

Titre: Conception et implémentation en langage P4 de la fonction de passerelle d'accès de la 5G, et caractérisation de ses performances sur un commutateur logiciel
Title:

Auteur: Mohammed-Ayoub Elharti
Author:

Date: 2023

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Elharti, M.-A. (2023). Conception et implémentation en langage P4 de la fonction de passerelle d'accès de la 5G, et caractérisation de ses performances sur un commutateur logiciel [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/55072/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/55072/>
PolyPublie URL:

Directeurs de recherche: Tarek Ould-Bachir, & Yvon Savaria
Advisors:

Programme: Génie informatique
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Conception et implémentation en langage P4 de la fonction de passerelle d'accès
de la 5G, et caractérisation de ses performances sur un commutateur logiciel**

MOHAMMED-AYOUB ELHARTI

Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Août 2023

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Conception et implémentation en langage P4 de la fonction de passerelle d'accès
de la 5G, et caractérisation de ses performances sur un commutateur logiciel**

présenté par **Mohammed-Ayoub ELHARTI**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Frédéric CUPPENS, président

Tarek OULD-BACHIR, membre et directeur de recherche

Yvon SAVARIA, membre et codirecteur de recherche

Alejandro QUINTERO, membre

DÉDICACE

Je dédie cet ouvrage

À ma maman bien-aimée, ma source infinie de soutien, d'amour et d'inspiration. Tes encouragements inconditionnels et ta foi en moi m'ont donné la force de poursuivre ce parcours académique. Je te suis profondément reconnaissant pour ta présence constante et ton amour inébranlable. Cette thèse est le fruit de nos sacrifices mutuels et de notre lien indéfectible.

Merci pour tout ce que tu es, Je t'aime maman !

À mon cher père, Je te dit Merci pour ton amour, ton soutien et ta bienveillance. Merci pour tout ce que tu as fait pour moi, Je suis fier d'être ton enfant.

je t'aime profondément, infiniment.

À ma sœur extraordinaire, T'es une bénédiction céleste qui éclaire notre famille, qui la rend céleste. Dans mon cœur, tu occupes un sage. Mon amour pour toi ne cessera jamais de grandir et de prospérer. Je t'aime mon étoile !

À mes frères, Complices de ma vie, piliers sacrés, mes héros invisibles. je vous suis à jamais reconnaissant pour tout. Je vous aime !

À mes amis, qui m'ont épaulé tout au long de mon chemin et avec qui j'ai mes meilleurs souvenirs. Je vous aime !

...

REMERCIEMENTS

Je souhaite exprimer ma profonde gratitude envers les personnes qui ont contribué à la réalisation de ce mémoire. Leur soutien inestimable a rendu ce projet possible et je tiens à leur témoigner toute ma reconnaissance sincère.

En premier lieu, j'aimerais réserver des mots de remerciement à mes directeurs de recherche. Un sincère merci à mon directeur de recherche, Tarek Ould-Bachir, pour sa confiance, son soutien et son accompagnement attentif qui m'ont permis de mener à bien ce mémoire. Ce fut un plaisir d'avoir eu l'opportunité d'être guidé par une personne d'une grande expertise professionnelle et personnelle. J'aimerais également exprimer ma gratitude à mon co-directeur, Yvon Savaria. Son soutien constant, ses encouragements et ses conseils constructifs ont été d'une importance capitale dans la réussite de ce travail ainsi que dans mon développement personnel, Je lui adresse un grand merci pour sa disponibilité et sa gentillesse.

Je tiens à exprimer ma profonde gratitude envers les membres du jury, le professeur Alejandro Quintero et Frédéric Cuppens. Leurs conseils précieux et leur lecture attentive sont décisifs pour bien conclure ce projet. Je suis honoré d'avoir bénéficié de leur expertise, et je leur suis profondément reconnaissant d'avoir consacré du temps et de l'énergie à évaluer mon travail avec rigueur et impartialité.

Je souhaite également exprimer ma profonde gratitude envers l'organisme MITACS et la compagnie Kaloom pour leur soutien financier généreux qui a rendu la réalisation de ce mémoire de maîtrise possible. Grâce à leur appui financier, j'ai pu consacrer pleinement mes efforts à la recherche et au développement de mon travail. En outre, j'aimerais adresser mes sincères remerciements au personnel de Kaloom qui s'est grandement impliqué dans ce projet. Je tiens à exprimer ma reconnaissance envers Ludovic Beliveau, Eric Tremblay, Ebin Mathews et Anandakumar Udayamurthy pour leur soutien technique, leurs conseils avisés et leur expertise précieuse. Leur engagement indéfectible, leur disponibilité et leur patience ont grandement contribué à la réussite de mon mémoire.

Je tiens également à exprimer ma sincère reconnaissance à Vincent Brouillard pour son rôle essentiel dans l'initiation et la structuration de ce projet. Sa vision, son expertise et sa passion pour le sujet ont été des éléments clés qui ont permis de donner naissance à cette recherche. Son implication a été d'une importance cruciale et a contribué de manière significative à l'aboutissement de ce mémoire de maîtrise. Merci sincèrement, Vincent, pour ta confiance et ton aide précieuse.

RÉSUMÉ

L'Internet a profondément imprégné notre quotidien, transformant la structure fondamentale de la société et la dynamique des entreprises. Face à l'augmentation exponentielle de la demande, des défis architecturaux majeurs ont vu le jour. La cinquième génération de technologie de communication mobile, la 5G, a émergé comme une réponse à ces défis, envisageant une convergence des réseaux filaires et mobiles. Par ailleurs, les fournisseurs de services Internet doivent sans cesse optimiser leurs services tout en maîtrisant les coûts.

Historiquement, les passerelles de réseau à large bande (BNG) étaient les garantes de cet accès réseau. Cependant, avec l'avènement de la 5G, cette approche semble dépassée. Les principaux défis résident dans la capacité à concilier les exigences variées des applications modernes, allant des jeux en ligne en temps réel aux communications nécessitant une latence ultra-faible. La 5G, bien qu'adossée aux principes du réseau défini par logiciel (SDN) et de la virtualisation du réseau (NFV), nécessite des changements profonds dans son architecture, en particulier concernant l'accès fixe. C'est dans cette optique que la passerelle d'accès (AGF) se présente comme une alternative à la BNG traditionnelle, offrant des solutions adaptées aux exigences spécifiques du cœur du réseau 5G.

Les objectifs de ce mémoire se concentrent sur la conception, le développement et l'évaluation de la fonction de passerelle d'accès (AGF) en utilisant le langage P4. Le but ultime étant de caractériser ses performances, en particulier sur le commutateur DPDK, qui a démontré sa capacité à gérer un grand nombre de sessions tout en maintenant un débit satisfaisant.

Ce travail de maîtrise a permis d'atteindre plusieurs jalons significatifs dans cette voie. Une des principales réalisations a été la conception et le développement réussis de la fonction de passerelle d'accès (AGF) en utilisant le langage P4. La validation de ce prototype à travers différentes plateformes, notamment BMv2, Tofino, et DPDK, a démontré sa robustesse et sa flexibilité. Nos expérimentations avec le code P4 de l'AGF ont mis en lumière de très bonnes performances optimales, confirmant sa pertinence pour répondre aux exigences modernes de la communication. De plus, notre étude approfondie sur DPDK a révélé son efficacité, notamment en termes de gestion d'un nombre de sessions élevées avec des débits satisfaisants. Ces découvertes consolident non seulement la viabilité de l'AGF comme successeur potentiel de la BNG mais attestent également de la pertinence du langage P4 dans le développement de solutions réseau innovantes pour l'ère de la 5G.

ABSTRACT

The internet has profoundly permeated our daily lives, reshaping the foundational structure of society and the dynamics of businesses. In the face of the exponential rise in demand, significant architectural challenges have arisen. The fifth generation of mobile communication technology, 5G, has emerged as a response to these challenges, envisioning a convergence of wired and mobile networks. Additionally, internet service providers constantly strive to optimize their service while keeping costs in check.

Historically, Broadband Network Gateways (BNG) were the mainstay of this network access. However, with the advent of 5G, this approach seems outdated. The primary challenges lie in reconciling the varied requirements of modern applications, ranging from real-time online gaming to communications requiring ultra-low latency. While 5G, underpinned by the principles of Software-Defined Networking (SDN) and Network Function Virtualization (NFV), requires profound shifts in its architecture, especially concerning fixed access. It's in this light that the Access Gateway Function (AGF) presents itself as an alternative to the traditional BNG, offering solutions tailored to the specific demands of the 5G core network.

The objectives of this thesis focus on the design, development, and evaluation of the Access Gateway Function (AGF) using the P4 language. The ultimate goal being to characterize its performance, particularly on the DPDK switch, which has shown its ability to manage a high number of sessions while maintaining a satisfactory throughput.

This master's work has enabled the achievement of several significant milestones along this path. One of the main accomplishments was the successful design and development of the Access Gateway Function (AGF) using the P4 language. Validating this prototype across various platforms, notably BMv2, Tofino, and DPDK, showcased its robustness and adaptability. Our experiments with the P4 code of the AGF highlighted optimal performances, affirming its relevance in meeting modern communication demands. Furthermore, our in-depth study on DPDK revealed its efficacy, especially in terms of handling high session counts with satisfactory throughputs. These findings not only solidify the AGF's viability as a potential successor to the BNG but also vouch for the relevance of the P4 language in developing innovative network solutions for the 5G era.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	x
LISTE DES FIGURES	xi
LISTE DES SIGLES ET ABRÉVIATIONS	xii
CHAPITRE 1 INTRODUCTION	1
1.1 Contextualisation et cadre du projet	1
1.1.1 Convergence des réseaux filaire et mobile	1
1.1.2 Réseaux définis par logiciel	2
1.1.3 Langages de programmation	2
1.2 Enjeux du déploiement de la FMC dans la 5G	3
1.3 Objectifs de recherche	4
1.4 Organisation du mémoire	4
CHAPITRE 2 CONCEPTS DE BASE ET REVUE DE LITTÉRATURE	6
2.1 La 5G : Une nouvelle famille de technologies	6
2.2 Technologies d'appui	6
2.2.1 Réseau défini par logiciel	6
2.2.2 Virtualisation des fonctions du réseau	8
2.3 Le réseau central 5G	8
2.3.1 La session PDU	9
2.3.2 La fonction du plan usager	9
2.3.3 Déploiement de l'UPF	10
2.3.4 Utilisation du langage P4 dans le déploiement de l'UPF	11
2.4 Convergence des réseaux filaire et mobile	12

2.4.1	L'accès sans fil au réseau 5G	12
2.4.2	Les scénarios du déploiement de l'accès filaire	14
2.4.3	Le Broadband Network Gateway	15
2.4.4	AGF	15
2.5	Environnement de développement P4	17
2.5.1	Le langage P4	17
2.5.2	Programmation du plan de données	18
2.5.3	DPDK	19
2.5.4	Les commutateurs programmables en P4	21
CHAPITRE 3 MÉTHODE PROPOSÉE POUR LA MISE EN ŒUVRE DU PLAN		
	DE DONNÉES DE LA FONCTION PASSERELLE D'ACCÈS AGF	23
3.1	Spécifications techniques de l'AGF	23
3.1.1	Modes de fonctionnement	23
3.1.2	AGF en mode direct : protocoles et fonctionnalités	25
3.2	Conception du prototype de l'AGF	30
3.2.1	Structure du pipeline de l'AGF	30
3.3	Implémentation de pipeline basée sur P4	35
3.3.1	Le parseur d'entêtes	36
3.3.2	Les tables de correspondance du pipeline	38
CHAPITRE 4 EVALUATION DES PERFORMANCES : RÉSULTATS ET ANALYSE		
		42
4.1	Configuration expérimentale sur DPDK	42
4.1.1	Un mot sur le générateur de trafic	42
4.1.2	Banc d'essai	43
4.2	Métriques	43
4.3	Résultats expérimentaux	44
4.3.1	Débits	44
4.3.2	Taux de perte de paquets	45
4.3.3	Nombre de sessions simultanées supportées	45
4.3.4	Utilisation des ressources matérielles	47
4.4	Conclusions et recommandations	49
CHAPITRE 5 CONCLUSION		
		50
5.1	Synthèse des travaux	50
5.2	Limitations de la solution proposée	50
5.3	Améliorations futures	51

RÉFÉRENCES 52

LISTE DES TABLEAUX

Tableau 4.1	Le débit mesuré de l'AGF en augmentant le nombre de sessions . . .	44
Tableau 4.2	Taux d'utilisation de CPU	48

LISTE DES FIGURES

Figure 2.1	La différence entre le réseau traditionnel et le réseau SDN	7
Figure 2.2	L'architecture modulaire de l'UPF [1]	10
Figure 2.3	Les protocoles de communication entre la 5G-RG et la RAN	13
Figure 2.4	Différents scénarios de déploiement pour connecter les RG au réseau de données	14
Figure 2.5	FMIF pour connecter la BNG au réseau 5G	15
Figure 2.6	Le modèle de transfert abstrait du langage P4	18
Figure 2.7	Information du flux de données	19
Figure 2.8	By-passing du noyau Linux par DPDK	20
Figure 3.1	Vue architecturale de l'AGF reliant les RG au 5GC via des réseaux d'accès filaires	23
Figure 3.2	Les informations de contrôle type AS	26
Figure 3.3	Les piles de protocoles de communication entre le 5G-RG et l'UPF par le biais de l'AGF	26
Figure 3.4	La structure de l'en-tête 5WE	28
Figure 3.5	Le format de l'en-tête GTP-U	30
Figure 3.6	Exemple de flux de données	32
Figure 3.7	Fonctionnalités du pipeline Upstream de l'AGF	33
Figure 3.8	Fonctionnalités du pipeline Downstream de l'AGF	35
Figure 3.9	La vue d'ensemble complète du prototype AGF-P4	36
Figure 3.10	Le parser de l'AGF	39
Figure 4.1	Une configuration de test avec un générateur de trafic	42
Figure 4.2	Banc d'essai utilisé pour l'évaluation de performances avec DPDK	43
Figure 4.3	Le taux de perte de paquets pour les différentes cas de test	46

LISTE DES SIGLES ET ABRÉVIATIONS

AGF	Access Gateway Function
AN	Access Network
ASIC	Application-Specific Integrated Circuit
BBF	Broadband Forum
BNG	Broadband Network Gateway
CN	Core Network
DPDK	Data Plane Development Kit
DSL	Domain-specific language
EAL	Environment Abstraction Layer
FMC	Fixed Mobile Convergence
GPRS	General Packet Radio Service
GTP	GPRS Tunneling Protocol
IP	Internet Protocol
NAS	Non-Access Stratum
NFV	Network Function Virtualisation
NIC	Network Interface Card
PDU	Protocol Data Unit
RAN	Radio Access Network
SDN	Software Defined Networking
UE	User Equipment
UPF	User Plane Function
AMF	Access and Mobility Management Function
SMF	Session Management Function
CPE	Customer Premises Equipment
WAN	Wide Area Network
FMIF	Fixed Mobile Interworking Function
PSA	Portable Switch Architecture
PNA	Portable NIC Architecture
FPGA	Field Programmable Gate Array
PPPoE	Point to Point Protocol over Ethernet
QFI	QoS Flow Identifier
DSCP	Differentiated Service Code Point
GBR	Guaranteed Bit Rate

UDP	User Datagram Protocol
MAC	Media Access Control
PCP	Priority Code Point
IP	Internet Protocol
VLAN	Virtual Local Area Network
ATSSS	Access Traffic Steering, Switching and Splitting

CHAPITRE 1 INTRODUCTION

1.1 Contextualisation et cadre du projet

Au cours des dernières décennies, l'Internet est devenu un élément indispensable dans notre vie quotidienne. Aujourd'hui, presque tout le monde a accès à Internet, ce qui a contribué à catalyser une transformation radicale de la société. Les entreprises ont numérisé leurs processus et ont évolué vers des modèles commerciaux en ligne. Ainsi, l'Internet a permis l'émergence de nouvelles opportunités de marché et transformé notre façon de nous instruire et de nous divertir, mais il a également intensifié la demande. Cette charge massive sur le réseau a créé de nouveaux défis architecturaux, dont le sujet de ce mémoire est un exemple éloquent.

C'est dans ce cadre que la cinquième génération de technologie de communication mobile, connue sous le nom de 5G, a été conçue. Elle vise à fournir des vitesses de données supérieures, à réduire les temps de latence et à augmenter la capacité du réseau. Cette génération est venue en réponse à l'émergence, avérée ou attendue, de nouvelles technologies de pointe telles que l'Internet des objets, les véhicules connectés ou les technologies vestimentaires (*wearable technologies*) [2]. Dans le sillage de ce développement, l'idée de faire converger les réseaux mobiles et filaire a émergé. Cette solution a notamment été rendue possible grâce à l'émergence des réseaux définis par logiciel.

1.1.1 Convergence des réseaux filaire et mobile

La convergence des réseaux filaires et mobiles dans la technologie 5G est une évolution stratégique importante pour plusieurs raisons. D'une part, cette fusion offre une plus grande flexibilité et efficacité dans l'utilisation des ressources de réseau. Les utilisateurs peuvent ainsi bénéficier de vitesses de connexion améliorées et d'une expérience plus fluide, indépendamment du type de connexion utilisé. D'autre part, une infrastructure réseau unifiée peut faciliter l'interconnectivité entre divers appareils et plateformes. Notons également qu'une architecture réseau intégrée peut aider à simplifier la gestion et la maintenance du réseau, ce qui peut se traduire par des économies de coûts [3].

Cependant, la convergence des réseaux filaires et mobiles dans la 5G présente plusieurs défis. Un des principaux défis est la nécessité d'une gestion complexe du trafic pour garantir un passage fluide entre les connexions filaires et sans fil. Cela nécessite le développement de nouveaux protocoles et mécanismes de gestion de réseau. Un autre défi est le besoin

d'investissements significatifs en infrastructure pour mettre à niveau les réseaux existants et construire de nouvelles installations compatibles 5G. Enfin, des problèmes de sécurité peuvent également survenir en raison de l'augmentation de la complexité et de l'interconnectivité du réseau, nécessitant des solutions de sécurité réseau plus sophistiquées et plus robustes.

1.1.2 Réseaux définis par logiciel

Un de ces concepts clé dans le déploiement de la 5G est celui de réseau défini par logiciel (*software defined networking* : SDN). Il s'agit d'une approche qui permet de réaliser un saut qualitatif capable d'élever les fonctionnalités du réseau à un niveau supérieur. Le SDN procède en découplant le plan des données du plan de contrôle, permettant de ce fait de les programmer séparément [4]. Les protocoles et les algorithmes réseau sont implémentés dans un logiciel du contrôleur centralisé, tandis que les dispositifs de plan de données, voire les commutateurs programmables, exécutent un ensemble configurable d'actions prédéfinies. Ces qualités ont fait du SDN la pierre angulaire dans le déploiement rapide de la 5G.

Le modèle SDN a également modifié d'importantes abstractions dans le domaine du réseau. Au lieu d'imposer des piles de protocoles prédéfinies dans les implémentations des commutateurs réseau traditionnelles, le SDN repose sur une séparation nette des préoccupations :

- Les utilisateurs développent des applications réseau en définissant des règles de réseau dans le plan de gestion sur la base de l'état du réseau signalé par l'entité centrale ;
- Un contrôleur centralisé capture l'état du réseau et communique avec le plan de données pour installer des règles de routage du trafic ;
- Le plan de données quant à lui transfère les paquets en utilisant des règles de routage définies.

1.1.3 Langages de programmation

Pour appuyer l'intégration du SDN dans le domaine de la réseautique, il était nécessaire de développer une interface spécifique et à accès ouvert permettant la communication entre le plan de contrôle et le plan de données. Pour cela en 2008, le protocole OpenFlow a été introduit comme interface standardisée permettant de configurer des actions prédéterminées dans le plan de contrôle pour qu'elles soient communiquées et utilisées dans le plan de données.

Afin de faciliter l'adoption du SDN en réseautique, il était impératif de concevoir une interface ouverte et dédiée pour la liaison entre les plans de contrôle et de données. Dans cette optique, le protocole OpenFlow a été lancé en 2008 comme une norme pour définir des actions dans le plan de contrôle, qui sont ensuite relayées et mises en œuvre dans le plan de données [5].

Rapidement, OpenFlow est devenu le langage *de facto* dans le domaine de la réseautique. Cependant, OpenFlow imposait certaines limitations, notamment en niveau de la granularité, l'évolutivité et la performance. Ces éléments ont conduit à la création du langage P4 pour son remplacement.

P4 est langage spécifique au domaine (*domain-specific language* : DSL) destiné à la programmation des commutateurs réseau dont le nom vient des quatre 'P' formant "*Programming Protocol-independent Packet Processors*" [6]. Il a été introduit en 2014 pour combler les lacunes d'OpenFlow. Le langage P4 permet aux utilisateurs de définir un comportement personnalisé de traitement des paquets qui n'est pas limité par les protocoles. Ainsi, il offre aux utilisateurs la possibilité de spécifier leur propre ensemble de champs de correspondance de trafics et de déterminer les actions à entreprendre le cas échéant.

