TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

IoT	Internet of Things
IdO	Internet des objets
OFAP	Optimized Flow Assignment Problem
ACK	Acknowledgment
INT	Interface
G-OFAP	Greedy-Optimized flow assignment problem.
D-OFAP	Dynamic-Optimized flow assignment problem.
SIA	Service-to-Interface assignment.
M2M	Machine-to-Machine.
ONLNE	Optimisation non linéaire en nombre entiers.

LISTE DES SYMBOLES ET UNITÉS DE MESURE

J	Joule.
Ko	Kilo octet.
Mo	Méga octet.

CHAPITRE 1

INTRODUCTION

1.1 Contexte et motivation

L'industrie de l'Internet des objets (IdO) est en plein essor et les entreprises, y compris les fournisseurs de services perçoivent les occasions d'affaires futures et sont en concurrence pour fournir les meilleures solutions IdO sur le marché. Accenture prévoit que la valeur ajoutée de l'IdO d'ici 2030 atteindra 14,2 trillions de dollars (Accenture, 2015). En effet, selon une dernière étude du marché, on estime que le nombre d'objets connectés atteindra 75,4 billions d'équipements d'ici 2025 (Ray, 2018). Comme ces rapports récents ont estimé que la portée du marché de l'IdO continuera de croître au cours des deux ou trois prochaines décennies, de nos jours, de nombreux chercheurs multidisciplinaires s'impliquent dans ce domaine. Le domaine de l'IdO a été aussi largement discuté dans plusieurs études récentes, car il s'agit d'une technologie permettant d'intégrer de manière transparente chaque objet au moyen d'Internet et fournit beaucoup d'imaginations différentes pour la vie future. Ces objets connectés sont omniprésents dans les maisons, les villes, les voitures, les usines et ils sont même intégrés dans les corps humains (Islam *et al.*, 2015).

Pour permettre aux objets connectés de réaliser ces différents scénarios, ils sont équipés par plusieurs interfaces réseau. Un environnement hétérogène entoure un objet connecté où un ensemble de réseaux mobiles et sans fils (Wi-Fi, Bluetooth, LoRa, RFiD, Zigbee, etc.) coexistent les uns avec les autres pour répondre aux exigences croissantes des applications IdO comme décrit à la figure 1.1. De plus, ces technologies utilisent des diverses normes et peuvent être classées en fonction de plusieurs paramètres distincts comme : la consommation d'énergie, débit binaire, la longueur du paquet, la portée et la topologie (Stočes *et al.*, 2016). Bien que les ressources réseaux disponibles soient souvent limitées, les objets connectés fournissent une grande variété de services. Certainement, une assignation plus efficace des services IdO aux ressources réseau disponibles améliorerait les performances du système IdO. Cependant, l'as-

signation optimale des flux IdO aux interfaces réseau de la passerelle n'est pas simple compte tenu de la nature hétérogène d'un environnement IdO où la sélection d'une interface réseau dépend de plusieurs paramètres.

Cette hétérogénéité de l'IdO a créé de nouveaux défis dans l'industrie et donc de nouveaux axes de recherche. Chaque technologie est dotée de ses propres forces et faiblesses. En conséquence, chaque technologie réseau est utilisée dans un contexte différent et un domaine d'application bien spécifié en fonction des exigences des applications IdO. Par exemple, les applications de l'agriculture intelligente qui intègrent des objets connectés pour améliorer et assurer la qualité de la production sont des applications qui nécessitent des technologies réseau qui garantissent la faible consommation d'énergie. De plus, ce type d'application peut intégrer des services IdO qui utilisent le même réseau et qui ont des différentes exigences comme un débit binaire élevé (Partha, 2017). Dans ce cas, la passerelle et les objets connectés doivent coopérer et mettre leurs ressources à disposition afin de satisfaire les exigences de ces applications affectées au réseau.

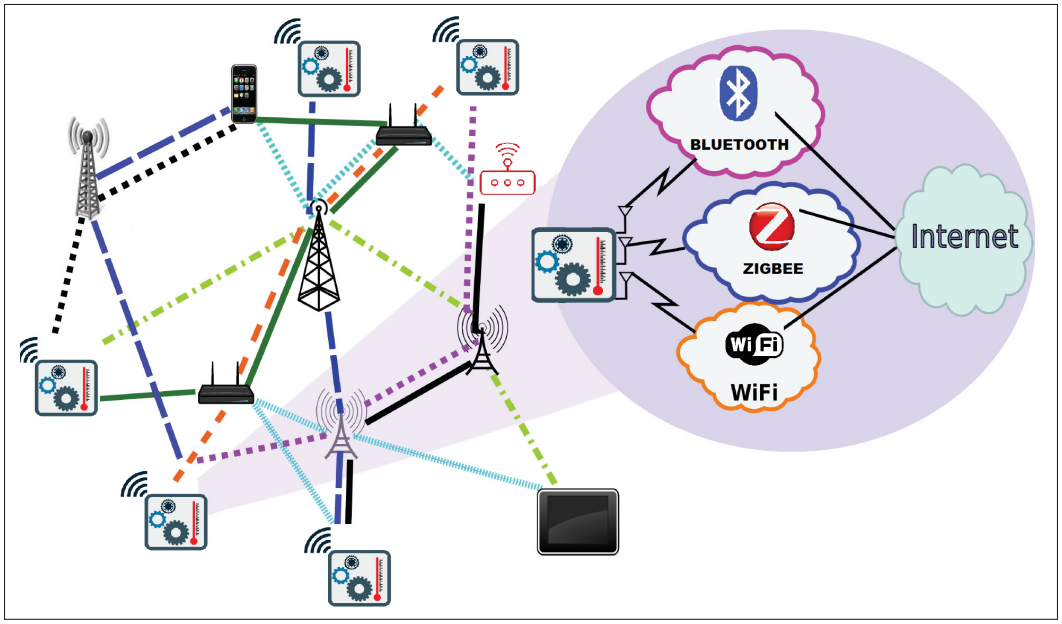

Figure 1.1 Réseaux hétérogènes
(Tosh & Sengupta, 2015)

Certes, cette diversité a contribué à faire de l'IdO un écosystème riche. Néanmoins, cette richesse a créé de nouvelles classes d'application. Leurs exigences sont variables en termes de débit, de latence et de gigue. Par suite, ces propriétés de l'application accroissent la complexité de l'assignation des services aux interfaces réseau afin de garantir une qualité de service satisfaisante. De plus, les différentes façons de connecter les objets complexifient le développement d'applications à grande échelle d'autant plus que les développeurs dans le domaine de l'IdO ne possèdent pas des outils qui permettent de sélectionner dynamiquement le réseau en fonction de la charge du trafic en temps réel. Toutefois, avec l'absence d'un mécanisme qui permet la sélection dynamique des interfaces, les services seraient affectés statiquement aux ressources réseau. Par suite, cette richesse de l'environnement de l'IdO ne peut pas être exploité qu'à travers des systèmes automatisés et dynamiques. Dans un tel environnement hétérogène, un objet connecté à un réseau sans fils doit être toujours bien servi.

1.2 Problématique

De nos jours, les appareils IdO sont souvent équipés de plusieurs interfaces. Ils échangent les données à travers différents protocoles de communication tels que Zigbee, WiFi, Bluetooth, LoRa et 3G avec une passerelle multi-interfaces. Cette hétérogénéité offre une richesse et nous permet de garantir une meilleure performance du réseau. De plus, l'accès simultané à une multitude d'interfaces améliorera considérablement l'utilisation du réseau, la latence globale, et fournira une solution robuste aux équipements IdO.

Cependant, en tenant compte d'une passerelle et d'équipements dotés de plusieurs interfaces réseau, les équipements IdO ne peuvent pas transmettre ses données à une interface appropriée sans un mécanisme efficace qui affecte les flux aux interfaces disponibles d'une manière optimale et qui permet la satisfaction des exigences de l'application. L'absence d'un mécanisme d'assignation de flux peut entraîner une congestion, en particulier lorsque le nombre d'appareils qui transmettent des données en utilisant la même interface augmente. Prenons l'exemple d'une application de suivi des plantes qui consiste à surveiller des espaces verts à grande échelle en collectant des informations telles que l'humidité du sol, le vent, le pH et le CO_2 (Stočes *et al.*,

2016). En outre, cette application envoie les images des feuilles au serveur *cloud* à l'aide d'une caméra haute définition, ce qui permet aux experts de détecter les anomalies qui peuvent toucher aux plantes. Dans ce scénario, l'appareil IdO est équipé de deux interfaces réseaux, WiFi et Zigbee. Les flux sont attribués d'une manière aléatoire et statique aux interfaces réseau. Pour cet exemple, les mesures d'humidité du sol, de température et de CO_2 sont attribuées de manière statique à l'interface Zigbee. Le service d'envoi des images des feuilles est attribué à l'interface WiFi car il nécessite beaucoup plus de bande passante que les autres services. Toutefois, le trafic réseau peut varier et de nouveaux services peuvent être attribués à l'interface Zigbee ou WiFi de la passerelle. Et donc, la capacité d'une interface telle que Zigbee peut être saturée au cours du temps. Par conséquent, une attribution statique des flux peut provoquer une congestion du réseau et des valeurs de latences importantes. De plus, lorsque le trafic est affecté d'une manière statique aux interfaces, cela peut entraîner une saturation de certaines interfaces. C'est pour cela, comme la coexistence de différentes technologies d'accès sans fils est davantage une réalité qu'une hypothèse, les équipements IdO dotés de multiples interfaces réseau devront relever le défi consistant à exploiter simultanément plusieurs technologies sans fil disponibles, en s'adaptant à l'aspect dynamique des environnements IdO où le nombre d'objets et le trafic changent dynamiquement au cours du temps. Pour remédier à ce problème, un module de contrôle d'interface devra être implémenté au niveau de la passerelle. Ce module a pour objectif de gérer simultanément et d'une façon dynamique les interfaces et d'affecter les flux des appareils IdO aux interfaces appropriées qui permettent de répondre aux exigences des applications.

Dans notre recherche, nous introduisons un modèle mathématique qui modélise les ressources disponibles dans les équipements IdO multi-interfaces et modélise les flux échangés avec la passerelle. Nous définissons le problème d'assignation des flux aux interfaces (*Optimized Flow Assignment Problem*) dénoté dans le reste du rapport par OFAP. L'objectif d'OFAP est de maximiser la quantité de données valides acceptées par la passerelle, en tenant compte des ressources réseau disponibles et en satisfaisant la qualité de service des flux (Amor *et al.*, 2019).

1.3 Questions de recherche

Dans notre travail, nous abordons les questions de recherches suivantes :

- QR1 : Le domaine de l'Internet des objets s'appuie sur des réseaux hétérogènes où les passerelles et les appareils IdO sont équipés par une multitude d'interfaces réseau. Ces équipements fournissent des services respectant des différentes exigences comme la bande passante et le temps de réponse.

  Dans un tel environnement hétérogène, comment peut-on échanger les données d'une manière optimale en maximisant la quantité des données acceptée tout en satisfaisant les capacités des ressources réseau de la passerelle et les exigences des applications IdO ?

- QR2 : Comme expliqué dans l'introduction, le nombre d'applications IdO ne cesse à augmenter. De plus, de nouveaux appareils équipés de plusieurs interfaces sont mise en place pour servir à ces applications qui ont des spécifications différentes. En conséquence, cette diversité en termes de ressources et applications a rendu de l'environnement IdO un environnement riche. Ainsi, nous posons la question suivante :

  Comment peut-on garantir une meilleure exploitation des interfaces réseau disponibles afin de servir les différentes applications existantes ?

- QR3 : Pour assurer une meilleure performance des réseaux IdO où les appareils et la passerelle IdO sont équipés par une multitude d'interfaces réseau, un objet connecté aura besoin de migrer entre ces différentes interfaces. Et donc, cela nous mène à notre troisième question :

  Comment peut-on migrer efficacement les flux envoyés par les équipements IdO d'une interface à une autre ?

1.4 Objectifs

L'objectif principal de la présente recherche consiste à maximiser les flux acceptés par une passerelle IdO équipée de plusieurs interfaces réseau tout en satisfaisant les échéances des applications et la capacité des ressources réseau.

Cet objectif est traduit en trois sous objectifs :

O1 : La conception d'un modèle d'optimisation qui permet de représenter notre système IdO composé d'une passerelle et d'appareils équipés de différentes interfaces réseau et qui permet de maximiser la quantité de données acceptées par la passerelle en satisfaisant les échéances des flux et la capacité du réseau.

O2 : Le développement d'une solution pour contrôler les flux et les attribuer d'une manière efficace aux interfaces réseau d'une passerelle IdO.

O3 : La conception et l'implémentation d'un protocole de *handover* qui permet le transfert d'une interface réseau à une autre.

1.5 Principales contributions

Dans le cadre de ce mémoire, trois principales contributions peuvent être considérées :

- Nous avons introduit un modèle mathématique qui caractérise les ressources réseau disponibles dans les équipements IdO multi-interfaces et modélise les flux échangés avec la passerelle. Nous définissons le problème d'optimisation du flux (OFAP) qui permet de maximiser la quantité de données acceptées par la passerelle, en tenant compte de la capacité des ressources réseau disponibles et en satisfaisant les exigences des flux IdO.
- Nous proposons deux algorithmes pour résoudre le problème d'assignation optimale des flux (OFAP) et nous montrons leur efficacité en calculant le taux d'acceptation des flux et en variant différents paramètres.

- Nous avons conçu et implémenté le module de contrôle de la passerelle et un protocole de *handover* qui permet le transfert des flux entre les différentes interfaces réseau disponibles en assurant l'établissement de connexion entre l'équipement IdO et la passerelle tout au long le processus de *handover*.

1.6 Plan du mémoire

Cette thèse sera divisée en trois principaux chapitres :

- Dans le premier chapitre, nous présenterons la revue de littérature et la mise en contexte en introduisant les différents travaux de recherche qui traitent les problèmes d'assignation et d'ordonnancement des flux :
  - Tout d'abord, nous introduisons les technologies nécessaires pour la bonne compréhension de notre projet de recherche.
  - Nous décrivons les technologies des réseaux sans fils les plus utilisées dans le domaine de l'Internet des objets et nous discutons la différence entre ces réseaux.
  - nous présentons des travaux connexes qui ont adressé le même problème que le nôtre, et nous discutons leurs limites.
- Le troisième chapitre présentera notre méthodologie pour atteindre nos objectifs énumérés précédemment dans la section 1.4 :
  - Nous présentons notre solution à implémenter au niveau d'une passerelle à interfaces multiples, qui est le module de contrôle de la passerelle et qui permet la gestion optimale des flux reçus et des ressources réseau disponibles.
  - Nous définissons notre modèle d'optimisation qui permet d'attribuer efficacement les flux aux interfaces réseaux.
  - Nous introduisons aussi dans ce chapitre deux algorithmes D-OFAP et G-OFAP qui permettent de résoudre le problème d'OFAP.
- Dans le quatrième chapitre, nous nous concentrerons sur les simulations effectuées pour valider notre solution. Nous présentons également les détails de l'implémentation de notre système.

CHAPITRE 2

REVUE DE LITTÉRATURE

2.1 Introduction

L'augmentation du nombre d'appareils IdO dans les environnements urbains conduira certainement à de plus grands échanges de flux entre les objets connectés, les serveurs de données, les commutateurs et les passerelles. Dans la littérature, de nombreuses approches ont été proposées pour réduire la charge du réseau, garantir des délais acceptables, un débit élevé et une faible consommation d'énergie en exploitant les interfaces réseau disponibles. Les décisions de sélection des interfaces reposent à la fois sur les caractéristiques du réseau et les exigences de l'application, ce qui entraîne un problème d'assignation de flux influencé par plusieurs critères.

Dans ce chapitre, nous présentons quelques solutions proposées dans la littérature pour optimiser l'assignation de flux aux interfaces et qui utilisent des différents techniques pour la résolution de leurs problèmes. Ensuite, nous exposons des travaux connexes qui utilisent des approches de *handover* vertical basées sur les préférences de l'utilisateur et les métriques du réseau. Enfin, nous présentons une étude comparative qui met en évidence les avantages et les limites de chaque solution existante.

2.2 L'internet des objets : Mise en contexte

L'acronyme Machine-to-Machine (M2M) représente la communication directe entre deux machines dotées de mêmes fonctions en utilisant des voies de communication filaire ou sans fils. Les technologies M2M offrent une manière efficace pour mettre en place des communications automatisées entre les machines sans l'interaction humaine (Theoleyre & Pang, 2013). Grâce à ces technologies, le domaine de l'Internet des objets (IdO) a évolué. Maintenant, un grand nombre d'appareils IdO se coopèrent afin de fournir des services automatisés. Dans cette section, nous présentons les architectures IdO et les types de technologies utilisées dans ce domaine.

2.2.1 Protocoles de communication d'IdO

Le domaine de l'Internet des objets fait face maintenant à une hétérogénéité sur différentes couches. D'une part, les technologies de l'IdO sont utilisées dans plusieurs domaines d'applications comme la logistique, la sécurité publique, les villes et les maisons intelligentes, etc. D'autre part, les architectures IdO utilisent des différentes technologies de réseaux pour envoyer les données de capteurs vers la passerelle. Dans cette section, nous présentons et nous montrons l'hétérogénéité des réseaux sans fils les plus utilisés dans le domaine de l'IdO.

Les réseaux dans le domaine de l'IdO sont un ensemble de capteurs autonomes distribués qui coopèrent ensemble au moyen de protocoles de communication. Le progrès phénoménal de l'IdO a donné naissance à d'autres protocoles de communications comme LoRa et Sigfox. Aujourd'hui, plusieurs réseaux et technologies sont utilisés pour communiquer les données des objets connectés aux centres de données à l'infonuagique. Dans les sections suivantes, nous donnerons un aperçu sur quelques protocoles de communications.

- **WIFI**

  WiFi est basé sur le standard 802.11 introduit par IEEE en 1997 (Morrow, 2004). L'objectif de ce standard est de décrire un réseau sans fil (WLAN) qui permet de fournir des services équivalents à un réseau filaire avec un débit élevé. L'architecture de ce standard est flexible et peut supporter des réseaux à grandes et moyennes échelles. Avec des débits de données bruts allant jusqu'à 11Mb/s-1Gb/s et un débit réel au niveau de la couche d'application d'environ 5Mb/s. WIFI fournit des débits suffisamment rapides pour des applications telles que FTP, HTTP, Streaming vidéo, etc.

- **Bluetooth**

  Bluetooth est basé sur le standard 802.15.1 de IEEE. Ce réseau supporte diverses applications à courte portée (environ 10 mètres) où le débit binaire est de 0.5 à 2 Mb/s (Morrow, 2004). Bluetooth fonctionne principalement en mode *ad-hoc*. Le réseau est appelé *piconet*. L'appareil qui initie la communication s'appelle "*master*" et les autres éléments du réseaux sont les "*slaves*". Chaque appareil Bluetooth peut fonctionner en tant que *slave* ou *master*.

Quand un seul *slave* est utilisé, la communication est appelé *point-to-point*. Le *master* peut contrôler au maximum sept appareils *slave* dans une architecture *point-to-multipoint*.

- **LoRa**

  LoRa, qui signifie «Long Range», est un système de communication sans fil à longue portée, développé par LoRa Alliance. Cette technologie fonctionne sur les fréquences 868 ou 915 Mhz. LoRa est utilisé dans les appareils alimentés par batterie où la consommation d'énergie est d'une importance primordiale. Ce réseau utilise pour la couche MAC le protocole LoRaWAN. C'est un protocole qui fournit un mécanisme de contrôle d'accés permettant à nombreux périphériques de communiquer avec la passerelle en utilisant la modulation LoRa (Augustin *et al.*, 2016).

- **Zigbee**

  Le standard IEEE 802.15.4 LR-WPAN (Zigbeee) supporte les communications sans fils à faible débit entre les appareils fixes et mobiles nécessitant une consommation de batterie faible. Ce réseau a une portée de 10 à 50 mètres (Morrow, 2004). Les domaines d'applications de ce standard sont : les maisons et villes intelligentes, détecteurs de fumée et de carbon, éclairage à distance, etc.

2.2.2 Comparaison des réseaux

Dans cette section, nous présentons un tableau qui résume les caractéristiques et la différence entre les réseaux les plus utilisés dans le domaine de l'IdO (voir tableau 2.1).

