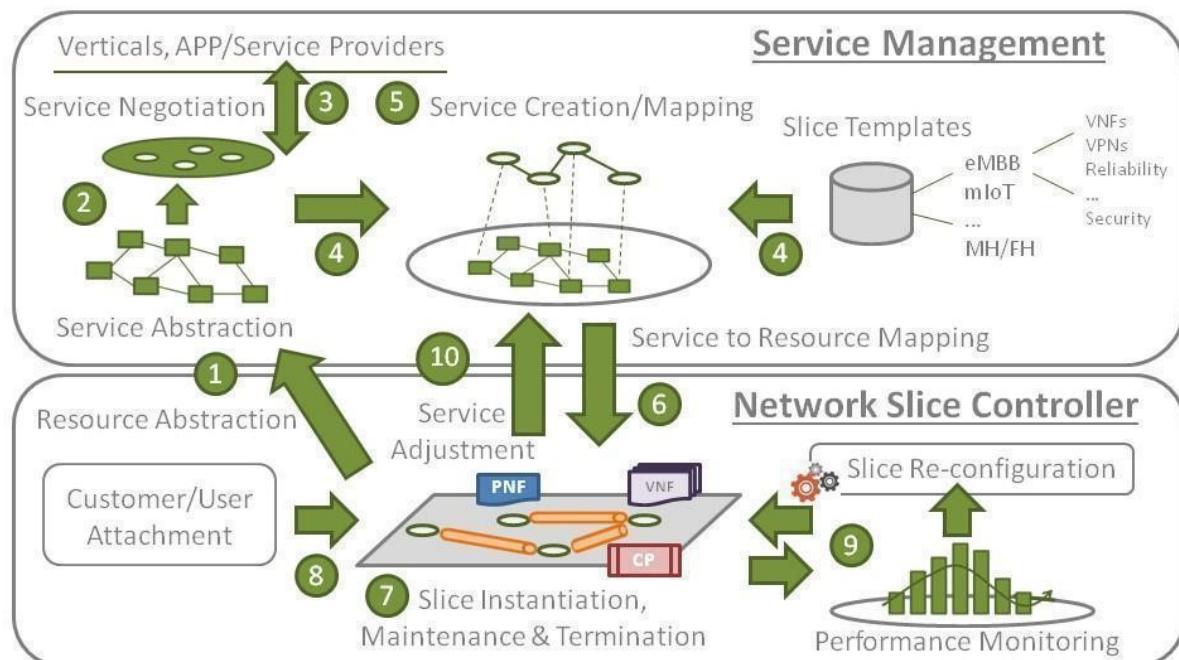


FIGURE 1.4 – Gestion de Service et contrôle de Slices (Afolabi et al., 2018).

Couche de management de service

Cette couche gère les opérations de service comme l'abstraction, négociation et contrôle d'admission des verticaux ainsi que des tierces parties (Etapes 2 et 3), mais aussi la création de service lorsque la demande de création de Slice est approuvée, selon les besoins de celui-ci (Etapes 4 et 5). Le service demandé est fourni à travers des VNF et PNF (Fonctions réseau physiques) exposées sur la couche basse (Etape 6).

Couche de contrôle de Network Slices

Cette couche fournit une abstraction de ressources pour la gestion des services (Etape 1) et gère la gestion des ressources des Slices comme l'instanciation des ressources Slices pour le mapping de Services (Etape 7) et la maintenance de la performance par l'analyse des métriques et la procédure de reconfiguration (Etape 9) et la sélection de Slices, l'attachement et le support pour la connectivité Multi-Slice (Etape 8). Cette couche agit donc après la création de Slices, tandis que la première couche sert à instancier les Slices sur le réseau.

Après avoir abordé le Network Slicing en détail, nous passons à un autre concept récent de la 5G, qui permet l'optimisation de la gestion des réseaux, intitulé le Software Defined Networking.

1.4 Software Defined Networking

Tout comme le Network Slicing, l'architecture SDN s'est vite affirmée comme étant un potentiel moteur pour la 5G. Nous décrivons cette architecture et ses avantages dans ce qui suit.

1.4.1 Définition

Les architectures réseau traditionnelles sont en même temps complexes et difficiles à gérer. Une des raisons pour cela est que les données de contrôle et des utilisateurs ne sont pas séparées sur le réseau et transitent de la même façon sans distinction entre elles. Pour pallier ce problème, une architecture réseau appelée SDN a vu le jour, dont l'objectif est de permettre de gérer l'ensemble du réseau à travers un programme central, tout en séparant les données utilisateur et de contrôle (Tijare et Vasudevan, 2016).

(Kim et Feamster, 2013) eux définissent le SDN comme étant un paradigme réseau, où un programme central gère et contrôle le comportement global d'un réseau. De par ce fait, les équipements réseau deviennent de simples transmetteurs de données (Data Plane), tandis que la logique de contrôle du réseau est implémentée dans le programme

central qui est aussi appelé contrôleur (Control Plane).

(Kreutz et al., 2014) quant à eux définissent un réseau SDN selon quatre critères :

- Les flux de données et de contrôle sont séparés. La capacité de contrôle est retirée des équipements réseau.
- Les décisions de transfert au sein du réseau sont basées sur orientées flux plutôt que destination. C'est-à-dire que l'on s'intéresse plus au flux qui transite qu'à la destination des paquets. Un flux étant une succession de paquets entre une source et destination.
- La logique de contrôle est transférée vers une entité externe appelée Contrôleur SDN.
- Le réseau est programmable par le biais d'applications s'exécutant au-dessus du contrôleur.

1.4.2 Architecture

On peut distinguer dans l'architecture SDN trois couches fondamentales : La couche Infrastructure, la couche Contrôle et la couche Application comme l'indique la figure 1.5 :

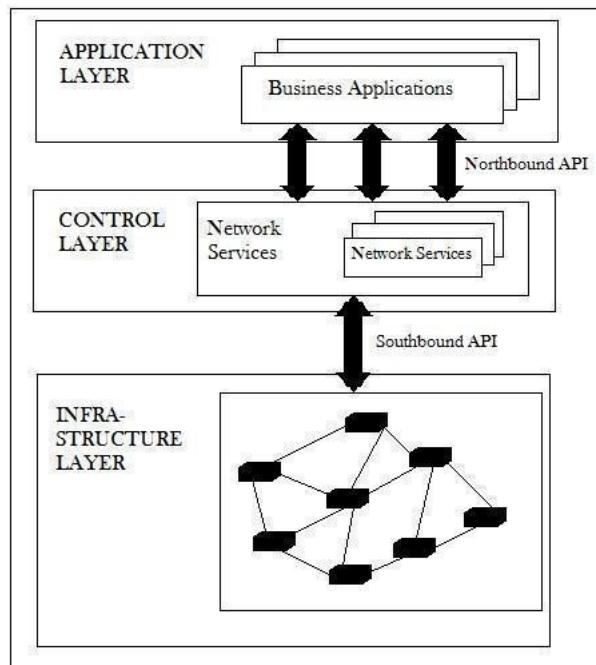


FIGURE 1.5 – Architecture SDN (Tijare et Vasudevan, 2016).

La couche la plus basse est la couche Infrastructure, responsable de l'acheminement des flux de données. Elle contient donc les équipements réseau de transmission. La couche intermédiaire représente la couche Contrôle. Elle est responsable de la programmation et du management du transfert des données dans la couche plus basse. Pour ce faire, elle récupère depuis la couche Infrastructure des informations de routage et de transmissions par le biais d'un ensemble d'interfaces standardisées, plus communément appelé API Sud à cause de sa position dans l'architecture. La couche Application quant à elle contient des applications qui peuvent utiliser les équipements réseau de la couche la plus basse, en utilisant la couche intermédiaire à travers d'autres méthodes standardisées appelées quant à elle API Nord. Ces applications réseau peuvent parfois même apporter de nouvelles caractéristiques au réseau, comme la sécurité ou de la configuration, ou peut même assister la couche Contrôle dans son management. (Braun et Menth, 2014). La figure 1.6 représente l'architecture de façon plus détaillée :

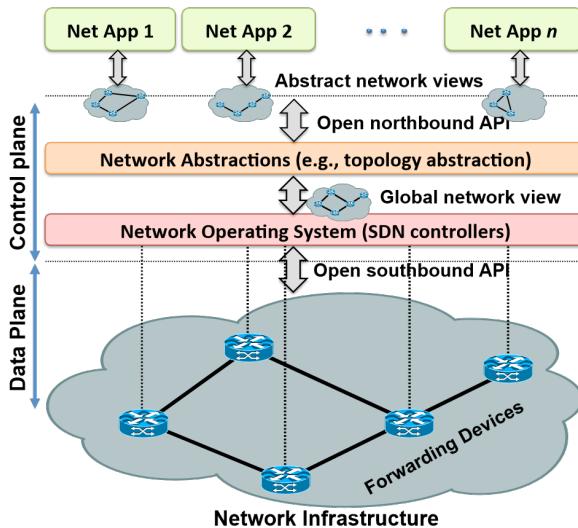


Fig. 4. SDN architecture and its fundamental abstractions.

FIGURE 1.6 – Architecture détaillée SDN (Kreutz et al., 2014).

1.4.3 API Nord/Sud

L'architecture SDN est une évolution de l'informatique en réseau basée sur la suppression des limites physiques existantes sur les commutateurs, les routeurs et les contrôleurs, et la suppression de couplage étroit existant entre ces équipements et la logique de contrôle via des API bien définies. Une interface de programmation d'application (API) est utilisée pour définir l'interaction logicielle entre les systèmes (Fishnet Security, 2014). En SDN, ces « systèmes » font référence aux applications réseau et au matériel tels que les routeurs, les commutateurs, etc. Cette capacité de programmation est ce qui donne aux API la flexibilité contrairement aux autres échanges de systèmes. Parallèlement à cela,

leur simplicité et leur efficacité sont ce qui le rend inestimable pour les SDN. Les API permettent donc essentiellement à un équipement réseau d'envoyer une demande ou de répondre à des requêtes (Tijare et Vasudevan, 2016).

Dans l'architecture SDN, il existe deux types d'API. La première est l'API Sud, qui assure la communication entre la couche Infrastructure et la couche Contrôle. Elle permet une gestion du réseau, en conférant au Contrôleur SDN le pouvoir d'effectuer des changements en temps réel sur les équipements réseau se trouvant dans la couche physique, et de la même façon à la couche physique de remonter des informations qui permettent le suivi des performances du réseau. OpenFlow est une initiative lancée par Open Networking Foundation (OFN) pour développer une API Sud standard, qui fonctionne sur des équipements propres à elle comme des switches OpenFlow (McKeown et al., 2008).

La deuxième API est l'API Nord. Elle est responsable de la communication entre la couche Application et la couche Contrôleur. On peut la voir donc comme le point d'entrée de l'architecture SDN, étant donné que toute application réseau voulant être déployée sur le réseau devra interagir avec cette API. Contrairement à l'API Sud, elle ne possède pas à ce jour de standard semblable à OpenFlow.

Nous nous intéresserons dans ce qui suit à l'API Nord, étant donné qu'elle est l'objet principal de notre stage.

1.4.4 API Nord

Dans l'architecture SDN, le lien avec les applications utilisant l'infrastructure physique se fait à travers l'API Nord. Son importance n'en est donc que plus grande ; si bien que selon SDX Central¹, elle est l'entité la plus importante dans l'architecture SDN. Dans ce qui suit nous définissons l'API Nord et relatons ses différents types.

Définition

Les nombreux avantages de SDN sont infuctueux si les applications ne peuvent en bénéficier. L'adoption du SDN dépend sur sa capacité à prendre en charge un large éventail d'applications. Les API Nord (North Bound Interface NBI) jouent un rôle intégral pour les développeurs d'applications et fournissent une interface commune entre les contrôleurs et plan de gestion. Elles aident à fournir les informations concernant les dispositifs sous-jacents pour le développement d'applications qui pourront interagir avec le Contrôleur SDN (Latif et al., 2019). L'importance de ces API est due au fait que la valeur d'une architecture SDN, comme dit précédemment, est liée aux types d'applications qu'elle peut

1. www.sdxcentral.com

supporter. Le contrôleur ne doit pas être une application monolithique ne fonctionnant que sur un type précis d'applications. Il doit pouvoir s'interfacer avec différents systèmes et permettre le suivi des mesures de performance réseau, et permettre même offrir même aux applications l'orchestration du réseau (Tijare, 2016).

Types

Selon la façon d'implémenter l'API Nord, on peut avoir affaire à deux types principaux :

1. **REST API** : Representational State Transfer (REST) API désigne toute API respectant les contraintes de l'architecture REST. Ces contraintes ont été définies par (Fielding, 2000) :
 - **Débuter par le style NULL** : Afin de commencer la conception d'une architecture, le concepteur commence par un ensemble vide et construit l'architecture en enrichissant cet ensemble à partir de composants familiers, jusqu'à ce cela satisfasse les besoins du système escompté.
 - **Client-Server** : La séparation de l'interface utilisateur du stockage de données, la portabilité de l'interface utilisateur est améliorée à travers plusieurs plateformes, ainsi que la scalabilité en simplifier le traitement au niveau du Serveur.
 - **Sans état** : La communication se doit d'être sans état, car chaque requête doit comporter toutes les informations nécessaires pour être comprise par le serveur, sans se servir de contexte stocké au niveau de celui-ci.
 - **Cache** : Afin d'améliorer l'efficacité du réseau, une mémoire cache peut être ajoutée au niveau du client, afin de stocker certaines réponses du Serveur, qui peuvent être utilisées dans le futur pour les mêmes requêtes.
 - **Interface uniforme** : L'architecture REST se distingue des autres architectures par le fait qu'elle uniformise les interfaces entre les composants. De par ce fait, l'architecture s'en trouve grandement simplifiée et la visibilité des interactions en est augmentée.
 - **Système en couches** : L'architecture en couches permet à chaque composant de n'interagir et de n'avoir de visibilité qu'avec la couche suivante, ce qui engendre une meilleure gestion de la sécurité de l'application en implémentant des

politiques de parcours de données à travers l'architecture entre chaque couche.

- **Code à la demande :** L'architecture REST permet au Client la fonctionnalité d'être étendue en téléchargeant et exécutant des codes sous forme de scripts, ce qui permet aux applications Client d'être plus simplifiées et de ne pas avoir besoin d'un grand nombre de fonctionnalités pré-implémentées.

Les avantages d'une telle architecture sur le paradigme SDN sont selon (Zhou et al., 2014) le fait qu'elle permette à des éléments de réseau comme les routeurs, switches... etc. d'être déployés dynamiquement, mais aussi le fait que REST soit flexible, c'est-à-dire que l'API Nord, étant susceptible d'être fréquemment mise à jour, les modifications ne toucheront que les composants directement concernés, vu que l'architecture sépare bien entre les composants. Quelques exemples d'API Nord de type REST sont : DISCO (Phemius et al., 2014), Floodlight²... La figure 1.7 montre le fonctionnement d'une architecture REST.

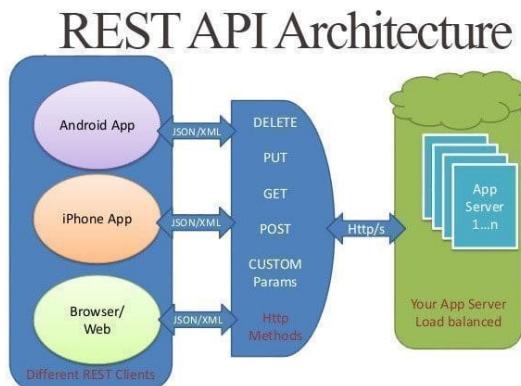


FIGURE 1.7 – Architecture REST (Fiverr).

2. **Langage de programmation :** Un autre type d'API Nord offert par les contrôleurs SDN sont sous la forme de langages de programmation à part entière (Sezer et al., 2013). L'intérêt d'une telle solution est qu'elle permet une abstraction plus haut-niveau, mais aussi la facilité qu'ont les développeurs à la mettre en place, car cela leur permet de résoudre des problématiques directement reliées à l'environnement en question. Les langages de programmation en SDN peuvent être comparés sur leur niveau d'abstraction : Les langages qui agissent plutôt en bas niveau pousseront les développeurs à traiter des problèmes directement reliés à l'API Sud, tandis que plus le langage sera haut niveau poussera les développeurs à se concentrer sur les problématiques plus générales.

2. www.floodlight.atlassian.net/wiki/spaces/floodlightcontroller/overview

Parmi les langages de programmation SDN les plus connus, nous pouvons citer Procera (Voellmy et al., 2012). C'est un langage de contrôle réseau haut niveau, qui permet aux opérateurs de réseaux de mettre en place des politiques de contrôle réseau réactives : c'après certaines règles déclaratives que le système appliquera lorsqu'une condition sera vérifiée. Cela permet à l'opérateur de garder la main sur l'état de son réseau et de réagir très vite lorsqu'un changement est détecté. La figure 1.8 détaille l'architecture dans laquelle Procera fonctionne.

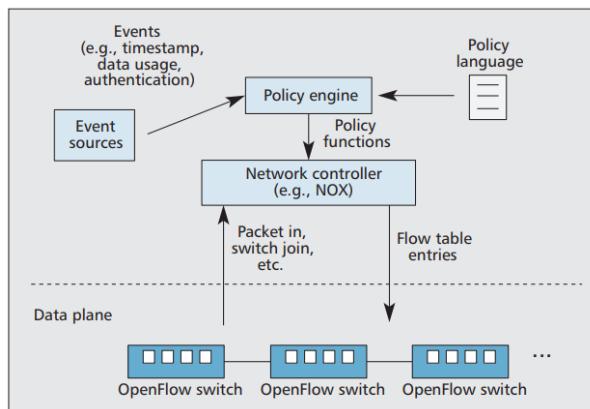


FIGURE 1.8 – Architecture Procera (Kim et Feamster, 2013)

Tout d'abord, un évènement quelconque (Horodatage, authentification...) déclenche l'exécution du système de gestion de politique, qui lui tirera les règles mises en place directement d'un catalogue de règles. Ensuite, en fonction de l'évènement et des règles mises en place, le système applique des fonctions qui seront transférées au contrôleur réseau, qui lui les transmettra à l'API Sud, qui elle les transmettra à la couche physique (OpenFlow Switches). Il existe une multitude d'autres langages de programmation similaires à Procera : Frenetic (Foster et al. ,2011), Pyretic (Reich et al., 2013) ... etc.

Dans la section suivante, nous parlons du concept d'orchestration réseau étant donné que l'objet principal de notre stage de fin d'études est le développement d'un orchestrateur de Slices.

1.5 Orchestration réseau

Avec l'émergence des avancées technologiques telles que la 5G, les réseaux SDN et les fonctions réseau virtualisées (NFV), le besoin d'orchestration s'est rapidement fait ressentir au sein des fournisseurs de services réseau. La virtualisation a d'ailleurs largement contribué à faciliter l'orchestration, car parmi les intérêts de la virtualisation se trouvent la gestion décentralisée et l'isolation. Les outils d'orchestration et de gestion de plateformes

réseau doivent être capables de mesurer les performances du réseau, de stockage, mais aussi les aspects sécurité. Des efforts de recherche et de développement sont consacrés à l'automatisation des processus d'orchestration, coordonner et gérer le déploiement et l'exploitation des services de réseau.

Nous nous intéressons dans cette partie à quelques solutions d'orchestration existantes sur le marché.

1.5.1 Solutions Open Source

Les fondations Open Source telles que Apache et Linux deviennent de plus en plus les entités d'hébergement pour de grands projets Open Source collaboratifs dans le domaine des réseaux. Plusieurs projets ont pour but de créer une plate-forme de services d'orchestration. Il est nécessaire noter que pour les réseaux 5G, la normalisation et l'open source sont essentiels pour une innovation rapide. Les vendeurs, les opérateurs et les communautés parient sur l'Open Source (Katsalis et al., 2016). Malgré cela, les solutions existantes ne sont toujours pas suffisamment matures, et des plates-formes d'orchestration de services réseau avancés sont manquantes (De Sousa et al., 2019). Toutefois, certaines initiatives sont en déploiement en collaboration entre les fournisseurs de VNF et les opérateurs réseau, dont voici quelques unes :

Open Source MANO

ETSI Open Source MANO (Open Source NFV Management and Orchestration)³ est un projet hébergé par l'ETSI (European Telecommunications Standards Institute) pour développer une plate-forme Open Source NFV-MANO alignée sur les modèles d'information NFV de l'ETSI et qui répond aux exigences des réseaux NFV de production. Le projet a lancé sa quatrième version (ETSI, 2018) en mai 2018 et a présenté des améliorations des capacités de la logique de modélisation et de mise en réseau. En outre, cette version prévoit une installation native dans le Cloud et une nouvelle API Nord, alignée sur la spécification ETSI NFV SOL005 (ETSI, 2018). L'architecture OSM présente une répartition claire de la fonction d'orchestration entre Resource Orchestrator et Service Orchestrator. L'orchestrateur de ressources prend en charge les environnements Cloud et SDN. L'orchestrateur de services assure la gestion du cycle de vie des VNF et des Slices et consomme des modèles d'information/données ouverts. Son architecture ne couvre qu'un seul domaine administratif.

3. <https://osm.etsi.org>

Cloudify

Cloudify⁴ est un outil d'orchestration centré sur l'optimisation de l'orchestration et de la gestion des NFV. Il fournit un NFVO (NFV Orchestrator) et un VNFM (VNF Manager) générique dans le contexte du NFV de l'ETSI, et peut interagir avec différents VIM (Virtual Infrastructure Manager), qui est l'infrastructure sur laquelle les VNFs sont instanciées, conteneurs et dispositifs et infrastructures non virtualisés. En outre, Cloudify fournit un cycle de vie complet de bout en bout de l'orchestration NFV grâce à un plan suivant une approche orientée modèle et centrée sur les applications. Il inclut l'implémentation de référence agile de l'automatisation comme moteur d'orchestration de base fournissant une gestion avancée et une automatisation continue.

Gohan

Gohan (NTT, 2015)⁵ est une initiative de MANO pour l'orchestration de SDN et NFV. L'architecture Gohan est basée sur des micro-services au sein d'un processus unique afin de garder l'architecture du système et le modèle de déploiement simples. Une définition de service Gohan utilise un schéma JSON (à la fois définition et configuration des ressources). Il comprend un serveur API basé sur REST, une base de données, une interface de ligne de commande (CLI) et une interface utilisateur web (Web UI). Enfin, Gohan peut être utilisé (i) dans le catalogue de services et la couche d'orchestration en plus des services de Cloud Computing et (ii) comme une sorte de MANO NFV qui gère à la fois le VIM de Cloud Computing et les dispositifs de réseau existants.

ONAP

Sous la bannière de la Linux Foundation, l'Open Network Automation Platform (ONAP)⁶ est le résultat de l'union de deux initiatives Open Source MANO (OPEN-O⁷ et OpenE-COMP⁸). La plateforme logicielle ONAP déploie une architecture unifiée et avec de solides capacités pour la conception, la création, l'orchestration, la surveillance et la gestion du cycle de vie des fonctions de réseaux physiques et virtuels. De plus, les fonctionnalités ONAP devraient permettre un déploiement et une gestion automatisés ainsi qu'une optimisation des politiques de gestion grâce à une exploitation intelligente des ressources du réseau à l'aide de données volumineuses et d'une intelligence artificielle (AI) (Converge Network Digest, 2017). ONAP est actuellement soutenu et encouragé par les plus grands opérateurs de réseaux et de Cloud computing et par les fournisseurs de technologie du monde entier (Cloudify Onap, 2018). Par conséquent, ONAP peut être utilisé pour conce-

4. <http://cloudify.co/>

5. <http://gohan.cloudwan.io/>

6. Linux Foundation, ONAP – Open Network Automation Platform, 2017, <https://www.onap.org/>

7. Linux Foundation, Open Orchestrator, <https://www.open-o.org/>

8. Linux Foundation, ONAP Developer Wiki, 2017, <https://wiki.onap.org/>

voir, développer et mettre en œuvre des services de réseau dynamiques sur le réseau du fournisseur de services et/ou dans son propre Cloud.

Le tableau 1.2 résume les principales caractéristiques des chaque projet open source. Les informations contenues dans ce tableau sont organisées comme suit : entités leaders, définition VNF, domaines de ressources (Cloud, SDN, NFV, Legacy), blocs fonctionnels NFV MANO implémentés (NFVO, VNFM, VIM ou support VIM), interface de gestion (CLI, API, GUI), et le champ d'application (mono/multi-domaine). (De Sousa et al., 2019)

| Solution | Leader | Domaine | | | | Interface | | |
|----------|------------------|---------|-----|-----|--------|-----------|-----|-----|
| | | Cloud | SDN | NFV | Legacy | CLI | API | GUI |
| ONAP | Linux Foundation | X | | X | | X | X | X |
| Gohan | NTT Data | X | X | X | X | X | X | X |
| OSM | X | X | X | | X | X | X | X |
| Cloudify | GigaSpace | X | | X | | X | X | X |

TABLE 1.2 – Solutions Open Source d'orchestration réseau (De Sousa et al., 2019)

1.5.2 Solutions commerciales

Le marché des solutions d'orchestration commerciales est en hausse, et est constitué de divers types d'entreprises, de fournisseurs de services informatiques, de fournisseurs de VNF et de vendeurs d'équipements réseau traditionnels (Cisco Inc, 2019). Voici quelques uns des fournisseurs majeurs :

Cisco Network Services Orchestrator

Cisco propose un produit appelé Network Services Orchestrator⁹. Il s'agit d'une plateforme d'orchestration qui permet d'automatiser le cycle de vie des services pour les réseaux hybrides (c'est-à-dire multifournisseurs). Cisco NSO permet de concevoir et de fournir des services plus rapidement et propose une orchestration de bout en bout sur plusieurs domaines. La plateforme déploie certaines fonctions de gestion et d'orchestration telles que l'orchestration des services réseau, Orchestration Cloud, NFVO et VNFM.

Oracle Communications Network Service Orchestration

La solution Oracle Communications Network Service Orchestration¹⁰ orchestre, automatise et optimise les VNF et le réseau la gestion du cycle de vie des services par l'intégration avec les BSS/OSS (Systèmes de gestion d'un opérateur télécom), les portails

9. www.cisco.com

10. www.oracle.com

de services et les orchestrateurs. Elle dispose de deux environnements pour déployer les services réseau : un environnement de conception pour concevoir, définir et programmer les capacités, et un environnement d'exécution pour exécuter la logique programmée et la gestion du cycle de vie.