1.2 Enjeux du déploiement de la FMC dans la 5G

Ce mémoire s'articule autour de la problématique de la convergence des réseaux mobiles et fixes (FMC) dans le cadre du développement de la 5G. Traditionnellement, ce sont les passerelles de réseau à large bande (*Broadband network gateway* : BNG) qui assurent l'accès réseau résidentiel. Ce sont des fonctions réseaux exclusivement conçues pour fournir des performances élevés, mais qui sont coûteuses à mettre en place. Bien que l'architecture du réseau 5G repose sur les principes du SDN et de la NFV, l'utilisation d'une passerelle BNG ne semble pas être appropriée pour offrir un accès fixe au cœur du réseau 5G. En effet, la BNG ne prend pas en charge les interfaces essentielles permettant de communiquer avec le réseau 5G, ni les autres fonctionnalités nécessaires pour assurer la gestion du trafic des utilisateurs finaux.

De ce fait, la passerelle d'accès (*Access Gateway Function* : AGF) s'est imposée comme le successeur de la BNG dans le réseau 5G [7]. L'AGF offre une solution plus globale, spécifiquement conçue pour prendre en charge les interfaces nécessaires pour interagir avec à la fois le plan de données et le plan de contrôle du réseau 5G. De plus, l'AGF répond également à d'autres exigences liées au cœur de la 5G, telles que la qualité de service, la gestion du trafic et de l'accès. Ce travail vise ainsi à explorer et à contribuer aux efforts déployés pour concrétiser cette vision de la convergence, en garantissant une transition fluide et une interopérabilité optimale entre les deux types de réseaux.

Les principaux facteurs de coût dans ce contexte sont les fonctions réseau à haut rendement nécessaires au niveau de la périphérie d'accès. En plus de fournir une connectivité Internet, la périphérie d'accès doit également gérer les authentifications et faire respecter les aspects

de qualité de service dans les contrats d'accès à Internet des abonnés résidentiels. Ceci se comprend d'autant plus que les exigences des clients et des fournisseurs de contenu forcent les fournisseurs de services Internet à fournir un haut rendement aux abonnés résidentiels à des coûts de plus en plus bas.

Dans cette optique, ce mémoire se propose d'étudier la pertinence d'une solution de commutation réseau basée sur un processeur à usage général, comme le x86 ou ARM, pour la gestion des commutations réseau. Plus précisément, nous allons nous pencher sur l'utilisation du commutateur logiciel DPDK (*Data Plane Development Kit*) [8], qui promet une optimisation des performances tout en potentiellement réduisant les dépenses. Cette exploration vise à déterminer si une telle solution peut offrir un bon équilibre entre efficacité opérationnelle et maîtrise des coûts.

1.3 Objectifs de recherche

Ce mémoire s'inscrit dans un effort global de conception de la fonction de passerelle d'accès AGF à l'aide du langage P4. En plus des tâches reliées à l'implémentation, nous nous fixons pour objectif de tester notre solution sur différentes plateformes logicielles et matérielles, notamment la mise en œuvre sur le commutateur logiciel DPDK. Nous veillerons ensuite à en caractériser les performances.

Ainsi, nos objectifs spécifiques s'énoncent comme suit :

- Détailler toutes les exigences de l'application étudiée en termes de qualité de services, débit et l'accès en ressources matérielles.
- Développer un prototype de la fonction de passerelle d'accès sur le langage P4.
- Valider les fonctionnalités du prototype sur plusieurs plate-formes, nommément BMv2 et DPDK.
- Caractériser les performances du prototype d'AGF développé sur le commutateur logiciel DPDK.

1.4 Organisation du mémoire

Le présent mémoire est structuré de la manière suivante : Le Chapitre 2 offre une exploration approfondie des concepts fondamentaux, ainsi qu'une revue de divers travaux pertinents issus de la littérature qui ont joué un rôle crucial pour l'accomplissement de nos objectifs de recherche. Dans le Chapitre 3, nous détaillons les exigences fonctionnelles spécifiques de la fonction de passerelle d'accès AGF. De plus, nous décrivons la méthodologie adoptée

pour élaborer un modèle de l'AGF en exploitant le langage P4 et les différentes étapes impliquées dans l'implémentation de ce modèle, que ce soit sur le commutateur BMv2, le Tofino ou le commutateur logiciel DPDK. Le Chapitre 4 se consacre à la présentation de notre configuration expérimentale, en mettant l'accent sur la méthodologie utilisée et l'analyse détaillée des résultats obtenus. Finalement, le Chapitre 5 offre une conclusion récapitulative de l'ensemble du mémoire, soulignant les principales réalisations et conclusions.

CHAPITRE 2 CONCEPTS DE BASE ET REVUE DE LITTÉRATURE

2.1 La 5G : Une nouvelle famille de technologies

La 5G est la plus récente version du réseau mobile qui est actuellement en cours de déploiement dans plusieurs régions du monde. Les avancées technologiques récentes en matière de réseaux et de communication radio ont permis l'introduction de nouveaux cas d'utilisation et d'applications, en mettant l'accent sur trois scénarios principaux [9] :

- Enhanced Mobile BroadBand (eMBB) : La 5G améliore l'expérience de l'utilisateur mobile en mettant l'accent sur une bande passante plus élevée, une latence plus faible et un meilleur support pour les zones densément peuplées.
- Ultra-Reliable-Low latency communications (URLLC) : La 5G répond aux besoins des trafics critiques avec des exigences particulières en termes de latence et de fiabilité, tels que les réseaux véhiculaires.
- Massive machine type communications (mMTC) : En se concentrant sur le trafic substantiel de machines à machines qui se développera en raison de l'expansion des dispositifs IoT.

Afin de soutenir chaque scénario, il est nécessaire d'intégrer les technologies les plus performantes disponibles tout en optimisant les ressources et en offrant une grande flexibilité au système. Plus précisément, cette approche doit mettre en œuvre une architecture qui peut prendre en charge tous les scénarios énumérés sur une infrastructure physique commune.

Un des défis majeurs que la technologie 5G ambitionne de relever est la convergence des réseaux mobiles et fixes. Cette convergence vise à harmoniser l'accès et les services sur les deux types de réseaux, offrant ainsi une expérience utilisateur sans coupure, tout en optimisant les ressources réseau. Dans ce chapitre, nous entreprenons un survol des concepts essentiels qui sous-tendent cette technologie, en mettant l'accent sur ladite convergence. Le travail fourni dans ce mémoire propose également des contributions à cette thématique de recherche.

2.2 Technologies d'appui

2.2.1 Réseau défini par logiciel

Au lieu de donner aux utilisateurs des réseaux la possibilité d'avoir le contrôle sur le réseau et de répondre eux-mêmes à leurs divers besoins, c'est le matériel réseau standard à fonctionnalité fixe qui a historiquement défini le comportement du réseau et comment les

trafics devraient être transférés [10]. Les commutateurs ASIC traditionnels, dont les cycles de conception sont long, ont perdu la faveur des acteurs du marché dans l’environnement Internet actuel, qui évolue rapidement. Au lieu de développer constamment de nouveaux commutateurs pour répondre à l’évolution des besoins, les utilisateurs préfèrent être en mesure de déployer de nouvelles applications quand ils le souhaitent en utilisant une interface logicielle. En outre, alors que la création de nouveaux commutateurs ASIC peut prendre des mois, voire des années, les applications des utilisateurs évoluent beaucoup plus rapidement.

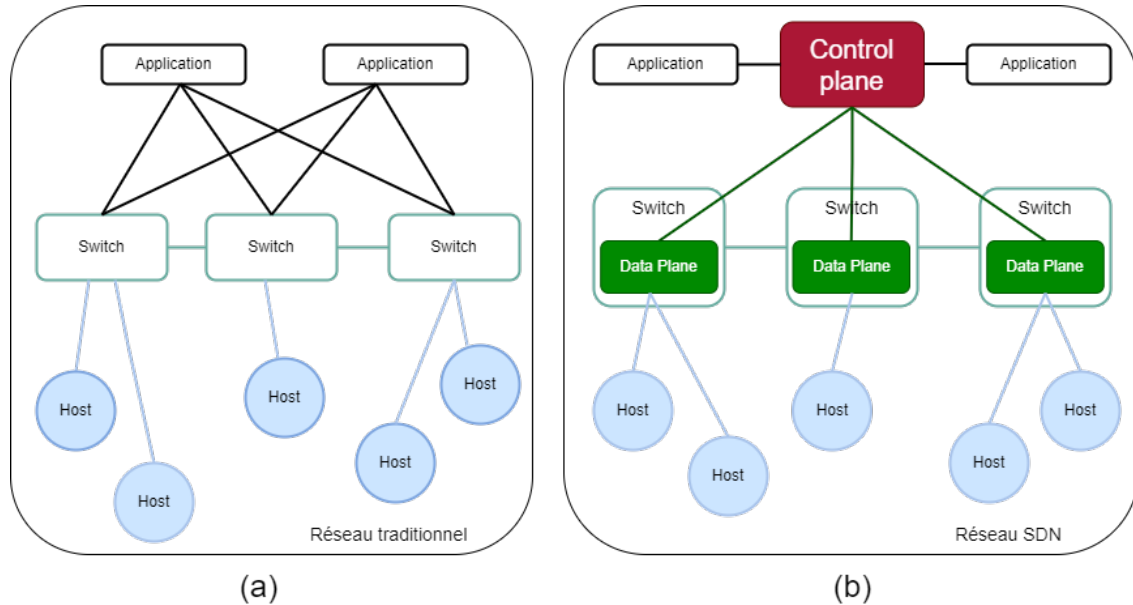


FIGURE 2.1 La différence entre le réseau traditionnel et le réseau SDN

Cette exigence a donné naissance au paradigme du réseau défini par logiciel (*Le Software-Defined Networking : SDN*) [11]. Les plans de contrôle et des données des dispositifs de mise en réseau sont entièrement séparés par le SDN, tel qu’illustré à la Figure 2.1. Les applications qui ont besoin d’informations sur l’ensemble du réseau n’ont plus qu’à les demander au plan de contrôle au lieu d’interroger chaque commutateur pour s’informer de l’état du réseau comme c’était le cas traditionnellement.

De plus, grâce à l’abstraction par l’utilisateur des implémentations matérielles de bas niveau, le SDN a rendu le réseau quasi-entièrement programmable. Par conséquent, les utilisateurs n’ont plus qu’à se préoccuper de créer les applications réseau qu’ils souhaitent. Les commutateurs sont ainsi réduits à un ensemble d’opérations effectuées sur des paquets conformément à des règles de correspondance préétablies [10]. Dans le SDN, un contrôleur chargé de mettre en place et de superviser un groupe de commutateurs SDN sert de centre logique de l’intel-

ligence du réseau. L'interface entre le contrôleur et les commutateurs est claire et simple. Le paradigme SDN aspire à une grande capacité d'évolution et est indépendant des fournisseurs et des protocoles.

2.2.2 Virtualisation des fonctions du réseau

Dans les réseaux traditionnels, il existe une connexion directe entre le matériel et les fonctions. Cependant, dans un réseau basé sur la virtualisation des fonctions du réseau (*Network function virtualization* : NFV), les différentes fonctions ne sont pas liées à des dispositifs physiques spécifiques. Au lieu de cela, elles sont exécutées sur des serveurs standards capables de gérer une variété de tâches.

Pour clarifier le concept, considérons l'exemple d'un routeur typique. Dans un réseau traditionnel, les fonctions de routage sont exécutées sur un dispositif dédié et spécialisé, qui est le routeur lui-même. Avec l'approche NFV, ces fonctions peuvent être exécutées sur un serveur standard, dans un environnement virtuel [12].

Comparé aux réseaux traditionnels, cette virtualisation offre une grande flexibilité, permettant aux développeurs de réseaux de mettre à l'échelle les différentes ressources réseau en fonction de la demande réelle ou des besoins d'un cas d'utilisation particulier [13]. Par conséquent, en ajoutant des ressources supplémentaires au système physique et en introduisant plus d'instances des fonctions nécessaires, le déploiement d'un réseau peut être facilement mis à l'échelle et modifié. L'approche NFV est généralement mise en œuvre en étroite collaboration avec la technologie SDN, ce qui permet de créer un vaste système de réseau programmable qui peut être facilement géré et adapté pour répondre à différents scénarios d'utilisation.

2.3 Le réseau central 5G

Le système 5G peut être divisé en deux parties, le réseau d'accès (*Access Network* : AN) et le réseau central (*Core Network* : CN). L'AN connecte les équipements utilisateur au reste de l'infrastructure, tandis que le CN permet l'accès à un ou plusieurs réseaux de données et comprend le routage et d'autres fonctions de réseau. Le réseau des données peut être un réseau générique, tel que l'Internet, ou un réseau spécifique créé par l'opérateur lui-même tel qu'un réseau d'accès local [14].

2.3.1 La session PDU

L'objectif principal du réseau 5G est de permettre l'accès aux réseaux des données externes via une session dite d'unité des données de protocole (*Protocol Data Unit session* : PDU session). Dans le contexte de la 5G, la session PDU sert de liaison entre l'équipement utilisateur et le réseau des données cible [15]. À un niveau élevé, la session PDU peut être décrite comme la connexion entre l'équipement usager (*user equipment* : UE) et le réseau des données cible, tandis qu'à un niveau inférieur, elle fournit un service de transport d'unités de données de protocole (PDU) - telles que les paquets utilisateur - à travers le plan utilisateur.

La session PDU dans l'environnement 5G utilise une série de tunnels pour encapsuler les paquets transmis entre l'UE et la fonction du plan de l'usager (*User Plane Function* : UPF). Cela permet des fonctionnalités clés de l'environnement 5G, telles que la mobilité et l'accès aux réseaux non-3GPP. En général, la session PDU est mise en œuvre à l'aide d'un protocole de tunnélisation IP, tel que le GTP [14].

2.3.2 La fonction du plan usager

L'UPF est une composante critique du réseau central 5G. Sa fonction principale est de gérer les paquets de données provenant de l'UE et de les diriger vers d'autres réseaux. Essentiellement, il sert de point d'entrée et de sortie pour l'UE lorsqu'il se connecte à des réseaux externes tels que l'Internet. Au-delà d'Internet, il existe d'autres réseaux des données auxquels l'équipement utilisateur peut accéder via l'UPF dans le réseau central 5G. Il s'agit notamment de réseaux des données spécifiques à une application et de réseaux restreints à l'opérateur, telle que la voix sur IP.

La Figure 2.2 représente le module architectural de l'UPF et ses principales fonctions. Elle est principalement composée de deux parties : un sous-module de contrôle et un sous-module de transfert. Le sous-module de contrôle est spécifiquement conçu pour gérer les fonctions de contrôle non temps réel, tandis que le sous-module de transfert est responsable du traitement de l'acheminement des paquets de données et du suivi des statistiques entre les interfaces N3, N6 et N9.

Le module de contrôle de l'UPF communique avec la session de gestion de fonction (*Session management function* : SMF) via l'interface N4. Ces modules échangent les informations des différentes sessions sur le réseau, surtout au moment de l'établissement de nouvelles sessions ou au moment de sur-utilisation. Dans ce cas, l'UPF reçoit les informations des nouvelles sessions utilisateur du SMF via l'interface N4. Ce module garde dans le plan de contrôle un contexte des différentes informations reçues, voire la session ID, la bande passante

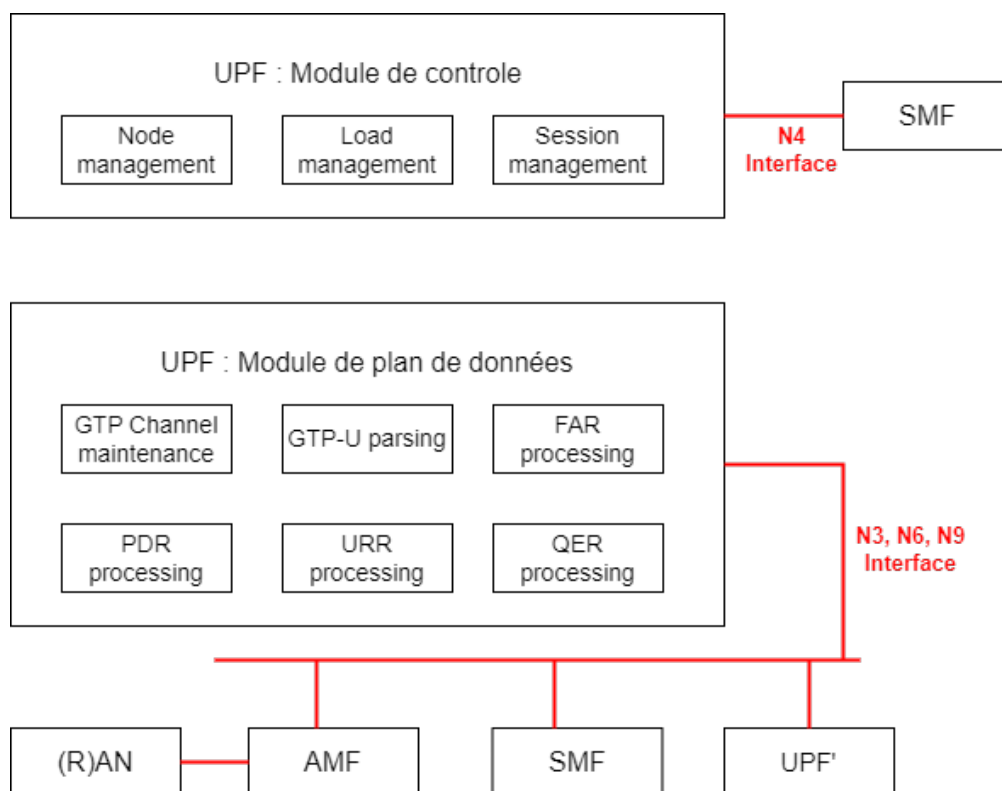


FIGURE 2.2 L'architecture modulaire de l'UPF [1]

accordée à cet utilisateur, etc [16]. Afin de router et de transmettre les données entre les interfaces N3 et N6 du plan utilisateur, et de transmettre les informations d'utilisation de l'utilisateur via l'interface N4. L'interface N4 est utilisée pour traiter les messages de protocole de contrôle entre l'élément de réseau de plan de contrôle SMF et est responsable de l'analyse et du traitement des messages de plan de contrôle. Cela implique une logique de traitement complexe qui nécessite beaucoup de recherches de table et d'opérations de correspondance, par rapport au plan utilisateur. Cependant, les exigences en temps réel pour l'interface N4 sont généralement faibles.

2.3.3 Déploiement de l'UPF

L'UPF est couramment exécutée sur des processeurs à usage général, basés sur la pile de protocoles réseau du noyau [1]. Toutefois, elles rencontraient des limitations de performances en raison des charges induites par le traitement des données sur le noyau du CPU. L'un des problèmes principaux à résoudre est celui des interruptions. Lorsqu'un volume important de paquets de données est reçu dans le réseau, cela entraîne la génération de nombreuses de-

mandes d'interruption matérielles. En conséquence, ces interruptions matérielles deviennent sujettes à des interruptions elles-mêmes. Si la fréquence des interruptions est élevée, cela entraînera une surcharge de performance plus importante. Un autre problème causé par le traitement basé sur le noyau est que les paquets de données reçues subissent une série d'opérations de copie en mémoire lorsqu'ils traversent la carte réseau en accord avec le programme d'application. Ce processus de copie prend 57.1% du temps total de traitement. Pour surmonter cette limitation, plusieurs approches ont été proposées dans la littérature [1, 15, 17].

Le prototype de l'UPF développé par [1] est une UPF à haute performance, basée sur une solution entièrement logicielle. Cette solution utilise la technique de contournement complet du noyau du système d'exploitation. Elle permet de traiter directement les paquets dans l'espace usager. Pour cela, le prototype utilise des infrastructures logicielles telles que le Data Plane Development Kit (DPDK), qui exploite efficacement les cœurs du CPU en effectuant une surveillance continue de la carte d'interface réseau (NIC) pour les paquets entrants [8].

La technique proposée dans [15] consiste à exécuter une partie du traitement des paquets avant qu'ils n'entrent dans la pile réseau du noyau. Cela est réalisé en exploitant deux fonctionnalités techniques du noyau Linux : le filtre de paquets BSD étendu et le chemin de données eXpress. Ces approches permettent de filtrer les paquets et de les acheminer vers un port de destination en fonction de leur nature spécifique. En utilisant ces techniques, le traitement des paquets peut être accéléré, améliorant ainsi les performances globales du système.

