UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MISE EN PLACE D'UNE SOLUTION DE SANTÉ MOBILE BASÉE SUR L'AUTHENTIFICATION ANONYME ET LA FAIBLE LATENCE

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

SYRINE RAJHI

FÉVRIER 2022

REMERCIEMENTS

Au terme de ce mémoire, je tiens à exprimer mes sincères remerciements ainsi que toute ma gratitude à Pr.Halima ELBIAZE ma directrice de recherche pour m'avoir encadrée et guidée tout au long de ma maitrise ainsi que pour l'expérience enrichissante et pleine d'intérêt qu'elle m'a offerte. Tout au long de l'élaboration de ce mémoire, j'ai eu l'honneur de bénéficier de son soutien scientifique, moral et financier.

Je tiens particulièrement à remercier Pr. Sébastien GAMBS mon co-directeur de recherche pour sa disponibilité ainsi que ses conseils avisés et ses orientations précieuses et aides apportés lors des différentes étapes de ma maitrise.

Que ce travail soit le modeste témoignage de ma haute considération et mon profond respect.

Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Sans oublier de remercier vivement le corps professoral de la faculté des sciences de l'UQAM.

Je réitère mes remerciements et respects envers mes très chers parents, mon frère et mes sœurs pour leur soutien indéfectible et inconditionnel. J'espère qu'ils retrouveront à travers ce travail, un gage de l'estime et de la gratitude que je leur porte.

Syrine RAJHI

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ACRONYMES

API	*Application Programming Interface*
CEP	*Complex Event Processing*
CCN	*Content-Centric Networking*
CPU	*Central processing unit*
CS	*Content Store*
DONA	*Data-Oriented Network Architecture*
FaX	*Find and Execute*
FIB	*Forwarding Information Base*
FoPax	*Find or Pull and Execute*
FoX	*Find or Execute*
FSX	*Find, Split and Execute*
ICN	*Information Centric Networking*
IdO	Internet des Objets
INC	*In-Network Computing*
IP	Protocole Internet
M-IdO	IdO Médical
m-santé	santé mobile
NACK	*Negative Acknowledgment*
NBR	*Name Based Routing*
NDO	*Named Data Objects*
NDN	*Named Data Networking*
NetInf	*Network of Information*
NFN	*Named Function Networking*
NRS	*Name Resolution Service*
OF	OpenFlow
ONF	*Open Networking Foundation*

OMS	Organisation Mondiale de la Santé
PIT	*Pending Interest Tables*
PKI	*Public Key Infrastrucutre*
PSIRP	*Publish-Subscribe Internet Routing Paradigm*
PURSUIT	*Publish Subscribe Internet Technology*
QoE	*Quality of Experience*
QoS	*Quality of Service*
RH	*Resolution Handler*
R2C	*Request to Compute*
SDN	*Software Defined Networking*
TLS	*Transport Layer Security*
UDP	*User Datagram Protocol*
URL	*Uniform Resource Locator*
vCPU	*virtual CPU*
Zo	Zettaoctet

RÉSUMÉ

Au cours des dernières années, l'Internet des objets (IdO) s'est développé dans de nombreux domaines, dont principalement l'IdO médical et particulièrement la santé mobile (m-santé). En effet, les applications de m-santé ont pour objectif de générer et de produire une quantité de données à traiter afin de prendre des décisions médicales fiables dans un délai restreint et sont caractérisées par des exigences strictes. De plus, la sécurité, la confidentialité et la protection de la vie privée des utilisateurs doivent être rigoureusement assurées du fait que ces applications échangent des informations médicales et personnelles hautement confidentielles. Compte tenu des différents avantages offerts aux patients, médecins et autorités sanitaires en termes de temps de réponse faible et la rapidité d'accès, les applications de m-santé sont devenus primordiaux comme étant des outils de prévention et de lutte contre les infections.

Étant donné que des milliards d'appareils IdO connectés à Internet génèrent une quantité massive de données, l'infrastructure actuelle de l'infonuagique (en anglais, *cloud computing*) et de l'informatique en périphérie (en anglais, *edge computing*) sont remises en question en termes de temps de réponse et de fiabilité des résultats en raison de leurs délais de réponse élevés dûs à la surcharge du réseau et de leurs distances par rapport à l'utilisateur final. En outre, ces deux technologies représentent une cible parfaite pour les attaquants en raison de leurs faiblesses en termes de sécurité, de confidentialité et de leurs expositions à un grand nombre de menaces et attaquants.

L'objectif de ce mémoire consiste à offrir un délai restreint et une fiabilité des résultats pour les applications IdO en utilisant le réseau des fonctions nommées (en anglais, *Named Function Networking* (NFN)) et le réseau défini par logiciel (en anglais, *Software Defined Networking* (SDN)) dans un contexte de santé mo-

bile. Notre solution de m-santé est capable de collecter et d'analyser les données physiologiques des utilisateurs afin de détecter la présence des symptômes d'un virus en un temps restreint et empêcher sa propagation. En effet, l'intégration de NFN dans une architecture basée sur SDN a pour but de minimiser le temps de réponse total et d'offrir une vue globale de l'architecture ainsi qu'une gestion dynamique. Étant donné que nous manipulons des données hautement confidentielles et personnelles, nous mettons en place un protocole de sécurité qui se base sur l'authentification anonyme et qui assure la confidentialité et le respect de la vie privée des utilisateurs en établissant un processus d'enregistrement et d'authentification.

En établissant une analyse informelle du schéma de sécurité, nous avons vérifié les fonctionnalités que le protocole de sécurité proposé réalise afin de protéger la vie privée des utilisateurs. À travers des scénarios de simulation, les résultats obtenus montrent que notre solution surpasse les performances d'un scénario de comparaison établi dans l'informatique en périphérie et réalise d'excellents résultats en termes de temps de réponse moyen. En outre, une évaluation analytique réalisée a montré que les résultats numériques et simulés en ce qui concerne le temps de latence global se correspondent.

Mots clés : Réseau des fonctions nommées, Réseau défini par logiciel, Réseau centré sur l'information, Infonuagique, Informatique en périphérie, Sécurité, Vie privée, Santé mobile

CHAPITRE I

INTRODUCTION GÉNÉRALE

1.1 Contexte

Le phénomène de l'Internet des Objets (IdO) a révolutionné le monde de l'Internet tout en apportant un confort et une facilité d'utilisation à la société. Certes, l'apparition des pandémies n'a fait qu'augmenter le nombre de recherches effectuées dans le but d'améliorer les services de santé tout en utilisant l'IdO. Par exemple, pendant la pandémie de Covid-19, une augmentation de 65% a été remarquée pour le nombre de téléchargements des applications de santé mobile (m-santé) (Statista, 2021a).

En utilisant des capteurs corporels personnels et des appareils médicaux mobiles, les applications de m-santé permettent l'analyse et la surveillance automatique et à distance de l'état de santé des patients. L'objectif de ces applications consiste à améliorer les soins fournis, réduire les risques de contagion ainsi que diminuer les coûts et les frais liés aux honoraires des médecins (Hu et Bai, 2014; Farahani *et al.*, 2018).

L'explosion de l'utilisation des applications de m-santé dans le but de capturer, analyser et enregistrer des informations médicales personnelles (ex. taux d'oxygénation, température, toux, fréquence cardiaque, taux d'insuline, etc.) a engendré la génération d'un volume de données énorme demandant à être filtré, évalué,

traité et stocké en temps réel afin de prendre des décisions rapides et fiables.

En effet, le contexte de m-santé exige que les décisions soient prises en un temps opportun tout en assurant la sécurité, la confidentialité et le respect de la vie privée des utilisateurs (Gama *et al.*, 2008). Toutefois, ces exigences ainsi que le volume de données énorme généré par les appareils IdO connectés mettent en question l'efficacité des centres de données de l'infonuagique (en anglais, *cloud computing*) et de l'informatique en périphérie (en anglais, *edge computing*) qui se distinguent par leur grande capacité de calcul et de stockage pour le traitement des données massives.

Face au besoin d'offrir un temps de latence faible pour les applications sensibles au délai, le réseau des fonctions nommées (en anglais, *Named Function Networking* (NFN)) a été récemment proposé dans la littérature (Tschudin et Sifalakis, 2014; Sifalakis *et al.*, 2014). Ce nouveau paradigme vise à utiliser les équipements de l'infrastructure réseau comme les routeurs et les commutateurs à des fins de calcul et de traitement pour satisfaire les exigences et la qualité de service (en anglais, (en anglais, *Quality of Service* (QoS)) attendues par les utilisateurs finaux. En effet, NFN permet d'exécuter des fonctions nommées à l'intérieur du réseau et offre une mise en cache des résultats et des fonctions aussi dans le réseau. De ce fait, NFN permet d'améliorer la stabilité et les performances en termes de latence et de rapidité d'accès.

1.2 Motivation

La déclaration d'une pandémie par l'Organisation mondiale de la santé (OMS) engendre un blocage et une perturbation du rythme de vie humaine de plusieurs manières. Les systèmes de santé sont par conséquent surchargés et paralysés devant le taux de contagion et de décès gigantesque. De plus, des perturbations économiques, sociales, culturelles et éducatives importantes sont provoquées par

les pandémies entrainant une récession et une crise mondiale soulevant une question sur la vie et la civilisation humaine. Les préoccupations primordiales lors de l'émergence d'une pandémie concernent généralement la mise en place des gestes barrières (ex. distanciation physique, port du masque, lavage des mains, etc.) et des règles de confinement dans le but de lutter ou du moins minimiser le nombre d'infections et de ralentir la contagion. Néanmoins, le respect rigoureux de ces mesures préventives doit encore être assuré, même avec le développement des vaccins.

Dans le contexte des pandémies où l'explosion de l'utilisation des applications m-santé requiert une réduction de délai de réponse ainsi qu'une amélioration de la fiabilité des résultats, nous proposons l'utilisation de NFN et du réseau défini par logiciel (en anglais, *Software Defined Networking* (SDN)) dans un contexte de m-santé. Notre proposition permet la collecte et l'analyse des données physiologiques générées en temps réel par les téléphones intelligents des utilisateurs et de détecter les cas suspects potentiellement porteurs de virus afin d'éviter la contagion. En outre, elle assure aussi la sécurité et respecte la vie privée des utilisateurs en établissant un protocole de sécurité avant de commencer la transmission des données physiologiques. Son objectif consiste à :

— Assurer plusieurs exigences de sécurité telles que l'authentification mutuelle, l'anonymat, la confidentialité, etc.

— Transmettre et analyser les données physiologiques des utilisateurs dans le but de détecter les cas suspects porteurs de virus tout en assurant un temps de réponse faible.

— Envoyer les données physiologiques des personnes suspectes vers le réseau hospitalier approprié pour une analyse approfondie du dossier médical afin de prendre des décisions fiables.

1.3 Problématique

Dans le but de reprendre le rythme normal de la vie suite à une pandémie et d'éviter les infections à la chaine ainsi que le non-respect des règles et des gestes barrières, nous mettons en place une solution de m-santé permettant l'analyse et la détection des symptômes de virus chez les individus et la prise de décision afin de prévenir la propagation.
Cette solution se base sur deux exigences primordiales : (i) minimiser le temps de réponse et (ii) assurer la sécurité, la confidentialité et le respect de la vie privée des utilisateurs. Pour pouvoir satisfaire les besoins des utilisateurs en termes de temps de réponse, il convient de bien choisir la technologie de déploiement de la solution qui permettra d'offrir un temps de latence faible. Compte tenu des informations personnelles sensibles qui peuvent être transmises dans le réseau en utilisant les applications de m-santé, l'établissement d'un protocole de sécurité robuste face aux : menaces, failles et attaques est primordial.

Notre choix de technologie s'est reposé sur NFN compte tenu de ses multiples avantages en termes de réduction de trafic et de temps de réponse, ainsi que de récupération rapide des données et des résultats, vu qu'il utilise les périphériques réseau pour le traitement et le calcul des fonctions. NFN met en place aussi des stratégies de transfert afin de trouver l'emplacement optimal pour l'exécution d'une fonction nommée, et qui sont utilisées dans le but d'améliorer la QoS et la qualité d'expérience de l'utilisateur (en anglais, *Quality of Experience* (QoE)) (Tschudin et Sifalakis, 2014; Sifalakis *et al.*, 2014).

Bien que NFN présente de nombreux avantages, plusieurs défis sont à relever, en particulier ceux de la flexibilité et de la gestion dynamique qui sont devenues des exigences strictes pendant les dernières années. Contrairement à l'infonuagique et l'informatique en périphérie qui utilisent largement le SDN (Nunes *et al.*, 2014),

aucune proposition n'existe à ce jour pour l'intégration de NFN dans une architecture SDN, ce qui peut par conséquent affecter la QoS particulièrement pour les systèmes médicaux à distance qui ne tolèrent pas le retard. En effet, SDN vise à séparer le plan de contrôle du plan de données dans le but de dissocier les opérations complexes de contrôle et de gestion dynamique des équipements réseau. En divisant l'architecture réseau en trois couches : applications, contrôle et données (Kreutz *et al.*, 2014), SDN offre une vue globale et centralisée de l'architecture à partir du contrôleur disponible dans la couche de contrôle. L'administrateur bénéficie cependant d'une vue globale de sa topologie réseau et des statistiques des flux. De plus, il peut configurer, surveiller et mettre en place des règles d'équilibrage de charge et de sécurité dans la couche application (Nunes *et al.*, 2014; Braun et Menth, 2014; Goransson *et al.*, 2016).

1.4 Objectifs

Lors d'une pandémie, et après de nombreux mois de confinement, les préoccupations gouvernementales concernent généralement le déconfinement et le retour du rythme normal de la vie. Toutefois, la vigilance doit rester de mise avec le port du masque et le respect des gestes barrières pour les espaces clos achalandés ou mal aérés. En effet, les portes d'accès de ces endroits constituent un potentiel goulot d'étranglement propice à une infection à la chaine.

Dans ce mémoire, nous étudions la mise en place d'une solution d'analyse et de détection m-santé basée sur NFN et sur SDN dans le but de tirer parti des deux approches. L'objectif de ce projet de maîtrise consiste à ce que chaque personne voulant accéder à un endroit public clos puisse utiliser une solution d'analyse des données physiologiques générées à partir des téléphones intelligents dans le but de détecter les cas suspects et d'éviter la contamination.
Cette solution met de l'avant deux objectifs importants, soient (i) assurer la sé-

curité des communications dans le but de garantir la sécurité et le respect de la vie privée des utilisateurs et (ii) minimiser le temps de réponse afin d'éviter les files d'attente devant les portes d'entrée. Notre premier objectif est assuré en adaptant un protocole d'authentification mutuelle pour le problème étudié. Une analyse informelle est présentée pour essayer de démontrer que le protocole satisfait plusieurs exigences de sécurité. Et, le deuxième objectif est atteint en utilisant une combinaison gagnante entre les principes des réseaux de fonctions nommées et ceux des réseaux définis par logiciel. À travers plusieurs simulations que nous avons effectuées, les résultats ont démontré la supériorité de cette combinaison par rapport à un modèle simple basé l'informatique en périphérie.

1.5 Contributions

Ce mémoire apporte les contributions suivantes :

- La description d'un cas d'utilisation permettant d'analyser les données médicales dans le but de détecter les cas suspects potentiellement porteurs du virus.
- Une adaptation d'un protocole de sécurité à partir du travail de (Alzahrani *et al.*, 2020) permettant d'assurer plusieurs exigences de sécurité telles que l'authentification mutuelle, l'anonymat, la confidentialité,etc.
- Une proposition d'une solution permettant l'utilisation de NFN dans une architecture basée sur SDN dans le but de minimiser le temps de latence et de bénéficier d'une vue globale de notre architecture.
- Une analyse des fonctionnalités du schéma de sécurité qu'il est capable d'assurer.
- Une évaluation des performances de notre solution en se basant sur des scénarios de simulation ainsi qu'une évaluation analytique.

1.6 Plan du mémoire

Ce mémoire se compose de six chapitres et s'organise comme suit :

— À la suite de ce premier chapitre, nous allons présenter dans le deuxième chapitre, le concept des applications IdO. Plus précisément, nous allons définir l'IdO et son utilisation dans le domaine médical. Ensuite, nous nous focalisons sur les applications de m-santé et leur exécution dans l'infonuagique et l'informatique en périphérie.

— Ensuite, dans le chapitre 3, nous présentons dans un premier temps les concepts liés au traitement dans le réseau (en anglais, *In-Network Computing* (INC)) et au réseau centré sur l'information (en anglais, *Information Centric Networking* (ICN)) et ses architectures existantes. Ensuite, nous définissons le NFN en faisant le point sur ses composantes principales.

— Dans le chapitre 4, nous introduisons le SDN et son architecture. Puis, nous nous intéressons au protocole OpenFlow et nous définissons ses multiples avantages.

— Le chapitre 5 sera consacré à l'étude et à la mise en place de notre solution. Nous commençons ce chapitre par décrire notre modèle de système. Ensuite nous présentons nos solutions proposées, à savoir un schéma de sécurité permettant l'authentification des utilisateurs, et une solution d'intégration de NFN dans une architecture basée sur SDN. Finalement, nous achevons ce chapitre par mettre en place une analyse informelle des fonctionnalités de sécurité assurées par notre protocole.

— Dans le dernier chapitre, nous évaluons les performances de notre solution en termes de temps de réponse global et de latence en effectuant des simulations basées sur plusieurs scénarios. Une modélisation par file d'attente des équipements du système est ensuite présentée, menant enfin à une évaluation

de notre solution.

— Enfin, une conclusion générale mettra un terme à ce mémoire.

CHAPITRE II

LES APPLICATIONS IDO

2.1 Introduction

Afin de mieux cerner l'étendue de notre problématique, nous présentons dans ce chapitre les concepts liés à l'IdO et à l'IdO médical (M-IdO). Nous présentons ensuite les défis majeurs rencontrés par les applications M-IdO. Finalement, nous définissons l'infonuagique et l'informatique en périphérie, leurs architectures et leurs avantages et nous mettons en évidence leurs limitations rencontrées par les applications M-IdO.

2.2 Internet des Objets

Les innovations technologiques ont favorisé l'émergence de milliards d'appareils, qui ont permis l'interconnexion des personnes et des objets à travers Internet. Cet écosystème cyberphysique appelé IdO (Gubbi *et al.*, 2013; Li *et al.*, 2015b) a joué un rôle primordial au fil des années du fait qu'il a facilité l'accès à l'information et aux services à travers le monde. En utilisant des milliards de capteurs et d'appareils hétérogènes qui peuvent être limités en ressources, puissants ou virtualisés, l'IdO permet l'interaction, la communication, la collecte et l'échange des données via une infrastructure de communication connectée à travers des milliards de nœuds IdO.

L'IdO est utilisé pour créer un réseau d'objets intelligents qui coopèrent à travers des applications dans le but d'automatiser des tâches et de fournir des services précis aux utilisateurs. Bien que ces applications soient différentes dans leur forme et leurs objectifs, elles partagent un principe commun, à savoir optimiser notre environnement et notre quotidien. Des exemples notables du IdO sont : les transports et la logistique, l'agriculture, la santé, la réalité virtuelle et la réalité augmentée, etc.

L'utilisation massive des applications IdO a augmenté de façon considérable le nombre des objets connectés à Internet. En effet, en 2003, il y avait environ 500 millions de périphériques connectés à Internet (DuBravac et Ratti, 2015). En 2015, ce nombre est passé à 15 milliards et, en 2025, il dépassera 75 milliards de périphériques connectés à Internet (Statista, 2021b). Ce nombre énorme d'appareils et de capteurs IdO connectés à Internet n'a fait qu'accroître le volume des données générées, qui nécessitent dans certains scénarios d'être traitées en temps réel. Par conséquent, les temps de réponse courts ne sont pas envisageables, comme pour le cas du M-IdO.

2.3 Utilisation dans le domaine médical

La santé représente un domaine d'application important pour l'IdO qui permet : (i) d'améliorer la qualité de vie, l'expérience utilisateur et la QoS, (ii) de gérer des maladies en temps réel (iii) et de réduire les coûts liés aux honoraires des médecins (Thakar et Pandya, 2017; Javdani et Kashanian, 2018). Le M-IdO se base principalement sur l'utilisation d'un certain nombre de capteurs et d'appareils médicaux qui collectent les informations de santé (température, glycémie, pression artérielle, fréquence cardiaque, etc.). Étant donné que les applications M-IdO utilisent des appareils ayant des limitations de stockage et de puissance de calcul, elles envoient les grandes quantités de données générées aux centres de

données de l'infonuagique ou à l'informatique en périphérie pour réaliser les tâches d'analyse, de traitement et de prise de décisions (Masip-Bruin *et al.*, 2018). Par conséquent, l'état de santé d'un patient peut être surveillé et analysé à distance. Le M-IdO offre plusieurs services selon les besoins de l'utilisateur, à l'instar de (Zhao *et al.*, 2011; Xu *et al.*, 2014) :

— L'assistance de vie qui propose une assistance médicale pour les personnes âgées et les personnes ayant une déficience en les surveillant dans leurs domiciles, et en émettant des notifications et des alarmes en cas d'urgence ou de rappel de prise de médicament.

— L'analyse de la santé mentale (De la Torre Díez *et al.*, 2019; Gutierrez *et al.*, 2021) qui permet de surveiller l'évolution d'une maladie mentale en analysant des données relatives au comportement, à la psychologie et aux interactions sociales.

— La m-santé qui offre des services axés sur l'acquisition et la transmission des informations médicales et physiologiques pour fournir des services de santé. En transmettant ces données sur Internet pour un traitement en temps réel, l'état de santé d'un patient peut être analysé et surveillé automatiquement et à distance. En effet, le développement et la mise en place des solutions m-santé offrent une approche efficace pour la prévention et la détection précoce de plusieurs maladies. Cependant, l'utilisation du m-santé joue un rôle crucial pour lutter contre la propagation des maladies et des virus qui génèrent des scénarios d'urgence sanitaire comme les pandémies (Dong et Yao, 2021).

2.4 Les défis des applications de m-santé

Avec l'explosion du grand volume de données produites par les dispositifs IdO, CISCO prévoit à ce que nous atteindrons 2,3 Zo de données générées par an en

2020 par ces appareils (Networking, 2016). Par conséquent, les applications M-IdO et principalement celles du m-santé font face à plusieurs défis dont nous citons (Ksentini *et al.*, 2021) :

— La latence : le maintien d'une faible latence représente une exigence clé pour les scénarios d'urgence qui nécessitent une intervention rapide ou à temps réel.

— La fiabilité des résultats : Une solution de m-santé doit garantir la fiabilité des résultats générés puisque la vie d'une personne ou d'un groupe de personne peut être en danger. Cependant, étant donné que ces applications sont basées sur les prédictions à partir de signaux physiologiques, nous ne pouvons pas garantir d'avoir une fiabilité de 100% en raison des variations physiologiques. De plus, plusieurs symptômes peuvent être relatifs à différentes maladies, ce qui peut par conséquent engendrer des faux positifs.

— La sécurité : En envoyant les informations personnelles sur le réseau Internet, on expose les données à des pirates ou à des logiciels malveillants, ce qui peut entraîner des conséquences majeures. Assurer la sécurité des informations et des données représente une exigence primordiale qui doit être prise en compte pour maintenir la résilience du système face aux menaces de sécurité.

2.4.1 Traitement dans l'infonuagique

Pour effectuer l'analyse et le traitement des données physiologiques, les dispositifs IdO de santé envoient la quantité de données vers Internet en raison de leurs ressources limitées. Cependant, l'infonuagique représente un concept important et notable de prestation de calcul et de stockage ayant des ressources informatiques configurables (Vilela *et al.*, 2020). L'infonuagique se distingue par son grand nombre de centres de données et offre une grande capacité de calcul et de stockage permettant l'exécution des applications des utilisateurs distants.

Cependant, la distance entre l'utilisateur final et les centres de données utilisés par l'infonuagique représente une source non négligeable de délais. De plus, avec le volume de données énorme transmis vers l'infonuagique et l'encombrement du réseau, l'exécution des services de m-santé sensibles aux retards sera inadaptée en raison d'un problème d'explosion du temps de latence dû à la surcharge de la couche infonuagique (Dang *et al.*, 2019; Ksentini *et al.*, 2021). Par conséquent, dans des situations d'urgence, les retards peuvent conduire à des prises de décisions inexactes et peuvent mettre la vie d'une personne à risque.

D'un autre côté, les informations médicales et de santé sont hautement confidentielles, et leur exposition peut aussi mettre la vie privée des personnes en danger. En effet, l'infonuagique représente une cible idéale pour les attaquants en raison de sa faiblesse en termes de sécurité, de confidentialité et de contrôle d'accès vu qu'elle regroupe plusieurs formes de données issues de multiples utilisateurs dans une seule et unique infrastructure. Des problèmes d'écoute clandestine, d'usurpation d'identité, de contrôle d'accès, d'authentification, etc. sont évidemment connus dans l'infonuagique en raison de l'utilisation de la communication sans fil (Ahuja *et al.*, 2012; Darwish *et al.*, 2019).

