

Ordonnancement de données multiprotocoles des objets connectés dans la maison intelligente

par

Milad AMIRIYAN

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN CONCENTRATION GÉNIE RÉSEAUX ET
TÉLÉCOMMUNICATION
M. Sc. A.

MONTRÉAL, LE 20 DÉCEMBRE 2021

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC



Milad Amiriyani, 2021



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

Monsieur Kim Khoa Nguyen, directeur de mémoire
Département de génie électrique à l'École de Technologie Supérieure

Madame Ghizlane El Boussaidi, présidente du jury
Département de génie logiciel et des TI à l'École de Technologie Supérieure

Monsieur Aris Leivadeas, membre du jury
Département de génie logiciel et des TI à l'École de Technologie Supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 1 DÉCEMBRE 2021

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Ce travail n'aurait pas pu être réalisé sans les aides et soutiens de mon directeur, Professeur Kim Khoa Nguyen. Je tiens à exprimer ma gratitude envers lui. Je remercie également tous mes enseignants à l'École Technologie Supérieure qui ont transmis leur savoir afin de parvenir au terme de ce travail.

J'adresse également ma gratitude envers Mr Benoit Gendron, fondateur et innovateur de société Latence Technologies, de m'avoir accepté comme stagiaire au sein de son entreprise.

Mes derniers remerciements vont à ceux qui m'ont soutenu de près ou de loin. Je pense particulièrement à ma famille et mes proches : Mon père, mon frère et ma sœur, qui m'ont toujours soutenu moralement et financièrement durant mes séjours ici à Québec.

Je dédie spécialement ce travail à notre défunte mère, Fereshteh Zibaei ; sans ses conseils et encouragements, je n'aurais jamais pu continuer.

Ordonnancement de données multiprotocoles des objets connectés dans la maison intelligente

Milad AMIRIYAN

RÉSUMÉ

Au cours des dernières années, le nombre d'appareils et d'applications de l'Internet des objets (IdO) et leur volume de données générées ont augmenté rapidement. Afin de répondre aux diverses exigences des utilisateurs en termes de transmission de données et de réduire des embouteillages dans le réseau domestique intelligent, des objets connectés intelligents multi-interfaces ont été déployés pour offrir la flexibilité d'accès à plusieurs réseaux d'accès hétérogènes. Dans ce mémoire, nous proposons un mécanisme pour affecter dynamiquement les flux de trafic de ces objets aux interfaces des passerelles d'IdO tout en satisfaisant leur qualité de service (QoS), et simultanément augmenter la quantité de données valides acceptées par les passerelles.

De ce fait, nous formulons le problème d'assignation optimale des flux de données provenant des appareils IdO ayant plusieurs interfaces vers les passerelles IdO (OFAP-MIG) en tant qu'un modèle de programmation mixte non entière (PMNE). Ensuite, nous proposons une solution basée sur une approche d'apprentissage par renforcement profond (ARP) pour résoudre ce problème NP-difficile. Les agents déployés dans les passerelles IdO et le nuage informatique utilisent les réseaux de neurones profond (RNP) pour calculer une solution optimale basée sur un espace d'état, un espace d'action et une fonction de récompense. Les résultats de simulation montrent que notre solution proposée surpasse la solution centralisée, qui a été présentée dans des travaux antérieurs, en termes de la quantité totale de données transférées par les passerelles et du taux d'acceptation.

Mots-clés: IdO, Optimisation, Ordonnancement de Trafic, Multi-Interface Hétérogène, Apprentissage par Renforcement Profond, Maison Intelligente

Multiprotocol Flow Assignment in Smart Home IoT Network

Milad AMIRIYAN

ABSTRACT

The number of Internet of Things (IoT) devices and applications is growing rapidly, as well as the amount of data generated by these devices. To afford various user requirements in data transmission, and reduce bottlenecks, multi-interface intelligent radio devices have increasingly been deployed to provide the flexibility of simultaneous access to several heterogeneous access networks. In this article, we propose a mechanism to dynamically assign the flows of these devices to the appropriate gateway interfaces while satisfying their QoS, and simultaneously increase the valid data accepted by the gateways of our IoT network.

In this thesis, we formulate the problem of optimal assignment of data flows coming from IoT devices having multiple interfaces to several multi-interface IoT gateways (OFAP-MIG) as a Mixed Non-Integer Programming model (MNIP), and propose a solution based on deep reinforcement learning (DRL) approach to solve this NP-hard problem. The agents deployed in the IoT gateways and the central cloud employ Deep Neural Network (DNN) to find the optimal solution based on a state space, an action space and a reward function. The simulation results shows the proposed solution outperforms the centralized one, which has been presented in prior work, in terms of the total amount of data transferred by the gateways, and the acceptance rate.

Keywords: IoT, Optimization, Flow Assignment, Heterogeneous Multi-Interface, Deep Reinforcement Learning, Smart Home

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 ÉTAT DE L'ART	9
1.1 L'Internet des Objets	9
1.2 Éléments de l'écosystème IdO	10
1.2.1 Passerelles IdO	10
1.2.1.1 Fonctionnalité	11
1.2.2 Objets Connectés	12
1.2.3 Protocoles de Communication IdO	12
1.3 Gestion des Flux dans les Réseaux Hétérogènes	15
1.4 Approches Récentes	16
1.5 Discussion	27
1.6 Conclusion	29
CHAPITRE 2 MÉTHODOLOGIE	31
2.1 Description du Système	31
2.1.1 Agent	32
2.1.1.1 Module de Surveillance	33
2.1.1.2 Module d'Optimisation	33
2.1.2 Mécanisme de Commutation	33
2.1.3 Exemple de Motivation	34
2.2 Formulation du Problème	35
2.2.1 Problème d'Optimisation	36
2.2.2 Contraintes	37
2.2.2.1 Contrainte de type d'interface pour la connexion entre l'objet connecté et la passerelle	37
2.2.2.2 Intégrité du Flux	37
2.2.2.3 Capacité Énergétique	37
2.2.2.4 Écheance des Flux	38
2.2.2.5 Capacité du Réseau	39
2.3 Solutions Proposées	40
2.3.1 Complexité du Problème	40
2.3.2 DRL-OFAP-MIG	42
2.3.3 GA-OFAP-MIG	47
2.4 Conclusion	52
CHAPITRE 3 RÉSULTATS EXPÉRIMENTAUX	53
3.1 Configuration de la simulation	53
3.1.1 Environnement de simulation	53
3.1.2 Scénarios de simulation	54

3.2	Résultats de la simulation	56
3.3	Discussion	57
CONCLUSION ET RECOMMANDATIONS		61
BIBLIOGRAPHIE		63

LISTE DES TABLEAUX

	Page
Tableau 1.1	Tableau comparatif des appareils IdO 12
Tableau 1.2	Tableau comparatif des réseaux 16
Tableau 1.3	Objectifs des approches 27
Tableau 1.4	Caractéristiques des méthodes récentes d’optimisation de réseau IdO 28
Tableau 2.1	Saisie de scénario 35
Tableau 2.2	Liste des notations 40
Tableau 3.1	Caractéristiques des interfaces pour les simulations 54
Tableau 3.2	Quantité de données générées 55
Tableau 3.3	Taille du réseau 55
Tableau 3.4	Durée du processus 55

LISTE DES FIGURES

	Page
Figure 0.1 Réseau IdO Hétérogène	2
Figure 0.2 Maison Intelligente	4
Figure 1.1 Passerelle IdO	11
Figure 1.2 Technologies de Connectivité IdO Sans Fil	13
Figure 1.3 Justification de l'Architecture Muppet	19
Figure 1.4 Système MEC NB-IoT et Modèle de File d'Attente	22
Figure 1.5 Cadre du système	25
Figure 1.6 Architecture du Système	26
Figure 2.1 Architecture du Système	31
Figure 2.2 Exemple de Problème d'Assignment de Flux	34
Figure 2.3 Cadre de RL	42
Figure 2.4 Architecture DRL	44
Figure 2.5 Organigramme de la Méthode GA	47
Figure 2.6 Codage de Chromosome	49
Figure 2.7 Croisement en un Point	51
Figure 2.8 Mutation	52
Figure 3.1 Scénario 1 - Quantité de Données Générées (Somme)	56
Figure 3.2 Scénario 1 - Quantité de Données Générées (Taux d'Acceptation)	57
Figure 3.3 Scénario 2 - Taille du Réseau (Somme)	58
Figure 3.4 Scénario 2 - Taille du Réseau (Taux d'Acceptation)	58
Figure 3.5 Durée du Processus	59

LISTE DES ALGORITHMES

	Page
Algorithme 2.1 Le cadre Basé sur les DRL	45

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

IoT	Internet of Things
IdO	Internet des Objets
OFAP	Optimized Flow Assignment Problem
OFAP-MIG	OFAP-Multi Interface Gateway
QoS	Quality of Service
QdS	Qualité de Service
DSF	Dynamic Switch-based Forwarding
SPLL	Shortest Path and Less Number of Links
CTMDP	Continuous-Time Markov Decision Process
PMNE	Programmation Mixte Non Entière
MNIP	Mixed Non-Integer Programming
DRL	Deep Reinforcement Learning
ARP	Apprentissage par Renforcement Profond
DNN	Deep Neural Network
DQN	Deep Q-learning Network
RNP	Réseau de Neurones Profond
WSN	Wireless Sensor Networks
M2M	Machine-to-Machine
WPAN	Wireless Personal Area Network
WLAN	Wireless Local Area Network
SDN	Software Defined Networking

INTRODUCTION

L'Internet des objets (IdO) est une nouvelle technologie qui se compose d'un continuum d'objets connectés qui sont capables de communiquer entre eux via la réseau Internet, avec l'aide de l'informatique omniprésente, la technologie de détection, et les protocoles tels que l'IPv6 et d'autres protocoles (Kang, Kim & Choo, 2017a). Les objets dans le réseau IdO sont typiquement des actionneurs et des capteurs, qui peuvent être connectés à tout moment et n'importe où.

Entre 2018 et 2025, le rapport GSMA IoT prévoit que le nombre d'appareils IdO triplera dans le monde, pour atteindre 25 milliards (Ray, 2018). L'IdO peut être appliqué à de nombreux domaines tels que les villes intelligentes, la fabrication, les transports, la santé, l'agriculture, etc (HIR, 2019).