2.2.3 Passerelle IdO

La passerelle IdO est une composante importante du réseau. Sa fonctionnalité principale consiste à transmettre des flux à l'internet pour être traités aux centres de données hébergés à l'infonuagique. Les appareils IdO ne sont pas toujours capables de se connecter directement à l'Internet à cause de l'absence d'une normalisation des protocoles de communication IdO (Kim *et al.*, 2015) et doivent toujours passer par une passerelle. Il y en a aussi d'autres facteurs, tels que la capacité limitée de traitement des appareils IdO, les ressources de puissances limitées, etc. Par

Tableau 2.1 Tableau comparatif des réseaux
(Ray & Pratim, 2017)

Paramètres	WiFi	Bluetooth	LR-WPAN	LoRa
Standard	IEEE 802.11 a/c/b/d/g/n	IEEE 802.15.1	IEEE 802.15.4	LoRaWAN
Bande de fréquence	5Ghz-60Ghz	2.4Ghz	868/915 Mhz/2.4 Ghz	868/900 Mhz
débit binaire	1Mb/s-1Gb/s	1-24Mb/s	40-250Kb/s	0.3-50 Kb/s
Couverture	20-100m	8-10m	10-20m	<30Km
Consommation d'énergie	Élevé	Bluetooth Medium : Bas	Bas	Bas
Coût	Élevé	Bas	Bas	Élevé

conséquent, la passerelle IdO intervient pour être l'intermédiaire entre l'Internet et les appareils IdO.

Un scénario typique est une maison connectée comportant de nombreux appareils IdO comme décrit à la figure 2.1. Pour un scénario pareil, nous avons besoin d'une passerelle qui est capable de gérer les appareils dans son environnement.

Les procédures principales à exécuter par la passerelle IdO sont :

- Renforcer la gestion des appareils IdO et des interfaces disponibles.
- Inter-connecter les centres de donnée aux réseaux IdO.

L'objectif principal de la passerelle IdO est similaire à celui d'une passerelle conventionnelle, mais certaines fonctionnalités supplémentaires sont présentées dans la section suivante.

2.2.3.1 Les fonctionnalités d'une passerelle IdO

Dans cette section, nous présentons les fonctionnalités de base d'une passerelle IdO. Premièrement, elle doit disposer d'une multitude d'interfaces réseau à différentes caractéristiques. Dans les architectures IdO, les appareils envoient leurs données en utilisant des différents protocoles de communication. Par exemple, une technologie de communication à courte portée est nécessaire pour fournir la connexion à des appareils équipés d'émetteurs à courte portée. Donc, la passerelle doit prendre en charge plusieurs protocoles pour pouvoir échanger des données

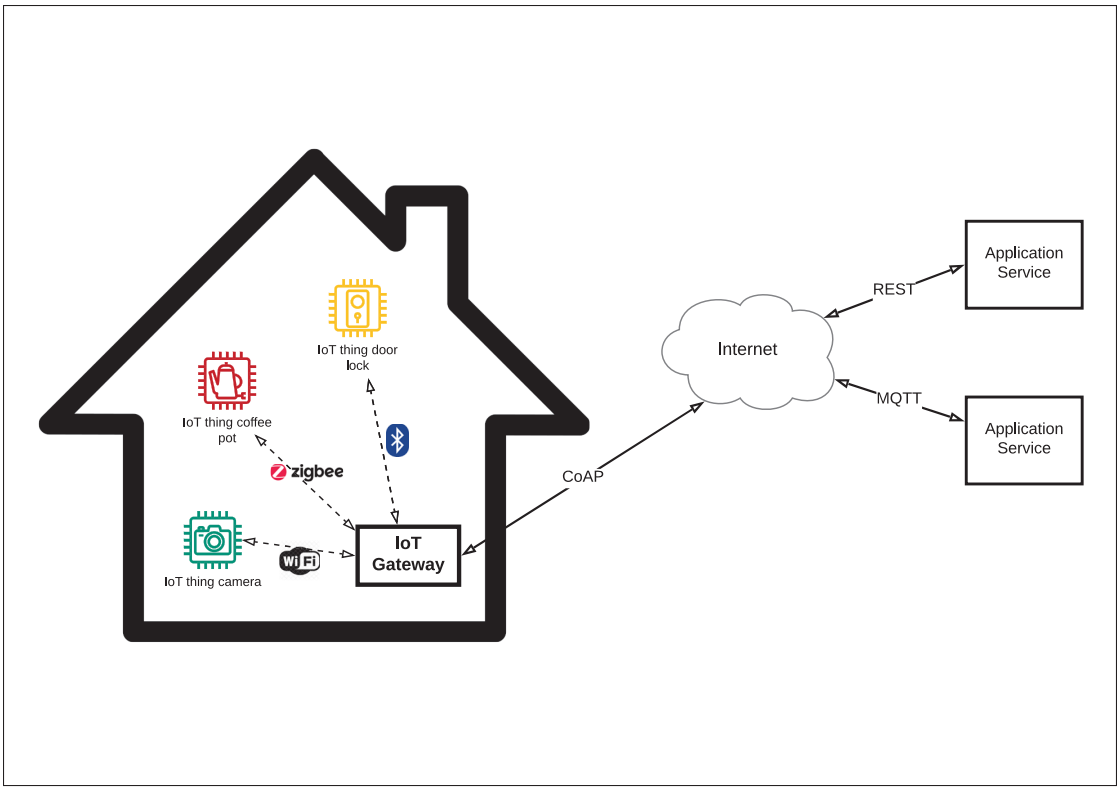

Figure 2.1 Exemple d'une passerelle IdO d'une maison connectée (Desai *et al.*, 2015)

avec les appareils IdO. En outre, la passerelle doit être équipée de communications filaires, de protocoles de communication WLAN et cellulaires, notamment Wi-Fi et LTE ainsi que de protocoles de communication à courte portée existants, tels que ZigBee, Bluetooth et Z-wave (Abbas & Yoon, 2015).

Deuxièmement, la gestion du réseau est une fonction primordiale pour la passerelle IdO. Cette dernière ne se limite pas aux fonctions de gestion du réseau conventionnel. Elle devrait, par exemple, affecter les services IdO aux interfaces dynamiquement, mettre à jour les modules logiciels associés à l'appareil, etc. En conséquence, le concept de gestion du réseau dans la passerelle IdO s'est étendu et les techniques permettant de renforcer ce concept sont essentielles pour améliorer la performance du réseau et la qualité de service.

Enfin, la passerelle IdO devrait assurer une gestion efficace des ressources et interfaces hétérogènes que ce soit pour l'équipement IdO ou pour la passerelle. Donc, des mécanismes robustes et des processus de *handover* efficaces doivent être implémentés pour gérer les différentes interfaces.

2.2.3.2 Architecture d'une passerelle IdO

Dans cette section, nous présentons des détails sur la structure générale de la passerelle IdO, comme illustrée à la figure 2.2, où les composantes nécessaires à la transmission et à la réception de données sont présentées. La figure 2.2 présente une structure générique d'une passerelle IdO dotée de plusieurs interfaces et montre également qu'il y a deux types d'interfaces dans un tel appareil :

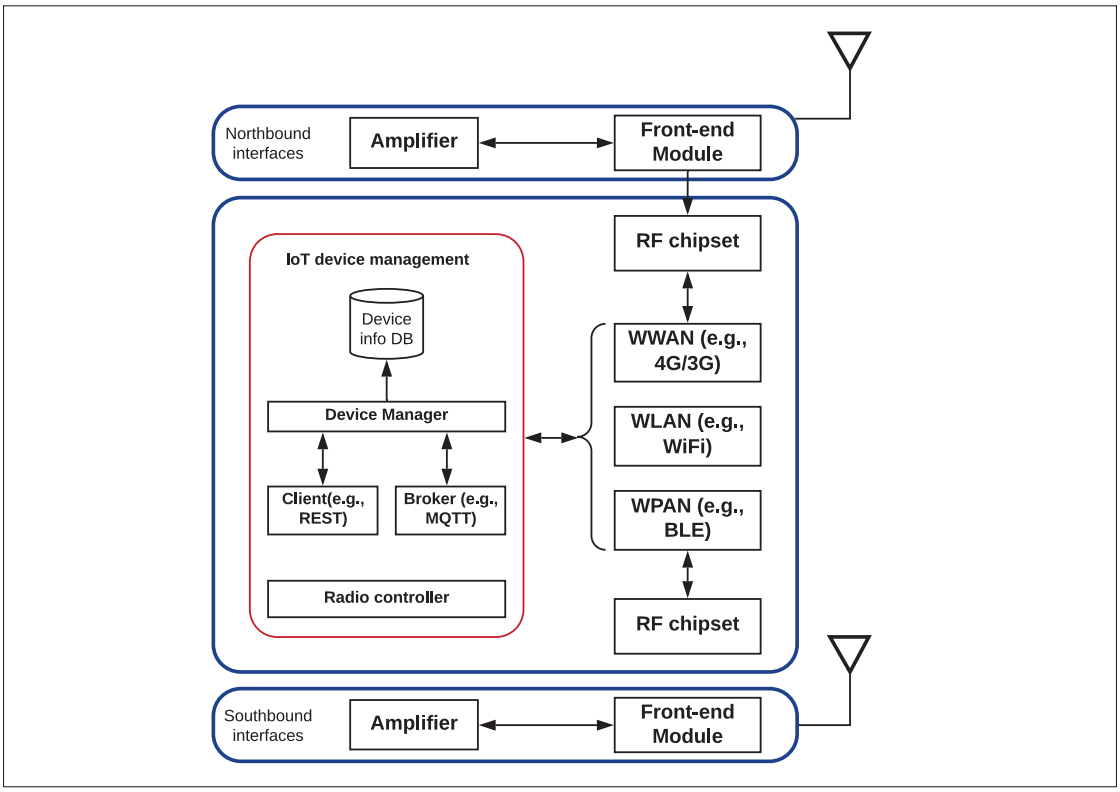

Figure 2.2 Structure générale d'une passerelle IdO
(Abbas & Yoon, 2015)

- *Northbound interfaces* : Ce sont les interfaces qui communiquent les données reçues en utilisant les réseaux cellulaires 3G/4G ou filaires. Les protocoles de transport utilisés pour transmettre les données des capteurs peuvent être MQTT ou des requêtes REST pour communiquer avec l'API du serveur de données (Datta *et al.*, 2014).
- *Southbound interfaces* : Ce sont les interfaces qui reçoivent les données de capteurs. Chaque appareil IdO envoie ses données à travers une technologie réseau différente. Pour cette raison, la passerelle doit être dotée de plusieurs interfaces (WPAN et WLAN) pour couvrir le plus grand nombre d'appareils (Datta *et al.*, 2014).

La figure 2.2 montre aussi qu'une passerelle IdO est composée de différents modules de communication (*RF Chipset*, *Front-end module*, *Radio controller*, antennes) qui sont mise en place pour gérer la communication de données au niveau de la couche physique par diverses technologies d'accès (Abbas & Yoon, 2015). De plus, une passerelle intègre un module pour la gestion des appareils IdO et leurs données reçues. Ce module utilise un broker pour la réception des données et l'établissement des connexions des appareils IdO. Le module de gestion des appareils inclut également un service appelé *Device Manager* qui permet l'écoute des messages reçus au niveau du broker et les stocker dans la base de données *device info DB*. Cette dernière contient toutes les informations sur les appareils IdO et leurs données (ID, adresse MAC/IP, emplacement, nombre et le type des capteurs connectés, etc.). Ce module permet aussi la transmission des données au nuage informatique par le biais de clients appropriés à l'architecture (Client MQTT, requêtes REST, CoAP, etc.)

2.2.3.3 Exemples de passerelle IdO

Plusieurs recherches ont été menées pour concevoir des passerelles qui pourront répondre aux exigences mentionnées dans la section précédente. Pour résoudre les problèmes de l'exploitation efficace des différents réseaux disponibles dans l'environnement IdO, la recherche de Chang et al (Chang *et al.*, 2015) a pour objectif de concevoir et d'implémenter une passerelle qui supporte des différents protocoles de communication sans fil. La passerelle conçue peut améliorer l'efficacité de la transmission en exploitant tous les réseaux disponibles. Elle est im-

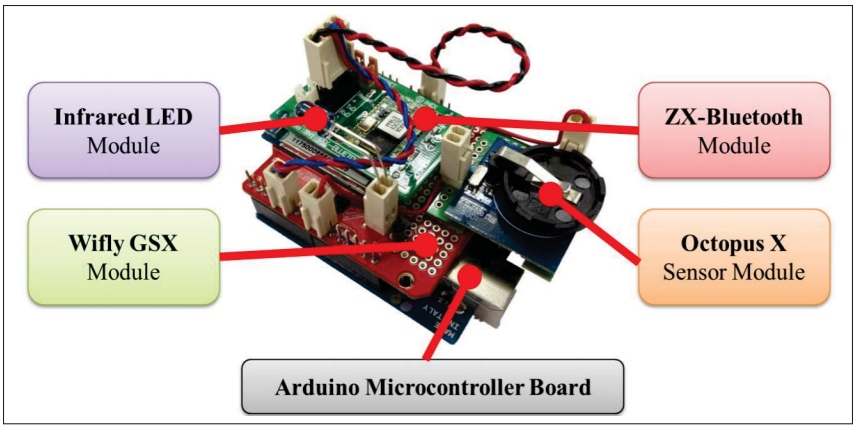

Figure 2.3 Les composants matériels d'une passerelle IdO
(Chang *et al.*, 2015)

plémentée par l'intégration de modules réseau (comme *ZX-Bluetooth & WiFly GSX*) qui sont installés sur une carte Arduino. Les composants matériels qui ont permis le développement de la passerelle IdO sont illustrés à la figure 2.3. Ces différents modules sont connectés à la carte Arduino au moyen de communications séries (UART et SPI) comme décrit à la figure 2.4.

Un autre travail, présenté à l'article (Min *et al.*, 2014), propose une passerelle IdO à réseaux hétérogènes. Cette passerelle a trois caractéristiques principales. Premièrement, elle transmet les données entre Internet et les réseaux de capteurs en utilisant plusieurs types de protocoles de communication comme Bluetooth, Zigbee et Ethernet. Deuxièmement, pour assurer la sécurité et la fiabilité des données de la passerelle IdO, certains protocoles des couches supérieures sont implémentés et mis en œuvre comme lwIP (qui est un protocole basé sur TCP/IP). Troisièmement, un algorithme d'ordonnancement basé sur l'assignation dynamique des priorités est également conçu et utilisé dans la passerelle pour résoudre le problème de la concurrence des données arrivant au même instant et aussi pour améliorer les performances en temps réel en planifiant efficacement les tâches à exécuter.

2.2.4 Exemples d'appareils IdO

La demande croissante de nouvelles applications mène certainement à penser à exploiter simultanément tous les réseaux d'accès disponibles pour fournir un débit élevé. Donc, avec l'exis-

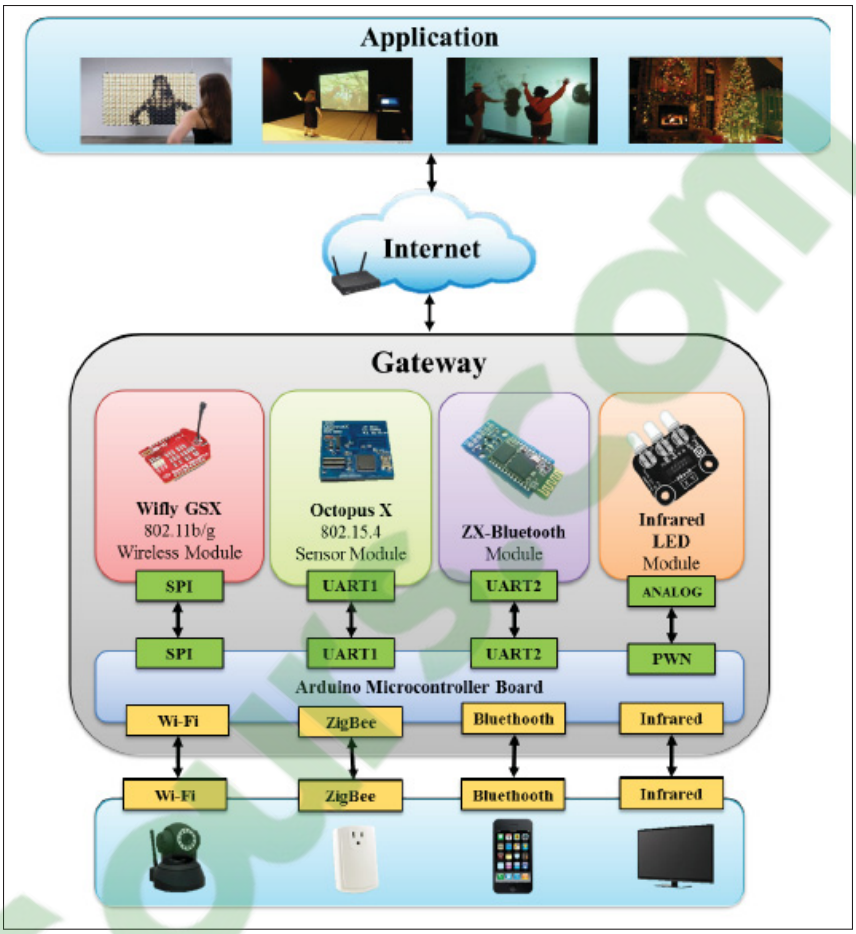

Figure 2.4 Conception d'une Passerelle à interfaces multiples (Chang *et al.*, 2015)

tence d'une multitude de technologies d'accès sans fil (WiFi, UMTS, 3G, HSPA, LTE, WiMAX, etc.), il est devenu courant qu'un appareil IdO ait plusieurs interfaces à la fois. Comme la coexistence d'appareils IdO dotés de multiples interfaces est davantage une réalité qu'une hypothèse, nous présentons dans cette section des exemples d'appareils IdO à interface multiple.

Actuellement, les appareils IdO sont souvent dotés de plusieurs interfaces réseau avec différents protocoles de communication, allant de courtes portées (Zigbee, Bluetooth) aux longues portées (WiFi, WiMAX, LTE). De plus, une zone géographique donnée peut être couverte par plusieurs technologies d'accès présentant des spécifications différentes. Par exemple, les ré-

seaux cellulaires peuvent fournir une couverture géographique à côté de réseaux WLAN (*Wireless Local Area Network*) contrairement au passé, lorsqu'une zone géographique est couverte par une seule technologie ou deux au plus. La disponibilité de plusieurs interfaces dans un seul appareil IdO donne la possibilité de choisir parmi de multiples réseaux disponibles, ce qui peut fournir une meilleure qualité de service des applications IdO.

Dans ce contexte, plusieurs appareils IdO ont été conçus et commercialisés par l'industrie tels que *WiPy*, *Lopy* et *Gpy* qui sont développés par *Pycom* (Pycom, 2019). Ce sont des cartes IdO programmables en *microPython* et qui sont dotées de plusieurs interfaces. Dans la littérature, plusieurs travaux de recherche ont implémenté des appareils IdO efficaces à bas coût et équipés d'une multitude d'interfaces en utilisant des modules matériels comme *ESP8266*, *Xbee*, *SX1278* pour LoRa, etc. Les auteurs dans (Zainuddin *et al.*, 2019), ont conçu un appareil IdO qui permet de surveiller les indices de la qualité d'eau à l'aide des capteurs. Ce système est conçu pour surveiller les valeurs de température et de pH. Ils ont mis en place un équipement IdO intermédiaire entre les capteurs et la passerelle en se basant sur une carte *Arduino* sur laquelle ils ont installé un module WiFi (*ESP8266*) pour communiquer avec la passerelle et un module *Xbee* pour collecter les données des capteurs. On peut aussi trouver Raspberry Pi qui est doté de trois interfaces qui sont : Bluetooth, WiFi et RJ45. Bien que cet appareil est plutôt considéré comme un *nano-ordinateur*, on pourrait également l'utiliser comme un appareil IdO puisqu'on peut y connecter des capteurs grâce aux ports GPIO.

Le tableau suivant résume la différence entre les appareils IdO mentionnés ci-dessus.

Tableau 2.2 Tableau comparatif des appareils IdO
(Pycom, 2019)

Caractéristiques	**WiPy**	**LoPy**	**Gpy**	**(Zainuddin *et al.*, 2019)**	**Raspberry Pi**
WiFi	Oui	Oui	Oui	Oui	Oui
LoRaWan	Non	Oui	Non	Non	Non
Bluetooth	Oui	Oui	Oui	Non	Oui
Zigbee	Non	Non	Non	Oui	Non
GPIO	24	24	22	20	20
Ram	4MB	512KB	4MB	8KB	1GB

2.2.5 Hétérogénéité des applications IdO

La communication avec une grande variété d'objets connectés, tels que les appareils ménagers, les véhicules, les drones, les caméras de surveillance et les capteurs de températures, permettra le développement de nouvelles applications qui utilisent cette énorme quantité de données créées par ces objets. De nos jours, on trouve les équipements IdO partout et dans tous les domaines comme la domotique, la santé, l'assistance aux personnes âgées, l'automobile, l'automatisation industrielle, la gestion du trafic, etc (Gaur *et al.*, 2015). Cependant, pour un domaine d'application aussi hétérogène, la satisfaction des exigences de tous les scénarios d'application possibles crée un défi pour l'industrie. Prenons l'exemple des villes intelligentes où plusieurs applications coexistent pour fournir des services aux citoyens comme la qualité d'air dans une zone donnée, la gestion du trafic et le suivi de la consommation d'énergie.