2.4.2 Traitement dans l'informatique en périphérie

Dans le but de résoudre certains défis rencontrés par l'infonuagique, l'informatique en périphérie a été mise en place. Elle se caractérise comme étant une plateforme décentralisée pour le traitement et le stockage des données. L'informatique en périphérie étend la capacité de l'infonuagique jusqu'à la périphérie du réseau en intégrant une couche intermédiaire qui contient des périphériques distribués et hétérogènes déployés à proximité des utilisateurs finaux (Yu *et al.*, 2017). L'architecture de l'informatique en périphérie est présentée dans la figure 2.1 et elle est composée de trois couches (Lyu *et al.*, 2018) :

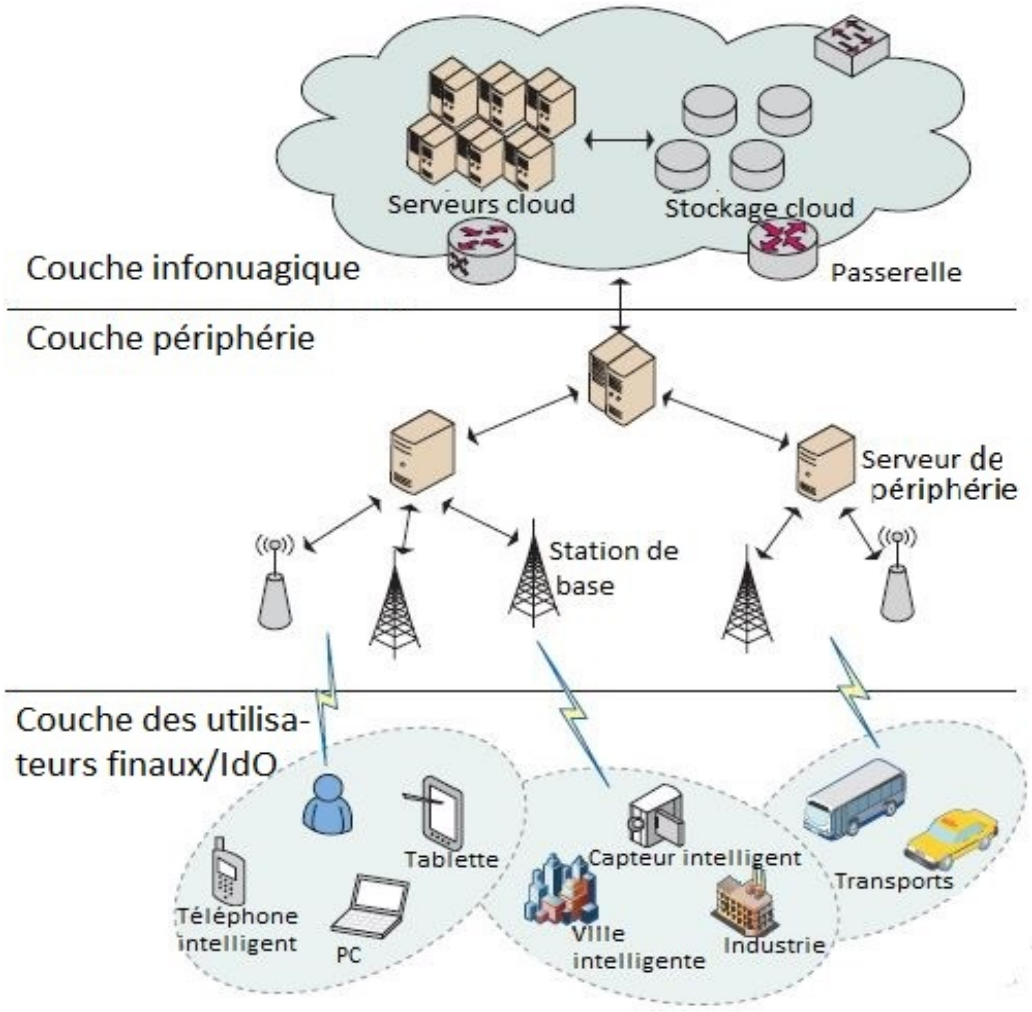

Figure 2.1: Architecture à trois couches : infonuagique, périphérie et IdO (Lyu *et al.*, 2018)

— La couche des utilisateurs finaux/IdO : elle contient tous les objets IdO connectés comme les capteurs, les téléphones intelligents, les montres intelligentes, les véhicules, etc.

— La couche périphérie : elle se compose de plusieurs périphériques ayant des caractéristiques différentes en termes de puissances de calcul et de stockage. Ces périphériques sont disponibles à proximité des utilisateurs finaux qui

envoient leurs données pour le traitement.

— La couche infonuagique : elle se distingue par son grand nombre de serveurs disponibles dans les centres de données et ayant une très grande capacité de calcul et de stockage. La couche de périphérie achemine les données vers la couche de l'infonuagique en cas de besoin.

L'exécution des tâches IdO dans les périphériques disponibles à proximité des utilisateurs a permis par conséquent d'améliorer l'efficacité et les performances et de réduire le temps de latence et le volume de données envoyées. Les requêtes et les données sont cependant acheminées vers la couche périphérie minimisant ainsi les retards potentiels des centres de données. En outre, la fiabilité est améliorée du fait que le placement des ressources de traitement à proximité des utilisateurs finaux évite que le système soit dépendant de la disponibilité de la connexion réseau. À la différence du modèle de l'infonuagique, la sécurité et la confidentialité sont améliorées étant donné que l'informatique en périphérie réduit le nombre de sauts que les données des utilisateurs doivent parcourir, ce qui par conséquent atténue le risque d'exposer les données aux attaquants (Goyal *et al.*, 2020; Vilela *et al.*, 2020).

Dans le contexte du m-santé, l'informatique en périphérie souffre tout de même de certaines limitations. En effet, l'allocation de la charge de travail et aussi l'encombrement du réseau de périphérie par les appareils IdO et les données volumineuses qui sont générées représentent des problèmes cruciaux. Ayant des capacités de calcul différentes dans chaque couche, l'informatique en périphérie doit gérer ses périphériques afin d'équilibrer la charge et d'éviter l'augmentation du temps de latence. En raison des délais variables qu'une application m-santé peut subir, la fiabilité des résultats n'est pas assurée, ce qui peut par conséquent réduire la QoS et la QoE (Shi *et al.*, 2016; Hartmann *et al.*, 2019).

Garantir la sécurité, la confidentialité et la vie privée des utilisateurs dans l'informatique en périphérie n'est pas aussi simple en raison du grand nombre de menaces et d'attaquants qui peuvent être présents entre la couche de périphérie et la couche des utilisateurs finaux. Les algorithmes de calcul peuvent également faire l'objet de plusieurs attaques paralysant ainsi le système de santé et permettant à des intrus d'utiliser des données à des fins illégales. L'utilisation d'algorithmes cryptographiques robustes et puissants ajoutera une surcharge en termes de délai de réponse ce qui peut affecter négativement la qualité de service attendue (Shi *et al.*, 2016; Abdellatif *et al.*, 2019).

2.5 Conclusion

Dans ce chapitre, nous avons présenté les notions de base associées à l'IdO et aux applications M-IdO. Ensuite, nous avons présenté les défis rencontrés par les applications M-IdO en raison de l'explosion du volume des données générées par les applications IdO. Nous avons aussi défini l'infonuagique et l'informatique en périphérie et leurs architectures. Finalement, nous avons présenté les avantages et les limitations de chaque technologie dans le contexte du M-IdO.

CHAPITRE III

RÉSEAU DES FONCTIONS NOMMÉES

3.1 Introduction

Dans ce chapitre, nous commençons dans un premier temps par présenter le traitement dans le réseau et ses avantages. Ensuite, nous définissons les concepts liés aux réseaux centrés sur l'information, leurs caractéristiques ainsi que leurs différentes architectures. Finalement, nous nous concentrons sur le réseau des fonctions nommées et ses principales composantes à savoir le flux de travail et les stratégies de transfert.

3.2 Traitement dans le réseau (INC)

Afin d'alléger la charge de l'infonuagique et de l'informatique en périphérie, l'INC a été proposé comme un nouveau paradigme qui vise à utiliser la couche réseau pour effectuer le traitement des fonctions. En effet, l'INC est un nouveau domaine de recherche en pleine émergence depuis quelques années. Le traitement dans le réseau consiste à se servir des équipements réseau déjà existants dans l'infrastructure réseau qui sont habituellement utilisés pour le transfert de trafic comme les routeurs et les commutateurs à des fins de calcul et de traitement (Sapio *et al.*, 2017; Tokusashi *et al.*, 2019).

Étant donné que l'INC se caractérise par le traitement au sein du réseau, les

requêtes sont traitées localement sans atteindre les couches distantes de l'infonuagique et de l'informatique en périphérie. De ce fait, l'INC permet par conséquent de réduire la latence, de minimiser le trafic ainsi que la charge du réseau et de diminuer les coûts d'utilisation des ressources de l'infonuagique et de l'informatique en périphérie (Sapio *et al.*, 2017; Tokusashi *et al.*, 2019).

3.3 Réseau centré sur l'information (ICN)

Le réseau centré sur l'information est un paradigme mis en place dans le but d'offrir un service d'infrastructure réseau plus adéquat à l'Internet du futur et aux attentes des utilisateurs. En effet, il procure plus de robustesse face aux pannes et aux perturbations engendrées par les données massives qui sont en train de saturer l'architecture actuelle de l'Internet. Il offre aussi l'opportunité de mieux représenter et exprimer les besoins et les attentes des utilisateurs particulièrement dans les scénarios IdO avec des exigences strictes en termes de QoS (Ahlgren *et al.*, 2012; Xylomenos *et al.*, 2014).

Différent de l'approche TCP/IP qui est centrée sur l'hôte, ICN est plutôt centré sur le contenu et s'intéresse particulièrement sur QUOI échanger plutôt qu'avec QUI échanger. Ce nouveau paradigme se base sur une approche de dénomination des objets qui sert à utiliser les noms au lieu des adresses IP pour la récupération du contenu. Dans une architecture ICN, un consommateur qui est en mouvement, peut tout simplement continuer à envoyer les demandes de récupération de contenu vers une nouvelle source supprimant ainsi le problème de gestion de connexion de bout en bout. L'ICN se base sur un ensemble de nouvelles fonctionnalités orientées contenu dans le réseau comme la dénomination des objets, le routage de contenu, la mise en cache et la sécurité basée sur les données. En effet, c'est une architecture prometteuse pour la prise en charge des différents scénarios IdO (Amadeo *et al.*, 2016; Bracciale *et al.*, 2019).

3.3.1 Concepts de base

ICN se base sur des approches innovantes qui lui permettent d'être un paradigme révolutionnaire. En effet, il attribue des noms aux contenus au sein de son réseau pour permettre leurs récupérations. De plus, ICN se base sur un principe de récupération pilotée par le récepteur, dans lequel il utilise pour le routage des requêtes un service de résolution de noms (en anglais, *Name Resolution Service* (NRS)) avec un routage basé sur le nom (en anglais, *Name Based Routing* (NBR)). Finalement, ICN adopte la mise en cache dans le réseau dans le but d'accélérer l'accès aux données et réduire le trafic.

- **Dénomination :** Pour récupérer un contenu, ICN se base sur une nouvelle approche qui se base sur les objets de données nommés (en anglais, *Named Data Objects* (NDO)). Ce concept consiste à nommer les objets qui peuvent être des pages web, des vidéos, des documents, etc. avec un nom unique et permanent. Les NDOs servent en effet à récupérer les contenus à travers leurs noms au lieu d'utiliser les adresses IP et doivent conserver leurs noms ainsi que leurs identités indépendamment de l'emplacement et de la méthode de communication et de stockage (Ahlgren *et al.*, 2012; Amadeo *et al.*, 2016). En outre, les noms des NDOs peuvent être hiérarchiques ou plats conformément à l'architecture ICN utilisée. Les noms hiérarchiques sont compréhensibles par l'humain et possèdent une structure semblable aux « Uniform Resource Locator (URL) » avec des longueurs variables où chaque chaîne de caractères est séparée par un « / ». En utilisant cette sémantique, chaque contenu peut être divisé en plusieurs morceaux portant des versions. Par contre, les noms plats comportent des identifiants de longueur fixe sans structure sémantique. De plus, ils ne sont pas lisibles par l'humain du fait qu'ils sont autocertifiés en établissant une liaison directe de hachage

de contenu à l'identifiant lui-même. Par conséquent, l'intégrité des données dans l'espace de noms plats est vérifiée sans l'utilisation d'une infrastructure à clé publique (en anglais, *Public Key Infrastrucutre* (PKI)) (Detti *et al.*, 2013; Dutta *et al.*, 2021).

- **Récupération pilotée par le récepteur :** Le routage consiste en un processus d'acheminement des demandes des consommateurs vers le fournisseur des données à travers le réseau, et par la suite la transmission de ces dernières vers le consommateur. Dans ICN, le traitement et la gestion des demandes de récupération de contenu s'appuient sur deux approches (Ahlgren *et al.*, 2012; Amadeo *et al.*, 2016).

  La première approche consiste à utiliser NRS pour stocker une liaison entre un NDO et le localisateur qui peut être un producteur ou un nœud de cache qui conserve une copie du contenu souhaité. Étant donné que chaque NRS couvre une zone spécifique, le consommateur doit rediriger son message vers un NRS approprié à la récupération des informations demandées. Lors de cette phase, l'utilisation du NBR est possible afin de faciliter la récupération des informations de routage. La dernière étape de cette approche consiste à acheminer les données vers le consommateur en suivant le chemin inverse.

  La deuxième approche consiste à envoyer directement les demandes vers les fournisseurs ou les nœuds possédant une copie sans qu'il soit obligatoire de passer par un localisateur pour la résolution des noms. Pour accomplir cette approche, chaque nœud entre le consommateur et le fournisseur doit connaître une partie des informations de routage. À la réception de la requête, le fournisseur envoie les données au consommateur en utilisant le chemin inverse.

- **Mise en cache dans le réseau :** Les architectures ICN disposent d'une fonctionnalité primordiale qui consiste à mettre en cache dans le réseau

des copies du contenu. Dans les architectures ICN, les nœuds possèdent des caches qui ont pour but de satisfaire les demandes des consommateurs pour un contenu. Le nœud disponible le plus proche détenant une copie du contenu demandé répondra alors à la requête du consommateur. La mise en cache dans le réseau permet par conséquent de réduire le trafic qui circule dans le réseau, d'accélérer la récupération des données et d'atténuer l'accès au serveur (Zhang *et al.*, 2013).

Afin d'optimiser de multiples mesures de performances telles que le taux d'accès au cache, la distance d'accès au cache et le coût opérationnel et aussi éviter le gaspillage d'énergie et la consommation des ressources, des politiques de décision de mise en cache et de remplacement sont mise en place qui prennent en charge des exigences telles que la popularité et la fraîcheur des données, etc (Ahlgren *et al.*, 2012; Zhang *et al.*, 2015).

- **Sécurité basée sur les données :** Dans le modèle TCP/IP, la sécurité est offerte en protégeant le canal de communication entre un client et un serveur en utilisant des protocoles de sécurité dans la couche de transport tel que « *Transport Layer Security (TLS)* ». Dans l'architecture ICN, la sécurité est fournie en sécurisant les données dans la couche réseau, visant ainsi à restreindre l'accès aux données à un utilisateur spécifique ou à un groupe d'utilisateurs. Ce schéma de sécurité basé sur le nom garantit que l'authentification et l'intégrité des données soient vérifiées localement en éliminant ainsi le besoin de faire confiance à un nœud intermédiaire (Amadeo *et al.*, 2016; Xu *et al.*, 2019).

3.3.2 Architectures existantes

De nombreuses architectures ont été proposées pour ICN telles que Data-Oriented Network Architecture (DONA), Publish-Subscribe Internet Routing Paradigm

(PSIRP), Network of Information (NetInf) et Named Data Networking (NDN). Nous définissons ces architectures en suivant leur apparition chronologique comme montré dans la figure 3.1.

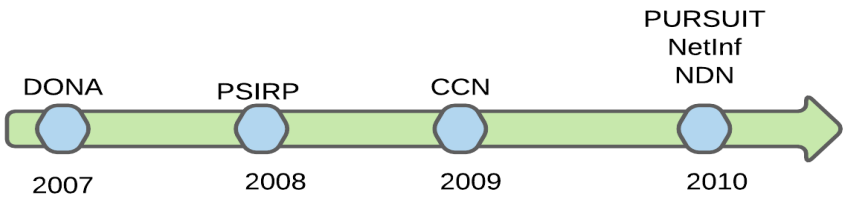

Figure 3.1: Chronologie des architectures ICN

- **Data Oriented Network Architecture (DONA) :**

  Elle est considérée comme étant parmi les premières architectures complètes prenant en charge le paradigme ICN. Dans l'architecture DONA (Koponen *et al.*, 2007), les nœuds autorisés à répondre aux demandes de contenu doivent s'enregistrer dans une infrastructure de résolution composée par un gestionnaire de résolution (en anglais, *Resolution Handler* (RH)). Ce dernier achemine les demandes de manière hiérarchique en essayant de trouver une copie du contenu la plus proche possible du consommateur. Ensuite, les paquets de données sont transmis vers le consommateur soit via le chemin RH inverse soit via un chemin direct. Les noms utilisés dans l'architecture DONA sont des noms plats rendant l'identification par un tiers difficile.

- **Publish-Subscribe Internet Routing Paradigm (PSIRP) :**

  PSIRP (Ahlgren *et al.*, 2012) se base sur un protocole de publication et d'abonnement et sur la paire « scope ID » et « rendez-vous ID » qui sont tous les deux des noms plats utilisés pour identifier les objets. Dans PSIRP, les nœuds rendez-vous sont utilisés pour établir la liaison entre les éditeurs

et les abonnés. En effet, un éditeur envoie un message de publication à un nœud de rendez-vous pour l'informer de la disponibilité d'une information, et l'abonné émet un message d'abonnement à un nœud rendez-vous pour établir la liaison. Les publications et les abonnements sont par la suite regroupés par un système de rendez-vous. PSIRP a aujourd'hui une version plus récente qui est Publish Subscribe Internet Technology (PURSUIT).

- **Network of Information (NetInf) :**

  L'architecture NetInf (Ahlgren *et al.*, 2012) utilise des noms plats pour les NDOs et propose deux modèles de récupération de données : soit à travers la résolution des noms, soit en utilisant le routage basé sur les noms. En effet, l'approche basée sur la résolution des noms consiste à ce que les sources publient les NDOs en enregistrant la liaison nom/localisateur dans un NRS. Ensuite, le récepteur envoie un paquet « Resolve » au NRS pour récupérer la copie des données à partir de la meilleure source disponible. En utilisant le routage basé sur les noms, les sources annoncent les informations de routage via un protocole de routage. Par conséquent, le récepteur peut envoyer une requête « GET » avec le NDO qui sera redirigé vers une copie disponible en s'appuyant sur le routage basé sur le nom. Aussitôt qu'une copie est trouvée, les données sont envoyées vers le récepteur en suivant le chemin inverse. L'utilisation des deux modèles offre l'avantage d'avoir une souplesse d'adaptation dans divers environnements réseau.

- **Named Data Networking (NDN) :**

  C'est le successeur de Content Centric Networking (CCN) (Jacobson *et al.*, 2009) qui est une architecture révolutionnaire dans le paradigme de l'ICN. Dans NDN (Zhang *et al.*, 2014), un consommateur envoie un paquet d'intérêt pour demander un contenu, qui lui sera renvoyé à travers un paquet de

données. Les noms utilisés dans les paquets d'intérêt et de données sont hiérarchiques et lisibles permettant d'identifier chaque élément de façon unique. Le nœud NDN se compose de trois tables qui sont représentées dans la figure 3.2 :

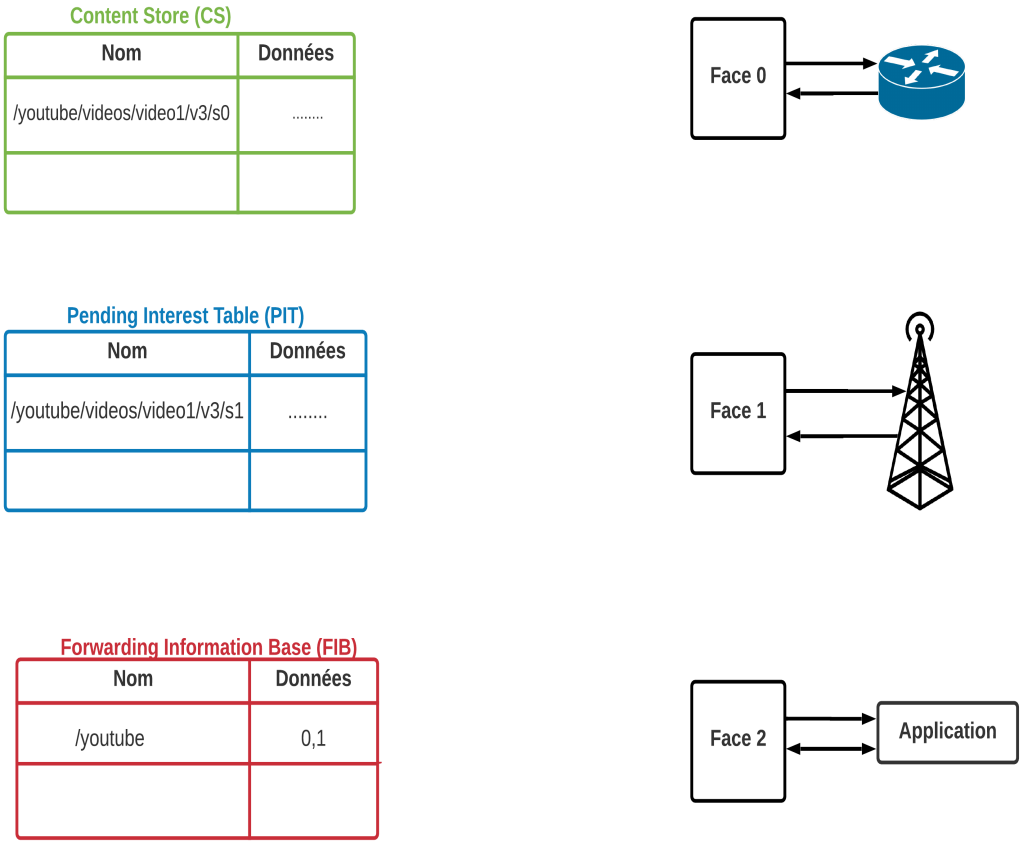

Figure 3.2: Composantes du nœud NDN

— *Content Store (CS)* : permet la mise en cache des paquets de données traversant le nœud. Le choix de la politique de décision de mise en cache repose essentiellement sur les ressources de stockage disponibles et le scénario utilisé. Néanmoins, quand l'espace de stockage est saturé et qu'une nouvelle donnée va être mise en cache, le NDN a recours à

une stratégie de remplacement par défaut visant à éliminer la donnée la moins utilisée récemment.

— *Pending Interest Tables (PIT)* : enregistre les paquets d'intérêts entrants non satisfaits . Lors de la réception d'un paquet de données, d'un accusé de réception négatif (NACK) ou si la durée de vie expire, le paquet d'intérêt est supprimé du PIT.

— *Forwarding Information Base (FIB)* : achemine les paquets d'intérêts vers les interfaces correspondantes en vérifiant le préfixe disponible dans le paquet. Dans FIB, deux interfaces sont disponibles dans chaque nœud NDN. La première interface est appelée « NetDeviceFace » et permet la communication avec les réseaux physiques. La deuxième « AppFace » et permet la communication avec les applications.

Quand un consommateur envoie un paquet d'intérêt, chaque nœud NDN vérifie si une copie est mise en cache dans son CS. Si une copie est repérée, le nœud envoie le paquet de données sur la même interface de réception du paquet d'intérêt. Sinon, le nœud cherche une correspondance dans le PIT. Si une entrée correspondante est disponible, cela indique qu'un paquet d'intérêt a été déjà envoyé et que le nœud attend le paquet de données. Par conséquent, le nœud met à jour le PIT et rejette la demande. Si aucune correspondance n'est trouvée dans le PIT, le nœud vérifie le FIB pour effectuer l'envoi du paquet d'intérêt vers une interface sortante. En cas d'échec, un NACK est renvoyé au consommateur et le paquet d'intérêt est rejeté.