HPE Service Director

Une autre solution commerciale est le Service Director HPE de la Hewlett Packard Enterprise. Le produit est une solution OSS d'orchestration de services qui gère le service de bout en bout et fournit une planification basée sur l'analyse et une automatisation en boucle fermée en utilisant un modèle de service basé sur les déclarations. Il prend en charge les VNF multi-fournisseurs, les VIM multi-fournisseurs, diverses versions d'OpenStack et plusieurs contrôleurs SDN.

Ericsson Network Manager

Ericsson propose quelques solutions dans le domaine du cloud, du SDN et de l'orchestration. L'une d'entre elles est le gestionnaire de réseau Ericsson¹¹ qui fournit un réseau unifié multicouche et multi-domaine (SDN, NFV, radio, transport et de base) et joue divers rôles tels que la VNFM, le Network Slicing et l'analyse du réseau. Un autre produit est l'Ericsson Orchestrator qui prend en charge l'orchestration des ressources, la gestion du cycle de vie des VNF et l'orchestration des services de bout en bout. Il utilise l'architecture ETSI NFV-MANO comme référence, jouant le rôle de NFVO, VNFM et SO.

1.6 Conclusion du premier chapitre

Nous avons vu dans ce chapitre les spécifications techniques de la 5G, son architecture, ainsi que les technologies majeures qui y sont utilisées. En effet, le Network Slicing et le Software Defined Networking sont des concepts extrêmement importants pour permettre à la 5G d'atteindre son potentiel escompté. Nous avons aussi présenté un bref comparatif entre les solutions d'orchestration réseau existantes sur le marché. Dans le prochain chapitre, nous parlons des drones de façon détaillée, sur le plan applicatif, ainsi que sur le plan de la connectivité.

11. www.ericsson.com

Etat de l'art sur les réseaux de Drones 5G

L'application de véhicules aériens sans pilote (Unmanned Aerial Vehicles), ou drones volants, a été lancée dans un large contexte d'activités civiles et commerciales, telles que la livraison de marchandises, l'agriculture précise, l'inspection à distance, la photogrammétrie, le vidéo streaming, et la collecte de données pour Internet des Objets (IoT). Certains changements réglementaires, les progrès technologiques, et la réduction du coût des capteurs ainsi que des unités de contrôle à bord ont rendu les drones plus accessibles aux consommateurs et aux application civiles. Il existe un large consensus dont les domaines d'activité s'étendront considérablement dans les années à venir. Pour l'Europe en particulier, le Single European Sky ATM Research¹ indique qu'il existe un potentiel sur le marché européen dépassant les 10 milliards d'euros par an d'ici 2035, et 15 milliards d'euros d'ici 2050, identifiant les services utilisant des drones (Drone-as-a-Service) comme la plus grande opportunité de marché des années à venir.

Nous commençons par donner une définition des drones, leurs divers domaines d'application et leurs besoins respectifs surtout en termes de connectivité. Nous détaillons ensuite l'architecture de UTM, qui est une entité globale permettant la gestion de vols de drones multiples de façon simultanée, pour assurer la sécurité des vols. Nous terminons en évoquant certaines recherches effectuées dans le domaine de l'optimisation de vols de drones.

2.1 Définition

Les véhicules aériens sans pilote (UAV), communément appelés drones, sont des véhicules aériens pilotés à distance. Ils vont des simples systèmes manuels à courte portée aux longs systèmes volant en haute altitude et nécessitant une piste d'atterrissement (Brooke-Holland, 2013).

Les UAV sont de petits avions volants sans pilote humain à bord. Ils peuvent être

1. (SESAR, Single European Sky ATM Research <https://www.sesarju.eu/>)

commandés manuellement à l'aide d'une télécommande ou de manière autonome à l'aide d'un pilote automatique intégré. Néanmoins, dans les deux cas, les UAV sont exploités à l'aide d'un système d'aéronef sans pilote (Unmanned Aerial System UAS) qui se compose d'une station de contrôle au sol GCS (Ground Control Station), de drones et d'une technologie de communication entre eux pour échanger des messages de commandement et de contrôle (C2). En outre, les opérations des UAV sont réglementées et contrôlées par les agences de sécurité aérienne telles que la Federal Aviation Administration (FAA) aux États-Unis et l'Agence européenne de la sécurité aérienne (EASA), qui mènent de vastes projets de recherche et développement pour l'innovation et le développement de nouveau systèmes pour les drones, qui permettent l'intégration des drones dans l'espace aérien à basse altitude.

Les drones sont principalement utilisés pour leur petite taille, et leur habileté à se déplacer dans des endroits étroits (par exemple les zones urbaines), ce qui rend leurs opérations critiques et vitales. Il y a aussi une énorme demande de drones plus gros pouvant transporter de lourdes charges, et pouvant voler sur de longues distances. Ces drones doivent être capables de transporter des charges utiles telles que le fret ou de lourds capteurs. Le temps de vol de ces drones peut varier entre 15 minutes à 2 heures, et la distance de vol peut varier entre 2 km et 20 km.

2.2 Domaines d'application

Le champ d'application des drones est tellement vaste et épars que la classification des domaines d'application reste une tâche difficile. En effet, ceux-ci sont sujets à différentes demandes et divers objectifs. (Hayat et al., 2016) proposent de les classifier en quatre catégories : Recherche et sauvetage (Search and Rescue SAR), Couverture (Coverage), Transport de biens (Delivery of goods) et Construction. Cette classification a pour objectif d'englober la majeure partie des applications de drones, même s'il existe forcément quelques applications qui ne sont pas comprises dans cette classification, car cela s'avère être une tâche quasi-impossible. L'intérêt de cette classification est qu'elle permet d'avoir divers domaines d'application dont les besoins en termes de communication diffèrent : Par exemple pour le domaine de la couverture, les drones peuvent être perçus comme des outils de relais réseau, ce qui n'est pas le cas pour la construction. La figure 2.1 illustre ces domaines d'applications ainsi que leurs besoins en termes de connectivité, où les traits continus indiquent un besoin primordial, et les traits discontinus signifient un besoin optionnel.

Nous détaillons dans ce qui suit chacune des catégories précédemment citées :

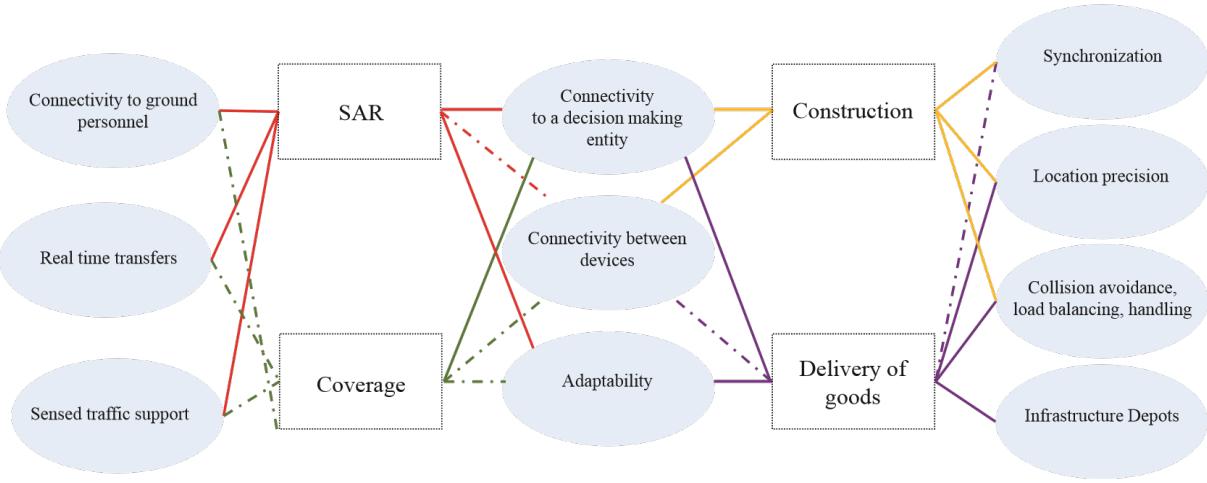


FIGURE 2.1 – Domaines d’application UAV (Hayat et al., 2016).

2.2.1 Recherche et Rescousse

C'est parmi les domaines d'application les plus prisés des drones. Dans une mission de recherche et rescousse, les drones sont censés rechercher et détecter une ou plusieurs cibles stationnaires ou en mouvement, et traquer l'évolution de leur position et ce afin de faciliter aux équipes de sauvetage leur opération (Chmaj et Selvaraj, 2015). Les drones peuvent s'avérer très utiles dans certains cas, par exemple dans le cas d'avalanches de neige, où l'espérance de survie des victimes sous la neige est de 20 minutes selon (Ackerman, 2014), les équipes au sol sont très limitées en temps pour localiser les victimes dans un premier temps, puis pour réagir. Tandis que si l'on utilise des drones, la localisation peut se faire en un temps record.

Certaines méthodologies ont été mises en place pour ce genre de situation. (Pack et al., 2009) considèrent la possibilité de localiser diverses cibles et proposent l'utilisation d'un centre de décision décentralisé au niveau de chaque drone afin de mener à bien la mission. Les décisions sont prises au niveau de chaque drone selon les données reçues à leur niveau à partir de leurs capteurs. Les approches décentralisées s'avèrent plus efficaces lorsque les zones d'action sont inaccessibles et sujettes à des changements suite à des catastrophes naturelles (Hayat et al., 2016). Toutefois, certaines approches centralisées ont été proposées, comme (Xargay et al., 2013), où les coûts de calcul à effectuer au niveau des drones est moindre.

2.2.2 Couverture

C'est l'un des domaines les plus vastes dans les applications de drones. Il comporte deux sous-domaines majeurs, qui sont la couverture de zones qui peut servir à la surveillance ou au monitoring, et la couverture réseau, où les drones servent de relais ou de « mules »

afin d'assurer la connectivité à d'autres équipements. Le tableau 2.1 (Hayat et al., 2016) relate une liste d'applications relatives aux deux types de couverture énoncés :

| | Petite zone | Moyenne zone | Grande zone |
|--------------------|---|---|--|
| Couverture de zone | Monitoring de surface (Eschmann et al., 2012), Mapping de site archéologiques (Cappellini et al., 2013), Inspection de ponts (Hallermann et al., 2007) | Monitoring de végétation (Rango et al., 2009), Photographie aérienne (Eisenbiss, 2008), Couverture d'évènements (Corcoran, 2014), Analyse de sites volcaniques (McGonigle et al., 2008) | Surveillance d'autoroutes (Srinivasan et al., 2004), Surveillance de frontières (Freed et al., 2005) |
| Couverture réseau | Couverture de réseau dans une zone accidentée (Burdakov et al., 2010), Relais de communication pour les véhicules autonomes sous-marins (Johansen et al., 2014) | Couverture réseau temporaires pour grand public (Guo et al., 2014) | Couverture réseau d'urgence après catastrophe (Ono et al., 2013) |

TABLE 2.1 – Couverture de drones (Hayat et al., 2016)

Pour les scénarios de couverture de zones étendues tels que la surveillance et la cartographie, les drones peuvent offrir une solution rapide. Par exemple, l'efficacité du temps, l'efficacité du traitement et la main-d'œuvre sont des métriques utilisées comme mesures par (Darwin et al., 2014) pour illustrer les avantages potentiels de l'utilisation des systèmes d'UAV par rapport aux techniques conventionnelles . Le rôle des drones est également crucial dans l'approvisionnement de la couverture réseau, comme en cas de catastrophe naturelle où l'infrastructure du réseau terrestre est perdue, les appareils aériens peuvent être le seul moyen de maintenir la communication pour faciliter les premières interventions des équipes au sol (Reynaud et al., 2011). Dans ce dernier cas, les drones sont équipés de cellules réseau, offrant de la connectivité aux personnes au sol en proximité du drone.

2.2.3 Transport de biens

Le transport de marchandises a attisé l'intérêt de plusieurs firmes internationales comme « Amazon » mais aussi quelques startups comme « Matternet » qui utilisent les drones pour le transport de premiers soins (kits médicaux) dans des situations d'urgence (catastrophe naturelle...). L'objectif est le gain en temps, en coût de main d'œuvre, et en pollution. En effet, les drones volent sur batterie. Si les batteries sont rechargées par des systèmes d'énergie renouvelable, comme l'énergie solaire, le vol est sans pollution. Le seul inconvénient de l'utilisation des drones est qu'ils font un bourdonnement assez

contraignant : Ce bruit n'est pas très perceptible lorsqu'un seul drone vole dans les airs ; cependant, un drone est assez bruyant lorsqu'il est à proximité. À l'avenir, il se peut que des couloirs de vol de drones trop actifs bourdonnent constamment. Afin de pallier ce problème, le chercheur Edge Herrera a conçu un drone sans hélice, qui se déplace grâce au vent et en générant un flux d'air conducteur (figure 2.2).



FIGURE 2.2 – Drone UAV sans hélices (Designboom, 2017).

2.2.4 Construction

La construction à l'aide de robots aériens est un nouveau domaine d'application des drones, qui consiste en l'utilisation de petits robots aériens soulevant soulèvent les éléments de construction et les positionnant à leur emplacement précis. Il est prévu que la construction à l'aide de robots aériens puisse être moins coûteuse et plus efficace puisque ces derniers peuvent voler dans des endroits étroits et placer des blocs de construction dans des endroits habituellement difficiles d'accès avec les machines de construction habituelles (Willman et al., 2012). En revanche, le nombre de drones à utiliser varie en fonction de la capacité de charge utile du robot et le poids des blocs à transporter. Il est donc très possible que plus d'un UAV soit nécessaire pour transporter les diverses structures, telles que les poutres ou les blocs de construction.

Pour ce genre d'applications, la difficulté réside dans la synchronisation entre les drones dans le cas où ceux-ci sont multiples. En effet, faire déplacer les drones ensemble de sorte à ce qu'un objet soit placé dans un endroit bien précis est encore un problème de recherche à ce jour, car plusieurs difficultés liées à l'équilibre des charges dans le cas où plusieurs drones portent un objet, peuvent apparaître, comme évoqué dans (Augugliaro et al., 2014). Le maintien de la séquence correcte des éléments constitutifs est également un paramètre à prendre en considération. Les robots enregistrent le nombre de blocs et leur disposition disponibles dans la zone, c'est-à-dire ceux qui ont déjà été placés et où placer le prochain. En définitive, pour l'application de construction utilisant des robots

aériens, la planification de la trajectoire, l'évitement des collisions, la localisation, la coordination, la synchronisation et le séquençage sont les exigences générales à prendre en compte.

Certains travaux, comme dans (Alejo et al., 2014) effectuent la synchronisation entre les drones de façon centralisée, où le mouvement et la localisation de chaque drone sont envoyées à l'unité centrale qui effectue le calcul des trajectoires, et qui envoient en fonction de celles-ci des commandes de contrôle aux drones évitant les conflits, comme l'indique la figure 2.3 :

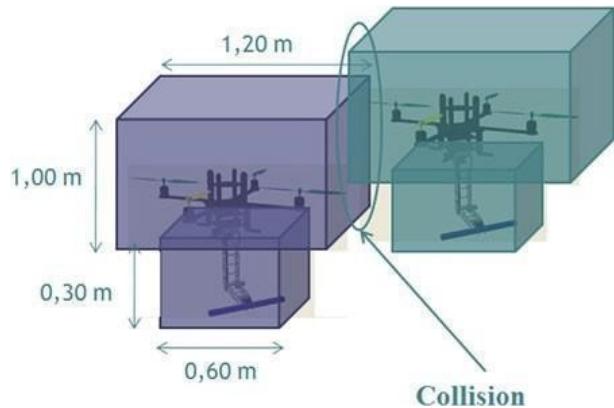


FIGURE 2.3 – Algorithme de détection de collisions (Alejo et al., 2014).

2.3 Besoins

Afin de mener à bien une mission de drones, il est également nécessaire de satisfaire les exigences qualitatives liées à la communication dans les réseaux aériens, qui dépendent des catégories d'application identifiées précédemment. La Figure 2.1 illustre la relation entre les applications et les exigences qualitatives. Observez que la relation peut être obligatoire (lignes pleines) ou facultatives (lignes pointillées) en fonction de l'application. Le lien est établi entre ces exigences obligatoires et la communication en précisant le trafic qui doit être échangé en tant que résultat de ces exigences qualitatives. Compte tenu des contraintes, des hypothèses et des exigences qui ont été identifiées à partir des travaux existants, il s'avère que pour chaque application, il existe différentes solutions possibles. Nous listons dans ce qui suit les différents besoins à prendre en considération :

2.3.1 Connectivité

Selon (Hayat et al., 2016), la connectivité peut être répartie en trois catégories. Premièrement, la demande de connectivité à une entité de prise de décision DME (Decision

Making Entity) est une nécessité dans tout réseau de drones, car les trajectoires des drones doivent être suivies à tout moment pour des raisons de sécurité. Deuxièmement, les drones peuvent être tenus de se connecter à certains clients au sol, par exemple le personnel au sol en cas des situations de catastrophe. Enfin, la connectivité entre les drones pendant les drones.

Connectivité avec les DME

La connectivité avec une DME peut être considérée comme une exigence générale quel que soit le scénario, en particulier lorsque plusieurs drones doivent être coordonnés pour une mission. Un DME est défini comme un appareil qui peut suivre la trajectoire de vol des drones dans le système et peut intervenir en cas d'urgence, par l'envoi de commandes de contrôle. Ainsi, les drones doivent être connectés à DME à tout moment. La DME peut être une station de commande manuelle à distance RC (Remote Control) ou une ou plusieurs stations de base au sol. La tâche minimale que ces DME se doivent de pouvoir accomplir est de permettre de suivre le vol des drones et interférer dans le déroulement normal de la mission (par exemple, pour un atterrissage d'urgence en utilisant le RC en cas de batterie faible). En outre, ces DME peuvent également agir comme coordonner les entités, décider du plan de mission pour tous les appareils dans le réseau. (Hayat et al., 2016).

Pour la recherche et la rescouasse, les scénarios de couverture et le transport de biens, il existe souvent une structure hiérarchique qui effectue la répartition des tâches. Cela signifie qu'une ou plusieurs entités du réseau peuvent être responsables de la prise de décision alors que d'autres ont la responsabilité de mener à bien les missions qui leur sont confiées par les DME, telles que la récupération de données par capteurs, comme c'est le cas dans (Olsson et al., 2010). Dans ce genre de scénarios, les drones maintiennent la connectivité à la DME, qui est chargé de coordonner les appareils. Les commandes de contrôle et de coordination sont envoyées par celle-ci suggérant des changements dans la trajectoire prévue afin d'éviter les collisions, d'interrompre la mission, ou effectuer le déplacement des drones vers des points de ramassage pour domaine de livraison... etc. Tous les drones envoient sur la liaison descendante leurs informations télémétriques (relatives à la position, la vitesse, l'orientation... etc.) au DME pour permettre le contrôle et le suivi. Ainsi, par exemple, pour la livraison de marchandises, où il est important que les drones livrent les colis à des endroits précis, la connectivité au DME garantit que les drones suivent les trajectoires prévues.

Pour la livraison des marchandises, il est prévu que les demandes des clients arrivent à certains dépôts de stockage de livraisons. Comme les demandes des clients affectent la planification de la trajectoire, les drones doivent se coordonner avec ces dépôts d'in-

frastructure même si un niveau d'autonomie de décision est utilisé à bord des appareils. Ces dépôts peuvent également servir de centres de réapprovisionnement en énergie pour les drones (Ackerman, 2011). Dans le domaine de la construction, la connectivité avec le DME est une exigence plus critique pour garantir que les structures sont construites selon le plan élaboré initialement. Les informations de séquençage et de suivi des éléments de construction peuvent être maintenues de manière centralisée et les drones peuvent envoyer un retour d'information au DME en conséquence. De plus, le mouvement des drones peut être suivi et les informations sur le cap, la vitesse et la trajectoire peuvent être renforcées pour éviter les collisions. La connectivité à un DME se traduit par la nécessité d'échanger des informations télémétriques sur la liaison descendante, ainsi que de contrôler le trafic sur la liaison montante.

Connectivité avec le personnel au sol

Pour des applications telles que la gestion des catastrophes, il y a une demande supplémentaire de maintien de la connectivité avec le personnel au sol ou les clients, car sans l'intervention du personnel au sol tel que les premiers intervenants, la mission ne peut être menée à bien.

Dans les projets SAR, les drones facilitent les opérations de sauvetage en fournissant une vue d'ensemble rapide de la zone afin de localiser les victimes en un temps record. Une fois l'opération de recherche par drones terminée, le personnel au sol est chargé de mener l'opération de sauvetage. Cela nécessite que les drones aient la capacité de communiquer leurs coordonnées GPS ainsi que toute information de capteur (images thermiques, vidéos...) au personnel au sol sur la liaison descendante.

Pour le domaine de couverture, la demande de connectivité continue avec le personnel au sol varie en fonction de la nature du domaine d'application. Par exemple, la couverture de zone dans le cas de la surveillance agricole ou structurelle (Eschmann et al., 2012) exige une image globale de la zone à surveiller. La seule exigence dans ces applications est que le drone soit capable de fournir des images haute résolution au client. Il y a peu de contraintes de temps dans de tels scénarios. Toutefois, pour la couverture de la zone dans la surveillance, les applications peuvent exiger une connectivité continue au sol (Freed et al., 2005). En cas de couverture du réseau, cette exigence est en fonction des cas, par exemple pour la couverture du réseau lors d'événements sportifs (Guo et al., 2014). Dans le cas de scénarios de catastrophe, le besoin de connectivité avec le personnel au sol peut s'avérer primordial, pour mettre aux victimes de contacter leurs proches. Pour la construction et les scénarios de livraison, la connectivité avec le personnel au sol fait ne se démarque pas comme une exigence de conception de réseau.

Connectivité entre drones

La connectivité entre les véhicules en réseau aérien est un paramètre de conception dépendant de la méthode de mise en œuvre. Ainsi, dans les situations où le terrain a beaucoup changé après une catastrophe naturelle, les opérations de secours en cas de catastrophe et de recherche et sauvetage doivent être effectuées dans de manière décentralisée (Edmondson, 2014), la connectivité entre les drones peut donc devenir une exigence. Toutefois, de nombreux projets développés pour la SAR utilise un processus décisionnel centralisé hors ligne, les drones participant à l'opération de recherche et de sauvetage ne sont pas obligés de communiquer entre eux. La connectivité entre les drones est une nécessité lorsque ces drones sont utilisés pour assurer la couverture du réseau, afin d'assurer la connectivité entre les clients et les réseaux externes dans des scénarios de catastrophe (Reynaud et al., 2011).

La connectivité entre drones peut s'appliquer à la livraison de marchandises dans des situations où plusieurs drones se coordonnent pour le transport conjoint de charges lourdes (Maza et al., 2010). Cependant, la coordination entre les drones n'est nécessaire que dans les situations où la capacité de transport de la charge utile d'un seul drone n'est pas suffisante pour transporter le colis et où plusieurs drones sont nécessaires pour le transport conjoint de charges. La connectivité entre les drones pour la livraison dépend donc de la méthodologie et n'est donc pas considérée comme exigence obligatoire.

2.3.2 Infrastructures

L'infrastructure en termes de réseaux aériens peut signifier aux stations de base au sol permettant la communication. Ces stations de base peuvent faire partie de l'infrastructure de réseau cellulaire déjà existante, mais la demande d'une infrastructure de communication pour la plupart des applications dépend de la méthode utilisée. Si l'on considère les mises en œuvre dans le monde réel, les domaines d'application concernant les situations de catastrophe (y compris la recherche et le sauvetage et la couverture dans les scénarios de catastrophe) supposent généralement une absence de fournitures et d'infrastructures de communication (Reynaud et al., 2011). Dans de telles situations, les drones sont utilisés pour fournir une infrastructure aérienne d'urgence pour l'approvisionnement du réseau de communication. Cependant, pour les applications de surveillance dans la couverture d'une zone, dans de nombreux projets, une infrastructure de communication terrestre est utilisée (Srinivasan et al., 2004). Ces infrastructures désignent souvent les DME.

Pour la livraison des marchandises, les infrastructures font référence aux dépôts de stockage où sont entreposés les colis et les instruments de réapprovisionnement en énergie. Pour la livraison des marchandises, une infrastructure de ces dépôts de stockage peut être

une nécessité pour la livraison à distance en dehors des zones que les drones ne peuvent atteindre sans être rechargés ou ravitaillés. Une telle infrastructure peut également fournir au réseau la connectivité pour le suivi des marchandises transportées par des drones ou l'envoi les commandes de contrôle des drones, si des stations de base sont installées dans le cadre de l'infrastructure (Cummings et al., 2011). Par conséquent, l'infrastructure peut servir à la fois de stockage et de communication. Il est nécessaire que les drones de transmettre leurs données de télémétrie à, et de recevoir des informations de coordination provenant de ces dépôts d'infrastructure. Ceci est nécessaire pour que les drones puissent tenir les dépôts à jour sur leur situation actuelle en matière et d'énergie, et rester informés sur les demandes des clients et les plans de vol actualisés en ligne. Pour la construction, les dépôts de stockage peuvent se référer aux bacs où les pièces et éléments de construction sont stockés. Au fur et à mesure que la zone sur dont le domaine de la construction devrait être restreint, il n'est pas nécessaire d'avoir une infrastructure de tels dépôts de stockage.

Après avoir décrit de manière globale le domaine des drones, nous abordons dans la partie qui suit une architecture servant à la gestion et à la coordination des vols de drones à une échelle continentale, qui est l'UTM.

2.4 UAV Traffic Management

Les applications commerciales et les divers champs d'exploitation des drones, dans différents secteurs, a rendu indispensable la présence d'une autorité de gestion et de coordination centralisée à grande échelle. Cette autorité aurait pour but de permettre le déroulement d'opération de drones dans l'environnement réglementaire aérien de chaque zone, de façon sécurisée, de sorte à éviter les collisions entre les drones, mais aussi de contrôler les différents propriétaires de drones à travers des systèmes d'authentification d'utilisateurs. Tous ces besoins ont poussé la direction de l'Administration fédérale de l'aviation (FAA) et de la National Aeronautics and Space Administration (NASA), à développer et à affiner un concept d'exploitation (ConOps) pour la gestion du trafic des UAV, sous le nom de Unmanned Traffic Management (UTM).