Dans [17], les auteurs ont implémenté l'UPF et comparé cette implémentation avec d'autres, allant d'une UPF basée uniquement sur le logiciel DPDK à des implémentations exploitant du matériel programmable. Après évaluation, bien que des techniques de délégation améliorent les performances dans certains cas, elles présentent des limites. Surpasser ces contraintes demande une délégation de fonctions plus complexes.

2.3.4 Utilisation du langage P4 dans le déploiement de l'UPF

L'émergence des puces dédiées (*Application-specific integrated circuit* : ASIC) programmables et des langages de haut niveau spécifiques aux réseaux, tels que P4, a ouvert de nouvelles possibilités pour la conception et la mise en œuvre des différentes fonctionnalités de l'UPF sur des plateformes matérielles. Dans le domaine de la mise en œuvre de l'UPF pour la 5G, plusieurs études ont souligné l'importance du langage P4.

Les auteurs de [18] proposent des solutions hybrides de pipelines pour l'UPF qui exploitent le langage P4 pour allouer des chemins de traitement de paquets en fonction des caractéristiques

de flux. Cette approche, tout en facilitant la mise à l'échelle, réduit également la latence jusqu'à 50%.

Dans [19], les auteurs introduisent une architecture de coexistence pour l'UPF, baptisée CeUPF. Dans ce contexte, l'accent est mis sur le rôle de CeUPF permettant une cohabitation entre les fonctions logicielles existantes et les performances élevées de l'acheminement matériel, en particulier en déchargeant vers un commutateur P4. Les résultats montrent que le déchargement de fonctionnalités (*offload*) vers le commutateur P4 présente des avantages significatifs en termes de bande passante et de débit.

Enfin, [20] offre une perspective sur la conception des réseaux mobiles en utilisant deux programmes P4 pour implémenter le plan de données 5G, mettant en évidence la flexibilité et l'efficacité du langage pour s'adapter aux besoins changeants des réseaux. La mise en œuvre est actuellement en déploiement limité, permettant de nouvelles recherches sur les réseaux mobiles.

2.4 Convergence des réseaux filaire et mobile

Les opérateurs voient dans l'arrivée de la 5G une occasion de fusionner les réseaux fixes et mobiles au-delà de la simple convergence structurelle, où les fonctions fixes et mobiles coexistent sur une infrastructure partagée [21]. La convergence fonctionnelle, en particulier, permet d'avoir un seul plan de contrôle pour les connexions filaires et sans fil. Les motivations de cette convergence sont multiples, notamment offrir aux clients une expérience de service fluide et indépendante de l'accès, permettre une connectivité multi-accès, rationaliser les fonctions réseau, réaliser une intégration et des services communs entre les divisions fixes et mobiles, et étendre la portée géographique des réseaux 5G. En réalisant cette convergence, les opérateurs peuvent optimiser leurs opérations, améliorer l'expérience client et exploiter les capacités de la technologie 5G dans leur infrastructure réseau.

Dans cette partie, nous allons voir le mode d'accès traditionnel au réseau 5G qui se contentait sur l'accès mobile. Par la suite, nous allons élaborer les différents scénarios de déploiement de l'accès filaire que le BBF en collaboration avec le 3GPP avaient proposé pour rendre cette convergence possible.

2.4.1 L'accès sans fil au réseau 5G

La standardisation du réseau d'accès radio (*Radio Access Network* : RAN) représente une évolution majeure de la technologie 4G LTE, notamment en ce qui concerne la pile de protocoles, les fonctionnalités et les capacités. La Figure 2.3 décrit la pile des protocoles du

plan utilisateur utilisés pour l'échange des sessions PDU entre un équipement 5G-RG et un RAN dans un réseau 5G [22], elle est constituée de deux couches : la couche 1, qui définit la couche physique (PHY), et la couche 2, qui englobe les sous-couches MAC (contrôle d'accès au support), RLC (contrôle de liaison radio), PDCP (protocole de convergence de données en paquets) et SDAP (protocole d'adaptation des données de service). Ces entités standardisées visent à introduire la flexibilité, à améliorer le découplage et à favoriser la compatibilité entre différents fournisseurs. De plus, le contrôle des ressources radio (RRC) est responsable de tâches telles que l'établissement, la reconfiguration et la libération de porteurs radio, l'exécution de procédures de mobilité, ainsi que la gestion des notifications de pagination (*paging*) ou des informations de diffusion système.

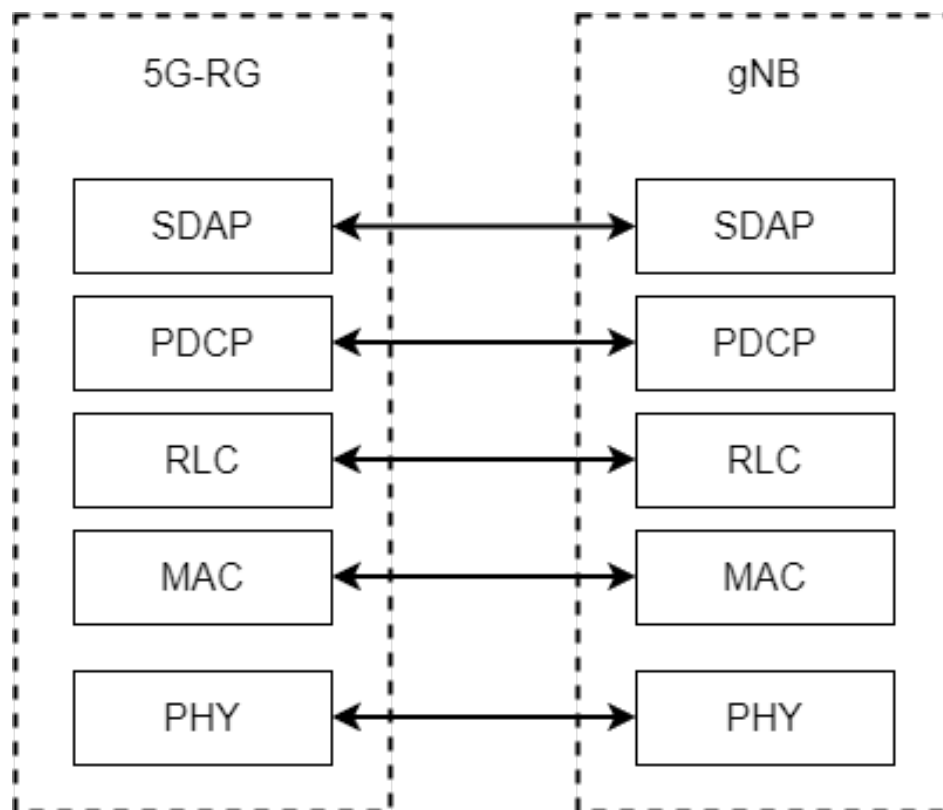


FIGURE 2.3 Les protocoles de communication entre la 5G-RG et la RAN

2.4.2 Les scénarios de déploiement de l'accès filaire

Les efforts déployés par le Broadband Forum (BBF) dans le domaine de la convergence fixe-mobile 5G visent à permettre plusieurs scénarios pour prendre en charge différents environnements de réseau, points de départ et priorités.

La Figure 2.4 représente trois scénarios de déploiement possibles présentant différents types de passerelles résidentielles (RG), réseaux d'accès et modèles d'interface avec le cœur 5G. Dans ces scénarios, les dispositifs prenant en charge les procédures 3GPP (UE) peuvent accéder au réseau cœur 5G par le biais d'un nœud d'accès qui peut être soit un multiplicateur d'accès DSL (DSLAM) ou une terminaison de ligne optique (OLT) en utilisant le RG comme intermédiaire.

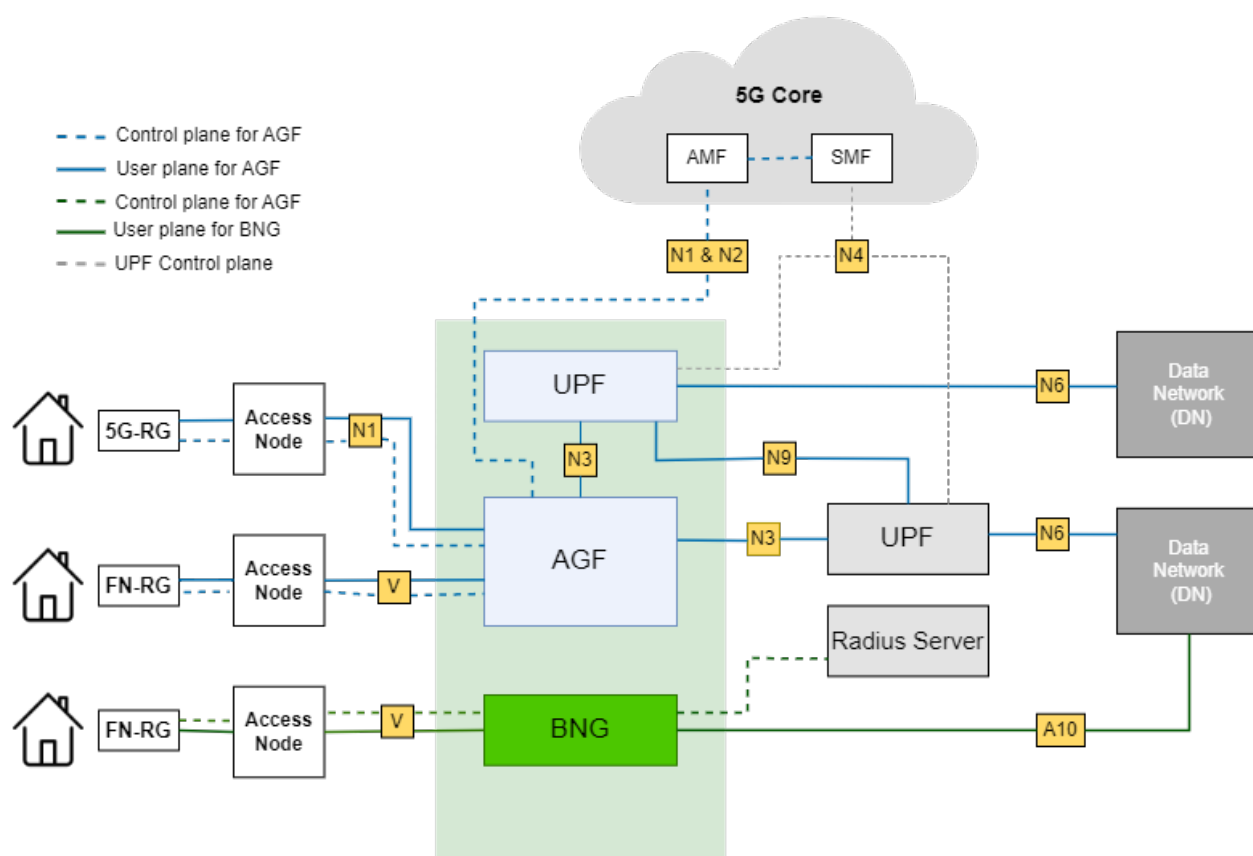


FIGURE 2.4 Différents scénarios de déploiement pour connecter les RG au réseau de données

L'architecture est conçue pour prendre en charge deux catégories de types d'équipements d'abonné en local (*Customer Premises Equipment* en anglais : CPE) : les 5G-RG et les RG de réseau fixe (FN-RG). Les 5G-RG sont ceux utilisés dans l'accès mobile, et ils ont été améliorés pour supporter l'accès filaire aussi. ils peuvent être équipés soit d'interfaces filaires

uniquement, soit d'interfaces sans fil uniquement (accès sans fil fixe), soit des deux types d'interfaces de réseau étendu (WAN), contrairement aux FN-RG qui ne supporte que l'accès filaire et qui ne prennent pas en charge les procédures 3GPP.

2.4.3 Le Broadband Network Gateway

Le premier scénario de déploiement représente l'accès traditionnel au réseau résidentiel dont la passerelle de réseau à large bande (BNG) est responsable. Dans ce déploiement, Le FN-RG est connecté au BNG qui agit en tant que point d'entrée principal pour les abonnés résidentiels, leur fournissant une connectivité au réseau de données de leur fournisseur de services spécifique. Lorsque les abonnés se connectent à la BNG, celle-ci assume la responsabilité de gérer leur accès en effectuant des tâches importantes telles que l'authentification, l'autorisation et la comptabilité des abonnés (AAA), ainsi que d'autres fonctions de gestion pertinentes voire assigner des adresses IP, assurer la qualité de service et la gestion des trafics [23].

La BNG toute seule n'est pas compatible pour se connecter et interagir avec le réseau 5G, parce qu'elle ne supporte pas les interfaces N1, N2 et N3 telles que définies dans les spécifications de la 3GPP pour la 5G. Cependant, il est important de noter qu'une mise en place d'une connexion entre la BNG et le réseau 5G reste possible, mais il va falloir dans ce cas là établir une connexion entre la BNG et la fonction d'interopérabilité fixe-mobile (*Fixed Mobile Interworking Function* en anglais : FMIF) comme montré dans la figure 3.4. Cette dernière supporte les interfaces N1, N2 et N3 et c'est elle qui s'occupe de la communication avec le 5G, tandis que la BNG se contente sur la communication avec la FN-RG pour l'authentification et la gestion des accès résidentiels [24].

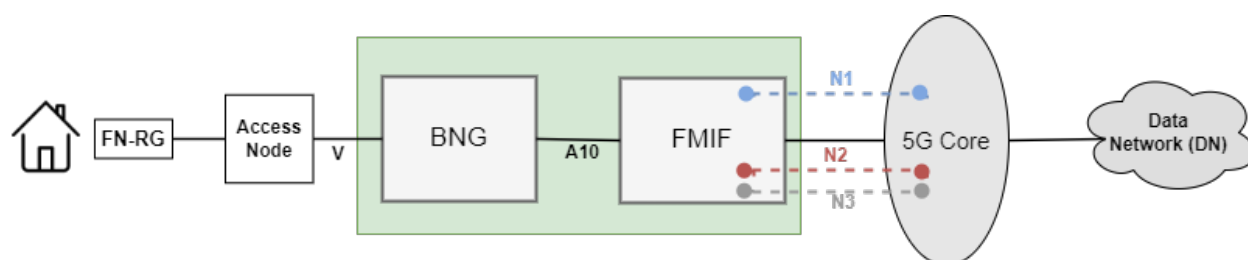


FIGURE 2.5 FMIF pour connecter la BNG au réseau 5G

2.4.4 AGF

En 2018, l'organisme de normalisation Broadband Forum (BBF) a décidé de s'attaquer à la convergence fixe-mobile (*Fixed-mobile convergence* : FMC) dans les réseaux 5G, avec la

possibilité que l'accès fixe sans fil 5G atténue ou soutienne le besoin de boucles locales. Pour y parvenir, le BBF collabore avec l'organisme 3GPP pour définir une fonction de passerelle d'accès (*Access Gateway Function* : AGF) qui se situe entre l'infrastructure d'accès filaire et le réseau principal sans fil.

Le rapport technique RT-470 expose les raisons de la FMC 5G [21], tandis que les exigences de l'AGF sont définies dans le document de travail TR-456 [25]. De plus, le 3GPP travaille sur la spécification technique pour la convergence sans fil-filaire 5G, appelée TS 23.316 V0.2.0 Services de groupe et aspects système Prise en charge de la convergence sans fil-fil pour le système 5G (5G WWC). La fonction de passerelle d'accès (AGF) est placée entre l'infrastructure d'accès filaire et le réseau de cœur sans fil pour offrir des services aux abonnés fixes à partir du même cœur 5G que celui qui prend en charge les utilisateurs mobiles. Cette configuration permet des économies de coûts et une meilleure efficacité opérationnelle.

Mode direct

Le second scénario de déploiement de l'accès filaire représente l'intégration en mode direct (5G-RG). La 5G-RG est connectée au réseau 5G via un réseau d'accès filaire se constituant d'une fonction de passerelle d'accès (AGF). Grâce à sa capacité de supporter les interfaces N1 et N2, l'AGF fait office d'intermédiaire entre le réseau d'accès filaire (aggrégé au niveau de la couche 2) et le réseau central 5G. Dans ce scénario, le réseau est dit intégré en mode direct à la faveur de la 5G-RG qui a la capacité de s'authentifier, de s'enregistrer et d'établir des signaux de type NAS (*Non-Access Stratum*) directement avec le réseau central 5G via l'interface N1 sans pour autant avoir besoin de passer par l'AGF.

Mode adaptatif

Le troisième mode d'intégration possible de l'accès filaire au réseau 5G est celui de l'intégration en mode adaptatif (FN-RG). Ce déploiement est similaire au deuxième scénario, il est aussi basé sur l'AGF qui, comme déjà mentionné, joue le rôle d'une station de connection entre le réseau d'accès et le réseau central 5G. Ce qui distingue ce mode du mode direct est que la FN-RG ne supporte pas l'interface N1. Dans ce cas là, c'est l'AGF qui émule l'interface N1, ainsi elle s'occupe de la génération et les échanges des signaux de signalisation de la couche Non-Access Stratum (NAS) avec l'AMF (*Access and mobility management function*) au nom de la FN-RG.

2.5 Environnement de développement P4

2.5.1 Le langage P4

P4, ou Programmation de processeurs de paquets indépendants du protocole, est un langage spécifique à un domaine pour configurer le plan de données d'un commutateur réseau. Bosshart *et al.* ont initialement introduit le langage en 2014 [6]. Une version révisée du langage a été présentée par Budiou et Dodd en 2016 [26]. La version mise à jour est appelée $P4_{16}$ tandis que l'originale est appelée $P4_{14}$ pour distinguer les deux versions [27].

P4 est un langage déclaratif de haut niveau qui peut décrire le traitement de paquets pour n'importe quel plan de données en utilisant des abstractions de réseau de haut niveau. Trois objectifs majeurs ont été pris en compte lors du développement du langage P4 : la reconfigurabilité, l'indépendance de protocole et l'indépendance de cible. La reconfigurabilité fait référence à la capacité d'un contrôleur à modifier le comportement de transfert de champ. Il appartient au codeur de spécifier le format de l'en-tête de paquet et les règles de traitement, ce qui est appelé l'indépendance de protocole. L'indépendance de cible fait simplement référence à la capacité d'un programme P4 à être compilé pour différentes cibles de commutateur sans nécessiter de connaissance préalable du matériel. Les abstractions P4, qui conçoivent une chaîne de traitement de plan de données conformément au modèle abstrait de la figure 2.2, comprennent des entêtes, des tables, des actions, etc. Le modèle abstrait pris en charge par P4 consiste en un parseur d'entêtes de paquet - en anglais *packet parser*, des tables de correspondance et d'action - en anglais *match action tables*, ainsi que le déparseur des entêtes du paquet - en anglais *packet deparser* [28]. Ces trois niveaux d'abstraction sont placés entre l'entrée (*ingress*) et la sortie (*egress*) du commutateur cible. Les en-têtes des paquets entrants sont analysés avant d'être envoyés aux ressources de table de correspondance, qui effectuent des recherches de champ d'en-tête et appliquent des actions aux en-têtes de paquet lorsqu'il y a une correspondance dans la table. Ensuite, le déparseur sérialise les métadonnées dans le paquet avant que celui-ci ne soit enfin envoyé à la sortie (*egress*).

Le flux de données dans le pipeline de traitement d'un code P4 est représenté dans la Figure 2.6. Les pipelines du P4 sont inspirés du concept de métadonnées de paquet de l'architecture PISA et sont composé de différents blocs qui ensemble offrent une interface uniforme. Les métadonnées de paquets, telles que les en-têtes, les métadonnées définies par l'utilisateur ou intrinsèques, sont utilisées pour transférer des informations entre ces blocs, permettant ainsi le bon ordonnancement et acheminement des paquets en cours de traitement.