La sécurité dans NDN est associée aux données en intégrant la signature numérique obligatoire qui couvre le nom, le contenu et les métadonnées. Cette signature permet d'établir la liaison sécurisée entre un nom et un contenu de chaque paquet de données. La vérification de la signature numérique permet par conséquent de garantir l'intégrité des données et d'assurer l'authentifi-

cation du producteur du fait qu'elle est calculée avec sa clé privée. De plus, l'utilisation des primitives cryptographiques comme les fonctions de hachage et le chiffrement des données garantissent à la fois le contrôle d'accès et la confidentialité. En effet, l'inspection des en-têtes des paquets dans NDN ne révèle aucune information sur le demandeur du contenu puisque les adresses sources et destinations ne sont pas utilisées. Par conséquent, la distribution de contenu basée sur les noms favorise la protection de la confidentialité et de la vie privée (Zhang *et al.*, 2010; Lounis *et al.*, 2012).

Le tableau 3.1 fournit un récapitulatif des caractéristiques des différentes architectures ICN (Xylomenos *et al.*, 2014).

Tableau 3.1: Résumé des caractéristiques des architectures ICN

	Dénomination		Routage		Noms lisibles	Sécurité	Mobilité
	Noms Plats	Noms hiérarchique	NRS	Basé sur les noms			
DONA	X			X		X Auto-certification des noms, PKI indépendant	X
PSIRP/ PURSUIT	X		X			X Auto-certification des noms, Authentification niveau paquet (PLA) pour les paquets individuels	X
NetInf	X		X	X		X Hachage de contenu ou de signature, PKI indépendant	X
NDN/ CCN		X		X	X	X Signature incluse dans le paquet, source de confiance externe	X

3.4 Réseau des fonctions nommées (NFN)

Afin de bénéficier des avantages de l'INC et de l'ICN, le réseau des fonctions nommées a été proposé en se basant sur l'architecture NDN. Outre l'accès aux données en utilisant des NDOs, le principe du NFN consiste à offrir la possibilité d'exécuter des fonctions nommées dans le réseau permettant ainsi une communi-

cation plus proche à l'utilisateur. En effet, les fonctions NFN sont indépendantes de l'emplacement offrant ainsi la possibilité aux fonctions de pouvoir être transférées et exécutées dans n'importe quel nœud NFN stockant le code de la fonction. NFN étend la sémantique de dénomination dans NDN en utilisant des expressions qui peuvent représenter un NDO, une fonction permettant l'exécution des NDOs ou une combinaison de noms et de fonctions (Tschudin et Sifalakis, 2014; Sifalakis *et al.*, 2014).

Le paradigme NFN se base sur deux composantes principales : le flux de travail qui permet d'orchestrer les appels des fonctions, et la stratégie de transfert qui prend la décision pour choisir l'emplacement de l'exécution de la fonction. En effet, dans NFN, le réseau se charge de trouver un emplacement optimal pour exécuter une fonction (Scherb *et al.*, 2019b). Toutefois, pour des raisons de sécurité ou de droits d'auteur, certaines fonctions sont épinglées sur les nœuds NFN dans le but d'empêcher leurs transferts sur le réseau, et à obliger la transmission des données vers la fonction épinglée (Scherb *et al.*, 2017).

3.4.1 Flux de travaux (Workflow)

NFN définit son flux de travail en utilisant le λ- calcul qui est une notation formelle des fonctions basée sur la logique mathématique. Trois opérations sont possibles dans une expression de λ- calcul (Scherb *et al.*, 2018) :

— Variable : v, où v est une variable

— Abstraction : (λ x.M), où M est un λ- calcul est x est une variable

— Application : (M N), où M et N sont des λ- calculs

Pour que le λ- calcul soit utilisé dans NFN, il a été étendu par une fonction « call » permettant d'appeler le code de fonction. En outre, l'appel de la fonction est suivi par des paramètres d'entrée qui peuvent être également un autre flux de travail comme dans l'exemple suivant (Scherb et Tschudin, 2018) : call/lib/func/data1

(call/lib2/func2/data2).

Toutefois, dans une architecture NDN, le traitement de ce flux de travail ne peut pas être réalisé étant donné qu'il ne correspond pas aux entrées du FIB. Cependant, le flux de travail a été modifié pour qu'il commence par un NDO interprétable en utilisant la correspondance du préfixe le plus long, par exemple :
/data1(λx.call/lib/func x(call/lib2/func2/data2))

3.4.2 Stratégies d'exécution dans NFN

La stratégie d'exécution NFN a pour but de trouver un emplacement optimal pour l'exécution d'une fonction NFN qui est caractérisée par sa distance de la source des données. Afin d'éviter la congestion des centres de données et de réduire le trafic dans le réseau, il est préférable d'exécuter les fonctions le plus près possible des données surtout si les données sont volumineuses (Scherb *et al.*, 2019a).

Le mécanisme d'orchestration de fonction par défaut dans NFN est « Find or Execute (FOX) » qui consiste à ce que le réseau envoie le paquet d'intérêt vers la source des données et simultanément une recherche d'un résultat mis en cache sur le même chemin est effectuée. Si la recherche du résultat mis en cache échoue, la source des données récupère la fonction et l'exécute. À l'exception des contraintes liées à la sécurité ou à des droits d'auteurs empêchant le transfert de la fonction vers la source des données, le réseau transmet le paquet d'intérêt à la source de la fonction pour chercher un résultat mis en cache ou bien exécuter la fonction (Tschudin et Sifalakis, 2014; Scherb et Tschudin, 2018).

Toutefois, une stratégie de transfert NFN doit suivre les caractéristiques des scénarios IdO en prenant en considération les contraintes des ressources et en permettant la génération des résultats dynamiques dans le temps. Par conséquent, la stratégie de transfert FoX devient inadéquate pour plusieurs scénarios IdO. Dans

ce contexte, et en prenant en considération les différentes caractéristiques des scénarios IdO, plusieurs stratégies d'exécution NFN ont été proposées dans le but d'améliorer la QoS attendue par l'utilisateur et de réduire le trafic dans le réseau.

Les réseaux IdO stationnaires :

Les réseaux IdO stationnaires, comme les maisons intelligentes, se caractérisent par des nœuds qui sont peu ou pas mobiles, un nombre élevé de capteurs ayant une faible puissance de calcul et de capacité de batterie. Avec plusieurs nœuds dans un réseau IdO stationnaire où chaque appareil génère une faible quantité de données, la stratégie FoX n'est pas recommandée étant donné la faible capacité et puissance des dispositifs pour l'analyse et l'interprétation des données émises. Cependant, si beaucoup de demandes sont exécutées en se basant sur FoX, les capacités du réseau et de calcul diminueront et le temps de traitement explosera (Scherb *et al.*, 2018). Par conséquent, plusieurs stratégies d'exécution ont été proposées dans la littérature pour ce type de scénarios telles que edgeFox, IoT-NCN et NDN-Fog.

La stratégie d'exécution EdgeFox (Scherb *et al.*, 2018) vise à adapter la stratégie d'exécution FoX pour qu'elle soit utilisée dans le réseau de périphérie où chaque nœud annonce la disponibilité des fonctions qu'il exécute dans le but de réduire la quantité de données transportées vers le réseau d'infrastructure (CORE). Deux situations sont considérées dans EdgeFox : (i) si un ou plusieurs nœuds de calcul sont disponibles, le calcul doit être exécuté sur l'un d'eux (ii) si aucun nœud de périphérie n'est disponible, un paquet d'intérêt est envoyé vers la source des données pour vérifier s'il existe un résultat mis en cache. Si cette étape échoue, le nœud exécutera localement la fonction (Scherb *et al.*, 2018; Amadeo *et al.*, 2019). Toutefois, compte tenu du temps d'exécution et de recherche d'un nœud de périphérie disponible trop long, la stratégie de transfert EdgeFox ne peut pas être adaptée pour les scénarios IdO sensibles à la latence.

La stratégie IoT-NCN (Amadeo *et al.*, 2019) vise à sélectionner l'exécuteur de la fonction dans le réseau de périphérie selon les exigences du service, les ressources de traitement disponibles et la proximité à la source des données. Les services IdO de référence étudié dans IoT-NCN peuvent être soit gourmands en données nécessitant une exécution proche des sources, soit des services gourmands en calcul qui vont être exécutés dans le réseau edge du fait qu'ils ont besoin d'avoir des ressources de calculs plus puissants. Cependant, le choix des exécuteurs restreint aux nœuds sur le chemin vers la source limite IoT-NCN étant donné qu'il est très probable d'envoyer un paquet d'intérêt vers l'infonuagique pour l'exécution.

NDN-Fog (Amadeo *et al.*, 2019) quant à elle, vise à exécuter les fonctions NFN dans le réseau de brouillard (en anglais, *Fog computing*) où chaque nœud du fog vérifie s'il possède un résultat mis en cache localement ou s'il peut être un exécuteur. Le choix du nœud exécuteur repose sur un appel d'offres du plus court temps de service fourni par un nœud fog qui est une estimation du temps de traitement de la fonction et du temps de la collecte des données IdO. Si aucun résultat n'est mis en cache et qu'aucun nœud fog n'est capable de satisfaire la demande, le paquet d'intérêt est envoyé vers l'infonuagique (Amadeo *et al.*, 2019). Toutefois, NDN-Fog augmente le trafic dans le réseau fog étant donné le grand nombre des paquets d'intérêts envoyés à tous les nœuds fog ce qui peut par conséquent encombrer le réseau et augmenter les délais de réponse.

Les réseaux IdO hautement mobiles :

Les réseaux IdO hautement mobiles comme les voitures connectées se distinguent par un haut degré de mobilité, une bande passante faible ou limitée ainsi qu'un temps de réponse élevée. Dans ce type de réseau IdO, les stratégies d'exécution NFN doivent se concentrer sur une livraison efficiente des données sensibles au délai (Amadeo *et al.*, 2019). Étant donné le haut degré de déplacement dans

les réseaux IdO hautement mobiles, l'utilisation de la stratégie d'exécution FoX engendre une perte de connexion avec les nœuds qui exécutent les fonctions. Par conséquent, aussitôt qu'ils seront connectés à un autre nœud NFN, les demandes sont renvoyées afin de réexécuter la fonction dès le début, ce qui peut être contre-productif.

La stratégie d'exécution Find and Execute (FaX) (Grewe *et al.*, 2020) consiste à commencer l'exécution d'une fonction dans le premier nœud de périphérie disponible et chercher simultanément un résultat mit en cache. Si le réseau arrive à trouver un résultat mis en cache avant la fin de l'exécution de la fonction, le traitement sera arrêté et le résultat sera envoyé au demandeur. Si la recherche d'un résultat mis en cache échoue, alors un résultat sera livré au demandeur (Amadeo *et al.*, 2019; Grewe *et al.*, 2020). Bien que FaX augmente les possibilités de livraison de résultat et fonctionne convenablement avec les petits calculs, la probabilité de démarrer un traitement plusieurs fois dans le réseau pour les calculs qui nécessitent beaucoup de temps peut diminuer l'efficacité et encombrer le réseau.

La stratégie Find or Pull and Execute (FoPax) (Grewe *et al.*, 2020) est similaire à Fax du fait qu'elle démarre le traitement immédiatement. Pour la récupération des résultats calculés par les nœuds de périphérie, FoPaX utilise des messages R2C « Request to Compute » qui offrent la possibilité de récupérer les résultats intermédiaires des autres nœuds d'exécution (Amadeo *et al.*, 2019; Grewe *et al.*, 2020). Cependant, FoPaX résout le problème de FaX en ce qui concerne la perte de connexion avec les nœuds exécuteurs et permet aussi de réduire le temps de traitement global en se servant des messages R2C. Néanmoins, la probabilité de lancer une exécution plusieurs fois dans le réseau demeure possible dans FoPax ce qui peut par conséquent retarder la livraison de réponse.

Les réseaux IdO à évènements complexes :

Le traitement des évènements complexes (en anglais, *Complex Event Processing* (CEP)) présente une approche robuste pour les scénarios IdO nécessitant d'utiliser des opérations de corrélation, de filtrage et d'agrégation sur un ensemble de données pour fournir des évènements significatifs et des notifications en réponse à la requête du demandeur. Toutefois, utiliser les CEPs dans une architecture ICN est une tâche compliquée, et la stratégie d'exécution FoX ne présente pas de solution pour les gérer.

La stratégie Find, Split and Execute (FSX) (Scherb et Tschudin, 2018) a pour objectif de décomposer les calculs en sous-calculs indépendants et de choisir une stratégie d'exécution pour chaque sous-calcul. Ensuite, un paquet d'intérêt est envoyé vers un nœud de périphérie en se basant sur la stratégie EdgeFox qui doit décider si l'intégralité de la fonction va être exécutée localement ou bien adopter une décomposition du flux de travail qui consiste à calculer un ou plusieurs sous-calculs et envoyer le reste vers l'infonuagique (Scherb et Tschudin, 2018). En adoptant une exécution des fonctions de façon parallèle, FSX permet de réduire le trafic et de minimiser le temps de latence. Néanmoins, les services qui nécessitent un temps de réponse en temps réel peuvent être affectés par le temps de la prise de décision et aussi par l'envoi de la décision de l'exécution vers l'infonuagique.

L'architecture INetCEP (Luthra *et al.*, 2019) a été proposée dans le but de trouver une solution au traitement de flux de données périodiques qui doit être initié par le producteur afin d'éviter l'encombrement du réseau par les demandes envoyées par le consommateur. INetCEP propose de modifier l'architecture CCN en ajoutant un moteur CEP ainsi que d'autres types de paquets permettant de prendre en charge aussi ce nouveau modèle d'interaction. De plus, un langage de requête CEP a été défini en utilisant des opérateurs qui permettent de détecter les évènements

dans l'architecture ICN permettant d'offrir un traitement de requête centralisé ou distribué et de gérer un grand nombre des paquets d'intérêts en attente (Luthra *et al.*, 2019). Toutefois, INetCEP peut être non fiable et une perte des paquets peut être constatée si la bande passante ou les ressources disponibles dans les nœuds sont inférieures au taux d'envoi des données.

3.5 Conclusion

À travers ce chapitre, nous avons présenté les notions de base associées au NFN. Plus précisément, nous avons défini l'INC qui est un nouveau domaine de recherche, ainsi que ses avantages. Nous avons également présenté l'ICN, ses concepts de base et ses principales caractéristiques existantes dans la littérature. Finalement, nous avons détaillé les deux composantes principales, à savoir le flux de travail et les stratégies d'exécution du NFN et nous avons établi une étude des différentes stratégies proposées dans la littérature.

CHAPITRE IV

RÉSEAU DÉFINI PAR LOGICIEL (SDN)

4.1 Introduction

Nous présentons à travers ce chapitre le réseau défini par logiciel (SDN) qui est présenté par l'«Open Networking Foundation (ONF)» comme étant une architecture réseau composé de trois couches : données, contrôle et application (Kreutz *et al.*, 2014).

Étant donné que les réseaux informatiques actuels peuvent subir plusieurs pannes qui nécessitent des efforts importants, leurs gestions sont devenues une tâche complexe. En effet, ces réseaux se composent par un nombre énorme d'équipements exécutant des logiciels de contrôle complexe qui peuvent varier d'un fournisseur à un autre.

Cependant, afin de résoudre les problèmes connus par ces réseaux, offrir une gestion plus simple et permettre une supervision et un contrôle logiquement centralisé, SDN a changé la façon de fonctionnement des réseaux en découplant le plan de données du plan de contrôle (Nunes *et al.*, 2014; Xia *et al.*, 2014). Dans une architecture SDN, le plan de données composé par les équipements du réseau, est seulement responsable de la transmission des données. Le plan de contrôle composé par un ou plusieurs contrôleurs logiquement centralisés prend en charge les tâches de gestion et de prise de décisions de transmission de trafic. Afin d'optimiser les performances du réseau, SDN utilise la couche application pour déployer

diverses applications permettant par exemple de gérer les charges élevées du trafic, de garantir la sécurité et de virtualiser le réseau (Kreutz *et al.*, 2014; Xia *et al.*, 2014). Dans ce chapitre, nous commençons dans un premier temps par présenter l'architecture du SDN, ses couches et ses interfaces. Dans un second temps, nous définissons le protocole OpenFlow qui a été reconnu et standardisé par l'ONF et qui permet la communication entre la couche de contrôle et celle de données.

4.2 Architecture SDN

L'architecture de SDN présentée dans la figure 4.1, se base principalement sur un concept de séparation des couches avec un contrôle centralisé. La couche application est la plus haute et héberge toutes les applications SDN. Elle communique avec un ou plusieurs contrôleurs disponibles dans la couche de contrôle à travers une interface de programmation d'application (en anglais, *Application Programming Interface* (API)) en direction du nord. Le contrôleur permet une gestion et un contrôle centralisé de toute l'architecture et communique avec le plan de données qui est la couche la plus basse à travers une API en direction du sud.

4.2.1 Couche de données

La couche de données se compose d'un ensemble d'équipements de transfert, généralement des commutateurs qui peuvent être matériels ou logiciels (Rawat et Reddy, 2016) . Étant donné qu'un commutateur dans SDN n'a aucune intelligence, il se base sur le contrôleur pour acheminer le trafic en fonction des différentes règles de routage qui sont stockées dans la table de flux locale du commutateur pour une utilisation ultérieure. La plupart des commutateurs utilisés dans SDN suivent une conception basée sur le protocole OpenFlow et transfèrent les paquets en utilisant leurs tables de flux qui contiennent un ensemble d'entrées. Les équipements de la couche des données sont aussi responsables de la collecte et du stockage de l'état

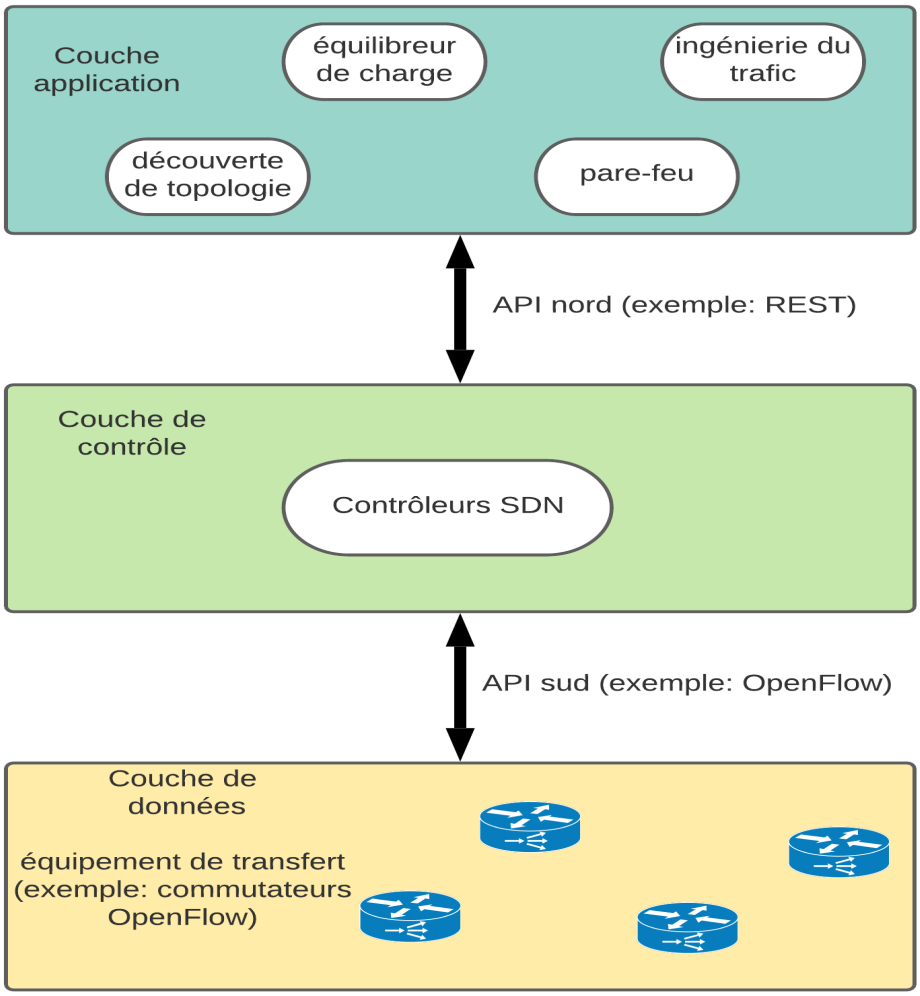

Figure 4.1: Architecture SDN

du flux réseau comme les statistiques du trafic avant de les envoyer aux contrôleurs (Nunes *et al.*, 2014; Rawat et Reddy, 2016).

4.2.2 Couche de contrôle

La couche de contrôle représente la partie intelligente de l'architecture SDN qui se compose par un ou plusieurs contrôleurs logiquement centralisés. Cette couche intermédiaire entre la couche des données et la couche des applications est res-

ponsable des décisions de routage ainsi que de la collecte des informations et des statistiques de la topologie du réseau. La couche de contrôle partage une vue globale et abstraite de l'infrastructure avec la couche application via une API nord. L'échange des informations entre la couche de contrôle et la couche des données est réalisé via une API sud permettant ainsi de simplifier la configuration et la gestion du réseau (Braun et Menth, 2014; Xia *et al.*, 2014; Rawat et Reddy, 2016). Toutefois, avoir un seul contrôleur centralisé dans une grande architecture réseau représente un point de défaillance et ne peut pas être en mesure de gérer efficacement les équipements de la couche des données. Cependant, vu qu'il effectue une tâche critique, d'autres contrôleurs de sauvegarde distribués peuvent exister dans la couche de contrôle pour assurer l'exécution des tâches si le contrôleur maître connaît une panne (Nunes *et al.*, 2014). Ces contrôleurs peuvent communiquer à l'aide d'une interface est-ouest.

Aujourd'hui, Ryu, Floodlight, OpenDaylight et NOX sont les contrôleurs les plus utilisés, permettant l'implémentation des fonctions comme la gestion des commutateurs et la découverte de la topologie. Le tableau 4.1 énumère les différents langages d'implémentations de chaque contrôleur ainsi que les API nord utilisés et les organismes de développement.

Tableau 4.1: Caractéristiques de contrôleurs SDN (Kreutz *et al.*, 2014; Xia *et al.*, 2014)

Caractéristique / Contrôleur	Langage d'implémentation	API nord	Organisme de développement
Ryu	Python	REST	NTT, OSRG group
Floodlight	Java	REST, Java RPC	Big switch networks
Opendaylight	Java	REST, RESTCONF	Linux Foundation
Nox	C++	ad-hoc API	Stanford/Nicira
Beacon	Java	ad-hoc API	Stanford
Trema	Ruby/C	ad-hoc API	NEC

4.2.3 Couche application

La couche application communique avec la couche de contrôle dans le but d'avoir une vue globale du réseau . Elle se compose d'une ou plusieurs applications ayant pour but de collecter des informations sur le réseau qui seront utilisées par la suite pour des prises de décisions (Xia *et al.*, 2014) . Cependant, les applications peuvent être classées sous différentes catégories par exemple pour la gestion de réseau et l'ingénierie du trafic, l'équilibrage de charge, la sécurité et le contrôle d'accès au réseau et la virtualisation de réseau (Braun et Menth, 2014).

4.2.4 Interfaces de programmation

Dans une architecture SDN, nous trouvons principalement trois types d'API qui permettent la communication entre les différentes couches :

— Nord : qui permet de fournir une interface de haut niveau dont le rôle consiste à gérer la connexion et la communication entre les couches de contrôle et application. Étant donné que les exigences d'une application réseau varient d'une application à l'autre, cette API ne dispose présentement d'aucune norme standard, laissant les contrôleurs spécifier l'API nord à utiliser (Braun et Menth, 2014).

— Sud : qui assure la communication entre la couche de contrôle et la couche des données en fournissant une interface d'interaction entre le contrôleur et les équipements de transfert de la couche de données. En effet, à travers cette interface, le contrôleur est capable de gérer en temps réel les équipements qui peuvent être physiques ou virtuels en échangeant des messages et des politiques. OpenFlow et ForCES sont des exemples d'APIs sud qui sont largement utilisés. Toutefois, OpenFlow, est aujourd'hui plus populaire puisqu'il est normalisé par l'ONF (Braun et Menth, 2014).

— Est-ouest : qui permet la communication entre plusieurs contrôleurs dans une architecture multi-contrôleurs dans le but d'orchestrer l'état du réseau (Braun et Menth, 2014).

4.3 Protocole OpenFlow

OpenFlow (OF) a été initialement proposé par l'Université de Stanford (McKeown *et al.*, 2008), et il est maintenant normalisé par l'ONF. À travers l'interface offerte par OpenFlow, le contrôleur peut définir des règles pour la gestion du trafic des données dans les commutateurs. En effet, les commutateurs utilisant OF peuvent être matériels créés par exemple par Cisco, IBM, Huawei ou bien logiciels comme Open Vswitch (Lara *et al.*, 2013; Akyildiz *et al.*, 2014). OF utilise trois catégories de messages pour établir la communication entre le contrôleur et le commutateur OF (Braun et Menth, 2014) :

— Asynchrone : envoyé par le commutateur OF et sert à notifier le contrôleur de l'arrivée des paquets, du changement d'état ou d'une erreur.