Depuis quelques années, les chercheurs ont conçu et développé diverses plates-formes matérielles et logicielles IdO pour répondre aux demandes croissantes d'applications dans l'industrie et dans la vie quotidienne. Ces plates-formes sont composées de différents types de réseaux hétérogènes tels que WSN (Wi-Fi, Bluetooth, LoRa, Zigbee, etc.), MCN (3G/4G/LTE/5G) et WMN, pour collecter des données à partir des objets intelligents comme décrit dans la figure 0.1. Ces technologies réseautiques peuvent être classées en fonction de certains paramètres tels que : la consommation électrique, le débit binaire, la portée et la topologie. La quantité de données produites par le réseau IdO augmente chaque jour et pose des problèmes concernant la sécurité, la scalabilité, l'hétérogénéité et l'interopérabilité. L'optimisation du réseau IdO est alors décisive pour minimiser l'interférence des flux de trafic des différents services. Afin de satisfaire la contrainte des ressources disponibles dans le réseau IdO, une approche d'allocation de ressources est nécessaire afin de gérer les services et les assigner aux ressources disponibles. Les objets connectés envoient les données détectées, appelées flux de données IdO, aux passerelles à différents intervalles de temps. Une passerelle IdO est un appareil physique ou une plate-forme virtuelle qui connecte des capteurs, des modules IdO et des appareils intelligents à l'info

nuagique ou agit comme une unité centrale elle-même pour traiter les données collectées. Les passerelles servent de portail d'accès sans fil pour permettre aux appareils IoT d'accéder à Internet. Cependant, l'assignation optimale et efficace des flux de données IdO aux interfaces appropriées des passerelles du réseau IdO doit tenir compte de plusieurs paramètres telles que l'hétérogénéité, la bande passante des interfaces des passerelles, la date limite de l'application IdO, la consommation d'énergie, etc..

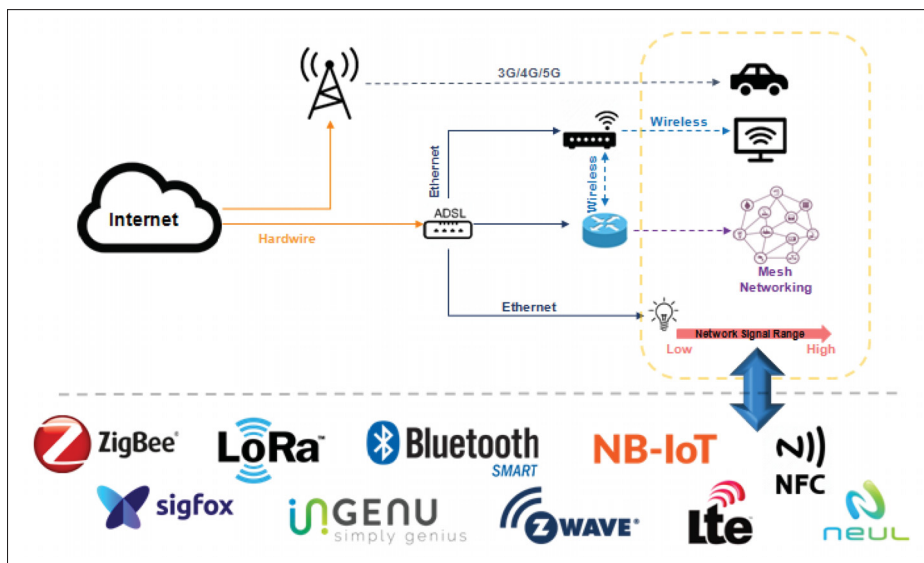


Figure 0.1 Réseau IdO hétérogène
Tirée de Silva et al. (2019)

Dans ce mémoire, nous étudions un mécanisme d'assignation de flux qui maximise la quantité de données valides acceptées par les passerelles, c'est-à-dire les données reçues dans leur délai respectif. Nous affectons les flux aux interfaces des passerelles IdO appropriées pour satisfaire non seulement les exigences de temps et de débit de données des flux, mais également la capacité énergétique des appareils. Contrairement aux travaux antérieurs, notre modèle prend en considération plusieurs protocoles et plusieurs interfaces de communication des objets connectés et des passerelles IdO.

L'allocation optimale des flux dans un tel réseau est un problème difficile que les techniques d'optimisation traditionnelles ne peuvent résoudre en temps réel (Chen et al., 2021). En conséquence, un contrôleur de réseau centralisé n'est pas capable de calculer la solution optimale instantanément. Nous développons dans ce mémoire une solution distribuée basée sur l'approche d'apprentissage par renforcement en profondeur multi-agents (DRL) qui est prometteuse pour l'allocation des ressources en temps réel. Notre solution consiste en plusieurs agents qui apprennent de manière dynamique en interagissant avec leur environnement. Les décisions optimales seront prises localement par chaque passerelle, puis améliorées par le modèle global produit dans le nuage informatique.

0.2 Problématique

Aujourd'hui, les objets connectés sont de plus en plus équipés de plusieurs interfaces réseaux, et utilisent différentes technologies de communication telles que le Wi-Fi, Bluetooth, LoRa, RFID, Zigbee, etc. 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. De ce fait, les passerelles IdO intelligentes introduites récemment dans le marché sont souvent conçues avec plusieurs interfaces de réseau pour servir une large gamme d'objets connectés (Tsai & Liu, 2018).

Cependant, ces objets ne peuvent pas transmettre leurs données à une interface appropriée d'une passerelle sans un mécanisme efficace qui alloue de manière optimale les flux de données aux interfaces disponibles. L'absence d'un tel mécanisme peut entraîner une congestion, surtout pour un grand nombre d'objets connectés (Amor, Nguyen, Pham & Cheriet, 2019).

Prenons l'exemple d'une maison intelligente avec plusieurs objets IdO comme décrit dans la figure 0.2. Ces appareils sont équipés d'interfaces ZigBee et WiFi et envoient leurs données vers un serveur sur l'infonuagique. Ces flux de données sont alloués de manière statique et

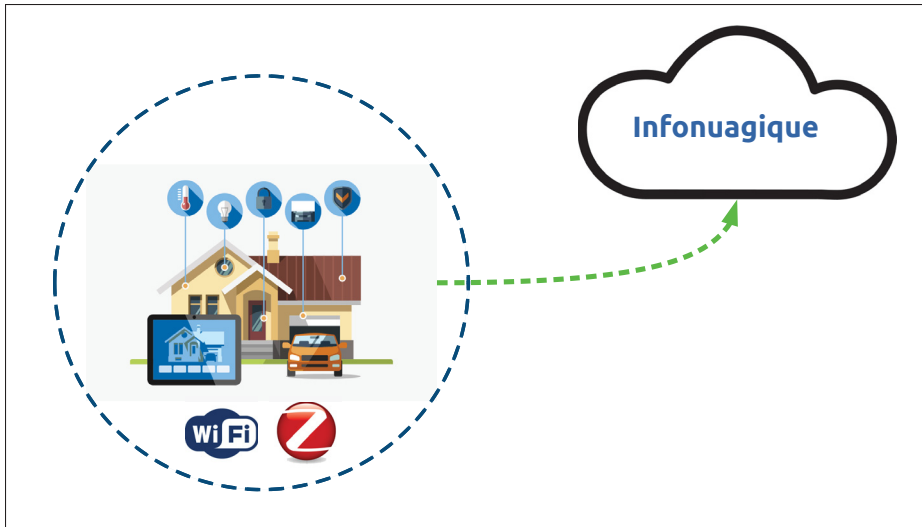


Figure 0.2 Maison Intelligente

aléatoire aux interfaces d'une passerelle. La caméra haute définition est affectée à l'interface WiFi et d'autres capteurs sont affectés à l'interface ZigBee. Toutefois, le trafic réseau peut varier en raison d'un autre appareil IdO qui essaie d'envoyer ses données vers le serveur sur l'infonuagique. En conséquence une affectation de flux statique peut provoquer une congestion ou une saturation de certaines interfaces dans le réseau. C'est pourquoi, la coexistence de différentes technologies d'accès sans fil étant plus une réalité qu'une hypothèse, les équipements IdO à interfaces réseaux multiples devront relever le défi d'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 de manière dynamique au cours du temps. Pour remédier à ce problème, un contrôleur doit être implémenté au niveau des passerelles. Afin de gérer simultanément et dynamiquement les interfaces et d'allouer les flux de données aux interfaces appropriées.

Comme mentionné, avec l'augmentation du nombre des objets connectés, une passerelle n'est pas capable de gérer tous les objets connectés, ce qui entraînera une diminution de la qualité de service. Pour résoudre ce problème, nous supposons que le système est composé de plusieurs

passerelles et également de la répartition des données entre ces passerelles de manière à augmenter les données globales acceptées par les passerelles tout en respectant leur QoS.

Dans ce mémoire, nous proposons un modèle mathématique pour le problème d'assignation des flux aux interfaces des passerelles multi-interface (Optimized Flow Assignment Problem-Multi Interface Gateway) dénoté dans le reste du mémoire par OFAP-MIG. L'objectif d'OFAP-MIG est de maximiser la quantité de données valides acceptées par les passerelles, en tenant compte des ressources disponibles et en satisfaisant la qualité de service des applications d'IdO.

0.3 Questions de recherche

Les questions de recherches suivantes nous permettront de subdiviser notre problématique :

(Q1) : Le domaine de l'Internet des objets repose sur plusieurs réseaux de connexion tels que les réseaux WSN et M2M, ainsi que les réseaux cellulaires. Tous ces réseaux utilisent des appareils 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 le temps de réponse et la bande passante.

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

(Q2) : Dans l'introduction, nous avons mentionné que le nombre d'applications IdO est en forte croissance. De plus, nouveaux appareils qui sont équipés de plusieurs interfaces, sont introduits pour servir ces applications qui ont des spécifications différentes. Cette variation en termes de ressources et d'applications a fait de l'environnement IdO un domaine très compliqué. Alors, la question est :

Comment peut-on assurer une meilleure exploitation des interfaces réseau dans nos passerelles du réseau IdO, pour servir les applications existantes et leurs exigences ?

0.4 Objectifs de la recherche

L'objectif principal de cette recherche est de proposer une solution pour assigner dynamiquement les flux de trafic des objets connectés équipés de plusieurs interfaces dans le réseau, aux interfaces des passerelles appropriées pour maximiser les données valides acceptées par les passerelles tout en satisfaisant la QoS du trafic. Cet objectif peut être divisé en deux sous-objectifs suivants :

- **Sous-objectif1 (SO1) :** Formuler un modèle du réseau IdO qui est composé de plusieurs passerelles IdO et de plusieurs objets connectés, chacun peut être équipé de plusieurs interfaces et peut communiquer en utilisant des protocoles différents. Ce modèle doit permettre de maximiser la quantité de données acceptées par les interfaces de ces passerelles tout en respectant les délais des flux et la capacité de chaque interface. Ce sous-objectif vise à répondre à la question de recherche Q1.
- **Sous-objectif2 (SO2) :** Développer une solution pour résoudre le modèle dans SO1 en quasi temps réel. Cette solution vise à assigner les flux de trafic dynamiquement et efficacement aux interfaces des passerelles en maximisant la quantité de données acceptées. Ce sous-objectif correspond à la question Q2.