Le parseur s'occupe de la division du paquet reçu en en-têtes distincts et en charge utile restante qui contient souvent le contenu de l'application envoyé, puis il fait l'extraction des

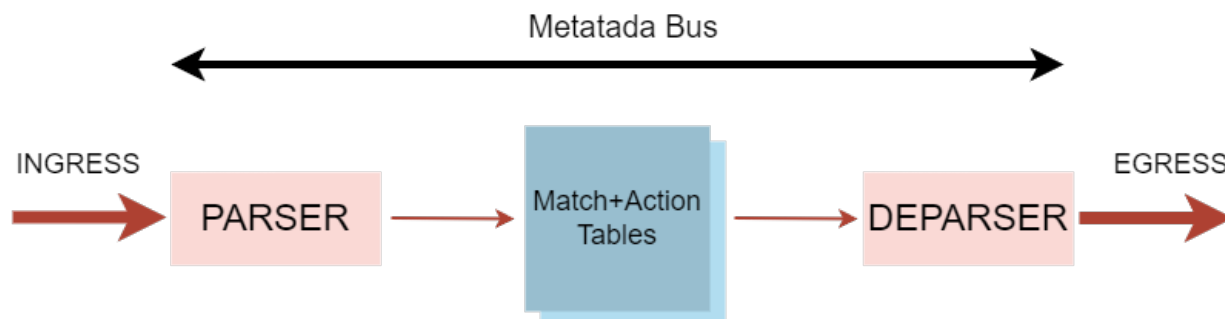


FIGURE 2.6 Le modèle de transfert abstrait du langage P4

champs contenus dans ces en-têtes selon la description et séquences effectués par le programmeur du code P4. par la suite, ces en-têtes et métadonnées sont acheminés au bloc des tables de correspondance qui consiste de plusieurs unités de [Clé-Action]. Au niveau de chaque unité se fait une comparaison d'une clé de recherche à une collections de paires Clé :Action et une action est exécutée en cas d'une comparaison positive. L'action exécutée peut être une addition de nouveaux champs, modification ou même une suppression de quelques autres, une action peut aussi générer de nouvelles métadonnées. C'est au niveau de ce module que se font les modifications apportées aux paquets traités. Force de mentionner que la charge utile ne subit aucun traitement au niveau du pipeline et est transférée séparément du reste du paquet. Après avoir traité le paquet au niveau du pipeline des tables de correspondance, il se transfère au bloc du réassemblage là où le deparseur reconstitue le paquet en émettant les en-têtes spécifiés suivis de la charge utile du paquet d'origine. La sortie du paquet est configurée avec des métadonnées intrinsèques qui incluent d'autres informations supplémentaires telles qu'un indicateur d'invalidation de paquets, le port de sortie souhaité, etc.

2.5.2 Programmation du plan de données

La programmation du plan de données fait référence à la capacité des utilisateurs, des administrateurs réseau ou des concepteurs d'équipement utilisant un ASIC de traitement de paquets à spécifier le plan de données et ses algorithmes. Bien que les algorithmes du plan de données aient été fréquemment exécutés sur des processeurs généraux au cours de la majeure partie de l'histoire de l'industrie des réseaux, la programmation du plan de données existait en réalité à cette époque. La programmation du plan de données n'est devenue un problème que lors de l'introduction de liaisons haut débit, qui dépassaient les limites de traitement des CPU et ont conduit au développement d'ASIC de traitement de paquets.

Les algorithmes utilisés dans le plan de données sont responsables du traitement de chaque

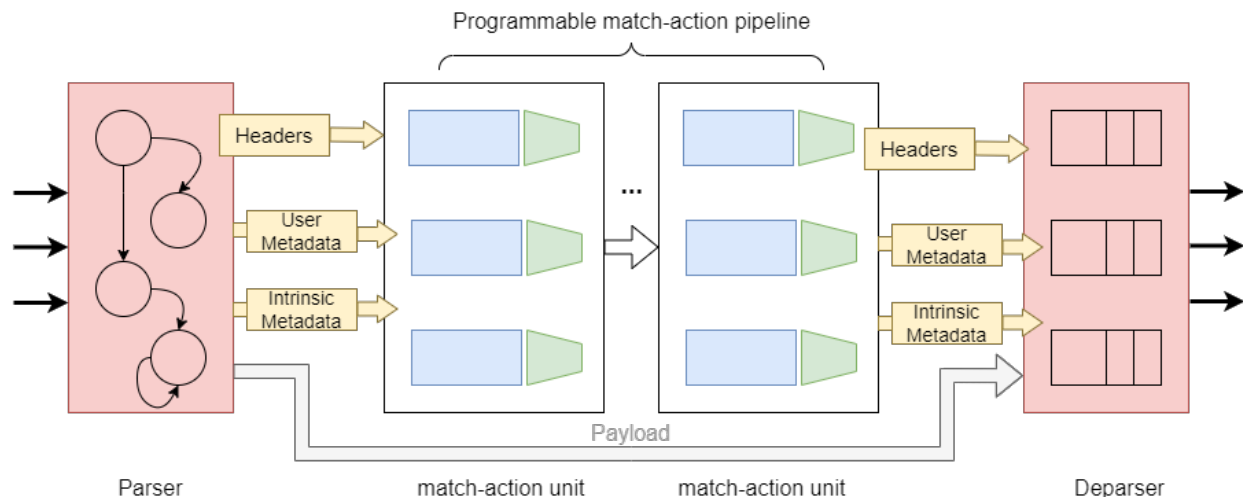


FIGURE 2.7 Information du flux de données [27]

paquet qui circule dans un réseau de communication. Par conséquent, ils déterminent en fin de compte comment ces systèmes fonctionnent, se comportent et évoluent. Habituellement, toute tentative d'intégrer la fonctionnalité du plan de données dans le plan de contrôle entraîne une baisse de performance considérable. La mise à disposition de la programmation du plan de données modifie de manière qualitative le pouvoir des utilisateurs. Ils sont maintenant capables de créer un équipement réseau unique sans pour autant sacrifier les performances, l'évolutivité, la vitesse ou la consommation d'énergie.

De nouveaux plans de contrôle et de nouvelles applications SDN peuvent être créés pour des réseaux uniques, et les utilisateurs peuvent eux-mêmes créer des algorithmes de plan de données spécialement adaptés à ces réseaux. La programmation du plan de données n'implique pas nécessairement la fourniture d'API utilisateur ou la prise en charge de plans de contrôle externes. Les fournisseurs de matériel pourraient toujours choisir de créer un plan de contrôle propriétaire et d'utiliser la programmation du plan de données exclusivement pour leurs propres avantages, sans nécessairement rendre leurs systèmes plus ouverts.

2.5.3 DPDK

Le Data Plane Development Kit (DPDK) est une collection de bibliothèques et d'outils logiciels qui facilite la création d'applications réseau sophistiquées [29]. Il est disponible gratuitement et est fourni sous la licence BSD. Le principal développeur de cette technologie est Intel, tandis que sa maintenance est supervisée par la fondation DPDK, composée d'un consortium de développeurs d'autres entreprises. Intel, ARM, Ericsson et la Fondation Linux

font partie des organisations qui apportent leur soutien à ce projet, qui compte de nombreux collaborateurs issus d'entreprises et d'universités.

La caractéristique principale du DPDK réside dans sa capacité à contourner la pile réseau du noyau Linux (Figure 2.8) [30], Ce qui permet à l'application construite dans l'espace utilisateur (*User Space*) d'établir une communication directe avec les équipements physiques de réseau. En conséquence, il offre un traitement amélioré et accéléré des paquets réseau provenant de ces applications. De plus, cette technologie permet également aux applications de tirer parti des fonctionnalités avancées des interfaces réseau contemporaines qui ne sont pas accessibles via les pilotes Linux traditionnels.

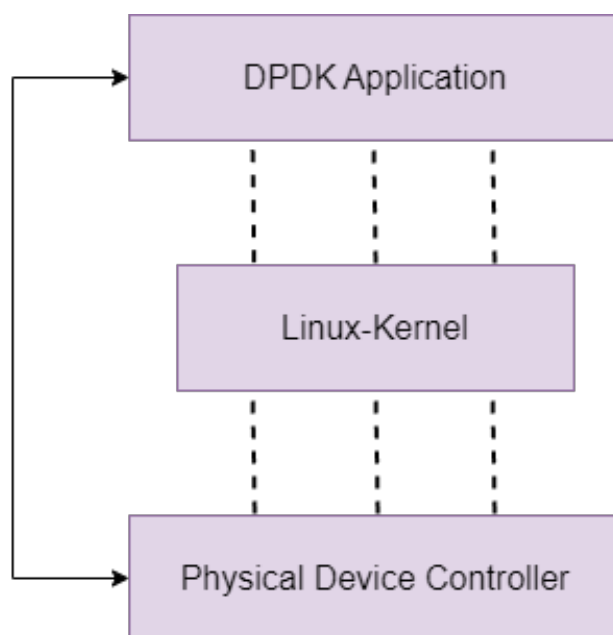


FIGURE 2.8 By-passing du noyau Linux par DPDK

L'utilisation de DPDK peut également servir à minimiser la surcharge supplémentaire imposée par la pile réseau du noyau Linux. Cela est réalisé en utilisant la technique d'attente active du CPU (*CPU Polling* en anglais) plutôt que les interruptions, ce qui entraîne une réduction des latences, notamment dans les environnements à haute vitesse.

De plus, le framework DPDK est utilisé par un ensemble de générateurs de trafic conçus pour générer un grand nombre de paquets tout en utilisant efficacement les ressources CPU. Plusieurs projets open source, tels que Moongen, Pktgen et TRex [31], exploitent cet outil. Plus précisément, TRex a été utilisé lors de la phase finale de mesure de ce travail de recherche, comme expliqué en détail dans le Chapitre 4.

Récemment, l'intégration du langage P4 avec DPDK a ouvert de nouvelles perspectives. Programmer DPDK en P4 offre l'avantage de rendre le plan de données plus flexible et programmable tout en maintenant des performances élevées, permettant ainsi aux développeurs de définir précisément le comportement souhaité de leurs applications réseau [32, 33].

T4P4S se présente comme un environnement de développement efficace pour cette intégration [34]. Il facilite le processus de compilation et de déploiement de programmes P4 sur DPDK, combinant ainsi la puissance de DPDK avec la flexibilité du langage P4. Grâce à T4P4S, les développeurs peuvent bénéficier d'un écosystème robuste pour explorer de nouvelles architectures de réseau tout en garantissant des performances optimales.

2.5.4 Les commutateurs programmables en P4

L'avènement du langage P4 a permis l'avancement de nouvelles technologies de commutateurs réseau avec une promesse significative. Les concepteurs de commutateurs réseau consacrent actuellement leurs efforts au développement de commutateurs avancés dotés de capacités améliorées. Cette orientation stratégique vise à leur permettre de tirer parti des avantages offerts par la technologie P4, optimisant ainsi leur rentabilité.

À ce jour, divers types de commutateurs programmables en P4 sont disponibles sur le marché. Certains d'entre eux sont des commutateurs logiciels spécialement conçus pour servir d'outils de test permettant d'évaluer la fonctionnalité des codes P4 dans des environnements de simulation. Ces commutateurs sont extrêmement efficaces et offrent des résultats rapides, ce qui en fait des dispositifs de choix pour les développeurs souhaitant évaluer les performances de leurs codes P4.

Parmi ces commutateurs logiciels, le BMv2 (Behavioral Model v2) [35] est particulièrement remarquable. Il est largement utilisé et recommandé par la communauté P4, en particulier pour ceux qui souhaitent se familiariser avec le langage P4 et débiter leur exploration dans ce domaine. Les commutateurs logiciels tels que le BMv2 se basent généralement sur l'architecture Portable Switch Architecture (PSA), laquelle est supportée par DPDK. Il convient de noter que DPDK prend en charge exactement deux architectures : PSA (Portable Switch Architecture) et PNA (Portable NIC Architecture). Dans notre implémentation, nous avons choisi d'utiliser l'architecture PNA pour atteindre nos objectifs.

De base, les commutateurs logiciels tels que BMv2 ne sont pas destinés à être utilisés dans des réseaux réels. Cependant, plusieurs travaux menés dernièrement ont montré la capacité de quelques commutateurs logiciels à être déployés dans un réseau réel. Le commutateur DPDK qui fait l'objet de ce travail en fait partie.

En revanche, divers commutateurs physiques ont été conçus pour être programmé en P4. On distingue deux type de commutateurs physiques [36] :

- Les commutateurs ASIC qui sont spécialement conçus pour accélérer le traitement du paquet afin d'offrir une latence faible et un débit très élevé [37], allant jusqu'à 3 Tbps pour le commutateur Tofino, qui est, à date, le commutateur ASIC le plus rapide au monde.
- Les commutateurs réseaux basés sur les FPGA. Ces derniers sont devenus des cibles d'immense importance pour le langage P4, leur capacité d'accélérer significativement les opérations de traitement et de routage de paquets a ouvert la voie à plusieurs travaux de recherche se concentrant principalement sur le déchargement des pipelines du plan de données des fonctions réseaux tels que la BNG par le biais des FPGA [38] [39].

CHAPITRE 3 MÉTHODE PROPOSÉE POUR LA MISE EN ŒUVRE DU PLAN DE DONNÉES DE LA FONCTION PASSERELLE D'ACCÈS AGF

3.1 Spécifications techniques de l'AGF

3.1.1 Modes de fonctionnement

Cette section fournit de plus amples informations sur les exigences fonctionnelles de l'AGF, que nous avons découvertes par le biais d'une analyse de la documentation existante dans le rapport technique TR-456, et sur la base de nos propres exigences et de recherches antérieures.

La Figure 3.1 montre que l'AGF peut être conçue en deux parties : une partie de plan de contrôle (AGF-CP) qui est responsable de l'authentification et la gestion des abonnés ainsi que toutes les informations de contrôle qui leur sont liées, et une partie de plan de données (AGF-UP) qui s'occupe du transfert des paquets du plan utilisateur de et vers le réseau de données.

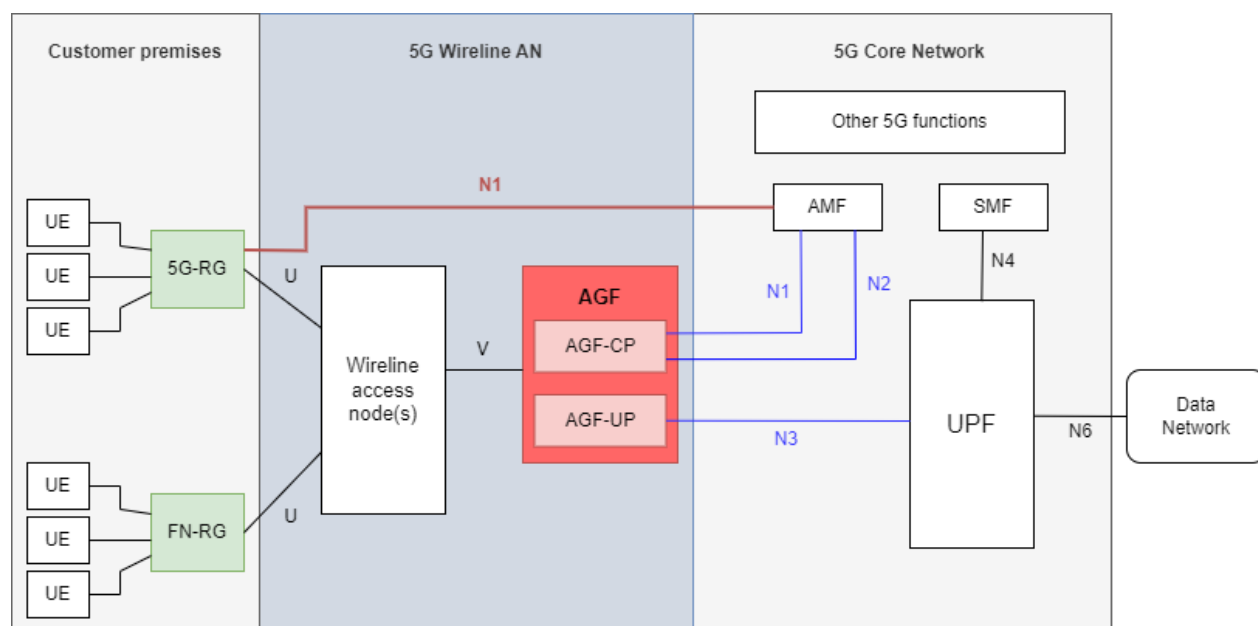


FIGURE 3.1 Vue architecturale de l'AGF reliant les RG au 5GC via des réseaux d'accès filaires

L'AGF peut être configurée pour fonctionner en deux modes : Le mode direct qui connecte l'AGF à la 5G-RG, et le mode adaptatif là où l'AGF est connectée à la FN-RG. L'opérateur de réseau peut décider d'opter pour une configuration directe seulement, adaptative seulement

ou pour une configuration qui fait fonctionner la 5G-RG et la FN-RG en parallèle. Les protocoles supportés pour chaque mode ne sont pas les mêmes que ce soit dans le plan de données ou le plan de contrôle.

Dans le mode adaptatif, L'AGF permet de connecter la FN-RG au réseau 5G. Cette communication se fait par le biais des protocoles traditionnels supportés par la FN-RG pour l'accès filaire au réseau de données. Les protocoles et les fonctionnalités de la FN-RG sont décrits dans le rapport technique TR-101.

Contrairement à la RG-5G, la FN-RG n'a pas été amélioré pour supporter les fonctionnalités de la 5G, seules les fonctionnalités décrites dans TR-101 sont prises en charge lors d'une connexion basée sur la FN-RG. Ces fonctionnalités ne supportent pas en particulier l'interface N1 pour la signalisation NAS, il n'y a pas de prise en charge de l'ATSSS (Service de sélection de la tranche d'accès), de la qualité de service réfléchie (en anglais : *Reflective QoS*) ou d'autres fonctionnalités similaires. Par conséquent, tous les avantages du cœur 5G ne sont pas disponibles dans le cas d'utilisation de la FN-RG.

Tout de même, l'AGF peut être conçue pour adresser quelques fonctionnalités que la FN-RG ne prend pas en charge, comme supporter l'interface N1, l'AGF s'occupera dans ce cas des échanges des signaux de signalisation Non-Access Stratum (NAS) avec la 5G au nom de la FN-RG. Cependant, quelques fonctionnalités liées au plan de données, voire la qualité de service réfléchie, ne peuvent pas être adressées par l'AGF, parce que même si l'AGF peut supporter de nouveaux protocoles qui prennent en charge la qualité de service réfléchie comme le protocole 5WE que l'on va voir par la suite, elle ne sera pas capable de les communiquer avec la FN-RG qui ne les supporte pas.

En revanche, la 5G-RG est plus globale par rapport à la FN-RG, elle est plus conforme et adaptée pour facilement se connecter au réseau 5G. Étant donné que la 5G-RG supporte l'interface N1, elle est capable d'établir une communication directe avec le cœur du réseau 5G pour le transfert des informations de contrôle, notamment les messages de signalisation NAS.

La 5G-RG peut aussi se connecter à la 5G par le biais de l'AGF, que ce soit pour le transfert des informations de contrôle ou bien les paquets de données. Le protocole utilisé pour la communication des informations de contrôle entre la 5G-RG est le PPPoE (*Point-to-Point Protocol over Ethernet*). Tandis que pour le transfert des paquets de données entre la 5G-RG et l'AGF, un nouveau protocole a été introduit qui est le 5WE (*5G Wireless Wireline Convergence User Plane Encapsulation*). Ce dernier est conçu spécifiquement pour permettre une convergence filaire dans le réseau 5G, il est le responsable de l'acheminement des paquets du plan de données entre la 5G-RG et l'AGF.

Dans notre travail, on s'est contenté sur le mode d'accès direct basé sur la 5G-RG, les différents protocoles et fonctionnalités requises pour ce mode de fonctionnement seront élaborés en détails dans les sections qui suivent.

3.1.2 AGF en mode direct : protocoles et fonctionnalités

Une mise en œuvre de l'AGF avec une séparation complète de son plan de contrôle du plan de données facilite la communication entre la 5G-RG et le réseau 5G. La partie du plan de données est responsable de l'acheminement à haut débit des paquets des différents abonnés qui sont connectés au réseau, Tandis que la partie de contrôle est généralement responsable de l'authentification et la gestion de ces abonnés.

Notre projet porte plutôt sur la conception et l'implémentation de la partie du plan de données (AGF-UP). Cependant, une petite description de la partie contrôle (AGF-CP) nous semble utile.

La partie de contrôle de l'AGF s'occupe du traitement et du transport de deux types de messages : les messages NAS qui met en œuvre une signalisation commun qui effectue la gestion du cycle de vie pour toutes les sessions du plan utilisateur, et les messages AS pour l'échange des différentes informations liées à l'établissement des sessions, la configuration d'abonnement d'accès et de la configuration des session PDU. la figure 3.2 donne une description des messages AS et leur contenu qui contient l'identifiant des sessions, le nombre des identifiants QFI (en ang : *QoS flow identifier*) que nous allons voir en détails dans les prochaines sections, ainsi que d'autres informations liées à la gestion de la bande passante qui sont communiquées dans d'autres messages AS.

Généralement, La partie de contrôle n'affecte la performance globale du système qu'en moment de l'établissement des sessions d'abonnés. En mode d'opération normale, le système ne requiert aucune interaction avec le plan de contrôle, à l'exception de la partie concernant le peuplement des tables et leurs actions pour qu'elles soient utilisées dans le plan de données.

De ce fait, comme notre prototype est conçu pour fonctionner en mode normale, nous nous concentrons sur la partie de traitement de paquets à haut débit, celle de plan de données ; seulement la partie du peuplement des tables sera abordée pour la partie du contrôle.