2.4.1 Définition

Dans (FAA, 2018), L'UTM est définie comme étant la manière dont la FAA soutiendra les opérations pour les UAS (drones) opérant principalement dans l'espace aérien à basse altitude. Il s'agit d'un système de gestion du trafic, où les opérateurs sont responsables de la coordination, de l'exécution et de la gestion des opérations, avec des règles de route établies par la FAA. L'UTM est conçu pour répondre à la demande et aux attentes d'un large éventail d'opérations dont la complexité et les risques ne cessent d'augmenter. Le

Le terme "UTM" fait référence à un ensemble de services fédérés et à un cadre global pour la gestion de multiples opérations UAS. Ces services sont distincts, mais complémentaires de ceux fournis par le système ATM (Air Traffic Management), et sont principalement basés sur le partage d'informations entre les opérateurs sur les prévisions de vol et les contraintes de l'espace aérien.

2.4.2 Rôle

Les services de l'UTM sont utilisés pour la gestion des multiples opérations de drones. En effet, l'UTM est responsable de la gestion des vols de drones, puisque cette structure est supposée détenir les informations (présence, trajectoire...etc.) de tous les drones volant dans les zones dont elle est responsable. Ainsi, elle peut valider un plan de vol ou le rejeter pour des raisons de sécurité. Par exemple, le vol peut être refusé si la région est déjà occupée par des plans de vol en cours, car cela peut provoquer une ou plusieurs collisions. L'intérêt de l'UTM est qu'elle fournit (FAA, 2018) :

- Une approche innovante pour répondre aux exigences de service, en tirant parti des services commerciaux qui imposent au gouvernement une charge (coût) d'infrastructure et de main-d'œuvre beaucoup plus réduite.
- Un environnement sûr et stable permettant aux opérateurs d'opérer et de répondre aux besoins des entreprises grâce à une connaissance partagée de la situation et à un cadre opérationnel composé de normes, de réglementations et de protocoles communs qui réduisent les risques et maintiennent la stabilité.
- Une structure flexible et extensible qui peut s'adapter et évoluer au fur et à mesure que l'espace commercial change et mûrit.
- Une construction qui permet à la FAA de maintenir son autorité sur l'espace aérien, tout en permettant à l'industrie de gérer les opérations dans les zones autorisées pour les vols UAS à basse altitude.

2.4.3 Architecture

Avec l'UTM, la FAA conserve son autorité réglementaire et opérationnelle en matière d'espace aérien et de trafic. Il s'agit d'organiser, coordonner et gérer les vols de drones par un ensemble fédéré d'acteurs dans un réseau distribué de systèmes automatisés via des interfaces de programmation d'applications (API). La figure 2.4 illustre une architecture qui identifie visuellement, à un haut niveau, les différents acteurs et composants, leurs relations contextuelles, ainsi que les fonctions de haut niveau et les flux d'information. La

ligne noire dans la figure 2.4 représente la démarcation entre la FAA et les responsabilités de l'industrie dans l'infrastructure, et les entités qui interagissent dans le cadre de l'UTM. Comme on peut le voir, l'UTM comprend les relations entre la FAA, l'opérateur et les différentes entités fournissant des services et/ou démontrant une demande de services dans l'environnement UTM. L'illustration met en évidence une qui fait largement appel à des entités tierces pour soutenir la FAA et l'opérateur dans leurs rôles et responsabilités respectifs.

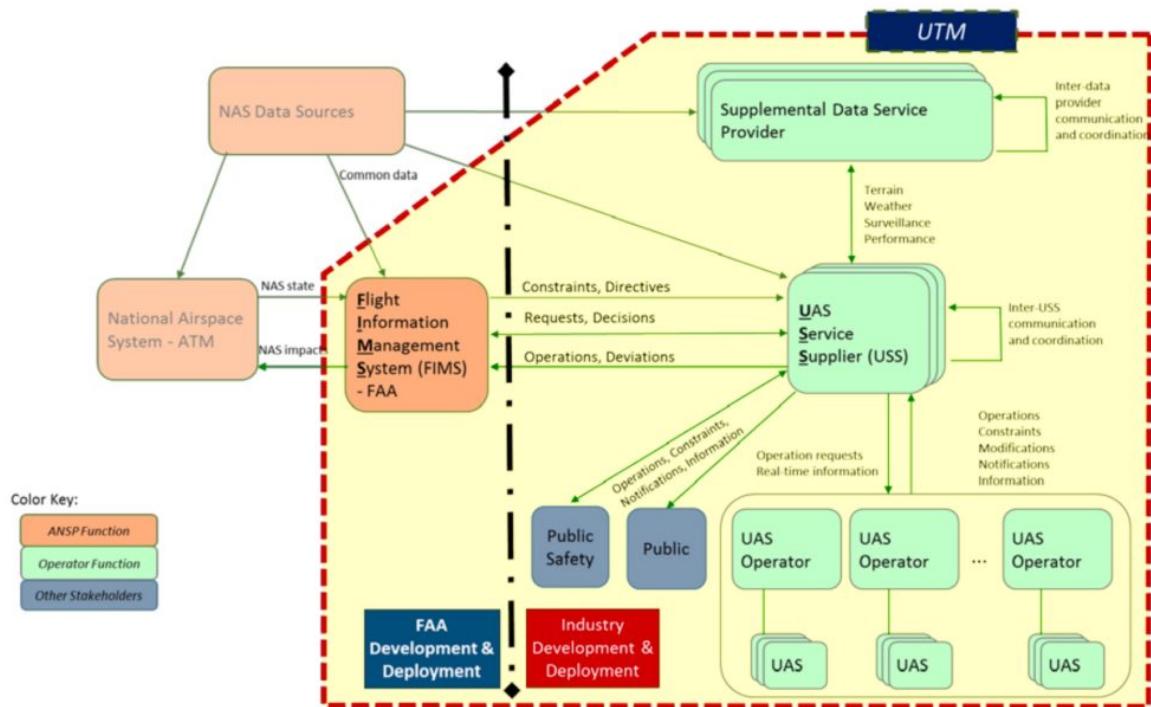


FIGURE 2.4 – Architecture UTM (Federal Aviation Agency, 2018).

Les différentes parties prenantes dans l'architecture, selon (FAA, 2018), sont :

- **Opérateur UAS** : L'opérateur est la personne ou l'entité qui planifie les vols de drones et les opérations, partage les informations sur les intentions de vol et mène les opérations en toute sécurité en utilisant toutes les informations disponibles. L'utilisation du terme "Opérateur" inclut les utilisateurs de l'espace aérien qui choisissent de participer à l'UTM, y compris les opérateurs d'aéronefs pilotés. Cette entité est chargée de créer le plan de vol des drones, qui est un document contenant des informations pertinentes sur le scénario, comme les heures de début et de fin de la mission du drone, la trajectoire de vol et les exigences nécessaires en termes de qualité de service de communication (c'est-à-dire latence, fiabilité, bande passante, etc.).

- **Pilote à distance (RPIC) :** Le RPIC est la personne responsable de la sécurité de chaque vol de drone. Une personne peut être à la fois l'Opérateur et le RPIC. Ce dernier adhère aux règles opérationnelles de l'espace aérien dans lequel le drone vole, évite les autres aéronefs, le terrain et les obstacles, évalue et respecte les contraintes de l'espace aérien et les restrictions de vol, et évite les conditions météorologiques/environnements incompatibles. Le RPIC est capable d'effectuer la surveillance des performances de vol et de l'emplacement du véhicule aérien. Si la sécurité du vol est compromise, en raison de la dégradation des capteurs ou les vulnérabilités environnementales, le RPIC doit être conscient de ces facteurs et doit intervenir de manière appropriée.
- **Fournisseur de Service UAS (USS) :** Un USS est une entité qui fournit des services visant à soutenir l'utilisation sûre et efficace de l'espace aérien en fournissant des services à l'opérateur pour répondre aux exigences opérationnelles de l'UTM. Un USS (i) agit comme un pont de communication entre les acteurs fédérés de l'UTM pour soutenir la capacité des opérateurs à répondre aux exigences réglementaires et opérationnelles pour les opérations UAS, et (ii) fournit à l'opérateur des prévisions de la demande pour un volume d'espace aérien afin que l'exploitant puisse s'assurer de sa capacité à mener à bien sa mission, et (iii) archive les données relatives aux opérations dans des bases de données historiques à des fins d'analyse. En général, ces fonctions clés permettent à un réseau des USS d'assurer la gestion coopérative des opérations à basse altitude sans l'intervention directe d'autorités plus hautes.
- **Réseau USS :** Le terme "réseau USS" fait référence au mécanisme par lequel les opérateurs et les services de soutien partagent des données ou interagissent entre eux (par exemple, un USS met les informations sur les intentions à la disposition de tous les autres USS). Dans l'UTM, plusieurs USS peuvent opérer dans la même zone géographique et peuvent donc soutenir des opérations "en chevauchement" qui nécessitent une orchestration et une synchronisation. Dans cet environnement, le réseau USS partage l'intention opérationnelle et d'autres détails pertinents dans l'ensemble du réseau afin de garantir le partage de la sensibilisation des participants à l'UTM. Compte tenu de la nécessité pour les USS d'échanger un ensemble minimal de données, le réseau USS doit mettre en œuvre un paradigme partagé, avec des méthodes de gestion de conflits ou de négociation, et des normes pour la transmission efficace des intentions de missions et des changements d'intention. Cela réduit et améliore la capacité et l'efficacité globales de l'espace partagé. L'USS devrait également faciliter la mise à disposition des données à la FAA et à d'autres entités pour assurer le fonctionnement sûr de la NAS, et toute autre fonction de

partage collectif d'informations, y compris la sécurité et l'identification.

- **Fournisseur de données supplémentaires UAS :** Les USS peuvent accéder à des fournisseurs de services de données supplémentaires (SDSP) via le réseau USS pour des services essentiels (par exemple, données de terrain et d'obstacles, données météorologiques spécialisées, surveillance). Les SDSP peuvent également fournir des informations directement aux USS ou aux opérateurs par l'intermédiaire de sources de réseau non UTM (par exemple, des sites internet publics/privés).
- **Système de gestion d'information de vols (FIMS) :** Le FIMS est une passerelle pour l'échange de données entre les participants de l'UTM et les systèmes de la FAA, grâce à laquelle la FAA peut fournir des directives et mettre les informations pertinentes sur les NAS (National Airspace System) à la disposition des exploitants de SAMU par l'intermédiaire du Réseau USS. La FAA utilise également cette passerelle comme point d'accès aux informations sur les opérations et est informée de toute situation pouvant avoir un impact sur la NAS. Le FIMS fournit un mécanisme de connaissance de la situation commune à tous les participants de l'UTM et constitue une composante de l'écosystème global de l'UTM. Le FIMS est la composante UTM que la FAA construira pour parvenir à soutenir les opérations de l'UTM.
- **Autorité fédérale d'aviation (FAA) :** La FAA est l'autorité fédérale chargée de l'exploitation des aéronefs dans tout l'espace aérien, et le régulateur et l'autorité de surveillance des opérations des aéronefs civils. La FAA maintient une autorité de surveillance des opérations aériennes civiles au sein de la NAS, qui garantit aux usagers de l'espace aérien l'accès aux ressources nécessaires pour répondre à leurs objectifs opérationnels et que l'utilisation partagée de l'espace aérien puisse être réalisée de manière sûre et équitable. La FAA élabore des règles, des règlements, des politiques et des procédures, selon les besoins, pour soutenir ces objectifs. Avec l'UTM, le rôle principal de la FAA est de fournir un cadre réglementaire et opérationnel pour les opérations et de fournir aux utilisateurs de l'espace aérien des données sur les contraintes de l'espace aérien provenant de la FAA. Cette dernière interagit avec l'UTM à des fins d'échange d'informations/de données selon les besoins, et a accès aux données à tout moment (via le FIMS) pour remplir ses obligations de surveillance réglementaire et opérationnelle.

2.5 Optimisation des vols de drones

2.5.1 Architecture U-space

Afin de fournir un accès efficace, sûr et sécurisé à l'espace aérien, l'Union européenne a récemment lancé un ensemble de nouveaux services appelés "U-space", permettant de soutenir la gestion des véhicules aériens sans pilote (UAV) et de prévenir les conflits liés aux vols dans l'espace aérien. Ces services sont basés sur la technologie de communication, qui devrait être le principal outil permettant de libérer le potentiel sous-jacent des opérations des UAV. À cet égard, la 5G est envisagée comme la norme de communication pour soutenir les diverses opérations et applications des drones.

Au cours de la dernière décennie, les drones ont connu une expansion sans précédent, ils ont été impliqués dans de nombreuses applications rendant la vie plus facile. Différents projets ont été lancés pour développer et déployer des solutions visant à améliorer la gestion du trafic des drones. Dans ce contexte, une série de projets SESAR² ont été lancés. Par exemple, le projet CORUS (CORUS, 2020) a été établi dans le cadre du programme SESAR dans le but de développer et d'écrire un concept d'opérations pour le U-space. Le projet porte également sur l'intégration des drones dans un espace aérien à basse altitude. En outre, afin de concrétiser le concept de U-space, différents projets de démonstration ont été lancés. L'un des projets importants est le GOF (Golfe de Finlande) (GOF, 2020) qui vise à tester les opérations avancées de drones, y compris les aéronefs pilotés et non pilotés dans l'espace aérien partagé, en vue d'accélérer l'adoption du U-space en Europe (en commençant par l'espace aérien à basse altitude).

De plus, comme les technologies de communication sans fil représentent un outil clé pour débloquer le potentiel des drones, l'utilisation de réseaux mobiles pour les drones présente un attrait important pour les chercheurs universitaires et industriels. Plusieurs évaluations et essais réels ont été élaborés sur la base de la normalisation du 3GPP, car certains travaux de recherche du 3GPP ont fourni différents défis pour appliquer les 4G et 5G au domaine des drones. Dans (3GPP, 2018), les améliorations du LTE pour l'atténuation des interférences, pour l'amélioration des performances de mobilité et l'identification aérienne des UE ont été proposées. Pour le réseau 5G, les exigences en matière de support des UAS liées à l'identification à distance des UAS et à la gestion du trafic aérien, ainsi que les exigences fonctionnelles et de performance pour les cas spécifiques d'utilisation des UAV (exposition du réseau, positionnement, etc.) ont été définies. Récemment, le 3GPP a commencé l'étude sur la prise en charge de la connectivité, de l'identification et du suivi des UAS (3GPP, 2019) où la vue initiale de l'architecture de référence compo-

2. SESAR Joint Undertaking, Single European Sky ATM Research. <https://www.sesarju.eu>

sée de l'UAV, de son contrôleur, de l'UTM, des tierces parties et de la 5G, qui agit non seulement comme un relais, mais expose des interfaces pour les interactions avec d'autres acteurs, a déjà été montrée. En parallèle, l'étude sur le support de la couche application pour les UAS est également menée dans (3GPP, 2020). Les résultats de ces études contribueront aux futures modifications de l'architecture et des mécanismes du réseau 5G.

Dans (3GPP, 2020), les auteurs ont fourni différentes évaluations réelles des données de la liaison montante et de la liaison descendante pour un drone volant basé sur le réseau 4G. Dans (Der Bergh et al., 2016) les auteurs ont utilisé un réseau 4G pour employer une connectivité à large bande et sans fil afin d'étudier un cas d'utilisation d'un essaim. (Yuan et al., 2018) quant à eux se sont concentrés sur l'étude d'un cas d'utilisation d'une surveillance congestionnée, basée sur un réseau mobile déchargé par un calcul intensif et envisageant une opération de reconnaissance faciale.

Une vision complémentaire des travaux en cours sur le U-space a récemment été publiée dans (Kuklinski et al., 2020), où le cadre architectural du 3GPP 5GS (3GPP, 2020) a été analysé quant à sa capacité à être intégré au U-space afin de répondre aux exigences techniques et de soutenir les processus commerciaux du cadre du U-space et de l'environnement de la SAMU. En outre, dans (Tomaszewski et al., 2020), l'architecture de mise en œuvre de la 5GS conforme à l'intégration du U-space a été étudiée et les modifications fonctionnelles nécessaires pour combler les lacunes identifiées ont été proposées. D'autre part, la plate-forme multi-opérateurs (à la fois en termes de réseau et d'opérateurs de drones) construite autour du système UTM a été proposée dans (Kuklinski et al., 2020) où le courtage de services de drones est utilisé pour trouver la meilleure offre de vol pour le client des services de drones, basée sur l'ensemble d'opérateurs de drones disponibles, la couverture de la zone de vol par différents réseaux 5G et la disponibilité de l'espace aérien pour la période demandée. Le concept présente également la caractéristique d'une instantiation dynamique et variable des éléments de l'architecture globale pendant le cycle de vie du vol.

2.5.2 Optimisation du Plan de vol

En plus de l'amélioration possible au niveau de l'architecture UAV, une autre optimisation possible est celle du plan de vol, en d'autres termes du chemin parcouru par le drone. Au cours d'un vols de drone, la condition nécessaire au bon déroulement de celui-ci est la fiabilité de la communication sans fil maintenue avec les drones afin de contrôler leur trajectoire et d'éviter les collisions avec d'autres drones en vol. Le lien de communication entre le sol, ou le pilote du drone, et les drones est connu dans la terminologie

des drones sous le nom de lien de contrôle et de commande, ou C2 Link (Organisation Internationale d'Aviation Civile, 2019). Il est prévu que le C2 link soit crucial, car en plus de transmettre les commandes du Drone Pilot pour contrôler la trajectoire des drones, il recueille des informations sur les drones, telles que la position ou la vitesse. Ces informations permettent au pilote du drone de détecter tout risque de collision avec un obstacle ou d'autres drones. Par conséquent, cette liaison doit être sécurisée et bénéficier d'une très faible latence visant à garantir que les demandes de contrôle soient effectuées en temps réel.

En parallèle, l'essor récent des technologies 5G, et leur capacité à prendre en charge des services à haute fiabilité et faible latence, via le service Ultra Reliable Low Latency (uRLLC) (Ksentini et al., 2018), représente une opportunité intéressante pour l'expansion des applications et des services des drones. En effet, la technologie 5G est censée garantir une bande passante plus large et une connectivité à très faible latence, ce qui la place comme un élément clé des services et applications basés sur les drones. La 5G met en lumière plusieurs nouveaux concepts qui peuvent être bénéfiques pour les drones. La 5G New Radio (NR) offre une bande passante plus large (jusqu'à 100 MHz dans la bande de fréquences < 6 GHz, et jusqu'à 400 MHz dans la bande de fréquences > 6 GHz) (GSMA, 2020) pour répondre aux applications exigeant un débit de données élevé, telles que la Réalité Augmentée/Virtuelle, le streaming vidéo 4K. En outre, la 5G NR utilise une nouvelle numérologie de la couche physique qui réduit considérablement la latence du réseau d'accès radio (RAN), ce qui, combiné aux capacités de Mobile Edge Computing (MEC) (Aissiou et al., 2018) à proximité du réseau radio, permettra d'obtenir une très faible latence pour la liaison C2.

Nous décrivons dans ce qui suit le déroulement classique d'un vol de drone à partir de la préparation, et ce afin de mieux cerner les optimisations possibles.

Déroulement du vol

En général, le déploiement des drones doit suivre plusieurs étapes (Khana et al., 2017), réparties en trois blocs : (i) le bloc de définition du champ d'application ; (ii) le bloc du drone qui comprend la planification du vol, la mise en œuvre du vol et l'acquisition des données ; (iii) le bloc du logiciel où sont effectuées l'analyse, l'interprétation et l'optimisation des données. Dans le premier bloc, la mission est clairement établie et des objectifs précis sont définis ; ex. mission pour la surveillance du trafic, l'analyse du trafic sur le réseau et les études comportementales. Le second bloc consiste en la préparation du vol, la prise en compte des conditions de sécurité et d'environnement et la planification du chemin. Le troisième et dernier bloc se déroule pendant le vol, où toutes les opérations (Analyse, Interprétation et Optimisation) sur les données acquises par les drones sont exécutées.

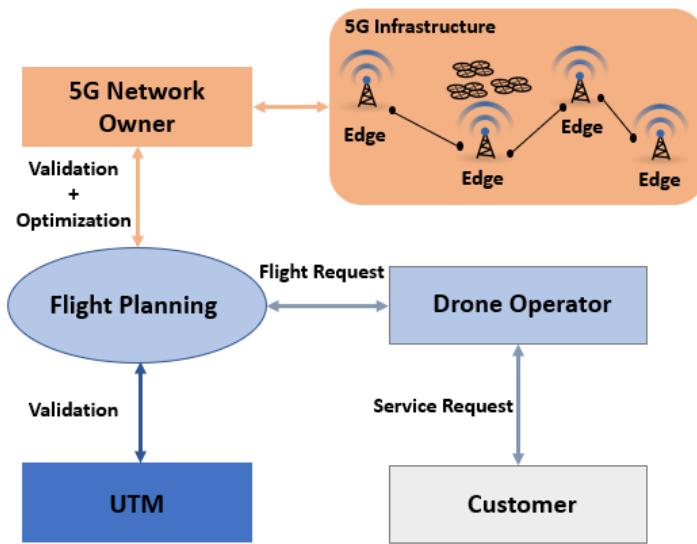


FIGURE 2.5 – Acteurs de planification de vol UAV.

Préparation de vol : La préparation d'un plan de vol pour la mission consiste à proposer une trajectoire (ou route) suivie par le drone pendant un intervalle de temps donné, en fonction des informations de l'infrastructure de l'opérateur réseau. Le plan de vol doit être validé ultérieurement par l'UTM. L'information que le NOP doit indiquer pour préparer le plan de vol est la couverture du réseau de la zone de vol, pour vérifier si l'infrastructure peut offrir la couverture nécessaire du réseau 5G. Par exemple, si à un instant donné, l'infrastructure du réseau est surchargée par un trafic gourmand en bande passante, il peut être difficile d'assurer une latence très faible à la liaison C2 pour les drones. Dans ce cas, le plan de vol doit être modifié et un autre temps de vol doit être proposé. Bien entendu, nous supposons que le NOP est conscient de l'état du réseau dans son infrastructure pendant la journée. Une autre information que le NOP devrait fournir est le nombre de serveurs MEC et leur correspondance avec les stations de base 5G. L'algorithme que nous proposons procède à ce niveau, en aidant l'Opérateur du drone à trouver un chemin qui réduit le nombre de migrations de services, et évite les stations de base qui sont surchargées. En ce qui concerne l'UTM, la validation consiste à vérifier si la zone est sûre en termes de conditions environnementales (Météo, zones accidentées...) et en termes de collisions des drones ; par exemple, si le plan de vol proposé se trouve dans une zone où d'autres drones volent à la même altitude, il y aura alors un risque de conflits et de collisions entre les drones, ce qui signifie que le plan de vol sera rejeté et qu'un autre plan de vol devra être proposé. L'étape de préparation du vol se termine par un plan de vol validé et accepté, qui contient la liste des cellules suivies par le drone, et le temps correspondant. La figure 2.5 résume les différentes étapes de la préparation de vol.

2.6 Conclusion du deuxième chapitre

Nous avons vu dans ce chapitre les différentes applications des drones, leurs besoins en termes de connectivité... etc. mais aussi l'entité extrêmement importante qu'est le UAV Traffic Management. Pour cette dernière, nous avons donné son rôle, son architecture, ainsi que l'ensemble des parties prenantes et le rôle de chacune. Puis, nous avons brièvement présenté certaines recherches qui sont menées pour l'optimisation du vol des drones. Après avoir vu les deux concepts majeurs autour desquels notre projet se situe, qui sont la 5G et les Drones, nous détaillons dans le chapitre suivant l'ensemble des contributions apportées durant le stage.

DEUXIÈME PARTIE

Travail effectué

Conception

Le travail que nous avons effectué durant ce stage de fin d'études est dans le contexte du projet Européen 5G !Drones, dans lequel l'établissement d'accueil EURECOM est un partenaire important. Nous présentons dans ce chapitre le projet 5G !Drones et ses objectifs, puis on présente la plateforme d'EURECOM sur laquelle nous avons eu à travailler. Le travail que nous avons effectué peut se répartir en deux parties : La première consistant en un livrable logiciel permettant d'orchestrer les Slices réseau sur la plateforme d'EURECOM, et l'autre en deux solutions permettant l'optimisation des vols de drones et de l'architecture U-space. Nous présentons donc dans ce chapitre l'étape de conception consistant en l'analyse des besoins du logiciel ainsi que son architecture, mais aussi la formulation mathématique du problème d'optimisation de vols de drones, ainsi que l'architecture proposée.

3.1 Présentation du projet

Le projet 5G !Drones vise à tester des cas d'utilisation de drones sur des installations 5G. Il s'agit d'un programme financé par l'Union européenne dans le cadre de la phase du programme public-privé 5G (5GPPP) qui vise à tester des scénarios de verticaux en utilisant des installations 5G. Un vertical est une entité à part entière offrant des services quelconques, qui dans notre cas s'appuient sur les drones. Le consortium du projet est composé de 20 partenaires, dont les opérateurs de réseaux 5G, drones verticaux, groupes industriels, petits et moyennes entreprises (PME), centres de recherche et universités. Le projet 5G !Drones vise à gérer plusieurs essais de drones, tels que la sécurité publique, la gestion des catastrophes naturelles...etc. En effet, les scénarios de suivi des drones et les cas d'utilisation avant le déploiement commercial sont très critiques, car ces scénarios impliquent des avions en vol, qui peuvent s'effondrer si leur trafic n'est pas bien contrôlé grâce à une connectivité C2 fiable et stable. D'autre part, le projet 5G !Drones permettra aux verticaux des UAV de tester leurs outils et applications, tels que le contrôle à distance et les applications de commande, l'intégration UTM et le streaming vidéo tout en validant les drones et les mesures de performance 5G (Key Performances Indicators KPI). Enfin, le projet 5G !Drones alimentera les organismes de normalisation, tels que le 3GPP, avec les améliorations pour soutenir pleinement et en toute sécurité les scénarios de drones dans 5G et au-delà.

3.1.1 Objectifs

Les principaux objectifs du projet 5G !Drones sont :

- Analyse des exigences de performance des applications et des modèles commerciaux des drones verticaux en 5G.
- Conception et mise en œuvre de la couche logicielle (ou système) 5G !Drones pour réaliser des essais de drones.
- Concevoir un langage descripteur de scénario de haut niveau pour exécuter et analyser les résultats de l'essai des UAV.
- Conception et mise en œuvre de facilitateurs 5G !Drones pour les essais et les opérations de drones (Slices).
- Valider les KPI (Key Performance Indicators) 5G qui démontrent l'exécution des cas d'utilisation de drones (latence, volume de données mobiles, nombre d'appareils).
- Des outils d'analyse de données avancés pour visualiser et analyser en profondeur les résultats des essais, et fournir un retour d'information à l'écosystème 5G et UAV.