0.5 Contributions

Dans ce mémoire, une revue de la littérature sur les méthodologies utilisées pour l'allocation optimale des ressources sur les interfaces des objets connectés est un bon début pour atteindre ces objectifs. Ensuite, trois contributions principales peuvent être envisagées :

- Un modèle mathématique pour maximiser la quantité de données valides acceptées par les passerelles IdO dans le réseau IdO tout en respectant les délais des flux de données.

- Une solution basée sur algorithme génétique pour résoudre l’assignation de flux optimisée de plusieurs passerelles IdO. Cet algorithme va déterminer la borne de la solution optimale.
- Une solution basée sur les DRL pour résoudre le problème en quasi temps réel.
- Plusieurs simulations ont été réalisées pour comparer nos solutions proposées avec les travaux antérieurs.

0.6 Plan du mémoire

Pour bien mener notre travail, nous allons articuler ce mémoire en 3 chapitres. Le premier chapitre est constitué par l’état de l’art. Ainsi, nous verrons d’abord l’écosystème IdO et ses éléments. Notre revue est basée sur des articles scientifiques qui traitent des problématiques similaires à la notre. Nous verrons également plusieurs méthodes utilisées par les chercheurs pour optimiser l’allocation de ressources sur le réseau IdO.

Le deuxième chapitre contient la méthodologie que nous allons adopter pour la résolution de notre problématique. Dans ce chapitre, nous allons présenter de la méthode d’assignation des flux multiprotocol dans un réseau IdO de maison intelligente. Nous abordons aussi, deux méthodes basées sur l’algorithme génétique et DRL pour résoudre le problème.

Le troisième chapitre se focalise sur les résultats expérimentaux et leurs interprétations. Les configurations de la simulation, les résultats obtenus et leurs interprétations font parties des subdivisions de ce chapitre.

Enfin nous allons conclure notre travail en récapitulant les différentes parties de nos contributions. Une proposition pour les travaux futurs complétera ceci et sera discutée dans cette conclusion.

CHAPITRE 1

ÉTAT DE L'ART

Dans la littérature, de nombreuses approches ont été proposées pour réduire la charge du réseau IdO, garantir des délais acceptables, maintenir un débit élevé et une faible consommation d'énergie en exploitant des interfaces réseau disponibles. Les décisions de sélection des interfaces se font selon 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 d'abord les définitions de l'IdO et de ses éléments qui incluent les protocoles de communication que les équipements IdO utilisent pour se connecter. Ensuite, dans la revue de littérature, nous exposons quelques solutions et ses différentes techniques utilisés pour résoudre le problème d'optimisation de l'affectation des flux aux interfaces. Enfin, nous présentons une étude comparative qui met en évidence les avantages et les limites de chaque solution existante.

1.1 L'Internet des Objets

L'Internet des objets (IdO) est un nouveau paradigme à l'époque actuelle qui a donné naissance à l'avènement de l'informatique omniprésente. Il aide un grand nombre d'équipements IdO à échanger des données détectées, à réagir aux événements et à interagir avec l'environnement en se connectant à Internet. Cette interaction entre des appareils hétérogènes permet des applications omniprésentes à l'IdO. Comme le nombre d'objets connectés à Internet est énorme et que l'IdO englobe une gamme extrêmement large d'applications, il est donc crucial d'avoir une architecture qui permet un contrôle flexible et la connectivité (Farhan, Kharel, Kaiwartya, Hammoudeh & Adebisi, 2018).

De plus, pour éviter les congestions, le trafic doit être géré de manière décentralisée (Uddin, Mukherjee, Chang & Lakshman, 2018). Il existe de nombreux protocoles pour connecter des

appareils IdO. Dans les réseaux IdO, nous avons de nombreux équipements et connexions, ce qui fait de ce réseau un système complexe et évolutif (Lei, Xu, Xiong, Zheng & Xiang, 2019).

L'environnement IdO se compose de diverses architectures de service, protocoles et conception de réseau pour gérer des millions d'appareils IdO pour l'échange de données. L'IdO doit prendre en charge des appareils hétérogènes basés sur des interfaces variantes avec un grand nombre de ressources disponibles (Samie, Bauer & Henkel, 2020).

1.2 Éléments de l'écosystème IdO

Dans cette section, nous allons présenter les composants de l'écosystème IdO qui est divisé en trois parties : les passerelles, les objets connectés et les protocoles de communications.

1.2.1 Passerelles IdO

La passerelle IdO est une composante importante du réseau (Desai, Sheth & Anantharam, 2015). Sa principale fonctionnalité est de transmettre des flux de données aux centres de données hébergés dans l'infonuage pour traitement (Amor *et al.*, 2019). Elles ont également un petit système d'exploitation avec des fonctionnalités et des ressources limitées en termes de mémoire, de capacité de calcul et de consommation d'énergie. Ainsi, la passerelle effectue de petites opérations telles que le filtrage des données et la conversion de données de format différent en un format unique (Aloi, Caliciuri, Fortino, Gravina, Pace, Russo & Savaglio, 2017).

Dans un grand écosystème IdO, les passerelles sont gérées par un contrôleur à hautes performances. Ainsi, une passerelle dans ce système doit réaliser les communications suivantes (Kang & Choo, 2018) :

- appareil IdO vers Passerelle ;
- passerelle vers l'application IdO ;
- passerelle vers le contrôleur ;

1.2.1.1 Fonctionnalité

Premièrement, une passerelle doit faire face à l'hétérogénéité et aux différents types de données collectées à partir de différents capteurs. Elle doit prendre en charge plusieurs protocoles pour pouvoir échanger des données avec les appareils IdO. Ensuite, elle doit envoyer les données au contrôleur ou à l'application IdO à l'aide d'Internet, comme indiqué dans la figure 1.1.

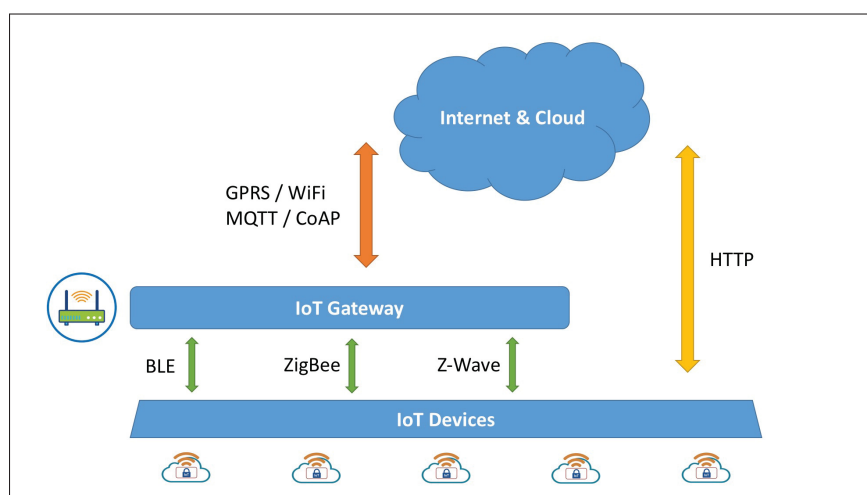


Figure 1.1 Passerelle IdO

Deuxièmement, la gestion du réseau est une fonction primordiale pour la passerelle IdO, ce qui concerne l'amélioration de performance du réseau et la qualité de service. Par exemple, la passerelle doit affecter des flux de données IdO à ses interfaces d'une manière dynamique, ou mettre à jour des modules logiciels dans les objets connectés.

Enfin, la passerelle IdO doit assurer une gestion efficace de ses ressources et interfaces hétérogènes ainsi que celles des objets connectés.

De plus, les passerelles IdO fonctionnent selon trois modes :

1. **Passive** : Dans ce mode, il est nécessaire d'ajouter de nouveaux objets ou de supprimer des objets existants manuellement.
2. **Semi-Automatisé** : Dans ce mode, il existe un lien entre les objets qui sont ajoutés et la passerelle pour la connexion. Il est basé sur une architecture plug-configurable qui montre

que les objets peuvent être branchés en fonction des besoins du réseau. Il est plus flexible que les passerelles passives en raison de l'interface externe (Guoqiang, Yanming, Chao & Yanxu, 2013).

3. **Entièrement Automatisé :** Dans ce mode, les passerelles ont la capacité d'auto-configuration. Il n'est pas nécessaire de prendre l'autorisation de l'administrateur. Elles peuvent ajouter ou supprimer les objets connectés eux-mêmes. (Gaur & Tahiliani, 2015).

1.2.2 Objets Connectés

Les objets connectés sont des éléments de base du système IdO. Le tableau 1.1 résume la différence entre certains objets connectés connus sur le marché (Amor *et al.*, 2019). Il montre une comparaison de différents appareils IdO en fonction de leurs ressources réseau, c'est-à-dire des protocoles de communication qu'ils utilisent et des ressources de traitement, ainsi que de leur prise en charge des appareils externes. Ces objets connectés sont présents dans toutes les couches de l'architecture IdO. Ils peuvent être open source ou propriétaires. Les objets connectés ont des capacités limitées en terme de stockage, mémoire, puissance, capacité de calcul et batterie.

Tableau 1.1 Tableau comparatif des appareils IdO

Caractéristiques	WiPy	LoPy	Gpy	Proposé dans (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

1.2.3 Protocoles de Communication IdO

Le progrès phénoménal de l'IdO a donné naissance à de nouveaux 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 (Kang, Kim & Choo, 2017b).