Andrea et al ont présenté dans leur recherche une étude de cas sur la ville de Padova en Italie (Zanella *et al.*, 2014) où plusieurs applications ont été déployées. Chacune de ces applications possède des différentes exigences en termes de débit binaire, latence tolérée, etc. comme détaillé dans le tableau 2.3.

Tableau 2.3 Tableau comparatif des applications (Zanella *et al.*, 2014)

Service	Réseau	Débit	Latence tolérée
Suivi de la qualité de l'air	IEEE 802.15.4, Bluetooth and Wifi	1pkt / 10min	5min for data
La gestion des déchets	Wifi, 3G et 4G	1pkt/hour	30min for data
Parc intelligent	802.15.4	à la demande	1 min
Suivi de la consommation d'énergie	PLC et Ethernet	1pkt / 10min	5 min for data, et exigences plus stricte pour le contrôle

2.3 Gestion des flux dans les réseaux hétérogènes

Les réseaux sans fils hétérogènes ont plusieurs caractéristiques différentes comme détaillé à la section 2.2.1. Cette hétérogénéité crée des défis lors de la sélection des interfaces réseau pour envoyer les flux. Cela fait de la gestion des flux dans les réseaux sans fils hétérogènes un problème de décision complexe. Pour obtenir des performances efficaces du réseau IdO, le mécanisme d'assignation des flux nécessite un algorithme de sélection de réseau robuste, capable de gérer des problèmes de prise de décision complexes. Cela représente un défi pour les concepteurs des réseaux sans fils destinés au domaine de l'IdO, car une sélection de réseau non optimale peut entraîner des effets indésirables sur le réseau, tels que : la perte de données critiques, saturation du réseau et une mauvaise qualité de service. Dans cette section, nous présentons des travaux connexes qui ont proposé des mécanismes d'ordonnancement des flux et des approches de sélection de réseaux dans un environnement IdO hétérogène.

2.3.1 Approches de sélection de réseaux

La nature dynamique des applications IdO nécessite la mise en œuvre d'un mécanisme robuste d'assignation de flux dans les réseaux hétérogènes. Dans le reste de cette section, nous examinons les approches de sélection d'interfaces réseau proposées dans la littérature qui permettent la satisfaction des besoins des utilisateurs. La figure 2.5 présente notre classification des travaux existants.

2.3.1.1 Approches basées sur les métriques du réseau

Ces approches sont basées sur des paramètres comme le RSSI, *Bit Error Rate* (BER), la bande passante des interfaces réseau, l'état actuel du réseau, etc. Contrairement aux approches basées sur les métriques de l'appareil/utilisateur, la collecte de ces paramètres peut également introduire une consommation d'énergie supplémentaire au niveau des équipements IdO. Cependant, ces paramètres reçus lors de la collecte d'informations sont importants pour la sélection des réseaux et sa précision. Par conséquent, un compromis entre la précision et le coût doit être

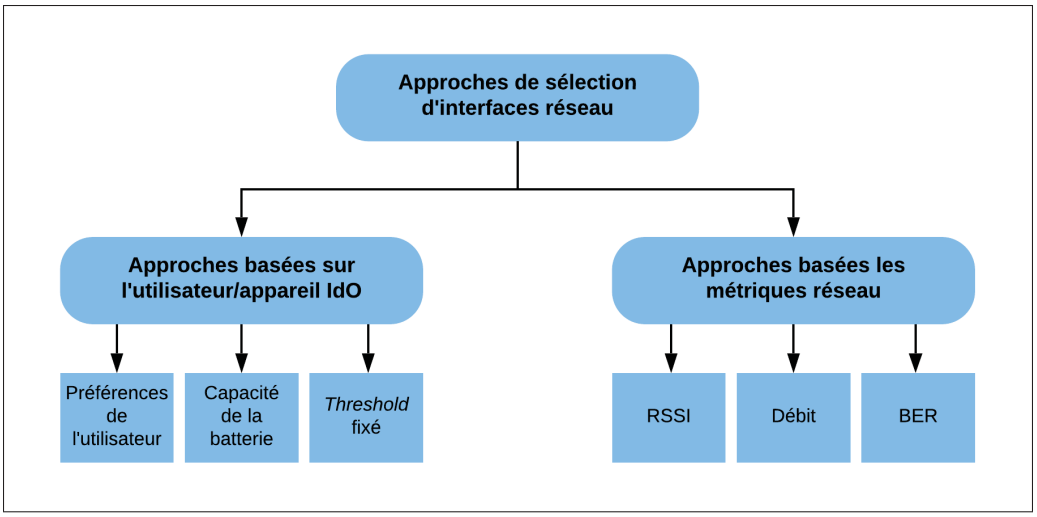

Figure 2.5 Classification des travaux existants

envisagé. Dans la plupart des travaux existants, le processus de *handover* et la sélection des réseaux sont effectués sur la base du *Received Signal Strength Indication* ou RSSI qui est la mesure de la puissance du signal reçu, de sorte que les appareils sélectionnent un réseau possédant le plus fort RSSI. Il existe également d'autres mécanismes de sélection de réseaux qui ont des objectifs différents comme l'économie d'énergie, l'amélioration de la qualité de service en connectant l'appareil IdO à une interface réseau à efficacité énergétique ou QoS supérieure par rapport aux mesures RSSI. Comme chaque réseau a des caractéristiques spécifiques et afin d'augmenter la précision de décision, un processus de *handover* vertical doit évaluer chaque réseau disponible avant la prise de décision.

Dans (Sha *et al.*, 2017), les auteurs ont présenté leur solution ARTPoS *(Adaptive Radio and Transmission Power Selection system)* qui prend en considération plusieurs technologies sans fil (comme WiFi, Zigbee et Bluetooth) et sélectionne le réseau qui convient le mieux au trafic et aux conditions du réseau en minimisant la consommation d'énergie. Compte tenu de la nature dynamique des réseaux dans le domaine de l'Internet des objets, une seule technologie ne peut pas relever les défis dans des conditions variables du réseau et du trafic. C'est pour cela, dans cette recherche, les auteurs ont conçu ARTPoS. L'architecture du système est décrite à la figure 2.6. Le module de sélection choisit l'interface radio la plus adaptée en fonction du

débit binaire spécifié par le service associé à l'appareil et de la capacité résiduelle du réseau. De plus, ARTPoS intègre des modules de contrôle des interfaces radio (WiFi, Bluetooth et Zigbee). Chaque contrôleur gère l'état de l'interface (c'est à dire activée ou désactivée) et calcule la consommation d'énergie en fonction de la décision prise par le module de sélection. Finalement, l'architecture d'ARTPoS intègre un module d'interface graphique qui gère les interactions avec l'utilisateur.

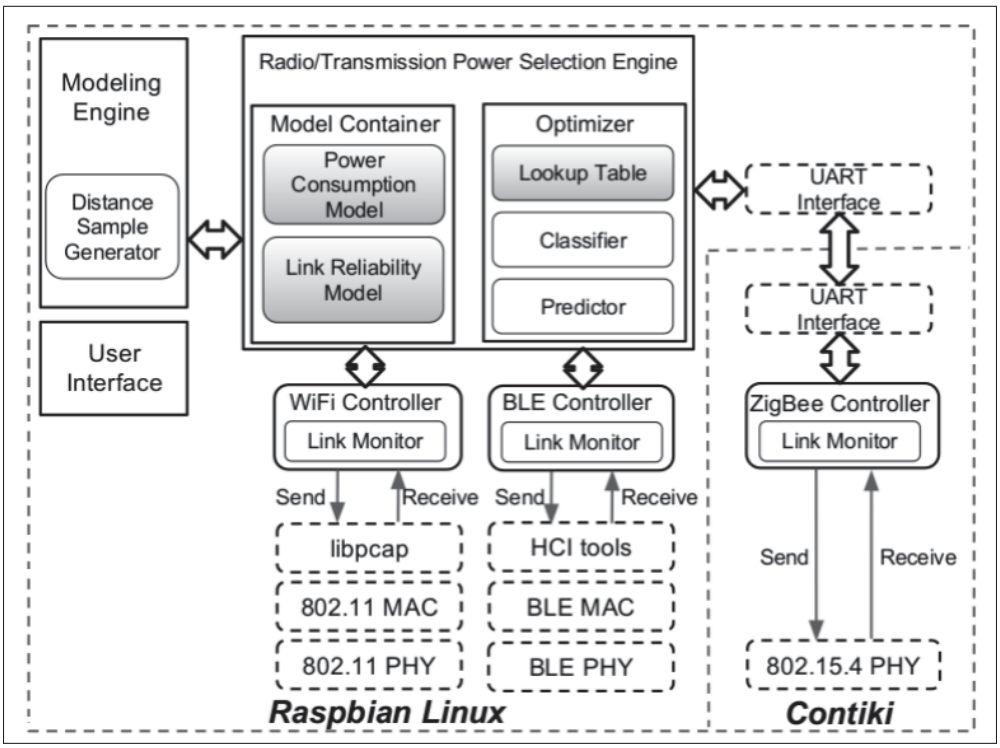

Figure 2.6 Architecture d'ARTPoS (Sha *et al.*, 2017)

Le travail de (Tosh & Sengupta, 2015) aborde le problème de la sélection de réseaux dans un environnement hétérogène en utilisant la théorie des jeux. Bien que les architectures IdO offrent une variété de ressources réseau, il est possible que les objets connectés ne bénéficient pas de toutes les conditions favorables pendant la période de service. Par exemple, une technologie pourrait offrir un meilleur débit, mais consommer plus d'énergie, tandis qu'une autre pourrait offrir une transmission de données fiable, mais coûterait plus cher. De plus, les appareils IdO

risquent de connaître une saturation surtout que le nombre d'objets sélectionnant le même réseau d'accès augmente et que la demande de bande passante pour le réseau augmente. Pour remédier à ce problème, les auteurs modélisent un mécanisme de sélection de réseau comme un jeu non coopératif en formulant une fonction d'utilité basée sur les paramètres suivants : capacité du réseau, coût d'utilisation des ressources et la latence. Entre autres, ils cherchent à trouver l'interface réseau convenable qui permettrait de maximiser la fonction d'utilité définie.

Dans cet article, les auteurs ont considéré un scénario dans lequel M réseaux d'accès sont disponibles et offrent des services distincts à un coût variable. Pour exploiter cette hétérogénéité, N objets connectés entrent en concurrence pour accéder à l'un ou un sous-ensemble des M réseaux. Chaque objet connecté est équipé d'une multitude d'interfaces réseau et peut accéder simultanément à plusieurs réseaux d'accès. Pour ce problème, les auteurs ont considéré les appareils IdO comme des joueurs. Ces derniers sont davantage intéressés par l'utilisation simultanée de plusieurs interfaces afin d'optimiser leurs performances globales en utilisant d'une manière efficace les réseaux sélectionnés. Les auteurs ont conçu une fonction d'utilité pour les objets connectés équipés de plusieurs interfaces en tenant en compte de la latence du réseau et du coût d'envoi des données par unité de temps.

2.3.1.2 Approches basées sur l'utilisateur/appareil IdO

Les paramètres utilisés dans ce type d'approche sont les préférences de l'utilisateur, la capacité de la batterie, des valeurs seuils de *handover* fixés par l'utilisateur, etc. Ces paramètres peuvent aider les appareils à interfaces multiples à économiser de l'énergie et améliorer la qualité de service des applications qu'ils servent. Dans le reste de cette section, nous présentons les travaux existants qui ont abordé le problème de la sélection de réseau en se basant sur les métriques de l'utilisateur ou/et l'appareil IdO.

Dans (Awad *et al.*, 2016), les auteurs ont abordé un scénario de réseau dans lequel plusieurs technologies d'accès radio peuvent être exploitées simultanément par les utilisateurs. Ils ont proposé un mécanisme dynamique de sélection de réseau permettant une exploitation efficace

en termes d'énergie des ressources disponibles. Dans l'approche proposée, un utilisateur assume la responsabilité de décision de sélection du réseau en donnant ses préférences. Les auteurs ont formulé le problème de sélection comme un problème d'optimisation où la fonction objective dépendrait des préférences de l'utilisateur dans le choix des valeurs des poids affectées aux critères de la sélection du réseau. La fonction objective permet de minimiser une fonction d'utilité qui prend en considération trois paramètres, à savoir la consommation d'énergie, le coût monétaire et la latence. Chacun de ces trois critères est multiplié par un coefficient précisé par l'utilisateur.

Dans la littérature, il existe d'autres travaux qui ont considéré la qualité d'expérience (QoE) dans la décision de sélection du réseau comme (Liu *et al.*, 2016). Dans cette recherche, les auteurs proposent un mécanisme de *handover* axé sur l'utilisateur et basé le framework MIH du standard IEEE 802.21 (Khattab & Alani, 2013). Son objectif est de maintenir un QoE acceptable pour différentes applications et de sélectionner un réseau approprié en fonction des préférences de l'utilisateur. Dans cette recherche, le coût et le QoE sont les préoccupations les plus importantes à considérer lors du processus de la sélection du réseau. Étant donné que les utilisateurs ont également des exigences différentes en matière de coût et QoE pour différents services d'application, il est nécessaire d'inclure les préférences des utilisateurs dans les décisions de sélection. Les procédures du mécanisme proposé sont décrits dans la figure 2.7. Les *QoE-Estimators* collectent les paramètres de réseau et de service afin de prédire la QoE en termes de MOS (*Mean Opinion Score*). Ensuite, la valeur de MOS prévue est envoyée à un algorithme de décision de *handover*. Les utilisateurs peuvent définir leurs préférences en fonction de leurs exigences. La décision prise par l'algorithme de décision est ensuite transmise au module *MIH Function* qui permet d'exécuter le processus du *handover*. Enfin, des commandes MIH sont produites par cette procédure pour gérer les différentes interfaces réseau.

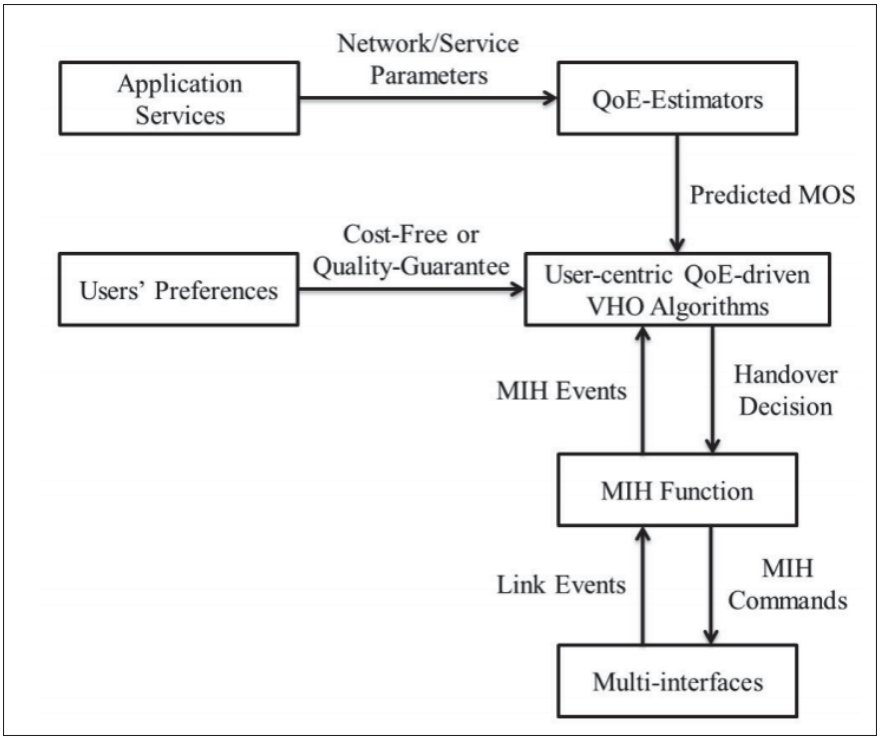

Figure 2.7 Les procédures du mécansime de *handover* centré sur l'utilisateur (Liu *et al.*, 2016)

2.4 Les problèmes d'ordonnancement des flux et l'allocation des ressources dans les réseaux M2M

Dans l'industrie, plusieurs applications IdO sont implémentées telles que les services destinés pour la santé, les applications de la ville intelligente (comme les voitures connectées, les applications de surveillance des feux de signalisation, services de congestion du trafic). La plupart de ces applications sont destinées pour le suivi des objets intelligents en temps réel. Prenons l'exemple de l'application de contrôle du trafic. Cette application permet aux citoyens d'obtenir des informations sur le trafic et la circulation dans la ville. Dans le cas d'un accident (embouteillages importants), les données doivent être envoyées en temps réel avec des valeurs de latence faibles afin que les citoyens puissent choisir les routes les plus rapides et les moins encombrées. En outre, une faible consommation d'énergie est également une exigence pour les réseaux M2M puisque ces équipements sont alimentés par la batterie. Pour remédier à ce problème, de nombreux protocoles de communications à faible consommation d'énergie sont

utilisés comme Zigbee et LoRa. Aussi, des nombreux mécanismes d'ordonnancement et d'allocation de ressources ont été proposés dans la littérature. Ces mécanismes permettent d'allouer les ressources réseau des équipements IdO afin de répondre aux exigences des applications IdO tout en garantissant la qualité de service des applications. Dans cette section, nous présentons des travaux qui ont proposé des mécanismes d'ordonnancement et d'allocation de ressources dans les réseaux IdO.

2.4.1 SIA

Dans cette recherche, *Angelakis et al.* se sont concentrés sur des équipements IdO dotés d'interfaces multiples. Chaque interface donne accès à une collection de ressources hétérogènes telles que le débit binaire, l'espace mémoire tampon, etc. Puisque les équipements IdO sont appelés à fournir une grande variété de services avec des différentes exigences, il est évident qu'une allocation efficace de ces ressources IdO améliorerait les performances de ce réseau. Dans ce travail, les auteurs ont abordé le problème de l'attribution des services aux interfaces de l'équipement en question. Ils ont noté ce problème *Service-to-Interface Assignment (SIA)*. Une instance du problème SIA est présentée à la figure 2.8.

Dans cette figure, on peut observer une instance d'une allocation dans un appareil IdO à trois interfaces (i = 1, 2, 3) offrant trois ressources différentes (k = 1, 2, 3). Avant l'assignation, deux services exigent des ressources ; le premier (jaune) et le deuxième (orange) service demandent d1 = (8, 10, 13) et d2 = (8, 5, 0) de ces ressources, respectivement. Après l'attribution, deux divisions de service se sont produites : les demandes du premier service (jaune) sont servies par les deuxième et troisième interfaces, tandis que les ressources du deuxième service (orange) sont divisées en deux interfaces ; le premier et le troisième. Cette figure montre un exemple du problème du SIA où l'appareil IdO devrait décider la manière avec laquelle les services sont assignés aux interfaces disponibles.

Dans (Angelakis *et al.*, 2016), les auteurs proposent une solution au problème de l'assignation des services, avec des exigences différentes, aux ressources réseau de l'équipement IdO, tout

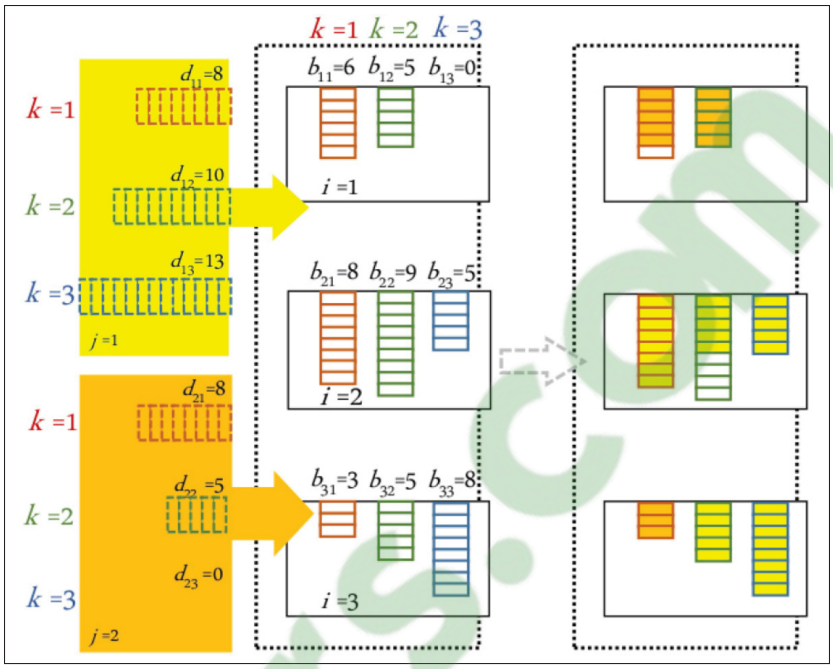

Figure 2.8 Une instance du problème de *SIA* (Angelakis *et al.*, 2016)

en minimisant le coût d'utilisation des interfaces. Ils ont présenté une formulation en programmation linéaire en nombres entiers mixtes (*Mixed Integer Linear Programming*) du problème de l'assignation de services aux interfaces avec des ressources hétérogènes. L'objectif est de minimiser le coût total d'utilisation des ressources, tout en satisfaisant la qualité de service des applications. Ils ont considéré le coût total comme la somme du coût d'utilisation de chaque unité de ressources et le coût de l'activation de chaque interface utilisée pour servir un service.