3.1.2 Systèmes UAV Trial

Pour effectuer les essais de drones, 5G !Drones s'appuie sur une architecture comportant différents composants, chacun responsable d'une partie du projet, avec quelques spécifications à respecter : (i) le système d'essai doit abstraire les détails de bas niveau pour permettre aux verticaux UAV de décrire le scénario d'essai, en termes de planification de la mission et de ressources de calculs et de réseau nécessaires, en fournissant des interfaces intuitives ; (ii) chaque essai consiste à exécuter une ou deux Network Slices (une uRLLC et une eMBB ou mMTC) ; (iii) chaque installation 5G utilise un modèle différent pour déployer et exécuter un Slice. La figure 3.1 représente l'architecture UAV Trial 5G !Drones.

L'élément clé de l'architecture 5G !Drones est le moteur d'essai (Trial Engine) 5G, qui comprend plusieurs fonctions de l'opérateur de drone, telles que la préparation et la validation du plan de vol, le déploiement du scénario d'utilisation du drone en plus d'une des installations 5G disponibles. Un portail web est défini, qui correspond à l'interface entre un opérateur de drone et le Trial Engine. Ce dernier aura un composant qui permettra au

propriétaire de l'essai de visualiser les résultats de l'essai, c'est-à-dire une liste sélectionnée d'indicateurs de performance clés. Dans ce qui suit, nous décrivons les composants qui composent l'architecture.

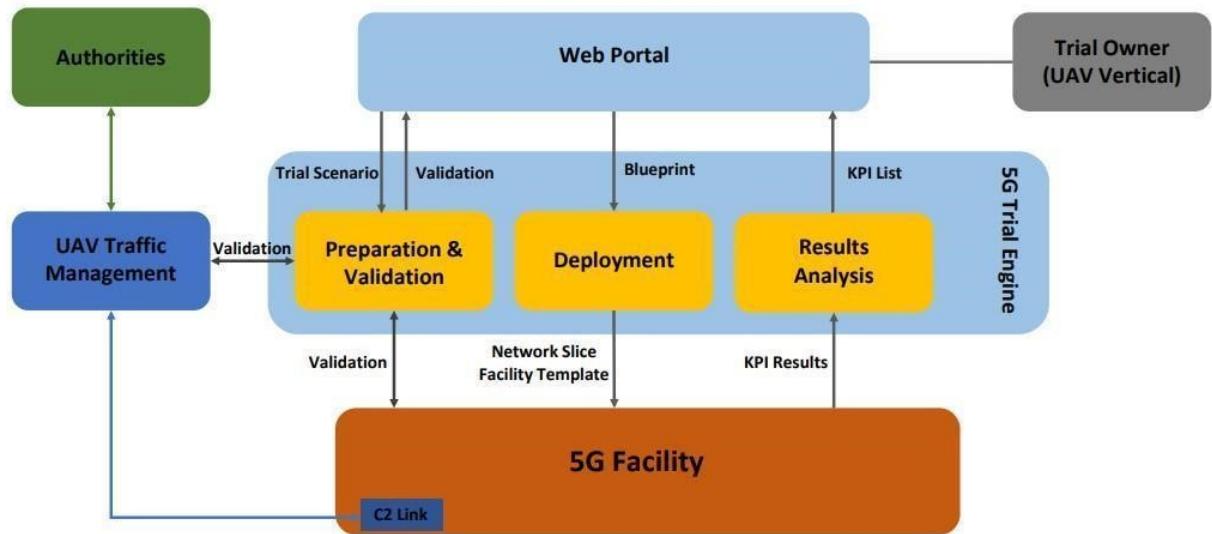


FIGURE 3.1 – Architecture 5G !Drones UAV Trial.

Vertical UAV

Il représente l'utilisateur qui vise à exécuter et à tester un scénario d'utilisation de drone. Il interagit avec le moteur d'essai 5G par le biais d'un portail web.

Portail Web

Le portail Web représente l'interface qui interagit entre les verticaux UAV et le moteur d'essai, qui est chargé de faire respecter le déroulement du scénario d'essai. Le portail Web se compose de deux sous-modules. Le premier est utilisé pour préparer la planification des essais, et le second pour suivre l'état des essais pendant leur exécution. Dans la planification des vols, le propriétaire de l'essai fournit les informations nécessaires à la description de ses scénarios, y compris la période d'exécution, la trajectoire de vol et le choix des installations. Ces informations seront traitées par le moteur d'essai pour la validation. Une fois l'essai validé, le propriétaire de l'essai indique, via le portail web, le réseau et les ressources informatiques nécessaires à l'exécution de l'essai, tels que : le type du Network Slice à exécuter (c'est-à-dire uRLLC, eMBB ou mMTC), la liste des fonctions virtualisées de réseau (VNF) à déployer (par exemple les applications C2, les services de streaming vidéo), et les KPI à mesurer, entre autres. Ensuite, pendant l'exécution, le portail web permet au propriétaire de l'essai de suivre l'évolution de l'essai en présentant le statut de la mission mis à jour à l'aide des données télemétriques reçues, des mesures des KPI et des données collectées par les drones.

Trial Engine

Comme indiqué précédemment, la plupart des fonctions de l'opérateur du drone sont incluses dans le moteur d'essai. Comme l'illustre la figure 3.1 le moteur d'essai est composé de trois blocs fonctionnels :

- **Préparation & Validation** : Cette fonction est déclenchée à la réception d'une demande du propriétaire de l'essai via le portail Web, y compris le scénario d'essai. Dans un premier temps, le plan de vol est préparé, puis vient sa validation. Pour valider l'essai et garantir un accès sûr à l'espace aérien par les drones, le moteur d'essai interagit avec l'UTM et l'installation 5G (5G Facility) par le biais d'une interface dédiée, afin d'éviter toute collision avec un autre vol, de vérifier l'autorisation et la disponibilité des ressources 5G. En cas de rejet, le propriétaire de l'essai est prié de mettre à jour le plan de vol en fonction des limites rencontrées. Ce processus est répété jusqu'à ce que le vol soit approuvé et validé, ou que le propriétaire de l'essai décide d'annuler la demande.
- **Déploiement** : Après la validation du plan de vol, et la réception du réseau ainsi que des besoins en ressources informatiques, sous la forme d'un « Blueprint », qui est un document générique décrivant le plan de vol ainsi que les ressources utilisées, le module de déploiement traduit ce dernier en un « Network Slice Template » (NST), qui sera utilisé pour demander la création d'un Network Slice au niveau de l'installation 5G sélectionnée pour effectuer l'essai. Comme il n'existe pas de norme mais seulement des recommandations pour le NST, chaque installation a son propre modèle de NST. Par conséquent, le moteur d'essai doit traduire le plan en un NST selon l'installation sélectionnée.
- **Analyse des résultats** : L'objectif principal du système étant d'effectuer des essais et de valider l'utilisation de la 5G, il est important de collecter des KPI sur le scénario d'exécution et de les afficher au propriétaire de l'essai via le portail Web. Cela lui permet d'observer et d'analyser les résultats en temps réel. Un module spécifique du moteur d'essai est dédié à la surveillance et à la visualisation des KPI. En effet, après le déploiement des services requis, le module d'analyse des résultats utilise l'API fournie par l'installation 5G pour collecter et surveiller les KPI conformément à la demande du propriétaire de l'essai. Le module d'analyse des résultats effectue également la collecte de statistiques avant d'afficher les résultats au propriétaire de l'essai par le biais d'une interface web dédiée.

Infrastructure 5G

Comme indiqué précédemment, les installations 5G du projet fournissent une infrastructure 5G pour tester les cas d'utilisation des drones, y compris les capacités de calcul, sous la forme de Edge Cloud ou de ressources centralisées sur le Cloud ; et la connectivité 5G en mode autonome (Standalone) et en mode non autonome (Non-standalone). Toutes les installations : (i) comprennent des stations de base NR 5G ; (ii) supportent le Network Slicing ; (iii) gèrent la gestion du cycle de vie (LCM) des VNF.

3.1.3 Architecture globale du système

Plateforme d'EURECOM

La figure 3.2 illustre l'architecture voulue et prévue de l'installation 5G d'EURECOM 5GEVE-Sophia Antipolis et ses composantes. Pour effectuer un essai, 5GEVE-SA devra fournir un portail Web qui permet à un vertical de décrire son essai, de télécharger le Network Service Descriptor (NSD) et son ensemble de Virtual Network Functions (VNF) ou Application Descriptor (AppD) à déployer sur MEC ou Cloud. L'équivalent des fonctions de portail sera disponibles via une API nord (NBI).

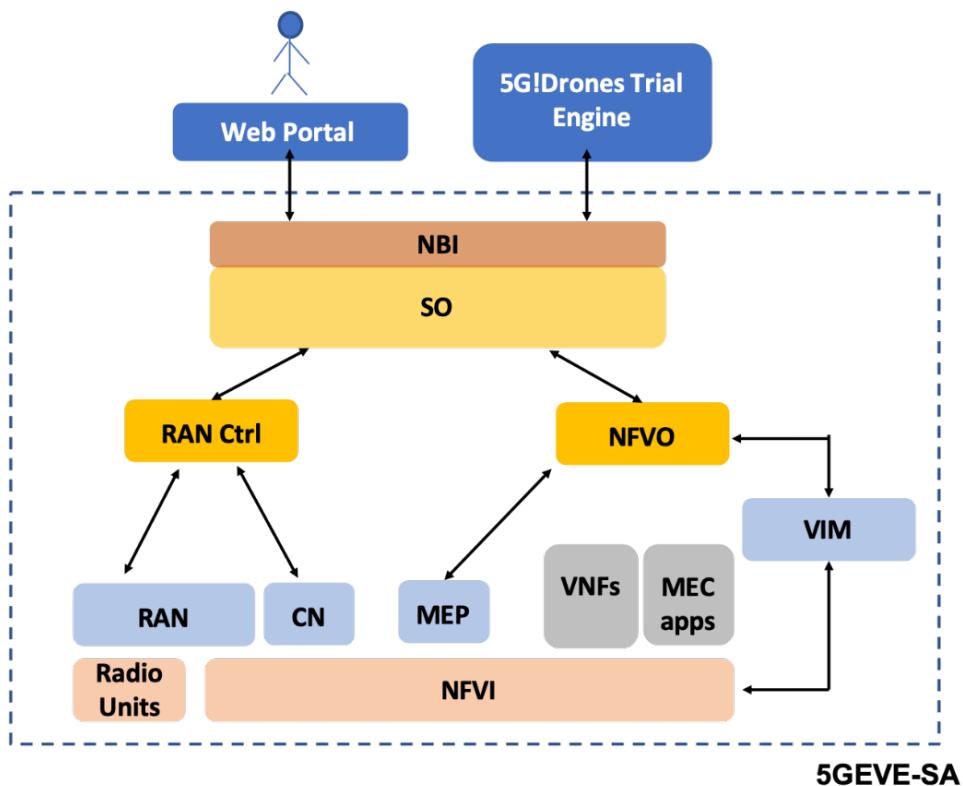


FIGURE 3.2 – Architecture 5GEVE.

5GEVE-SA permettra le déploiement d'un réseau de bout en bout, y compris le Slicing du RAN. Le SO sera chargé de déployer un Slice selon un Network Slice Template

(NST), qui sera détaillé ultérieurement. Le Slice Orchestrator (SO) aura pour rôle de recevoir les demandes du Trial Engine (ou de l'utilisateur) sous forme d'un NST, et effectuer l'instanciation sur l'infrastructure physique sur le NFVO et le RAN Ctrl. Le NFVO (Network Function Virtualized Orchestrator) est le contrôleur des fonctions virtuelles implémentées dans le réseau, comme les fonctions du Core Network (MME, PGW...) mais aussi l'instanciation des applications à déployer sur les équipements utilisateur (Comme les drones...) au niveau du Edge. Le NFVI est l'infrastructure sur laquelle ces VNFs sont déployées, et le VIM est le contrôleur de cette dernière. En ce qui concerne la partie Radio, elle est contrôlée par un RAN Contrôleur dont le rôle est la création et la gestion de Slices RAN, ainsi que la communication entre le Core Network et l'eNodeB (la station de base).

Afin de faire le parallèle avec cette architecture et celle du UAV Trial de 5G !Drones précédemment donnée, nous présentons l'architecture suivante (Figure 3.3) :

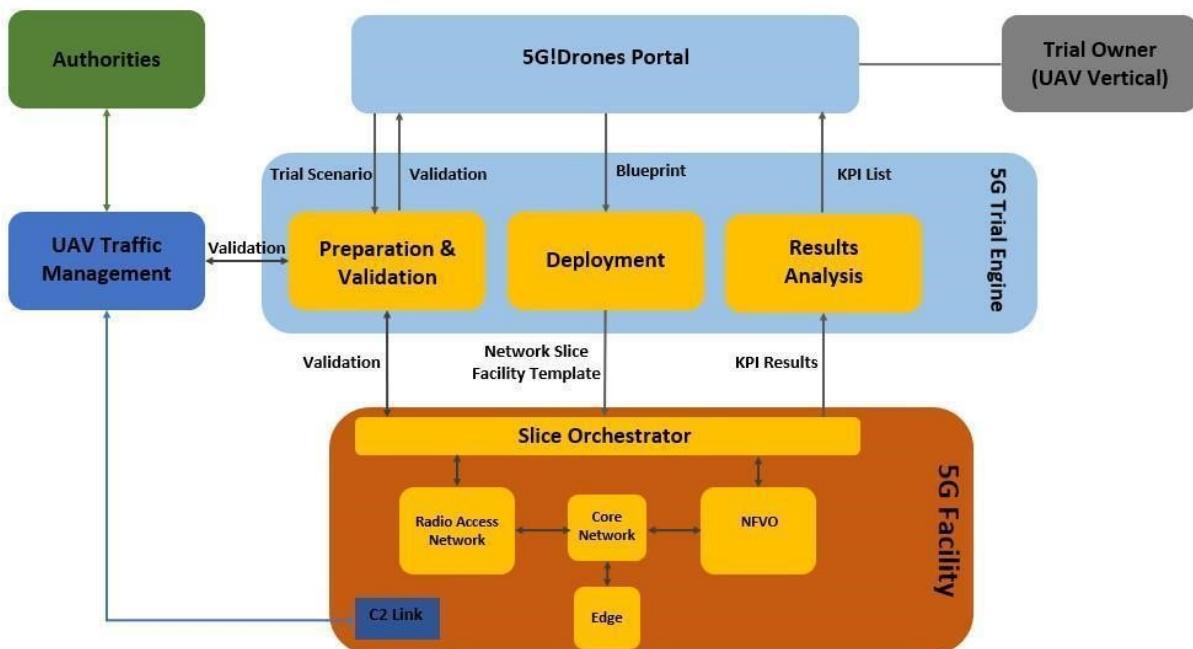


FIGURE 3.3 – Architecture 5GDrones-5GEVE.

Cela signifie que les composants tels que le Slice Orchestrator, le NFVO et le RAN Controller sont en fait des parties de la 5G Facility précédemment décrite. Ainsi, le plan de vol sera transmis à la Facility à travers le SO, qui est en fait le point d'entrée de celle-ci, il sera d'abord validé après avoir confirmé avec la partie RAN et NFVO que les ressources demandées sont disponibles, puis un NST lui sera proposé et il demandera l'instanciation du Network Slice demandé sur les parties plus basses, par le biais d'une API Nord.

3.1.4 Slice Orchestrator

La principale tâche de notre stage est le développement du Slice Orchestrator, par le biais d'une API Nord (Ensemble de méthodes permettant la création et la gestion de Slices réseau sur l'infrastructure 5GEVE-SA). Nous détaillons dans cette section le travail réalisé. Comme cité précédemment, le SO prend comme entrée une NST pour déployer un Network Slice de bout en bout au-dessus de l'installation 5G. L'une des premières tâches que nous avons eues à effectuer est l'élaboration d'un NST complet regroupant l'ensemble des informations nécessaires à la création d'un Network Slice. Le NST auquel nous avons abouti est composé de trois parties principales :

1. Les métadonnées du Slice, telles que la durée, l'identifiant du propriétaire, le statut du Slice... etc.
2. La partie RAN du Slice qui décrit les ressources RAN nécessaires, telles que le type du Slice (uRLLC ou eMBB ou mMTC), la liste des UE (IMSI), la largeur de bande passante demandée, la latence maximale, etc.
3. La partie NSD du Slice qui décrit les ressources virtuelles qui doivent être déployées sur le Edge ou dans le Cloud.

Pour chaque partie, la liste des KPI qui doivent être relevés est mentionnée. Des détails sur les KPI seront donnés dans la section suivante. La figure 3.4 montre la structure du NST, en particulier les champs de métadonnées et la partie RAN. Afin d'aboutir à cette structure, nous avons mené de nombreuses discussions avec les membres de l'équipe responsables de la partie Radio ainsi que de la partie Cloud respectueusement. L'objectif est que le NST fournit à chaque partie les informations nécessaires afin de permettre l'instanciation bout en bout du Slice sur la plateforme.

Une fois le NST prêt, nous avons entamé le développement du SO. Pour ce faire, nous avons d'abord préparé la liste exhaustive des fonctions qui devraient être disponibles sur l'API Nord de la plateforme, selon les besoins des partenaires, qui ont été établis dans le projet 5G !Drones. Le tableau 3.1 contient l'ensemble des fonctions disponibles dans la NBI (proposée donc par le SO) ainsi que leur description.

En parallèle à l'élaboration du NST, nous avons aussi eu à proposer une liste de mesures (KPI) à relever sur la plateforme, qui pourraient être utiles pour le suivi de l'évolution du Network Slice, ainsi que la vérification de si les besoins escomptés sont bien satisfaits. Le tableau 3.2 comprend la liste des KPI à relever sur l'infrastructure :

```
{
  "metadata": {
    "name": "Test Slice 1",
    "status": "Not running",
    "provider": "Eurecom",
    "version": "1.0",
    "startDate": "2020-04-01 13:23",
    "endDate": "2020-04-01 14:23",
    "kpi-list": ["Service Decommissioning Time"]
  },
  "ranSubSlice": {
    "type": 1,
    "region": 1,
    "latency": 10,
    "interruption": 0,
    "trafficOrientation": "Uplink",
    "UE": [20000000507],
    "kpi-list": ["Latency", "Throughput"]
  },
  "networkServiceSubSlice": {
    "name": "MEC app package of a robot control application",
    "version": "1.0",
    "provider": "1",
    "checksum": "1234",
    "userDefinedData": {
      "region_id": 2
    },
    "appD": { ... }
  }
}
```

FIGURE 3.4 – Network Slice Template.

3.1.5 Portail Web

Architecture Portail Web

Afin de permettre l'utilisation de l'API Nord de façon plus interactive, nous avons eu à développer un Portail Web pour permettre aux utilisateurs d'employer le SO de manière plus conviviale, pour ainsi comprendre le fonctionnement de l'architecture 5GEVE-SA. Le Portail Web se doit de permettre aux utilisateurs la création de Slices, en spécifiant leurs besoins. Ces derniers seront transformés par le Portail en NST, et sera transféré au SO, qui effectuera l'instanciation.

Ce portail Web peut être réparti en trois fonctions majeures : Le Trial enforcement, qui permet la création de Slice, le Life-Cycle Management qui permet de gérer le Slice après sa créations, comme son arrêt, sa suppression... etc. et la partie KPI Monitoring qui

| Méthode | Détails |
|---|---|
| createSlice (SliceID, NST) | Instancier le Slice équivalent au NST sur la facilité (NFVO et Contrôleur RAN) |
| getSlice (SliceID) | Retourne le Network Slice Template associé au SliceID |
| stopSlice (SliceID) | Stoppe le Slice sans supprimer les ressources allouées sur la facilité (Les ressources sont toujours allouées mais ne sont pas utilisées) |
| resumeSlice (SliceID) | Relance l'exécution du Slice stoppé avec les mêmes ressources précédemment allouées |
| deleteSlice (SliceID) | Supprime le Slice stoppé en libérant les ressources allouées |
| listSlices() | Retourne tous les slices disponibles (en exécution et stoppés) |
| updateStoppedSlice(SliceID, Paramètres) | Met à jour les ressources d'un Slice stoppé |
| updateRunningSlice(SliceID, Paramètres) | Met à jour les ressources d'un Slice en exécution |

TABLE 3.1 – Méthodes de l'API Nord 5GEVE

| KPI | Niveau | Détails |
|----------------------------|--------|--|
| Slice-deployment-duration | SO | Temps requis pour la création d'un Slice de bout en bout |
| Slice-time-decommissioning | SO | Temps requis pour libérer les ressources du Slice |
| Latency-RAN | RAN | Latence au niveau du RAN |
| Uplink-data-rate | RAN | Débit de données Uplink |
| Downlink-data-rate | RAN | Débit de données Downlink |
| Packet-Loss-rate | RAN | Perte de paquets au niveau du RAN |
| IP-rate | RAN | Débit de paquets dans la couche PDCP |
| Latency-eNB-CN | RAN | Latence mesurée entre le RAN et le Core Network |
| Bandwidth | RAN | Bande passante des cellules |
| CPU-utilization | NFVO | Pourcentage d'utilisation du CPU par les VNF |
| Memory-utilization | NFVO | Pourcentage d'utilisation de la mémoire par les VNF |
| Number-instances | NFVO | Nombre d'instances Docker par VNF |
| Latency-edge | NFVO | Latence entre le Core Network et le Edge |
| Latency-VIM | NFVO | Latence entre le Core Network et le VIM |

TABLE 3.2 – Liste des KPI à mesurer 5GEVE

permet à l'utilisateur d'offrir des dashboards lui affichant les mesures (Key Performance Indicators) du Slice afin de pouvoir contrôler ses performances. Le portail Web contient aussi deux bases de données : une pour les Slices (ou les Trials), et l'autre pour les KPI,

comme l'indique la figure 3.5.

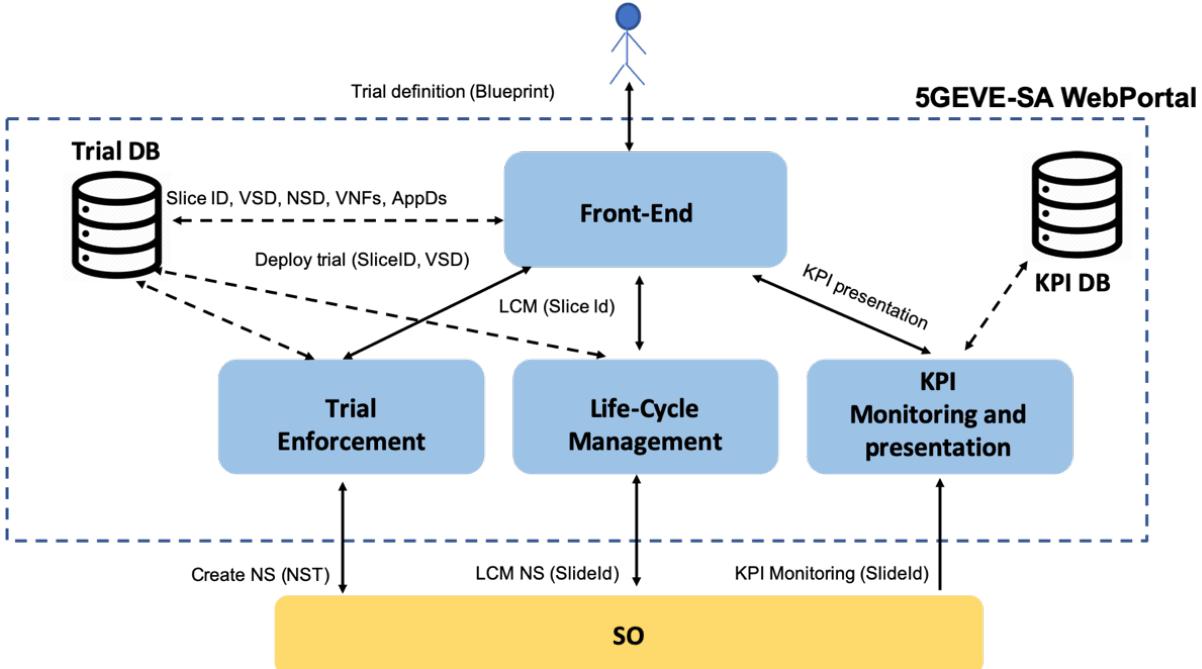


FIGURE 3.5 – Architecture Portail Web.

3.2 Analyse des besoins

3.2.1 Spécifications fonctionnelles

Nous présentons dans ce qui suit l'ensemble des spécifications fonctionnelles que notre système (SO + Portail Web) devra respecter :

1. Le système doit permettre de créer des Slices réseau.
2. Le système doit permettre de remplir les informations utilisateur relatives aux Slices réseau.
3. Le système doit permettre de supprimer des Slices réseau.
4. Le système doit permettre de stopper des Slices réseau.
5. Le système doit permettre de relancer des Slices réseau stoppés.
6. Le système doit permettre de modifier les caractéristiques des Slices réseau en cours.
7. Le système doit permettre de modifier les caractéristiques des Slices réseau stoppés.
8. Le système doit permettre de redéployer des Slices réseau supprimés.
9. Le système doit permettre de lister les Slices réseau créés.

10. Le système doit permettre d'afficher les temps de création et de suppression de Slices réseau.
11. Le système doit permettre de rediriger l'utilisateur vers une page de suivi de performances (KPI Monitoring) d'un Slice.

3.2.2 Spécifications techniques

D'un point de vue non fonctionnel, la solution devrait répondre aux spécifications techniques suivantes :

1. Le système doit être basé sur l'architecture REST.
2. Le système doit utiliser des bases de données non relationnelles de types NoSQL MongoDB.
3. Le portail Web doit être développé avec le Framework Flask pour Python.
4. L'API du Slice Orchestrator doit être documentée en utilisant l'outil Swagger.
5. Le système doit être implémenté à l'aide de Threads.
6. Le système doit être utilisable sous forme de Container Docker.