La figure 1.2 montre certaines des technologies de connectivité sans fil IdO avec leurs caractéristiques. Dans cette partie, nous présenterons certains des protocoles les plus utilisés :

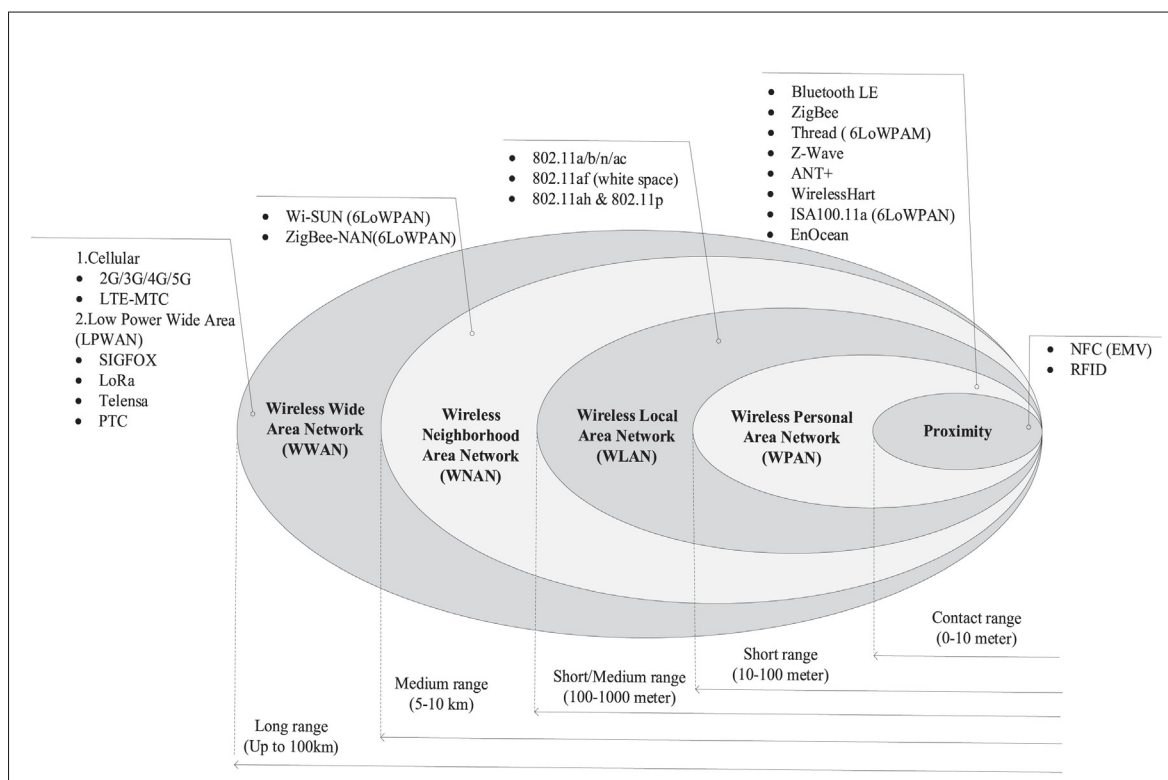


Figure 1.2 Technologies de Connectivité IdO Sans Fil
Tirée de Mahmoud et al. (2016)

1. **NFC** : La portée de ce protocole étant très courte, (e.g, les objets mobiles qui l'utilisent peuvent communiquer entre eux sur quelques centimètres). Les données peuvent être transmises en quelques secondes entre les appareils NFC (Al-Saedi & Azim, 2017).
2. **Réseau Personnel Sans Fil à Faible Consommation Énergétique (6LowPAN)** : 6LowPAN permet aux objets intelligents de se connecter à l'Internet à l'aide du protocole IPv6. 6LowPAN élimine les obstacles à l'utilisation du protocole IPv6 et permet une capacité de traitement limitée, un faible débit de données (Honggang, Chen & Leyu, 2018).

3. **Bluetooth (BLE) :** BLE permet une transmission rapide de paquets de données jusqu'à 2 Mbps dans les bandes industrielles, scientifiques et médicales (ISM) (Baert, Camerlynck, Crombez & Hoebeke, 2019). Les appareils qui implémentent le protocole BLE sont classés en deux types ; maître et esclave où le maître peut se connecter à plusieurs esclaves (Dian, Yousefi & Lim, 2018).
4. **ZigBee :** La norme IEEE 802.15.4 LR-WPAN (Zigbee) a été conçue afin de fournir une connectivité sans fil évolutive, à faible coût et à faible consommation pour une grande variété d'applications (Pan, He, Wu, Fang, Cao & Liao, 2018).
5. **Low Power Wide-Area-Networks (LPWAN) :** Les protocoles LPWAN sont des protocoles à faible consommation, à faible bande passante et à faible coût, en particulier dans les communications sur de longues distances (e.g., avec des portées de transmission de 1 m à 50 km) (Liya & Aswathy, 2020). Voici une brève présentation des caractéristiques de chaque protocole :
 - **Long Range Wide-Area-Networks (LoRaWAN) :** Il s'agit d'un protocole de communication de couche physique, avec une faible consommation d'énergie et une longue durée de vie de la batterie pouvant atteindre 10 ans. LoRaWAN est utilisé dans les services et applications de réseau étendu (WAN), tels que le M2M, les applications industrielles et les villes intelligentes, qui nécessitent de longues distances de communication allant de (2 à 5) km dans les zones urbaines et jusqu'à 15 km dans les zones suburbaines (Lehong, Isong, Lugayizi & Abu-Mahfouz, 2020). Le débit de données de ce protocole varie de 0,3 kbps à 50 kbps (Liya & Aswathy, 2020).
 - **Narrowband Internet of Things (NB-IoT) :** C'est une nouvelle technologie d'accès radio basée sur la norme d'évolution à long terme (LTE) mais avec des fonctionnalités minimales afin de réduire la consommation d'énergie des appareils IdO à ressources limitées. Il fonctionne sur (180-200) kHz et utilise également la modulation QPSK (3GPP, 2016).
 - **SigFox :** Ce protocole fonctionne sur une bande de fréquence de 868 MHz, où le spectre est divisé en 400 canaux de 100 Hz. Les appareils IdO peuvent transmettre jusqu'à 140

paquets par jour avec un débit de données allant jusqu'à 100 bps et son signal peut atteindre des distances de (30 à 50) km dans les zones rurales (Osman & Abbas, 2018).

6. **WiFi** : Cette technologie permet des services équivalents à un réseau filaire avec un débit élevé. L'architecture de ce réseau 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 est suffisamment rapide pour des applications telles que FTP, HTTP, Streaming vidéo, etc (Sobin, 2020).
7. **Cellular** : Tout service IdO qui fonctionne sur de longues distances peut bénéficier du déploiement des technologies du système mondial de communication mobile (GSM) telles que les protocoles de communication cellulaire 3G, 4G et 5G, car ils ont une capacité de transmission élevée, en particulier 4G et 5G. Cependant, la communication via des protocoles cellulaires est coûteuse et consomme beaucoup d'énergie.

Le tableau 1.2 compare les fonctionnalités et les caractéristiques des protocoles IdO les plus populaires dans la couche de liaison de données.

1.3 Gestion des Flux dans les Réseaux Hétérogènes

Les réseaux sans fil hétérogènes sont complexes, en raison des différentes caractéristiques des protocoles de communication. Cette hétérogénéité crée des défis pour la sélection des interfaces réseau pour envoyer les flux de données, ce qui complique la gestion des flux dans les réseaux sans fil hétérogènes. Un algorithme robuste capable de prendre des décisions difficiles est requis pour obtenir la performance voulue. Une décision sous-optimale peut entraîner des effets indésirables sur le réseau, tels que : la perte de données critiques, la saturation du réseau et de faible performance ou faible qualité de service.

Dans la section suivante, nous présentons des travaux connexes qui ont proposé des mécanismes d'ordonnancement de flux dans un environnement IdO hétérogène.

Tableau 1.2 Tableau comparatif des réseaux

Protocol	6LowPAN	BLE	ZigBee	NB-IoT	Cellular
Norme de Réseau	IEEE 802.15.4	802.15.1	IEEE 802.15.4	3GPP	LTE/5G
Type de Réseau	Star/Mesh	Star	Hybrid	Star	Mobile
Bande de Fréquence	2.4GHz	(2.40–2.48) GHz	2.4GHz 915Mhz 868Mhz	3.75kHz 15kHz 180-200kHz 850-900MHz	(1800–2400) Mhz 3G (2–8GHz) 4G
Portée de Transmission	(10–100)m	up to 100m	(10–100)m Sub-GHz up to 1km	1km(urban) 10km(rural)	10km/h*t (3G) 500km/h*t (4G)
Consommation d'Énergie	*	15 mA	30 mA	(3–50) μ A	800mA (3G) (1000–3500) mW (4G)
Limitation des Nœuds	65000	65535 nodes	65000	100K devices per cell	4000 devices/ km^2 (4G)
Débit de Données	(20,40,250) kbps	125Kbps (1,2)Mbps	250 kbps	(30-60)kbps 200kbps	144kb-2Mbps (3G) 100Mb-1Gbps (4G)
Applications	Maison intelligente, Agriculture intelligente, IdO industriel structurel, Surveillance des soins de santé	Téléphones obiles, Jeux, Maisons intelligentes, Wearables, PC, Sécurité, Soins de santé, Sports et fitness, Industriel, etc.	Maison intelligente, Surveillance médicale, Capteurs IA environnements, Électronique grand public	Automatisation de la fabrication de compteurs électriques, Terminaux de point de vente au détail, Ville intelligente	Vidéoconférence, GPS et TV mobile (3G), Appareils portables, Applications haut débit et TV mobile (4G)

1.4 Approches Récentes

L'un des défis les plus importants consiste à décharger des données des appareils IdO vers les passerelles pour traiter ces données ou les envoyer vers l'infonuage à distant. Ainsi, envoyer des données aux passerelles de manière dynamique, fiable et robuste est un défi intéressant.

Wang, Xie, Liu & Zhu (2018) ont proposé une couche réseau appelée "Exploiting Cross-Technology (ECT)" pour interconnecter différents protocoles IdO en utilisant la technologie croisée et sa transmission de données simultanée du nœud ZigBee à un autre nœud ZigBee et appareil WiFi. Cette couche vise à réduire le délai de livraison sous la contrainte que les priorités des nœuds ZigBee changent dynamiquement.

Dans leur méthode, les auteurs ont mis l'accent sur les données alors que dans les méthodes précédentes qui utilisent la technologie croisée, les capteurs et le délai pour différents services ainsi que la consommation d'énergie des appareils IdO sont plus importants. Chacune de ces méthodes a son propre problème, il est donc préférable d'analyser les données de chaque appareil et ensuite de prendre une décision sur les priorités des paquets. Dans Wang *et al.* (2018), les données doivent être envoyées à un serveur ou un contrôleur pour les analyser et prendre des décisions, ce qui augmente le délai.

Pour résoudre les problèmes susmentionnés, dans cette approche, les données importantes et les données brutes qui sont collectées par les capteurs sont envoyées à différentes interfaces de passerelle. Les données importantes sont envoyées par le Wi-Fi qui a le débit de données plus élevé. Le contrôleur peut décider la priorité des données et leur délai respectif, ensuite il envoie ses instructions aux appareils IdO pour envoyer des données brutes vers leur interface ZigBee.

Pour mettre en œuvre la méthode proposée, les auteurs ont analysé le réseau, et calculé le délai dans le réseau avec différents modes, puis, essayé de minimiser ce délai en utilisant des techniques d'optimisation.