— Contrôleur-commutateur : transmis par le contrôleur pour récupérer et détecter les caractéristiques, les configurations et les informations des commutateurs OF.

— Symétrique : envoyé par le commutateur OF ou le contrôleur pour initier la communication.

4.3.1 Table de flux

Le commutateur OF utilise des tables de flux afin de stocker les règles reçues du contrôleur pour assurer la gestion du trafic. Ces tables contiennent des champs appelés entrées de flux ou aussi règles de flux ayant chacune une priorité de vérification, et permettant de définir comment un paquet sera traité (Nguyen *et al.*,

2015; Goransson *et al.*, 2016). En effet, chaque entrée de flux est composée de trois champs qui sont les suivants :

— En-tête : utilisé pour comparer les en-têtes des paquets entrants dans un ordre de priorité. Quand une correspondance complète est trouvée, l'action sera exécutée sur ce paquet. Le champ d'en-tête peut être parfois remplacé par un astérisque (*) qui indique un champ non pertinent à la correspondance et qui ne sera pas vérifié. Cependant, selon les besoins de l'application, nous pouvons utiliser soit tous les champs d'en-tête, soit seulement les champs de la couche 2 ou 3 du modèle TCP/IP. Les champs de l'en-tête d'une table de flux sont présentés dans la figure 4.2.

— Compteur : sert à collecter des statistiques sur les flux telles que le nombre des paquets et d'octets reçus, la durée du flux et le nombre des paquets abandonnés.

— Instructions : définit comment un paquet doit être géré quand il y a une correspondance avec une entrée de flux. En effet, ce champ permet la modification du paquet, de l'ensemble des actions ou même du processus de traitement.

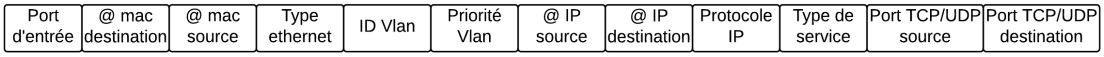

Figure 4.2: Champs d'une entrée de flux

4.3.2 Les instructions

Dans un commutateur OF, chaque entrée de flux contient un champ d'instruction permettant d'appliquer une liste d'actions si le paquet correspond au champ d'entête de l'entrée. Les instructions permettent d'établir des modifications sur les paquets, l'ensemble des actions ou sur le processus de traitement (Braun et Menth, 2014; Found, 2015). Parmi les instructions les plus communes, nous citons :

— Apply-Actions : applique directement les actions sur le paquet sans effectuer des changements sur l'ensemble des actions. Elle est utilisée pour établir une modification entre deux tables ou pour exécuter différentes actions du même type.

— Write-Actions : ajoute les actions spécifiées à l'ensemble d'actions.

— Clear-Actions : vide l'ensemble d'actions.

— Goto : permet de poursuivre le processus de traitement dans une autre table de flux référencée. Afin d'éviter les boucles de traitement, il faut spécifier un identifiant de table de flux plus élevé que la table actuelle.

4.3.3 Les actions

Un commutateur OF supporte différentes actions appliquées sur les paquets quand il y a une correspondance avec le champ d'en-tête (Braun et Menth, 2014; Found, 2015). En effet, les paquets qui n'ont aucune action définie explicitement sont supprimés sans avoir besoin d'utiliser une action permettant d'établir la suppression. Les actions de base sont :

— Output « numéro-port » : transfère le paquet vers un port OF qui peut être physique, logique ou réservé.

— Set-Queue « identifiant-queue » : précise la queue du port vers laquelle le paquet est transmis.

— Group « identifiant-groupe » : effectue le traitement du paquet selon les politiques du groupe spécifié.

— Meter « identifiant-meter » : dirige le paquet vers le compteur spécifié.

4.3.4 Illustration des opérations d'OpenFlow

Le traitement des paquets dans OpenFlow suit un processus bien défini qui est illustré dans la figure 4.3.

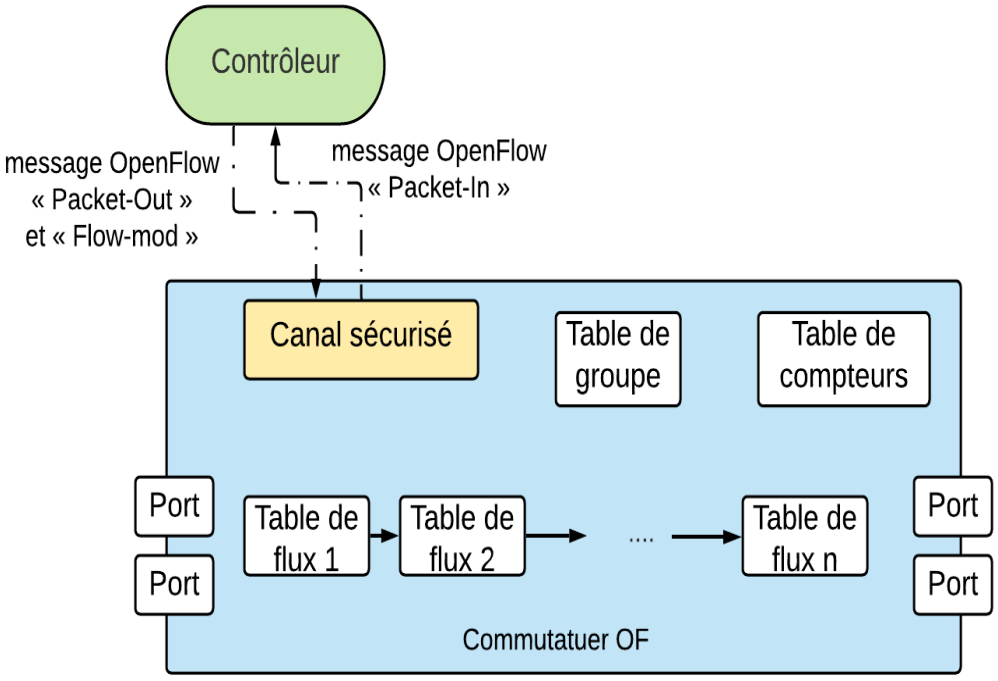

Figure 4.3: Composantes d'un commutateur OpenFlow (Found, 2015)

En utilisant le protocole OF, le contrôleur et le commutateur échangent les messages via un canal sécurisé qui se base sur le protocole TLS pour le cryptage asymétrique. À la réception d'un paquet, le commutateur commence par une analyse de l'en-tête du paquet en effectuant une correspondance avec la table de flux. Cependant, trois situations peuvent exister (Found, 2015) :

— Une correspondance est trouvée dans la table de flux, alors cette entrée est prise en compte ;

— Plusieurs entrées peuvent aussi correspondre avec l'en-tête du paquet, dans ce cas l'entrée la plus précise et ayant la priorité la plus élevée sera choisie ;

— Si aucune correspondance n'est trouvée, le paquet est encapsulé et envoyé vers le contrôleur via un canal sécurisé dans un message « Packet-In » permettant de notifier au contrôleur qu'aucune entrée de flux n'a été trouvée pour ce paquet. Par conséquent, le contrôleur établit les règles correspondantes au paquet et envoie des messages « Packet-Out » et « Flow-Mod » vers le commutateur pour mettre à jour la table de flux avec les nouvelles entrées.

Dans les deux premières situations, après avoir effectué la correspondance avec l'entrée de flux, le commutateur met à jour le compteur et exécute les instructions appropriées.

4.4 Conclusion

Dans ce chapitre, nous avons présenté le réseau défini par logiciel et plus précisément, son architecture générale, ses couches et ses interfaces. Grâce à son agilité et sa flexibilité ainsi que sa capacité d'optimiser la configuration et la performance du réseau, SDN est très populaire aujourd'hui auprès de la communauté de l'infonuagique et il est déployé dans de nombreux centres de données. Ensuite, nous avons présenté le premier protocole reconnu et standardisé par l'ONF qui est OpenFlow et qui permet la communication entre les couches de contrôle et de données dans l'architecture SDN.

CHAPITRE V

SOLUTION M-SANTÉ BASÉE SUR NFN ET SDN

5.1 Introduction

À travers ce chapitre, nous proposons une solution m-santé basée sur NFN et SDN qui peut être utilisée par chaque personne voulant accéder à un endroit public clos achalandé. Cette solution permet de collecter et d'analyser les données physiologiques générées en temps réel par les téléphones intelligents des utilisateurs et de détecter les cas suspects potentiellement porteurs de virus afin d'éviter la contagion. En effet, elle a pour but de minimiser le temps de latence afin d'éviter les files d'attente devant les portes d'entrée et de garantir plusieurs exigences de sécurité telles que l'authentification mutuelle, l'anonymat, la confidentialité.

Nous commençons ce chapitre par parcourir la littérature et donner l'état de l'art en relation avec le problème étudié, à savoir la mise en place de ICN dans une architecture SDN étant donné qu'il n'existe pas à ce jour des travaux qui prennent en considération l'intégration de NFN dans SDN. Ensuite, nous présentons une modélisation de la solution que nous proposons en décrivant son architecture et le scénario proposé ainsi que ses différentes exigences. Nous présentons aussi dans ce chapitre nos solutions proposées en termes (i) de protocole de sécurité permettant l'authentification des utilisateurs tout en protégeant leurs vies privées et (ii) d'intégration du paradigme NFN dans l'architecture SDN afin de bénéficier

des avantages de deux approches. Finalement, une analyse des fonctionnalités du protocole de sécurité proposée sera élaborée.

5.2 État de l'art

Le succès et la réussite du ICN viennent du fait qu'il résout les différents problèmes liés à l'architecture traditionnelle et présente divers avantages en termes de réduction de trafic dans le réseau, d'amélioration de QoS et de QoE, de fiabilité et de rapidité des résultats attendus. D'un autre côté, le SDN simplifie la programmabilité du réseau et offre une nouvelle approche pour configurer les différentes applications de sécurité, d'équilibrage de charge, d'ingénierie de trafic et de découverte de topologie. Toutefois, dans le but de bénéficier pleinement des avantages de ces deux approches et de résoudre les problèmes que rencontrent les réseaux traditionnels, leur intégration est devenue un objectif ultime (Dutta *et al.*, 2021).

Dans ce contexte, la prise en charge de ICN par SDN dans la littérature est envisagée en se basant sur deux approches différentes. Un premier groupe de solutions se base sur une stratégie d'implémentation à court terme c'est-à-dire que l'approche proposée utilise les commutateurs OF ainsi que les API déjà existantes pour réaliser les fonctionnalités ICN. Le deuxième groupe de solutions s'appuie sur une stratégie d'implémentation à long terme qui envisage les futurs commutateurs OF compatibles avec ICN en surmontant les limitations de OF et en proposant des modifications et des extensions.

5.2.1 Les solutions à court terme

Dans (Nguyen *et al.*, 2013), les auteurs ont proposé une solution basée sur l'architecture CCN qui permet le transfert basé sur le nom dans les commutateurs OpenFlow. L'idée de cette solution consiste à hacher le nom du contenu du pa-

quet d'intérêt en un entier et de le remplacer dans les champs que le commutateur OF peut gérer. Un nouveau module appelé « wrapper » est mis en place entre le commutateur OF et le nœud CCN permettant de lire les paquets d'intérêts ICN que le commutateur reçoit, de hacher le nom de contenu, de l'encapsuler dans un paquet IP et finalement de le renvoyer vers le commutateur OF pour commencer la transmission et la surveillance des paquets en fonction de leur nom de contenu. Cette solution se base sur une approche qui ne nécessite pas la modification ni du protocole OF, ni du ICN. Toutefois, elle souffre du problème de devoir changer la sémantique des champs utilisés pour stocker les informations de nom en le remplaçant par un hachage du nom de contenu de ICN ce qui résulte à une perte de signification des noms des champs. De plus, les opérations supplémentaires réalisées sur chaque paquet ont un impact important sur les performances de transmission.

Les auteurs de (Vahlenkamp *et al.*, 2013) ont présenté une solution de superposition dans laquelle les services offerts par ICN sont déployés sur les réseaux IP tout en utilisant une architecture SDN et en mettant en place un nouvel identifiant de protocole afin de reconnaitre les paquets ICN. Un utilisateur voulant récupérer un contenu à partir du domaine ICN doit envoyer un paquet d'intérêt au commutateur qui le transmet par la suite au contrôleur qui offre une vue globale de l'architecture. Ce dernier vérifie le domaine ICN responsable du contenu demandé et renvoie une adresse réseau publique du domaine ICN au commutateur. Finalement, une communication est établie entre le commutateur et le domaine ICN. Néanmoins, cette solution présente un processus trop long et complexe qui peut impacter les performances de transmission, ce qui ne peut pas correspondre avec les scénarios à temps réel.

Dans (Eum *et al.*, 2015), les auteurs ont utilisé un identifiant de données pour le domaine ICN ayant une longueur fixe et qui sert à identifier les données à récupé-

rer. Cet identifiant est composé de deux champs : « Label (L) » qui est attribué par le producteur de contenu et il est unique sous son domaine et « Principal (P) » qui représente un hachage cryptographique de la clé publique du producteur de contenu. Ils ont aussi utilisé un identifiant de transfert qui est utilisé dans le domaine OF-ICN ayant une longueur fixe et permettent d'identifier un chemin de transfert entre le consommateur et le producteur. Ces deux identifiants sont inclus dans les champs des paquets UDP (en anglais, *User Datagram Protocol*) pour remplacer le port source et destination. Par conséquent, à cause de ces modifications, cette solution souffre du problème de changement de la sémantique des paquets UDP et peut provoquer des retards de traitement pour coder et décoder ces identifiants.

Finalement, une solution à court terme a été proposée par (van Adrichem et Kuipers, 2015) qui se base sur l'utilisation de la couche application de l'architecture SDN et dans laquelle le commutateur et le contrôleur utilisent des plug-ins supplémentaires pour pouvoir traiter les paquets d'intérêts ICN. De plus, dans cette proposition, les auteurs ont ajouté un autre canal de communication séparé du canal de communication OpenFlow existant. En outre, le contrôleur utilise un module ICN supplémentaire et les commutateurs sont améliorés en implémentant des capacités ICN via CCN et son démon CCNx. Pour pouvoir traiter les paquets d'intérêts ICN, les commutateurs doivent établir des connexions additionnelles et gérer plusieurs protocoles. Par conséquent, le module ajouté pour la découverte de topologie et l'établissement de liens dans le contrôleur afin de mieux gérer le module ICN peut en effet conduire à des retards de traitement et des décisions opposées.

5.2.2 Les solutions à long terme

Dans (Aubry *et al.*, 2015), les auteurs ont proposé une solution qui repose uniquement sur ICN et n'utilise pas les protocoles de communication traditionnels. Pour ce faire, un schéma de routage SDN pour CCN a été proposé qui se base sur de nouveaux types de messages qui ont été créés afin d'effectuer la communication entre le commutateur et le contrôleur tout en prenant en considération ICN. Ces messages sont transmis entre le commutateur et le contrôleur uniquement via des paquets d'intérêt et de données. Toutefois, cette solution génère un nombre élevé de messages échangés, ce qui peut réellement saturer le réseau et faire exploser les délais de transmission.

La proposition de (Li *et al.*, 2015a) est basé sur la modification du commutateur OF ainsi que l'identification du type de paquet à traiter. En effet, les auteurs ont apporté des changements sur le commutateur OpenFlow en ajoutant les tables CS, PIT et FIB qui sont utilisées dans l'architecture NDN pour prendre en charge le stockage de contenu, la gestion des paquets d'intérêts en attente et les informations de transfert. De plus, pour pouvoir identifier les paquets de données, le nom de contenu a été haché dans les paquets. En révisant la structure des tables de flux et des entrées de flux, le commutateur est capable d'identifier le type du paquet reçu et le traiter selon le processus de transfert correspondant.

(Veltri *et al.*, 2012) ont proposé une solution basée sur une architecture ICN nommée CONET qui utilise un mécanisme de transmission des messages d'intérêt appelé «lookup-and-cache». Cette solution prévoit que les nœuds du réseau qui ne connaissent pas le prochain nœud pour transmettre un message d'intérêt, envoient une requête à un NRS disponible dans la couche de contrôle pour fournir les informations de routage nécessaires. Cependant, les nœuds du réseau ne peuvent exécuter que des opérations de transfert, tandis que le contrôleur a été étendu

pour agir comme un NRS afin d'effectuer la création et le maintien des règles de transfert. En outre, des tables de transfert basées sur les noms dans le nœud ICN ont été mises en place pour lier les noms de contenu aux nœuds suivants. Néanmoins, ce processus ne fait qu'augmenter le temps de réponse si un contenu demandé ne figure pas dans la table de transfert locale et le nœud doit demander au contrôleur de chercher le nom dans les autres nœuds.

5.3 Modélisation du système

En nous référant au contexte actuel de la pandémie de COVID-19 et du déconfinement, nous présentons dans la figure 5.1 une modélisation de notre système. Le but de notre cas d'utilisation consiste à collecter et à analyser les données physiologiques (température, saturation en oxygène et fréquence cardiaque) des professeurs, employés et étudiants qui veulent accéder à un campus universitaire. Notamment, des cas de faux positifs peuvent exister si la personne souffre d'une autre maladie (ex. hypertension, thyroïde, etc.). Pour éviter cette situation, lorsque les résultats de l'analyse sont positifs, les données physiologiques sont envoyées vers le réseau hospitalier afin d'établir un examen du dossier médical et des antécédents. Ce processus d'analyse et de détection n'est effectué qu'après une authentification mutuelle de l'utilisateur auprès du réseau universitaire et aussi du réseau hospitalier. Nous décrivons dans la section suivante l'architecture ainsi que les différents composants de notre cas d'utilisation et finalement nous définissons le fonctionnement ainsi que les différentes étapes de notre cas d'utilisation.

5.3.1 Architecture du modèle proposé

L'architecture globale de notre modèle de système est modélisée dans la figure 5.2 et elle est composée :

— d'un réseau universitaire dans lequel il existe (i) un point d'accès sans fil

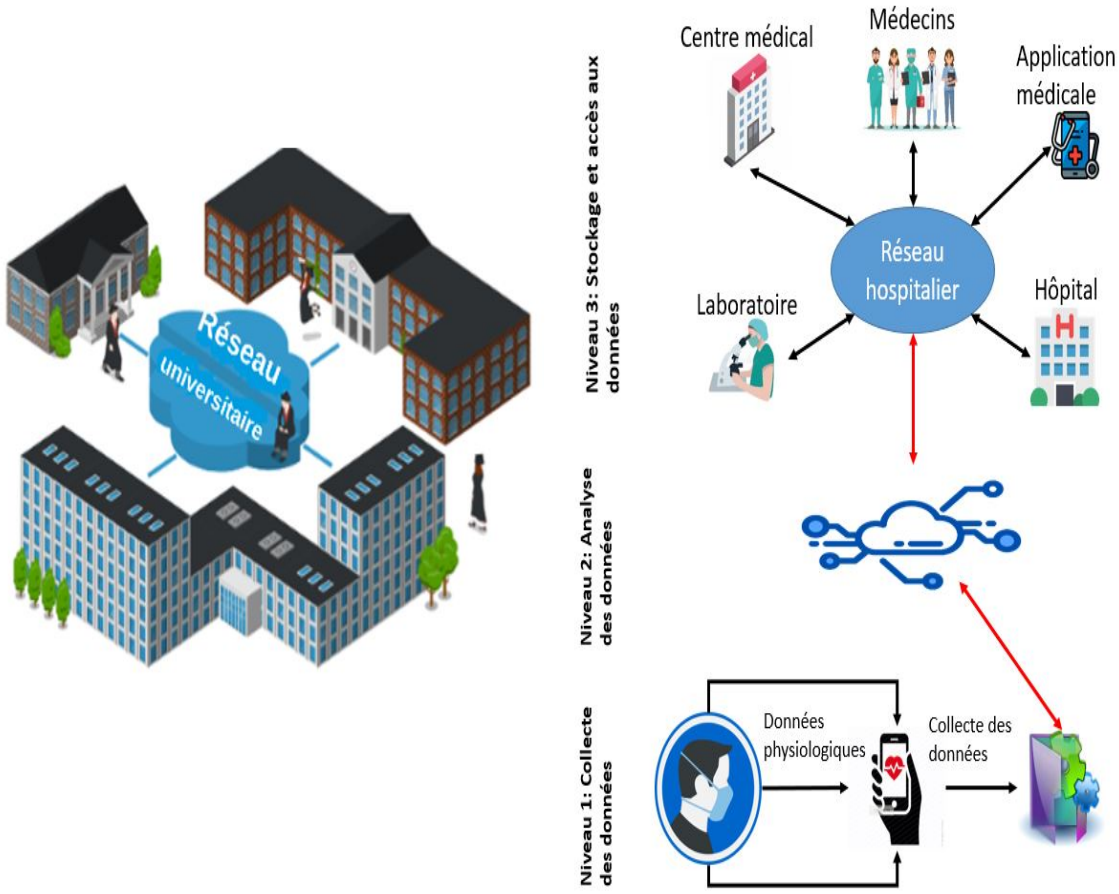

Figure 5.1: Modélisation du système

qui permet la communication du téléphone intelligent de l'utilisateur avec le réseau universitaire ainsi que la liaison du réseau avec Internet et (ii) des commutateurs que nous appelons OF-NFN du fait qu'ils sont capables de supporter OF et NFN. Ces commutateurs permettent l'exécution des fonctions NFN pour l'analyse des données physiologiques ;

— d'un réseau hospitalier disposant d'une passerelle réseau vers l'Internet afin de permettre la communication avec le point d'accès du réseau universitaire, et d'un commutateur OF-NFN qui permet l'exécution des fonctions NFN pour l'examen de l'historique médical de l'utilisateur ;

— d'un gestionnaire d'identité disponible dans la couche application qui permet d'authentifier l'utilisateur au sein du réseau universitaire et hospitalier dans le but d'assurer la sécurité, la confidentialité et de garantir la vie privée ;

— les deux réseaux (hospitalier et universitaire) sont supervisés et gérés chacun par un contrôleur SDN disponible dans la couche réseau afin d'avoir une vue globale et de bénéficier de la gestion et la flexibilité de chaque réseau ;

— le protocole OF est utilisé pour établir la communication entre la couche des données composées par les réseaux universitaire et hospitalier et la couche de contrôle ;

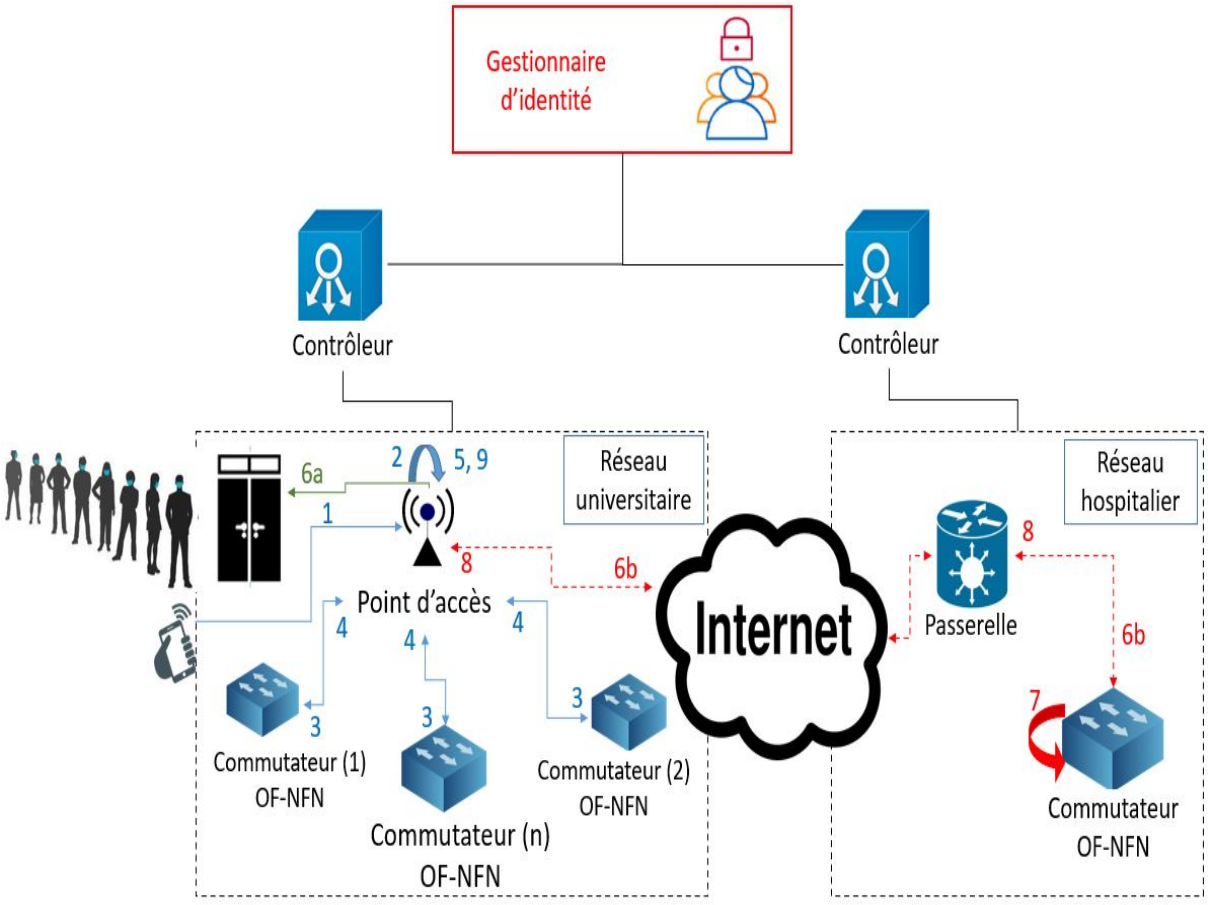

Figure 5.2: Description du cas d'utilisation

5.3.2 Cas d'utilisation

Cette section décrit les différentes étapes effectuées par notre système proposé. Afin de commencer le processus d'analyse des données physiologiques, chaque utilisateur est authentifié auprès du réseau universitaire et hospitalier par le biais du gestionnaire d'identité pour détecter les cas suspects. Les étapes sont présentées dans la figure 5.2 et elles sont expliquées ci-dessous :

- **étape 1 :** Les données physiologiques comme la température, la saturation en oxygène et la fréquence cardiaque sont envoyées à partir du téléphone intelligent de l'utilisateur vers le point d'accès du réseau universitaire ;
- **étape 2 :** Le point d'accès crée des paquets UDP contenant les données physiologiques reçues ainsi que le nom des fonctions à exécuter ;
- **étape 3 :** Ensuite, il envoie chaque paquet créé vers un commutateur OF-NFN pour lancer l'analyse des symptômes à travers des fonctions NFN ;
- **étape 4 :** Les commutateurs OF-NFN établissent l'analyse des données physiologiques de chaque utilisateur et renvoient les résultats vers le point d'accès ;
- **étape 5 :** Ce dernier vérifie les résultats afin : **étape 6-a :** d'autoriser l'accès à la personne ou **Étape 6-b :** d'envoyer les données suspectes vers le réseau hospitalier ;
- **étape 7 :** Le commutateur OF-NFN du réseau hospitalier effectue l'examen du dossier médical et des antécédents en exécutant des fonctions NFN ;
- **étape 8 :** Ils renvoient les résultats obtenus vers le point d'accès ;
- **étape 9 :** Finalement, le point d'accès prend la décision définitive pour autoriser ou refuser l'accès de l'utilisateur au campus universitaire s'il s'avère qu'il est probable d'avoir le COVID-19.