3.3 Optimisation de vols

3.3.1 Architecture U-space

Afin de permettre une optimisation des vols de drones, nous avons proposé, avec l'aide de partenaire du projet 5G !Drones une nouvelle architecture de référence visant à intégrer le système 5G dans l'écosystème U-space. L'architecture proposée vise à montrer une conception de référence qui démontre la manière dont la 5G peut prendre en charge les services du U-space, et détaille l'interaction entre les différents éléments permettant d'exécuter plusieurs services de drones en utilisant le système 5G comme infrastructure de communication. Nous la détaillons dans ce qui suit.

Architecture proposée

Une vue représentative de l'architecture de référence proposée, intégrant les systèmes 5G dans le U-space, est illustrée dans la Figure 3.6. Cette architecture vise à décrire les acteurs concernés et les interactions entre eux. Elle est divisée en trois blocs : Le côté client, composé de l'utilisateur final et du Business Provider, le U-space avec ses différents composants, et l'infrastructure 5G. La figure indique également les différentes étapes du déploiement d'une mission : (i) la phase de pré-déploiement qui correspond à toutes les étapes nécessaires pour préparer le plan de vol et le valider, avec l'UTM et le propriétaire du réseau 5G ; (ii) la phase de post-déploiement qui correspond aux procédures pour faire

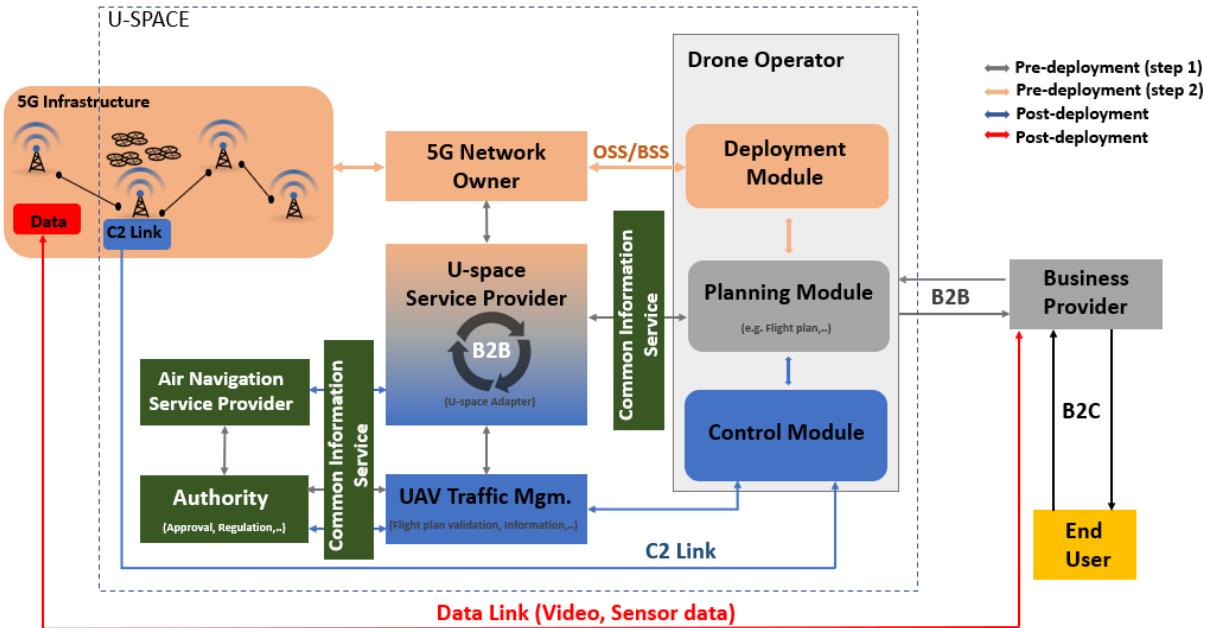


FIGURE 3.6 – Architecture USP-5G.

voler les drones en toute sécurité.

La section suivante décrit les différents composants et interfaces de l'architecture.

— Acteurs :

- **Client** : Une personne, ou une entité, qui souhaite bénéficier d'un service utilisant un drone. Il peut s'agir, par exemple, d'un client qui commande un colis et souhaite qu'il soit livré par un drone, ou d'une société de sécurité qui souhaite surveiller un bâtiment à l'aide de drones équipés de caméras.
- **Business Provider** : Cet acteur est considéré comme le point d'entrée du système, car il offre aux utilisateurs finaux la possibilité d'utiliser des services basés sur les drones. Les utilisateurs finaux doivent passer par le fournisseur de services pour obtenir un service, en lui communiquant les informations les plus pertinentes sur leurs besoins.
- **U-space** : comprend plusieurs composants (opérateur de drones, propriétaire du réseau 5G, infrastructure 5G, fournisseur de services U-space, UTM, autorités) nécessaires à la gestion des drones, comme l'illustre la figure 3.6. Le U-space représente l'ensemble des services fédérés et des fonctions associées dans un cadre complet (SESAR Joint Undertaking, 2019). L'objectif principal de cette entité consiste à permettre et à soutenir des opérations multiples et si-

multanées sûres et efficaces des drones. Dans l'architecture que nous proposons, le fournisseur de services doit contacter U-space pour des raisons de déploiement, de sécurité et de couverture des drones.

- **UTM :** Défini en 2.4.1.
- **Autorités :** Désignées par un État ou une organisation internationale, les autorités sont des entités qui assurent la réglementation, la supervision et l'orientation de l'écosystème des drones, par exemple : l'Agence de l'aviation civile, l'Agence de radiocommunication et d'autres autorités compétentes pour décider des restrictions ou des autorisations pour les vols de drones, les zones (zones d'exclusion aérienne (NFZ) permanentes ou temporaires) et l'utilisation des équipements de radiocommunication. L'une de leurs tâches les plus importantes est, entre autres, de garantir la sécurité des opérations dans le domaine de l'aviation.
- **Service des informations communes :** Entité qui garantit que toutes les informations puissent être échangées entre les différentes organisations pour remplir leurs obligations, et qui permet l'échange d'informations essentielles entre les fournisseurs de services du U-space, les Opérateurs de drones, les fournisseurs de services de navigation aérienne et tous les autres participants à l'espace aérien du U-space (EASA, 2020).
- **Propriétaire du réseau :** L'opérateur du réseau 5G. Le propriétaire du réseau 5G détient des informations sur la couverture disponible et la qualité de service des communications offertes par le réseau mobile. Ainsi, il est responsable de fournir à l'opérateur du drone, via des interfaces spécifiques, des informations sur la connectivité 5G disponible sur une zone sélectionnée dans un temps donné.
- **Infrastructure 5G :** L'infrastructure exploitée par le propriétaire du réseau 5G. Elle se compose de stations de base radio 5G (c'est-à-dire gNB), du Core Network 5G (CN), du Edge et du centre de données. Tous les services et applications nécessaires à la communication et à l'exploitation des drones seront déployés sur cette infrastructure. La partie de l'infrastructure 5G qui soutient le domaine des UAS doit être considérée comme l'une des ressources du U-space, et donc être soumise aux processus de gestion des ressources du U-space. En particulier, ces processus coopéreront avec les processus internes de gestion des ressources, de planification et de reconfiguration du réseau du propriétaire du

réseau. Par conséquent, la nécessité d'une validation préalable des plans de vol est un avantage pour le propriétaire du réseau et permet de préparer à l'avance la configuration nécessaire pour assurer la sécurité des vols et la qualité de la transmission des charges utiles.

- **Opérateur de drones :** l'entité qui pilote les drones et peut offrir des services basés sur les drones. Elle est chargée de créer le plan de vol des drones, qui est un document contenant des informations pertinentes sur le scénario, comme les heures de début et de fin de la mission du drone, la trajectoire de vol et les exigences nécessaires en termes de qualité de service de communication (c'est-à-dire latence, fiabilité, bande passante, etc.). Il est composé de trois grands modules :
 1. **Module de planification :** Responsable de la préparation du plan de vol à exécuter par les drones pour remplir la mission demandée, en tenant compte des conditions de couverture fournies par les réseaux 5G et de la sécurité de l'espace aérien assurée par l'UTM. Ce module doit communiquer avec le propriétaire du réseau 5G et l'UTM par l'intermédiaire du fournisseur de services afin de valider le plan de vol.
 2. **Module de déploiement :** Une fois le plan de vol défini et validé par les trois parties (à savoir le propriétaire du réseau 5G, l'UTM et les autorités), ce module communique au propriétaire du réseau 5G les exigences de communication du service fourni par l'UAV. Le propriétaire du réseau 5G fait ensuite respecter le déploiement des Network Slices et des applications associées pour servir l'UAV pendant sa mission.
 3. **Module de contrôle :** Une fois le vol commencé, des informations en temps réel concernant la progression du vol (par exemple, la localisation, le niveau d'énergie, etc.) sont envoyées par l'UAV au module de contrôle, en utilisant la liaison C2. Ensuite, ce module les communique à son tour à l'UTM, qui envoie une alerte à l'opérateur du drone ; si nécessaire, il peut demander un changement de trajectoire s'il le juge approprié pour éviter d'éventuelles collisions.
- **Fournisseur de services U-space (USSP) :** La partie du U-space qui est chargée d'assurer la communication entre le module de planification de l'opérateur de drone, le propriétaire du réseau 5G et l'UTM, par l'intermédiaire d'une interface Business To Business (B2B) (décrise ci-dessous).

- **Interfaces :** Après avoir présenté les différents acteurs de l'architecture proposée, nous décrivons ici les différentes communications et interactions entre eux. Comme nous l'avons vu précédemment, les acteurs appartenant à différents domaines, différents types d'interfaces sont mis en place, comme les communications entre Business To Customer (B2C) et Business To Business(B2B). Bien entendu, toutes ces communications suivent un certain ordre que nous avons divisé dans la figure en Pré & Post déploiement comme suit :
 - **Utilisateur final – Business Provider (B2C) :** L'utilisateur final communique au Business Provider certaines informations sur le vol demandé, par exemple la destination, les heures de début et de fin du vol. De même, le fournisseur de services communique à l'utilisateur final certaines informations de contrôle en temps réel, comme la localisation du drone, ce qui lui permet de suivre la progression du vol. De plus, il peut communiquer à l'utilisateur final d'autres types de données, comme la vidéo en continu de la caméra à bord du drone.
 - **Business Provider – Opérateur de drones (B2B) :** Le Business Provider utilise cette interface pour transmettre les informations reçues de l'utilisateur final. D'autres informations peuvent être ajoutées, telles que l'emplacement d'un colis dans le cas d'un service de livraison. De l'autre côté, l'opérateur du drone communique au Business Provider des informations sur le déroulement du vol.
 - **Module de planification – Propriétaire du réseau 5G – UTM (B2B) :** Cette interface est considérée comme la plus importante, puisque ces trois acteurs sont impliqués dans la validation/rejet du plan de vol proposé par l'opérateur de drones. En effet, ce dernier, via le module de planification, connaît les heures de début et de fin ainsi que le lieu de départ et la destination de la mission du drone, il peut donc proposer un plan de vol qui consiste en une séquence de coordonnées GPS à suivre par le drone. Ce plan de vol sera transmis au fournisseur de services du U-space par l'intermédiaire du CIS, et au propriétaire du réseau 5G afin d'obtenir des informations sur la zone de couverture pour la période spécifiée. Une fois ces informations obtenues, le plan de vol est transmis à l'UTM pour la phase de validation. Cette dernière garantit que le plan de vol proposé n'engendre pas de conflits avec d'autres vols prévus et vérifie si le plan de vol est conforme aux restrictions de l'espace aérien (par exemple, les zones d'interdiction de vol). En outre, l'UTM peut demander des informations supplémentaires aux autorités, afin de garantir la sécurité du vol.

En cas de rejet du plan de vol, une mise à jour est nécessaire, et un nouveau plan est alors proposé par l'opérateur du drone. Ce processus est répété jusqu'à la validation d'un des plans de vol proposés ou l'annulation de la mission du drone par l'opérateur du drone ou le prestataire de services.

- **Module de déploiement – Propriétaire du réseau 5G :** Le module de déploiement est utilisé pour transmettre le plan de vol validé au propriétaire du réseau 5G, en demandant l'allocation des ressources nécessaires au drone pendant sa mission, ce qui inclut principalement l'allocation des instances Network Slice (NSI). À cette fin, le module de déploiement communique avec le propriétaire du réseau 5G en utilisant le portail du système (OSS/BSS) sur la base d'un plan qui comprend les informations nécessaires pour le déploiement et l'exécution des services requis par le drone pendant son vol. Ensuite, le propriétaire du réseau 5G demande le déploiement physique sur l'infrastructure 5G par le biais d'un Slice pour chaque service.
- **Module de contrôle – C2 Link :** Ce lien est utilisé pendant l'exécution du vol du drone. Il permet au module de contrôle de contrôler et de surveiller le vol de l'UAV en temps réel. Il convient de mentionner que le lien C2 doit être associé à une instance de Network Slice uRLLC, afin de fournir des latences très faibles.
- **Module de contrôle – UTM link :** Cette liaison est utilisée par le module de contrôle pour transmettre à l'UTM les données télémétriques reçues pendant le processus de surveillance du vol. Ces données seront utilisées par l'UTM pour vérifier que tous les vols de drones sont conformes à la restriction négociée pendant la phase de validation du vol. L'UTM peut demander une mise à jour de la trajectoire au module de contrôle, qui à son tour, via la liaison C2, transmet les modifications demandées aux UAV.
- **Payload data – Fournisseur de services :** Ce lien n'est utilisé que dans certains cas, par exemple si les drones déployés sont chargés d'enregistrer des vidéos, ou de collecter certaines mesures. Ces données sont recueillies par les drones et directement envoyées au fournisseur d'entreprises, qui les transmet à son tour à l'utilisateur final par l'intermédiaire de l'interface B2C.
- **UTM – Autorités :** L'UTM maintient un lien de communication permanent avec les autorités, par l'intermédiaire du CIS, ce qui lui permet d'avoir des informations en temps réel sur les vols en cours. Ces informations sont cruciales pour éviter les conflits et garantir un environnement sûr pour les vols des drones.

- **Autorités – ANSP :** Les autorités maintiennent un lien permanent avec le fournisseur de services de navigation aérienne, afin d'obtenir des informations en temps réel sur le contrôle du trafic aérien, à des fins de sécurité.
- **ANSP – USSP :** Le lien entre le fournisseur de services de navigation aérienne et le fournisseur de services du U-space passe par le CIS, et vise à échanger des informations essentielles comme les restrictions de l'espace aérien, le statut de l'espace aérien et les informations disponibles sur le trafic, afin que l'USSP puisse fournir des services à l'opérateur de drone.

Conclusion

Nous avons proposé une nouvelle architecture intégrant les systèmes 5G avec le U-space, où l'objectif est d'utiliser les réseaux 5G pour soutenir les opérations et les services des applications de drones et les vols en toute sécurité. L'objectif est que cette architecture soit utilisée dans le projet 5G !Drones pour mettre en œuvre les composants de l'architecture afin de construire un système permettant d'intégrer des composants de drones et la 5G, dans le but de tester plusieurs cas d'utilisation de drones. À terme, le projet 5G !Drones permettra de valider l'utilisation de la 5G comme système de connectivité pour les drones et les UAS, et donc dans le U-space.

3.3.2 Plan de vol

Outre l'utilisation de la 5G pour la communication avec les drones, l'application permettant de piloter les drones et qui recevra les commandes de contrôle sera hébergé sur des serveurs MEC, ce qui garantira une communication à très faible latence pour la liaison C2. Le déploiement de MEC (Romana et al., 2018) est supposé être très distribué, c'est-à-dire que plusieurs serveurs MEC seront déployés à proximité des utilisateurs finaux, qui dans le cas présent sont les drones eux-mêmes. Un serveur MEC couvrira un ensemble de stations de base, pouvant gérer ainsi une zone géographique limitée. Cependant, comme les drones sont très mobiles, ils peuvent sortir de la zone de couverture d'un serveur MEC, ce qui peut augmenter la latence, et donc perturber la liaison C2 (et donc créer une menace contre la sécurité du vol des drones). Une solution généralement utilisée pour conserver l'avantage de la MEC, en termes de connexion à faible latence, consiste à faire migrer l'application Drone Pilot entre les serveurs MEC (appelée migration de services), en suivant la mobilité des drones. Ainsi, l'application Drone Pilot sera toujours hébergée par le serveur MEC couvrant la zone (ensemble de bases de stations) où se trouvent les drones. Néanmoins, la migration du service a un effet négatif, car pendant la migration, le service

est hors service (appelé "downtime") pendant quelques secondes. Cela peut avoir un effet négatif sur la liaison C2, les migrations de service doivent donc être réduites au minimum.

Nous proposons donc ce qui suit une solution d'optimisation que nous avons proposée afin de réduire le temps d'arrêt de la migration de service en minimisant le nombre de migrations de Drone Pilot parmi les serveurs MEC. Pour atteindre cet objectif, nous proposons un algorithme à utiliser pendant la phase de planification de la mission de drone, où l'opérateur du drone prépare le plan de vol en accord avec l'opérateur du réseau 5G. L'algorithme proposé vise à sélectionner la trajectoire de vol, du point de départ au point d'atterrissement (plan de vol du drone), en considérant non seulement le chemin le plus court mais aussi en réduisant le nombre de migrations de services.

Solution proposée

1. Modélisation du système : Comme indiqué précédemment, l'algorithme que nous proposons intervient à l'étape de planification de l'itinéraire de la mission. Comme habituellement modélisé, nous considérons notre réseau mobile comme étant composé d'un ensemble de stations de base, où chaque station de base a une couverture hexagonale (Levine et al., 1997). Dans la figure 3.7, nous montrons un exemple de topologie de réseau mobile, où chaque serveur MEC (Edge) couvre une zone composée d'un groupe de cellules. L'application Drone Pilot peut être déployée sur le serveur MEC, afin d'assurer une faible latence. Nous avons utilisé des couleurs pour montrer la relation entre un serveur MEC et le groupe de cellules qu'il couvre. Supposons qu'une mission consiste à faire voler un drone d'un point A à un point B (voir Figure 3.7 (b)), dans différentes zones. Nous considérons alors que l'application Drone Pilot est d'abord instanciée dans le serveur Edge1 car elle couvre la position initiale du drone. Ensuite, l'application Drone Pilot est migrée entre les serveurs en fonction de la mobilité du drone.

Évidemment, le chemin droit entre les deux points minimisera la distance parcourue par le drone. Mais, dans certains cas, comme celui illustré sur la figure 3.7 (a) le trajet rectiligne entre les deux points ne nécessitera pas moins de trois migrations de services, puisque le drone passera par les Edges 1, 6, 3 et enfin 4 ; ce qui nécessite de migrer l'application Drone Pilot entre les Edges parcourus. Cela aurait un impact sur les performances de la liaison C2, car la durée de la migration pourrait être conséquente. Toujours d'après la figure 3.7, on peut voir qu'un autre chemin est beaucoup plus intéressant, en termes de migrations de services ; si le drone passe de l'Edge1 à l'Edge2, puis à l'Edge4, il atteindra sa destination finale avec seulement deux migrations de services, ce qui réduira considérablement la durée de downtime. Il faut donc un algorithme qui renvoie le meilleur chemin entre les

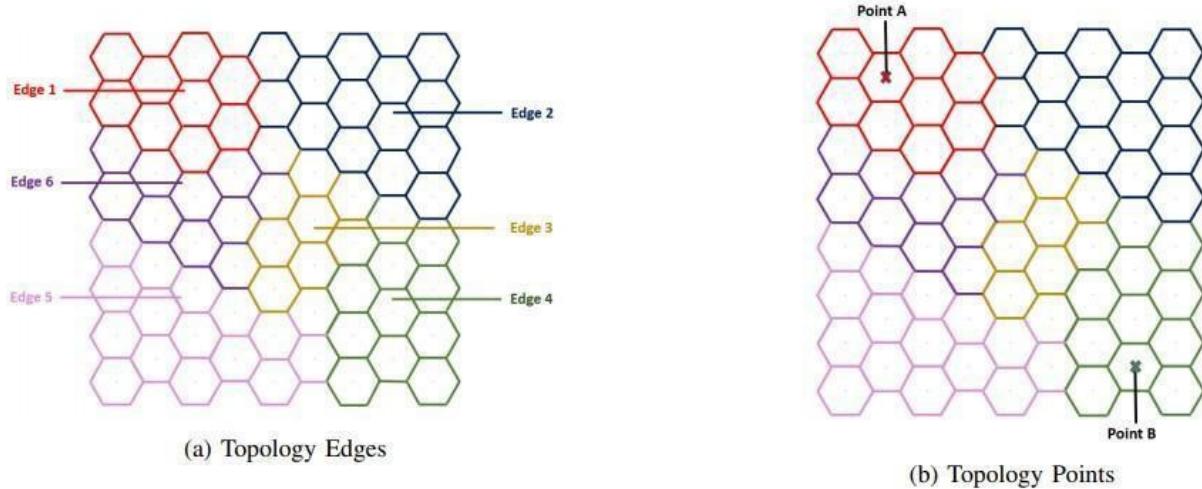


FIGURE 3.7 – Topologie MEC du réseau.

deux points, en termes de minimisation du nombre de migrations de services.

2. Formulation du problème : Nous proposons de modéliser la topologie du réseau de la figure 3.7 par un graphe orienté, où : les sommets représentent les cellules, les arêtes sont pondérées soit par la distance entre les deux cellules, soit par le coût de la migration des services dans le cas où les deux cellules sont sous la couverture de Edges différents. L'objectif est de trouver la trajectoire optimale, c'est-à-dire un ensemble de cellules à traverser, qui réduit les migrations de service du point de départ au point d'arrivée.

Nous désignons par E_i , le serveur Edge couvrant la zone $\{C_{1i}, C_{2i}, \dots, C_{ni}\}$, qui consiste en un ensemble de cellules, où C_{ji} représente la cellule identifiée par j dans la zone couverte par l'Edge E_i . Comme indiqué précédemment, nous modélisons la topologie comme un graphe orienté (V, E) où V est l'ensemble des cellules $\{C_{11}, C_{21}, \dots, C_{n1}, \dots, C_{1k}, C_{2k}, \dots, C_{nk}\}$, k est le nombre de cellules par Edge, et n est le nombre de Edges.

Nous désignons par $w_{(i,j)(k,m)}$ le poids entre deux voisins, c'est-à-dire C_{ij} and C_{km} , qui représente le coût de la migration du service si les deux cellules ne sont pas sous la même couverture de bord, ou la distance entre elles si elles sont dans la même zone. Dans notre cas, comme la topologie a une forme hexagonale, toutes les distances sont similaires, et égales à 1 pour plus de simplicité. Un tel graphe est illustré dans la figure 3.8, où C est le coût fixe de la migration des services entre deux Edges.

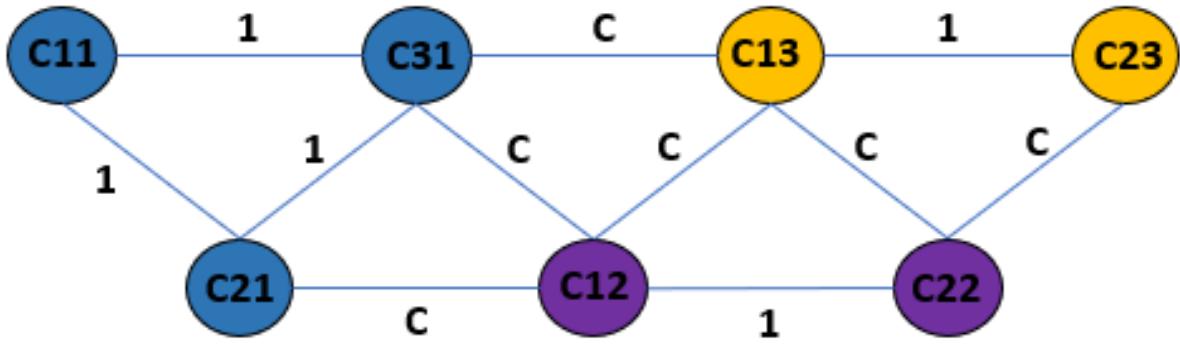


FIGURE 3.8 – Structure du graphe.

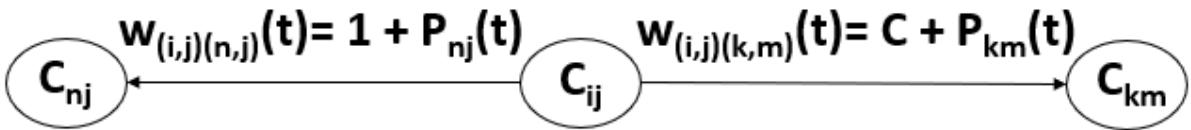


FIGURE 3.9 – Poids des arêtes.

Un autre paramètre qui peut avoir un impact sur les performances de la liaison C_2 (c'est-à-dire la latence) est le fait qu'une cellule sélectionnée peut être surchargée pendant la période de vol, par d'autres types de trafic réseau. Par conséquent, nous ajoutons un autre paramètre au modèle, qui est la probabilité de surcharge de la cellule (notée $P(t)$) qui indique la surcharge d'une cellule à un instant t . Par exemple, à un moment donné, chaque cellule de la topologie aura une probabilité d'être surchargée ; en effet, la probabilité de trouver une cellule occupée à l'heure de pointe est différente des autres périodes de la journée.

Pour introduire ce paramètre dans notre modèle, nous l'incluons dans le poids des arêtes entre les cellules. De cette façon, le chemin choisi tiendra compte de la distance, du coût de migration du service et de la surcharge de la cellule de destination. $P(t)$ peut être calculé à l'aide d'un modèle de prévision, formé à partir des données collectées sur la dynamique du trafic du réseau mobile (Alawe et al., 2018). Nous dérivons le poids $w_{(i,j)(k,m)}$ d'une arête (Figure 3.9) comme suit :

$$w_{(i,j)(k,m)}(t) = \begin{cases} 1 + P_{km}(t) & \text{if } j = m \\ C + P_{km}(t) & \text{else} \end{cases} \quad (3.1)$$

où C_{ij} est la cellule source, C_{km} la cellule de destination, t l'instant demandé et $P_{km}(t)$ la probabilité que la cellule C_{km} soit surchargée à l'instant t .