À l'aide de ce modèle, chaque nœud ZigBee utilise une approche de programmation dynamique pour découvrir sa séquence de transfert optimale. Puisque des liens ZigBee sont peu fiables, il est possible que les données n'atteignent pas la destination. Alors, il faut définir des schémas de retransmission dans différents scénarios qui introduisent des délais additionnels dans la livraison des données.

Pour calculer les délais de livraison attendus et le taux de livraison attendu, les auteurs ont utilisé la qualité du lien entre les différents nœuds et la probabilité de réception des données dans différents scénarios ainsi que le délai d'attente d'un nœud pour passer à l'état de réception. Pour optimiser la séquence de transfert, ils ont défini des limites pour différentes métriques introduites. Ensuite, le serveur place ces priorités dans une table nommée carte de priorité. Les points d'accès envoient cette carte de priorité aux nœuds ZigBee. Cette carte de priorité réduit

la surcharge du réseau qui est générée par les nœuds ZigBee lorsqu'ils essaient d'envoyer des informations supplémentaires aux points d'accès.

Enfin, les auteurs ont proposé le transfert de données simultanée pour réduire davantage le délai de livraison en utilisant les points d'accès WiFi pour livrer les données brutes au serveur si leur transmission de données importantes a été terminée. Leur résultat montre que leur approche réduit considérablement le délai de livraison des données, qui est 5,4 fois inférieur à la méthode de transfert dynamique basé sur le commutateur (DSF) en termes de cycle d'utilisation. En ce qui concerne les paquets de données générés, le délai de livraison du schéma proposé n'est pas comparable au DSF dans des scénarios de trafic élevé et alors que la probabilité de congestion augmente.

Comme mentionné ci-dessus, les appareils IdO s'interconnectent avec 2 types de connexion différents. Dans Uddin *et al.* (2018), les auteurs ont proposé une approche pour profiter des avantages des deux méthodes et éliminer les inconvénients. Leur méthode proposée de traitement multiprotocole à la périphérie pour l'IdO (Muppet) utilise une structure basée sur SDN et un commutateur contrôlé par SDN nommé BLESS. La méthode SDN permet de bénéficier des avantages des connexions peer-to-peer. Dans cette méthode BLESS est un commutateur P-4 qui élimine les inconvénients des connexions centrées sur l'information nuagique en implémentant à la périphérie du réseau pour réduire autant que possible la latence. Cette solution est illustrée dans la figure 1.3. Cette figure montre comment l'approche Muppet utilise différents aspects des approches "peer-to-peer" et "cloud-centric" en utilisant différents éléments tels que : les nœuds de périphérie, le contrôleur SDN et les commutateurs basés sur p-4, pour tirer parti de tous leurs avantages. Cette approche peut être utilisée dans différents cas d'utilisation, tels que :

1. **Intra-Protocol** : Lorsque les appareils utilisent le même protocole de couche de liaison sous-jacent.
2. **Inter-protocole** : Lorsque deux protocoles de couche de liaison différents interagissent.

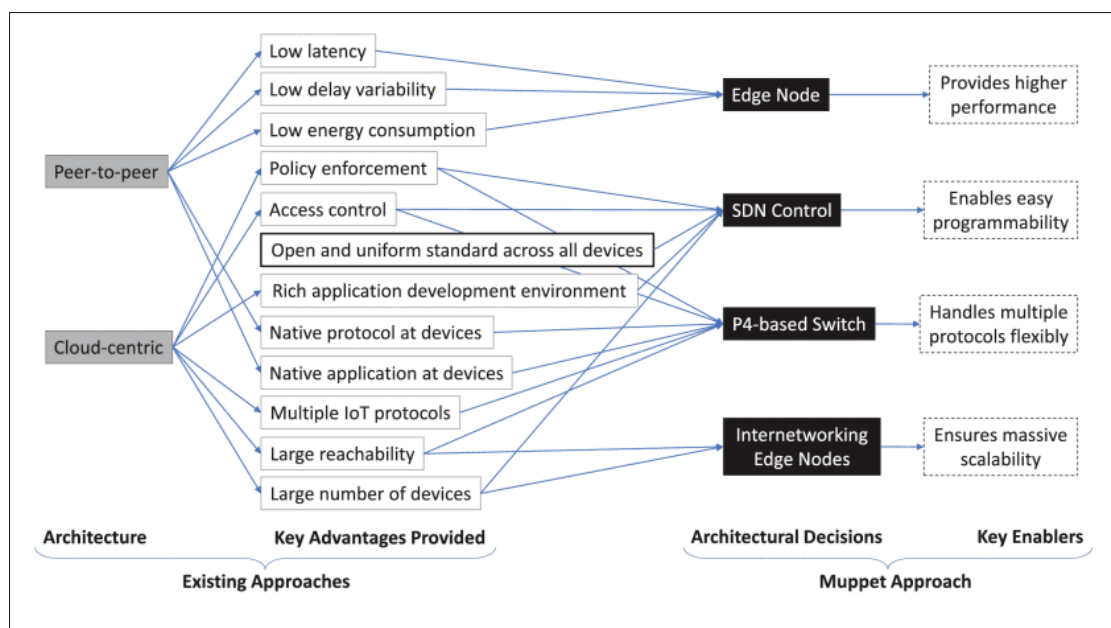


Figure 1.3 Justification de l'Architecture Muppet
Tirée de Uddin et al. (2018)

Les auteurs ont développé un commutateur logiciel programmable (*Muppet*) comme intermédiaire pour capturer tout le trafic afin de permettre une automatisation des services en réseau basée sur des règles. L'architecture de Muppet a trois niveaux :

1. **Plan de Gestion :** Le rôle le plus important de ce plan est d'interconnecter différents appareils IdO hétérogènes en créant une adresse public unique au monde pour les paquets tout en conservant leur propre adresse et protocoles, en maintenant les adresses uniques dans chaque port. Pour activer l'application de la politique au niveau de la couche liaison, les auteurs mettent le contrôleur Muppet dans la boucle pour certaines transactions de demande/réponse spécifiques. Ce plan se compose de trois modules différents pour effectuer les tâches susmentionnées :

- module de gestion BLE ;
- module de gestion ZigBee ;
- module de gestion IP ;

2. **Plan de Données :** Ce plan reçoit les paquets de différents appareils IdO à partir de différents protocoles, ports et interfaces IdO, puis les analyse en fonction des politiques définies pour différents protocoles et décide des actions. Ce plan a quatre composants clés :
 - analyse de paquets ;
 - flux de contrôle ;
 - actions ;
 - recirculation des paquets ;
3. **Plan de Contrôle :** Dans cette approche Muppet, on a besoin d'une unité centrale pour prendre des décisions sur toutes les différentes politiques et caractéristiques des appareils, applications et services IdO dans le réseau. Cette unité peut être physique ou logique. Dans cette approche pour réduire ce problème, les auteurs ont proposé de garder le contrôleur temporairement hors du circuit d'automatisation normal.

Cette méthode présente de nombreux avantages pour les réseaux IdO traditionnels, car elle surmonte l'accessibilité limitée des protocoles IdO natifs et forme une infrastructure IdO à large spectre en fournissant une couche de service logique commune aux protocoles IdO. Comme Muppet n'envoie que les notifications au-dessus d'un seuil défini, cette approche réduit significativement la consommation d'énergie par rapport au modèle infonuagique et peer-to-peer. De plus, cette approche présente moins de délai que le modèle infonuagique avec des connexions WiFi et LTE. Les auteurs ont aussi testé l'évolutivité de leur approche en émulant les ports et le débit et en calculant l'utilisation des ressources. Leurs résultats montrent que la tâche de chiffrement/déchiffrement impose une surcharge au système mais cette surcharge augmente linéairement avec le nombre de ports.

Dans Farhan *et al.* (2018), les auteurs ont proposé un nouveau schéma pour optimiser la planification des paquets envoyés à la passerelle à partir des appareils ou capteurs IdO tout en utilisant la route optimale pour envoyer ces données. La nouveauté de leur approche réside dans l'optimisation conjointe du réseau avec un objectif de conservation de l'énergie et un schéma de gestion des ressources tout en considérant les approches de routage. Ils définissent l'itinéraire

optimal comme le chemin le plus court vers la passerelle afin que la consommation d'énergie de l'appareil diminue. Dans leur approche, ils ont proposé un nouveau schéma de routage chemin le plus court et moins de liens (SPLL) où de nombreux objectifs de conception tels qu'un faible délai, une faible consommation d'énergie, un débit élevé et une variance limitée de la qualité de la connexion sont considérés comme un équilibre de charge. Leur approche améliore l'équilibre de charge en transférant des données aux nœuds qui ont moins de nœuds voisins.

Les auteurs ont utilisé le modèle de file d'attente M/M/1 pour calculer le taux de service et le taux d'arrivée des paquets provenant des nœuds. Ils ont utilisé un algorithme d'ordonnancement des messages Long Hop (LH) qui sert d'abord les paquets provenant d'une plus longue distance. Les paquets acheminés utilisent des nœuds Cluster Head (CH) jusqu'à leur récepteur final. Pour le dimensionnement de la mémoire tampon du réseau, ils ont utilisé la règle empirique car les flux de chaque CH sont faibles.

Cependant, leur méthode n'est pas efficace dans les réseaux à petite échelle. Elle fonctionne bien dans les grands réseaux en équilibrant la charge de trafic et en réduisant l'utilisation de nombreux nœuds intermédiaires pour fournir les données à la BS.

Comme cette méthode a un nombre moyen de sauts et une distance de transmission limitée, surtout pour les grandes zones, elle présente une latence moyenne légèrement élevée. Ceci réduit le délai d'expiration des paquets et la retransmission des paquets et améliore le transfert de données qui pourrait respecter la QoS de certains services sensibles au délai. Par contre, cette méthode n'est pas réellement optimale pour les réseaux qui ont un délai de service limité car cette méthode sert en premier les paquets qui viennent de plus grande distance. En ce qui concerne le débit, cette méthode améliore le débit du réseau car elle donne la priorité aux paquets avec une distance plus longue. Cependant, le risque de perte de paquets augmente.