La figure 3.3 décrit la pile de protocoles utilisée par l'AGF pour le transfert des paquets de plan de données entre le 5G-RG et l'UPF. Le transfert de paquets entre la 5G-RG et l'AGF est basé sur la communication point à point qui se fait par le biais du protocole PPPoE version 2, appelé 5WE, et des VLAN. Tandis que le transfert de paquets entre l'AGF et l'UPF sur l'interface N3 est assuré en utilisant la technique du tunneling standard basée sur

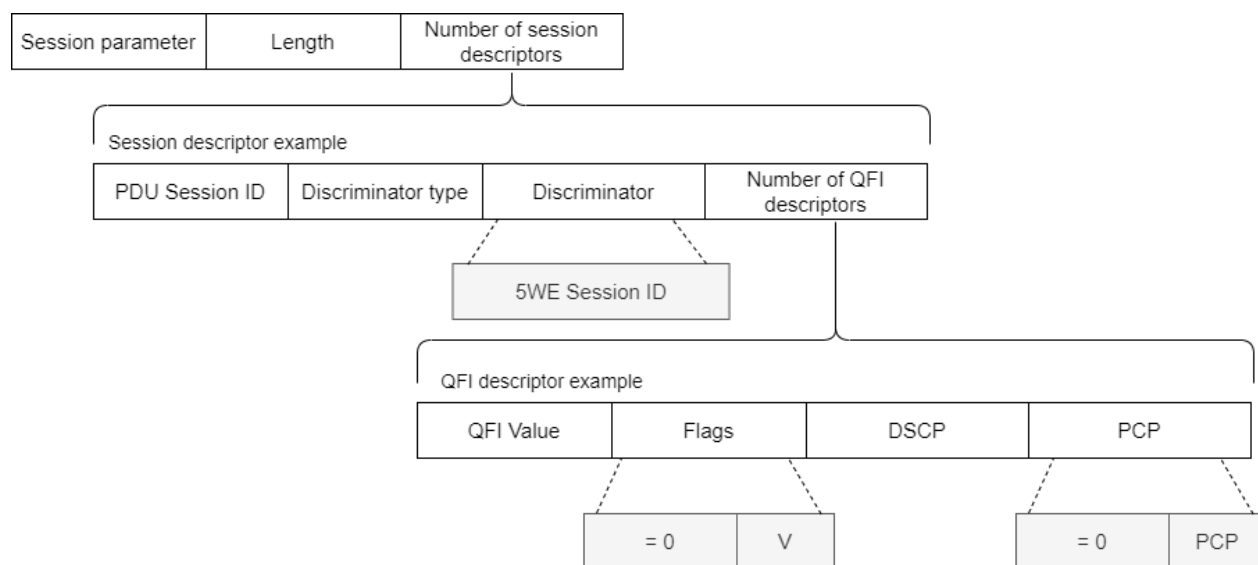


FIGURE 3.2 Les informations de contrôle type AS [25]

l'encapsulation de la charge utile "Payload" dans des en-tête GTP-U [40] [41] .

Ces protocoles sont élaborés en détails dans les sections qui suivent.

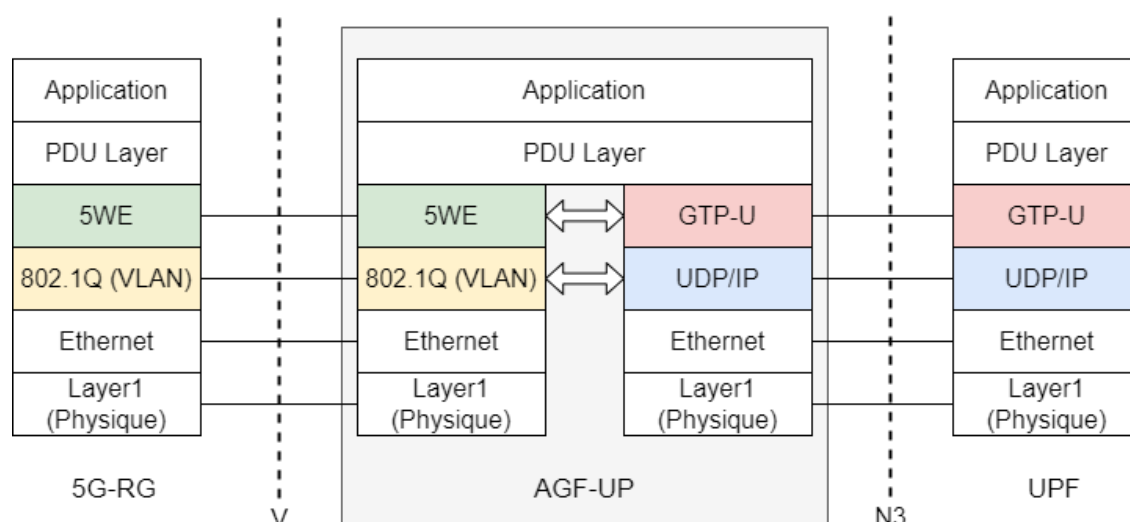


FIGURE 3.3 Les piles de protocoles de communication entre le 5G-RG et l'UPF par le biais de l'AGF

Les lignes d'accès

Un élément essentiel du concept AGF sont les lignes d'accès qui permettent d'établir la connexion entre les 5G-RG et le cœur du réseau par le biais de différentes technologies tels que les réseaux DSL, fibre optique, câble et mobiles.

En effet, les lignes d'accès servent de points d'entrée dans le réseau et sont responsables de fournir aux utilisateurs finaux un accès aux réseaux de données tels que Internet et autres services ainsi de gérer la quantité de bande passante qui peut être accordée à un abonné. Parmi les protocoles les plus couramment utilisés par les lignes d'accès, on trouve le protocole PPPoE qui est utilisé pour encapsuler les trames PPP (*Point-to-Point Protocol* en anglais) dans des trames Ethernet afin d'établir une connexion point-à-point entre l'équipement utilisateur (User Equipment) et le nœud d'accès via un réseau Ethernet.

le PPPoE se compose de deux phases distinctes : la phase de découverte et la phase de session PPP. Lorsqu'un hôte d'une certaine 5G-RG souhaite initier une session PPPoE, il doit d'abord effectuer une découverte pour identifier l'adresse MAC Ethernet de la paire (AGF dans notre cas) et établir une SESSION_ID PPPoE. Cette phase de découverte se fait sur 4 étapes [42] que l'on va projeter sur notre application AGF pour expliquer leur fonctionnement :

- PADI (*The PPPoE Active Discovery Initiation*) : Dans notre cas, la 5G-RG envoie un paquet de diffusion avec une étiquette qui contient le nom de service demandé, l'étiquette sera "5G-Service-Name" si la 5G-RG sollicite un accès au réseau 5G.
- PADO (*The PPPoE Active Discovery Offer*) : Lorsque les noeuds reçoivent le "PADI", celui qui offre ce service répondra par un message PADO. Dans le cas d'un service 5G, c'est l'AGF qui va répondre en envoyant un PADO à la 5G-RG.
- PADR (*The PPPoE Active Discovery Request*) : lorsque l'hôte qui demande accès au service (la 5G-RG dans notre cas) reçoit le PADO, il va d'abord garder l'Adresse MAC du serveur (AGF dans notre) dans sa table MAC pour les prochaines utilisations. Puis, il va répondre par un "PADR" pour demander l'accès au noeud qui offre ce service (AGF dans notre cas).
- PADS (*The PPPoE Active Discovery Session-Confirmation*) : lorsque l'AGF recevra le PADR, il va initialiser une nouvelle session PPP et générer un identifiant de session unique (Session_ID) qui sera utilisé dans la communication PPPoE entre la 5G-RG et l'AGF.

Dans notre projet, on s'est servi de ces paquets d'établissement de sessions PPP pour tester la fonctionnalités du plan de données de l'AGF lorsqu'il reçoit des paquets qui sont destinés à être acheminer au plan de contrôle.

Un nouveau protocole : L'encapsulation 5WE

L'encapsulation 5WE correspond à l'encapsulation du plan utilisateur de la convergence 5G sans fil-filaire (*5G Wireless Wireline Convergence User Plane Encapsulation* en anglais), c'est un nouveau mécanisme utilisé dans les réseaux 5G pour encapsuler et transporter les

paquets de données de l'utilisateur entre la passerelle résidentielle(RG) et le réseau central 5G. Il est conçu pour permettre un accès filaire au réseau 5G, offrant une expérience utilisateur fluide sur différents types de technologies réseau. L'encapsulation 5WE permet un transport efficace des paquets de données de l'utilisateur sur une variété de types de réseau, y compris Ethernet, Wi-Fi et réseaux cellulaires, tout en assurant des performances élevées et une faible latence.

En effet, cette encapsulation représente une version modifiée du protocole PPPoE, spécialement adaptée pour répondre aux besoins du réseau 5G. Elle offre également une communication point à point et est conçue pour être utilisée dans des environnements où la transmission point à point sur Ethernet (PPPoE) est déjà déployée. Par conséquent, les implémentations doivent être en mesure de vérifier le numéro de version pour distinguer la nature du trame. Une version 1 indique que le paquet est de type PPPoE, tandis qu'une version 2 indique que le paquet est de type 5WE.

La charge utile d'Ethernet (*Ethernet Payload*) pour le protocole PPPoE est identifiée par une valeur d'Ethertype de 0x8864, Les informations du paquet qui suivent cet Ethertype utilisent une valeur de 2 dans le champ VER afin de réutiliser le PPPoE comme encapsulation 5WE. L'encapsulation 5WE est structurée comme montré dans la Figure 3.4 [43].

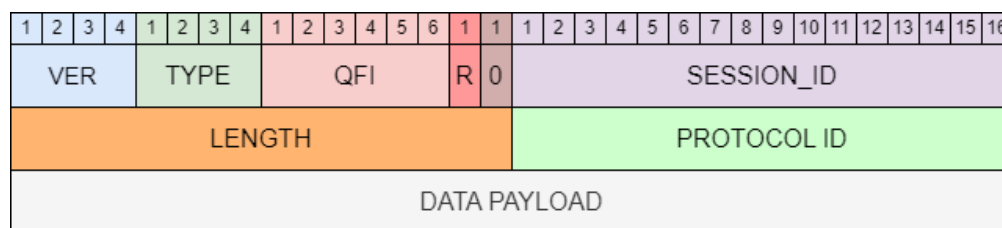


FIGURE 3.4 La structure de l'en-tête 5WE

Le champ "Ver" réfère à la version du PPPoE, il doit être mis à 0x02 pour indiquer qu'il s'agit bien d'une encapsulation 5WE. Le champ "Type" est fixé à 1.

Le champ "QFI" est un champ de 6 bits qui joue un rôle crucial dans l'établissement des politiques de qualité de service dans le réseau 5G. Il est responsable de l'encodage des identifiants des flux de qualité de service de la 5G, également appelés *5G QoS Flow Identifier* en anglais. La 5G-RG et le cœur du réseau 5G s'appuient sur ce champ pour identifier les valeurs correspondantes de la PCP (Priorité de la Classe de Service) et du DSCP (Code Point de Différenciation de Service) associées.

Le bit "R" sert d'indicateur pour la qualité de service réfléchi. Lorsqu'il est défini à 1, la 5G-RG n'a pas besoin d'établir de nouvelles règles de qualité de service pour les paquets

qu'elle va construire. Elle se contente de copier les mêmes règles de qualité de service qui ont été appliquées aux trames reçus en provenance de la 5G (Downstream forwarding). Ce champ est utile que pour les paquets de downstream et est ignoré pour les paquets Upstream.

Le champ "Session_ID" est un entier non signé de 16 bits. Il sert à différencier les multiples sessions PDU des abonnés qui se trouvent dans le VLAN. La valeur 0xffff est réservée pour une utilisation future et ne doit en aucun cas être utilisée.

La valeur du champ "Length", qui est codée sur 16 bits, indique en octets la taille de la charge utile de données (Payload), y compris l'identifiant initial de protocole (Protocol ID).

le "Protocol ID" indique le type de la charge utile qui est encapsulée et transférée à l'intérieur de l'en-tête 5WE. A date, seulement trois valeurs sont valides :

- 0x0021 : pour encapsuler une charge qui se constitue de paquets IPv4
- 0x0031 : pour encapsuler l'Ethernet dans le 5WE
- 0x0057 : pour encapsuler des paquets IPv6 dans le 5WE

Le Tunneling : Le Protocole GTP-U

Le protocole GTP-U (*GPRS Tunneling Protocol for User Plane*) est utilisé dans les réseaux mobiles, que ce soit dans les réseaux initiaux via la 3G et la 4G, ou dans le réseau 5G où il joue un rôle vital dans le transport des paquets entre les nœuds d'accès et le cœur 5G, notamment l'UPF.

Le principal objectif du protocole GTP-U est de fournir un mécanisme de tunnel permettant le transport de données utilisateur entre différents nœuds au sein du réseau. Il est responsable de l'encapsulation des Unités de Données de Protocole (PDU) transportées entre l'Équipement Utilisateur (UE) et d'autres nœuds en utilisant la pile de protocoles IP/UDP. Chaque nœud d'extrémité se voit attribuer une valeur d'Identifiant d'Extrémité de Tunnel (*Tunnel Endpoint Identifier* en anglais :TEID) pour indiquer l'association d'une PDU spécifique avec un tunnel particulier. Ces valeurs ne sont généralement pas symétriques entre les deux extrémités, car chaque nœud alloue individuellement sa propre valeur de TEID.

Le protocole GTP-U est toujours encapsulé dans l'UDP, et l'IANA [44] a réservé le numéro de port 2152 en tant que port de destination UDP pour le GTP-U. L'allocation du port source UDP dépend du nœud d'extrémité émetteur.

La figure 3.5 présente la structure de l'en-tête GTP-U, qui comprend plusieurs champs essentiels pour garantir à la fois le bon acheminement des paquets et l'identification du contenu de la charge encapsulée dans cet en-tête.

Le champ "Ver" désigne la version de GTP-U utilisée, qui peut être soit la version 0 soit la version 1. La version 1 est spécifiquement utilisée dans la 5G en raison de ses extensions,

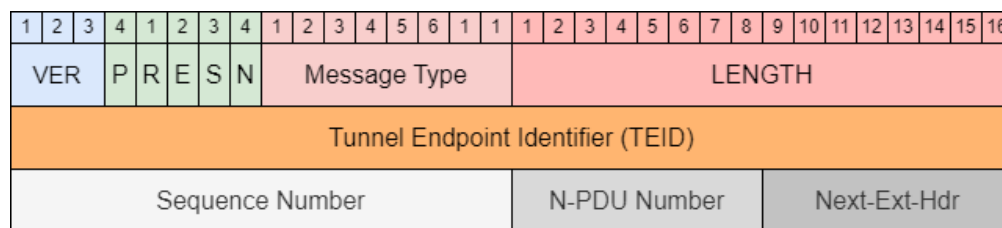


FIGURE 3.5 Le format de l'en-tête GTP-U

qui incluent des champs importants tels que le QFI pour le traitement des plans utilisateurs dans le réseau 5G.

Trois indicateurs, à savoir l'indicateur "E", l'indicateur "S" et l'indicateur "N", sont utilisés pour indiquer l'existence de champs optionnels. Une extension de l'en-tête GTP-U (*GTP-U Extension header*) existe si l'indicateur "E" est mis à 1. La mise des champs "S" et "N" à 1 impliquera l'existence d'un numéro N-PDU et du numéro de séquence qui indique le numéro de séquence croissant des sessions PDU transportées via les tunnels GTP-U.

le champ "TEID" est le champ le plus important dans l'en-tête GTP-U, il est décrit sur 32 bits et permet d'identifier de manière unique un tunnel dans le réseau. Lorsqu'un paquet est encapsulé dans un tunnel GTP-U, le champ "TEID" est ajouté à l'en-tête pour indiquer le tunnel de destination. Une fois le paquet reçu à l'extrémité de réception du tunnel, le champ TEID est utilisé pour déterminer à quel tunnel il appartient et le paquet est ensuite décapsulé et transféré vers la destination appropriée.

le champ "Next-Ext-Hdr" indique l'en-tête d'extension suivante si elle existe. Dans notre travail, nous avons utilisé l'extension "PDU Session Container" où la valeur du champ "Next-Ext-Hdr" est définie comme 10000101 [45]. Cette extension spécifique, "PDU Session Container", est particulièrement pertinente pour la 5G, car elle contient le champ "QFI" utilisé pour garantir que les différentes sessions PDU bénéficient des politiques de qualité de service appropriées dans le réseau 5G.

3.2 Conception du prototype de l'AGF

3.2.1 Structure du pipeline de l'AGF

Lors de l'établissement d'une nouvelle session, la 5G-RG reçoit des règles de qualité de service (QoS) ainsi que les descripteurs de classe de trafic correspondants, tels que ceux illustrés dans la figure 3.2. Ces descripteurs incluent les flux de qualité de service (QFIs) qui sont attribués à la session, ainsi que le Session-AMBR (bit rate maximum agrégé) qui gère la quantité de

bande passante allouée à chaque flux de données de la session. Cela permet de contrôler la qualité de service pour chaque flux de trafic dans le réseau 5G, garantissant ainsi une utilisation efficace et équitable des ressources réseau pour chaque abonné.

Toutes ces informations seront communiquées par la suite à l'AGF qui doit maintenir un contexte pour chaque session PDU, ce contexte contient l'identifiant de la ligne (Line ID), l'identifiant de la session PDU, les adresses MAC de la 5G-RG et des UPFs qui lui sont connectées, ainsi que la correspondance des valeurs des QFIs avec les valeurs des champs PCP pour l'Ethernet et celles des DSCP pour les adresses IP afin de permettre la classification et la gestion des paquets. Aussi l'AGF doit garder un contexte pour la correspondance des valeurs de TEID avec la valeur de chaque session PDU en vue de pouvoir identifier les paquets en provenance de l'UPF et les acheminer vers la 5G-RG qui leur correspond.

Après avoir établi la nouvelle connection avec la 5G-RG et l'UPF, et gardé dans son plan de contrôle un contexte sur toutes les sessions PDU actives, l'AGF passe au traitement des trafics de plan de données dépendamment de leur provenance et de leur priorité. La figure 3.6 montre un exemple de type de trafics que peut recevoir l'AGF que ce soit de la part de la 5G-RG (transfert dans la direction Upstream) ou bien de la part de l'UPF (transfert dans la direction Downstream). On distingue deux type de flux de qualité de service : Les flux de qualité de service à débit garanti (*Guaranteed Bit rate Qos Flow* en anglais : GBR) qui constituent les trafic pour les services de haute priorité (Jeux en ligne par exemple), et il y'a ainsi le flux de qualité de service à débit non garanti (Non-GBR) qui sont des trafics de faible priorité.

L'AGF doit pouvoir identifier, classifier et traiter tous les types de trafics qu'elle reçoit des différentes sessions PDU actives en se basant sur le type des flux de données que transportent ces trafics. Pour ce faire, un pipeline pour décrire les fonctionnalités de l'AGF et répondre à ses exigences a été conçu. ce pipeline est composé de trois parties : une partie pour la transmission des paquets Upstream (*Upstream forwarding* en anglais), une partie pour la transmission des paquets Downstream (*Downstream forwarding* en anglais), et une dernière partie pour le traitement et l'acheminement du paquets de contrôle. On va élaborer les fonctionnalités de chacune de ces parties dans ce qui suit.