Les auteurs ont conçu deux solutions pour le problème *SIA*. Dans la première solution, les ressources disponibles des interfaces peuvent répondre à l'ensemble des demandes en une itération. Ils ont trouvé la solution optimale pour le problème à l'aide d'un *solver*, qui est considéré comme une référence pour les deux algorithmes qu'ils ont développé pour approcher la solution optimale. Dans la deuxième solution, lorsque les demandes des services dépassent les ressources disponibles des objets IdO, le mécanisme conçu peut répondre à ces demandes en plusieurs itérations. En d'autres termes, ils ont utilisé un nombre approprié d'itérations pour

optimiser l'allocation des ressources, tout en obtenant le coût total minimal d'utilisation des interfaces de l'appareil IdO. Ils ont appelé ce problème le *SIA multiround*.

Tandis que cette recherche montre des résultats prometteurs concernant la minimisation de consommation d'énergie en assignant les services aux ressources hétérogènes, la formulation mathématique ne considère pas l'échéance des flux IdO et le délai de transmission du flux en utilisant une interface donnée. De plus, le mécanisme d'assignation des services est implémenté au niveau de chaque appareil IdO et donc, la décision de ce mécanisme ne tient pas en compte la capacité de la passerelle (tient en compte seulement des ressources de l'appareil IdO en question) ce qui peut entraîner une saturation au niveau de la passerelle.

2.4.2 Mécanismes d'ordonnancement centralisés et distribués

L'hétérogénéité des déploiements IdO, comme expliqué dans la section 2.2, pose de nouveaux problèmes pour les chercheurs et l'industrie dans les réseaux M2M. Afin d'exploiter d'une manière efficace les ressources disponibles dans un tel réseau, des mécanismes d'ordonnancement de flux ont été proposés pour les assigner aux interfaces appropriées. Dans la littérature, il existe deux types de mécanismes d'ordonnancement : centralisés et distribués. Pour les mécanismes centralisés, les décisions d'ordonnancement sont prises généralement au niveau de la passerelle ou un contrôleur central. Ce mécanisme est plus efficace pour l'économie d'énergie. Cependant, avec l'augmentation exponentielle du nombre d'appareils IdO, les mécanismes centralisés auront de sérieuses implications en termes d'évolutivité et de fiabilité. Par contre, les mécanismes distribués, où les décisions d'ordonnancement sont traitées à chaque appareil IdO, peuvent résoudre le problème de saturation d'un contrôleur central. Mais, ce type d'ordonnancement peut ne pas être souhaitable car les calculs exécutés à chaque appareil IdO causent une consommation d'énergie supplémentaire. En conséquence, un compromis doit être envisagé entre la consommation de l'énergie et la fiabilité de l'architecture proposée lors du choix des mécanismes d'ordonnancement.

Di Wu et al. (Wu *et al.*, 2015) ont abordé le problème de la gestion des flux et de la mobilité en milieu urbain où des réseaux hétérogènes coexistent pour fournir une multitude de services aux usagers des objets connectés. Les auteurs ont utilisé la technologie du *Software Defined Networks (SDN)* pour contrôler les flux dans les architectures IdO. Ils ont donc proposé UbiFlow, qui est un mécanisme basé sur des contrôleurs distribués qui permettent l'ordonnancement et la gestion des flux échangés entre les passerelles et les appareils IdO. Cette solution sert à la gestion de la mobilité, l'optimisation de commutation entre les réseaux hétérogènes. Ces contrôleurs échangent des informations sur les appareils mobiles/IdO et les passerelles afin d'avoir une vue globale sur le réseau avant de sélectionner une interface réseau/point d'accès et réaliser un équilibrage de charge efficace.

Dans (Fu *et al.*, 2012), les auteurs ont abordé le problème d'envoi des données en temps réel tout en minimisant la consommation d'énergie des appareils IdO. Pour chaque type de données reçues des appareils, les auteurs définissent une période de suivi comme une contrainte temporelle. Ils considèrent les données comme valides s'ils sont collectés par un nœud IdO et reçus par la passerelle IdO dans la même période. Et donc, nous pouvons considérer la fin de la période de suivi comme une échéance avant laquelle le flux d'un service donné doit être reçu. Pour garantir un compromis entre la validité des données et la consommation d'énergie, les auteurs ont proposé deux mécanismes d'ordonnancement (centralisé ECR et distribué EDR) pour l'envoi des données à la passerelle. En ECR, la passerelle IdO est responsable de l'ordonnancement des flux envoyés par les nœuds IdO. Le compromis entre la validité des données et la consommation énergétique est transformé en problème de minimisation d'énergie et formulé comme un problème d'optimisation linéaire. Cependant, pour EDR, la décision de la transmission des données et l'ordonnancement sont pris par l'appareil IdO. Pour satisfaire le compromis de la validité énergétique, les auteurs ont conçu un algorithme d'ajustement dynamique basé sur un seuil qui permet à l'appareil d'ajuster sa période d'inactivité.

2.5 Discussions

Dans cette section, nous avons présenté un bref résumé (tableau 2.4) des solutions les plus pertinentes par rapport à notre problème de recherche. Le tableau suivant met en évidence les principales différences entre ces recherches et leurs limitations.

Tableau 2.4 Tableau comparatif des travaux connexes

Travaux	**Méthodologie**	**Contributions**	**Limitations**
(Sha *et al.*, 2017)	- Les auteurs ont formulé le problème de sélection d'interfaces comme un problème d'optimisation et ils ont modélisé un système de sélection qui repose sur la puissance du signal reçu et du taux de réception des paquets.	- La conception d'un système noté ARTPoS (Adaptive Radio and Transmission Power Selection) qui met à disposition plusieurs technologies sans fil et sélectionne l'interface qui est la plus appropriée aux conditions du réseau. - La formulation du problème de la sélection des interfaces comme un problème d'optimisation et le développement d'une solution satisfaisant les exigences des applications mobiles.	Les auteurs dans cette recherche ont considéré la minimisation de la perte des données et la consommation d'énergie sans tenir compte de l'échéance des flux reçus.

Tableau 2.4 Tableau comparatif des travaux connexes (suite)

Travaux	**Méthodologie**	**Contributions**	**Limitations**
(Wu *et al.*, 2015)	- Pour contrôler les flux et optimiser les *handover* dans l'architecture IdO à réseaux hétérogènes, *Ubiflow* utilise des contrôleurs SDN distribués. - Une structure *overlay* distribuée basée sur le hachage est proposée pour maintenir l'évolutivité du réseau.	- Un algorithme d'assignation optimal pour faire correspondre les meilleurs points d'accès disponibles aux équipements IdO, avec une analyse de l'état du réseau et des demandes de flux en tant qu'entrées. - Un mécanisme d'équilibrage de charge pour les contrôleurs distribués pour remédier au problème de la surcharge.	- La formulation du problème prend seulement en considération le *handover* homogène entre les points d'accés. - La formulation ne tient pas en compte du délai du *handover* lors du basculement vers un autre point d'accès.
(Awad *et al.*, 2017)	- Dans cet article, les auteurs utilisent une approche basée sur un modèle linéaire, combinée à une approche basée sur la théorie des jeux, pour résoudre le problème de la sélection des réseaux dans un environnement hétérogène.	- La formulation d'un problème d'optimisation multi-objective qui vise à sélectionner les interfaces d'une manière optimale en fonction des objectifs de l'utilisateur et des caractéristiques du réseau. - Le développement d'un mécanisme d'assignation dynamique qui vise à maximiser le temps de fonctionnement de l'appareil de l'utilisateur.	- Les auteurs ont modélisé la demande des services, mais ils n'ont pas donné une formulation exacte des délais du transfert des données, délai de traitement, délai du *handover*, etc. - Les auteurs n'ont pas considéré et n'ont pas conçu un mécanisme de *handover* qui permet de changer la connexion d'un appareil IdO d'une interface à une autre.

Tableau 2.4 Tableau comparatif des travaux connexes (suite)

Travaux	**Méthodologie**	**Contributions**	**Limitations**
(Angelakis *et al.*, 2016)	Les auteurs ont présenté une formulation en programmation linéaire en nombres entiers mixtes (MILP) du problème de l'assignation des services aux interfaces avec des ressources hétérogènes.	- La formulation du problème d'assignation de services IdO aux ressources réseau comme un problème d'optimisation. - La preuve de la complexité du problème SIA et le développement de deux algorithmes pour résoudre le problème d'optimisation.	- La formulation mathématique ne considère pas le délai de transmission du flux en utilisant une interface donnée et l'échéance de chaque flux échangé. - Le mécanisme d'assignation ne tient pas en compte de la capacité de la passerelle (tient en compte seulement des ressources de l'appareil IdO en question) ce qui peut entraîner une saturation au niveau de la passerelle. - Cette recherche ne propose pas une implémentation dans un environnement réel de leur solution.

Dans notre recherche, nous avons modélisé le problème d'assignation de flux comme un problème d'optimisation sous forme de programmation non linéaire en nombres entiers (*INLP : Interger Non Linear programming*).

CHAPITRE 3

MÉTHODOLOGIE

3.1 Introduction

Dans ce chapitre, nous introduisons la description de notre système et notre solution qui permet de répondre à la problématique de recherche. Ensuite, nous présentons la modélisation mathématique qui est sous forme d'un problème d'optimisation où l'objectif est de maximiser la quantité de données acceptée par la passerelle. Enfin, nous détaillons les solutions proposées à ce problème qui sont G-OFAP et D-OFAP.

3.2 Description du système

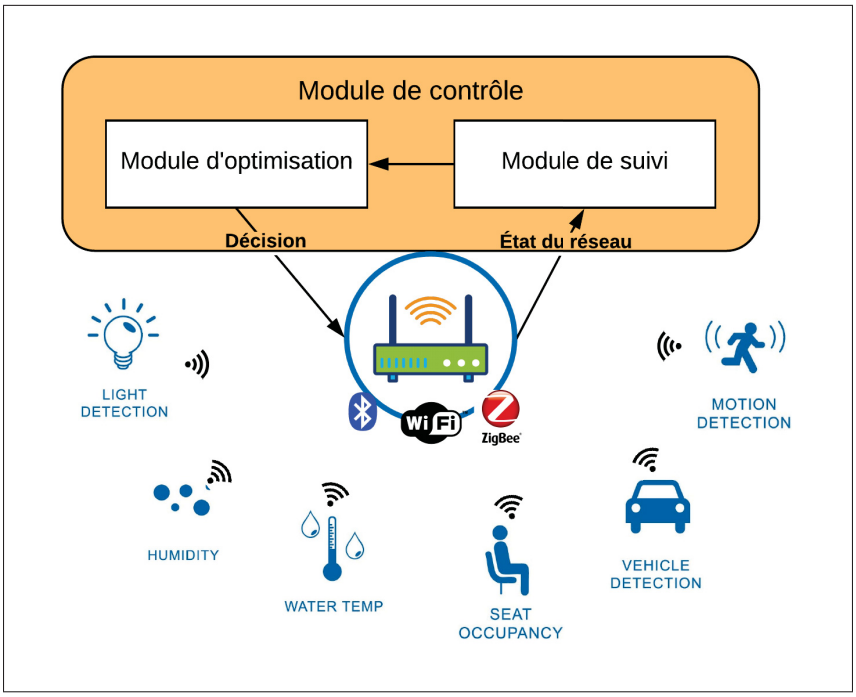

Figure 3.1 Architecture du système (Amor *et al.*, 2019)

Dans notre travail, nous considérons une architecture IdO qui consiste à une passerelle équipée de plusieurs interfaces réseaux telles que Bluetooth, WiFi et Zigbee. La passerelle communique avec un ensemble d'équipements IdO comme décrit à la figure 3.1. Chaque appareil IdO est équipé d'au moins une interface réseau. Nous supposons que la force du signal est largement suffisante pour pouvoir se connecter à la passerelle et que le temps est divisé en créneaux horaires dans lesquels les appareils envoient des données discrètes. Les flux envoyés à chaque créneau horaire sont connus et chaque équipement envoie au plus un flux de donnée durant un créneau horaire.

Dans notre problème, nous considérons l'assignation des flux envoyés durant un intervalle de temps aux interfaces réseau disponibles. Supposons que chaque flux devrait être reçu avant son échéance respective qui est le délai toléré pour chaque flux IdO. Dans ce travail, nous visons à affecter un ensemble de flux donné, transmis dans la même plage horaire, aux interfaces réseau disponibles tout en maximisant la quantité de données valides reçues par la passerelle.

Le nombre augmentant des appareils IdO dans les zones urbaines denses peut inévitablement mener à une saturation du réseau et une dégradation de service. En conséquence, la perte de donnée peut être énorme. Pour remédier à ce problème, les passerelles dans notre système, intègrent un module de contrôle qui permet l'assignation des flux aux interfaces réseau de la passerelle dynamiquement comme détaillée dans la section 3.2.1. En utilisant ce module, la passerelle peut aussi prendre la décision pour migrer les flux d'une interface saturée vers une autre légèrement chargée pour améliorer la qualité de service et augmenter la quantité de données acceptées.

3.2.1 Module de contrôle de la passerelle

Notre solution applique l'assignation des flux aux interfaces en exécutant les modules suivants :

3.2.1.1 Module de suivi

Ce module est responsable de collecter les informations des appareils IdO et ses services comme le débit binaire d'envoi, les exigences de service en termes de latence, échéances, etc. Ces informations sont extraites des messages échangés (*DISCOVER message*) lors du processus de l'établissement de connexion. Ce processus ainsi que ce type de message sont expliqués dans la section 3.2.2. Ces informations collectées sont stockées pour être utilisées par suite par le module d'optimisation. En utilisant ces données, la passerelle peut avoir une vue globale sur les flux envoyés et leurs exigences en termes de qualité de service.

3.2.1.2 Module d'optimisation

Après l'exécution du module de suivi, la passerelle va affecter les flux aux interfaces adéquates en utilisant le module d'optimisation. Ce module exécute les algorithmes proposés à la section 3.4. Cette assignation de flux dépend de la capacité de la passerelle et des exigences des services. L'objectif de ce module consiste à maximiser l'utilisation du réseau en maximisant la quantité des flux attribués à la passerelle. Ce module prend en entrée l'ensemble des appareils IdO et la capacité de leurs interfaces réseaux, la taille et les échéances des flux envoyés. Il permet d'attribuer les flux à la meilleure interface réseau qui garantit la demande des flux en terme de bande passante et de délais tolérés. Le module d'optimisation peut exiger d'un appareil IdO de passer d'une interface à une autre. Par conséquent, nous concevons un protocole de *handover* comme décrit dans la section suivante.

3.2.2 Protocole de *handover*

Pour changer d'une interface réseau à une autre, nous avons conçu un protocole qui permet d'assurer un *handover* en minimisant le temps de discontinuité de service. Ce protocole est décrit par le diagramme de séquence à la figure 3.2. Tout d'abord, lorsqu'un appareil se connecte à la passerelle, il envoie un message *DISCOVER* qui permet à la passerelle de retrouver des informations sur le flux et l'appareil tels que l'*ID*, les interfaces réseaux disponibles à l'appareil

et le débit binaire d'envoi des données. Ensuite, la passerelle exécute l'algorithme d'assignation de flux. Le résultat de cet algorithme attribue d'une manière optimale l'appareil à l'interface réseau en tenant compte des exigences de l'application et de la passerelle comme l'échéance des flux, la capacité de la passerelle et les interfaces disponibles dans l'objet connecté. Un appareil est désormais incapable de se connecter à la passerelle si la capacité est dépassée. Dans ce cas, l'appareil qui a initié la connexion reçoit un message *NOT AUTHORIZED*. Les appareils qui doivent changer vers une autre interface reçoivent le message *SWITCH-InterfaceID* qui indique l'identifiant de l'interface vers laquelle ils doivent changer. Ensuite, pour établir la connexion en utilisant l'interface sélectionnée, deux messages seront échangés *HELLO* et *HELLO-ACK*. Après la réception de *HELLO-ACK*, la connexion est établie et donc l'appareil peut commencer à envoyer les données en utilisant la nouvelle interface sélectionnée.

3.3 Formulation du problème

Dans le cadre de notre méthodologie, nous proposons un modèle d'optimisation qui permet de répondre à nos premier et deuxième objectifs définis dans la section 1.4 qui consistent à maximiser la quantité de données acceptées par la passerelle en satisfaisant la qualité de service des applications IdO.

Dans cette section, nous désignons l'ensemble des appareils (équipements) IdO par $\mathcal{I}$. $\mathcal{F} = \{f_i\}_{\forall i \in \mathcal{I}}$ désigne l'ensemble des flux où f_i est le flux envoyé par l'appareil i. Dans ce travail, nous supposons que chaque appareil i envoie au plus un flux f_i dans le créneau horaire considéré. La taille d'un flux f_i est égale à S_i. Chaque appareil IdO dispose d'une capacité de batterie limitée à C_i. Nous notons la consommation d'énergie par seconde pour la transmission de données à travers l'interface j par E_j. Soit $\mathcal{J}$ l'ensemble d'interfaces réseau disponibles dans la passerelle. Chaque interface j dispose d'un taux de transmission qui est égal à T_j. La capacité réseau de l'interface j est égale à R_j. A est une matrice donnée avec $|\mathcal{I}|$ lignes et $|\mathcal{J}|$ colonnes, représentant les interfaces réseau disponibles dans les appareils IdO. Un appareil i est équipé de l'interface j si $A_{i,j} = 1$, sinon $A_{i,j} = 0$. Nous définissons la variable binaire de décision $x_{i,j}$ pour représenter l'assignation des flux aux interfaces de la passerelle :

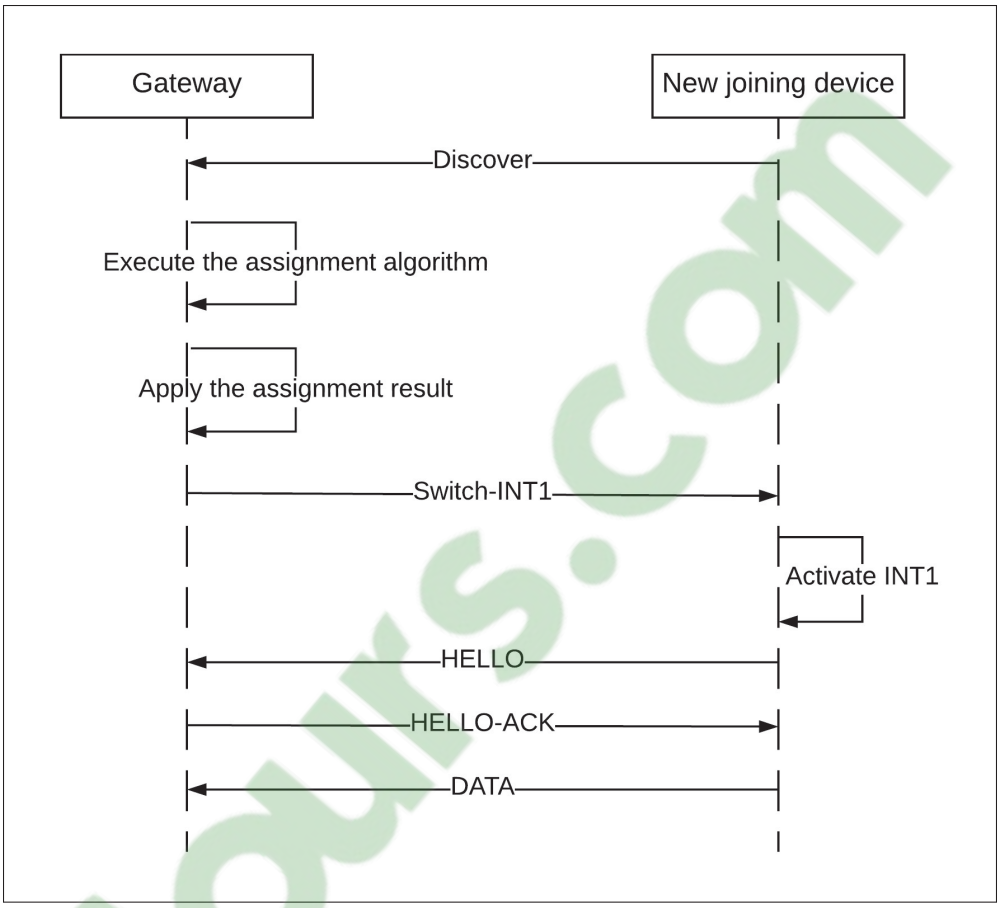

Figure 3.2 Le protocole de *handover* proposé

$$x_{i,j} = \begin{cases} A_{i,j} & \text{si le flux } f_i \text{ est affecté à l'interface } j \text{ de la passerelle,} \\ 0 & \text{sinon.} \end{cases}$$

3.3.1 Fonction objective

L'objectif de notre problème d'optimisation d'assignation de flux est de maximiser la quantité de données valides acceptées par la passerelle tout en satisfaisant les contraintes d'échéances, d'énergie et de la capacité du réseau. Il peut être formulé comme suit :

$$\max \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} S_i \, x_{i,j} \tag{3.1}$$

3.3.2 Contraintes

La fonction objectif définie à l'équation 3.1 est soumis aux contraintes suivantes :

3.3.2.1 Capacité d'énergie

Cette contrainte garantit que lorsque le flux f_i est transmis à travers l'interface j, sa consommation d'énergie ne doit pas dépasser la capacité de la batterie C_i de l'appareil. Cette consommation d'énergie est calculée en multipliant le délai de transmission par la consommation d'énergie par seconde E_j en utilisant l'interface j.

$$\sum_{j \in \mathcal{J}} \frac{S_i}{T_j} \, E_j \, x_{i,j} \leq C_i, \;\; \forall i \in \mathcal{I}. \tag{3.2}$$

3.3.2.2 Échéances des flux

Chaque flux f_i doit être reçu avant son échéance D_i. Nous définissons trois délais pour chaque flux f_i : délai de *handover* $\theta_{i,j}$, délai de transmission $\beta_{i,j}$ et délai de mise en mémoire tampon $\alpha_{i,j}$.