Pour ajuster l'impact de C et P_{km} sur le poids de l'arête, et pour orienter la solution, nous introduisons un coefficient, noté α . Le poids de l'arête est maintenant exprimé

comme suit :

$$w_{(i,j)(k,m)}(t) = \begin{cases} 1 + (1 - \alpha)P_{km}(t) & \text{if } j = m \\ \alpha C + (1 - \alpha)P_{km}(t) & \text{else} \end{cases} \quad (3.2)$$

Où ($0 \leq \alpha \leq 1$)

Ainsi, on peut utiliser la valeur de α pour orienter la solution en donnant une plus grande priorité à la réduction de la migration des services, ou à la visite de cellules moins chargées. Comme $P_{km}(t)$ exprime une probabilité, sa valeur est comprise entre 0 et 1, ce qui n'est pas le cas pour le coût de la migration de service (C). Par conséquent, les deux valeurs sont normalisées pour leur donner le même ordre de grandeur, afin qu'elles influencent le modèle de la même manière.

Après avoir défini le poids du lien reliant deux noeuds adjacents (cellules voisines), nous notons $P = (c_1, c_2, c_3, \dots, c_n)$ un chemin en V ; où $c_i \in V$, et c_i, c_{i+1} sont deux noeuds adjacents. On note par $f(c_i, c_{i+1})$ la fonction qui renvoie le poids du lien entre c_i et c_{i+1} tel que défini dans l'équation (3.2). Le problème consiste maintenant à trouver un chemin P qui minimise $\sum_{n=1}^{i=1} f(c_i, c_{i+1})$.

3. **Résolution :** Pour trouver le chemin optimal P , en tenant compte à la fois de la migration des services et de la surcharge des cellules, du point de départ au point d'arrivé, nous proposons deux algorithmes. Le premier est basé sur l'algorithme de Djikstra (Djikstra, 1959); tandis que le second est basé sur un algorithme glouton Prim (Prim, 1957).

4. Algorithmes :

- **Basé sur Djikstra :** L'algorithme basé sur Djikstra calcule pour chaque noeud, la distance la plus courte entre le noeud source et celui-ci. Pour ce faire, il initialise d'abord le noeud initial avec une distance égale à 0 et les distances des noeuds restants avec l'infini. Ensuite, il définit le noeud non visité avec la plus petite distance actuelle comme étant le noeud actuel (S). Pour chaque voisin (N), il ajoute le poids de la connexion entre S et N à la distance de la source à S. Si la nouvelle valeur est inférieure à la distance précédente de la source à N, il met à jour cette dernière avec la valeur calculée. Il répète ce processus jusqu'à ce que tous les noeuds soient visités. Notre contribution à l'algorithme Djikstra est l'utilisation de la probabilité de surcharge du noeud cible lorsque nous comparons et mettons à jour les distances du noeud source. En effet, au lieu de ne prendre que le poids de la connexion entre les noeuds, nous ajoutons la valeur de la probabilité à cette connexion.

Algorithm 1: Djikstra-based Algorithm

```

P := {};
d[S] :=  $\infty$  for all S in the graph;
d[start] := 0;
while There is a node out of P do
    Choose a node S out of P with min distance d[S];
    Put S in P;
    for Each node B out of P and neighbor of a do
        if  $d[B] > d[S] + \text{weight}(S,B) + \text{overload\_probability}(B)$  then
            d[b] := d[a] + weight(S,B) + overload_probability(B);
            previous[B] := S;
        end
    end
end

```

- **Basé sur Prim :** En plus de l'algorithme basé sur Dijkstra, nous en introduisons un autre, de type Greedy (Glouton), à savoir l'algorithme Prim. Il crée à partir d'un graphe donné, le Minimum Spanning Tree (MST), qui est un autre graphe extrait du graphe initial, où tous les sommets sont reliés par un chemin, et où la somme de tous les poids est la plus petite possible, en prenant en compte la migration des services ainsi que la probabilité de surcharge. Cet algorithme initialise d'abord le MST comme un ensemble vide, puis prend à chaque étape le bord de poids minimum du graphique initial, et l'ajoute au MST dans le cas où un bord est valide. Une arête valide entre deux noeuds se produit lorsqu'une de ses extrémités est déjà incluse dans le MST et l'autre ne l'est pas. Ces étapes sont répétées, et le nombre d'arêtes dans le MST (nbEdges) est incrémenté à chaque étape jusqu'à ce que le MST contienne un nombre d'arêtes égal au nombre de noeuds du graphique initial (graphSize) moins 2 (nbEdges = graphSize - 2). Une fois que le MST est formé, nous reconstruisons le chemin d'une source à un noeud cible en utilisant l'algorithme de recherche Breadth-first (Zuse, 1972), qui est un algorithme d'exploration d'un graphe en passant par tous les noeuds voisins à la profondeur actuelle, puis en passant aux noeuds au niveau de profondeur suivant et ainsi de suite, jusqu'à trouver le noeud cible. De cette façon, nous donnons le chemin attendu en utilisant l'algorithme Prim.

Algorithm 2: Prim-based Algorithm

```
nbEdges := 1;  
MST := {};  
while nbEdges < graphSize - 1 do  
    | Find minimum weight edge E;  
    | If E is valide edge then add it to MST;  
    | nbEdges := nbEdges +1;  
end
```

3.4 Conclusion du troisième chapitre

Nous avons présenté tout au long de ce chapitre l'ensemble de nos contribution aussi bien sur le plan technique, avec le développement du Slice Orchestrator et du Portail Web, qui permettront l'automatisation de la gestion de Network Slices pour le vol de drones sur la plateforme d'EURECOM, que sur la plan théorique, avec nos deux contributions en Recherche qui permettront d'optimiser les vols de drones sur une échelle globale en étant adoptées par les partenaires du projet 5G !Drones.

Tests et résultats

4.1 Slice Orchestrator

Une fois la liste des méthodes de l'API Nord établie, nous avons développé le Slice Orchestrator en écrivant le corps des fonctions requises en utilisant le langage Python, puis nous les avons intégrées en utilisant un outil Open-Source permettant de créer des API Web, qui est Swagger. L'ensemble des fonctions écrites utilisent donc les API offertes par les deux autres contrôleurs : Le NFVO et le Contrôleur RAN. Swagger permet aussi d'avoir une documentation Web de l'API développée, avec l'ensemble des méthodes, leurs chemins respectifs, leurs paramètres, ainsi que le type de valeurs qu'elles retournent. La figure 4.1 indique la documentation Swagger du SO :

Slice Orchestrator API

This interface provides the offered methods of our Slice Orchestrator API, their parameters and their responses

| Slices | | Show/Hide List Operations Expand Operations |
|--------|-----------------------------------|--|
| GET | /createSlice/{sliceId} | Create a Slice that will be instantiated in our platform |
| POST | /deleteSlice | Delete a Slice |
| GET | /getSlice/{sliceId} | Retrieve a Slice from its ID |
| GET | /listSlices | List the available Slices |
| GET | /resumeSlice/{sliceId} | resume a Slice that will be instantiated in our platform |
| POST | /stopSlice | Stop a Slice |
| POST | /updateRunningSlice | Update a running Slice |
| POST | /updateStoppedSlice | Update a stopped Slice |
| POST | /ValidateResources | Validate the resources needed |
| KPIs | | Show/Hide List Operations Expand Operations |
| GET | /getDashboardKpi/{userId} | Get the Dashboard for KPI monitoring of a Slice |
| GET | /getDecommissioningTime/{sliceId} | Get the Decommissioning time of a Slice |
| GET | /getDeploymentTime/{sliceId} | Get the Deployment time of a Slice |
| Trial | | Show/Hide List Operations Expand Operations |
| POST | /ValidateTrial | Validate the trial |

FIGURE 4.1 – Slice Orchestrator, API Nord

4.2 Portail Web

La figure 4.2 représente la page d'accueil du portail Web développé, permettant à un utilisateur de créer des Slices, de consulter les slices déjà créés, ceux qu'il a supprimés...etc. Il a été développé en utilisant Flask, qui est un environnement de développement Web en Python.

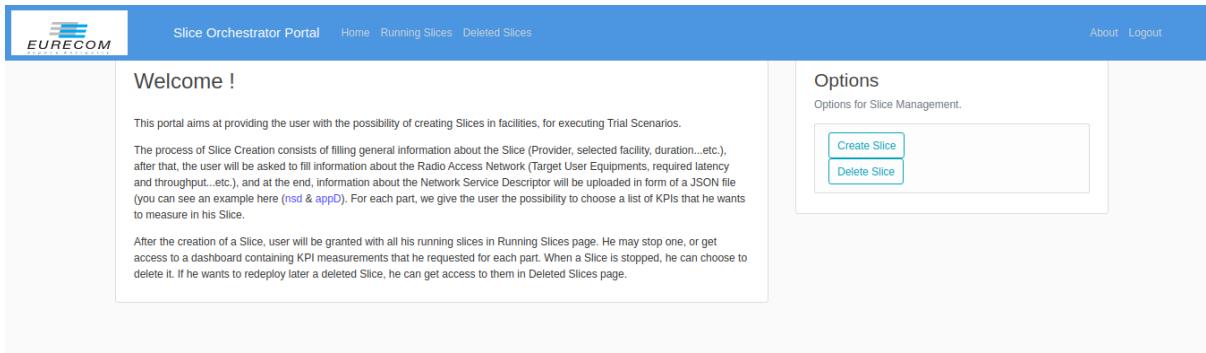


FIGURE 4.2 – Page d'accueil Portail Web.

Après la création du Slice, l'utilisateur a la possibilité de visualiser l'ensemble des Slices créés, les stopper, les modifier ou même les supprimer, selon les méthodes définies au niveau du SO, comme l'indique les figures 4.3, 4.4 et 4.5.

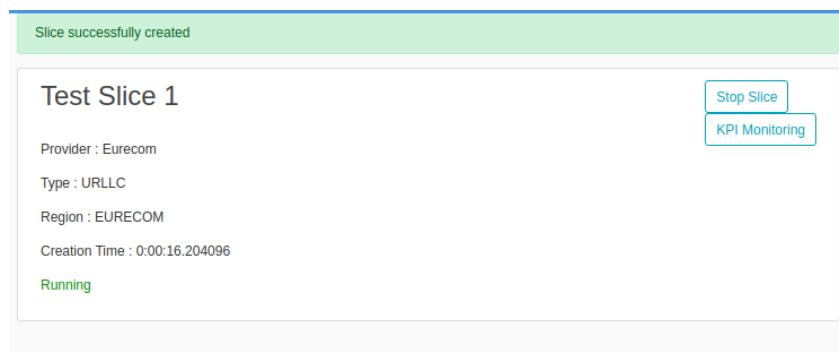


FIGURE 4.3 – Crédit d'un Slice.

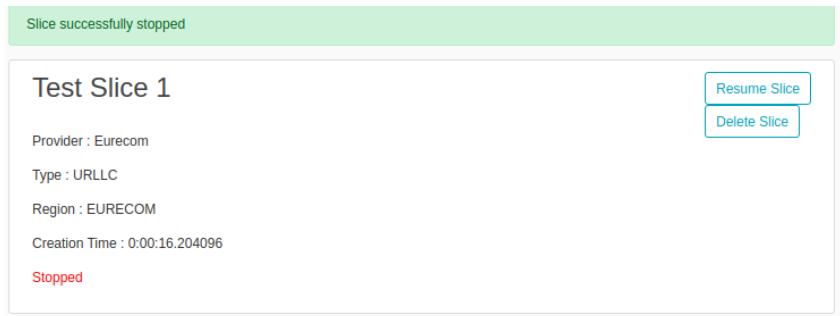


FIGURE 4.4 – Arrêt d'un Slice.

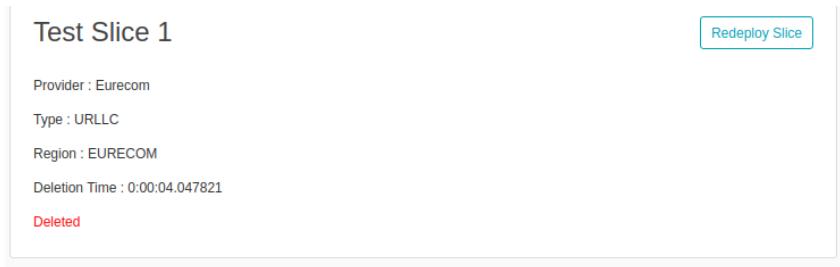


FIGURE 4.5 – Suppression d'un Slice.

4.3 Scénario d'exécution

La figure 4.6 représente l'architecture détaillée de la plateforme 5GEVE-SA sur laquelle nous avons effectué les tests. Mis à part les parties déjà détaillées dans la figure 3.2, d'autres parties y sont ajoutées qui sont :

- **Equipement Utilisateur** : qui peut être un téléphone mobile se connectant au réseau 5G NSA.
- **Core Network** : L'ensemble de fonctions réseau représentant le Core du réseau 5G NSA. Il se répartit en MME (Mobility Management Entity), qui est l'équipement qui gère la signalisation (plan de contrôle) entre les équipements utilisateur et le Core Network, et effectue l'authentification de l'équipement utilisateur dans le réseau, en interagissant avec le HSS (Home Subscriber Server) qui détient une base de données des utilisateurs dans le réseau, et les S-Gateway et P-Gateway qui sont deux équipements qui participent à l'acheminement des données dans le réseau.
- **DNS (Domain Name Server)** : Permettant la translation des noms de domaines auxquels accède l'équipement utilisateur, en adresse IP pour permettre l'acheminement des paquets.
- **Plateforme MEC** : La plateforme sur laquelle sont exécutées les applications

utilisées par l'équipement utilisateur, sous forme de Docker containers.

- **DNS Rule** : Fonction du NFVO permettant d'ajouter dans le serveur DNS le nom de domaine ainsi que l'adresse des applications déployées sur la plateforme MEC.
- **KPI + Logs Manager** : Le contrôleur responsable de gérer l'évolution de l'exécutions des applications en remontant les KPIs précédemment listés et en affichant les logs à un utilisateur extérieur, pour lui permettre de suivre l'évolution de ses applications.
- **eNodeB** : La station de base à laquelle l'équipement utilisateur se connecte et qui redirige le trafic vers le Core Network.

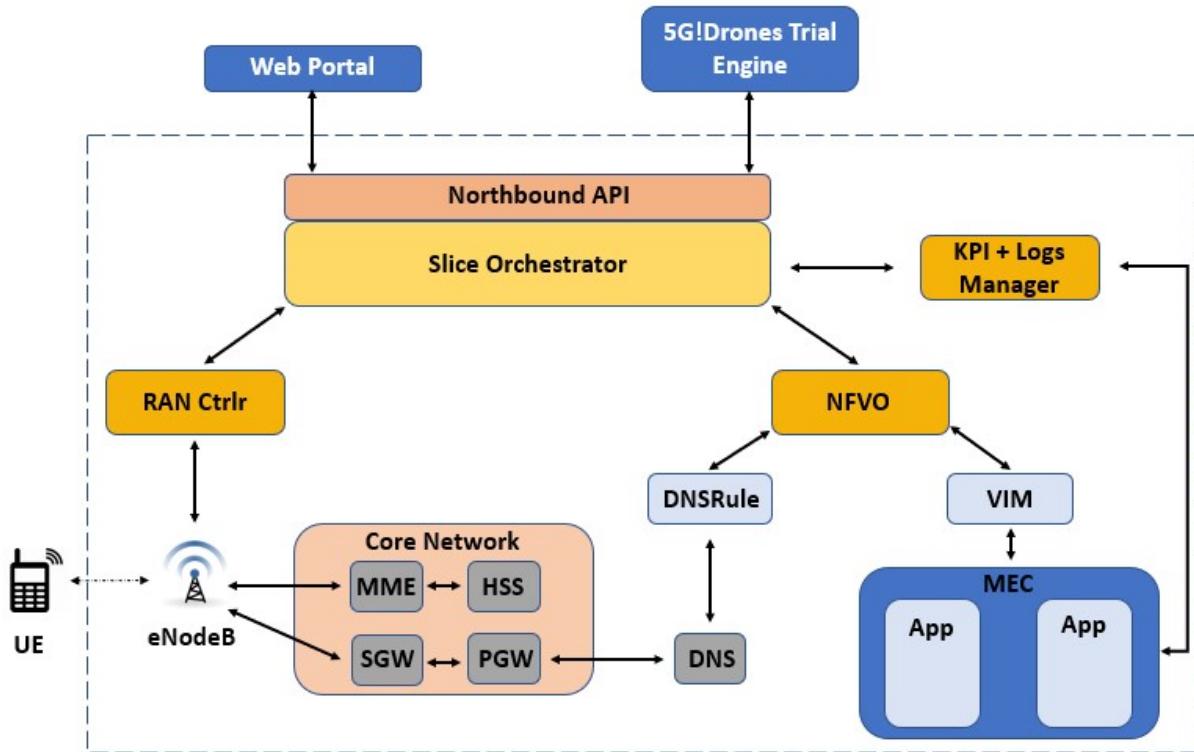


FIGURE 4.6 – Architecture détaillée 5GEVE-SA.

Afin de vérifier le bon fonctionnement de la plateforme, nous effectuons le test de la création de bout en bout d'un Slice, c'est-à-dire à partir du Portail Web, passant par le Slice Orchestrator, la partie RAN et la partie NFV. Pour ce faire, nous avons développé une application client/serveur permettant de récupérer à partir du client les coordonnées GPS (Longitude, latitude, vitesse... etc.) et de les envoyer au serveur, à travers un socket, qui sont des structures systèmes permettant de faire transiter des informations logicielles

entre deux ports dans une ou plusieurs machines d'un même réseau. Après avoir implémenté l'application Serveur, celle-ci a été déployée sous forme d'un Docker container, et a été insérée dans le catalogue des applications de la plateforme MEC, pour pouvoir être instanciée au niveau du Slice.

En premier lieu, afin de créer un Slice, l'utilisateur utilise le Portail Web afin de remplir les informations générales sur le Slice, comme l'indique la figure 4.7.

| Slice Info | |
|----------------------|---|
| Name | Test Slice 1 |
| Provider | EURECOM |
| Type | <ul style="list-style-type: none"> <input checked="" type="radio"/> uRLLC <input type="radio"/> eMBB <input type="radio"/> Hybrid |
| Version | 1 |
| Start Date Time | 06/17/2020 |
| End Date Time | 06/17/2020 |
| Start Hour | 09:59 |
| End Hour | 09:59 |
| KPI List | <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Service Deployment Duration <input checked="" type="checkbox"/> Reconfiguration Execution Duration |
| Next | |

FIGURE 4.7 – Formulaire Métadonnées.

Ensuite, les informations sur la partie Radio sont entrées par l'utilisateur (Figure 4.8). Il est à noter que la partie IMSI est extrêmement importante car elle censée contenir l'identifiant de la puce qui sera au niveau de l'équipement utilisateur qui se connectera au Slice. Cet IMSI sera envoyé par le Contrôleur RAN à l'eNB afin d'identifier l'équipement utilisateur dans le Core Network.

RAN Info

Region

- 5G-EVE

Ue List

- IMSI

Add UE Remove UE

Latency (ms)

10.00

KPI List

- Latency
- Packet Delay Variation
- Uplink Date Rate
- Downlink Date Rate
- Packet Loss Rate
- Bandwidth
- Interruption Time
- IP Rate

Next

FIGURE 4.8 – Formulaire RAN.

Enfin, les informations sur la partie NFV sont remplies, comme l'indique la figure 4.9. Le champ Name doit contenir le nom de l'application dans le catalogue des applications dans la plateforme MEC. La partie Software Image Url contient l'URL dans laquelle le NFVO pourra récupérer l'image de l'application Serveur précédemment insérée pour pouvoir l'instancier. Le DNS Domain Name indique la première partie du nom de domaine par lequel l'application Serveur pourra être accessible, en y concaténant « .eurecom.mec » (Cette tâche est effectuée automatiquement par le NFVO). Ainsi, le NFVO instanciera l'application qui porte le nom entré, et se trouvant dans le catalogue se trouvant à l'adresse Software Image Url, dans la plateforme MEC, et ajoutera une entrée dans le serveur DNS associant son adresse IP et le nom de domaine inséré à travers le DNS Rule.

| NSD Info | |
|---|---|
| Name | <input type="text" value="gps-server"/> |
| Version | <input type="text" value="1"/> |
| Checksum | <input type="text" value="132"/> |
| Application Name | <input type="text" value="gps-server"/> |
| Application Description | <input type="text" value="gps"/> |
| Virtual Memory Size | <input type="text" value="120.00"/> |
| CPU Architecture | <input type="text" value="x86_64"/> |
| Number of Virtual CPU | <input type="text" value="500.00"/> |
| Container Format | <input type="text" value="kubernetes"/> |
| Min Disk number | <input type="text" value="1.00"/> |
| Min RAM Size | <input type="text" value="512.00"/> |
| Size | <input type="text" value="512.00"/> |
| Software Image URL | <input type="text" value="http://localhost:9000/download"/> |
| Operating System | <input type="text" value="linux"/> |
| Storage Size | <input type="text" value="1.00"/> |
| DNS Domain Name | <input type="text" value="gps-server."/> |
| IP Address Type | <input type="text" value="IP_V4"/> |
| TTL | <input type="text" value="0.00"/> |
| Min RAM Size | <input type="text" value="100.00"/> |
| Time Unit | <input type="text" value="ms"/> |
| <input type="button" value="Validate"/> | |

FIGURE 4.9 – Formulaire NFV

```

HOST = 'gps-server.eurecom.mec' # The server's hostname or IP address
#HOST = '172.29.248.16'
#HOST = '192.168.12.167'
PORT = 65432                 # The port used by the server

```

FIGURE 4.10 – Socket Host.

Après avoir créé le Slice, nous testons son fonctionnement en connectant à un ordinateur une dongle contenant la puce dont l'IMSI est celui rempli précédemment dans la partie Radio. Ainsi, l'ordinateur fera office d'équipement utilisateur. Ensuite, nous lui re-lions un capteur GPS permettant de récupérer les coordonnées, et nous les transmettons à l'application serveur en utilisant le nom de domaine rempli précédemment dans la partie NFV, comme l'indique la figure 4.10.

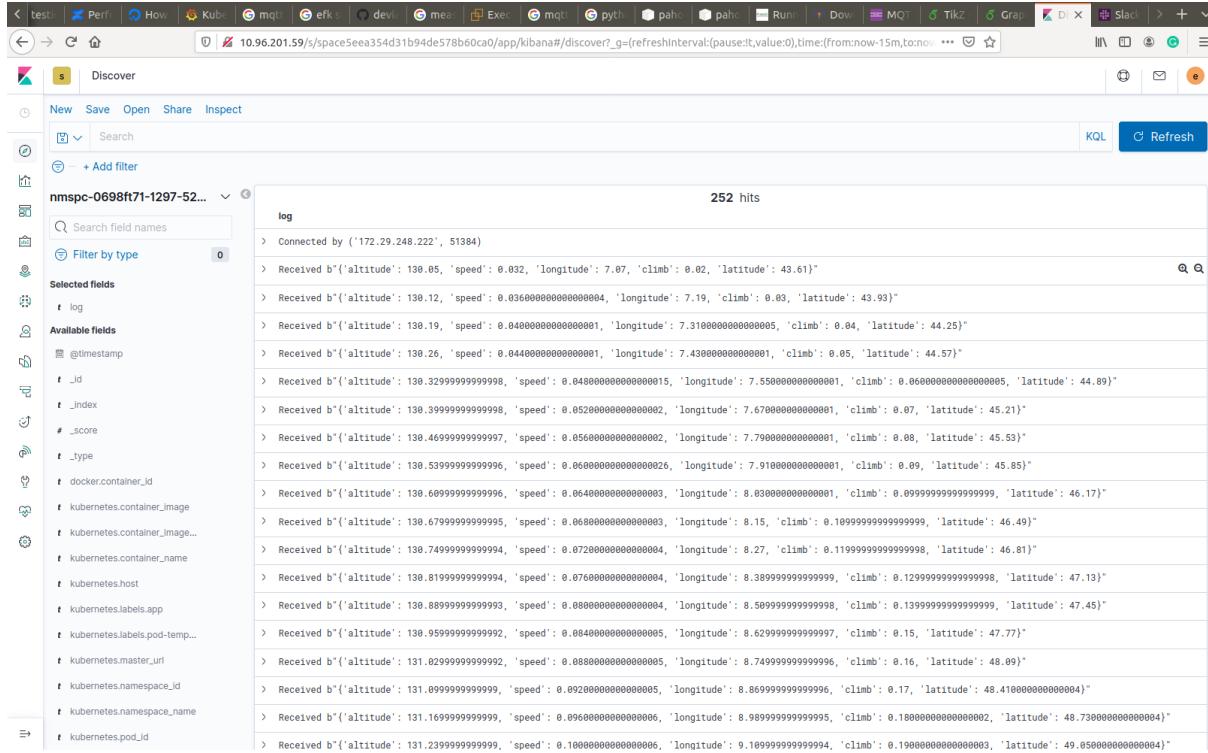


FIGURE 4.11 – Logs Serveur GPS.

Nous pouvons voir dans la figure 4.11 qui indique les logs de l’application que l’accès se fait de manière réussie, et ceci car le trafic émanant de l’équipement utilisateur est capté par l’eNodeB, qui effectue l’authentification de l’utilisateur dans le Core Network, puis en le relayant vers le PGW et le SGW, qui après avoir consulté le serveur DNS avec le nom de domaine de l’application Serveur, obtient l’adresse de celle-ci, et peut ainsi relayer le trafic à son tour vers l’application.

Nous avons donc effectué la création de bout en bout d'un Slice, en y intégrant la partie Radio et NFV. L'intérêt d'une telle application client serait de la faire déployer sur un drone, ce qui servirait à avoir ses coordonnées GPS en temps réel en les transmettant à l'application Serveur sur la plateforme MEC, en passant par l'eNodeB, ce qui permettrait de bénéficier de latences très courtes et donc de pouvoir suivre le parcours du drone en temps réel.