Il est essentiel dans une solution de m-santé de satisfaire les exigences attendues en termes de QoS, de temps de réponse, de fiabilité de résultats et de sécurité étant donné que le retard, les pertes d'informations ou les résultats erronés peuvent induire à des conséquences comme la création des longues files d'attente devant les portes d'entrée. Pire, l'obtention de faux positifs pour une personne ayant le COVID-19 lui autorisant l'accès et par conséquent propager le virus. De plus, la nature hautement confidentielle des données médicales exige un haut degré de garantie de sécurité, de confidentialité et de vie privée afin d'éviter de mettre à risque la vie des utilisateurs (ex. harcèlement par les attaquants, utilisation des données personnelles à des fins non légales, etc.).

5.4 Protocole de sécurité proposée

La mise en place d'une solution d'analyse et de détection m-santé exige la protection et la sécurité des données ainsi que la garantie de la vie privée des utilisateurs. S'inspirant du travail présenté par (Alzahrani *et al.*, 2020), nous présentons dans cette section un protocole d'enregistrement et d'authentification des utilisateurs auprès du réseau universitaire et du réseau hospitalier pour permettre leurs authentifications de façon sécuritaire et protéger leurs vies privées. En effet, (Alzahrani *et al.*, 2020) ont proposé un protocole qui permet l'authentification des capteurs corporels auprès d'un système de soins de santé via des nœuds intermédiaires.

Dans notre protocole, un utilisateur représenté par son téléphone intelligent est mutuellement authentifié auprès du réseau universitaire et du réseau hospitalier. Le processus d'authentification est établi à partir du module du gestionnaire d'identité disponible dans la couche application de l'architecture SDN via un point d'accès présent dans la couche des données de l'architecture SDN. Les notations utilisées dans ce protocole sont présentées dans le Tableau 5.1. Le protocole proposé repose sur trois phases à savoir : (i) initialisation des paramètres

(ii) enregistrement des utilisateurs et (iii) authentification mutuelle. Les étapes (i) et (ii) ne sont effectuées qu'une seule fois. Les détails de toutes les étapes sont illustrés dans les figures présentées ci-dessous.

Tableau 5.1: Liste des notations

Notations	Descriptions
ID_{smart}	Identifiant unique du téléphone intelligent
K_{smart}	Clé secrète du téléphone intelligent
TID_{smart}	Identité temporaire du téléphone intelligent
N_{im}, N_h	Nonce
PUB_{im},$PRIV_{im}$	Paire de clés publique/privée du gestionnaire d'identité
PUB_h, $PRIV_h$	Paire de clés publiques/privée du réseau hospitalier
$K_{session}$	Clé de session établie entre le gestionnaire d'identité et le téléphone intelligent
J_{im}, A_1,A_2,A_3,b_1,B_2,b_s	Paramètres de sécurité
x, y, z	Des secrets temporaires aléatoires
T_1,T_2,T_3,T_4	Horodatages
h(.)	Fonction de hachage à sens unique

5.4.1 Initialisation des paramètres

La phase d'initialisation consiste à générer les paramètres appropriés par le gestionnaire d'identité et elle est nécessaire avant de commencer l'enregistrement des utilisateurs. Les étapes suivantes sont effectuées dans cette phase :

- Le gestionnaire d'identité génère une paire de clés publique/privée (PUB_{im}, $PRIV_{im}$) en utilisant l'algorithme de chiffrement asymétrique RSA.
- Ensuite, il génère un nonce N_{im} en utilisant un générateur aléatoire cryptographique.

De même, le réseau hospitalier doit générer les paramètres requis pour la communication avec le gestionnaire d'identité. Les étapes suivantes sont établies dans cette phase :

- Le réseau hospitalier génère une paire de clés publique/pivée (PUB_h, $PRIV_h$) en utilisant l'algorithme de chiffrement asymétrique RSA.
- Ensuite, il génère un nonce N_h en utilisant un générateur aléatoire cryptographique.
- Finalement, il envoie au gestionnaire d'identité la clé publique PUB_h et N_h.

5.4.2 Enregistrement des utilisateurs

Chaque personne doit s'enregistrer auprès des réseaux universitaire et hospitalier pour obtenir les services de vérification nécessaire afin d'accéder au campus universitaire. Pour ce faire, le gestionnaire d'identité doit tout d'abord envoyer au téléphone intelligent sa clé publique PUB_{im}. Ensuite, le téléphone intelligent de chaque utilisateur enverra son adresse IMEI représentée par ID_{smart}, et qui sera cryptée avec PUB_{im} au gestionnaire d'identité à travers le point d'accès.
À la réception du ID_{smart} crypté avec PUB_{im}, le gestionnaire d'identité va le décrypter avec sa clé privée $PRIV_{im}$ pour récupérer ID_{smart}. Ensuite, il génère une clé de session $K_{session}$ pour établir une communication sécurisée avec le téléphone intelligent. Le gestionnaire d'identité calcule par la suite une clé privée du téléphone intelligent exprimée par $K_{smart} = h\ (ID_{smart} \| \mathrm{N}_{im})$, une identité temporaire du téléphone intelligent représentée par $TID_{smart} = h\ (\ ID_{smart} \| \mathrm{K}_{session})$ et un autre paramètre $J_{im} = ID_{smart} \oplus h(\mathrm{TID}_{smart} \| \mathrm{N}_{im})$. Finalement, le gestionnaire d'identité envoie un premier message vers le téléphone intelligent contenant K_{smart}, $K_{session}$, et TID_{smart} qui ne sont pas chiffrés via le point d'accès. Cependant, N_{im} ainsi que J_{im} sont gardés secrets et ne sont pas échangés avec le téléphone intelligent.
Les étapes effectuées dans cette phase sont décrites dans la figure 5.3.

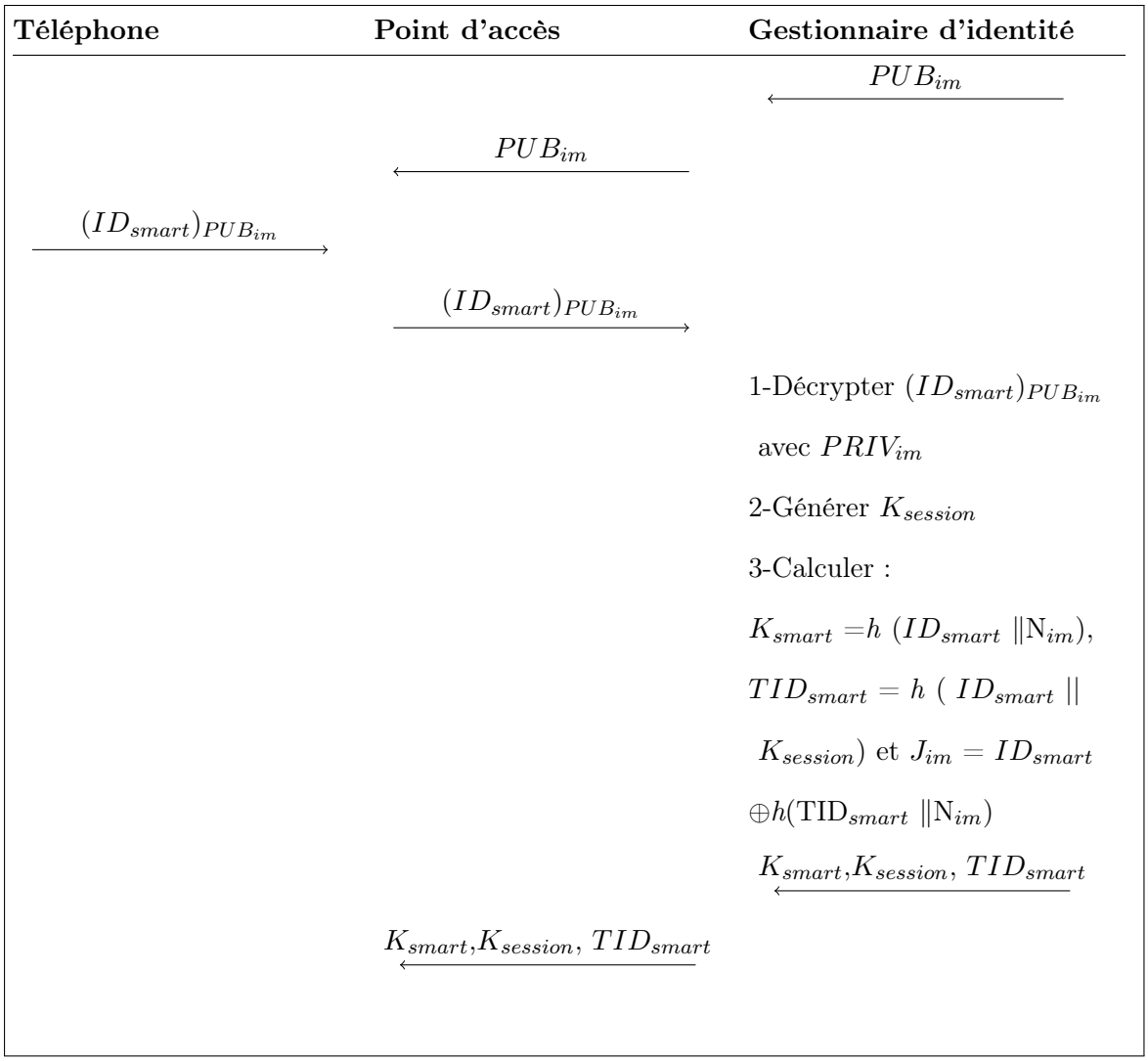

Figure 5.3: Phase d'enregistrement de l'utilisateur

5.4.3 Authentification mutuelle des utilisateurs

Cette phase permet d'authentifier les utilisateurs munis de leurs téléphones intelligents auprès des réseaux universitaire et hospitalier tout en garantissant plusieurs exigences de sécurité telles que l'anonymat, la non-traçabilité, la confidentialité, etc.

Dans cette étape, la mémoire du téléphone intelligent est initialisée avec les paramètres $\{PUB_{im}, ID_{smart}, K_{smart}, TID_{smart}$, et $K_{session}\}$ qui ont été envoyés auparavant par le gestionnaire d'identité, tandis que ce dernier est initialisé avec les paramètres suivants $\{PUB_{im}, PRIV_{im}, K_{session}, J_{im}, ID_{smart}, TID_{smart}, PUB_h, N_{im}$ et $N_h\}$. La procédure d'authentification mutuelle entre le téléphone intelligent et le gestionnaire d'identité est illustrée comme suit :

- **Étape 1 :** Le téléphone intelligent commence par choisir un entier aléatoire x et génèrer un horodatage T_1. Ensuite, il calcule $A_1 = \mathrm{x} \oplus h(\mathrm{K}_{smart} \| \mathrm{K}_{session})$, $b_s = h(K_{session})$ et $B_1 = h(ID_{smart} \| \mathrm{TID}_{smart} \| \mathrm{b}_s \| \ x \ \| \mathrm{T}_1)$. Puis, il envoie un message contenant $\{B_1, T_1, A_1, TID_{smart}\}$ vers le gestionnaire d'identité via le point d'accès.
- **Étape 2 :** Quand le gestionnaire d'identité reçoit le 1^{er} message envoyé par le téléphone intelligent, il génère un horodatage T_2 et vérifie l'authenticité du message reçu en calculant un seuil de différence T tel que $|T_2 - T_1| < T$. Le gestionnaire d'identité abandonne la connexion si le seuil est expiré, sinon il calcule $ID_{smart} = J_{im} \oplus h(\mathrm{TID}_{smart} \| \mathrm{N}_{im})$, ensuite $K_{smart} = h(ID_{smart} \| \mathrm{N}_{im})$, $x = A_1 \oplus h(\mathrm{K}_{smart} \| \mathrm{K}_{session})$ et vérifie finalement si $h(ID_{smart} \| \mathrm{TID}_{smart} \| \mathrm{b}_s \| \ x \ \| \mathrm{T}_1) \ ? = B_1$. Si les deux valeurs sont différentes, il abandonne la connexion. Sinon, il choisit entier aléatoire y et calcule un nouveau $K_{session} = h(ID_{smart} \ \| \mathrm{TID}_{smart} \| \mathrm{K}_{smart} \| \ x \ \| \ y \ \| \mathrm{T}_1)$, une nouvelle identité temporaire avec $K_{session}$ de la session actuelle et J_{im} et les remplace avec les nouvelles valeurs. En-

Téléphone	**Point d'accès**	**Gestionnaire d'identité**
1-Générer x et T_1		
2-Calculer A_1=x$\oplus h$(K$_{smart} \parallel K_{session}$)		
$b_s = h(K_{session})$ et $B_1 = h(ID_{smart} \parallel$		
$TID_{smart} \parallel b_s \parallel x \parallel T_1)$		
$\xrightarrow{B_1 \,;\, T_1 \,;\, A_1 \,;\, TID_{smart}}$		
	$\xrightarrow{B_1 \,;\, T_1 \,;\, A_1 \,;\, TID_{smart}}$	
		1-Générer T_2 et vérifier si $\lvert T_2 -$
		$T_1 \rvert < T$
		2-Calculer $ID_{smart} = J_{im} \oplus h($
		$TID_{smart} \parallel N_{im})$, $K_{smart} = h($
		$ID_{smart} \Vert$N$_{im})$ et $x = A_1 \oplus h($
		$K_{smart} \Vert$K$_{session})$
		3-Vérifier si $h(ID_{smart} \Vert$TID$_{smart}$
		$\Vert$b$_s \Vert\, x \Vert$T$_1) = B_1$.
		4-Générer y et calculer $K_{session}$
		$= h(ID_{smart} \Vert$TID$_{smart} \Vert$K$_{smart}$
		$\parallel x \Vert y \Vert$T$_1)$
		5-Mettre à jour TID_{smart} et J_{im}
		6-Calculer $A_2 = y \oplus h($K$_{smart} \Vert \Vert$
		$K_{session} \parallel x)$ et $B_2 = h(K_{session})$
		$\xleftarrow{B_2 \,;\, T_2 \,;\, A_2 \,;\, TID_{smart}}$
	$\xleftarrow{B_2 \,;\, T_2 \,;\, A_2 \,;\, TID_{smart}}$	

Figure 5.4: Étapes 1 et 2 de la phase d'authentification

suite, il calcule $A_2 = y \oplus h(\mathrm{K}_{smart} \| \mathrm{K}_{session} \| x)$ et $B_2 = h(K_{session})$. Finalement, il crée un message contenant $\{B_2, T_2, A_2, TID_{smart}\}$ et l'envoie au téléphone intelligent à travers le point d'accès. Les étapes 1 et 2 sont modélisées dans la figure 5.4.

- **Étape 3 :** En recevant le message envoyé par le gestionnaire d'identité, le téléphone intelligent vérifie la validité de T_3 en calculant $|T_3 - T_2| < T'$. Ensuite, il calcule $y = A_2 \oplus h(\mathrm{K}_{smart} \| \mathrm{K}_{session} \| x)$, $K_{session} = h(ID_{smart} \| \mathrm{TID}_{smart} \| \mathrm{K}_{smart} \| x \| \mathrm{T}_1)$ et vérifie si $h(K_{session}) \stackrel{?}{=} B_2$. Si la vérification échoue, le téléphone intelligent met fin à la session. Sinon, il met à jour $K_{session}$ par la nouvelle valeur après une vérification réussie de l'authenticité. Enfin, il crypte son numéro de téléphone « Numero » avec PUB_{im} et il l'envoie vers le gestionnaire d'identité à travers le point d'accès. Cette étape est représentée par la figure 5.5.

- **Étape 4 :** Dans cette étape, le gestionnaire d'identité décrypte le numéro de téléphone crypté reçu avec sa clé privée $PRIV_{im}$. Ensuite, il génère un horodatage T_4 et choisit un entier aléatoire z. Après, il calcule $A_3 = z \oplus h(\mathrm{N}_h)$ et $C = h(\mathrm{Numero} \| z \| \mathrm{T}_4)$. Finalement il crypte le numéro de téléphone avec la clé publique du réseau hospitalier PUB_h et crée un message contenant $\{A_3, C, T_4, (Numero)_{PUB_h}\}$ et le transmet au réseau hospitalier.

- **Étape 5 :** Finalement, le réseau hospitalier génère un horodatage T_5 et vérifie la validité de T_4 en calculant $|T_5 - T_4| < T''$. Ensuite, il décrypte le numéro de téléphone crypté reçu avec sa clé privée $PRIV_h$, calcule $z = A_3 \oplus h(\mathrm{N}_h)$ et vérifie si $h(\mathrm{Numero} \| z \| \mathrm{T}_4) \stackrel{?}{=} C$. Si la vérification réussie, il envoie un code de confirmation généré aléatoirement vers le numéro de téléphone décrypté qui doit ensuite confirmer la réception du code aléatoire en le renvoyant vers le réseau hospitalier pour que l'authentification soit complètement terminée. Sinon, il abandonne la session. La figure 5.6 représente les étapes 4 et 5.

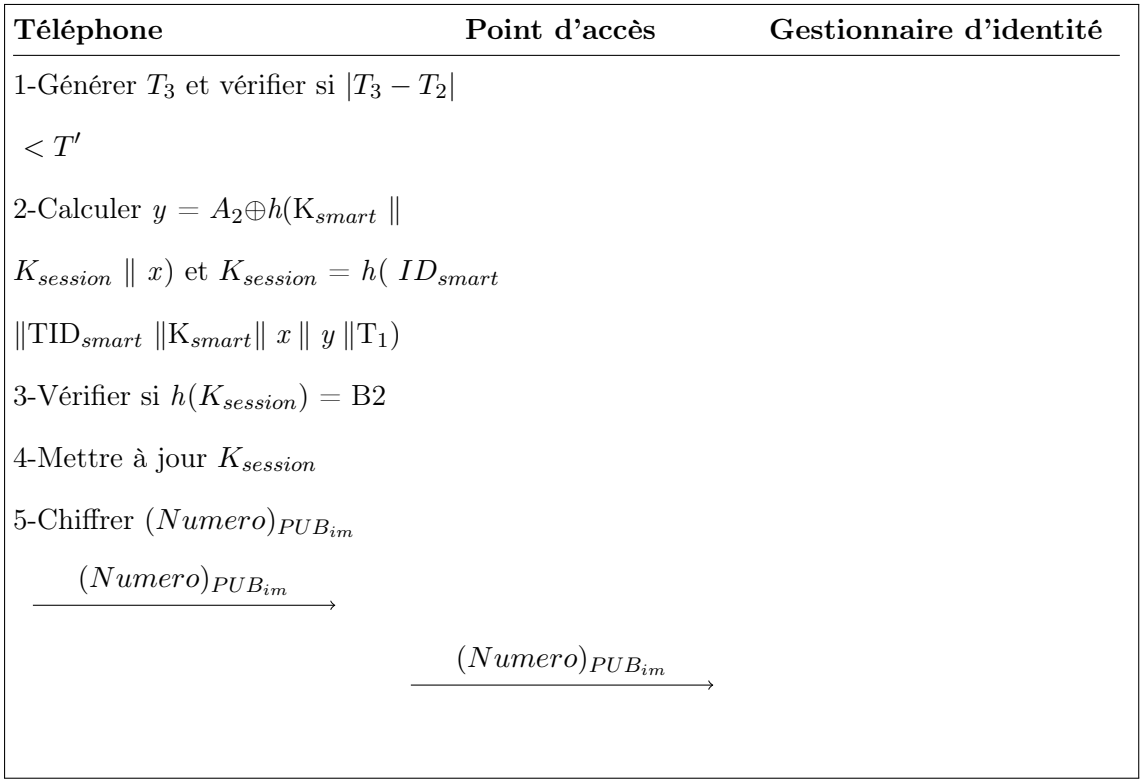

Figure 5.5: Étape 3 de la phase d'authentification

Téléphone	**Point d'accès**	**Gestionnaire d'identité**	**Hôpital**
		1-Décypter $(Numero)_{PUB_{im}}$ avec $PRIV_{im}$ 2-Générer z et T_4 3-Calculer $A_3{=}z{\oplus}h(\mathrm{N}_h)$ et $C = h(\mathrm{Numero}\| z \|\mathrm{T}_4)$ 4-Crypter $(Numero)_{PUB_h}$ A_3 ; C ; T_4 ; $(Numero)_{PUB_h}$ $\longrightarrow$	
			1-Générer T_5 et vérifier si $\|T_5 - T_4\| < T''$ 2-Décrypter $(Numero)_{PUB_h}$ avec $PRIV_h$ 3-Calculer $z{=}A_3{\oplus}h(\mathrm{N}_h)$ et vérifier si $h(Numero \| z \|\mathrm{T}_4) = C$ 4- Générer un code de confirmation aléatoire si la vérification est réussie 5- Envoyer le code au téléphone de l'utilisateur

Figure 5.6: Étapes 4 et 5 de la phase d'authentification

5.5 Solution d'intégration NFN/SDN proposée

Afin de pouvoir utiliser l'approche NFN dans une architecture SDN et de permettre la communication entre les différents nœuds et contrôleurs, nous proposons une solution basée sur l'utilisation des paquets UDP pour la transmission des données. Cette solution permet de gérer non seulement les paquets d'intérêt et de données, mais aussi les flux IP. Le fonctionnement de cette solution d'intégration NFN/SDN repose sur un processus à quatre étapes.