Le délai de commutation est le temps nécessaire pour passer de l'interface connectée actuelle à l'interface j et pour établir la connexion. Ce mécanisme est décrit à la section 3.2.2. Nous définissons $\theta_{i,j}$ comme le délai de commutation de l'appareil IdO i pour passer vers l'interface j à partir de l'interface connectée actuelle. Quant au délai de transmission, il est égal à la taille du flux, S_i, divisé par le taux de transmission T_j. Par conséquent, $\beta_{i,j}$ est défini comme suit :

$$\beta_{i,j} = \frac{S_i}{T_j}, \;\; i \in \mathcal{I}, \; j \in \mathcal{J}. \tag{3.3}$$

Le délai de mise en mémoire tampon $\alpha_{i,j}$ représente le délai de transmission des flux stockés dans la mémoire tampon. Ce délai est la somme de deux termes : (a) le délai de transmission de Q_j, qui est la quantité de données restantes dans la mémoire tampon qui est déjà transmise par l'interface j. (b) Le second terme est le délai de transmission des flux envoyés aux mêmes créneaux horaires que f_i et ordonnancés pour être transmis avant f_i. Dans notre travail, nous supposons que les flux dont l'échéance est la plus proche sont programmés pour être transmis en premier lorsqu'ils sont envoyés dans le même créneau horaire et affectés à la même interface. $\mathcal{I}'$ désigne l'ensemble des appareils qui envoient des flux avec une échéance antérieure à f_i où $\mathcal{I}' = \{i' \in \mathcal{I} \; / \; \{i' \neq i\}$ and $D_{i'} < D_i\}$. Par conséquent, $\alpha_{i,j}$ est défini comme suit :

$$\alpha_{i,j} = \frac{Q_j}{T_j} + \sum_{i' \in \mathcal{I}'} \frac{S_{i'} \, x_{i',j}}{T_j}, \;\; i \in \mathcal{I}, \; j \in \mathcal{J}. \tag{3.4}$$

La contrainte du délai garantit la satisfaction de l'échéance du flux. Lorsque le flux f_i est envoyé à travers l'interface j, le délai de commutation $\theta_{i,j}$ plus le délai de transmission $\beta_{i,j}$ à l'aide de l'interface j plus le délai de mise en mémoire tampon $\alpha_{i,j}$ ne doit pas dépasser l'échéance D_i de f_i.

$$\sum_{j \in \mathcal{J}} \theta_{i,j} \, x_{i,j} + \sum_{j \in \mathcal{J}} \beta_{i,j} \, x_{i,j} + \sum_{j \in \mathcal{J}} \alpha_{i,j} \, x_{i,j} \leqslant D_i, \;\; \forall i \in \mathcal{I}. \tag{3.5}$$

3.3.2.3 Assignation des flux

Nous supposons que les flux ne sont pas répartis sur différentes interfaces. Cette contrainte garantit que le flux f_i est transmis à travers une interface unique.

$$\sum_{j \in \mathcal{J}} x_{i,j} \leqslant 1, \ \ \forall i \in \mathcal{I}. \tag{3.6}$$

3.3.2.4 Capacité du réseau

Cette contrainte garantit que la quantité de données affectée à l'interface j ne dépasse pas la capacité du réseau R_j de cette interface.

$$\sum_{i \in \mathcal{I}} S_i \, x_{i,j} \leq R_j, \ \ \forall j \in \mathcal{J}. \tag{3.7}$$

Les notations sont résumées dans le tableau 3.1.

Tableau 3.1 Résumé des notations

Notation	Description
$\mathcal{I}$	Ensemble d'appareils (équipements) IdO
$\mathcal{J}$	Ensemble d'interfaces réseau disponibles dans la passerelle
$\mathcal{F}$	Ensemble des flux
T_j	Taux de transmission de l'interface réseau j
Q_j	Quantité de données restantes dans la mémoire tampon (en Mo) déjà transmise en utilisant l'interface j
D_i	Échéance du flux f_i
C_i	Capacité de la batterie de l'appareil i
E_j	Consommation d'énergie par seconde pour la transmission de données à travers l'interface j
R_j	Capacité de l'interface réseau j
S_i	La taille du flux f_i
A	Matrice représentant les interfaces réseaux disponibles aux appareils IdO
$\theta_{i,j}$	Délai de *handover* du flux i de l'interface actuelle vers l'interface j (en secondes)
$\alpha_{i,j}$	Délai de transmission des flux qui sont ordonnancés pour être transmis avant f_i en utilisant l'interface j
$\beta_{i,j}$	Délai de transmission de f_i en utilisant l'interface j
$x_{i,j}$	Variable de décision binaire indiquant que le flux f_i est affecté à l'interface j

Notre problème d'assignation de flux est un problème d'optimisation non linéaire en nombre entier (ONLNE), dû à la contrainte 3.5 (Amor *et al.*, 2019). La complexité des solveurs ONLNE étant élevée, nous proposons deux algorithmes approximatives pour résoudre ce problème dans la section suivante.

3.4 Solutions proposées

Dans cette section, nous concevons deux algorithmes pour notre problème d'assignation optimale des flux. Ensuite, nous présentons le *Baseline* avec lequel nous comparons nos solutions et qui est proposé par (Angelakis *et al.*, 2016) pour résoudre le problème de *Service-Interface-Assignment problem* (SIA).

3.4.1 G-OFAP : Approche Greedy pour OFAP

Cette approche est basée sur le problème du *Knapsack*. Nous modélisons chaque interface sous forme de *Knapsack*. La capacité de chaque *Knapsack* est égale à la capacité de l'interface réseau. Nous considérons les flux comme les objets à ajouter dans le sac. Nous modélisons le poids d'un flux par sa taille en Mo (Amor *et al.*, 2019).

Notre algorithme 3.1 prend en entrée la liste des flux candidats, l'ensemble des interfaces disponibles, les échéances des flux, la capacité disponible à la batterie de l'équipement, la consommation d'énergie par interface et la capacité réseau des interfaces. Le résultat de l'algorithme est une matrice binaire X, avec $|\mathcal{I}|$ lignes et $|\mathcal{J}|$ colonnes, qui représente l'assignation des flux aux interfaces disponibles. Au début, les interfaces réseau sont triées en fonction de leurs capacités réseau et l'ensemble des flux est trié en fonction de leurs tailles dans l'ordre croissant (ligne 3 et 4). Ensuite, nous appliquons la stratégie de meilleur ajustement (*Best Fit*). En d'autres termes, l'algorithme affecte un flux à une interface, qui a la plus faible capacité réseau, parmi les interfaces disponibles. Si l'interface a suffisamment de capacité pour accepter le flux f_i, on utilise la fonction *calcul_delay* (ligne 13) pour calculer la somme des trois délais définis dans la formulation du problème à savoir, le délai de transmission, de commutation et de la

Algorithme 3.1 G-OFAP (Amor *et al.*, 2019)

```
1  Input : I, F, J, R_j, D_i, C_i, E_j
2  Output : Assignment matrix X.
3  List interfaces = Sort(J)
4  List flux = Sort(F)
5  List somme = 0
6  X = 0
7  for i in flux do
8      Boolean check = False
9      j = 0
10     while (j ≤ size(interfaces)) and (check == False) do
11         inter = interfaces[j]
12         if (somme[j] + S_i ≤ R_j) then
13             delay = calcul_delay(f, F, D_i, T_j)
14             Tr_delay = S_i / T_j
15             if (delay ≤ D_i) and (Tr_delay.E_j ≤ C_i) then
16                 check = True
17                 X[i][j] = 1
18                 somme[j] = somme[j] + S_i
19             end
20         end
21         j = j + 1
22     end
23 end
24 return X
```

mise en mémoire tampon. L'interface sélectionnée doit satisfaire les contraintes de l'échéance du flux D_i et la capacité de la batterie de l'appareil C_i (ligne 15). Cependant, si un flux n'est affecté à aucune interface, il sera rejeté. L'algorithme itère l'ensemble des interfaces et s'arrête lorsque tous les flux sont visités.

3.4.2 D-OFAP : Approche basée sur la programmation dynamique

Afin de résoudre notre problème, nous proposons une deuxième solution basée sur la programmation dynamique qui est décrite à l'algorithme 3.3. Cette solution est basée sur le vecteur W calculé à l'aide d'un algorithme de programmation dynamique (algorithme 3.2) qui four-

nit une solution exacte à un problème de *Knapsack* simple. L'idée est donc de trouver un sous-ensemble optimal à partir de l'ensemble global des flux. Ce sous-ensemble contient les éléments qui permettent de maximiser la quantité des données affectées à une interface particulière. Ensuite, nous appliquons le reste des contraintes sur ce sous-ensemble pour générer un résultat valide (Amor *et al.*, 2019).

Tout d'abord, nous expliquons l'algorithme de la programmation dynamique qui est utilisé pour résoudre le problème du Knapsack. Nous avons basé notre conception sur l'implémentation de (Neal, 2012) qui donne une solution optimale pour le problème de *Subet-Sum Problem*. Nous avons appliqué cet algorithme pour assigner à chaque itération la plus grande quantité de flux possible à l'interface sélectionnée. Ceci est réalisé en calculant le vecteur W qui contient toutes les sommes possibles pour un sous-ensemble de tailles de flux. Nous calculons le vecteur W en j étapes successives, pour chaque élément de l'ensemble de flux. Prenons l'exemple de l'ensemble de tailles de flux (en Mo) $\mathcal{S} = \{2, 3, 4, 1\}$. Dans cet exemple, nous cherchons tous les sous-ensembles de $\mathcal{S}$ où la somme est égale ou inférieure à 7. Le tableau 3.2 représente un exemple d'un vecteur $work$ pour l'ensemble $\mathcal{S} = \{2, 3, 4, 1\}$. Notons que les index du tableau correspondent aux sommes des éléments de $\mathcal{S}$.

Tableau 3.2 Exemple de calcul du vecteur W (Neal, 2012)

somme	0	1	2	3	4	5	6	7	8	9	10
init	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
$\mathcal{S} = \{2\}$	0	-1	2	-1	-1	-1	-1	-1	-1	-1	-1
$\mathcal{S} = \{2, 3\}$	0	-1	2	3	-1	3	-1	-1	-1	-1	-1
$\mathcal{S} = \{2, 3, 4\}$	0	-1	2	3	4	3	4	4	-1	4	-1
$\mathcal{S} = \{2, 3, 4, 1\}$	0	1	2	3	4	3	4	4	1	4	1

Nous construisons le vecteur W en j étapes successives. Nous notons les index de l'ensemble $\mathcal{S}$ par k. Ci-dessous nous détaillons les conditions nécessaires pour calculer chaque élément du vecteur W (Neal, 2012) :

- $W[k] \neq -1$: Sinon une somme k n'est pas possible. Autrement dit, si $W[k] \neq -1$, c'est à dire que nous pouvons avoir une somme k à partir de l'ensemble donné (dans notre exemple c'est $\mathcal{S}$)
- $W[k] \neq s$: avec s est un élément de $\mathcal{S}$. Cette condition garantit que nous utilisons chaque élément de l'ensemble donné au plus une fois.
- $W[k+s] = -1$: Cette condition nous permet d'éviter d'ignorer une somme trouvée par les itérations précédentes.

Nous introduisons ces trois conditions à la ligne 13 de l'algorithme 3.2 de la programmation dynamique. Nous ajoutons la contrainte $k + s \leq R$ pour limiter les sommes à chercher à la capacité de l'interface.

Algorithme 3.2 Programmation dynamique (Neal, 2012)

```
1  Input : I, F, R
2  Output : W.
3  for k ← 0 to R do
4  |  W[k] ← −1
5  end
6  W[0] ← 0
7  for j ← 0 to R do
8  |  W[k] ← −1
9  end
10 for j ← 0 to size(F) do
11 |  s ← S_j
12 |  for k ← 0 to R do
13 |  |  if W[k] ≠ −1 and W[k] ≠ s and W[k + s] = −1 and k + s ≤ R then
14 |  |  |  W[k + s] ← s
15 |  |  end
16 |  end
17 end
18 return W
```

Dans ce qui suit, nous détaillons l'algorithme D-OFAP (voir algorithme 3.3). Cet algorithme prend en entrée l'ensemble des appareils $\mathcal{I}$, l'ensemble des flux $\mathcal{F}$, l'ensemble des interfaces

Algorithme 3.3 D-OFAP (Amor *et al.*, 2019)

```
1  Input : I, F, J, R_j, D_i, C_i, E_j
2  Output : Assignment matrix X.
3  List J ← Sort(J)
4  j ← 0
5  X ← 0
6  while j ≤ |J| and F ≠ 0 do
7      W ← DP(S_i, R_j, F)
8      assigned ← False
9      i ← size(W)
10     while (i ≠ 0) and (assigned = False) do
11         if (W[i] ≠ −1) and (W[i] ≠ 0) then
12             subset ← extract_subset(W[i])
13             apply the defined constraints on the selected subset.
14             if subset ≠ 0 then
15                 F ← F \ subset
16                 assigned ← True
17                 update the assignment matrix X based on the new subset.
18             end
19             else
20                 i ← i − 1
21             end
22         end
23     end
24     j ← j + 1
25 end
26 return X
```

réseau $\mathcal{J}$ et les caractéristiques des flux et des interfaces. La sortie de l'algorithme est une matrice binaire X, avec $|\mathcal{I}|$ lignes et $|\mathcal{J}|$ colonnes, qui représente l'assignation des flux aux interfaces disponibles.

Cet algorithme trie en premier les interfaces en fonction de leurs capacités réseau R_j. Ensuite, pour chaque interface, il calcule le vecteur W en utilisant la programmation dynamique (ligne 7). Ensuite, l'algorithme sélectionne un sous-ensemble d'appareils ayant la plus grande somme de tailles de flux, à partir de W (ligne 12). Tous les flux des appareils dans le sous-ensemble sélectionné doivent respecter les délais et les contraintes de capacités (ligne 13). S'il existe un

sous-ensemble valide pour l'interface j, le sous-ensemble généré sera soustrait de l'ensemble $\mathcal{F}$ (ligne 15). Ensuite, la matrice d'assignation sera mise à jour et le sous-ensemble de flux est attribué à l'interface qui est en train d'être traitée. Cet algorithme s'arrête lorsque toutes les interfaces sont traitées ou lorsque tous les flux sont affectés.

3.4.3 RAND-INIT-ALLOCATION

Pour évaluer la performance de nos solutions, nous allons comparer nos résultats avec celles du *Baseline* qui est un algorithme proposé par (Angelakis *et al.*, 2016) et qui est décrit par l'algorithme 3.4.

Cet algorithme est basé sur les priorités affectées aux interfaces et aux flux pour décider lesquels devront être affectés et pour quelles interfaces. L'idée principale de cet algorithme est de servir en premier les flux qui consomment le plus de ressources réseau. Dans notre cas, ce sont les flux qui ont les tailles les plus importantes. Puis, l'assignation des flux aux interfaces se fait par ordre des interfaces qui ont le plus faible coût d'utilisation. L'algorithme traite ensuite les flux avec des demandes en ressources inférieurs, etc.

La ligne 7 de l'algorithme (voir algorithme 3.4) calcule la valeur normalisée des tailles des flux. Ceci est fait en divisant la taille de chaque flux S_i par la valeur maximale de tous les flux, $maxS$. Le résultat est un vecteur de valeurs normalisées S'. Cette procédure de normalisation est aussi appliquée pour calculer les valeurs normalisées de la consommation énergétique. En conséquence, nous aurons $0 \leq S'_i \leq 1,\ \forall i \in \mathcal{I}$ (de même pour E'). La procédure RANDOM-INIT-EQUAL-SHARES aux lignes 12-13 prend en entrée les valeurs normalisées des vecteurs $S'etE'$ et identifie quels flux et interfaces qui ont des valeurs normalisées égales. Pour ces flux et interfaces, la procédure choisit au hasard l'ordre dans lequel le flux/interface sera servi/traité. Le résultat est un vecteur avec l'ordre dans lequel les flux seront servis. Après l'obtention d'un vecteur ordonné pour les flux et les interfaces, l'algorithme peut maintenant itérer sur ces vecteurs et affecter chaque flux à l'interface appropriée.

Le reste de l'algorithme (lignes 15-32) itère le vecteur d en commençant par le premier flux dans la liste et l'affecter aux ressources des interfaces disponibles. Initialement, nous définissons le vecteur $totalCost$ qui est constitué de $\mathcal{J}$ éléments nuls. Ce vecteur permet de stocker l'utilisation de chaque interface à chaque itération. Plus tard, l'élément $totalCost_j$ ($j^{ème}$ colonne de totalCost) sera mis à jour si le flux f_i de taille S_i est affecté à l'interface j.

Tout d'abord, l'algorithme identifie le flux f_i qui correspond à l'élément d_s (ligne 16). Ensuite, il tente d'affecter f_i à l'interface selon l'ordre défini dans les vecteurs. Si les contraintes décrites aux lignes 22 et 24 sont satisfaites, l'attribution se fait par la procédure *ASSIGN*. Puis, la matrice d'attribution X et la variable de consommation des ressources de l'interface sont mise à jour. Cet algorithme s'arrête lorsque tous les flux et les interfaces sont traitées ou lorsque tous les flux sont affectés.

3.5 Conclusion

Dans ce chapitre, nous avons présenté notre modèle mathématique visant à optimiser l'utilisation des interfaces disponibles dans une passerelle IdO. Notre objectif est la maximisation de la quantité de données acceptées par la passerelle en satisfaisant la qualité de service des applications. Nous avons proposé deux algorithmes pour approcher la solution optimale. Le premier repose sur l'approche Greedy qui permet de résoudre le problème du Knapsack. Le deuxième algorithme se base sur la programmation dynamique. Dans le dernier chapitre, nous allons implémenter les algorithmes mentionnés et évaluer les performances de notre travail en utilisant des paramètres spécifiques.