Dans Qin, Chen, Cao, Zeng, Li & Peng (2018), les auteurs ont proposé un déchargement de calcul conjoint et une planification multi-utilisateurs dans le système NB-IoT, comme le montre la figure 1.4. Ici, leur problème de déchargement cherche à transférer les données vers le serveur "Mobile Edge Computing (MEC)" pour décharger la tâche de calcul à un appareil central puissant

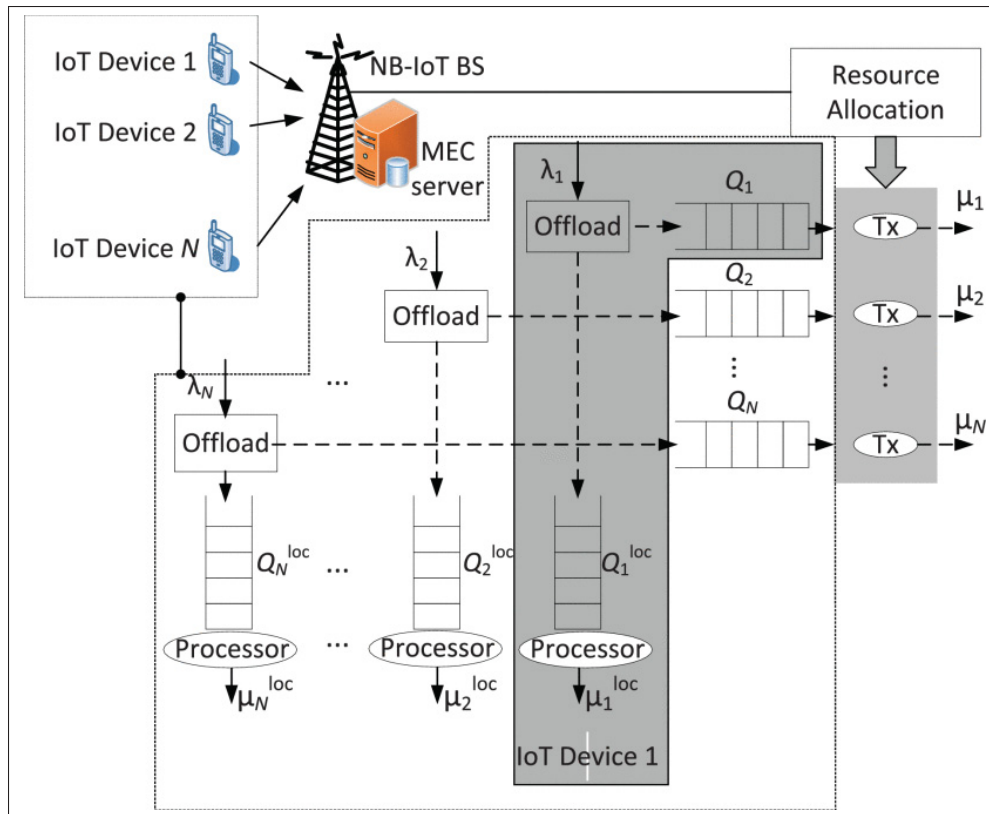


Figure 1.4 Système MEC NB-IoT et Modèle de File d'Attente
Tirée de Lei et al. (2019)

et réduire le délai de traitement comme indiqué en haut à gauche de la figure 1.4. Cependant, le déchargement des données sur le serveur pose certains problèmes, tels que l'augmentation du délai de transmission ou augmentation de l'électricité consommée de l'appareil IdO. Ensuite, ils planifient différents utilisateurs dans différentes files d'attente pour envoyer leurs données et allouer les ressources appropriées à ces utilisateurs, comme vous pouvez le voir sur la figure 1.4. Les auteurs proposent donc un modèle d'optimisation à l'aide du modèle processus de décision markovien en temps continu (CTMDP) pour résoudre ces contraintes. La raison pour laquelle ils ont utilisé le processus de décision de Markov en temps continu au lieu du processus de décision de Markov en temps discret est que le dernier a besoin d'informations supplémentaires sur les états du système et la mémorisation des états précédents. Il existe quelques facteurs importants qui affectent la décision de décharger les données sur le serveur comme :

- emplacement du serveur ;
- capacités locales de calcul et de stockage ;
- alimentation de l'appareil IdO ;
- type d'application ;
- allocation des ressources radio du NB-IoT ;

La décision d'allocation des ressources radio dans NB-IoT aura des effets importants sur le délai de transmission des données et la consommation d'énergie, ce qui affectera les performances de déchargement. Chaque appareil IdO contient des modules tels que la détection, la communication, la traitement et le stockage. Chaque appareil a deux files d'attente. L'une consiste à envoyer les données à traiter sur le serveur MEC, et l'autre à traiter les données sur l'appareil IdO lui-même localement. Les auteurs prennent en considération la partie communication et calcul des appareils IdO dans leur modèle. Dans NB-IoT, le contrôle des ressources radio a trois fonctions :

1. **Adaptation du Lien** : Cette fonction adapte le niveau de Modulation et de Codage (*MCS*) et le nombre de répétition pour s'assurer que le taux d'erreur de bloc satisfait le seuil. Ici, il n'est effectué que par les effets d'atténuation à grande échelle.
2. **Scheduling** : Dans NB-IoT, le problème de planification comporte deux parties, la liaison montante et la liaison descendante. La liaison descendante n'a besoin que d'allouer des ressources de domaine temporel tandis que la liaison montante a besoin à la fois de ressources de domaine temporel et fréquentiel. L'ordonnanceur est responsable de déterminer les configurations RU en plus d'ordonner la séquence de transmission des paquets et des trames. Le débit de transmission des données d'un appareil IdO est considéré comme une valeur constante.
3. **Contrôle de Puissance** : Pour réduire la complexité, NB-IoT utilise l'approche de contrôle de puissance en boucle ouverte, ce qui signifie que l'appareil IdO décide sa puissance. Cette décision est faite en considérant sa distance à la BS.

Cette méthode peut être mise en œuvre pour un grand nombre d'appareils IdO. Sa performance est flexible, c'est à dire que nous pouvons attribuer à l'appareil IdO une application qui est sensible à la latence ou à l'efficacité énergétique. Enfin, cette méthode est basée sur un modèle de tâche stochastique, autrement dit, elle considère l'impact de la tâche actuelle sur les performances futures du réseau en fonction de son délai et de sa consommation d'énergie.

Dans Lei *et al.* (2019), les auteurs se concentrent sur la consommation d'énergie de l'interface Wi-Fi et son temps active. Ils ont utilisé une interface ZigBee ou Bluetooth co-localisée pour résoudre l'écoute inactive haute puissance de l'interface WiFi tout en résolvant le délai de livraison élevé en l'absence d'interface Wi-Fi. Ils ont proposé une programmation DIPS (Dual-Interface Dual-Pipeline) pour les connexions multi-sauts.

Dans leur méthode, les interfaces ZigBee sont prévues pour envoyer des données de contrôle à la passerelle, afin de réveiller les interfaces Wi-Fi pour envoyer des données à la passerelle. Les interfaces Wi-Fi créent une chaîne permettant d'envoyer les données à la passerelle pour une livraison de données multi-sauts rapide et économe en énergie. Les interfaces Wi-Fi peuvent aider les interfaces ZigBee peu fiables.

L'objectif principal de ce modèle est de programmer le cycle d'utilisation des interfaces pour réduire la consommation d'énergie en utilisant des interfaces Wi-Fi rapides pour transmettre des données en respectant le délai d'échéance. Pour atteindre cet objectif, les auteurs ont conçu un cadre de système comme le montre la figure 1.5. Dans ce schéma, ils ont considéré un flux de données entre deux nœuds. Ce flux a une limite de délai de bout en bout pour respecter la QoS. Ainsi, cette méthode peut minimiser la consommation d'énergie tout en respectant la limitation du délai de bout en bout.

Cette méthode utilise un temps de réveil réactif. Elle a donc l'avantage de pouvoir s'adapter au trafic dynamique, ce qui la rend applicable à une large gamme d'applications IdO. Contrairement aux technologies croisées qui ne sont pas bonnes pour les communications multi-sauts, cette méthode est conçue pour minimiser la latence du réseau.

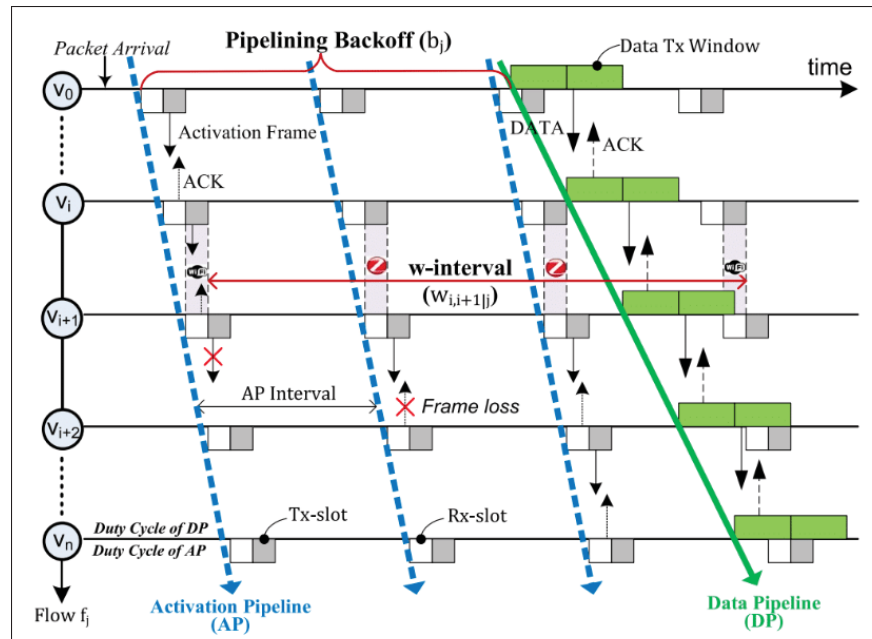


Figure 1.5 Cadre du système
Tirée de Qin et al. (2018)

Amor *et al.* (2019) ont proposé une assignation de flux optimisée dans les passerelles IdO multi-interfaces. Comme nous le savons, les appareils IdO multi-interfaces améliorent la capacité du réseau. Ainsi, en utilisant ces interfaces efficacement, les performances du réseau augmentent. Le défi dans les appareils IdO multi-interfaces est que les données peuvent ne pas être allouées à la meilleure interface en fonction de son contenu et de ses caractéristiques, ce qui entraîne une congestion. Il est donc nécessaire d'avoir une assignation de flux qui décide d'allouer quelles données à quelle interface par rapport à leur échéance de livraison. Cette méthode a proposé une assignation de flux qui maximise toute la quantité de données acceptées par la passerelle dans le réseau IdO. L'architecture proposée d'architecture comporte deux modules comme le montre la figure 1.6.

1. **Module de Surveillance :** Dans ce module, la passerelle collecte des informations sur les différents appareils IdO qui y sont connectés, leurs services et applications pris en charge, ainsi que leur bande passante, débit de données et délai de livraison respectifs. Après avoir collecté ces informations, il fournit une vue du réseau et des appareils connectés.