Le pipeline de transmission Upstream

Le pipeline de transmission Upstream, montré dans la figure 3.7, décrit le chemin emprunté par une requête pour être acheminée de l'équipement utilisateur (User Equipment) qui l'a créée vers le serveur qui fournit ce service. Pour le fonctionnement normal, l'équipement utilisateur construit un paquet qui représente la charge utile (*Payload*). Cette charge peut

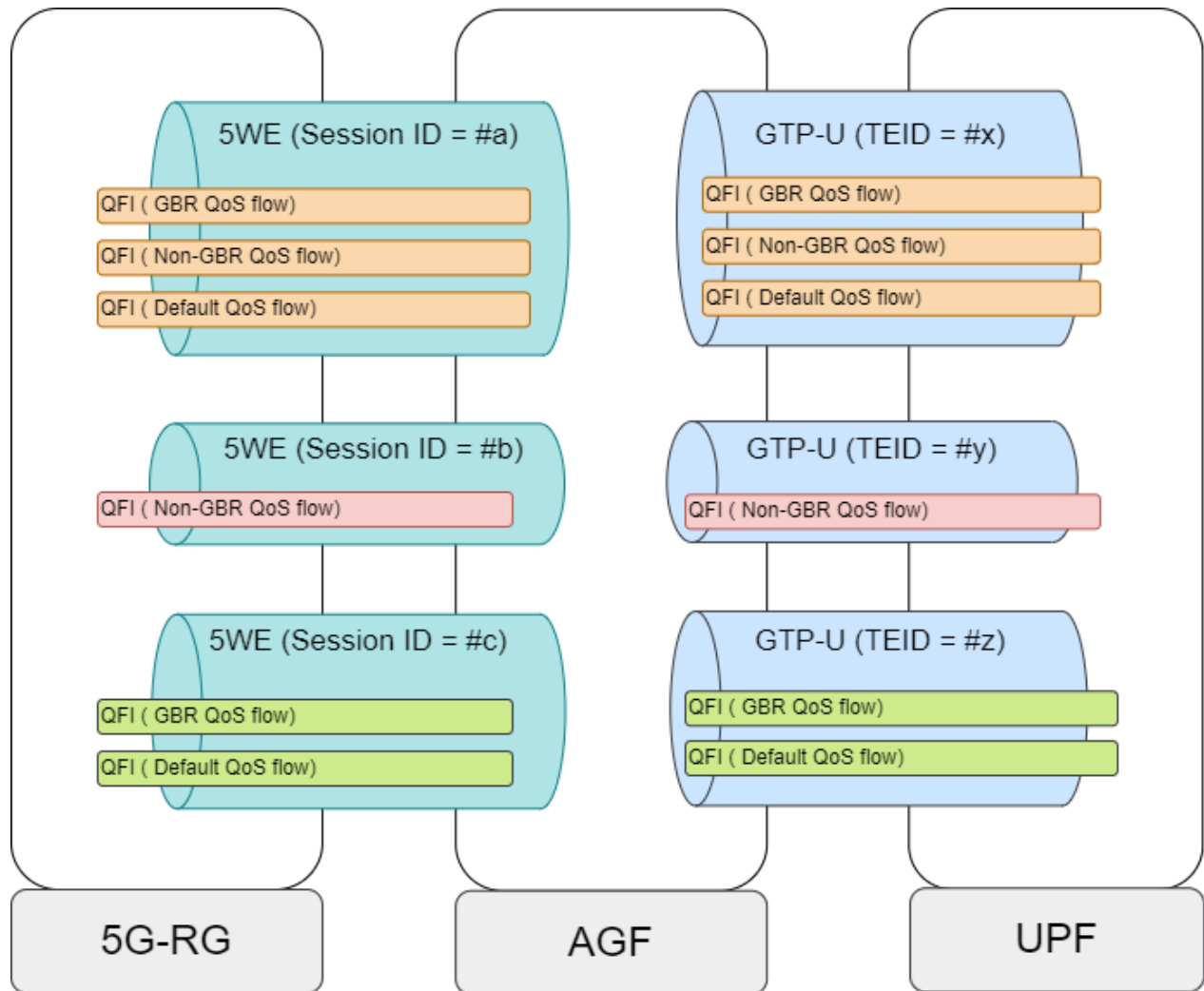


FIGURE 3.6 Exemple de flux de données

être soit "Ethernet", "IPv4" ou "IPv6" appelés (*Inner Ethernet, Inner IPv4, Inner IPv6* en anglais) afin de les distinguer des en-têtes Ethernet et IP extérieures qui servent pour l'encapsulation de cette charge. En cas d'Ethernet, cette charge doit contenir l'adresse MAC source de l'utilisateur ainsi que l'adresse MAC du serveur du réseau de données destination (exemple : Internet). Dans le cas d'une communication IP, l'adresse IP source sera celle de l'équipement utilisateur et l'adresse IP destination sera celle du serveur destination.

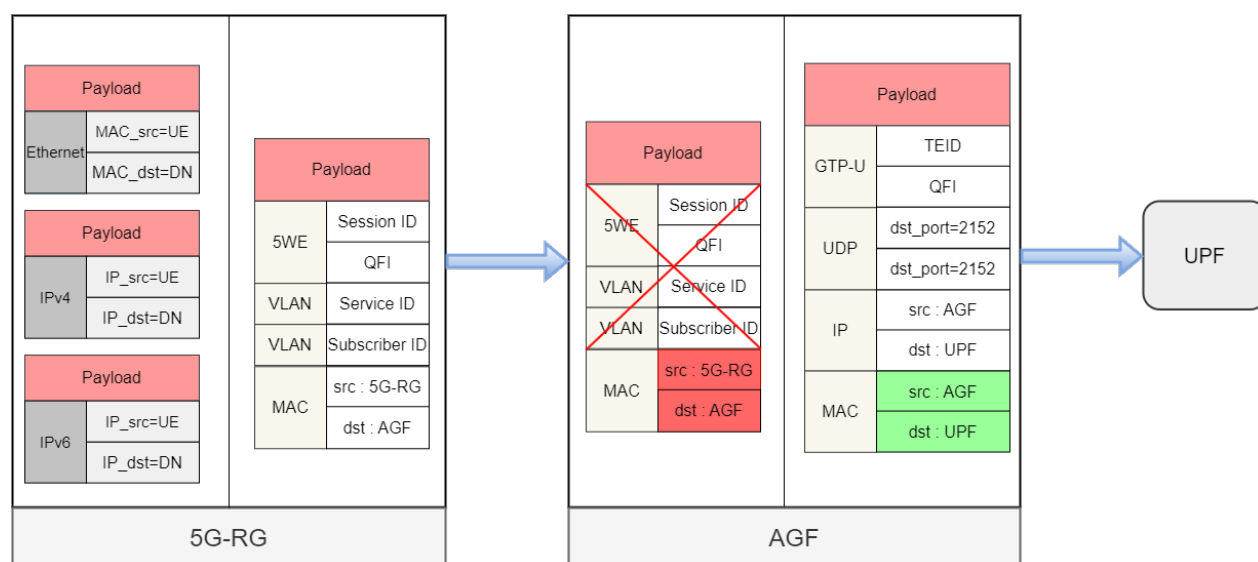


FIGURE 3.7 Fonctionnalités du pipeline Upstream de l'AGF

La charge est ensuite envoyée à la passerelle 5G-RG qui va l'encapsuler dans un paquet composé des en-têtes Ethernet, VLAN et 5WE. Dépendamment de quel utilisateur la 5G-RG a reçu le paquet, elle identifie la session "Session ID" et puis marque le "QFI" dans l'en-tête 5WE. La 5G-RG assigne les deux valeurs des VLANs "Service ID" et "Subscriber ID" tout en marquant le champ "pcp" qui correspond au "QFI". Par la suite, elle crée l'en-tête Ethernet en indiquant son adresse MAC ainsi que celle de la destination (AGF).

L'AGF peut se servir de plusieurs paramètres afin de vérifier la validité d'un paquet en provenance de la 5G-RG, voire l'identifiant de session de la 5WE, l'identifiant de VLAN, l'adresse MAC ou bien le port où le paquet a été reçu. Dans notre prototype, nous avons décidé de s'appuyer sur l'identifiant de session "Session ID" ainsi que les identifiants VLAN afin de permettre à l'AGF de vérifier si un paquet est valide ou pas. Notre prototype permet également de rejeter tout paquet reçu qui n'a pas un identifiant de protocole "Protocol ID" valide pour la session PDU (voir section 3.1.2).

Lorsque le paquet est valide, l'AGF se sert de la valeur de la "session ID" de l'en-tête 5WE

pour en identifier la session PDU "*PDU Session ID*" en cours de traitement, qui est stockée dans le plan de contrôle de l'AGF. Cette liaison permet à l'AGF d'obtenir toutes les valeurs nécessaires pour remplir les champs du paquet GTP-U qu'elle va devoir construire afin de transférer la charge utile encapsulée vers l'UPF approprié.

Ensuite, l'AGF entreprend la construction du paquet GTP-U. Elle commence par créer l'en-tête GTP-U ainsi que son extension en incluant les valeurs appropriées pour les champs "QFI" et "TEID" pour le tunneling. Ensuite, elle établit l'en-tête UDP avec le numéro de port de destination 2152. Le numéro de port UDP source n'a pas d'importance, et dans notre prototype, nous avons également décidé de le fixer à 2152. Par la suite, l'AGF génère un en-tête IP qui contient les adresses source (celle de l'AGF) et destination (l'adresse IP de l'UPF destinataire). Enfin, l'AGF modifie les adresses MAC de l'en-tête Ethernet en remplaçant celles reçues de la 5G-RG par celles de l'AGF et de l'UPF. Une fois le paquet ainsi construit, il est envoyé à l'UPF pour traitement ultérieur.

Le pipeline de transmission Downstream

Le pipeline de transmission Downstream, quant à elle, décrit le chemin que prend un paquet en provenance d'un réseau de données (DN) vers une passerelle résidentielle (RG).

La figure 3.8 donne une description des fonctionnalités de la fonction AGF en cas d'un traitement Downstream. Lorsque l'AGF reçoit un paquet en provenance de l'UPF, elle vérifie si le paquet était bien destiné à elle en vérifiant l'adresse IP destination. Puis, elle procède à l'identification de la session PDU qui est sujet de ce traitement en se basant sur la valeur du "TEID" ainsi que l'adresse IP source (celle de l'UPF).

Ensuite, l'AGF se servira de la valeur de la session PDU pour déterminer l'identifiant de session ("Session ID") à insérer dans l'en-tête 5WE, ainsi que les identifiants VLAN (VLAN ID) pour le service et l'abonné. Pour que la politique de qualité de service puisse s'appliquer sur le paquet lors de son passage du VLAN du réseau d'accès, l'AGF doit le classifier. Pour ce faire, elle se base sur la valeur du "QFI" reçue dans l'en-tête GTP-U pour en identifier la valeur des champs "PCP".

Une fois que l'AGF a collecté toutes les informations nécessaires pour construire le paquet 5WE à envoyer à la 5G-RG, elle procède à la création des en-têtes 5WE et VLAN en remplissant leurs champs avec les valeurs appropriées. Ensuite, elle modifie les adresses MAC de l'en-tête Ethernet pour les adapter aux destinataires, puis envoie le paquet à la 5G-RG de destination.

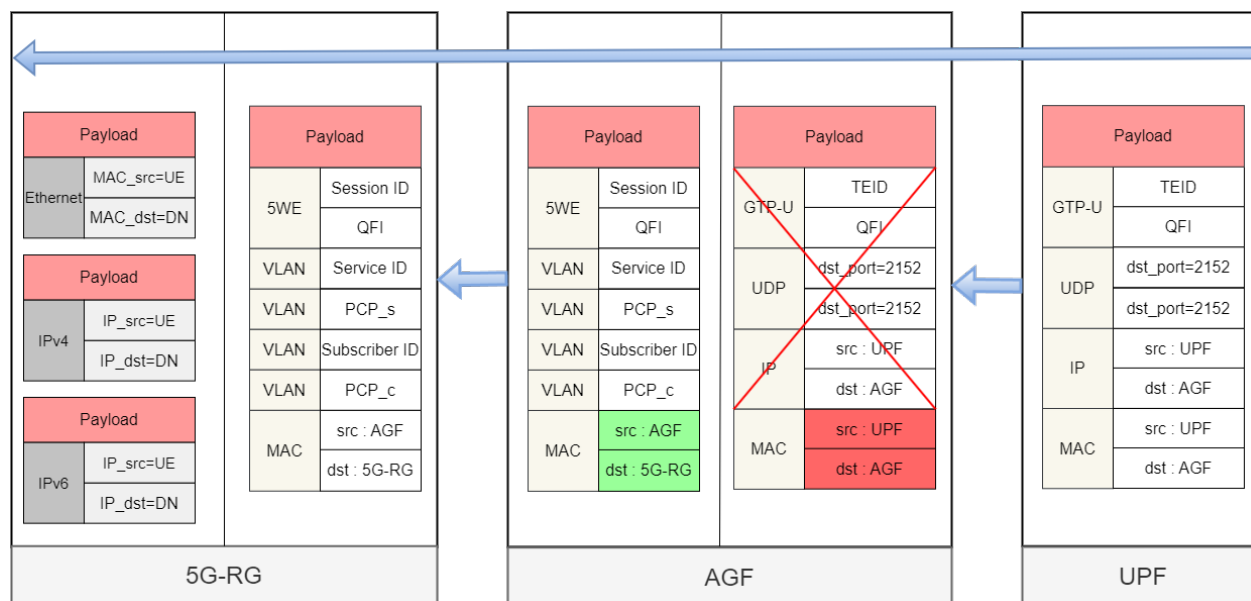


FIGURE 3.8 Fonctionnalités du pipeline Downstream de l'AGF

Le pipeline de contrôle

Le pipeline de contrôle que nous avons conçu dans notre prototype AGF ne prend pas en charge le traitement des paquets de contrôle. Ceci signifie qu'aucune modification n'est apportée aux champs des paquets de contrôle dans ce pipeline.

En effet, le pipeline conçu se contente seulement de distinguer les paquets du plan de données du plan de contrôle en provenance de la 5G-RG une fois qu'ils sont reçus par l'AGF. Cela est réalisé en se basant sur la valeur du champ *Version* du PPPoE ou du 5WE en cas des paquets de plan de données. De plus, le pipeline effectue la vérification de la validité des paquets PPPoE de contrôle en se basant sur la valeur du champ *Code* des paquets PPPoE.

Le pipeline de contrôle permet également d'acheminer les paquets qui sont en provenance du plan de contrôle de l'AGF vers le port de sortie de la 5G-RG destination.

3.3 Implémentation de pipeline basée sur P4

Dans cette section, nous présentons notre implémentation P4 pour le plan de données d'une fonction de passerelle d'accès (AGF), en se basant sur les exigences décrites dans les sections 3.3 et 3.4. Notre implémentation se compose de trois composants principaux : (1) une mise en œuvre du parseur, détaillée dans la section 3.5.1, qui permet la séparation des en-têtes des paquets, (2) des pipelines d'actions de correspondance programmables en P4, et (3) un

déparseur qui permet de reconstituer les en-têtes des paquets restants.

La figure 3.9 offre une vue d'ensemble complète du prototype AGF modélisé en P4. Comme discuté dans la section précédente, les pipelines de notre prototype subissent des variations significatives lorsqu'il s'agit de traiter le trafic upstream, downstream et la partie de contrôle dans un AGF. Par conséquent, le diagramme de flux représenté peut être distinctement divisé en trois parties : un pipeline upstream qui est représenté en bleu, un pipeline de traitement downstream en vert, et un pipeline des paquets de contrôle représenté en rouge.

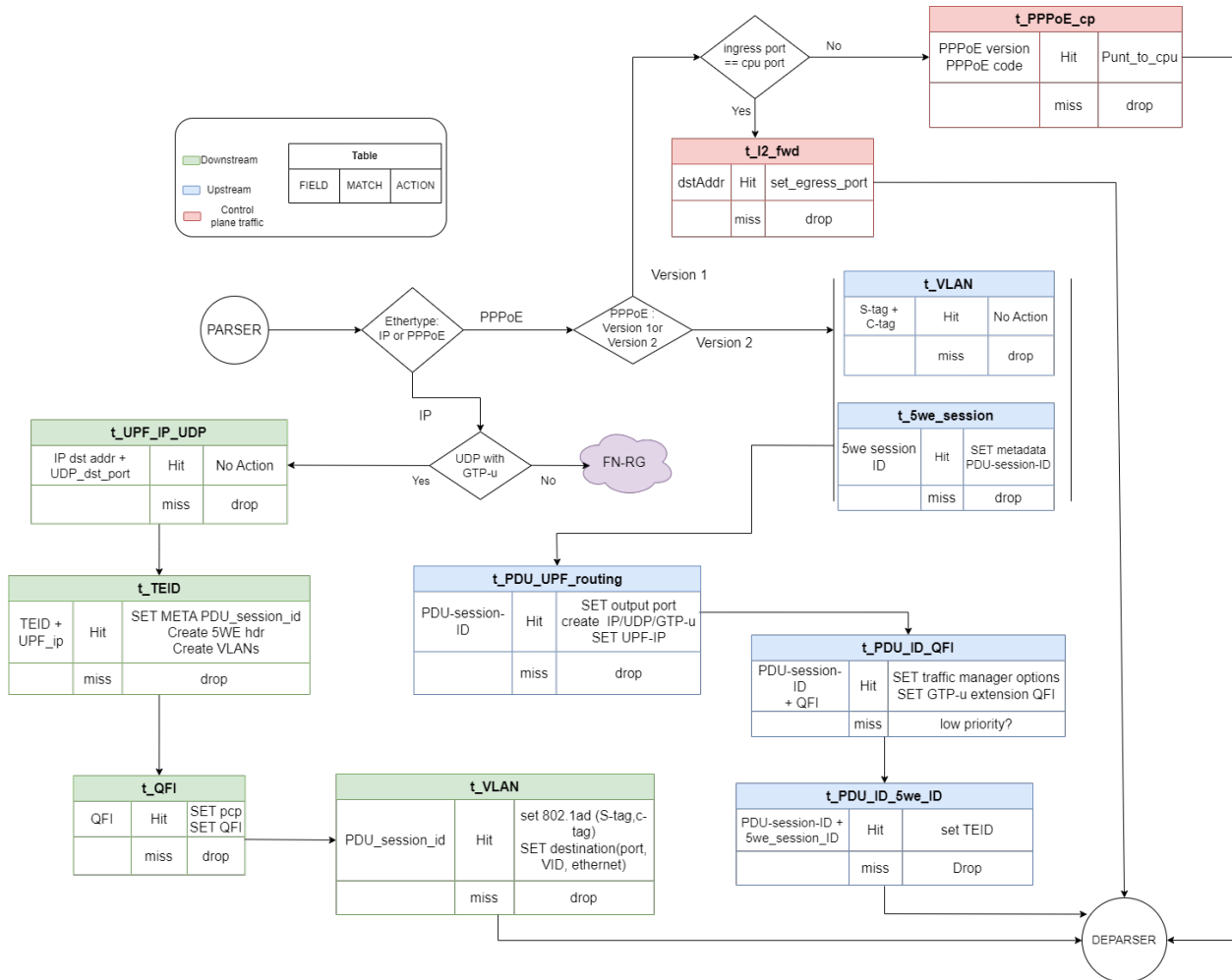


FIGURE 3.9 La vue d'ensemble complète du prototype AGF-P4

3.3.1 Le parseur d'entêtes

Après avoir pénétré dans le P4-AGF, les paquets passent par une étape initiale appelée le parser d'entrée (*ingress parser* en anglais). Pendant cette étape, le flux entrant de bits subit un

processus d'analyse pour le convertir en une représentation de données interne significative. Cette représentation est appelée le vecteur d'en-tête du paquet. Le vecteur d'en-tête du paquet contient des informations essentielles sur le paquet et peut être facilement accessible et utilisé dans les étapes ultérieures du pipeline d'actions de correspondance P4.

la figure 3.10 représente le flux de contrôle du parser de l'AGF qui est décrit comme une machine à états. Le point de départ est généralement l'état "start" (début), qui représente le point initial du processus d'analyse, puis l'extraction du premier en-tête ait lieu. Dans notre application AGF, l'état qui suit l'état "Start" est toujours le "Parse_Ethernet" qui s'occupe de l'extraction de l'en-tête Ethernet,. Cette dernière représente la premiere en-tête des paquets du trafic, que ce soit pour les trafics du downstream ou upstream.

En fonction de la valeur du champ "Ethertype" de l'en-tête Ethernet, l'état suivant du parser est déterminé. En cas d'en-tête VLAN ultérieur, il s'agira d'un paquet Upsteam. L'etherType serait soit 0x8100 ou bien 0x88a8 dépendamment si le paquet contient un VLAN à étiquette double ou un VLAN à étiquette simple. Dans le cas où le paquet est doublement étiqueté, le "Parse_802.1AD" serait le prochain état du parser. cet état permet d'extraire l'en-tête du VLAN du service (S-TAG dans le code P4) et aussi d'indiquer l'état suivant qui serait dans ce cas là le "Parse_802.1Q" où aura lieu l'extraction de l'en-tête du VLAN du client (C-TAG dans le code P4). Pendant cet état, la valeur du champ Ethertype du VLAN "C-TAG" indiquera si le paquet PPPoE est un paquet de découverte (PPPoED) ou pas ; les paquets de découverte PPPoED sont des paquets communiqués entre le 5G-RG et l'AGF au moment d'établissement de nouvelles sessions ; Si le paquet est un message de découverte, on passe à l'état "Parse_PPPoE_v1" qui impliquera l'extraction de l'en-tête PPPoE ainsi que l'exécution de l'état final "Accept" permettant d'accepter et acheminer le paquet au pipeline d'action-correspondance P4. Sinon l'état "Parse_PPPoE_v1_v2" sera exécuté, permettant de distinguer les paquets PPPoE version 1 des paquets PPPoE version 2 qui se constitue de l'en-tête 5WE, puis un des états "Parse_PPPoE_v1" ou "Parse_5WE" serait exécuté dépendamment de la valeur du champ version de l'en-tête PPPoE.