Algorithme 3.4 *Baseline* (Angelakis *et al.*, 2016)

Input : $\mathcal{I}$, $\mathcal{F}$, $\mathcal{J}$, R_j, C_i, E_j.
Output : *Assignment matrix* X.
$X = 0_{|\mathcal{I}|x|\mathcal{J}|}$
$maxS = \max_{i \in \mathcal{I}} S_i$
$maxE = \max_{j \in \mathcal{J}} E_j$
for *i=0 ...* $|\mathcal{I}|$ **do**
    $S'_i = \frac{S_i}{maxS}$
end
for *j=0 ...* $|\mathcal{J}|$ **do**
    $E'_j = \frac{E_j}{maxE}$
end
$d = RANDOM\_INIT\_EQUAL\_SHARES(S'_i)$
$c = RANDOM\_INIT\_EQUAL\_SHARES(E'_j)$
$totalCost = 0_{|\mathcal{J}|}$
for $s = 1...|\mathcal{I}|$ **do**
    Let f_i be the flow that corresponds to d_s.
    $assigned = False$
    $a = 0$
    while $(a \leq |\mathcal{J}|)$ *and* $(assigned == False)$ **do**
        Let j be the index of the interface that corresponds to c_a.
        $energy = CAL\_ENERGY(E_j, S_i)$
        if $(energy \leq C_j) and (totalCost_j + S_i \leq R_j)$ **then**
            $delay = CAL\_DELAY(f_i)$
            if $(delay \leq D_i)$ **then**
                $[X, totalCost_j] = ASSIGN(f_i, j)$
                $assigned = True$
            end
        end
        $a = a + 1$
    end
end
return X

CHAPITRE 4

PROTOCOLE EXPÉRIMENTAL ET VALIDATION

4.1 Introduction

Dans ce chapitre, nous présentons l'implémentation des algorithmes proposés et présentés aux sections 3.4.1 et 3.4.2. Nous développons des simulations pour étudier les performances des deux algorithmes *G-OFSP* et *D-OFSP*. Nous présentons en premier l'implémentation du système conçu dans un environnement réel. Ensuite, nous donnons des détails sur la conception de nos simulations. Nous présentons les résultats obtenus et nous comparons nos solutions avec la solution *Baseline*. Nous les comparons également avec la solution optimale obtenue en solvant le problème d'optimisation présenté à la section 3.3 à l'aide d'un *CPLEX solver*. La solution *Baseline* est présentée au chapitre de la méthodologie et est basée sur l'attribution de priorité aux flux et aux interfaces pour décider quel flux doit être traité en premier. Enfin, en nous basant sur les résultats obtenus au cours du processus de validation, nous soulignons les avantages et les limites de notre solution.

4.2 Architecture du testbed

Dans cette section, nous présentons les détails d'implémentation de la solution proposée dans le chapitre précédent. Nous introduisons en premier l'architecture de notre système. Puis, nous présentons le scénario que nous avons effectué pour tester le protocole et le module de contrôle de la passerelle.

4.2.1 Composantes de l'architecture

Pour simuler un scénario réel d'un environnement IdO hétérogène, nous avons utilisé des différents types d'appareils IdO qui sont équipés d'au moins une interface.

- **Passerelle** : Pour simuler la passerelle, nous avons utilisé un Raspberry Pi 3 qui est un micro-ordinateur à processeur ARM et qui possède des entrées-sorties pour y ajouter d'autres

périphériques, interfaces réseau, capteurs, etc (Pi, 2019). Ce modèle a suffisamment de ressources réseau et de traitement pour pouvoir exécuter les tâches associées à une passerelle comme le transfert des paquets, routage et gestion de l'interopérabilité entre les différents protocoles. Notre passerelle est équipée de deux interfaces réseau : WiFi 802.11n, Bluetooth 3.0 et une interface Ethernet pour pouvoir se connecter à Internet. Dans cette passerelle, nous avons implémenté le module de contrôle présenté à la section 3.2.1. Pour le module d'optimisation, qui fait partie du module de contrôle, nous avons utilisé l'algorithme D-OFAP pour la prise d'une décision optimale. Nous avons également implémenté l'entité principale de notre protocole de *handover* qui est conçu à la section 3.2.2.

- **Appareil IdO type 1** : Notre premier appareil IdO consiste à une carte Raspberry Pi Modèle zéro qui est moins puissant en la comparant au Raspberry Pi modèle 3 en termes de capacité du processeur, de la mémoire vive, nombre de GPIO (ports entrée-sortie), etc. Cet appareil dispose de deux interfaces réseau : Bluetooth et WiFi.

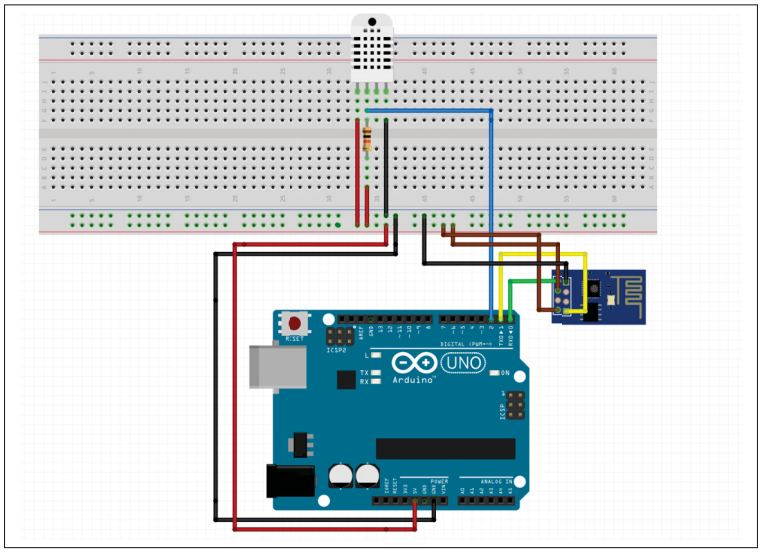

Figure 4.1 Appareil IdO type 2 (fritzing, 2019)

- **Appareil IdO type 2** : Cet appareil est un Arduino Uno qui est un microcontrôleur mono-puce et qui est équipé de 14 broches d'entrée-sortie (Arduino, 2019). Nous avons connecté à cette carte un module WiFi ESP8266. L'ESP8266 est un circuit intégrant un microcontrô-

leur et un module WiFi développé par le fabricant Espressif. Ce circuit intégré permet de connecter un microcontrôleur à un réseau pouvant de ce fait, diffuser du contenu sur internet et communiquer avec une passerelle. Nous avons également connecté à cet appareil un capteur de température qui permet de mesurer la température ambiante et l'envoyer périodiquement à la passerelle. Le module WiFi et le capteur sont connectés à la carte Arduino comme décrit à la figure 4.1.

4.2.2 Scénario et topologie

Pour tester notre solution, nous avons conçu l'architecture décrite à la figure 4.2. Nous avons connecté deux appareils de type 1 et 2 à la passerelle. L'appareil IdO 1 envoie des fichiers JSON qui contiennent des données générées aléatoirement. Tandis que l'appareil IdO 2 envoie les données du capteur de température. Les données sont envoyées sous format JSON en utilisant le protocole MQTT pour WiFi et RFCOMM pour Bluetooth. Au début du scénario, nous avons connecté l'appareil IdO de type 1 à la passerelle en utilisant l'interface WiFi. Puis, nous avons connecté l'appareil IdO de type 2 (Arduino) qui envoie des données critiques avec des courtes échéances aussi en utilisant WiFi. Ce nouvel équipement, qui a rejoint le réseau, a déclenché le processus d'assignation de flux qui est exécuté par le module de contrôle de la passerelle. Le résultat de l'exécution de l'algorithme D-OFAP est la décision qui contient l'assignation des appareils aux interfaces de la passerelle. Dans notre cas, la passerelle a envoyé une commande *Switch-to-BLE* à l'appareil IdO 1 pour lui indiquer de changer son point de connexion vers Bluetooth. Ceci est dû aux valeurs d'échéances courtes des flux de l'appareil IdO type 2, qui ne seraient pas satisfaites si les équipements IdO 1 et 2 envoient leurs données en utilisant la même interface. La figure 4.2 montre des exemples de deux types de notification qui sont :

- *DISCOVER* : Comme décrit à la section 3.2.2, cette notification contient des informations sur l'appareil IdO (nombre d'interfaces et emplacement), les flux (débit binaire d'envoi et échéances), etc. Ces informations vont être utilisées ensuite par le module d'optimisation.
- *SWITCH-TO-BLE* : Cette notification sert à déclencher le processus de *handover* vers l'interface Bluetooth.

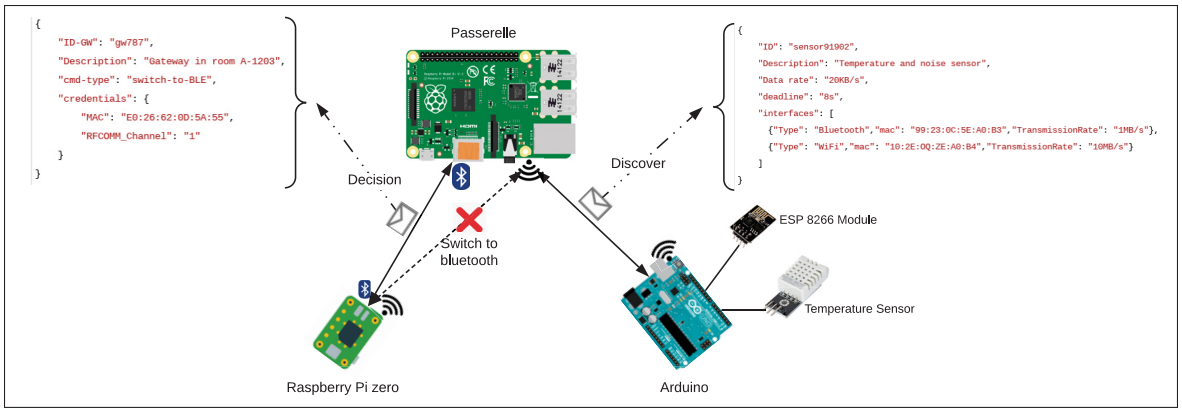

Figure 4.2 Architecture proposée

4.2.3 Implémentations et tests

Dans cette section, nous présentons les détails de l'implémentation du protocole de *handover* que nous l'avons développé et testé au niveau d'un Raspberry Pi 3 (la passerelle) et un- Raspberry Pi zero (l'appareil IdO) avec deux interfaces Bluetooth et WiFi. La figure 4.3 montre un aperçu général des modules développés pour implémenter le protocole de *handover*. Dans ce scénario, nous avons utilisé D-OFAP qui est implémenté au niveau du module d'optimisation pour la prise de décision d'assignation de flux. Le résultat de l'algorithme peut être soi déclencher le processus de *handover* pour changer d'interface, soi garder la connexion avec l'interface courante. Dans notre implémentation, nous avons développé nos modules en utilisant des scripts Bash et Python.

Comme décrit à la figure 4.3, deux scripts bash sont responsables au déclenchement du processus de *handover*. Dans cette section, nous prenons l'exemple de changement du point de connexion vers WiFi. Ce processus est déclenché au niveau de la passerelle en exécutant le script *switch-to-wifi.sh* suite à une décision de l'algorithme D-OFAP. Ce script va créer deux processus qui tournent en background. Le premier est *notif-ble-send.sh* qui permet d'envoyer la notification *SWITCH-WIFI* à l'appareil IdO en utilisant l'interface courante (Bluetooth) et le deuxième script permet de créer un processus qui sera à l'écoute des notifications ou données reçus de l'appareil IdO. Le processus d'écoute lancera le processus d'envoi des notifica-

tions pour l'interface WiFi une fois que la notification *HELLO* est reçue. Enfin, la notification *HELLO-ACK* est envoyée par *notif-wifi-send.sh* et la passerelle commence à recevoir les données de l'appareil IdO.

De même, nous avons développé des scripts Bash et Python au niveau de l'appareil IdO (Raspberry Pi Zero) pour pouvoir recevoir et répondre aux demandes de la passerelle. Les scripts *notif-ble-listener.sh* et *notif-wifi-listener.sh* permettent de recevoir les notifications de la passerelle par le biais des interfaces Bluetooth et WiFi respectivement. Les scripts *notif-wifi-send.sh*, *notif-ble-send.sh send_data.sh* permettent d'envoyer les notifications et les données à la passerelle.

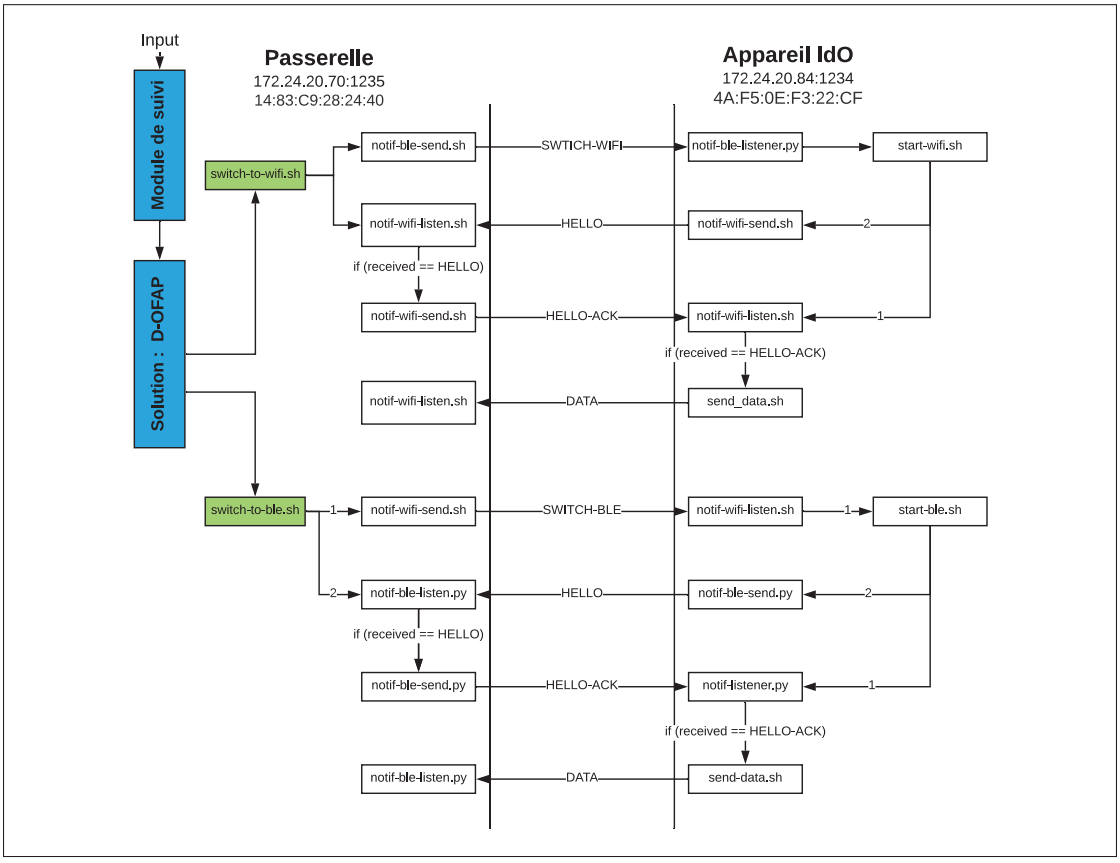

Figure 4.3 Implémentation du protocole de *handover*

4.2.4 Délais de *handover*

Ce délai inclut le délai d'établissement de connexion en utilisant le protocole proposé ainsi que le temps d'activation d'interfaces. La figure 4.4 montre les délais mis pour établir la connexion et envoyer les données. Le temps mis pour établir la connexion WiFi est au moyenne 0.9s. Par contre, pour l'interface Bluetooth, ce délai atteint 6s à cause du processus du couplage (*Pairing*) de Bluetooth. De plus, la figure 4.4 montre que notre implémentation garantit que l'envoi des données ne peut pas être interrompu que lorsque la connexion avec la nouvelle interface serait établie. Cela permettra évidemment de diminuer la quantité des données perdue lors du processus de *handover*.

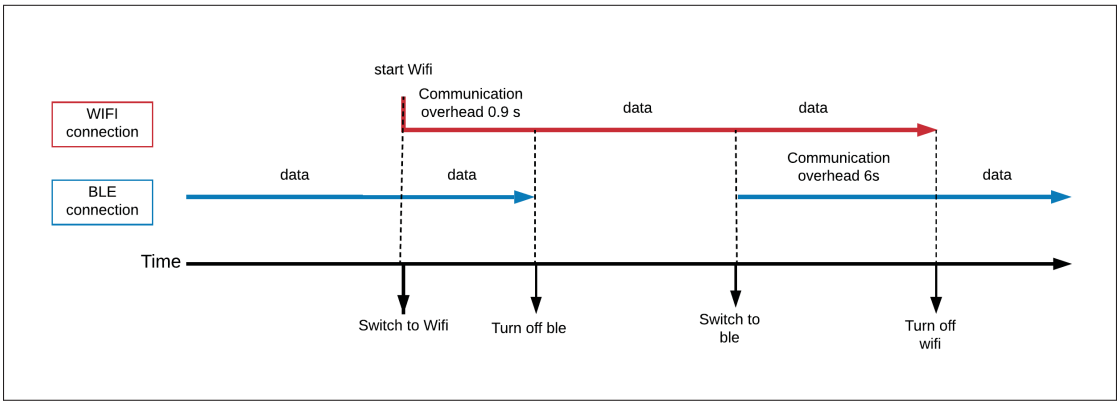

Figure 4.4 Délais de *handover*

4.3 Protocole de validation

Nos algorithmes sont déployés dans le module d'optimisation qui fait partie du module de contrôle de la passerelle décrit à la section 3.2.1.2. Nous avons effectué et développé nos simulations en utilisant Python. Afin de valider *G-OFAP* et *D-OFAP*, nous avons développé la solution *Baseline* et la solution optimale en utilisant *ILOG CPLEX IBM*. Pour tester la performance de nos algorithmes, nous avons effectué différents scénarios en les comparant avec la solution optimale et le *Baseline*. Le résultat de notre solution est une matrice qui indique

l'attribution des flux aux interfaces et qui satisfait les échéances et les demandes des flux en termes de bande passante.

Notre protocole de validation est présenté à la figure 4.5 et englobe les étapes suivantes :

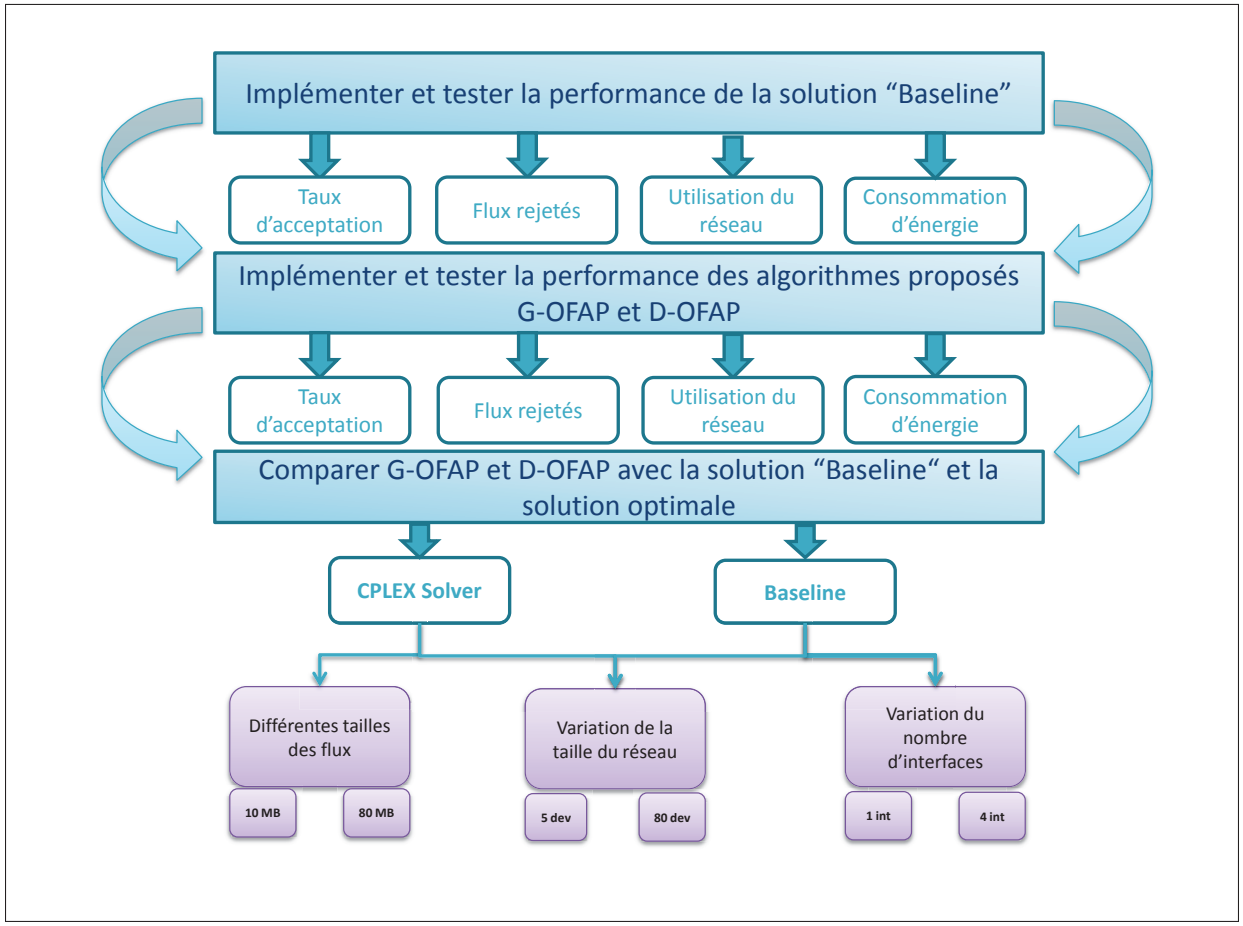

Figure 4.5 Protocole de validation

- Implémenter et tester la performance de la solution *baseline* en fonction de la quantité des flux acceptés et rejetés et la consommation d'énergie après l'assignation des flux aux interfaces.
- Implémenter et tester la performance des algorithmes *G-OFAP* et *D-OFAP* en fonction des mêmes critères utilisés pour l'évaluation de performance à l'étape précédente.