La figure 4.12 résume les étapes par lequel la création du Slice passe, et par la même occasion le fonctionnement général du Portail Web :

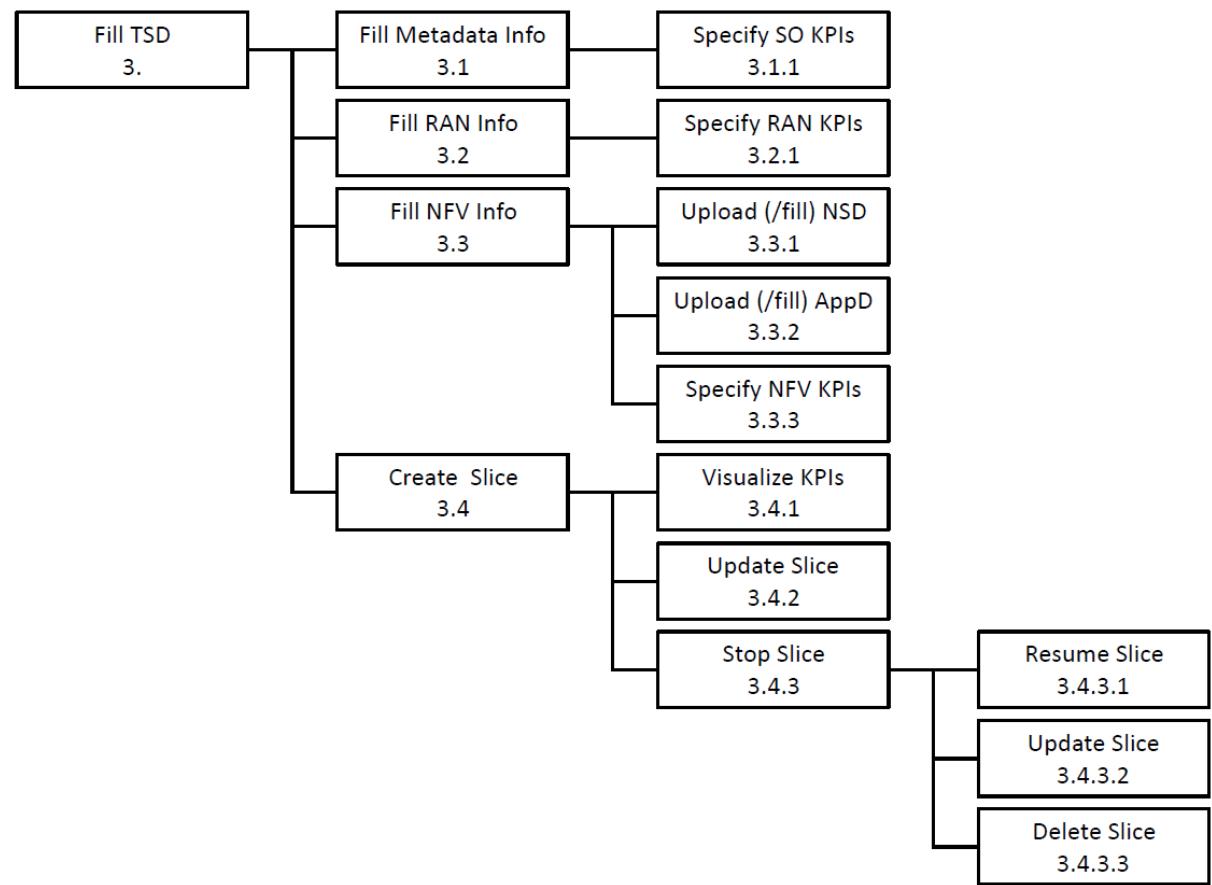


FIGURE 4.12 – Fonctionnement Portail Web.

4.4 Optimisation de vols

Nous présentons dans cette partie les résultats obtenus suite à l'exécution de la solution proposée pour optimiser les plans de vol de drones. Nous rappelons que cette solution consiste à choisir le chemin le moins coûteux selon certains critères (relocations de services, cellules surchargées...etc.).

4.4.1 Evaluation de performances

1. **Topologie et scénarios considérés :** Pour évaluer les performances des deux algorithmes, nous avons exécuté les deux algorithmes sur la topologie de la figure 3.7. Notez que cette topologie n'est qu'un exemple, nous supposons que chaque opérateur de réseau dispose d'un tel modèle pour tous les lieux géographiques couverts par son réseau mobile.

Nous avons envisagé deux scénarios pour modifier la solution attendue, soit en donnant une plus grande priorité à la minimisation des migrations de services, ou à l'évitement des cellules surchargées. Pour y parvenir, nous avons choisi différentes valeurs de α . Il convient de noter que pour tous les scénarios, le coût de migration C et la probabilité de surcharge $P(t)$ sont d'abord normalisées pour donner le même ordre de grandeur aux deux paramètres, en multipliant simplement $P(t)$ par 10, puisque la valeur par défaut du coût de migration du service est de 10, et que la probabilité de surcharge est comprise entre 0 et 1. Toute autre façon de normaliser les deux variables peut être facilement réalisable. À partir de ce point, toutes les valeurs de C et $P(t)$ sont normalisées.

- (a) **Scénario 1 :** Dans ce scénario, on accorde plus de poids à la migration du service, c'est-à-dire que la solution attendue tend à passer par une cellule surchargée qu'à migrer un service MEC. La valeur que nous avons utilisée pour alpha est : $\alpha = 0.8$, de sorte à ce que le coût de la migration soit supérieur à la probabilité de surcharge.
- (b) **Scénario 2 :** Contrairement au scénario 1, dans ce scénario, un poids plus important est attribué à la probabilité de surcharge, c'est-à-dire que la solution attendue évite les cellules surchargées et accepte davantage de migrations de services. Nous avons utilisé pour ce scénario $\alpha = 0.2$, ce qui fait que la probabilité de surcharge est supérieure au coût de la migration.

Comme indiqué précédemment, nous utilisons l'exemple de la figure 3.7, qui consiste en une topologie formée de 68 cellules. Pour chaque scénario, nous mesurons les performances des trois solutions en termes de nombre de migrations de services et de nombre de cellules parcourues qui sont surchargées. Pour mesurer le nombre de migrations de services, nous générerons d'abord de manière aléatoire les probabilités de surcharge des cellules et nous faisons varier le nombre de cellules par Edge MEC de 4 à 22 ; tandis que pour le nombre de cellules surchargées, nous fixons le nombre de cellules (9 par serveur Edge) couvertes par un serveur Edge et nous faisons varier le nombre de cellules surchargées dans la topologie. Nous supposons

qu'une cellule est surchargée lorsque sa probabilité de surcharge dépasse un certain seuil fixé par le propriétaire du réseau (0,5 par exemple). Pour avoir une possibilité de comparaison, nous exécutons également l'algorithme Djikstra (Noté par Default dans les figures) sans aucune contrainte, pour trouver le chemin le plus court, c'est-à-dire que les poids des arêtes du graphe sont pour cet algorithme tous égaux à 1. De cette façon, nous obtenons le chemin le plus court, en termes de distance, entre le point initial et le point final. Nous calculons ensuite le nombre de migrations de services et de cellules surchargées utilisées par le chemin trouvé par les trois algorithmes.

2. **Résultats :** Les figures 4.13 et 4.14 montrent les performances des trois solutions, en termes de nombre de migrations de services ainsi que de nombre de cellules surchargées, pour les scénarios 1 et 2, respectivement. Pour les deux scénarios, le nombre de migrations de services diminue à mesure que le nombre de cellules couvertes par un Serveur MEC augmente, ce qui est logique puisque plus le nombre de cellules à l'intérieur d'un Edge est élevé, moins le nombre de Edges est élevé, et moins le nombre de migrations de services est élevé aussi. De même, le nombre de cellules surchargées sélectionnées dans le chemin proposé augmente à mesure que le nombre de cellules surchargées dans la topologie augmente. Cela est évident car plus le nombre de cellules surchargées dans la topologie est élevé, plus le nombre de cellules surchargées sélectionnées dans le chemin proposé est élevé. En outre, nous remarquons que dans le scénario 1, les algorithmes basés sur Djikstra et les algorithmes par défaut obtiennent des résultats similaires et les meilleures performances en termes de nombre de migrations de services ; cependant, l'algorithme basé sur Djikstra donne de meilleurs résultats en termes de nombre de cellules surchargées retenues dans le chemin. Nous expliquons cela par le fait que l'algorithme par défaut trouve toujours la même solution puisque l'objectif est de sélectionner le chemin le plus court ; alors que l'algorithme basé sur Djikstra prend en considération les deux métriques, un poids plus important est donné à la réduction du nombre de migrations de services. On note également que l'algorithme glouton donne un résultat moins intéressant que les deux autres solutions pour les deux métriques.

Pour le scénario 2, nous voyons dans la figure 4.14 (a) que l'algorithme basé sur Djikstra se comporte comme la solution par défaut en termes de nombre de migrations de services, alors qu'il obtient les meilleurs résultats en termes de nombre de cellules surchargées sélectionnées dans un chemin. Nous expliquons cela par le fait que l'utilisation du chemin le plus court permet d'avoir des solutions intéressantes pour les migrations de services, alors que l'algorithme basé sur Djikstra recherche

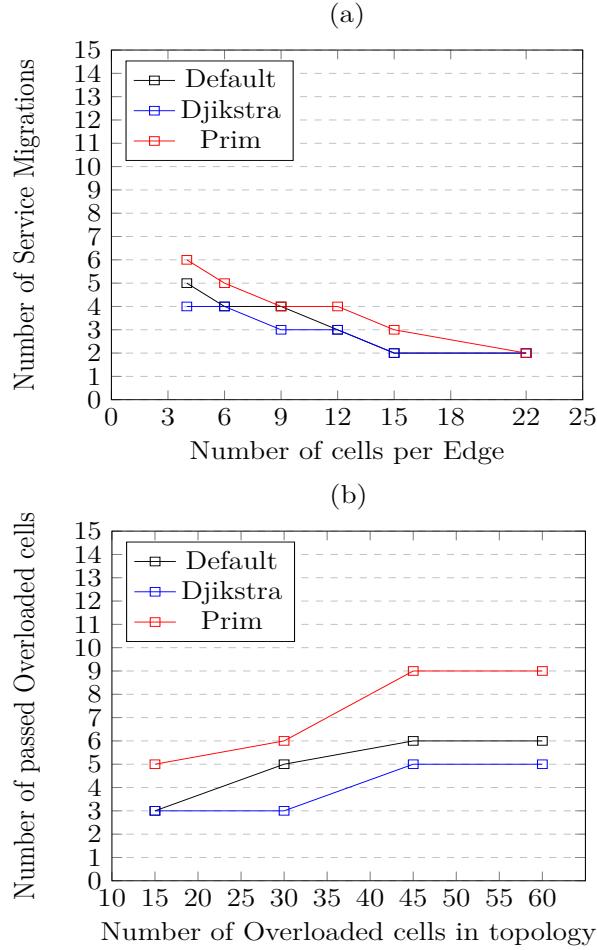


FIGURE 4.13 – Evolution des métriques du Scénario 1

des solutions qui trouvent plutôt un compromis entre les deux métriques, avec plus de poids donné pour réduire le nombre de cellules surchargées. En effet, nous observons clairement dans la figure 4.14 (b) que l'algorithme basé sur Djikstra obtient les meilleurs résultats en termes de nombre de cellules surchargées dans la topologie.

Il convient de mentionner que, dans le cas de l'algorithme basé sur Djikstra, le nombre de migrations de services dans le scénario 2 est plus élevé que dans le scénario 1 (il se situe principalement entre 3 et 8 dans le scénario 2, tandis que dans le scénario 1, il se situe entre 2 et 4). Ceci est dû au fait que l'algorithme basé sur Djikstra donne plus d'importance à l'évitement des cellules surchargées, ce qui signifie que le modèle favorise un chemin avec moins de cellules surchargées. Pour la même raison, le nombre de cellules surchargées est plus petit dans le scénario 2 que dans le scénario 1 (elles sont comprises entre 2 et 4 dans ce scénario alors que dans le scénario 1, elles sont comprises entre 3 et 5).

Ces résultats prouvent clairement que notre modèle est sensible aux poids des pa-

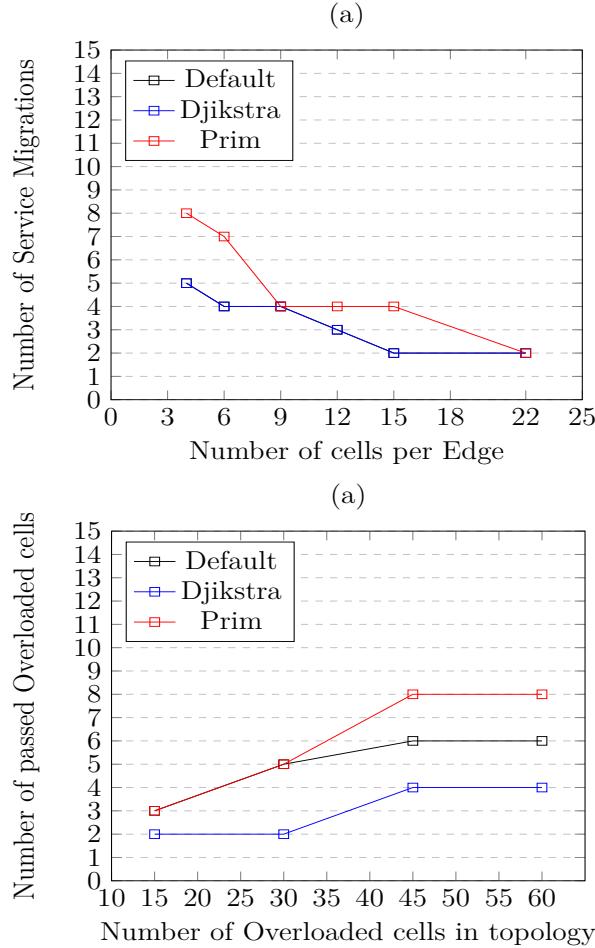


FIGURE 4.14 – Evolution des métriques du Scénario 2

ramètres, ce qui permet à l'opérateur de drones de régler le modèle en fonction de ses souhaits, qu'il s'agisse de réduire les migrations de services ou les cellules surchargées, ou même de considérer équitablement les deux paramètres. Il est à noter que la différence entre ces chiffres dans les deux scénarios n'est pas conséquente en raison de la topologie utilisée, où les Edges sont divisés de telle sorte que les migrations de services ne puissent pas dépasser un certain nombre. La différence devrait être plus conséquente et visible sur une topologie plus grande, où les Edges sont nombreux, ce qui est prévu en 5G.

Conclusion

Dans ce travail, nous avons introduit un nouvel algorithme qui intervient à l'étape de planification de vol d'une mission de drone pour réduire les migrations de service ainsi que les cellules surchargées que les drones en vol vont traverser, en calculant le chemin le plus efficace entre le point de départ et le point d'arrivée. Nous avons introduit une nouvelle métrique pour mesurer le coût de la migration des pilotes de drones à travers les serveurs

Edge. Sur la base de cette métrique, nous avons proposé deux algorithmes pour résoudre le problème : une solution gloutonne, basée sur l'algorithme de Prim, et un algorithme basé sur Djikstra. Les résultats de la simulation ont indiqué que l'algorithme basé sur Djikstra est plus efficace en ce qui concerne les deux mesures ciblées : le nombre de migrations de services et le nombre de cellules surchargées à traverser. Cela améliore la fiabilité globale du C2 link aux drones, ce qui est une exigence fondamentale pour permettre l'adoption à grande échelle des services de drones.

Conclusion Générale

Nous avons présenté dans ce mémoire le travail que nous avons réalisé au sein du centre de recherche EURECOM et portant sur le développement d'un Orchestrateur de Slices sur la plateforme 5G-EVE. Nous avons pu constater que le domaine de la 5G est en plein effervescence, surtout grâce aux initiatives telles que 5G !Drones.

A ce titre, nous avons d'abord effectué une étude bibliographique sur la 5G, le Network Slicing et le Software Defined Networking ainsi que le domaine des drones. Nous avons ensuite décrit le projet 5G !Drones dans lequel s'inscrit notre stage, puis la plateforme 5G-EVE et l'ensemble des contributions que nous avons pu apporter durant ce stage.

L'objectif majeur de ce stage de fin d'études est l'application des notions apprises durant les 5 années du cursus d'ingénieur. Cet objectif a été totalement atteint : D'un point de vue de communication, nous avons eu à travailler avec une équipe de personnes issues de différents pays, et les choses se sont à ce propos très bien passées. D'un point de vue développement, nous avons pu assimiler des outils de développement tels que Swagger ou Flask de façon rapide et efficace, ce qui nous a permis d'écrire du code maintenable et facilement compréhensible. D'un point de vue de proposition, nous avons eu à proposer des idées à maintes reprises surtout dans le cadre des papiers de recherche établis. Ainsi, les qualités majeures d'un ingénieur d'Etat en Informatique issu de l'Ecole nationale Supérieure d'Informatique ont été présentes.

L'originalité de ce stage nous a également été d'un très grand bénéfice car en plus de cette immersion dans un environnement de recherche de niveau international, nous avons dû, du fait de l'exigence de résultat qui nous a été exprimée, acquérir en un temps très court, des connaissances techniques extrêmement diverses, très actuelles, et d'un haut niveau de complexité pour nous intégrer pleinement dans l'équipe-projet.

C'est donc sur tout l'impact positif qu'a eu ce stage sur parcours professionnel que nous tenons à mettre l'accent en conclusion de ce rapport : il nous a permis d'enrichir nos connaissances et compétences dans le domaine technique, de connaître plus en détail le monde de la recherche, de découvrir le sentiment d'appartenance à une équipe, d'accroître le sentiment de responsabilité dans le travail effectué, de découvrir un environnement international en travaillant avec des personnes de nationalité différentes, et sans doute bien plus encore...

Références

3GPP ts 23.401, general packet radio service (gprs) enhancements for evolved universal terrestrial radio access network (e-utran) access" release 13, v13.5.0, 12-2015.

03-1974-computer-vm-survey-goldberg-Gol74.pdf. (n.d.).

3GPP Tech. Rep. 38.913, "Study on Scenarios and Requirements for Next Generation Access Technologies (Release 14)," v14.2.0, 2017

3GPP, "Study on application layer support for Unmanned Aerial Systems (UAS)," 3GPP Technical Report TR 23.754, ver. 0.7.0, April 2020.

3GPP, "Study on enhanced LTE support for aerial vehicles," 3GPP Technical Report TR 36.777, ver. 15.0.0, January 2018.

3GPP, "Study on supporting Unmanned Aerial Systems (UAS) connectivity, Identification and tracking," 3GPP Technical Report TR 23.754, ver. 0.1.0, January 2020.

3GPP, "Unmanned Aerial System (UAS) support in 3GPP," 3GPP Technical Specification TS 22.125, ver. 17.1.0, December 2019.

5G ! Drones. (n.d.).

A. Aissiou, A. Ksentini, A. M. Gueroui, T. Taleb , "On Enabling 5G Automotive Systems Using Follow Me Edge-Cloud Concept ,," IEEE Trans. Vehicular Technology 67(6) : 5302-5316, 2018.

A. Ksentini, P. Frangoudis, N. Nikaein, A. PC, "Providing low latency guarantees for slicing-ready 5g systems via two-level mac scheduling," IEEE Network Magazine, 2018. GSMA Public Policy Position. (2020, March) 5G Spectrum.

Aditya Gudipati, Daniel Perry, Li Erran Li, and Sachin Katti. 2013. SoftRAN : software defined radio access network. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (HotSDN '13). Association for Computing Machinery, New York, NY, USA, 25–30. DOI :<https://doi.org/10.1145/2491185.2491207>

- Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A., Flinck, H. (2018). Network slicing and softwarization : A survey on principles, enabling technologies, and solutions. *IEEE Communications Surveys and Tutorials*, 20(3), 2429–2453. <https://doi.org/10.1109/COMST.2018.2815638>
- Aissiou, A. Ksentini, A. M. Gueroui, T. Taleb , “On Enabling 5GAutomotive Systems Using Follow Me Edge-Cloud Concept ,”*IEEETrans. Vehicular Technology* 67(6) : 5302-5316, 2018.
- Akcay, H., Yiltas-Kaplan, D. (2017). Web-Based User Interface for the Floodlight SDN Controller. *Int. J. Advanced Networking and Applications Volume :*, 8(5), 3175–3180.
- An, N., Kim, Y., Park, J., Kwon, D. H., Lim, H. (2019). Slice management for quality of service differentiation in wireless network slicing. *Sensors (Switzerland)*, 19(12), 1–19. <https://doi.org/10.3390/s19122745>
- Architecture, F. A. (2014). Architecture Installation. (August).
- B. Furht and S. A. Ahson, Long Term Evolution : 3GPP LTE Radio and Cellular Technology. Boca Raton, FL : CRC Press, 2009, ch. 12, pp. 441–443.
- B. Naudts, et al., “Techno-economic Analysis of Software Defined Networking as Architecture for the Virtualization of a Mobile Network”, IEEE EWSDN, Darmstadt, Oct. 2012.
- B. V. Der Bergh, A. Chiumento, and S. Pollin, “LTE in the sky : trading off propagation benefits with interference costs for aerial nodes,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 44–50, May 2016.
- Berman, M., Chase, J. S., Landweber, L., Nakao, A., Ott, M., Raychaudhuri, D., ... Seskar, I. (2014). GENI : A federated testbed for innovative network experiments. *Computer Networks*, 61(2014), 5–23. <https://doi.org/10.1016/j.bjp.2013.12.037>
- Bhalla, M. R., Bhalla, A. V. (2010). Generations of Mobile Wireless Technology : A Survey. 5(4), 26–32.
- Bockelmann, C., Pratas, N., Nikopour, H., Au, K., Svensson, T., Stefanovic, C., ... Dekorsy, A. (n.d.). Massive Machine-type Communications in : Physical. 1–13.

Braun, W., Menth, M. (2014). Software-Defined Networking Using OpenFlow : Protocols, Applications and Architectural Design Choices. Future Internet, 6(2), 302–336. <https://doi.org/10.3390/fi6020302>

Burdakov, O., Doherty, P., Holmberg, K., Kvarnstrom, J., Olsson, P. M. (2010). Positioning unmanned aerial vehicles as communication relays for surveillance tasks. Robotics : Science and Systems, 5, 257–264. <https://doi.org/10.7551/mitpress/8727.003.0034>

Chen, X., Wu, T. (2017). Towards the Semantic Web Based Northbound Interface for SDN Resource Management. Proceedings - IEEE 11th International Conference on Semantic Computing, ICSC 2017, 40–47. <https://doi.org/10.1109/ICSC.2017.27>

Cisco Inc, Network Services Orchestrator Data Sheet - Cisco, <https://www.cisco.com/c/en/us/products-management/networkservices-orchestrator/datasheet-c78-734576.html>.

Converge Network Digest, The Open Network Automation Platform looks like a turning point for telecom architecture, 2017, <http://www.convergedigest.com/2017/04/the-open-network-automation-platform.html>, accessed 2017-09-05.

Corcoran, B. M. (2014). Drone Journalism : Newsgathering applications of Unmanned Aerial Vehicles (UAVs) in covering conflict , civil unrest and. (January). Retrieved from <https://assets.documentcloud.org/documents/1034066/final-drone-journalism-during-conflict-civil.pdf>

CORUS Project. (2020) Concept of Operation for EuRopean UTM Systems. [Online]. Available : <https://www.sesarju.eu/projects/corus>

Darwin, N., Ahmad, A., Zainon, O. (2014). The potential of unmanned aerial vehicle for large scale mapping of coastal area. IOP Conference Series : Earth and Environmental Science, 18(1). <https://doi.org/10.1088/1755-1315/18/1/012031>

David A. Levine, Ian F. Akyildiz and Mahmoud Naghshineh , “A Resource Estimation and Call Admission Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concept ,” IEEE/ACM Transactions on Networking, 1997.

E. Ackerman. (2014) Concept art hints at the awesome future of drones. [Accessed June-2015]. [Online]. Available : <http://spectrum.ieee.org/automaton/robotics/aerial-robots/concept-art-hints-at-the-awesome-future-of-drones>

E. W. Dijkstra, “A note on two problems in connection with graphs,” Numerische Mathematik, 1959.

Eschmann, C., Kuo, C. M., Kuo, C. H., Boller, C. (2012). Unmanned aircraft systems for remote building inspection and monitoring. Proceedings of the 6th European Workshop - Structural Health Monitoring 2012, EWSHM 2012, 2, 1179–1186.

ETSI ISG NFV, ETSI GS NFV-SOL 005 V2.4.1 : Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point, 2018, https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/005/02.04.01_60/gs_NFV-SOL005v020401p.pdf

European Union Aviation Safety Agency . (2020) High-level regulatory framework for the U-space

F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in Proc MCC Workshop on Mobile Cloud Comput., 2012, pp. 13–16.

Federal Aviation Administration. (2018) Unmanned Aircraft System (UAS) Traffic Management (UTM), Concept of OperationsMozaffari, M., Saad, W., Bennis, M., Debbah, M. (2019). Communications and control for wireless drone-based antenna array. IEEE Transactions on Communications, 67(1), 820–834. <https://doi.org/10.1109/TCOMM.2018.2871453>

Foster, N., Harrison, R., Freedman, M. J., Monsanto, C., Rexford, J., Story, A., Walker, D. (2011). Frenetic : A network programming language. Proceedings of the ACM SIGPLAN International Conference on Functional Programming, ICFP, 279–291. <https://doi.org/10.1145/2034773.2034812>

Freed, M., Fitzgerald, W., Harris, R. (2005). Intelligent Autonomous Surveillance of Many Targets with few UAVS. Proceedings of the Research and . . ., (January), 1–9. Retrieved from <http://www.entish.org/published/freed-fitzgerald-harris-05.pdf>

GOF Project. (2020) Gulf of Finland project. [Online]. Available : <https://www.sesarju.eu/node/3387>

Gramazio, F., Cadalbert, T., Augugliaro, F., Kohler, M., Willmann, J., D’Andrea, R. (2012). Aerial Robotic Construction Towards a New Field of Architectural Research. International Journal of Architectural Computing, 10(3), 439–460.

Guo, W., Devine, C., Wang, S. (2014). Performance analysis of micro unmanned airborne communication relays for cellular networks. 2014 9th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2014, 658–663. <https://doi.org/10.1109/CSNDSP.2014.6923909>

Gupta, A., Jha, R. K. (2015). A Survey of 5G Network : Architecture and Emerging Technologies. *IEEE Access*, 3(c), 1206–1232. <https://doi.org/10.1109/ACCESS.2015.2461602>

Gupta, A., Jha, R. K. (2015). A Survey of 5G Network : Architecture and Emerging Technologies. *IEEE Access*, 3(c), 1206–1232. <https://doi.org/10.1109/ACCESS.2015.2461602>

Gupta, L., Jain, R., Vaszkun, G. (2016). Survey of Important Issues in UAV Communication Networks. *IEEE Communications Surveys and Tutorials*, 18(2), 1123–1152. <https://doi.org/10.1109/COMST.2015.2495297>

Hallermann, N., Morgenthal, G. (2014). Visual inspection strategies for large bridges using Unmanned Aerial Vehicles (UAV). *Bridge Maintenance, Safety, Management and Life Extension - Proceedings of the 7th International Conference of Bridge Maintenance, Safety and Management, IABMAS 2014*, (December), 661–667. <https://doi.org/10.1201/b17063-96>

Han, S., I, C. L., Xu, Z., Rowell, C. (2015). Large-scale antenna systems with hybrid analog and digital beamforming for millimeter wave 5G. *IEEE Communications Magazine*, 53(1), 186–194. <https://doi.org/10.1109/MCOM.2015.7010533>

Hayat, S., Yanmaz, E., Muzaffar, R. (2016). Survey on Unmanned Aerial Vehicle Networks for Civil Applications : A Communications Viewpoint. *IEEE Communications Surveys and Tutorials*, 18(4), 2624–2661. <https://doi.org/10.1109/COMST.2016.2560343>

Imad Alawe, Adlen Ksentini, Yassine Hadjadj-Aoul, Philippe Bertin, “Improving traffic forecasting for 5g core network scalability : A machine learning approach,” *IEEE Network Magazine*, November 2018.