- **Étape 1 :** Lorsque le point d'accès reçoit les données physiologiques de l'utilisateur, il crée un paquet UDP. Le champ des données du paquet IP sera divisé en deux afin qu'il puisse supporter les deux nouveaux champs soient le nom de la fonction et les données.
- **Étape 2 :** Ensuite chaque paquet est envoyé vers un commutateur OF-NFN du campus universitaire disponible pour effectuer l'analyse des données physiologiques. À la fin de cette étape, chaque commutateur doit renvoyer un paquet UDP vers le point d'accès contenant dans le champ des données le résultat de l'exécution de la fonction d'analyse.
- **Étape 3 :** À la réception des résultats envoyés par les commutateurs OF-NFN, le point d'accès prend la décision pour l'ouverture des portes ou l'envoi des données physiologiques pour une analyse approfondie du dossier médical de l'utilisateur dans le réseau hospitalier. Cependant, le point d'accès encapsule cette fois les données physiologiques seulement dans un paquet UDP qui est envoyé vers la passerelle du réseau hospitalier pour le transmettre vers un commutateur OF-NFN.
- **Étape 4 :** Après un examen du dossier médical de l'utilisateur, le commutateur OF-NFN du réseau hospitalier envoie de la même façon que l'étape 2, un paquet UDP contenant le résultat de la vérification vers la passerelle

qui le transfère vers le point d'accès

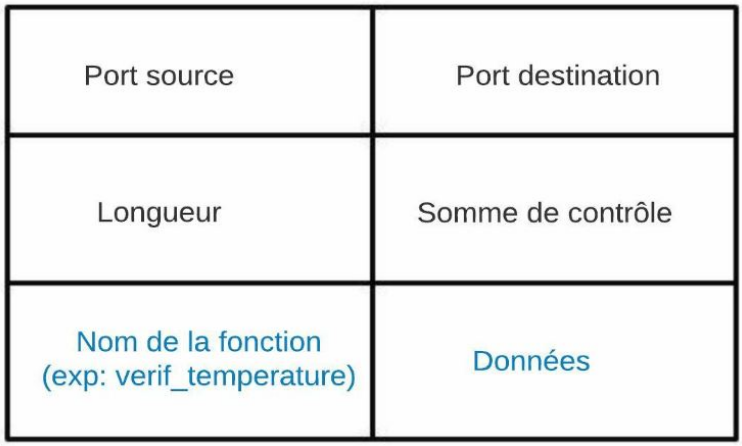

Figure 5.7: Paquet UDP créé par le point d'accès

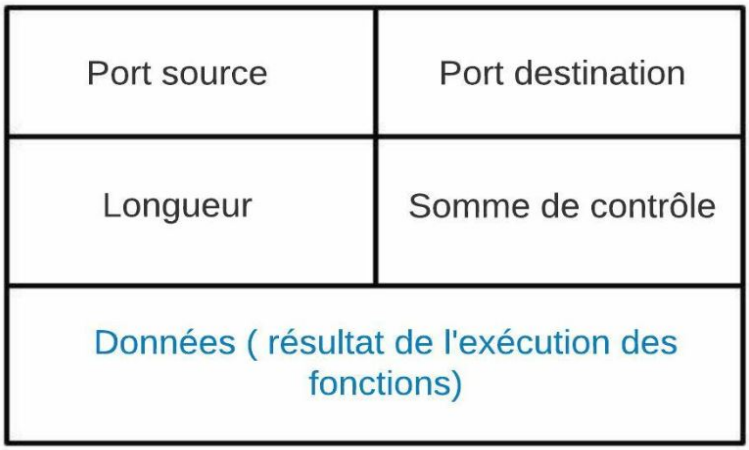

Figure 5.8: Paquet UDP créé par les commutateurs OF-NFN

5.6 Analyse de sécurité

Dans cette section, nous vérifions que notre protocole de sécurité proposé accomplit les exigences de sécurité et satisfait les propriétés requises. Pour ce faire, nous adoptons une méthode informelle pour la vérification.

5.6.1 Authentification mutuelle

Dans notre protocole de sécurité proposé, le gestionnaire d'identité authentifie le téléphone intelligent de l'utilisateur et valide son identité en vérifiant si $h(ID_{smart}\|\mathrm{TID}_{smart}\|\mathrm{b}_s\|\ x\ \|\mathrm{T}_1) = B_1$. Étant donné que les paramètres utilisés pour le calcul de B_1 et la vérification de son exactitude ne sont connus que par le téléphone intelligent, cette étape garantit que les paramètres ID_{smart}, TID_{smart}, b_s, x et T_1 proviennent de la bonne source et par conséquent la validation de son identité.

Le téléphone intelligent à son tour, confirme que le message est authentique et provient de la bonne source en vérifiant si $h(K_{session}) = B_2$. Afin de pouvoir calculer $K_{session}$, le téléphone intelligent doit calculer y en utilisant A_2 qui a été envoyé par le gestionnaire d'identité. La modification de l'un de ces paramètres empêche la validation de B_2 et par conséquent le gestionnaire d'identité ne sera pas reconnu comme étant une source fiable.

D'un autre côté, le réseau hospitalier s'assure de l'authenticité du gestionnaire d'identité en vérifiant si $h(\mathrm{Numero}\|\ z\ \|\mathrm{T}_4) = C$ et de la légitimité du téléphone intelligent en lui envoyant un code de confirmation généré aléatoirement qui doit lui être renvoyé par le destinataire. L'utilisation de cette vérification prouve que l'altération de l'un des paramètres utilisés pour la vérification de l'exactitude de C engendre un faux résultat et par conséquent l'inauthenticité du message.

L'utilisation des nombres aléatoires, les identités temporaires, les nonces et le chiffrement des données sensibles comme l'ID_{smart} et le numéro de téléphone rend le protocole plus robuste. Par conséquent, le protocole proposé garantit l'authentification mutuelle entre le gestionnaire d'identité, le téléphone intelligent et le réseau hospitalier.

5.6.2 Anonymat et non-traçabilité

L'utilisation des identités temporaires pour établir la communication au lieu d'utiliser les vraies identités garantit la non-révélation des identités des utilisateurs même si les variables de session temporaire $K_{session}$ ou la clé secrète du téléphone intelligent K_{smart} sont divulguées. En raison de leur utilisation combinée avec plusieurs autres paramètres, la divulgation de ces paramètres ne peut pas aboutir au dévoilement de l'identité réelle de l'utilisateur. De plus, les identités réelles ne sont jamais transmises en clair puisque le protocole proposé crypte l'adresse IMEI du téléphone intelligent avec la clé publique du gestionnaire d'identité pour qu'elle représente un identifiant unique du téléphone intelligent et qu'elle puisse être envoyée vers le gestionnaire d'identité.

La non-traçabilité garantit qu'un attaquant qui observe le canal de communication sur plusieurs sessions, ne peut pas retracer ou identifier les actions réalisées (Kumar et Chand, 2020). Cependant, les identités temporaires TID_{smart} sont modifiées chaque session, d'où un adversaire ne peut pas différencier ou lier les différentes sessions d'un même utilisateur. De ce fait, notre proposition garantit l'anonymat et la non-traçabilité des utilisateurs.

5.6.3 Confidentialité

Dans notre système, les informations transmises sont les données personnelles de l'utilisateur, il est donc important d'empêcher les entités non autorisées de pouvoir lire les messages échangés. La transmission des données sensibles sans cryptage permettra à un adversaire d'observer facilement l'échange en cours. Cependant, le protocole proposé assure la confidentialité des données en transmettant les données sensibles comme l'ID_{smart} et le numéro de téléphone sous une forme cryptée. Par conséquent, les attaques par écoute clandestine ne sont pas possibles vu que la

confidentialité des données échangées est assurée.

5.6.4 Résistance aux attaques d'informations temporaires spécifiques à la session

Un schéma d'authentification ayant une clé de sécurité sécurisée doit résister contre les attaques d'informations temporaires spécifiques à la session même si ces informations comme les entiers aléatoires, les horodatages,etc. sont compromises. En effet, dans notre schéma, la révélation de l'entier aléatoire x ne permet pas à un adversaire de récupérer K_{smart} ou $K_{session}$ à partir de A_1 s'il est divulgué puisqu'il utilise un hachage à sens unique qui garantit la non-révélation des informations.

De même, la connaissance de x par un attaquant ne lui permet pas de révéler ID_{smart} ou TID_{smart} à partir de B_1 pour construire $K_{session}$ parce qu'il doit connaître b_s qui ne peut être calculé qu'avec un hachage de $K_{session}$ de la session précédente. Un adversaire doit aussi compromettre K_{smart}, $K_{session}$ et x pour pouvoir récupérer y à partir de A2.
La clé de session $K_{session}$ dépend aussi de ID_{smart} qui est un identifiant crypté avec PUB_{im}, de TID_{smart} qui est un hachage de ID_{smart} et $K_{session}$ de la session précédente et de K_{smart} qui utilise le hachage de ID_{smart} et N_{im}. Ce calcul rend l'attaque par un adversaire plus complexe, voire impossible. Par conséquent, la connaissance des entiers aléatoires spécifiques à la session ne compromet pas la clé de session.

5.6.5 Résistance aux attaques de rejeu

L'attaque par rejeu ne devient valide que si les informations transmises précédemment peuvent être utilisées de façon malicieuse. Cependant, dans le protocole proposé, si l'adversaire rejoue les messages 1, 2 et 4 de la phase d'authentification,

l'utilisation des horodatages T_1, T_2 et T_4 va empêcher toute sorte d'attaques de rejeu. Par conséquent, la transmission des messages validés avec des horodatages permet d'éviter ce type d'attaques. De plus, de chaque côté, une vérification de la légitimité de l'émetteur et de l'authenticité du message est faite. Le gestionnaire d'identité vérifie si $h(ID_{smart} \| \mathrm{TID}_{smart} \| \mathrm{b}_s \| \ x \ \| \mathrm{T}_1) = B_1$ pour valider l'authenticité du téléphone intelligent. Ce dernier vérifie si $h(K_{session}) = \mathrm{B2}$ pour confirmer que le message est authentique et dissiper toute probabilité d'attaque par rejeu. Et, enfin le réseau hospitalier vérifie si $h(\mathrm{Numero} \| \ z \ \| \mathrm{T}_4) \stackrel{?}{=} C$ afin de s'assurer que les informations proviennent du gestionnaire d'identité.

5.6.6 Résistance aux attaques par usurpation d'identité du gestionnaire d'identité

L'attaque par usurpation d'identité consiste à ce qu'un tiers utilise à des fins illégales l'identité d'une autre personne en faisant semblant d'être une source de confiance. Un adversaire voulant imiter le comportement du gestionnaire d'identité et usurper son identité et ayant compromis la clé secrète du téléphone intelligent K_{smart} n'est pas en mesure de construire un message de réponse valide contenant A_2, B_2, T_2 et TID_{smart} puisqu'il ne connaît ni x ni $K_{session}$ pour la construction de A_2 et B_2. Par conséquent, même en connaissant K_{smart}, notre protocole résiste aux attaques d'imitation de clé.

5.6.7 Résistance aux attaques par usurpation d'identité

Cette attaque consiste à ce qu'un adversaire se fait passer pour un gestionnaire d'identité, un réseau hospitalier ou un téléphone intelligent vérifié comme entité légitime en modifiant les messages. En effet, les paramètres (A_1, B_1) (A_2, B_2) et (A_3, C) utilisés dans les messages ne peuvent être construits sans la connaissance de (K_{smart} et $K_{session}$), (N_{im} et $K_{session}$) et N_h respectivement. En outre, l'utilisation de cryptage, du hachage et de l'horodatage renforce encore plus le protocole

contre les attaques par usurpation d'identité.

5.6.8 Sécurité persistante

Le protocole de sécurité proposée génère une clé unique pour chaque session qui est composée de :

— nombres aléatoires générés indépendamment pour chaque session ;

— hachage à sens unique ;

— d'autres paramètres comme K_{smart} , ID_{smart}, etc.

La non-disponibilité de l'un des paramètres cités ci-haut nécessaires pour la construction de $K_{session}$ paralyse et bloque l'attaquant. Cependant, nous pouvons affirmer que notre protocole génère les clés secrètes du fait que la compromission de la clé de session actuelle ne permet pas de révéler la clé de session précédente et la révélation de la clé de session précédente ne révèle pas la clé de session future.

5.6.9 Sécurité des clés de session

Cette caractéristique consiste à empêcher la fuite de toute clé de session antérieure même si la clé secrète d'un utilisateur légitime est révélée. De ce fait, si la clé N_{im} qui est utilisé pour calculer la clé secrète du téléphone intelligent K_{smart} est divulguée à l'adversaire, un attaquant n'est pas en mesure de calculer les clés de session précédente, car il n'a pas accès à l'entier y.

5.6.10 Fraîcheur des clés de session

Dans notre protocole, chaque $K_{session}$ est composée par des nombres aléatoires et un horodatage qui offrent une fraîcheur à la clé pour chaque session de communication. Par conséquent, cela permet de générer une clé exclusive pour chaque session, ce qui permet de garantir la fraîcheur de la clé.

5.7 Conclusion

À travers ce chapitre, nous avons fourni un aperçu sur l'état de l'art en relation avec l'intégration du paradigme ICN dans une architecture basée sur SDN. Ensuite, nous avons présenté notre modèle de système en passant par un modèle réel de notre cas d'utilisation jusqu'aux différentes étapes du scénario. Nous avons aussi présenté au cours de ce chapitre, nos solutions afin de satisfaire les exigences attendues par notre solution. Enfin, nous avons établi une analyse de sécurité afin de vérifier les fonctionnalités assurées par notre protocole.

CHAPITRE VI

ÉVALUATION DES PERFORMANCES

6.1 Introduction

Dans ce chapitre, une évaluation des performances de notre solution m-santé en termes de temps de latence a été effectuée en établissant : (i) une évaluation par simulation en mettant en place divers scénarios et (ii) une évaluation analytique dans laquelle nous avons établi une modélisation par file d'attente pour les différents équipements utilisés. Nous présentons ces deux évaluations, les paramètres utilisés ainsi que les résultats obtenus dans les sections suivantes.

6.2 Évaluation par simulation

6.2.1 Environnement de simulation

Pour les besoins de nos simulations, nous implémentons notre solution m-santé à travers l'outil Mininet-Wifi (Mininet-Wifi, 2021) qui permet la création d'un réseau virtuel. En effet, en utilisant une seule machine et le langage Python, Mininet-Wifi exécute le code des applications sur des commutateurs, des stations sans fil, etc. Nous avons aussi utilisé le contrôleur SDN Ryu (RYU, 2021) qui utilise aussi le langage Python et qui prend en charge plusieurs protocoles de gestion des périphériques réseau tels que OpenFlow.

Dans notre simulation, nous avons utilisé deux types d'interfaces de communica-

tion :

— Sans fil : entre le téléphone intelligent et le point d'accès ;

— Filaire : entre les différents équipements du réseau universitaire, et aussi entre les périphériques du réseau hospitalier.

Afin de mettre en place nos solutions proposées en termes de protocole de sécurité et d'intégration de NFN avec SDN, nous avons modifié le contrôleur RYU pour qu'il soit capable de supporter les différentes fonctions relatives au schéma de sécurité géré par le gestionnaire d'identité. De plus, nous avons mis en place des fonctions capables de créer des paquets UDP comme proposé dans la solution afin de pouvoir utiliser NFN dans le point d'accès et dans les commutateurs OF-NFN. Des fonctions ont été aussi ajoutées dans les commutateurs OF-NFN qui permettent la vérification de la température, la saturation en oxygène, et la fréquence cardiaque au sein du réseau universitaire ainsi que l'examen du dossier hospitalier et des antécédents dans le réseau hospitalier.

Finalement, les fonctionnalités du point d'accès ont été étendues pour qu'il puisse dupliquer les données physiologiques reçues par l'utilisateur et les envoyer vers les commutateurs OF-NFN, et aussi pour qu'il prenne la décision et valide les résultats obtenus.

6.2.2 Scénarios de simulation

Dans ce travail, nous nous sommes concentrés principalement sur l'étude de l'impact des différentes technologies utilisées à savoir NFN et SDN sur les performances des applications en termes de temps de réponse. Plus précisément, nous examinons les métriques suivantes pour évaluer l'efficacité de nos solutions proposées :

— Le temps de réponse du protocole de sécurité aux requêtes des utilisateurs afin de permettre leurs authentifications ;

— Le temps de réponse du réseau universitaire aux requêtes des utilisateurs qui consiste au temps nécessaire pour que toutes les personnes reçoivent une réponse sans passer par le réseau hospitalier (aucun cas n'est suspect) ;

— Le temps de réponse des réseaux universitaire et hospitalier aux requêtes des utilisateurs. C'est le temps de réponse nécessaire afin que tous les utilisateurs aient une réponse s'ils sont tous des cas suspects ;

— Le temps de latence global du système pour montrer le potentiel d'extensibilité à mesure que le nombre de requêtes augmente. Cette métrique représente l'authentification de tous les utilisateurs ainsi que leurs passages par les réseaux universitaire et hospitalier afin de s'assurer de leurs antécédents.

Afin d'évaluer les performances de nos solutions proposées, nous considérons les scénarios suivants :

- Scénario 1 : Nous avons varié le nombre de CPU virtuels (en anglais, *virtual CPU* (vCPU)) alloués à la machine virtuelle afin de voir son impact sur le temps de réponse pour l'exécution (i) du protocole de sécurité et (ii) des fonctions d'analyse dans les commutateurs OF-NFN du réseau universitaire.
- Scénario 2 : Dans le but d'évaluer le temps de latence de notre solution basée sur NFN et SDN, nous avons mis en place une solution m-santé dans l'informatique en périphérie afin de comparer les résultats obtenus par les deux solutions. L'architecture de ce scénario est présentée dans la figure 6.1 et elle se compose d'un téléphone intelligent de l'utilisateur, d'un point d'accès permettant la liaison du téléphone intelligent avec le réseau universitaire ainsi que l'envoi des données physiologiques vers un serveur edge. Et finalement, d'un serveur edge qui analyse les données physiologiques des utilisateurs en

exécutant les fonctions simultanément. Nous avons aussi varié le nombre de ressources CPU alloués aux équipements dans Mininet-Wifi afin d'évaluer les variations et l'impact de ces derniers en termes de temps de latence.

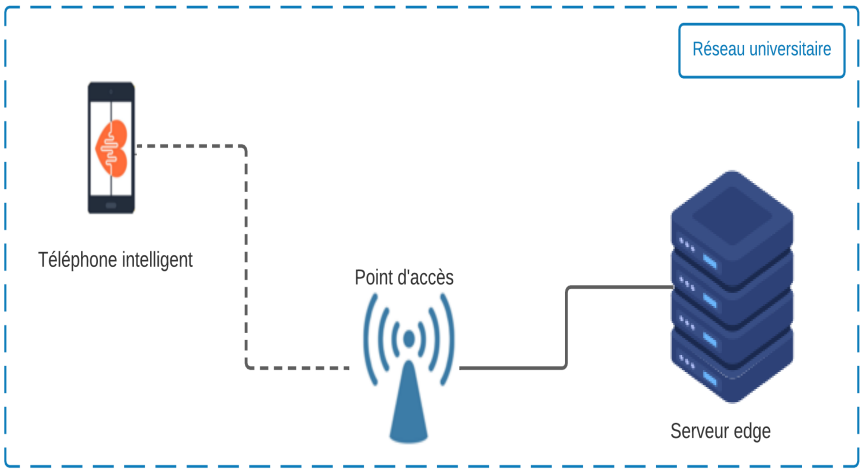

Figure 6.1: Scénario 2 de la simulation

- Scénario 3 : À l'issue de ce scénario, nous cherchons à évaluer la performance de notre solution en termes de temps de latence global pour les réseaux universitaire et hospitalier. Pour ce faire, nous avons comparé notre solution avec une version étendue du scénario 2. Outre le réseau universitaire dans l'informatique en périphérie, le scénario 3 contient aussi un réseau hospitalier qui se compose d'un commutateur permettant la liaison des réseaux universitaire et hospitalier, et d'un serveur de périphérie qui effectue l'examen du dossier hospitalier et des antécédents. Nous avons considéré le pire des scénarios, c'est-à-dire que toutes les personnes sont suspectées d'avoir le COVID-19, d'où le passage obligatoire par le réseau hospitalier. Ce scénario est présenté dans la figure 6.2 .
- Scénario 4 : Finalement, nous avons modélisé le temps de latence global nécessaire pour qu'un utilisateur puisse accéder au campus universitaire en

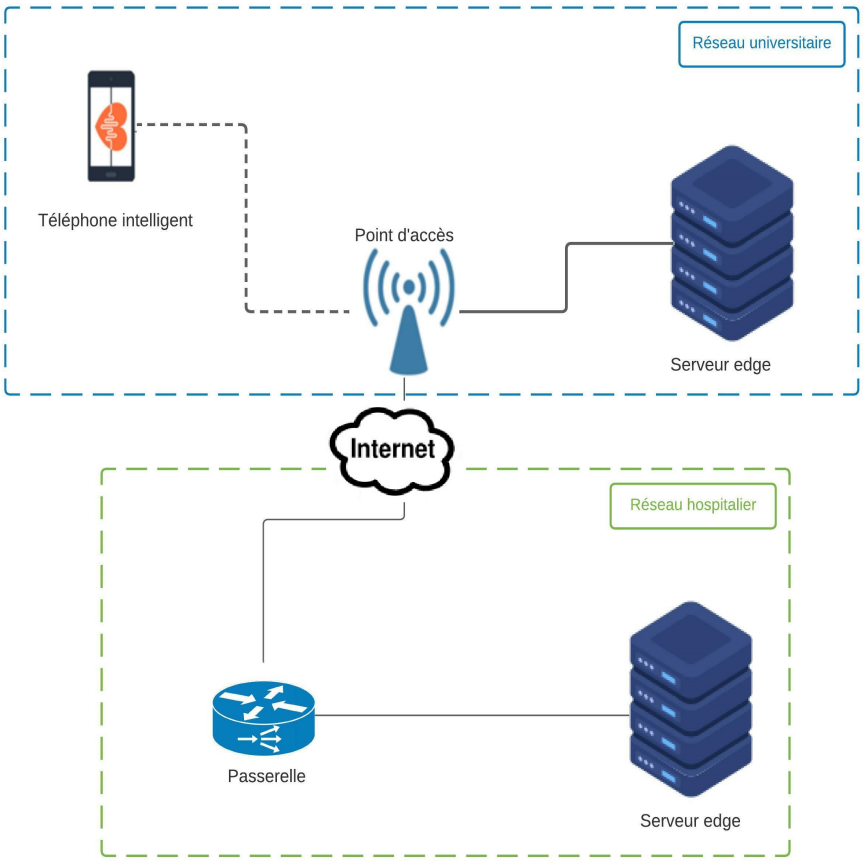

Figure 6.2: Scénario 3 de la simulation

établissant toutes les étapes du protocole de sécurité et en effectuant une analyse dans le réseau universitaire et hospitalier. De même, dans ce scénario, nous avons considéré que toutes les personnes sont suspectées d'avoir le COVID-19 et une vérification auprès du réseau hospitalier doit être faite.

6.2.3 Analyse des résultats

Les figures 6.3 et 6.4 présentent les résultats obtenus pour le scénario 1 dans lequel nous avons varié le nombre de vCPUs alloués à la machine virtuelle (de 1 jusqu'à 4). En effet, les performances de la machine virtuelle augmentent si nous

attribuons davantage de processeurs, ce qui va permettre par conséquent de traiter un nombre élevé de tâches simultanément. Plus spécifiquement, nous analysons dans la figure 6.3 le temps de réponse du protocole de sécurité qui inclut le temps de communication ainsi que le temps d'exécution.

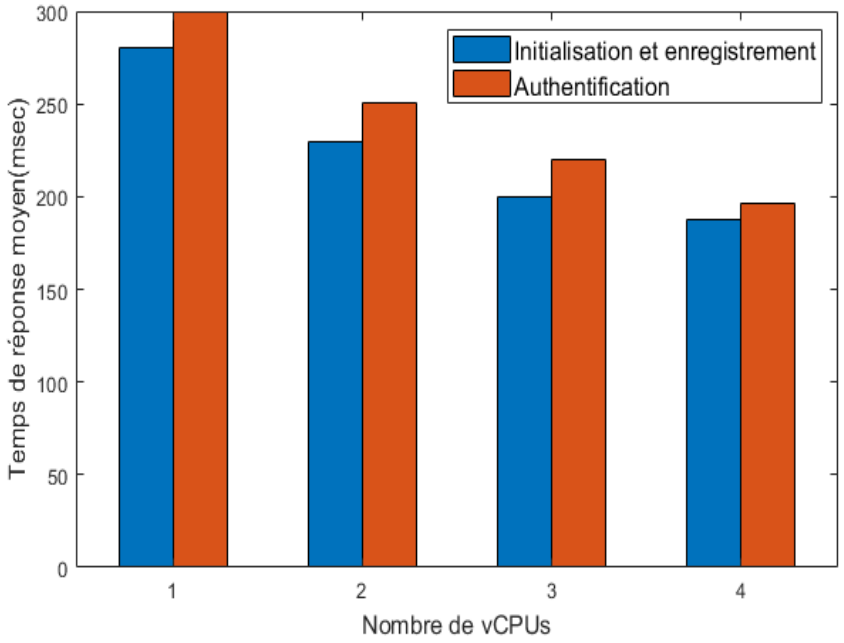

Figure 6.3: Variation du temps de réponse du protocole de sécurité en fonction du nombre de vCPUs

Nous notons une corrélation négative entre le nombre de vCPUs alloués et le temps de réponse du protocole de sécurité. Nous pouvons ainsi remarquer qu'à mesure que nous augmentons le nombre de vCPUs alloués, le temps d'exécution du protocole de sécurité diminue. Les résultats obtenus considèrent deux phases : la $1^{\text{ère}}$ phase qui est l'initialisation des paramètres et l'enregistrement, et la $2^{\text{ème}}$ phase qui est l'authentification de l'utilisateur au sein des réseaux universitaire et hospitalier.