- Comparer les performances en termes de la consommation d'énergie et de la maximisation de la quantité de données acceptées par notre algorithme avec les performances de la solution optimale et le *Baseline*.
- Effectuer des différents scénarios en variant la quantité de données générées, le nombre d'interfaces et le nombre d'appareils connectés à la passerelle en mesurant le taux d'acceptation et la consommation d'énergie totale.

Dans ce travail, nous considérons que les équipements IdO possèdent le même nombre d'interfaces que la passerelle. Nous considérons seulement le délai de transmission, nous supposons que les objets connectés sont placés à proximité de la passerelle de telle sorte que le délai de propagation est minime. Nous nous concentrons sur l'optimisation de l'assignation des flux aux interfaces par la maximisation de la quantité de données acceptées par la passerelle.

- Détails des simulations :

Dans nos simulations, nous avons considéré une architecture composée d'une passerelle et nous faisons varier le nombre d'équipements IdO se connectant à la passerelle. En ce qui concerne les données, chacun de ces équipements IdO envoie des données discrètes. Nous générons l'instance du problème avec des tailles de flux générés aléatoirement et ayant une somme fixe donnée en entrée. Les appareils IdO sont équipés d'interfaces multiples. Nous considérons quatre types de technologies de communication : WiFi, Bluetooth, Zigbee et Z-Wave, avec des débits binaires respectifs de 5 Mbps, 2 Mbps, 240 Kbps et 100 Kbps. Dans nos simulations, nous considérons deux types d'applications de ville intelligente IdO (Zanella *et al.*, 2014) : (a) les applications à faible débit de données pouvant tolérer des délais jusqu'à 2 minutes (par exemple, Smart Parking, surveillance de la qualité de l'air) et (b) des applications critiques pouvant tolérer des délais allant jusqu'à 20 secondes (par exemple, des alarmes pour des situations d'urgence, des applications de soins de santé) (Amor *et al.*, 2019).

Afin d'étudier les performances de *G-OFSP* et *D-OFSP*, nous comparons leurs résultats à ceux des résultats optimaux obtenus en solvant le problème d'optimisation formulé et aux résultats de la solution RAND-INIT-ALLOCATION présentée à la section 3.4.3. Nous avons résolu

notre problème d'optimisation d'assignation de flux en utilisant ILOG CPLEX (IBM, 2019). Pour un nombre et de tailles de flux raisonnables, le solveur offre des solutions optimales. Cependant, le temps de résolution augmente lorsque nous traitons une architecture IdO plus étendue. Une série de simulation a été menée pour calculer le taux d'acceptation dans différents scénarios. Ces derniers ont simulé les flux provenant des équipements IdO en faisant varier la quantité de données générées, le nombre d'interfaces par équipement et la taille du réseau. Les résultats des scénarios sont abordés dans la section suivante, où nous avons étudié l'impact de divers paramètres du réseau sur les solutions proposées (Amor *et al.*, 2019).

4.4 Résultats expérimentaux

Ces scénarios ont simulé les flux provenant d'équipements IdO en faisant varier les paramètres suivants :

4.4.1 Quantité de données générées

Dans ce scénario, le nombre de flux générés est fixé à 80 et le nombre d'interfaces par équipement est fixé à 3. Nous faisons varier la quantité totale de données générées. Pour chaque ensemble de flux généré, nous calculons le taux d'acceptation des données valides, qui est égal à la quantité totale de données acceptées divisée par la quantité totale de données générées.

La figure 4.6 représente les résultats des simulations du scénario 1 effectuées pour analyser les performances des solutions proposées. Cette figure indique le taux d'acceptation des données valides en modifiant la quantité des données provenant des équipements IdO. Au début, lorsque la quantité des données générées est comprise entre 10 et 15 Mo, le taux d'acceptation des algorithmes G-OFAP et D-OFAP varie entre 0,9 et 1, ce qui est comparable à la solution optimale. Cependant, lorsque la charge du réseau augmente, nous observons que D-OFAP a de meilleures valeurs de taux d'acceptation que l'algorithme Greedy et le Baseline. Par exemple, lorsque les équipements IdO génèrent des données dont la somme est égale à 25 Mo, D-OFAP atteint un taux d'acceptation de 0,69, ce qui est plus proche de la valeur de la solution optimale

et supérieur à la valeur de la solution *Baseline*. La baisse des valeurs du taux d'acceptation s'explique par le fait que la taille moyenne des flux augmente. Pour cette raison, il n'est pas possible de respecter les échéances prescrites pour tous les flux. Par conséquent, la passerelle serait obligée de rejeter les flux non valides où les échéances ne peuvent pas être respectées.

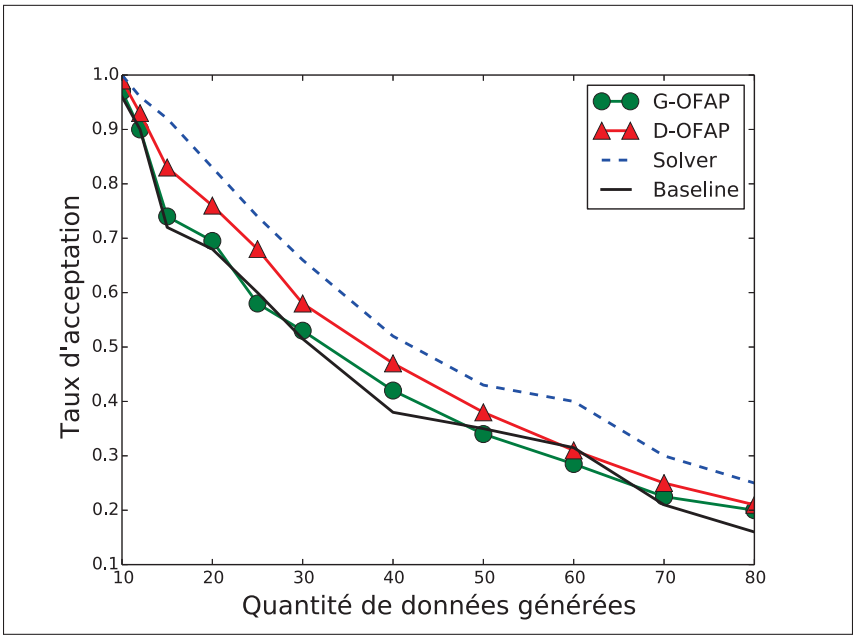

Figure 4.6 Effets de la variation de la quantité de données générées sur les valeurs du taux d'acceptation.

4.4.2 Nombre d'interfaces

Dans ce scénario, nous avons fixé la somme des flux à 75 Mo et le nombre de flux générés à 80. Nous effectuons nos simulations avec différents types interfaces et nous faisons varier le nombre d'interfaces par équipement.

La figure 4.7 représente le taux d'acceptation en variant le nombre d'interfaces. Nous pouvons observer que D-OFAP garantit une solution optimale dans le cas où la passerelle et les appareils IdO sont équipés d'une seule interface. En effet, notre solution D-OFAP qui est basé sur la programmation dynamique, donne une solution exacte au problème du Knapsack. Le scénario

où la passerelle est équipée d'une seule interface est une instance du problème du Knapsack où la capacité du Knapsack est la capacité de l'interface et les poids des objets à mettre dans le sac sont les tailles des flux. C'est pour cela que D-OFAP donne une solution optimale dans ce cas. Lorsque nous augmentons le nombre d'interface, nous pouvons constater que le taux d'acceptation augmente, car la capacité de gérer et accepter plus de trafic augmente également. La différence entre la solution optimale et les valeurs de D-OFAP est au moyenne 7%. Cependant, la solution optimale est coûteuse en terme du temps d'exécution. Le temps écoulé pour résoudre le problème à l'aide du solveur CPLEX est beaucoup plus long que les algorithmes D-OFAP et G-OFAP. Par exemple, pour l'instance du problème avec 25 équipements générant 15 Mo de données et avec une passerelle équipée de 5 interfaces, il faut 11 heures et 34 minutes au solveur pour générer la solution optimale, 8 secondes pour l'algorithme D-OFAP et 25 ms pour la solution Greedy. Nous pouvons également observer que la solution *Baseline* a relativement les même valeurs de taux d'acceptation que G-OFAP car ces deux algorithmes applique la même politique d'assignation.

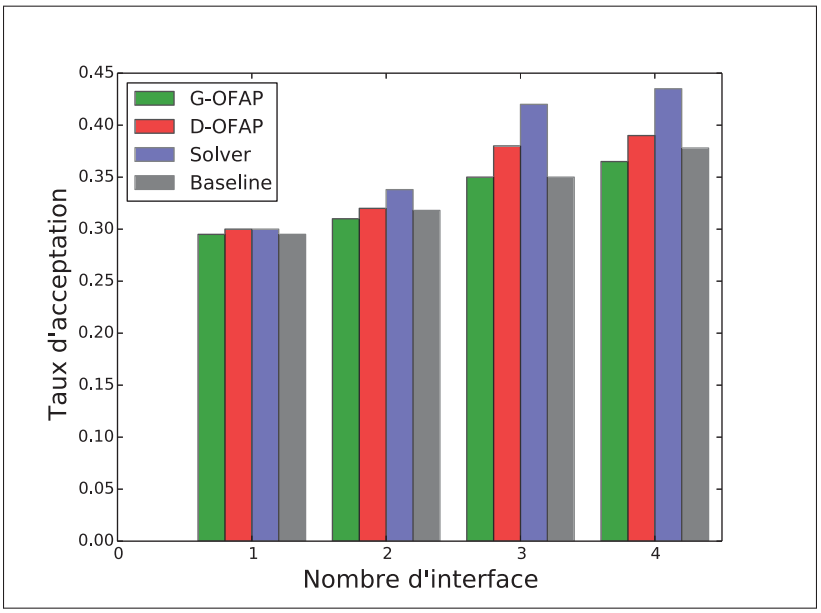

Figure 4.7 Effets de la variation du nombre d'interfaces sur les valeurs du taux d'acceptation.

4.4.3 Taille du réseau

Nous étudions ici l'impact de la variation de la taille du réseau. Dans cette section, la taille du réseau est définie en termes de nombre d'équipements qui se connectent à la passerelle et qui envoient des flux au cours d'un créneau horaire spécifié. Le nombre d'interfaces par équipement est fixé à 3 et les équipements IdO génèrent des flux dont la somme est égale à 40 Mo.

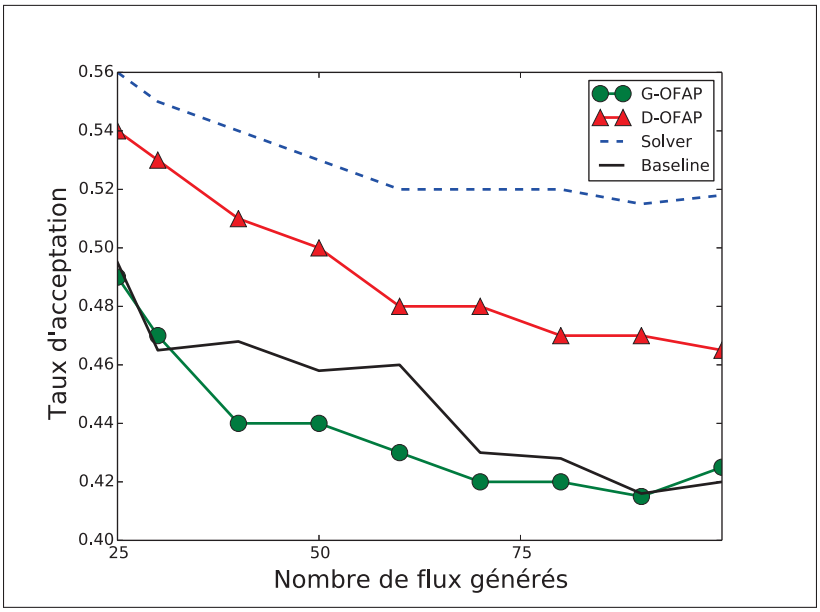

Figure 4.8 Effets de la variation de la taille du réseau sur les valeurs du taux d'acceptation.

La figure 4.8 montre que D-OFAP est plus performant que l'approche *Greedy* et la solution *Baseline*. Notons que le taux d'acceptation de données valides diminue lorsque nous augmentons le nombre de flux. En effet, de plus en plus de flux allouent les ressources réseau de l'interface et donc aucune ressource disponible n'est suffisante pour les flux qui ne sont pas encore traités. Cependant, nous pouvons constater que la variation du nombre de flux n'a pas une grande influence sur les valeurs du taux d'acceptation qui ne varie pas largement lorsque le nombre de flux augmente. Cela peut être expliqué par le fait que la quantité de données générées est fixe.

4.4.4 Utilisation des réseaux

Pour déterminer comment les charges sont distribuées entre les différents réseaux en exécutant nos solutions, nous avons effectué une comparaison de la distribution du trafic entre les différentes interfaces. Dans ce scénario, nous considérons trois types d'interfaces réseau par objet connecté qui sont WiFi, Bluetooth et Zigbee. Dans ce scénario, nous générons une quantité de données inférieure à la capacité des interfaces réseau. Concernant la configuration de l'architecture, nous avons généré les données avec 50 objets qui sont connectés à la passerelle. Nous avons calculé le pourcentage de l'utilisation de chaque interface en divisant la quantité des données acceptée par la capacité de l'interface.

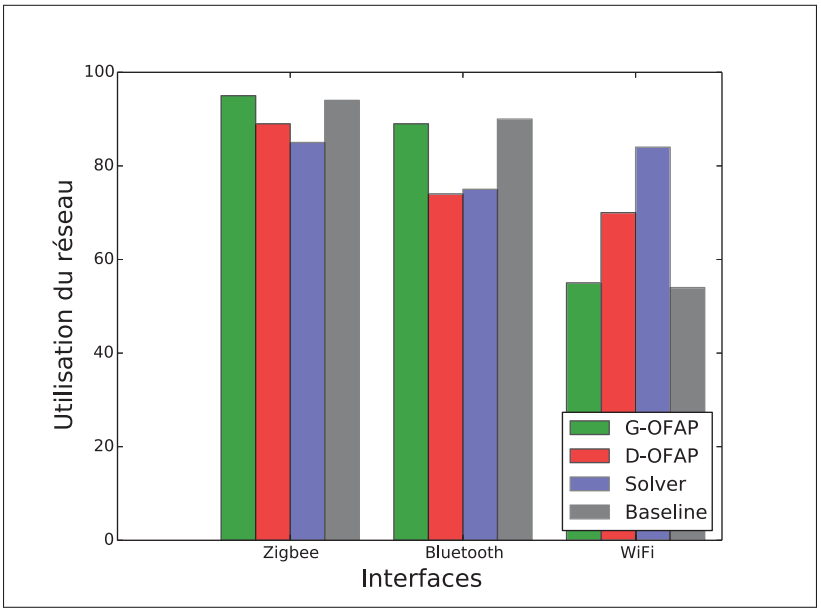

Figure 4.9 Utilisation des interfaces réseau.

La figure 4.9 présente la distribution de la charge du trafic sur les trois interfaces utilisées, WiFi, Bluetooth et Zigbee. Les résultats montrent que le solveur tend à saturer les interfaces avec les plus grandes capacités comme WiFi plutôt que Bluetooth et Zigbee. Nous constatons que le solveur favorise, dans ce scénario, l'assignation des flux à WiFi pour maximiser la quantité de données acceptées par la passerelle. En effet, le solveur ne donne pas une importance à

la priorité des interfaces avant d'affecter les flux. Cela est dû à la suffisance de la capacité des batteries qui permettrait l'envoi des flux avec les trois interfaces disponibles aux objets connectés. Donc, si la contrainte énergétique définie à l'équation 3.2 est satisfaite, la sélection des flux qui vont être affectés aux interfaces et leurs distributions entre les interfaces dépendent seulement de la quantité de données à maximiser.

Par contre, pour les algorithmes proposés D-OFAP et G-OFAP, les interfaces avec les plus faibles capacités vont être traitées en premier. En d'autres termes, la liste des interfaces est triée dans l'ordre ascendant selon la capacité de chaque interface. Ensuite, l'algorithme applique sa politique pour affecter les flux. G-OFAP commence par affecter les flux avec les tailles les plus faibles et D-OFAP applique l'algorithme de programmation dynamique pour déterminer la liste valide des flux choisis pour être envoyés à travers l'interface réseau choisie. Et donc, ces deux algorithmes tendent à maximiser la quantité des données affectée aux interfaces avec la capacité la plus faible comme le présente la figure 4.9. Ces résultats montrent que ces deux algorithmes affectent plus de données à l'interface Zigbee et Bluetooth puisque ces derniers sont les interfaces avec des capacités les plus faibles.

De même, la solution *Baseline* favorise la maximisation des flux affectés aux interfaces Zigbee et Bluetooth plutôt que WiFi. Ainsi, la solution *Baseline* tend plus à surcharger les interfaces avec les plus faibles coûts (en termes d'énergie) en premier. Ceci est dû à la fonction qui permet le tri des interfaces suivant leurs consommations énergétiques.

4.4.5 Quantité des flux rejetés

Le taux des flux rejetés est un autre aspect que nous analysons pour illustrer l'efficacité de nos solutions. Pour ce faire, nous considérons une architecture composée de 40 équipements connectés à une passerelle multi-interface. Tous ces équipements, incluant la passerelle, sont équipés de deux interfaces à savoir Bluetooth et WiFi. Nous varions la quantité des données générée et nous mesurons la quantité des flux rejetés. Nous rappelons que le rejet des flux survient lorsque soi le flux a dépassé son échéance, soi l'interface est surchargée ou soi parce

que l'énergie disponible à l'équipement n'est pas suffisante pour envoyer le flux sur l'une de ses interfaces disponibles.

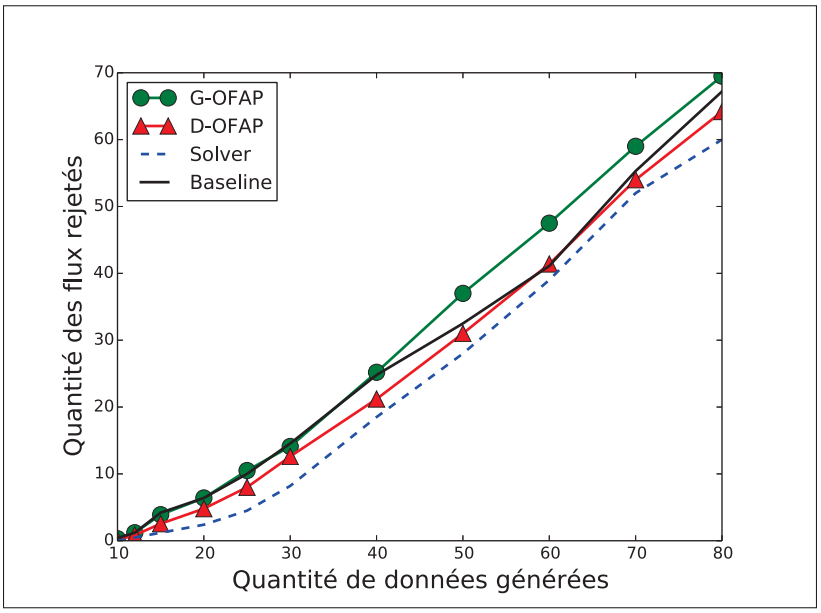

Figure 4.10 La quantité des flux rejetés.

La figure 4.10 illustre la quantité des flux rejetés en variant la quantité des flux générés. Nous remarquons que G-OFAP atteint des valeurs supérieures à la solution optimale et D-OFAP car cet algorithme commence par l'assignation des flux de tailles basses aux interfaces. En conséquence, les interfaces sont rendues surchargées par un grand nombre de flux de faible quantité de données. C'est pour cela que le nombre des flux rejetés pour G-OFAP est maximal. En augmentant la quantité de données générées, le nombre de flux rejetés augmente considérablement, car la taille des flux est rendue grande par rapport à la capacité des interfaces. Cependant pour la solution optimale, le choix des flux est basé sur la maximisation des flux acceptés par la passerelle.