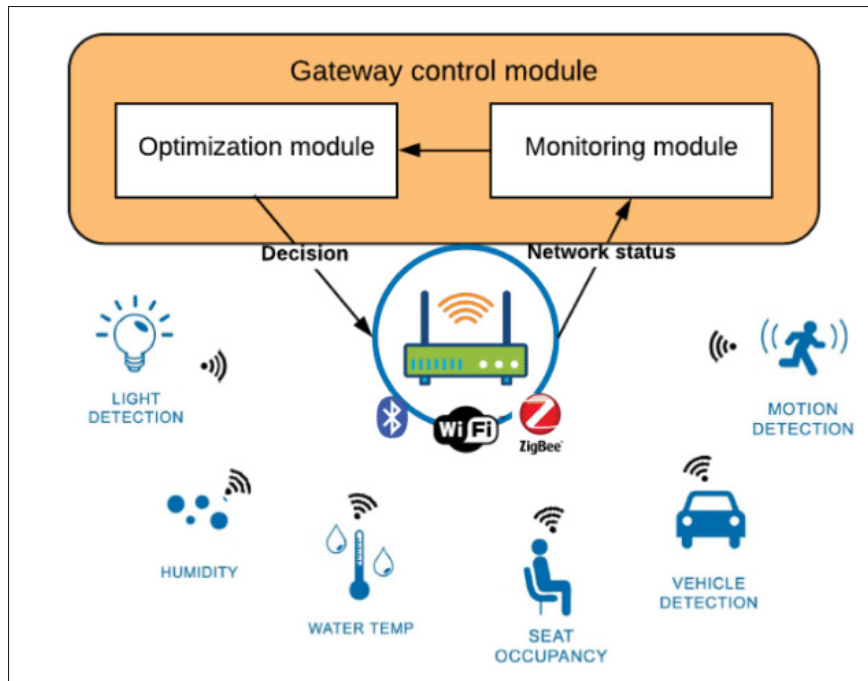


Figure 1.6 Architecture du Système
Tirée de Amor et al. (2019)

2. **Module d'Optimisation :** L'objectif de ce module est d'optimiser l'assignation des flux de manière à maximiser la quantité totale de données affectées aux passerelles IdO, mais il doit respecter les délais des applications et services respectifs et également respecter l'énergie, la capacité de l'appareil IdO et de données de la passerelle IdO. Pour optimiser le problème d'assignation de flux, les auteurs ont supposé que l'algorithme affecte un flux à une interface réseau qui a la plus petite capacité suffisante parmi les interfaces disponibles.

Cette méthode considère la capacité de la passerelle et l'assignation des flux aux différentes interfaces de cette passerelle comme un problème. En plus, elle prend en considération les délais respectifs des différents flux. Cependant, cette méthode simplement prend en compte la capacité énergétique de chaque appareil IdO et n'a pas optimisé la consommation d'énergie du réseau IdO.

1.5 Discussion

Dans cette section, nous allons résumer des travaux connexes présentés dans la section précédente qui traitent la gestion des flux, l'allocation des ressources et la planification des paquets dans les réseaux IdO.

Tableau 1.3 Objectifs des approches

Référence	Équilibrage de Charge	Qualité du Lien	Hétérogénéité	Durée de Vie du Réseau
Farhan <i>et al.</i> (2018)	✓	-	-	✓
Wang <i>et al.</i> (2018)	-	✓	✓	Partiel
Uddin <i>et al.</i> (2018)	✓	-	✓	✓
Lei <i>et al.</i> (2019)	Partiel	-	-	✓
Qin <i>et al.</i> (2018)	✓	✓	✓	✓
Amor <i>et al.</i> (2019)	✓	-	✓	-

Il existe de nombreux défis en termes de la QoS et de gestion des ressources dans les réseaux IdO hétérogène, tels que la conception et le respect du délai requis des applications, la consommation d'énergie, les contraintes de ressources, le débit, l'utilisation de la bande passante, la tolérance aux pannes et les intercouches de couplage pour un service IdO en temps réel. Les solutions sont différentes en termes de paramètres tels que les approches de routage multi-chemins, d'agrégation de données et d'allocation de ressources. Le tableau 1.3 montre la synthèse des méthodes étudiées en fonction de leur objectif.

Dans le tableau 1.4, nous présentons les méthodes susmentionnées et comparons leurs objectifs et leurs tendances.

Toutes les études surmentionnées visent à réduire la consommation d'énergie du réseau IdO tout en diminuant le délai moyen des flux dans le réseau. Certains d'entre eux ont utilisé la consommation d'énergie comme objectif principal et certains ont utilisé un délai et une latence moyens. Ces études ont utilisé des techniques d'optimisation différentes mais aucune n'a considéré la capacité de données de la passerelle sauf Amor *et al.* (2019), qui a proposé une assignation de flux pour utiliser la capacité maximale de la passerelle par rapport au temps

Tableau 1.4 Caractéristiques des méthodes récentes d'optimisation de réseau IdO

Référence	Objectif(s)	Paramètres de QoS Pris en Compte	Inconvénient(s)
Farhan et al. (2018) Vers l'informatique verte pour l'IdO : approche de planification des chemins et des messages orientée énergie	Minimiser la consommation d'énergie et gérer les ressources en utilisant des approches de routage	- Retard et débit en utilisant l'équilibrage de charge - Perte de paquets en utilisant un schéma multi-chemins	- Nécessite un temps supplémentaire pour former des clusters - La fiabilité et les liens ne sont pas pris en compte
Wang et al. (2018) ECT : Exploiter la transmission simultanée inter-technologies pour réduire les délais de livraison de paquets dans les réseaux IdO	Gestion des ressources par agrégation de données avec séquence de transfert et routage réseau optimaux	- Résoudre la congestion du réseau - Augmenter la fiabilité en utilisant un remplacement pour les liens ZigBee peu fiables	La consommation d'énergie n'est pas validée lors de la configuration du réseau
Uddin et al. (2018) Commutation de périphérie multiprotocole basée sur SDN pour l'automatisation des services IdO	Résoudre l'hétérogénéité des réseaux IdO et l'utilisation du SDN centrée sur l'info nuagique avec les avantages peer-to-peer	- Augmenter l'évolutivité - Diminuer le délai et la latence de bout en bout	Aucune simulation pour les réseaux à grande échelle
Lei et al. (2019) Déchargement de calcul conjoint et planification multi-utilisateurs	contrôlez la congestion dans le calcul de périphérie ou l'info nuagique	- Respecter le délai - Augmenter la durée de vie du réseau - Augmenter la fiabilité	Ne prendre pas en compte l'arrivée de paquets provenant de plusieurs sources
Qin et al. (2018) DIPS : Planification à double interface et à double pipeline pour une communication multi-sauts écoénergétique dans l'IdO	Schéma de planification au profit de l'agrégation des données et du contrôle de la congestion avec la gestion des flux	- Diminuer la congestion, la latence et la perte de paquets	Dans la gestion des flux, la capacité du réseau n'est pas prise en compte
Amor et al. (2019) Attribution de flux optimisée dans une passerelle IdO à interface multiprotocole	Résoudre le problème de congestion et augmenter la capacité du réseau	- Respecter le délai des prestations - Augmenter la capacité du réseau tout en respectant la QoS requise	Ne pas tenir compte de la qualité de la liaison et de la consommation d'énergie

limite des flux. Par contre, cette étude n'a envisagé qu'une communication à un seul saut avec une seule passerelle.

Selon les avantages et les inconvénients mentionnés dans le tableau 1.4, nous proposons une direction de recherche pour exploiter les avantages de toutes ces recherches afin de minimiser la consommation de l'énergie et minimiser le délai moyen dans le réseau tout en utilisant la capacité maximale des passerelles dans les réseaux IdO.

Pour atteindre cet objectif, nous pouvons réutiliser le mécanisme d'attribution de flux proposé dans Amor *et al.* (2019) en développant son mécanisme pour un réseau multi-passerelles multi-sauts. Nous avons donc besoin d'un routage dynamique et d'une méthode de planification dynamique pour livrer les flux à leurs destinations. Comme dans Uddin *et al.* (2018), nous pouvons implémenter un algorithme basé sur SDN sur nos passerelles pour bénéficier du "Edge Computing" et minimiser le trafic dans le cœur de réseau, ce qui réduit les congestions et les délais dans le réseau. Comme dans Qin *et al.* (2018), nous pouvons utiliser différents types de schémas de pipeline à double interface pour atteindre nos objectifs mentionnés.

1.6 Conclusion

Les solutions existantes ont étudiées divers aspects opérationnels des réseaux IdO, notamment l'ajustement des cycles d'utilisation, les schémas d'évitement des collisions/encombrements, les mécanismes de gestion du temps de veille des appareils en éteignant les radios ou en augmentant le temps d'attente, la planification efficace des ressources radio, la sélection intelligente d'interfaces radio hétérogènes, etc. Dans la section suivante, nous présentons notre description du système IdO qui a été mentionné dans le contexte et notre solution pour l'assignation de flux dans les réseaux IdO multiprotocoles tout en respectant les échéances respectives des flux et la consommation d'énergie du réseau.

CHAPITRE 2

MÉTHODOLOGIE

Dans cette section, nous présenterons notre système IdO et notre approche et solution pour résoudre le problème de recherche qui est l'assignation de flux multiprotocole dynamique dans le réseau IdO intelligent. Ensuite, nous formulerons le modèle mathématique du problème qui est une fonction d'optimisation visant à maximiser la quantité de données acceptées par les passerelles du réseau IdO de la maison intelligente tout en respectant la QoS des appareils et des flux. Enfin, nous étudierons la solution proposée, OFAP-MIG.