Dans la figure 6.4, nous évaluons le temps de réponse du réseau universitaire. Dans

ce cadre, nous avons varié le nombre d'utilisateurs envoyant des données physiologiques (de 25 jusqu'à 143 utilisateurs). Une corrélation inverse entre le nombre de vCPUs et le temps de réponse est également notée. Par conséquent, l'impact du nombre de vCPUs est évident en termes d'amélioration des performances réseau.

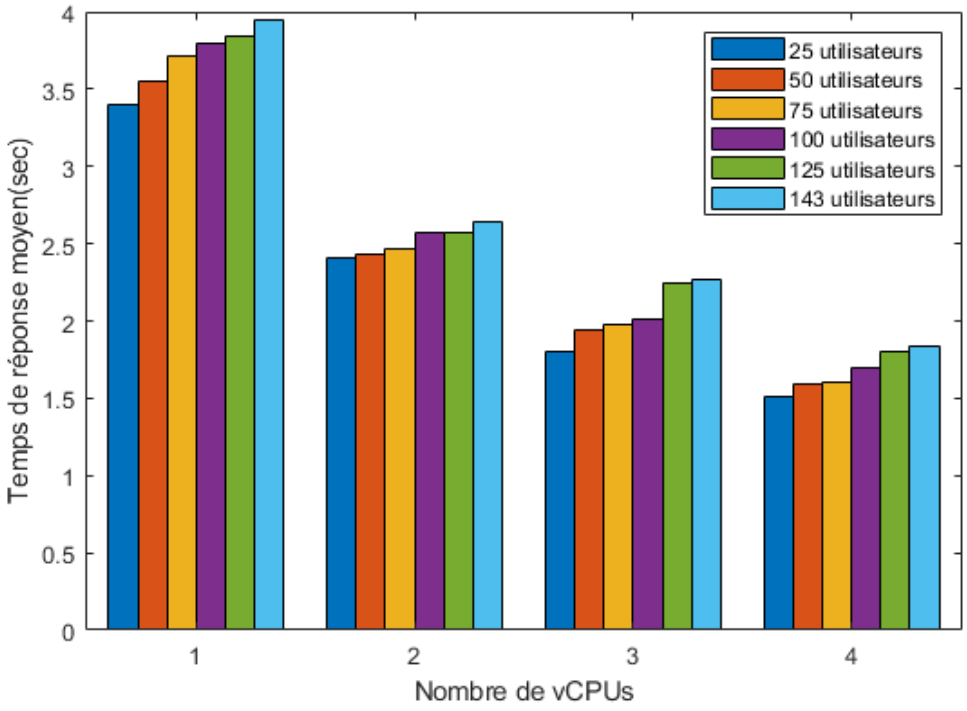

Figure 6.4: Variation du temps de réponse du réseau universitaire en fonction du nombre de vCPUs

Les résultats du scénario 2 sont présentés dans la figure 6.5. Nous avons choisi d'évaluer les performances de ce scénario en fixant le nombre de vCPUs alloués à 4 afin d'obtenir les meilleurs résultats. De plus, nous avons varié le nombre de ressources CPU pour les équipements utilisés sous Mininet-Wifi (0.5%, 1.5% et 1.5% de vCPUs alloués à la machine virtuelle). Nous remarquons que pour un CPU=0.5% et 1%, le scénario de l'informatique en périphérie devient de plus en plus mauvais en termes de temps de réponse moyen au fur et à mesure que le

nombre des utilisateurs augmente. Cependant, NFN offre un meilleur temps de réponse moyen ainsi qu'une évolution plus stable pour les deux premières valeurs de CPU. En revanche, pour un CPU= 1.5% le temps de réponse moyen dans NFN dépasse celui du scénario de l'informatique en périphérie, ce qui explique que NFN est plus adapaté pour les équipements à faibles ressources.

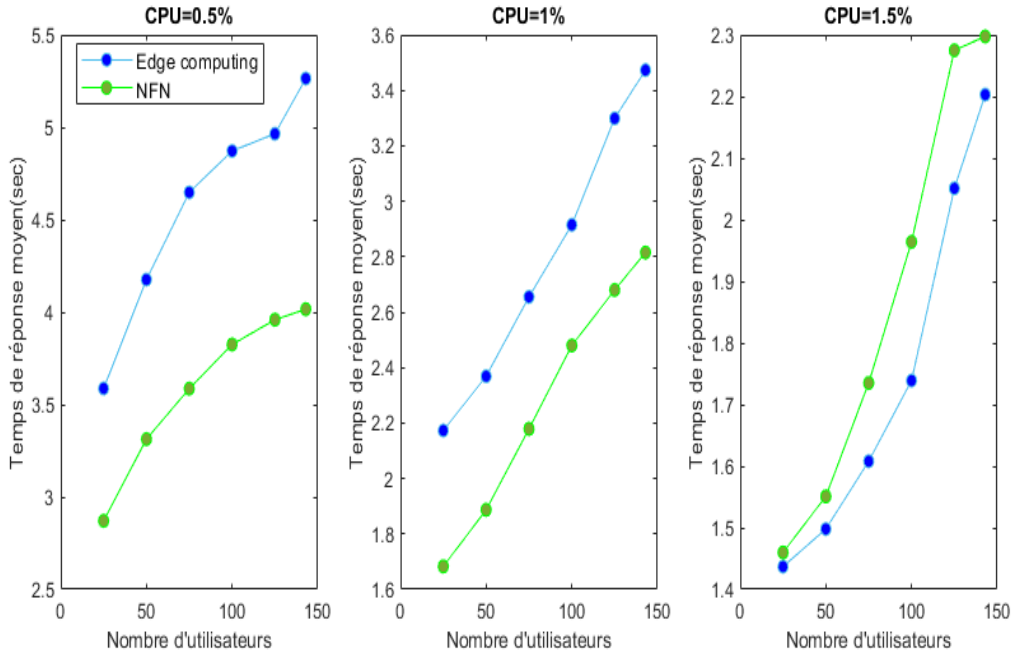

Figure 6.5: Variation du temps de réponse en fonction du nombre de CPU et du nombre d'utilisateurs

Sur la figure 6.6 qui représente le scénario 3, nous comparons le temps de latence global pour les réseaux universitaire et hospitalier dans notre solution basée sur NFN et SDN avec celui de l'informatique en périphérie. En effet, le temps de latence global correspond au temps de communication ainsi que le temps d'exécution des fonctions de collecte, d'analyse et de détection. Nous constatons une nette supériorité en termes de performance des résultats du NFN ce qui correspond avec notre objectif à savoir offrir un temps de réponse compatible avec les

besoins des solutions de m-santé.

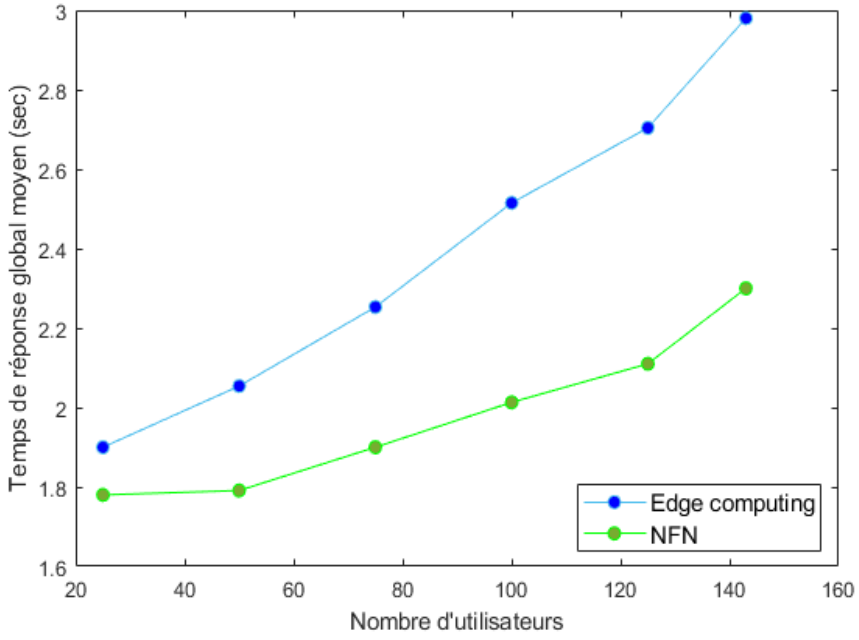

Figure 6.6: Comparaison temps de latence global moyen NFN et informatique en périphérie

Les résultats du dernier scénario sont illustrés dans la figure 6.7 qui évalue le temps de réponse global de la solution m-santé proposée. En effet, les résultats montrent que même pour une phase de surcharge du système avec 143 utilisateurs, la solution offre une réponse en 2,3 secs pour qu'une personne soit complètement authentifiée auprès du réseau universitaire et hospitalier et pour que toutes les analyses soient faites. En une phase normale ou peu chargée, une personne sera authentifiée et aura une décision concernant son accès au campus en 1,7 sec approximativement. Ces valeurs correspondent aux attentes en termes de temps de réponse moyen de notre solution d'analyse et de détection offrant ainsi une authentification et un accès en un temps faible.

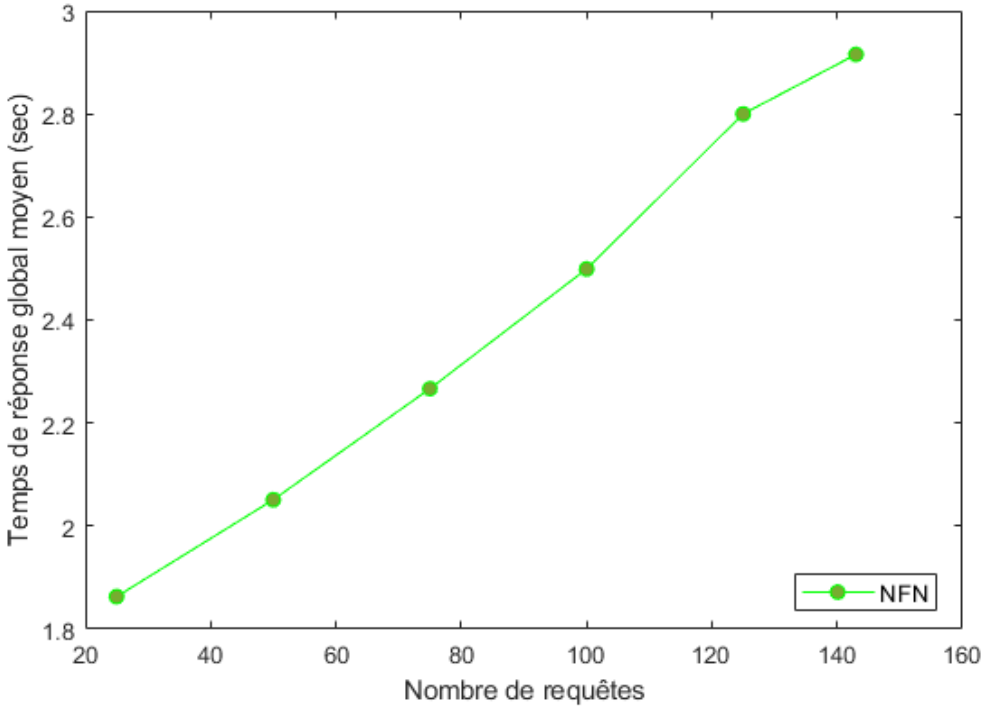

Figure 6.7: Temps de latence global moyen du système

6.3 Évaluation analytique

Dans le but de formuler et étudier le système d'analyse et de détection m-santé proposé, nous considérons une modélisation par file d'attente présentée dans la figure 6.8. Nous supposons dans notre modèle que les arrivées sont représentées par un processus de Markov avec un flux Poisson. Le taux d'arrivée moyen des données physiologiques représenté par λ, c'est-à-dire le temps d'interarrivée de chaque service de santé est indépendant et distribué de façon exponentielle. Et le taux de service moyen par serveur est donné par μ. À l'exception des commutateurs OF-NFN du campus universitaire qui ont un taux d'arrivée moyen $\lambda' = \frac{K}{N}\ \lambda$ étant donné qu'ils suivent un modèle «fork-join» où K représente le nombre de sous tâches et N le nombre total des serveurs fork-join. Après l'analyse des données

physiologiques par les commutateurs OF-NFN, les résultats obtenus sont dirigés vers la 2ème file d'attente du point d'accès afin de prendre la décision d'envoyer des données vers les le réseau hospitalier avec une probabilité P_h si la personne est suspecte, ou d'ouvrir les portes d'entrée du campus avec une probabilité 1-P_h.

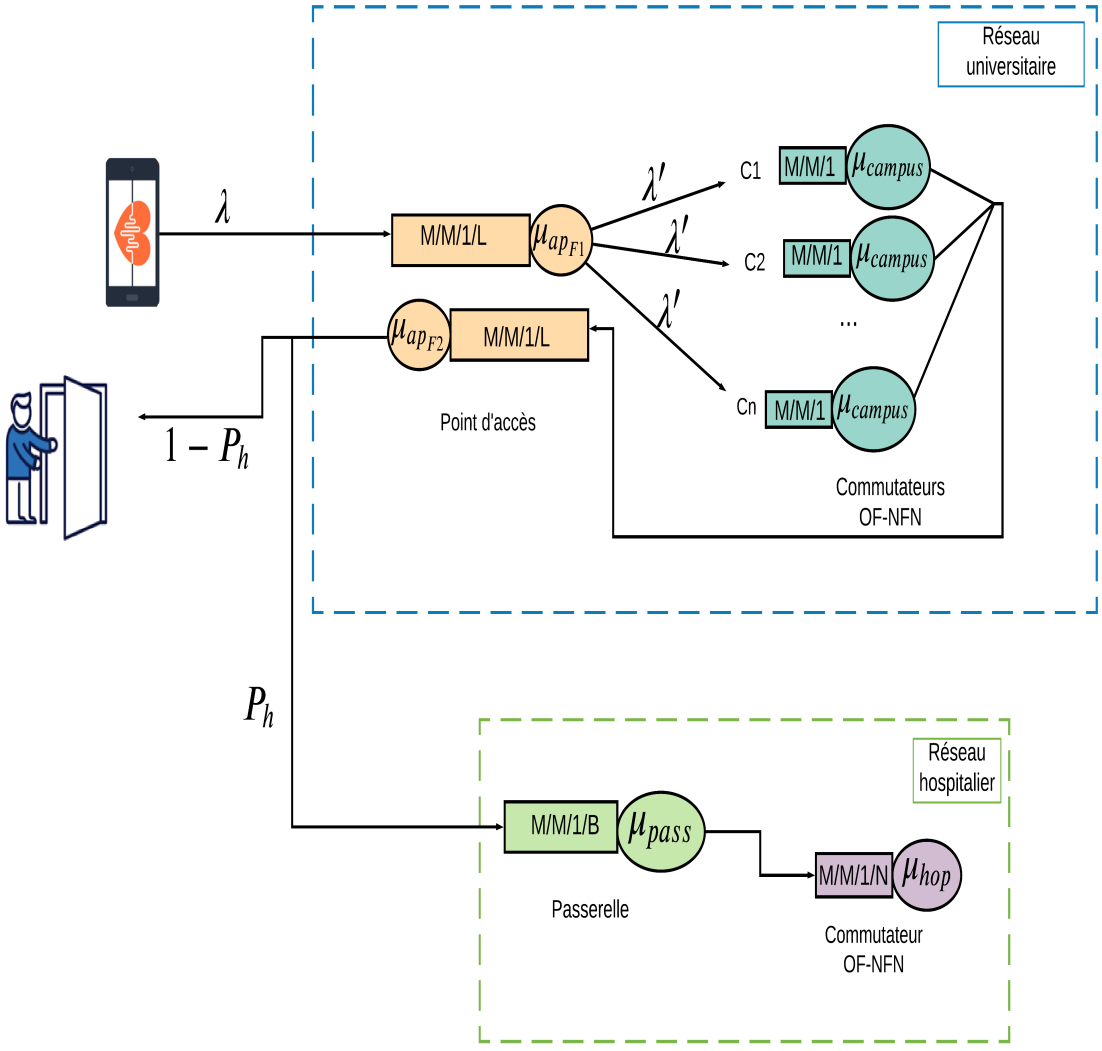

Figure 6.8: Modèle de mise en file d'attente du système proposé

6.3.1 Temps de réponse moyen du point d'accès

Le point d'accès se compose de deux files d'attente modélisée chacune par une queue M/M/1/L ayant un seul serveur, une taille de file d'attente égale à L et un taux de service moyen μ_{ap}. L'intensité du trafic d'une file du point d'accès est calculée par $\rho_{ap} = \frac{\lambda}{\mu_{ap}}$. Pour pouvoir exprimer le temps de réponse moyen d'une file d'attente du point d'accès, nous devons utiliser les équations suivantes :

- La probabilité de blocage c'est-à-dire le taux de perte est calculée comme suit :

$$\Pi_{B.ap} = \begin{cases} \frac{1}{L+1} & \text{si } \rho_{ap} = 1 \\ \frac{(1-\rho_{ap})\rho_{ap}^{L}}{1-\rho_{ap}^{L+1}} & \text{sinon.} \end{cases} \tag{6.1}$$

- Le débit effectif du point d'accès est donné par :

$$\lambda_{eff.ap} = \lambda(1 - \Pi_{B.ap}) \tag{6.2}$$

- Le nombre moyen des requêtes qui sont en attente et en cours de traitement est calculé par :

$$E_{ap} = \sum_{i=1}^{L} \Pi_i = \begin{cases} \frac{L}{2} & \text{si } \rho_{ap} = 1 \\ \frac{\rho_{ap}}{1-\rho_{ap}} - \frac{(L+1)\rho_{ap}^{L+1}}{1-\rho_{ap}^{L+1}} & \text{sinon.} \end{cases} \tag{6.3}$$

- Finalement, le temps moyen de réponse du point d'accès est obtenu par :

$$T_{ap} = \frac{E_{ap}}{\lambda_{eff.ap}} \tag{6.4}$$

6.3.2 Temps de réponse moyen des commutateurs OF-NFN du campus universitaire

Les commutateurs OF-NFN du campus universitaire permettent de vérifier l'état d'une personne en analysant ses données physiologiques. En effet, le taux d'arrivée λ au point d'accès est divisé sur plusieurs commutateurs pour que chacun

puisse vérifier une donnée physiologique et bénéficier d'un traitement en parallèle. Cependant, les commutateurs OF-NFN du campus universitaire ont un taux d'arrivée moyen λ' = $\frac{K}{N}$ λ. Le taux de service moyen dans ces commutateurs est donné par μ_{campus} et l'intensité du trafic est calculée par $\rho_{campus} = \frac{\lambda'}{\mu_{campus}}$.

En se basant sur le travail de (Varki *et al.*, 2008), le temps de réponse approximatif pour les commutateurs OF-NFN disponibles dans le réseau universitaire est calculé comme suit :

$$T_{campus} \approx \frac{1}{\mu_{campus}}(H_N + \frac{\rho_{campus}}{2 * (1 - \rho_{campus})}\Big(Sum_{N-\rho_{campus}} + (1 - 2\rho_{campus} * Sum_{N(N-\rho_{campus})})\Big) \tag{6.5}$$

où :

- H_N représente un nombre harmonique calculé comme suit : $1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{N}$
- $Sum_{N-\rho_{campus}}$ et $Sum_{N(N-\rho_{campus})}$ sont des sommes partielles obtenues comme suit :
  $Sum_{N-\rho_{campus}} = \frac{1}{1-\rho_{campus}} + \frac{1}{2-\rho_{campus}} + \frac{1}{3-\rho_{campus}} + ... + \frac{1}{N-\rho_{campus}}$
  $Sum_{N(N-\rho_{campus})} = \frac{1}{1-\rho_{campus}} + \frac{1}{2}\ \frac{1}{2-\rho_{campus}} + \frac{1}{3}\ \frac{1}{3-\rho_{campus}} + ... + \frac{1}{N}\ \frac{1}{N-\rho_{campus}}$

6.3.3 Temps de réponse moyen de la passerelle

La passerelle reliant le réseau du campus universitaire avec le réseau hospitalier est représentée par une file M/M/1/B qui se compose d'un seul serveur, d'une taille de file d'attente égale à B et un taux de service moyen μ_{pass}. L'intensité du trafic de la passerelle est calculée par $\rho_{pass} = \frac{\lambda}{\mu_{pass}}$. Nous devons utiliser les équations suivantes afin de trouver le temps de réponse moyen de la passerelle.

- La probabilité de blocage c'est-à-dire le taux de perte est calculée comme

suit :

$$\Pi_{B.pass} = \begin{cases} \frac{1}{B+1} & \text{si } \rho_{pass} = 1 \\ \frac{(1-\rho_{pass})\rho_{pass}^{B}}{1-\rho_{pass}^{B+1}} & \text{sinon.} \end{cases} \tag{6.6}$$

- Le débit effectif de la passerelle est donné par :

$$\lambda_{eff.pass} = \lambda(1 - \Pi_{B.pass}) \tag{6.7}$$

- Le nombre moyen des requêtes qui sont en attente et en cours de traitement est calculé par :

$$E_{pass} = \sum_{i=1}^{B} \Pi_i = \begin{cases} \frac{B}{2} & \text{si } \rho_{pass} = 1 \\ \frac{\rho_{pass}}{1-\rho_{pass}} - \frac{(B+1)\rho_{pass}^{B+1}}{1-\rho_{pass}^{B+1}} & \text{sinon.} \end{cases} \tag{6.8}$$

- Finalement, le temps moyen de réponse de la passerelle est obtenu par :

$$T_{pass} = \frac{E_{pass}}{\lambda_{eff.pass}} \tag{6.9}$$

6.3.4 Temps de réponse moyen du commutateur OF-NFN du réseau hospitalier

Le commutateur OF-NFN disponible dans le réseau hospitalier permettant de vérifier l'historique médical d'une personne suspecte est modélisé par une queue M/M/1/N ayant un seul serveur, une taille de file d'attente représentée par la variable N et d'un taux de service moyen μ_{hop}. L'intensité du trafic de la passerelle est calculée par $\rho_{hop} = \frac{\lambda}{\mu_{hop}}$. Afin de pouvoir calculer le temps de réponse moyen du commutateur OF-NFN disponible dans le réseau hospitalier, ces équations doivent être résolues.

- La probabilité de blocage c'est-à-dire le taux de perte calculée comme suit :

$$\Pi_{B.hop} = \begin{cases} \frac{1}{N+1} & \text{si } \rho_{hop} = 1 \\ \frac{(1-\rho_{hop})\rho_{hop}^{N}}{1-\rho_{hop}^{N+1}} & \text{sinon.} \end{cases} \tag{6.10}$$

- Le débit effectif du commutateur OF-NFN est donné par :

$$\lambda_{eff.hop} = \lambda(1 - \Pi_{B.hop}) \tag{6.11}$$

- Le nombre moyen des requêtes qui sont en attente et en cours de traitement est calculé par :

$$E_{hop} = \sum_{i=1}^{N} \Pi_i = \begin{cases} \frac{N}{2} & \text{si } \rho_{hop} = 1 \\ \frac{\rho_{hop}}{1-\rho_{hop}} - \frac{(N+1)\rho_{hop}^{N+1}}{1-\rho_{hop}^{N+1}} & \text{sinon.} \end{cases} \tag{6.12}$$

- Le temps moyen de réponse du commutateur OF-NFN est finalement obtenu par :

$$T_{hop} = \frac{E_{hop}}{\lambda_{eff.hop}} \tag{6.13}$$

6.3.5 Temps de réponse moyen global

En se basant sur les équations présentées précédemment, et étant donné que la deuxième file d'attente peut : (i) envoyer les données physiologiques vers le réseau hospitalier avec une probabilité P_h si l'historique médical de la personne doit être vérifié (ii) ou envoyer la décision d'ouverture de porte avec une probabilité 1-P_h, le temps de réponse moyen global du système proposé est calculé comme suit :

$$T_{global} = T_{ap.F1} + T_{campus} + T_{ap.F2} + P_h(2 * T_{pass} + T_{hop} + T_{ap.F2}) \tag{6.14}$$

où $T_{ap.F1}$ et $T_{ap.F2}$ représentent le temps de réponse moyen de la 1ère et la 2ème file du point d'accès respectivement.