Lorsqu'il s'agit d'un paquet Downstream, l'en-tête IPv4 sera extraite après avoir extrait l'Ethernet dans le premier état, puis une décision sera effectuée sur la valeur du champ "Protocol" pour transiter au prochain état "Pars_UDP" permettant d'extraire l'en-tête UDP. dans l'état "Parse_UDP", si la valeur du champ "src_port" vaut 0x2152, ceci indique que l'en-tête encapsulée à l'intérieur de l'UDP est bel et bien la 'GTPU'. Dans ce cas là, l'état suivant "Parse_GTPU" s'occupera de l'extraction de l'en-tête GTPU, ainsi de la transition au dernier état du parser de Downstream qui sera le "Parse_pdu_session_container" si la valeur du champ Next_hdr valait 0x85. Le reste du paquet excluant le "Payload" sera extrait

dans cet état et le paquet sera ensuite acheminé pour qu'il soit traité dans le pipeline d'action-correspondance P4.

3.3.2 Les tables de correspondance du pipeline

Le pipeline upstream commence d'abord par appliquer la table *t_vlan_5we*. Cette table vérifie la validité des paquets reçus en faisant correspondre les valeurs des clés aux valeurs des champs de l'identifiant VLAN (*VLAN ID*) du client et du service, ainsi de vérifier si la valeur du champ *Session_ID* de l'en-tête correspond elle aussi à la valeur de la clé de correspondance. Si les clés correspondent aux valeurs, l'action "set_pdu_session_id" sera exécutée pour mettre la valeur du champ *Session Id* dans le champ de la métadonnées qui lui correspond. Sinon, l'action par défaut "*Drop*" permet de rejeter tous les paquets inconnus et illégitimes pour des raisons de sécurité.

Si la session est légitime, la table *t_pdu_upf_routing* sera appliquée en vue de créer les en-têtes IP, UDP et la GTPU constituant la pile protocolaire qui va être envoyée et utilisée par le coeur du réseau 5G, notamment l'UPF. L'action "set_pdu_upf_hdr" exécutée par la table *t_pdu_upf_routing* permet également de décider vers quel port de destination le paquet GTPU créé devrait être envoyé.

Après la construction de la pile des protocoles IP/UDP/GTPU ainsi que l'extension PDU-Session-Container du GTPU, vient l'étape de la classification des paquets pour la qualité de service. La table *t_PDU_ID_QFI* permet de mettre la valeur reçue du champ *QFI* de l'en-tête 5WE dans celui de l'en-tête "PDU_Session_Container" construite. si la clé de la table ne correspond à aucune valeur, la valeur du *QFI* de l'en-tête PDU_Session_Container sera mise à 0 qui est la valeur de l'action par défaut de la table *t_PDU_ID_QFI*.

La dernière table du pipeline Upstream est la table *t_PDU_ID_5WE_ID*. Si la clé de cette table correspond bien à la valeur du champ *Session_Id* qui a été enregistrée dans les métadonnées, l'entrée *teid_id* de cette table sera utilisée pour remplir le champ *TEID* de l'en-tête GTPU, permettant ainsi de finaliser la construction de toutes les en-têtes du paquet qui sera acheminé dans ce cas là vers l'UPF destination. L'exécution de cette table entraînera également l'annulation des en-têtes VLAN de service, VLAN de client ainsi que l'en-tête 5WE qui étaient initialement présentes dans le paquet reçu en provenance du 5G-RG.

Comparé au pipeline de l'Upstream, le pipeline Downstream est plus simple et nécessite moins de traitement. Lorsqu'un paquet provenant de l'UPF entre dans l'AGF, nous commençons par vérifier s'il est légitime en effectuant une authentification de l'abonné basée sur l'adresse IP de destination et le port de destination de l'UDP. Ces deux paramètres servent de clés pour

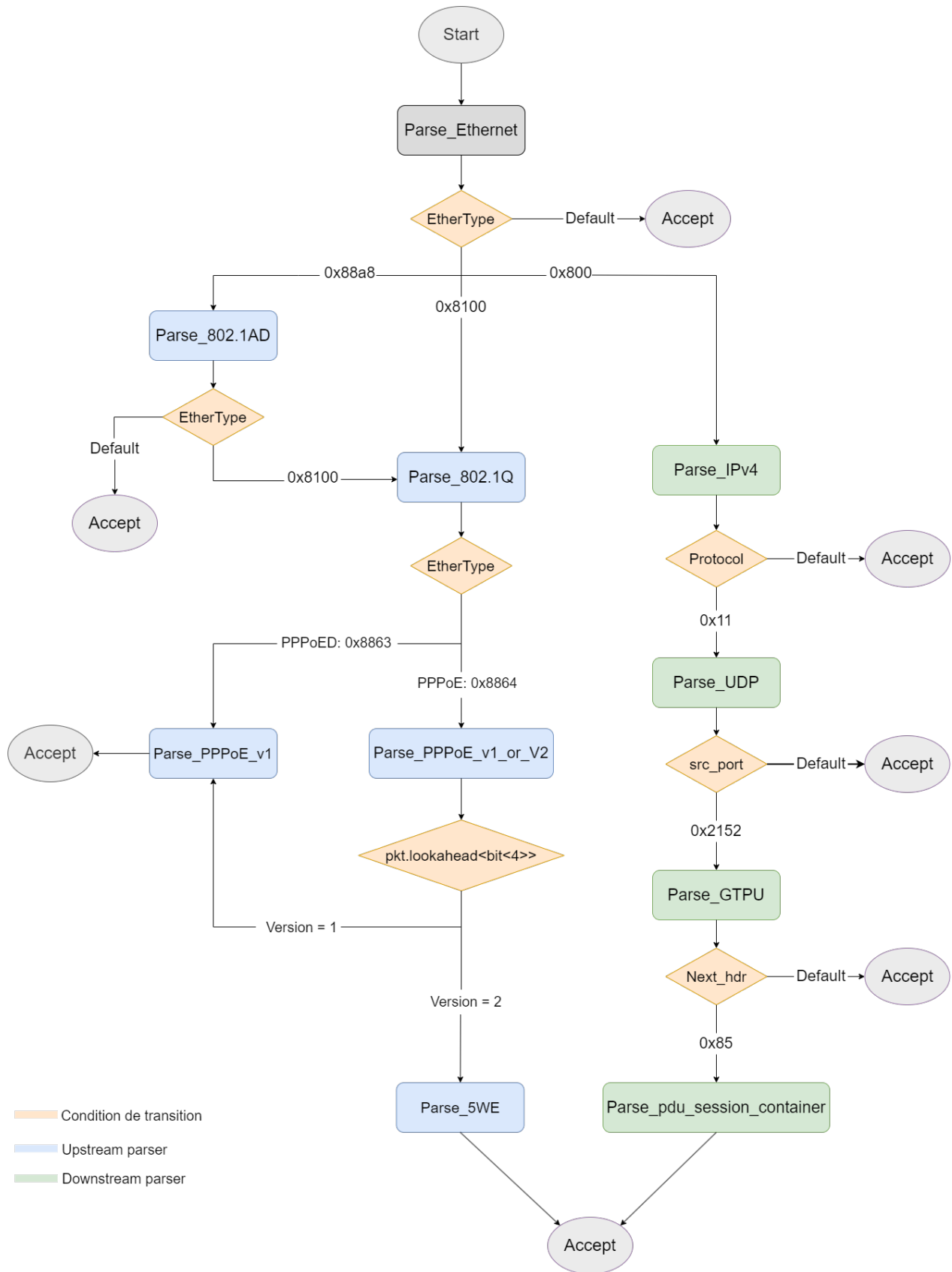


FIGURE 3.10 Le parser de l'AGF

la première table, appelée "*t_upf_ip_udp*", du pipeline Downstream. Cette table exécute uniquement l'action "NoAction", ce qui signifie qu'elle ne modifie pas le paquet en cours de traitement. Si le paquet est considéré comme valide, l'en-tête UDP est supprimé et le paquet est acheminé vers l'étape suivante du traitement. En revanche, si le paquet est considéré comme non valide, l'action par défaut "*Drop*" sera appliquée pour rejeter le paquet.

Après avoir authentifié l'abonné via l'adresse IP, l'adresse MAC de la passerelle résidentielle 5G-RG, l'identifiant de session 5WE négocié, ainsi que l'identifiant de la session PDU sont utilisés comme entrée pour la table *t_TEID*. Cette table enregistre l'identifiant de la session PDU dans les métadonnées du paquet. De plus, elle assigne la valeur de l'adresse MAC du 5G-RG au champ de l'adresse MAC destination de l'Ethernet, remplace la valeur de l'adresse MAC source par celle de l'AGF, et met la valeur appropriée de l'identifiant de session dans l'en-tête 5WE.

Après avoir créé les en-têtes qui composent le paquet Downstream, destiné à être transmis de l'AGF à la 5G-RG, il est nécessaire de mettre en place une politique de qualité de service (QoS). Selon la valeur du champ *QFI* reçue dans l'en-tête PDU-Session-Container, qui sert de clé pour la table *t_qfi*, l'AGF remplira le champ *QFI* de l'en-tête 5WE avec la valeur reçue de l'extension PDU-Session-Container du GTPU. Si aucune correspondance n'est trouvée dans la table, l'action par défaut sera appliquée, fixant ainsi la valeur du champ *QFI* de l'en-tête 5WE à "0".

Si une correspondance se produit, La dernière table nommée *t_VLAN* prend la décision sur quel port destination le paquet doit être envoyé. L'action correspondante permet ainsi de remplir les champs des deux identifiants des VLAN (*Vlan_ID*) du client et de service avec les valeurs indiquées dans l'entrée de cette action.

cette étape de traitement permet d'annuler la pile d'en-têtes d'origine composée de l'IP, de l'UDP, du GTPU et de son extension, et la remplacer par le paquet final qui se compose de l'Ethernet, les deux VLANs ainsi que de l'en-tête 5WE.

En ce qui concerne le traitement des paquets du pipeline de contrôle, décrit en rouge dans la Figure 3.9, deux tables de correspondance s'occupent de la gestion et le routage de ces paquets de contrôle. La table "*t_pppoe_cp*" permet de vérifier les champs Version et Code du PPPoE reçu de la 5G-RG pour déterminer la légitimité du paquet en vérifiant s'il est un paquet de contrôle pour l'AGF. Si la version vaut "1", il s'agit bien d'un paquet PPPoE du plan de contrôle. Puis, il reste à vérifier la valeur du champ Code, cette valeur doit correspondre aux valeurs définies par l'IANA [44] dépendamment du type de paquet, s'il est de type PADI, PADO, PADR ou PADS. si la valeur du code correspond à une de ces valeurs, le paquet est ensuite acheminé sans aucun traitement ultérieur vers le port de contrôle (Le port de sortie

de notre programme P4 prendra la valeur du port CPU). Sinon, le paquet sera rejeté.

La deuxième table de controle, *t_l2_fwd*, s'occupe plutôt du routage du paquets de l'autre direction. Ceux qui sont en provenance du plan de controle de l'AGF (Via le port CPU) et destinés à être envoyer à la 5G-RG. Cette table vérifie seulement l'adresse MAC destination du paquet pour déterminer le port de sortie où l'envoyer.

CHAPITRE 4 EVALUATION DES PERFORMANCES : RÉSULTATS ET ANALYSE

4.1 Configuration expérimentale sur DPDK

Cette section fournira une description détaillée du banc d'essai utilisé dans cette étude, ainsi que des méthodes et des outils employés pour obtenir les résultats présentés dans les sections suivantes.

4.1.1 Un mot sur le générateur de trafic

Les générateurs de trafic englobent une gamme d'outils logiciels et matériels conçus pour générer sur un port réseau donné un volume substantiel de trafic réseau, qui est ensuite prévu d'être reçu sur un autre port réseau. Le générateur de trafic (TG) est capable de fournir des mesures telles que la latence, le débit et d'autres métriques utilisées pour évaluer la performance du dispositif réseau connecté entre les ports du générateur de trafic, également connu sous le nom de dispositif en cours de test (*Device Under Testing* en anglais : DUT). La figure 4.1 représente une configuration de test réseau typique de générateur de trafic.

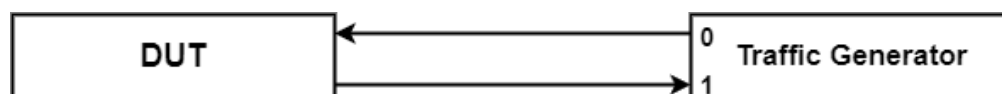


FIGURE 4.1 Une configuration de test avec un générateur de trafic

Bien que de nombreux générateurs de trafic commerciaux soient déjà disponibles sur le marché, il existe également une importante collection de projets open source qui s'efforcent de reproduire des modèles de trafic authentiques. Ces projets visent à soumettre l'équipement testé à des charges réalistes, améliorant ainsi sa fiabilité et ses performances. Parmi les différents projets, on trouve plusieurs projets notables tels que pktgen-DPDK, Moongen et TRex, tous construits sur l'architecture DPDK.

Particulièrement, TRex, qui a été développé par Cisco, offre aux clients un large éventail d'options de personnalisation, leur permettant de modifier facilement les profils de trafic. De plus, TRex inclut des API d'automatisation conviviales, simplifiant ainsi la mise en œuvre de tâches automatisées. En outre, la flexibilité de sa structure permet l'intégration transparente de comportements à la fois sans état et étatiques (*Stateless and stateful behaviors* en anglais), élargissant ainsi le champ des appareils pouvant faire l'objet de tests.

4.1.2 Banc d'essai

La figure 4.2 présente le banc d'essai utilisé pour évaluer la performance de notre prototype AGF sur le commutateur DPDK. Ce banc d'essai a été construit de deux machines basées sur des processeurs Intel, la machine en cours de test (DUT) qui a été utilisée pour configurer le prototype AGF et le rouler sur DPDK est basé sur le processeur Intel(R) Xeon(R) (8 cœur, avec 2 thread pour chaque cœur).

Chaque machine est équipée d'une carte réseau Intel XXV710, qui utilise des câbles de type SFP28 capables de transporter des paquets à des débits allant jusqu'à 25GbE.

Pour la génération de trafic, nous avons installé TREX sur l'autre machine, le débit maximal qu'on a pu générer avec TREX était de l'environ de 10Gbps.

Après avoir configuré le banc d'essai ainsi que DPDK pour le lancement de notre prototype, nous avons procédé aux expériences en mesurant nos métriques pour différents scénarios expérimentaux. Nous avons débuté en générant un trafic composé d'une seule session avec un seul flux. Ensuite, nous avons augmenté le nombre de sessions à 1000. Par la suite, nous avons mesuré la performance de notre implémentation pour 5000 sessions. Enfin, dans le dernier scénarionous, nous avons atteint le maximum de sessions que TREX était capable de supporter, soit 20000 sessions.

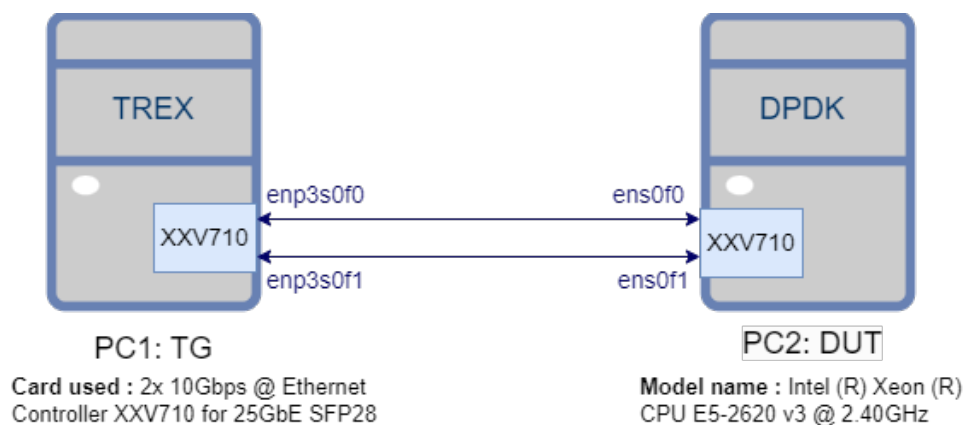


FIGURE 4.2 Banc d'essai utilisé pour l'évaluation de performances avec DPDK

4.2 Métriques

L'évaluation des performances de notre prototype AGF sur DPDK en utilisant un générateur de trafic pour différents scénarios expérimentaux nous conduit à analyser de près plusieurs métriques clés qui permettront d'évaluer l'efficacité et la stabilité de l'AGF dans un scénar-

rio de déploiement réel au sein d'un réseau 5G. Les expériences réalisées nous ont permis d'examiner quatre métriques majeures :

- Le débit : Cette mesure est essentielle pour évaluer la capacité de l'AGF à gérer un volume élevé de trafic de manière fluide et efficace.
- Le taux de perte de paquets.
- Le nombre de sessions simultanées que peut l'AGF supportée sur un commutateur DPDK.
- Le taux d'utilisation des ressources matérielles.

4.3 Résultats expérimentaux

4.3.1 Débits

Le tableau 4.1 fournit un aperçu des débits générés par le générateur de trafic pour chaque scénario expérimental. Lors de la première expérience avec une seule session générée, un débit total de 2.73 Gbps a été atteint. On a augmenté le nombre de sessions à 1000 sessions pendant la deuxième expérience et une légère baisse de débit a été constaté (2.41Gbps). Pour le troisième scénario expérimental avec 5000 flux de paquets générés, le débit total observé a diminué d'environ la moitié par rapport à la première expérience avec un seul flux a été observée (1,52 Gbps). Enfin, les résultats de la dernière expérience ont démontré une chute significative de débit total qui a été réduit à 1.15Gbps.

Il est essentiel de reconnaître qu'il y a une différence entre le débit total en transmission et celui en réception. Cette dissemblance de taille de paquets entre les paquets GTP-U et 5WE explique ces résultats. En effet, des paquets GTP-U sont envoyés tandis que des paquets 5WE sont reçus, et vice versa. De plus, quelques pertes de paquets qui ont eu lieu en augmentant le débit peuvent également justifier cette différence.

TABLEAU 4.1 Le débit mesuré de l'AGF en augmentant le nombre de sessions

Nombre de sessions	Débit (Tx)	Débit (Rx)
1 session	2.73 Gbps	1.36 Gbps
1 000 sessions	2.41 Gbps	1.12 Gbps
5 000 Sessions	1.52 Gbps	871 Mbps
20 000 sessions	1.15 Gbps	412 Mbps

4.3.2 Taux de perte de paquets

Les résultats montrent que les performances de l'exécution de l'AGF sur DPDK diminuent à mesure que le nombre de sessions augmente. En effet, la figure 4.2 indique que, pendant l'exécution des différents scénarios expérimentaux, des pertes de paquets ont été constatées. Pendant la première expérience avec un seul flux de paquets généré, aucune perte de paquet n'a eu lieu avec tous les débits générés.

Durant la deuxième expérience, on a remarqué qu'une perte de paquets a commencé à avoir lieu à partir des débits de 1.8Gbps. Environ 8% de paquets envoyés a été perdus.

Le troisième scénario, avec 5000 sessions générées, on a pu atteindre un débit de 1.15Gbps sans pour autant qu'il y'ait une perte de paquets. Après 1.15Gbps, la performance a diminué et une perte de paquet a commencé à avoir lieu.

La performance du système a significativement été réduit lorsqu'on a généré 20000 sessions, nous étions capable de seulement générer un débit de 800Mbps sans avoir des pertes de paquets.

Cette perte de paquets se justifie par le fait que l'augmentation de nombre de sessions implique aussi l'augmentation de nombre de champs à ajouter ou modifier pour les différentes en-têtes du système. Le thread PMD (Poll Mode Driver) utilisé par DPDK est responsable de la gestion des paquets d'entrée, voire les classifier une fois reçus ainsi qu'exécuter des actions sur ces paquets une fois classifiés. Si ce Thread n'arrive pas à consommer rapidement le trafic, la file d'attente matérielle de la carte NIC déborde et les paquets sont perdus.

Lorsqu'une seule session était utilisée, la DPDK nécessitait moins de traitement, et le thread PMD n'était jamais saturé. Cependant, lorsque le nombre de sessions a été augmenté, plusieurs règles ont été installées, et les paquets ont commencé à passer plus de temps dans le pipeline de traitement pour apporter des modifications aux en-têtes. Cela a conduit à une surcharge du thread PMD, qui a alors bloqué des paquets dans la file d'attente, entraînant ainsi une saturation et des pertes de paquets.

4.3.3 Nombre de sessions simultanées supportées

Une session, dans le cadre de notre application AGF, correspond à un flux spécifique de paquets qui subissent tous un traitement identique tout au long de leur processus de transfert de leur source vers leur destination. Le nombre maximal de sessions qui peuvent être supportées est déterminé par le nombre de règles que peut installer le commutateur dans ses tables de routages.