4.4.6 Consommation d'énergie

Dans cette partie, nous étudions la consommation d'énergie lors de la transmission des flux et nous comparons les résultats des solutions suggérées. Nous générons 60 flux et on varie la

somme de leurs tailles de 10 Mo à 80 Mo. Chaque appareil est équipé de 3 interfaces et envoie ses flux à une passerelle multi-interface.

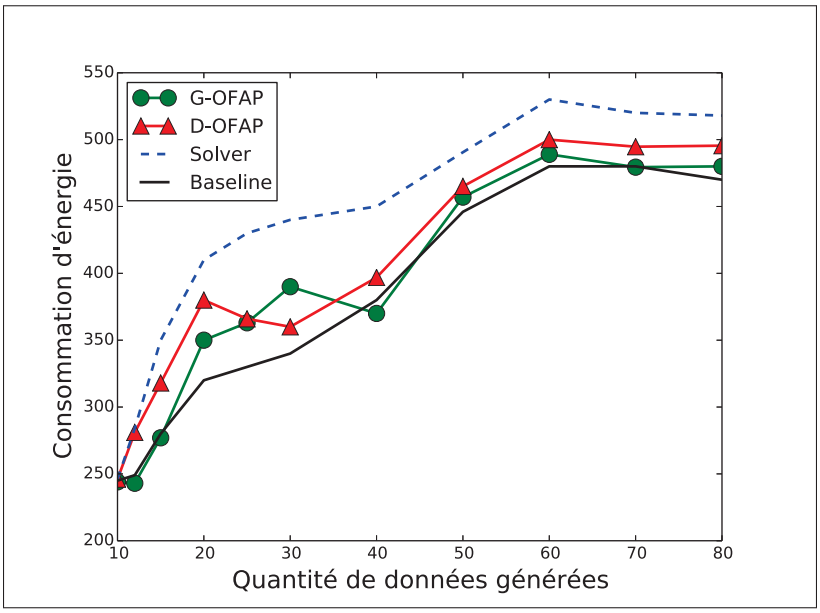

Figure 4.11 Consommation d'énergie.

La figure 4.11 illustre la consommation d'énergie de D-OFAP, G-OFAP et le *Baseline* lors de l'assignation des flux. Nous remarquons que la consommation d'énergie augmente avec la quantité des données générées. Nous constatons que les valeurs du D-OFAP sont relativement supérieures aux celles du *Baseline* et G-OFAP. Ceci est expliqué par le fait que D-OFAP a un taux d'acceptation supérieur au *Baseline* et G-OFAP. En effet, la solution *Baseline* donne une priorité aux interfaces qui économisent l'énergie des appareils connectés. Cet algorithme, présenté précédemment à la section 3.4.3, utilise une fonction pour trier les interfaces dans l'ordre croissant selon leurs consommations énergétiques. La solution *Baseline* utilise également une deuxième fonction qui a pour objectif de trier les flux selon leurs caractéristiques et de déterminer quel flux à traiter en premier. C'est ce qui explique les valeurs faibles de la consommation d'énergie de la solution *Baseline* (Figure 4.11) par rapport à G-OFAP et D-OFAP. Cependant, tandis qu'il ne tient pas en compte de la consommation d'énergie lors de l'assignation des flux, l'algorithme G-OFAP trie les interfaces selon leurs capacités réseau avant l'assignation des

flux. Et donc, dans notre scénario, ça va mener à donner la priorité aux réseaux qui ont une faible consommation d'énergie par rapport à WiFi comme Zigbee et Bluetooth.

4.5 Discussion

Nous discutons ci-après les résultats des simulations. Dans la section précédente, nous avons introduit notre protocole de validation qui inclut les simulations effectuées pour évaluer les performances des algorithmes proposées G-OFAP et D-OFAP en les comparant avec la solution optimale et la solution *Baseline* présentée à la section 3.4.3.

En résumé, comme observé dans les résultats des scénarios présentés, D-OFAP a des performances meilleures que G-OFAP et la solution *Baseline* mais il est surperformé par la solution optimale. Il convient de noter que, même si la différence entre le taux d'acceptation de D-OFAP et la solution optimale varie entre 4% et 7%, la complexité de la solution optimale est bien supérieure à celle de D-OFAP. Donc, ces simulations montrent l'efficacité de notre mécanisme à maximiser les flux acceptés par une passerelle IdO et les résultats montrent également l'importance d'exploiter simultanément les interfaces disponibles au sein des équipements IdO pour surmonter les problèmes de surcharge du réseau et augmenter la quantité de données acceptées.

Nous constatons également que la solution *Baseline* est plus efficace pour l'économie d'énergie des équipements IdO en la comparant avec nos solutions D-OFAP et G-OFAP. Ceci est expliqué par le fait que la solution *Baseline* a pour objectif d'affecter les services aux interfaces des équipements IdO tout en minimisant la consommation d'énergie totale de tous les équipements. De plus, la solution optimale ne tient pas en compte de la consommation énergétique lors de l'assignation des flux et dans le cas où la capacité des batteries des équipements est suffisante pour envoyer les flux vers l'interface sélectionnée (Amor *et al.*, 2019).

CONCLUSION GÉNÉRALE

Dans ce mémoire, nous avons abordé le problème d'assignation optimale des flux dans une passerelle IdO à interfaces multiples. Nous avons introduit un modèle mathématique qui modélise les ressources disponibles dans une passerelle et des équipements multi-interfaces et les flux échangés dans un réseau IdO. Nous avons défini le problème d'assignation optimale des flux, noté *Optimized Flow Assignment Problem (OFAP)*, comme un problème d'optimisation sous forme de programmation non linéaire en nombre entier (*INLP : Interger Non Linear programming*). L'objectif d'OFAP est de maximiser la quantité de données valides acceptées par la passerelle, en tenant compte des ressources réseau disponibles et en satisfaisant les échéances des flux.

Nous avons conçu en premier un mécanisme de contrôle de la passerelle qui applique l'assignation des flux aux interfaces en exécutant deux modules : le module de suivi et le module d'optimisation. Le module de suivi permet la collecte d'information sur l'état du réseau, ce qui est nécessaire pour la prise de décision de *handover*. Le module d'optimisation utilise les données collectées par le module de suivi pour exécuter l'algorithme qui permet l'assignation optimale des flux aux interfaces réseau. Nous avons également conçu un protocole de *handover* vertical qui permet de migrer les flux d'une interface à une autre. Le protocole proposé est déclenché par des commandes générées par le module d'optimisation de la passerelle.

Pour résoudre le problème d'optimisation, nous avons développé deux algorithmes qui exploitent efficacement les interfaces disponibles de la passerelle tout en satisfaisant les échéances des flux. Le premier algorithme (G-OFAP) est basé sur l'approche Greedy et le deuxième (D-OFAP) utilise la programmation dynamique. Nous avons ensuite effectué l'évaluation de performance de nos solutions en les comparant avec la solution optimale et un travail connexe. Les expérimentations montrent que les algorithmes proposés G-OFAP et D-OFAP produisent des résultats prometteurs pour différents scénarios. Les résultats ont montré l'efficacité des

algorithmes en fonction des paramètres suivants : 1) Taux d'acceptation des données valides, 2) nombre d'équipements IdO connectés à la passerelle, 3) Consommation d'énergie. Nous avons également implémenté notre solution, le module de contrôle de la passerelle, ainsi que le protocole de *handover* conçu dans un environnement réel IdO.

Dans les travaux futurs, nous comptons étendre notre formulation pour intégrer une fonction multi-objective et prendre en considération d'autres paramètres comme le coût d'activation d'interfaces et les préférences des utilisateurs. Ces paramètres permettront une meilleure assignation des flux aux interfaces en termes de coût et de qualité d'expérience (QoE). De plus, nous intéressons à élargir notre problème pour inclure plusieurs passerelles. Cela permettra aux équipements IdO d'exploiter les ressources réseau disponibles des passerelles voisines. En intégrant cette fonctionnalité, un équipement IdO sera désormais capable d'exécuter un *handover* horizontal entre deux passerelles IdO, si la qualité de service n'est pas garantie, en utilisant la passerelle courante.

ANNEXE I

ARTICLE PUBLIÉ DANS UNE CONFÉRENCE

Ce mémoire fait partie d'une publication, intitulé "Optimized Flow Assignment in a Multi-Interface IoT Gateway", qui a été publié à "International Wireless Communications & Mobile Computing Conference (IWCMC 2019, Tangier) " (Amor *et al.*, 2019).

Optimized Flow Assignment in a Multi-Interface IoT Gateway

Mohamed Ghazi Amor, Kim Khoa Nguyen, Chuan Pham, Mohamed Cheriet
École de Technologie Supérieure, University of Quebec, Canada
Email: {mohamed-ghazi.amor.1, kim-khoa.nguyen, chuan.pham.1, mohamed.cheriet}@etsmtl.ca

***Abstract*—The last few years have witnessed a significant increase in the deployment of heterogeneous Internet of Things (IoT) networks. IoT devices send data with different requirements such as tolerated delay and data rates. Emerging multi-interface IoT devices bring the flexibility of connecting to multiple heterogeneous access networks, which thus improves the network capacity. However, each network interface has its own constraints in terms of network coverage, capacity, packet loss rates, etc. An efficient utilization of the available multiple interfaces in IoT gateways would improve the network performance. Therefore, it is crucial to design a flow assignment mechanism to select the appropriate interface that best satisfies the flow's requirements and maximizes the amount of data accepted by an IoT gateway. In this work, we model and formulate the optimized flow assignment problem (OFAP) in a multi-interface IoT gateway. Then, we develop two heuristic algorithms to find a feasible solution for OFAP. The first algorithm is based on the greedy approach and the second uses dynamic programming to assign flows to interfaces. We provide simulation results that show the effectiveness of our algorithms.**

I. Introduction

Over the past decade, Machine to Machine (M2M) networks have witnessed evolution in their communication schemes [1]. Nowadays, Internet of Things (IoT) devices are often equipped with multiple interfaces. These smart objects send and receive data through different communication protocols such as Zigbee, WiFi, Bluetooth, LoRa and 3G to a multi-interface gateway [2]. This heterogeneity brings the advantage of accessing to multiple network resources which may make an improvement in the network performance and avoid congestion. However, each communication technology has its own specifications. Although available resources are often limited, IoT devices may be called on to provide low-latency guarantees to deadline sensitive applications that are very common in IoT use cases [3]. An example of a deadline sensitive application is an emergency medicine service with low-latency constraints associated with body sensors or patient health devices [4].

As a gateway has multiple network interfaces, IoT devices may not deliver data to a proper interface without an efficient mechanism that assigns the IoT flows to available interfaces while satisfying the application's requirements. The lack of an assignment mechanism may lead to congestion, in particular when the number of devices using the same interface increases. Consider an example of a plant monitoring application [5] which monitors large scale green spaces by collecting information such as soil moisture, humidity, wind, pH, and CO_2. This application sends pictures of leaves to a cloud server using a high-definition camera to allow experts to detect plant diseases. In this scenario, the IoT device is equipped with two different interfaces WiFi and Zigbee. The flows are assigned statically by the providers of end devices to the network interfaces. For this example, the soil humidity, temperature, and CO_2 measures are assigned statically to the Zigbee interface because it consumes less energy than WiFi. As for picture data, it will be assigned to WiFi interface since it requires much more bandwidth. However, the network traffic can vary and more services can be deployed, or new devices can join the network. As a result, Zigbee can be overloaded overtime. Therefore, a static assignment of the flows can cause network congestion and higher delay. Moreover, when traffic is assigned statically to interfaces, this may cause saturation of certain interfaces despite the availability of other interfaces which can handle more traffic. To the best of our knowledge, this issue has been investigated in prior research like [6], [7] and [8], which propose resource allocation and service to interface assignment to ensure the quality of service and to optimize the energy consumption of IoT devices. However, they do not take into account the deadline requirements of applications.

In this paper, we propose a flow assignment mechanism that maximizes the amount of valid data accepted by the gateway, which is the data received within its respective deadline. We perform this maximization by assigning the flows to the suitable network interface that satisfies not only the deadline and data rate requirements of flows but also the energy capacity of devices. Our contributions can be summarized as follows:

- We introduce a mathematical model that characterizes the available resources across multi-interface end devices and models the exchanged flows between the IoT devices and the multi-interface gateway. We define the optimized flow assignment problem (OFAP) to maximize the amount of valid data accepted by the gateway, taking into account the available network resources and satisfying other flow requirements.
- We design a switching mechanism between gateway interfaces that enables a seamless handover and ensures the continuity of the communication while switching.
- We propose two algorithms to solve the optimized flow

BIBLIOGRAPHIE

Abbas, Z. & Yoon, W. (2015). A Survey on Energy Conserving Mechanisms for the Internet of Things : Wireless Networking Aspects. *Sensors*, 15, 24818-24847.

Accenture. (2015, January, 19). Winning with the industrial Internet of Things. Repéré à https://www.accenture.com/us-en/insight-industrial-internet-of-things.

Amor, M. G., Khoa Nguyen, K., Pham, C. & Cheriet, M. (2019, June). Optimized Flow Assignment in a Multi-Interface IoT Gateway. *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, pp. 1379-1384. doi : 10.1109/IWCMC.2019.8766770.

Angelakis, V., Avgouleas, I., Pappas, N., Fitzgerald, E. & Yuan, D. (2016, Oct). Allocation of Heterogeneous Resources of an IoT Device to Flexible Services. IEEE Internet of Things Journal. 3(5), 691-700.

Apache. (2019a, Mars, 10). Apache Spark. Repéré à https://spark.apache.org/.

Apache. (2019b, Mars, 10). Apache Storm. Repéré à http://storm.apache.org/.

Arduino. (2019, Mars, 8). Arduino UNO documentation. Repéré à https://www.arduino.cc/en/Main/AboutUs.

Augustin, A., Yi, J., Clausen, T. H. & Townsley, W. (2016). A Study of LoRa : Long Range and Low Power Networks for the Internet of Things. *Sensors*, 16, 1466-1469.

Awad, A., Mohamed, A. & Chiasserini, C. (2016, April). User-centric network selection in multi-RAT systems. *2016 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 97-102.

Awad, A., Mohamed, A. & Chiasserini, C. (2017). Dynamic Network Selection in Heterogeneous Wireless Networks : A user-centric scheme for improved delivery. *IEEE Consumer Electronics Magazine*, 6(1), 53-60.

Chang, C.-T., Chang, C.-Y., Dario Borja Martinez, R., Chen, P.-T. & Chen, Y. (2015). An IoT Multi-Interface Gateway for Building a Smart Space. *Open Journal of Social Sciences*, 03, 56-60.

Datta, S. K., Bonnet, C. & Nikaein, N. (2014, March). An IoT gateway centric architecture to provide novel M2M services. *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 514-519.

Desai, P., Sheth, A. & Anantharam, P. (2015, June). Semantic Gateway as a Service Architecture for IoT Interoperability. *2015 IEEE International Conference on Mobile Services*, pp. 313-319.

Factory, S. (2018, Mars, 9). SIG 2014. Repéré à https://sig2014.esrifrance.fr/part_factorysystemes.aspx.

fritzing. (2019, Juin, 9). Fritzing projects. Repéré à http://fritzing.org/projects/dht22-esp8266-and-arduino-uno.

Fu, H., , & Fang, Y. (2012, March). Energy-efficient reporting mechanisms for multi-type real-time monitoring in Machine-to-Machine communications networks. *2012 Proceedings IEEE INFOCOM*, pp. 136-144.

Gaur, A., Scotney, B., Parr, G. & McClean, S. (2015). Smart city architecture and its applications based on IoT. *Procedia computer science*, 52, 1089–1094.

IBM. (2019, February, 3). CPLEX : IBM's Linear Programming Solver. Repéré à http://www.ilog.com/product/cplex/.

Islam, S. M. R., Kwak, D., Kabir, M. H., Hossain, M. & Kwak, K. (2015). The Internet of Things for Health Care : A Comprehensive Survey. *IEEE Access*, 3(9), 678-708.

Khattab, O. & Alani, O. (2013, April). Survey on Media Independent Handover (MIH) Approaches in Heterogeneous Wireless Networks. *European Wireless 2013 ; 19th European Wireless Conference*, pp. 1-5.

Kim, S., Choi, H. & Rhee, W. (2015, Aug). IoT home gateway for auto-configuration and management of MQTT devices. *2015 IEEE Conference on Wireless Sensors (ICWiSe)*, pp. 12-17.

Liu, L., Sun, L. & Ifeachor, E. (2016, Oct). User-centric QoE-driven vertical handover framework in heterogeneous wireless networks. *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1-8.

Min, D., Xiao, Z., Sheng, B., Quanyong, H. & Xuwei, P. (2014). Design and implementation of heterogeneous IOT gateway based on dynamic priority scheduling algorithm. *Transactions of the Institute of Measurement and Control*, 36(7), 924-931.

Morrow, M. (2004). *Wireless Network Coexistence* (éd. 1). New York Chicago : McGraw-Hill Networking.

Neal, W. (2012). Analysis of Algorithms : Subset Sum Problem [Notes de cours]. Repéré à http://www.cs.utsa.edu/~wagner/CS3343/ss/ss.html.

Partha, P. R. (2017). Internet of things for smart agriculture : Technologies, practices and future direction. *Journal of Ambient Intelligence and Smart Environments*, 4(9), 395-420.

Pi, R. (2019, Mars, 8). Raspberry Pi documentation. Repéré à https://www.raspberrypi.org/.

Porkodi, R. & Bhuvaneswari, V. (2014, March). The Internet of Things (IoT) Applications and Communication Enabling Technology Standards : An Overview. *2014 International Conference on Intelligent Computing Applications*, pp. 324-329.

Pycom. (2019, Mars, 8). Pycom : Next Generation Internet of Things Platform. Repéré à https://pycom.io/.

Raschellà, A., Bouhafs, F., Seyedebrahimi, M., Mackay, M. & Shi, Q. (2017). Quality of Service Oriented Access Point Selection Framework for Large Wi-Fi Networks. *IEEE Transactions on Network and Service Management*, 14(2), 441-455.

Ray & Pratim, P. (2017). Internet of things for smart agriculture : Technologies, practices and future direction. *Journal of Ambient Intelligence and Smart Environments*, 9(4), 395 - 420.

Ray, P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University - Computer and Information Sciences*, 30(3), 291 - 319.

Sha, M., Paul, S., Ravichandra, N. & Chowdhury, S. (2017, June). Adaptive radio and transmission power selection for Internet of Things. *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, pp. 1-10.

Shah, J. & Mishra, B. (2016). IoT enabled environmental monitoring system for smart cities. *2016 International Conference on Internet of Things and Applications (IOTA)*, pp. 383–388.

Sinha, R. S., Wei, Y. & Hwang, S.-H. (2017). A survey on LPWA technology : LoRa and NB-IoT. *ICT Express*, 3(1), 14 - 21.

Stočes, M., Vaněk, J., Masner, J. & Pavlík, J. (2016). Internet of Things (IoT) in Agriculture - Selected Aspects. *Agris on-line Papers in Economics and Informatics*, 8(1), 83-88.

Theoleyre, F. & Pang, A.-C. (2013). *Internet of Things and M2M Communications*. Wharton, TX, USA : River Publishers.

Tomasik, J. (2012). *Garantie de la qualité de service et évaluation de performance des réseaux de télécommunications*. (Habilitation à diriger des recherches, Université Paris Sud - Paris XI). Repéré à https://tel.archives-ouvertes.fr/tel-00661575.

Tosh, D. K. & Sengupta, S. (2015, March). Heterogeneous access network(s) selection in multi-interface radio devices. *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 117-122.

Wen, Y., Zhang, W. & Luo, H. (2012, March). Energy-optimal mobile application execution : Taming resource-poor mobile devices with cloud clones. *2012 Proceedings IEEE INFOCOM*, pp. 2716-2720.

Wu, D., Arkhipov, D. I., Asmare, E., Qin, Z. & McCann, J. A. (2015, April). UbiFlow : Mobility management in urban-scale software defined IoT. *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 208-216.

Zainuddin, Z., Idris, R. & Azis, A. (2019, Mars). Water Quality Monitoring System for Vannamae Shrimp Cultivation Based on Wireless Sensor Network In Taipa, Mappakasunggu District, Takalar. *First International Conference on Materials Engineering and Management - Engineering Section (ICMEMe 2018)*.

Zanella, A., Bui, N., Castellani, A., Vangelista, L. & Zorzi, M. (2014). Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1), 22-32.