Afin d'évaluer mathématiquement notre solution d'analyse et de détection m-santé, nous supposons que le taux d'arrivée des requêtes de données de santé λ varie de 25 à 143 requêtes par seconde. Le temps de service de chaque donnée de

santé dans le point d'accès pour la 1ère file et la 2ème file est $\frac{1}{\mu_{ap}} = 0{,}0014$ sec. Les commutateurs OF-NFN du réseau universitaire, la passerelle et les commutateurs OF-NFN du réseau hospitalier ont un temps de service $\frac{1}{\mu_{campus}} = 0{,}0058$ sec, $\frac{1}{\mu_{pass}} = 0{,}02$ sec et $\frac{1}{\mu_{hop}} = 0{,}1$ sec respectivement. Nous supposons aussi que L=200, B=40 et N=10.

Nous avons varié aussi la probabilité P_h de 0,2 jusqu'à 1 dans le but de voir les différentes variations en augmentant le nombre de personnes suspectes de COVID-19. En se basant sur le travail de (Vergados *et al.*, 2006), nous considérons que la taille des paquets contenant les données physiologiques (température, saturation en oxygène et fréquence cardiaque) est égale à 1048 bits.

Dans le but de comparer notre modélisation mathématique avec le principe de simulation du scénario 4 mais en variant P_h, nous avons présenté les résultats numériques obtenus dans la figure 6.9. Cependant, nous pouvons remarquer que les résultats de simulation correspondent avec les résultats analytiques avec une légère variation étant donné que les résultats sont une moyenne de plusieurs exécutions.

6.4 Conclusion

Dans ce chapitre, nous avons montré à travers des scénarios de simulation que notre solution d'analyse et de détection m-santé basée sur NFN et SDN offre de meilleurs résultats en termes de temps de réponse par rapport à un scénario basé sur l'informatique en périphérie que nous avons mis en place. Finalement, en établissant une modélisation par file d'attente des différents équipements de notre solution, nous avons conclu que les résultats numériques se concordent avec les résultats de simulations réalisées.

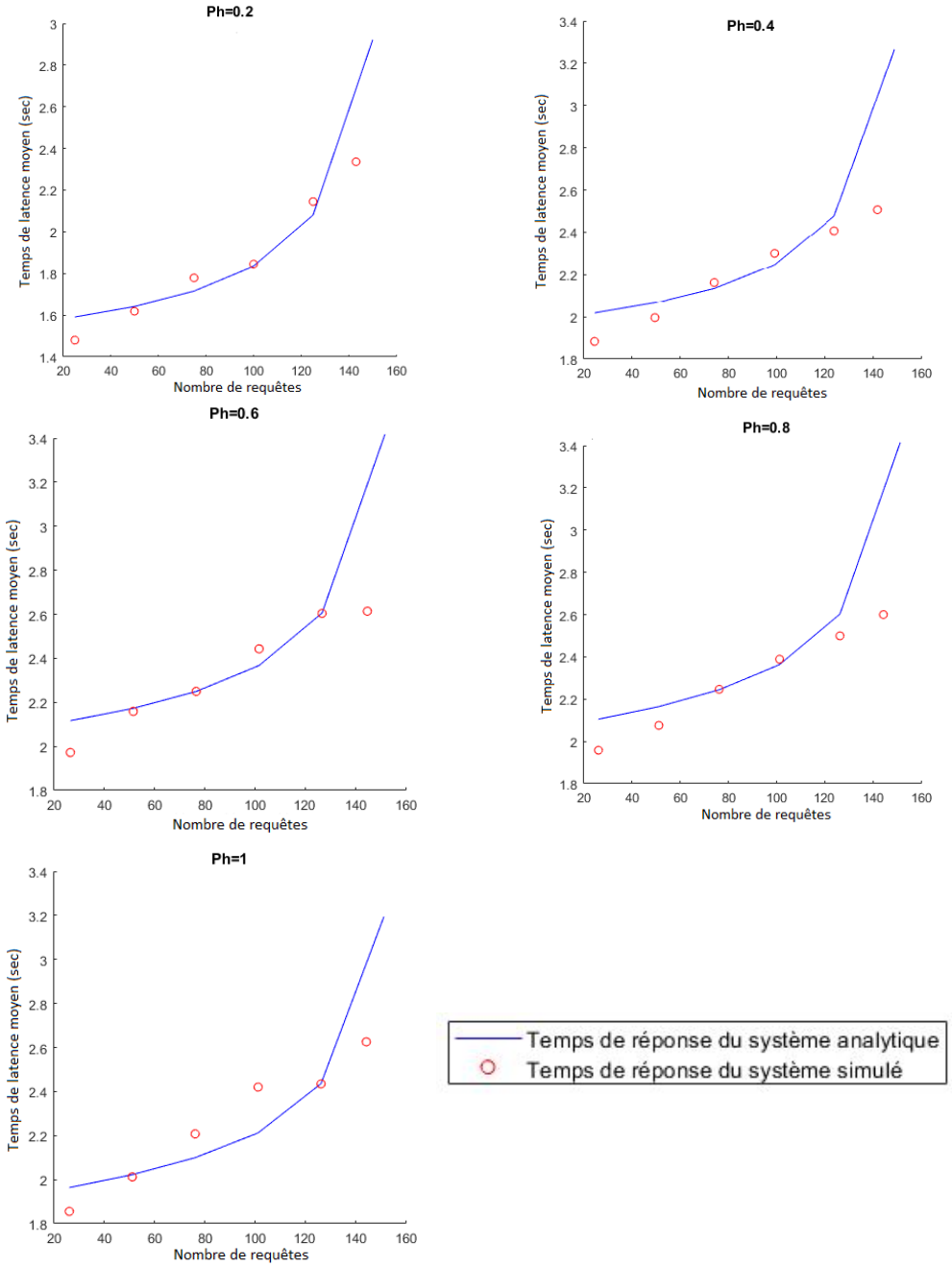

Figure 6.9: Temps de réponse : analytique vs simulation

CHAPITRE VII

CONCLUSION

Pendant les dernières années, l'Internet des objets s'est considérablement incrusté dans la vie quotidienne des utilisateurs étant donné le confort et la facilité d'utilisation apportés principalement dans le domaine de la santé. Compte tenu des circonstances de la pandémie que nous sommes en train de vivre, le nombre des utilisateurs de l'IdO médical et particulièrement de la santé mobile a explosé générant ainsi un volume de données énorme qui doit être collecté, analysé et stocké dans les plus brefs délais afin d'offrir une prise de décision médicale fiable.

Malheureusement, l'infonuagique ainsi que l'informatique en périphérie se trouvent incapables de répondre aux besoins des utilisateurs en termes de temps de réponse opportun et de satisfaction des exigences des applications m-santé en raison de l'augmentation du trafic dans le réseau et de la grande distance par rapport à l'utilisateur final. Par ailleurs, les applications m-santé exigent un haut niveau de sécurité, de confidentialité et de vie privée, ce qui ne peut être assuré ni dans l'infonuagique ni dans l'informatique en périphérie du fait qu'ils sont exposés à un grand nombre de menaces, de failles et d'attaquants.

À cet effet, le réseau des fonctions nommées (NFN) est récemment proposé dans la littérature afin d'offrir une prise en charge des tâches sensibles au délai. NFN se base sur deux paradigmes : (i) INC qui offre une exécution des tâches IdO dans

les équipements réseau afin de se rapprocher l'utilisateur final et par conséquent réduire le trafic dans le réseau et (ii) ICN qui est une nouvelle architecture permettant d'offrir un service d'infrastructure réseau plus adéquat à l'Internet du futur et aux attentes des utilisateurs.

Fort de ce constat, et dans le contexte des pandémies, nous avons proposé dans de ce mémoire une solution m-santé basée sur NFN et SDN capable d'offrir un temps de réponse dans les plus brefs délais tout en assurant la fiabilité des résultats. Notre solution permet en effet de collecter, de transférer et d'analyser les données physiologiques générées en temps réel par les téléphones intelligents et de détecter les cas suspects porteurs de virus afin d'éviter la transmission à la chaine devant les portes d'entrée d'un espace public fermé dans lequel il existe un grand nombre de personnes. La solution garantit aussi la sécurité et la vie privée des utilisateurs en établissant un protocole de sécurité avant de commencer la transmission des données physiologiques.

À travers une analyse informelle, nous avons validé que le protocole de sécurité proposé assure plusieurs fonctionnalités, notamment l'authentification mutuelle, la confidentialité, la résistance aux attaques par usurpation d'identité, etc. Nous avons aussi mis des scénarios de simulations afin de comparer notre solution basée sur NFN et SDN avec un scénario déployé dans l'informatique en périphérie. Les évaluations de ces scénarios ont montré que notre solution surpasse les performances des autres approches. Finalement, une modélisation par file d'attente pour les équipements utilisés a prouvé que les résultats numériques et simulés se correspondent.

Nous envisageons dans nos travaux futurs de mettre en place une application d'équilibrage de charge afin de distribuer les paquets entre les différents commutateurs OF-NFN de façon équitable en nous basant sur la disponibilité des

commutateurs dans le but d'améliorer le temps de réponse global.

RÉFÉRENCES

Abdellatif, A. A., Mohamed, A., Chiasserini, C. F., Tlili, M. et Erbad, A. (2019). Edge computing for smart health : Context-aware approaches, opportunities, and challenges. *IEEE Network*, *33*(3), 196–203.

Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D. et Ohlman, B. (2012). A survey of information-centric networking. *IEEE Communications Magazine*, *50*(7), 26–36.

Ahuja, S. P., Mani, S. et Zambrano, J. (2012). A survey of the state of cloud computing in healthcare. *Network and Communication Technologies*, *1*(2), 12.

Akyildiz, I. F., Lee, A., Wang, P., Luo, M. et Chou, W. (2014). A roadmap for traffic engineering in SDN-OpenFlow networks. *Computer Networks*, *71*, 1–30.

Alzahrani, B. A., Irshad, A., Albeshri, A., Alsubhi, K. et Shafiq, M. (2020). An Improved Lightweight Authentication Protocol for Wireless Body Area Networks. *IEEE Access*, *8*, 190855–190872.

Amadeo, M., Campolo, C., Quevedo, J., Corujo, D., Molinaro, A., Iera, A., Aguiar, R. L. et Vasilakos, A. V. (2016). Information-centric networking for the internet of things : challenges and opportunities. *IEEE Network*, *30*(2), 92–100.

Amadeo, M., Ruggeri, G., Campolo, C. et Molinaro, A. (2019). IoT Services Allocation at the Edge via Named Data Networking : From Optimal Bounds to Practical Design. *IEEE Transactions on Network and Service Management*, *16*(2), 661–674.

Amadeo, M., Ruggeri, G., Campolo, C., Molinaro, A., Loscrí, V. et Calafate, C. T. (2019). Fog Computing in IoT Smart Environments via Named Data Networking : A Study on Service Orchestration Mechanisms. *Future Internet*, *11*(11), 222.

Aubry, E., Silverston, T. et Chrisment, I. (2015). SRSC : SDN-based routing scheme for CCN. Dans *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, 1–5. `http://dx.doi.org/10.1109/NETSOFT.2015.7116130`

Bracciale, L., Loreti, P., Detti, A., Paolillo, R. et Melazzi, N. B. (2019). Lightweight Named Object : An ICN-Based Abstraction for IoT Device Programming and Management. *IEEE Internet of Things Journal*, *6*(3), 5029–5039. http://dx.doi.org/10.1109/JIOT.2019.2894969

Braun, W. et Menth, M. (2014). Software-defined networking using OpenFlow : Protocols, applications and architectural design choices. *Future Internet*, *6*(2), 302–336.

Dang, L. M., Piran, M., Han, D., Min, K., Moon, H. *et al.* (2019). A survey on internet of things and cloud computing for healthcare. *Electronics*, *8*(7), 768.

Darwish, A., Hassanien, A. E., Elhoseny, M., Sangaiah, A. K. et Muhammad, K. (2019). The impact of the hybrid platform of internet of things and cloud computing on healthcare systems : opportunities, challenges, and open problems. *Journal of Ambient Intelligence and Humanized Computing*, *10*(10), 4151–4166.

De la Torre Díez, I., Alonso, S. G., Hamrioui, S., Cruz, E. M., Nozaleda, L. M. et Franco, M. A. (2019). IoT-based services and applications for mental health in the literature. *Journal of medical systems*, *43*(1), 1–6.

Detti, A., Caponi, A., Tropea, G., Bianchi, G. et Blefari-Melazzi, N. (2013). On the interplay among naming, content validity and caching in Information Centric Networks. Dans *2013 IEEE Global Communications Conference (GLOBECOM)*, 2108–2113. IEEE.

Dong, Y. et Yao, Y.-D. (2021). IoT platform for COVID-19 prevention and control : A survey. *IEEE Access*, *9*, 49929–49941.

DuBravac, S. et Ratti, C. (2015). The internet of things : evolution or revolution ? *American International Group*.

Dutta, N., Sarma, H. K. D., Jadeja, R., Delvadia, K. et Ghinea, G. (2021). Security in ICN. In *Information Centric Networks (ICN)* 119–137. Springer.

Eum, S., Jibiki, M., Murata, M., Asaeda, H. et Nishinaga, N. (2015). A design of an ICN architecture within the framework of SDN. Dans *2015 Seventh International Conference on Ubiquitous and Future Networks*, 141–146. http://dx.doi.org/10.1109/ICUFN.2015.7182521

Farahani, B., Firouzi, F., Chang, V., Badaroglu, M., Constant, N. et Mankodiya, K. (2018). Towards fog-driven IoT eHealth : Promises and challenges of IoT in medicine and healthcare. *Future Generation Computer Systems*, *78*, 659–676.

Found, O. N. (2015). Openflow switch specification version 1.5. 1 (Protocol version

0x06).

Gama, Ó., Carvalho, P., Afonso, J. A. et Mendes, P. (2008). Quality of service support in wireless sensor networks for emergency healthcare services. Dans *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1296–1299. IEEE.

Goransson, P., Black, C. et Culver, T. (2016). *Software defined networks : a comprehensive approach*. Morgan Kaufmann.

Goyal, S., Sharma, N., Kaushik, I., Bhushan, B. et Kumar, A. (2020). Precedence & issues of IoT based on edge computing. Dans *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, 72–77. IEEE.

Grewe, D., Nayak, N., Ambalavanan, U. et Schildt, S. (2020). Towards In-Network Computing Infrastructures for Connected Vehicles.

Gubbi, J., Buyya, R., Marusic, S. et Palaniswami, M. (2013). Internet of Things (IoT) : A vision, architectural elements, and future directions. *Future generation computer systems*, *29*(7), 1645–1660.

Gutierrez, L. J., Rabbani, K., Ajayi, O. J., Gebresilassie, S. K., Rafferty, J., Castro, L. A. et Banos, O. (2021). Internet of Things for Mental Health : Open Issues in Data Acquisition, Self-Organization, Service Level Agreement, and Identity Management. *International Journal of Environmental Research and Public Health*, *18*(3), 1327.

Hartmann, M., Hashmi, U. S. et Imran, A. (2019). Edge computing in smart health care systems : Review, challenges, and research directions. *Transactions on Emerging Telecommunications Technologies*.

Hu, Y. et Bai, G. (2014). A systematic literature review of cloud computing in eHealth. *arXiv preprint arXiv :1412.2494*.

Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H. et Braynard, R. L. (2009). Networking Named Content. p. 1–12., New York, NY, USA. Association for Computing Machinery. `http://dx.doi.org/10.1145/1658939.1658941`. Récupéré de `https://doi.org/10.1145/1658939.1658941`

Javdani, H. et Kashanian, H. (2018). Internet of things in medical applications with a service-oriented and security approach : a survey. *Health and Technology*, *8*(1), 39–50.

Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker,

S. et Stoica, I. (2007). A Data-Oriented (and beyond) Network Architecture. *37*(4), 181–192. http://dx.doi.org/10.1145/1282427.1282402. Récupéré de https://doi.org/10.1145/1282427.1282402

Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S. et Uhlig, S. (2014). Software-defined networking : A comprehensive survey. *Proceedings of the IEEE*, *103*(1), 14–76.

Ksentini, A., Jebalia, M. et Tabbane, S. (2021). IoT/cloud-enabled smart services : A review on QoS requirements in fog environment and a proposed approach based on priority classification technique. *International Journal of Communication Systems*, *34*(2).

Kumar, M. et Chand, S. (2020). A lightweight cloud-assisted identity-based anonymous authentication and key agreement protocol for secure wireless body area network. *IEEE Systems Journal*, *15*(2), 2779–2786.

Lara, A., Kolasani, A. et Ramamurthy, B. (2013). Network innovation using openflow : A survey. *IEEE communications surveys & tutorials*, *16*(1), 493–512.

Li, P., Muqing, W., Ning, W. et Hongbao, L. (2015a). Supporting information-centric networking in SDN. *International Journal of Future Computer and Communication*, *4*(6), 386.

Li, S., Da Xu, L. et Zhao, S. (2015b). The internet of things : a survey. *Information Systems Frontiers*, *17*(2), 243–259.

Lounis, A., Hadjidj, A., Bouabdallah, A. et Challal, Y. (2012). Secure and scalable cloud-based architecture for e-health wireless sensor networks. Dans *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, 1–7. IEEE.

Luthra, M., Koldehofe, B., Höchst, J., Lampe, P., Rizvi, A. H., Kundel, R. et Freisleben, B. (2019). INetCEP : In-Network Complex Event Processing for Information-Centric Networking. Dans *2019 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, 1–13.

Lyu, X., Tian, H., Jiang, L., Vinel, A., Maharjan, S., Gjessing, S. et Zhang, Y. (2018). Selective offloading in mobile edge computing for the green internet of things. *IEEE Network*, *32*(1), 54–60.

Masip-Bruin, X., Marin-Tordera, E., Jukan, A. et Ren, G.-J. (2018). Managing resources continuity from the edge to the cloud : Architecture and performance. *Future Generation Computer Systems*, *79*, 777–785.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. et Turner, J. (2008). OpenFlow : enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, *38*(2), 69–74.

Mininet-Wifi (2021). Mininet-Wifi. Récupéré le 06-02-2021 de `https://mininet-wifi.github.io/`

Networking, C. V. (2016). Cisco global cloud index : Forecast and methodology, 2015-2020. white paper. *Cisco Public, San Jose*.

Nguyen, X.-N., Saucez, D., Barakat, C. et Turletti, T. (2015). Rules placement problem in OpenFlow networks : A survey. *IEEE Communications Surveys & Tutorials*, *18*(2), 1273–1286.

Nguyen, X. N., Saucez, D. et Turletti, T. (2013). Providing CCN functionalities over OpenFlow switches.

Nunes, B. A. A., Mendonca, M., Nguyen, X.-N., Obraczka, K. et Turletti, T. (2014). A survey of software-defined networking : Past, present, and future of programmable networks. *IEEE Communications surveys & tutorials*, *16*(3), 1617–1634.

Rawat, D. B. et Reddy, S. R. (2016). Software defined networking architecture, security and energy efficiency : A survey. *IEEE Communications Surveys & Tutorials*, *19*(1), 325–346.

RYU (2021). RYU. Récupéré le 06-02-2021 de `https://ryu-sdn.org/`

Sapio, A., Abdelaziz, I., Aldilaijan, A., Canini, M. et Kalnis, P. (2017). In-network computation is a dumb idea whose time has come. Dans *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, 150–156.

Scherb, C., Emde, S., Marxer, C. et Tschudin, C. (2019a). Data upload in mobile edge computing over icn. Dans *2019 IEEE Globecom Workshops (GC Wkshps)*, 1–6. IEEE.

Scherb, C., Grewe, D., Wagner, M. et Tschudin, C. (2018). Resolution strategies for networking the IoT at the edge via named functions. Dans *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 1–6.

Scherb, C., Marxer, C., Schnurrenberger, U. et Tschudin, C. (2017). In-network live stream processing with named functions. Dans *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, 1–6.

Scherb, C., Marxer, C. et Tschudin, C. (2019b). Execution Plans for Serverless

Computing in Information Centric Networking. Dans *Proceedings of the 1st ACM CoNEXT Workshop on Emerging in-Network Computing Paradigms*, 34–40.

Scherb, C. et Tschudin, C. (2018). Smart Execution Strategy Selection for Multi Tier Execution in Named Function Networking. Dans *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, 1–6.

Shi, W., Cao, J., Zhang, Q., Li, Y. et Xu, L. (2016). Edge computing : Vision and challenges. *IEEE internet of things journal*, *3*(5), 637–646.

Sifalakis, M., Kohler, B., Scherb, C. et Tschudin, C. (2014). An information centric network for computing the distribution of computations. Dans *Proceedings of the 1st ACM Conference on Information-Centric Networking*, 137–146.

Statista (2021a). Département de recherche Statista. Récupéré le 20-05-2021 de `https://www.statista.com/statistics/1181413/medical-app-downloads-growth-during-covid-pandemic-by-country/)`

Statista (2021b). Département de recherche Statista. Récupéré le 03-05-2021 de `https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/`

Thakar, A. T. et Pandya, S. (2017). Survey of IoT enables healthcare devices. Dans *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, 1087–1090. IEEE.

Tokusashi, Y., Dang, H. T., Pedone, F., Soulé, R. et Zilberman, N. (2019). The case for in-network computing on demand. Dans *Proceedings of the Fourteenth EuroSys Conference 2019*, 1–16.

Tschudin, C. et Sifalakis, M. (2014). Named functions and cached computations. Dans *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, 851–857.

Vahlenkamp, M., Schneider, F., Kutscher, D. et Seedorf, J. (2013). Enabling information centric networking in IP networks using SDN. Dans *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, 1–6. IEEE.

van Adrichem, N. L. M. et Kuipers, F. A. (2015). NDNFlow : Software-defined Named Data Networking. Dans *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, 1–5. `http://dx.doi.org/10.1109/NETSOFT.2015.7116131`

Varki, E., Merchant, A., Chen, H. *et al.* (2008). The M/M/1 fork-join queue with variable sub-tasks. Récupéré de `http://citeseerx.ist.psu.edu/viewdoc/`

download?doi=10.1.1.100.3062&rep=rep1&type=pdf

Veltri, L., Morabito, G., Salsano, S., Blefari-Melazzi, N. et Detti, A. (2012). Supporting information-centric functionality in software defined networks. Dans *2012 IEEE International Conference on Communications (ICC)*, 6645–6650. IEEE.

Vergados, D. J., Vergados, D. D. et Maglogiannis, I. (2006). Applying wireless diffserv for qos provisioning in mobile emergency telemedicine. Dans *IEEE Globecom 2006*, 1–5. IEEE.

Vilela, P. H., Rodrigues, J. J., Righi, R. d. R., Kozlov, S. et Rodrigues, V. F. (2020). Looking at Fog Computing for E-Health through the Lens of Deployment Challenges and Applications. *Sensors*, *20*(9), 2553.

Xia, W., Wen, Y., Foh, C. H., Niyato, D. et Xie, H. (2014). A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*, *17*(1), 27–51.

Xu, B., Da Xu, L., Cai, H., Xie, C., Hu, J. et Bu, F. (2014). Ubiquitous data accessing method in IoT-based information system for emergency medical services. *IEEE Transactions on Industrial informatics*, *10*(2), 1578–1586.

Xu, K., Wan, Y. et Xue, G. (2019). Powering Smart Homes with Information-Centric Networking. *IEEE Communications Magazine*, *57*(6), 40–46. http://dx.doi.org/10.1109/MCOM.2019.1800732

Xylomenos, G., Ververidis, C. N., Siris, V. A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K. V. et Polyzos, G. C. (2014). A Survey of Information-Centric Networking Research. *IEEE Communications Surveys Tutorials*, *16*(2), 1024–1049.

Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J. et Yang, X. (2017). A survey on the edge computing for the Internet of Things. *IEEE access*, *6*, 6900–6919.

Zhang, G., Li, Y. et Lin, T. (2013). Caching in information centric networking : A survey. *Computer networks*, *57*(16), 3128–3141.

Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., claffy, k., Crowley, P., Papadopoulos, C., Wang, L. et Zhang, B. (2014). Named Data Networking. *SIGCOMM Comput. Commun. Rev.*, *44*(3), 66–73. http://dx.doi.org/10.1145/2656877.2656887. Récupéré de https://doi.org/10.1145/2656877.2656887

Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thornton, J. D., Smetters, D. K., Zhang, B., Tsudik, G., Massey, D., Papadopoulos, C. *et al.* (2010). Named

data networking (NDN) project. *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, *157*, 158.

Zhang, M., Luo, H. et Zhang, H. (2015). A survey of caching mechanisms in information-centric networking. *IEEE Communications Surveys & Tutorials*, *17*(3), 1473–1499.

Zhao, W., Wang, C. et Nakahira, Y. (2011). Medical application on internet of things. Dans *IET international conference on communication technology and application (ICCTA 2011)*, 660–665. IET.