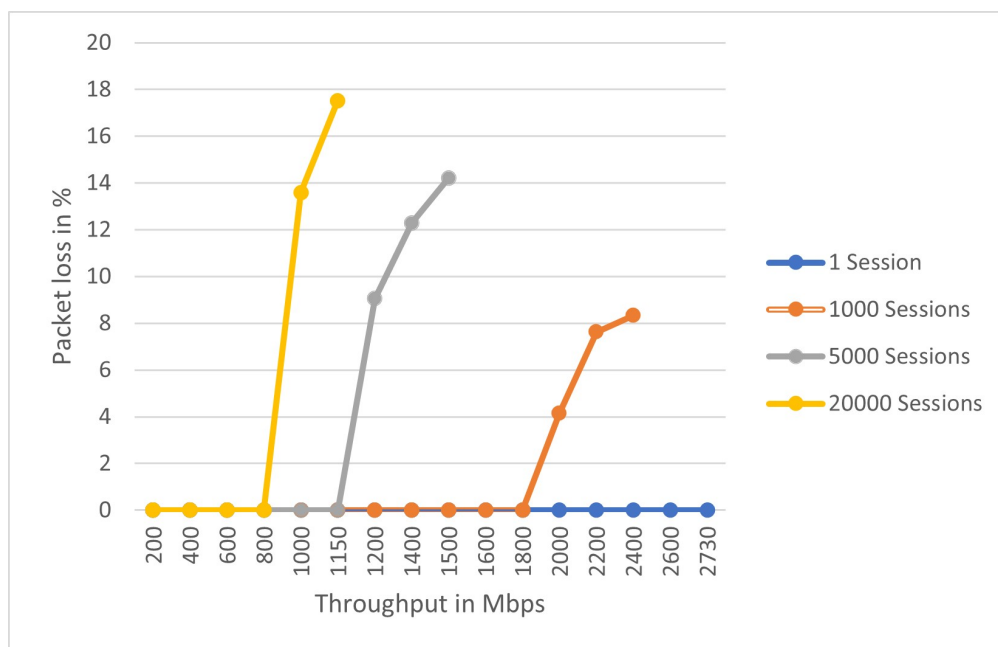


FIGURE 4.3 Le taux de perte de paquets pour les différents cas de test

Théoriquement, le nombre de règles que peut supporter une table de correspondance programmée en P4 est décrit par le nombre de combinaison de ses clés. Dans notre prototype, La table "t_TEID" effectue une correspondance sur deux clés qui sont l'adresse IP source et le champ TEID. Chaque clé est sur 32 bits, ce qui offre ($2^{32} \times 2^{32}$) combinaisons possibles. Cette valeur ne peut être supportée par aucun commutateur, qu'il soit logiciel ou matériel.

En pratique, lors de la programmation en langage P4, il est possible de définir la taille maximale d'une table en tant que paramètre au moment de sa déclaration dans le code P4. Si le compilateur parvient à compiler le code P4 sans erreur, cela signifie que le commutateur en question est capable de prendre en charge le nombre de règles qui a été indiqué dans cette table du code.

Dans notre cas, nous avons utilisé le compilateur P4-DPDK qui permet compiler notre code P4 en plusieurs fichiers permettant de configurer et mettre en place le pipeline DPDK. Afin d'évaluer le nombre de sessions maximales que le commutateur DPDK peut prendre en charge, nous avons compilé le code P4 avec différentes tailles de la table "t_TEID". Le nombre maximal d'entrées que nous avons réussi à obtenir pour la table "t_TEID" sans rencontrer d'erreurs de compilation est de 2 147 483 647.

Nous n'avons pas pu relever des métriques pour ce nombre de sessions, parce que le générateur de trafic, TREX, que nous avons utilisé ne permet pas de générer plus que 20000 sessions.

4.3.4 Utilisation des ressources matérielles

Le DPDK utilise un modèle d'exécution complète pour le traitement des paquets, ce qui nécessite la préallocation de toutes les ressources avant d'appeler les applications de la couche de traitement des données.

En effet, Le DPDK se sert de la couche d'abstraction de l'environnement (*Environment Abstraction Layer* : EAL) pour accéder aux ressources de bas niveau telles que le matériel et la mémoire. Au moment de l'initialisation d'une application DPDK, le processus qui en est responsable, *rte_eal_init()*, appelle des thread qui permettent d'initialiser l'application ainsi de choisir comment distribuer les diverses ressources physiques telles que la mémoire et le matériel.

Limitations matérielles

DPDK utilise les "hugepages" de Linux pour l'allocation du grand pool mémoire (*Large memory pool* en anglais) utilisé pour les tampons de paquets (*Packet buffers*). Cette fonctionnalité permet de réduire l'utilisation des tampons de traduction d'adresse (*Translation lookaside buffer* en anglais : TLB) et les pertes de cache.

L'utilisation des "hugepages" doit être fait avant le lancement de l'application DPDK, sinon l'application ne fonctionnera pas. Une application DPDK peut utiliser soit des "hugepages" de 2 Mo, soit des "hugepages" de 1 Go.

La machine que nous avons utilisé dans notre configuration expérimentale est basée sur une ancienne architecture Intel qui ne supporte pas les "hugepages" de 1Go. Seuls les "hugepages" de 2 Mo sont accessibles sur cette machine, cette contrainte pourrait avoir légèrement restreint les performances. De manière similaire, la carte mère ne disposent pas de connecteurs PCI 3.0, ce qui pourrait réduire le débit total des cartes réseau, car celles-ci atteignent leur performance maximale sur des plates-formes PCI 3.0.

Taux d'utilisation de la CPU

Pour calculer le taux d'utilisation de la CPU, nous avons utilisé l'outil *Flame graphs*, cet un outil Open source [46] donne une représentation graphique de données hiérarchiques pour visualiser les traces de pile (*stack traces*) d'un logiciel profilé, DPDK dans notre cas. le *Flame graphs* se sert de l'outil *Perf* pour profiler la consommation de la CPU.

Au moment de lancement de l'application DPDK. Dans notre cas, lors du lancement du script *bf_switchd* [47], plusieurs bibliothèques dont se sert le DPDK pour le traitement sont initialisés, Parmi ces bibliothèques, on distingue :

- *rte_swx_pipeline_run* : cette bibliothèque permet d'initialiser et exécuter le pipeline qui a

été créé pendant la compilation du code P4, la visualisation sur *Flame graphs* montre qu'elle consomme 4.28% de CPU.

- `rte_pktmbuf_free` : cette librairie est celle responsable de la création et la suppression des tampons (buffers) que l'application DPDK peut utiliser pour stocker des tampons de paquets. Ces tampons sont stockés dans un "mempool" (pool de mémoire) après l'exécution du script `bf_switchd`, cette librairie consomme environ 10% de la CPU.
- Les processus "`instr_hdr`", "`table_lookup`", "`instr_table_exec`" et d'autres processus dont Graph Flames n'a pas pu identifier le nom, sont responsables de l'extraction des en-têtes dans l'étage du parser, ainsi que l'exécution des tables et la modification des champs des différentes en-têtes lorsque le paquet entre le pipeline de traitement. Ces processus consomment environ 15% du total de la CPU.
- `lcore_worker_1` : La librairie est chargée de gérer l'interaction avec les interfaces NIC. Elle prend en charge la réception des paquets RX provenant de l'interface RX qui lui est dédiée, puis elle distribue ces paquets reçus vers les différents threads de travail pour un traitement ultérieur. De même, elle gère l'envoi des paquets TX depuis les différents threads vers l'interface de sortie. Cette librairie DPDK est responsable de l'utilisation d'une grande partie du CPU, représentant environ 22% de l'utilisation totale.

Les expériences réalisées ont démontré que l'augmentation du nombre de sessions par TREX n'avait pas un impact significatif sur l'utilisation du CPU. La consommation totale de CPU est restée constamment estimée à environ 50% (comme montré dans le tableau 4.2).

TABLEAU 4.2 Taux d'utilisation de CPU

Librairie	Taux d'utilisation de CPU
<code>rte_swx_pipeline_run</code>	4.28%
<code>rte_pktmbuf_free</code>	10.51%
<code>table_lookup</code>	5.26%
<code>instr_table_exec</code>	3.71%
<code>instr_hdr</code>	2.6%
<code>lcore_worker_1</code>	22.18%
autres paramètres	3.14%
Total	51.68%

4.4 Conclusions et recommandations

Les résultats de l'évaluation expérimentale démontrent la faisabilité et l'efficacité de l'implémentation du plan de données de la fonction AGF sur un commutateur DPDK programmable en P4. Bien que DPDK n'ait présenté aucune limitation en termes d'utilisation de mémoire, sa capacité de traiter de grand débits de données s'est avérée être un élément contraignant dans la gestion simultanée de plusieurs sessions.

CHAPITRE 5 CONCLUSION

5.1 Synthèse des travaux

L'objectif principal de ce travail de recherche était d'établir la viabilité d'une fonction de passerelle d'accès (AGF) grâce à l'examen de ses spécifications techniques et à la facilitation de son implémentation en P4. L'examen approfondi des spécifications nous a permis de comprendre les critères et les fonctionnalités nécessaires de l'AGF, facilitant ainsi sa mise en œuvre réalisable.

De plus, notre étude a permis une évaluation complète des avantages et des inconvénients du prototype AGF sur DPDK en utilisant le langage P4. Cette évaluation a fourni des informations quantitatives sur l'efficacité de DPDK dans le renforcement du plan de données de l'AGF et la gestion d'un volume substantiel de sessions simultanées.

Enfin, suite à cette recherche menée dans ce travail de maîtrise, des recommandations importantes ont été formulées en vue d'un déploiement potentiel de la solution AGF à l'échelle industrielle. Ces recommandations prennent en considération les performances observées du prototype, les limitations identifiées tout au long du processus de conception, ainsi que les exigences spécifiques de l'industrie.

5.2 Limitations de la solution proposée

Pour évaluer la capacité de DPDK à améliorer le traitement du plan de données de l'AGF, nous avons utilisé des outils open source de test et de génération de trafic. Bien que DPDK n'ait présenté aucune limitation en termes de mémoire tout en supportant un très grand nombre de sessions (des millions de sessions peuvent être prises en charge par DPDK), sa capacité de traiter de nombreux gigabits de données par seconde s'est avérée être un élément contraignant dans la gestion simultanée de plusieurs sessions.

Les expériences ont été également impactées par les limitations matérielles de l'équipement utilisé. La machine Intel Xeon sur laquelle DPDK a été configuré était relativement ancienne et ne supportait que les "Hugepages" de 2MB. De plus, le générateur de trafic avait également des limitations, étant incapable de générer des millions de sessions et se limitant à quelques milliers seulement.

5.3 Améliorations futures

Pour pallier les limitations associées à DPDK, diverses approches peuvent être envisagées, dont l'utilisation conjointe de DPDK avec une carte FPGA. Cette combinaison peut contribuer à améliorer les performances de l'AGF. Lorsque le pipeline de DPDK atteint sa saturation et que la file d'attente des cartes réseau connectées à DPDK est encombrée, les paquets peuvent être redirigés vers la FPGA pour poursuivre leur traitement. Cette décharge de travail vers la FPGA permet de soulager DPDK et d'optimiser le traitement global des paquets, offrant ainsi une meilleure efficacité dans le système de gestion du trafic réseau.

RÉFÉRENCES

- [1] H. Zhang, Z. Chen et Y. Yuan, “High-performance UPF design based on DPDK,” dans *2021 IEEE 21st International Conference on Communication Technology (ICCT)*. IEEE, 2021, p. 349–354.
- [2] M. Jaber, M. A. Imran, R. Tafazolli et A. Tukmanov, “5G backhaul challenges and emerging research directions : A survey,” *IEEE Access*, vol. 4, p. 1743–1766, 2016.
- [3] C. Behrens, S. Krauß, E. Weis et D. Breuer, “Technologies for convergence of fixed and mobile access : An operator’s perspective,” *J. Opt. Commun. Netw.*, vol. 10, n^o. 1, p. A37–A42, Jan 2018.
- [4] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky et S. Uhlig, “Software-defined networking : A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, n^o. 1, p. 14–76, 2015.
- [5] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker et J. Turner, “Openflow : Enabling innovation in campus networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, n^o. 2, p. 69–74, 2008.
- [6] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese *et al.*, “P4 : Programming protocol-independent packet processors,” *ACM SIGCOMM Computer Communication Review*, vol. 44, n^o. 3, p. 87–95, 2014.
- [7] “TR-456 : AGF functional requirements, Issue : 02,” BroadBand Forum, Rapport technique, 2022.
- [8] DPDK Community, “Data plane development kit (DPDK),” Web Page, 2023. [En ligne]. Disponible : <https://www.dpdk.org/>
- [9] R. Ricart-Sanchez, P. Malagon, J. M. Alcaraz-Calero et Q. Wang, “P4-NetFPGA-based network slicing solution for 5G mec architectures,” dans *2019 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*. IEEE, 2019, p. 1–2.
- [10] K. Benzekki, A. El Fergougui et A. Elbelrhiti Elalaoui, “Software-defined networking (sdn) : A survey. security and communication networks. vol. 9 (18). 5803–5833,” 2016.
- [11] M.-K. Shin, K.-H. Nam et H.-J. Kim, “Software-defined networking (SDN) : A reference architecture and open APIs,” dans *IEEE 2012 International Conference on ICT Convergence (ICTC)*, 2012, p. 360–361.

- [12] “Network Functions Virtualisation, an introduction, benefits, enablers, challenges and call for action,” https://portal.etsi.org/NFV/NFV_White_Paper.pdf.
- [13] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco et F. Huici, “{ClickOS} and the art of network function virtualization,” dans *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, 2014, p. 459–473.
- [14] “TS 23.501 : system architecture for the 5G system (5GS),” 3rd Generation Partnership Project (3GPP), Rapport technique.
- [15] J. Zhou, Z. Ma, W. Tu, X. Qiu, J. Duan, Z. Li, Q. Li, X. Zhang et W. Li, “Cable : A framework for accelerating 5G UPF based on eBPF,” *Computer Networks*, vol. 222, p. 109535, 2023.
- [16] C. Rotter et T. Van Do, “A queueing model for threshold-based scaling of UPF instances in 5G core,” *IEEE Access*, vol. 9, p. 81 443–81 453, 2021.
- [17] A. Bose, D. Maji, P. Agarwal, N. Unhale, R. Shah et M. Vutukuru, “Leveraging programmable dataplanes for a high performance 5G user plane function,” dans *Asia-Pacific Workshop on Networking (APNet 2021)*, 2021, p. 57–64.
- [18] S. K. Singh, C. E. Rothenberg, J. Langlet, A. Kassler, P. Vörös, S. Laki et G. Pongrácz, “Hybrid P4 programmable pipelines for 5G gNodeB and user plane functions,” *IEEE Transactions on Mobile Computing*, 2022.
- [19] Z. Cong, Z. Baokang, W. Baosheng et Y. Yulei, “CeUPF : Offloading 5G user plane function to programmable hardware base on co-existence architecture,” dans *Proceedings of the ACM International Conference on Intelligent Computing and its Emerging Applications*, 2021, p. 34–39.
- [20] R. MacDavid, C. Cascone, P. Lin, B. Padmanabhan, A. Thakur, L. Peterson, J. Rexford et O. Sunay, “A p4-based 5G user plane function,” dans *Proceedings of the ACM SIGCOMM Symposium on SDN Research (SOSR)*, 2021, p. 162–168.
- [21] BBF TR-470, “5G Wireless Wireline Convergence Architecture,” <https://www.broadband-forum.org/technical/download/TR-470.pdf>.
- [22] P. Vörös, G. Pongrácz et S. Laki, “Towards a hybrid next generation NodeB,” dans *Proceedings of the P4 Workshop in Europe*, 2020, p. 56–58.
- [23] H. Woesner, C. Lange, R. Schlenk, M. Schlosser et D. Kosiankowski, “Virtualization in high-throughput network elements and its impact on energy consumption,” dans *Photonic Networks ; 17. ITG-Symposium ; Proceedings of. VDE*, 2016, p. 1–6.
- [24] R. Kundel, L. Nobach, J. Blendin, W. Maas, A. Zimmer, H.-J. Kolbe, G. Schyguda, V. Gurevich, R. Hark, B. Koldehofe *et al.*, “OpenBNG : Central office network functions

- on programmable data plane hardware,” *International Journal of Network Management*, vol. 31, n^o. 1, p. e2134, 2021.
- [25] BBF TR-456, “AGF Functional Requirements,” <https://www.broadband-forum.org/technical/download/TR-456.pdf>.
- [26] M. Budiu et C. Dodd, “The p416 programming language,” *SIGOPS Oper. Syst. Rev.*, vol. 51, n^o. 1, p. 5–14, sep 2017. [En ligne]. Disponible : <https://doi.org/10.1145/3139645.3139648>
- [27] F. Hauser, M. Häberle, D. Merling, S. Lindner, V. Gurevich, F. Zeiger, R. Frank et M. Menth, “A survey on data plane programming with P4 : Fundamentals, advances, and applied research,” *Journal of Network and Computer Applications*, p. 103561, 2022.
- [28] P. G. K. Patra, F. E. R. Cesen, J. S. Mejia, D. L. Feferman, L. Csikor, C. E. Rothenberg et G. Pongracz, “Toward a sweet spot of data plane programmability, portability, and performance : On the scalability of multi-architecture P4 pipelines,” *IEEE Journal on Selected Areas in Communications*, vol. 36, n^o. 12, p. 2603–2611, 2018.
- [29] J. Viktorin et J. Korenek, “Packet processing on FPGA SoC with DPDK,” dans *International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2016, p. 1–2.
- [30] H. Singh, C. Huang, M. Sicard-Gagne, G. Shami, M. Lyonnais, D. Fedorov et R. Wilson, “INT-SDN : Evaluation of various P4 parameters using optical telemetry having reconfigurable data plane on 40 Gbps line rate,” dans *Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. IEEE, 2019, p. 1–6.
- [31] “TREX,” https://trex-tgn.cisco.com/trex/doc/trex_manual.html.
- [32] S. Laki, P. Vörös et F. Fejes, “Towards an AQM evaluation testbed with P4 and DPDK,” dans *Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos*, 2019, p. 148–150.
- [33] H. Harkous, M. Jarschel, M. He, R. Pries et W. Kellerer, “P8 : P4 with predictable packet processing performance,” *IEEE Transactions on Network and Service Management*, vol. 18, n^o. 3, p. 2846–2859, 2020.
- [34] P. Vörös, D. Horpácsi, R. Kitlei, D. Leskó, M. Tejfel et S. Laki, “T4P4S : A target-independent compiler for protocol-independent packet processors,” dans *International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2018, p. 1–8.
- [35] “Behavioral model : The reference p4 software switch,” <https://github.com/p4lang/behavioral-model>.

- [36] J. S. da Silva, T. Stimpfling, T. Luinaud, B. Fradj et B. Boughzala, “One for all, all for one : A heterogeneous data plane for flexible P4 processing,” dans *International Conference on Network Protocols (ICNP)*. IEEE, 2018, p. 440–441.
- [37] R. Kundel, A. Rizk, J. Blending, B. Koldehofe, R. Hark et R. Steinmetz, “P4-codel : Experiences on programmable data plane hardware,” dans *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, p. 1–6.
- [38] M. Kekely et J. Korenek, “Mapping of P4 match action tables to FPGA,” dans *International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2017, p. 1–2.
- [39] S. K. Singh, C. E. Rothenberg, G. Patra et G. Pongracz, “Offloading virtual evolved packet gateway user plane functions to a programmable ASIC,” dans *Proceedings of the ACM CoNEXT Workshop on Emerging in-Network Computing Paradigms*, 2019, p. 9–14.
- [40] C.-A. Shen, D.-Y. Lee, C.-A. Ku, M.-W. Lin, K.-C. Lu et S.-Y. Tan, “A programmable and FPGA-accelerated GTP offloading engine for mobile edge computing in 5G networks,” dans *Conference on Computer Communications Workshops (INFOCOM)*. IEEE, 2019, p. 1021–1022.
- [41] G. Matos, F. L. Verdi, L. M. Contreras et L. C. de Almeida, “When srv6 meets 5G core : Implementation and deployment of a network service chaining function in smartnics,” *arXiv preprint arXiv :2107.11966*, 2021.
- [42] “A Method for Transmitting PPP Over Ethernet (PPPoE),” <https://www.rfc-editor.org/rfc/rfc2516.html>.
- [43] IETF, “5G Wireless Wireline Convergence User Plane Encapsulation (5WE) ,” <https://datatracker.ietf.org/doc/html/rfc8822>.
- [44] “User Plane Protocol and Architectural Analysis on 3GPP 5G System,” <https://datatracker.ietf.org/doc/html/draft-ietf-dmm-5g-uplane-analysis-02>.
- [45] 3GPP TS 29.281, “General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U),” https://www.etsi.org/deliver/etsi_ts/129200_129299/129281/17.04.00_60/ts_129281v170400p.pdf.
- [46] “Flame Graphs,” <https://www.brendangregg.com/flamegraphs.html>.
- [47] “P4-DPDK-Target,” <https://github.com/p4lang/p4-dpdk-target>.