International Civil Aviation Organization. (2019) Remotely Piloted Aircraft System (RPAS) Concept of Operatins (ConOps) for International IFR Operations.

ITU-R. (2015). “IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond.” Itu-R M.2083-0, 0, https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.

Iwamura, M. (2015). NGMN view on 5G architecture. *IEEE Vehicular Technology Conference*, 2015. <https://doi.org/10.1109/VTCSpring.2015.7145953>

Ji, H., Kim, Y., Lee, J., Onggosanusi, E., Nam, Y., Zhang, J., ... Shim, B. (2017). Overview of Full-Dimension MIMO in LTE-Advanced Pro. IEEE Communications Magazine, 55(2), 176–184. <https://doi.org/10.1109/MCOM.2016.1500743RP>

Ji, H., Park, S., Yeo, J., Kim, Y., Lee, J., Shim, B. (2018). Ultra-Reliable and Low-Latency Communications in 5G Downlink : Physical Layer Aspects. IEEE Wireless Communications, 25(3), 124–130. <https://doi.org/10.1109/MWC.2018.1700294>

Johansen, T. A., Zolich, A., Hansen, T., Sorensen, A. J. (2014). Unmanned aerial vehicle as communication relay for autonomous underwater vehicle - Field tests. 2014 IEEE Globecom Workshops, GC Wkshps 2014, 1469–1474. <https://doi.org/10.1109/GLOCOMW.2014.7063641>

K. Phemius, M. Bouet and J. Leguay, "DISCO : Distributed multi-domain SDN controllers," 2014 IEEE Network Operations and Management Symposium (NOMS), Krakow, 2014, pp. 1-4, doi : 10.1109/NOMS.2014.6838330.

K. Zuse. Der plankalku. (1972) Der Plankalkül (BFS).

Kafle, V. P., Fukushima, Y., Martinez-Julia, P., Miyazawa, T. (2018). Consideration on automation of 5G network slicing with machine learning. 10th ITU Academic Conference Kaleidoscope : Machine Learning for a 5G Future, ITU K 2018, (November). <https://doi.org/10.23919/ITU-WT.2018.8597639>

K. Katsalis, N. Nikaein, A. Edmonds, Multi-domain orchestration for NFV : Challenges and research directions, in : 2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS), IEEE, 2016, pp. 189–195, <http://dx.doi.org/10.1109/IUCC-CSS.2016.034>, <http://ieeexplore.ieee.org/document/828601/>.

Kim, H., Feamster, N. (2013). Improving network management with software defined networking. IEEE Communications Magazine, 51(2), 114–119. <https://doi.org/10.1109/MCOM.2013.64611>

Kotulski, Z., Nowak, T. W., Sepczuk, M., Tunia, M., Artych, R., Bocianiak, K., ... Wary, J. P. (2018). Towards constructive approach to end-to-end slice isolation in 5G networks. Eurasip Journal on Information Security, 2018(1). <https://doi.org/10.1186/s13635-018-0072-0>

Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., Uhlig, S. (2015). Software-defined networking : A comprehensive survey. Proceedings of the IEEE, 103(1), 14–76. <https://doi.org/10.1109/JPROC.2014.2371999>

Ksentini, A., Nikaein, N. (2017). Toward Enforcing Network Slicing on RAN : Flexibility and Resources Abstraction. *IEEE Communications Magazine*, 55(6), 102–108. <https://doi.org/10.1109/MCOM.2017.1601119>

Laliberte, A. (2009). Unmanned aerial vehicle-based remote sensing for rangeland assessment, monitoring, and management. *Journal of Applied Remote Sensing*, 3(1), 033542. <https://doi.org/10.1117/1.3216822>

Latif, Z., Sharif, K., Li, F., Karim, M. M., Biswas, S., Wang, Y. (2020). A comprehensive survey of interface protocols for software defined networks. *Journal of Network and Computer Applications*, 156, 1–30. <https://doi.org/10.1016/j.jnca.2020.102563>

Lindsey, Q., Mellinger, D., Kumar, V. (2012). Construction with quadrotor teams. *Autonomous Robots*, 33(3), 323–336. <https://doi.org/10.1007/s10514-012-9305-0>

M. Mukherjee, L. Shu, Y. Zhang, Z. Zhou, K. Wang, Joint power and reduced spectral leakage based resource allocation for D2D communications in 5G, in 15th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Zhangjiajie, China, 18–20 November 2015

M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha, “The role of cloudlets in hostile environments,” *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 40–49, Oct. 2013.

Maza, I., Kondak, K., Bernard, M., Ollero, A. (2010). Multi-UAV cooperation and control for load transportation and deployment The original publication is available at www.springerlink.com in this link : *Journal of Intelligent Robot Systems*, 57, 417–449.

McKeown, N., Anderson, T., Peterson, L., Rexford, J., Shenker, S., Louis, S. (2008). *Sigcomm08_Openflow.Pdf*.38(2), 69–74. <https://doi.org/10.1145/1355734.1355746>

Mitra, R. N., Agrawal, D. P. (2015). 5G mobile technology : A survey. *ICT Express*, 1(3), 132–137. <https://doi.org/10.1016/j.icte.2016.01.003>

Muhammad Arsalan Khana, Wim Ectorsa, Tom Bellemansa, Davy Janssensa and Geert Wetsa, “UAV-Based Traffic Analysis : A Universal Guiding Framework Based on Literature Survey,” *Transportation Research Procedia*, 2017. [7] Federal Aviation Administration. (2018) Unmanned Aircraft System (UAS) Traffic Management (UTM), Concept of Operations.

N. H. Motlagh, M. Bagaa, and T. Taleb, “Uav-based iot platform : A crowd surveillance use case,” IEEE Communications Magazine, vol. 55, no. 2, pp. 128–134, February 2017.

Nakao, A., Du, P., Kiriha, Y., Granelli, F., Gebremariam, A. A., Tale, T., Bagaa, M. (2017). End-to-end network slicing for 5G mobile networks. *Journal of Information Processing*, 25, 153–163. <https://doi.org/10.2197/ipsjjip.25.153>

Nanda, S., Chiueh, T. (2005). A Survey on Virtualization Technologies. RPE Report, 179(Vm), 1–42. <https://doi.org/10.1.1.74.371>

NGMN. (2016). Description of Network Slicing Concept by NGMN Alliance. Ngmn 5G P1, 1(September), 19. Retrieved from https://www.ngmn.org/uploads/media/160113_Network_Slicing_v

Olsson, P. M., Kvarnström, J., Doherty, P., Burdakov, O., Holmberg, K. (2010). Generating UAV communication networks for monitoring and surveillance. 11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010, 1070–1077. <https://doi.org/10.1109/ICARCV.2010.5707968>

Ono, F., Ochiai, H., Takizawa, K., Suzuki, M., Miura, R. (2013). Performance analysis of wireless relay network using network coding and UAS. 2013 IEEE Globecom Workshops, GC Wkshps 2013, 1409–1414. <https://doi.org/10.1109/GLOCOMW.2013.682519>

Pack, D. J., DeLima, P., Toussaint, G. J., York, G. (2009). Cooperative Control of UAVs for Localization of Intermittently Emitting Mobile Targets. *IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics*, 39(4), 959–970. <https://doi.org/10.1109/TSMCB.2008.2010865>

Polyanskiy, Y., Poor, H. V., Verdú, S. (2010). Channel coding rate in the finite blocklength regime. *IEEE Transactions on Information Theory*, 56(5), 2307–2359. <https://doi.org/10.1109/TIT.2010.2043769>

Popovski, P., Trillingsgaard, K. F., Simeone, O., Durisi, G. (2018). 5G wireless network slicing for eMBB, URLLC, and mMTC : A communication-theoretic view. *IEEE Access*, 6, 55765–55779. <https://doi.org/10.1109/ACCESS.2018.2872781>

R. C. Prim, “Shortest connection networks and some generalizations,” *Bell System Technical Journal*, 1957.

Reich, J., Monsanto, C., Foster, N., Rexford, J., Walker, D. (2013). Modular SDN Programming with Pyretic. Log In, 38.

Report, T. (2017). TR 138 912 - V14.0.0 - 5G; Study on New Radio (NR) access technology (3GPP TR 38.912 version 14.0.0 Release 14). 0, 0–76.

Reynaud, L., Rasheed, T., Kandeepan, S. (2011). An integrated Aerial Telecommunications Network that supports emergency traffic. 2011 the 14th International Symposium on Wireless Personal Multimedia Communications : Communications, Networking and Applications for the Internet of Things, WPMC'11.

Rodrigo Romana, Javier Lopeza, Masahiro Mambo, “Mobile edge computing, fog et al. : A survey and analysis of security threats and challenges,” Future Generation Computer Systems, January 2018.

S. Davy et al., “Challenges to support edge-as-a-service,” IEEE Commun. Mag., vol. 52, no. 1, pp. 132–139, Jan. 2014.

Saleri, R., Cappellini, V., Nony, N., De Luca, L., Pierrot-Deseilligny, M., Bardiere, E., Campi, M. (2013). UAV photogrammetry for archaeological survey : The Theatres area of Pompeii. Proceedings of the DigitalHeritage 2013 - Federating the 19th Int'l VSMM, 10th Eurographics GCH, and 2nd UNESCO Memory of the World Conferences, Plus Special Sessions FromCAA, Arqueologica 2.0 et Al., 2, 497–502. <https://doi.org/10.1109/DigitalHeritage.2013.6744818>

Santhi, K. R., Srivastava, V. K., SenthilKumaran, G., Butare, A. (2003). Goals of true broad band's wireless next wave (4G-5G). IEEE Vehicular Technology Conference, 58(4), 2317–2321. <https://doi.org/10.1109/vetecf.2003.1285943>

Saraiva de Sousa, N. F., Lachos Perez, D. A., Rosa, R. V., Santos, M. A. S., Esteve Rothenberg, C. (2019). Network Service Orchestration : A survey. Computer Communications, 142–143(May), 69–94. <https://doi.org/10.1016/j.comcom.2019.04.008>

Scherer, J., Yahyanejad, S., Hayat, S., Yanmaz, E., Vukadinovic, V., Andre, T., ... Hellwagner, H. (2015). An autonomous multi-UAV system for search and rescue. DroNet 2015 - Proceedings of the 2015 Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use, 33–38. <https://doi.org/10.1145/2750675.2750683>

Schulz, P., Matthe, M., Klessig, H., Simsek, M., Fettweis, G., Ansari, J., ... Windisch, M. (2017). Latency Critical IoT Applications in 5G : Perspective on the Design of Ra-

dio Interface and Network Architecture. IEEE Communications Magazine, 55(2), 70–78. <https://doi.org/10.1109/MCOM.2017.1600435CM>

Selvaraj, H., Zydek, D., Chmaj, G. (2015). Preface. Advances in Intelligent Systems and Computing, 366, v. <https://doi.org/10.1007/978-3-319-08422-0>

SESAR Joint Undertaking. (2019) Initial view on Principles for the U-space architecture.

SESAR Joint Undertaking. (2020) Single European Sky ATM Research. [Online]. Available : <https://www.sesarju.eu/>

Sezer, S., Scott-Hayward, S., Chouhan, P., Fraser, B., Lake, D., Finnegan, J., ... Rao, N. (2013). Are we ready for SDN ? Implementation challenges for software-defined networks. IEEE Communications Magazine, 51(7), 36–43. <https://doi.org/10.1109/MCOM.2013.6553676>

Sharma, S., Miller, R., Francini, A. (2017). A Cloud-Native Approach to 5G Network Slicing. IEEE Communications Magazine, 55(8), 120–127. <https://doi.org/10.1109/MCOM.2017.1600942>

Si, L. L. S. (n.d.). 5G Architecture and Specifications Examples of Service Adaptive Slices with Dynamic Deployments of Flexible Radio , . 5.

Soret, B., Mogensen, P., Pedersen, K. I., Aguayo-Torres, M. C. (2014). Fundamental tradeoffs among reliability, latency and throughput in cellular networks. 2014 IEEE Globecom Workshops, GC Wkshps 2014, 1391–1396. <https://doi.org/10.1109/GLOCOMW.2014.7063628>

Srinivasan, S., Latchman, H., Shea, J., Wong, T., McNair, J. (2004). Airborne traffic surveillance systems - Video surveillance of highway traffic. VSSN'04 - Proceedings of the ACM Second International Workshop on Video Sureveillance and Sensor Networks, 131–135.

T. Halonen, J. Romero, and J. Melero, GSM, GPRS and EDGE Performance : Evolution Towards 3G/UMTS. New York : Wiley, 2003.

T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, “Mobile edge computing potential in making cities smarter,” in IEEE Commun. Mag., vol. 55, no. 3, pp. 38–43, Mar. 2017.

OF ENGINEERING SCIENCES RESEARCH TECHNOLOGY THE NORTHBOUND APIs OF SOFTWARE DEFINED NETWORKS. © International Journal of Engineering Sciences Research Technology, 501(January 2019). <https://doi.org/10.5281/zenodo.160891>

Voellmy, A., Kim, H., Feamster, N. (2012). Procera. 43. <https://doi.org/10.1145/2342441.2342451>

W. Fang, X. Yin, Y. An, N. Xiong, Q. Guo, J. Li, Optimal scheduling for data transmission between mobile devices and cloud. Inform. Sci. 301, 169–180 (2015)

Waharte, S., Trigoni, N., Julier, S. J. (2009). Coordinated search with a swarm of UAVs. 2009 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, SECON Workshops 2009, 00(c), 25–27. <https://doi.org/10.1109/SAHCNW.2009.5172925>

X. An, F. Pianese, I. Widjaja, and G. Acerr, “Dmme : A distributed lte mobility management entity,” Bell Lab. Tech. J., vol. 17, no. 2, Sep. 2012.

Xiang, W., Zheng, K., Shen, X. S. (2016). 5G mobile communications. 5G Mobile Communications, 1–691. <https://doi.org/10.1007/978-3-319-34208-5>

Yliopisto, O., Six, T., France, G. T. S., Scientific, F. O. R., Sr, A. K., Tilepikoinonies, C. K. (n.d.). Unmanned Aerial Vehicle Vertical Applications’ Trials Leveraging Advanced 5G Facilities.

Z. Yuan, J. Jin, L. Sun, K. Chin, and G. Muntean, “Ultra-reliable iot communications with uavs : A swarm use case,” IEEE Communications Magazine, vol. 56, no. 12, pp. 90–96, December 2018.

Zhang, C. (2006). AN UAV-BASED PHOTOGRAHMETRIC MAPPING SYSTEM.

Zheng, J., de Veciana, G. (2019). Elastic Multi-resource Network Slicing : Can Protection Lead to Improved Performance ? Retrieved from <http://arxiv.org/abs/1901.07497>

Zhou, W., Li, L., Luo, M., Chou, W. (2014). REST API design patterns for SDN northbound API. Proceedings - 2014 IEEE 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE WAINA 2014, 358–365. <https://doi.org/10.1109/WAINA.2014.153>

Zhou, X., Li, R., Chen, T., Zhang, H. (2016). Network slicing as a service : Enabling en-

terprises' own software-defined cellular networks. *IEEE Communications Magazine*, 54(7), 146–153. <https://doi.org/10.1109/MCOM.2016.7509393>

TROISIÈME PARTIE

Annexes

Outils de manipulation de drones

En plus des parties développement et recherche, nous avons eu l'occasion durant notre stage de manipuler et tester des drones réels (Voir Figure 5.1). Nous avons dû dans un premier temps apprendre à manipuler le véhicule aérien de façon manuelle d'abord, en utilisant une télécommande spécifique au drone, illustrée dans la figure 5.2.



FIGURE 5.1 – Drone utilisé.



FIGURE 5.2 – Télécommande de Drone.

Le dispositif qui permet le contrôle du drone et l'autopilotage est Pixhawk 4 (Voir figure 5.3). C'est un dispositif implanté à bord du drone comme l'indique la figure, permettant de contrôler des drones et de récupérer des informations sur le vol.

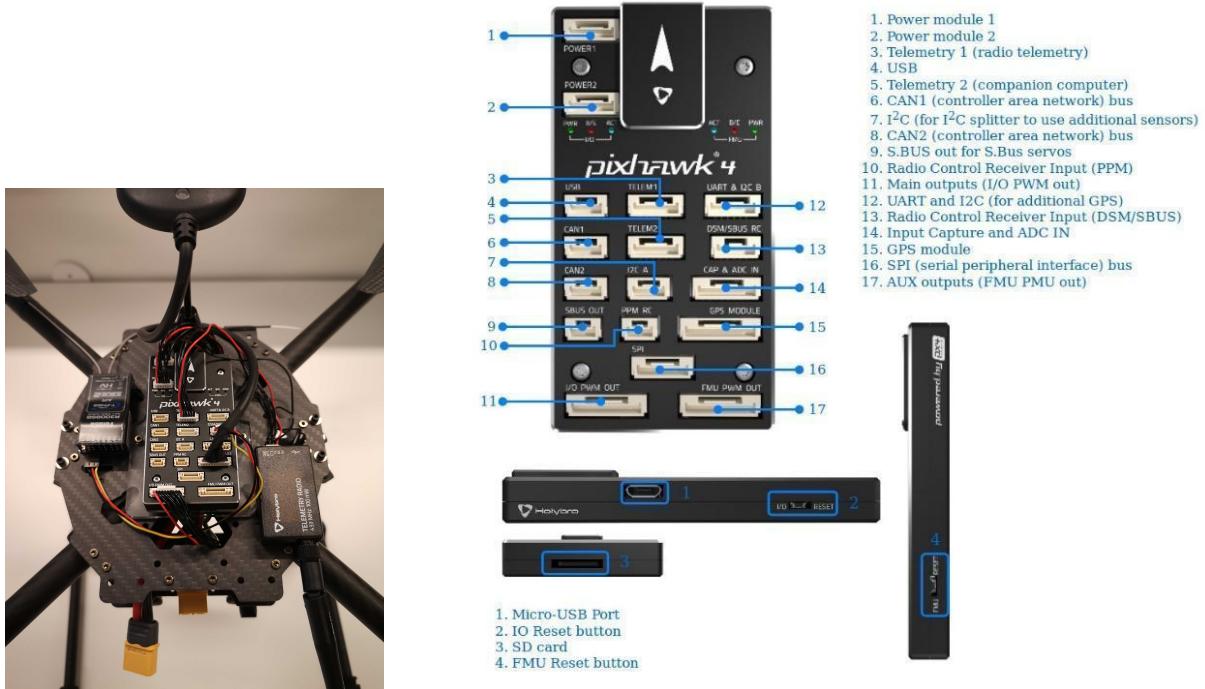


FIGURE 5.3 – Contrôleur Pixhawk4

Il possède de nombreux ports, permettant chacun de récupérer des informations précises sur le vol. En ce qui nous concerne, les ports nous intéressants sont le port GPS, et ce afin de garder une trace de la position du drone pendant le vol, mais aussi la télémétrie, qui permet de récupérer des informations de vol telles que la vitesse, l'altitude, l'accélération... etc. Ces informations peuvent s'avérer très utiles pour le suivi et l'analyse des vols.

Dans un second temps, nous avons eu à contrôler le drone à travers un logiciel Open-Source qui porte le nom de ARDUPilot Mission Planner (Figure 5.4). C'est un logiciel très utile permettant de visualiser les informations de télémétrie en temps réel durant le vol, ainsi que la planification de vol comme indiqué dans la figure, en définissant les points par lesquels le drone doit passer, les actions à entreprendre à chaque point (Attente, atterrissage...). Il permet aussi de garder les informations de vol sous forme de fichiers logs, qui peuvent plus tard servir à reproduire le vol dans d'autres logiciels de visualisation tels que Google Earth. Il est ainsi possible d'utiliser le drone pour faire de la reconnaissance, car à partir des fichiers logs on peut récupérer le champ de vision du drone pendant le vol.

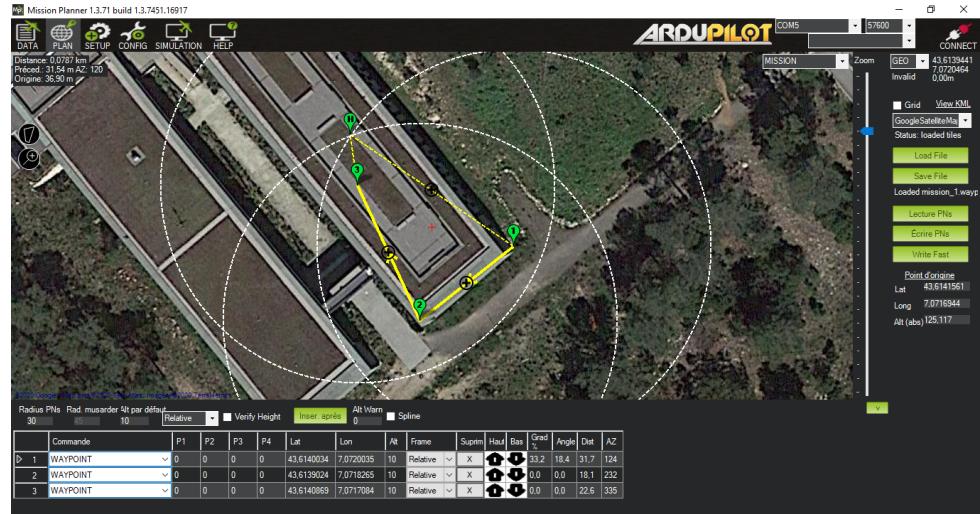


FIGURE 5.4 – ARDU Pilot Mission Planner.

Nous avons aussi l'occasion de programmer un mini-ordinateur, constitué des principales composantes physiques d'une carte mère (Processeur, RAM, ports USB... etc.) semblable à un système embarqué du type Raspberry (Voir figure 5.5).



FIGURE 5.5 – Mini-Ordinateur Drone.



FIGURE 5.6 – Configuration Mini-Ordinateur.



FIGURE 5.7 – Intégration Mini-Ordinateur au Drone.

Ce système a ensuite été programmé comme une machine ordinaire, c'est-à-dire en la connectant à un clavier, une souris et un écran et en y installant un système d'exploitation Linux (Ubuntu) comme l'indique la figure 5.6.

Ce dispositif a été connecté directement au drone comme l'indique la figure 5.7, pour permettre l'exécution de scripts comme l'application du GPS précédemment faite directement sur le drone, en utilisant une API appelée « Drone Kit ». Cette API fonctionne à l'aide d'un émetteur/récepteur de Télémétrie connectés au drone et à la machine sur laquelle nous souhaitons effectuer le contrôle du drone.

Elle permet entre autres de :

- Se connecter à un ou plusieurs drones à travers un script Python.
- Récupérer des paramètres de télémétrie à partir du drone en vol et en temps réel.
- Recevoir des notifications asynchrones des changements de situation du drone.
- Piloter le drone de façon précise.
- Faire exécuter des plans de vol préparés au préalable au drone.

Nous avons écrit un script Python (Figure 5.8) utilisant Drone Kit qui permet de récupérer les informations de télémétrie du drone pendant le vol, et les insère dans des fichiers JSON qui font office de fichiers logs de vol.

La connexion avec le véhicule se fait à la ligne 21, après avoir connecté le récepteur de télémétrie à l'ordinateur sur un port USB, et après avoir relayé le trafic entrant vers le port 14550 en utilisant un outil s'appelant MAVProxy, et en exécutant la commande :

```
mavproxy.py --master=/dev/ttyUSB0 -- out 127.0.0.1:14550
```

```

1  from dronekit import *
2  import time
3  import json
4  from datetime import datetime, date
5
6  init = {}
7  with open('telemetry_info.json', 'w') as json_file:
8      json.dump(init, json_file)
9
10 telemetry_infos = []
11
12 def fill_info(info):
13     with open('telemetry_info.json', 'r+') as file:
14         data = json.load(file)
15         data.update(info)
16         file.seek(0)
17         json.dump(data, file, sort_keys=True, indent=2)
18
19
20 #vehicle = connect('127.0.0.1:14550', wait_ready=True)
21 vehicle = connect('127.0.0.1:14550', wait_ready=False)
22
23 #vehicle.mode = VehicleMode("MISSION")
24 while (True):
25     # vehicle is an instance of the Vehicle class
26     print ("Autopilot Firmware version: ", vehicle.version)
27     #print ("Autopilot capabilities (supports ftp): ", vehicle.capabilities.ftp)
28     print ("Global Location: ", vehicle.location.global_frame)
29     print ("Global Location (relative altitude): ", vehicle.location.global_relative_frame)
30     print ("Local Location: ", vehicle.location.local_frame)      #NED
31     print ("Attitude: ", vehicle.attitude)
32     print ("Velocity: ", vehicle.velocity)
33     print ("GPS: ", vehicle.gps_0)
34     print ("Groundspeed: ", vehicle.groundspeed)
35     print ("Airspeed: ", vehicle.airspeed)
36     print ("Gimbal status: ", vehicle.gimbal)
37     print ("Battery: ", vehicle.battery)
38     print ("EKF OK?: ", vehicle.ekf_ok)
39     print ("Last Heartbeat: ", vehicle.last_heartbeat)
40     print ("Rangefinder: ", vehicle.rangefinder)
41     print ("Rangefinder distance: ", vehicle.rangefinder.distance)
42     print ("Rangefinder voltage: ", vehicle.rangefinder.voltage)
43     print ("Heading: ", vehicle.heading)
44     print ("Is Armable?: ", vehicle.is_armable)
45     print ("System status: ", vehicle.system_status.state)
46     print ("Mode: ", vehicle.mode.name)      # settable
47     print ("Armed: ", vehicle.armed)      # settable

```

FIGURE 5.8 – Dronekit pour la Télémétrie.