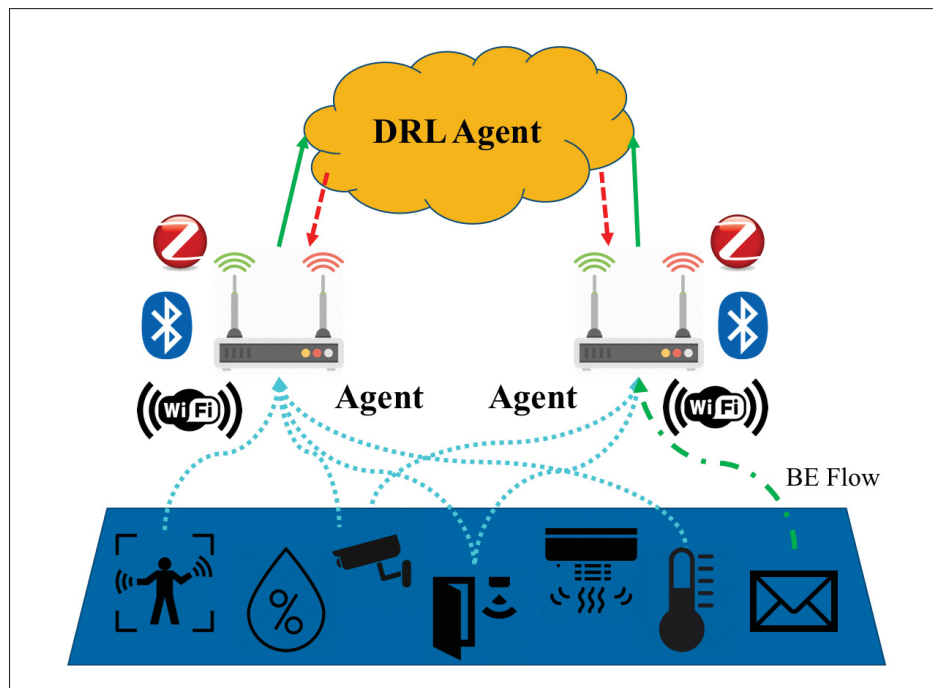


Figure 2.1 Architecture du Système

2.1 Description du Système

Nous considérons un système IdO composé de plusieurs passerelles avec plusieurs interfaces telles que WiFi, Bluetooth et Zigbee, etc. Les appareils IdO transmettent des données aux passerelles, qui sont connectées à l'infonuage, comme le montre la figure 2.1.

Contrairement aux travaux présentés dans Amor *et al.* (2019), nous supposons que les flux peuvent être répartis entre différentes interfaces d'appareils IdO. Les passerelles peuvent également servir d'autres trafics que ceux envoyés par les appareils IdO. En d'autres termes, "Best Effort (BE)" peut être envoyé via les passerelles vers Internet. Nous supposons que les appareils IdO sont fixés dans leurs locaux. Ces appareils IdO envoient des données discrètes dans des créneaux horaires définis qui sont déjà connus selon les configurations des capteurs de maison intelligente. S'il n'y a pas assez de bande passante dans une interface donnée, ou si le coût d'utilisation de différentes interfaces s'avère avantageux, une passerelle peut prendre la décision de répartir les flux sur plusieurs interfaces. Pour modéliser la charge imposée au système d'exploitation de l'appareil pour gérer de telles divisions, nous introduisons un délai fixe et l'appelons activation du délai d'interface. Ceci est utilisé comme paramètre pour évaluer le nombre de divisions (Uddin *et al.*, 2018).

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. Il est important de mentionner que nous ne prenons pas en considération les interférences entre les différents protocoles de communication. De plus, nous supposons que le délai entre les agents de périphérie et l'infonuage est zéro.

Le but de cette recherche est d'attribuer un ensemble de flux définis et transmis dans le même intervalle de temps aux interfaces de nos passerelles tout en maximisant la quantité de données valides reçues par nos passerelles.

2.1.1 Agent

L'agent est implémenté à la fois sur l'infonuage et sur chaque passerelle. Chaque agent surveille les flux connectés, et les affecte de manière optimale aux interfaces. L'agent résidant sur l'infonuage calcule l'assignation de flux optimale globale pour l'ensemble du réseau et met à jour les

agents s'exécutant sur les passerelles. Cette architecture réduit la complexité de calcul sur les passerelles qui ont une capacité de ressources de calcul limitée. Le module agent est composé de deux modules différents comme ci-dessous :

2.1.1.1 Module de Surveillance

Ce module collecte des informations en temps réel sur les appareils IdO et utilise la méthode push/fetch pour envoyer ces mises à jour au module d'optimisation afin d'affecter de manière optimale les flux aux interfaces des passerelles. Les informations requises incluent les appareils IdO, leurs débits de données, leurs services et leurs délais respectifs. 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.

2.1.1.2 Module d'Optimisation

Ce module est chargé de sélectionner la meilleure interface réseau répondant aux exigences du flux. Le module d'optimisation utilise cet information collectée par le module de surveillance pour affecter les flux aux interfaces disponibles des passerelles.

2.1.2 Mécanisme de Commutation

Dans Amor *et al.* (2019), les auteurs ont introduit un protocole qui permet un transfert transparent entre les interfaces réseau. Contrairement à ce travail, un flux de données peut être fractionné dans ce mémoire. Par conséquent, le délai de commutation est la somme :

- du temps d'établissement d'une connexion d'une ou plusieurs interfaces selon la décision de l'algorithme d'optimisation.
- du délai de commutation lorsque le module d'optimisation affecte le flux à une interface différente de celle à laquelle l'appareil est actuellement connecté.

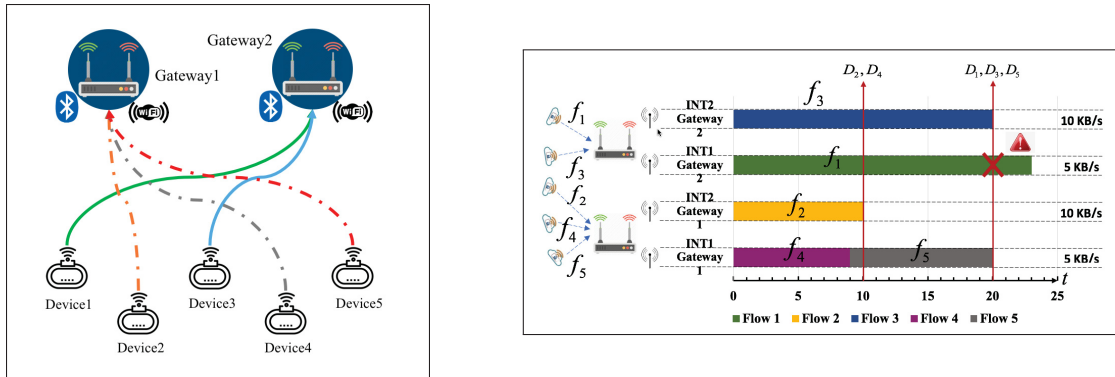


Figure 2.2 Exemple de Problème d'Assignment de Flux

2.1.3 Exemple de Motivation

Pour illustrer la sévérité et l'importance d'une méthode d'optimisation pour affecter dynamiquement des flux aux interfaces de passerelle, considérons un exemple dans la figure 2.2. Dans ce scénario, il y a cinq capteurs e-santé qui sont connectés à deux passerelles multiprotocoles d'une maison intelligente. Lorsque les flux sont envoyés dans le même intervalle de temps et affectés à la même interface, le flux avec l'échéance la plus proche est programmé pour être transmis en premier. Notre objectif est de maximiser la quantité de données valides, qui est égale à la quantité totale de données acceptées par les passerelles divisée par la quantité totale de données générées, tout en respectant les délais. Les flux avec leurs échéances sont résumés dans le tableau 2.1.

La figure 2.2 montre l'architecture de l'exemple qui comprend les composants suivants :

- deux passerelles IdO ayant chacune deux interfaces qui sont WiFi et BLE avec respectivement 10 Ko/s et 5 Ko/s de débits de transmission.
- cinq appareils IdO qui sont équipés de deux interfaces chacun comme passerelles. Chaque appareil IdO envoie un seul flux.

À $t = 0$ chaque appareil envoie son flux respectif aux passerelles. Tout flux doit être reçu par les passerelles dans son délai.

Tableau 2.1 Saisie de scénario

Flux	f_1	f_2	f_3	f_4	f_5
Taille des Flux (S_i) (in KB)	230	50	100	90	110
Délai de Transmission ($\beta_{i,j,k}$) (WiFi) (Secondes)	23	5	10	9	11
Délai de Transmission ($\beta_{i,j,k}$) (BLE) (Secondes)	46	10	20	18	22
Échéance de Flux (D_i) (Secondes)	20	10	20	10	20

Considérez l'assignation de flux pour l'appareil 1 dans la figure 2.2. L'appareil envoie le flux f_1 avec une taille de 230 Ko, ce qui prend 23 s pour l'envoyer via l'interface WiFi et 46 s via BLE. Le temps limite de ce flux est de 20s. Ainsi, si l'on affecte le flux à l'interface BLE de la passerelle 2, comme dans la figure 2.2, ou à l'interface WiFi de cette passerelle, son échéance sera dépassée. En conséquence, la passerelle supprime f_1 , ce qui peut être dangereux si elle contient des données patient critiques. Mais si on fractionne ce flux et qu'on l'envoie via le BLE de la passerelle 2 et le WiFi de la passerelle 1, la quantité de données valides acceptées par les passerelles du réseau augmente tout en respectant le temps limite de tous les flux.

Ici, les flux sont affectés aux interfaces de manière statique car il n'y a pas de mécanisme dynamique qui pourrait optimiser l'assignation de flux entre les interfaces. De plus, comme le trafic peut varier considérablement dans le temps, l'assignation et la réassignation des flux aux interfaces disponibles des passerelles doivent se faire de manière dynamique.

2.2 Formulation du Problème

Nous avons appelé le problème OFAP-MIG. L'optimisation de l'assignation des flux de plusieurs passerelles IdO dans lesquelles les flux peuvent être répartir entre les différentes interfaces disponibles d'un appareil IdO. De plus, le trafic au mieux (BE) peut également être envoyé via les passerelles pour accéder à Internet, à chaque fois qu'il y a de la capacité disponible sur les interfaces.

Nous désignons l'ensemble des appareils IdO qui ont plusieurs interfaces telles que WiFi, BLE et Zigbee par $\mathcal{I}$ et l'ensemble des flux comprenant les flux de données au mieux par $\mathcal{F}$, et définissons comme ci-dessous :

$$\mathcal{F} = \{f_i\} \quad \forall i \in \mathcal{I}$$

Où f_i est un ensemble de flux envoyés par un ensemble d'appareils IdO $\mathcal{I}$ et la taille de ces flux est définie en fonction des exigences de service dans la maison intelligente. La taille du flux f_i est notée par S_i . Nous supposons que chaque appareil envoie un seul flux. On note C_i la capacité de la batterie des appareils IdO et $E_{j,k}$ la consommation énergétique d'un appareil IdO pour envoyer le flux f_i à la passerelle k via l'interface j . Nous dénotons l'ensemble des interfaces réseau disponibles dans notre réseau $\mathcal{J}$ et l'ensemble des passerelles par $\mathcal{K}$. Chaque interface j de la passerelle k a un débit de transmission $T_{j,k}$ et une capacité $R_{j,k}$. $A_{j,k}$ est la matrice binaire des interfaces dans les passerelles ; $a_{j,k} = 1$ signifie que l'interface j est disponible sur la passerelle k . $B_{i,j}$ est la matrice binaire des interfaces dans les objets connectés ; $b_{i,j} = 1$ signifie que l'interface j est disponible sur l'objet i . Nous supposons que les interfaces utilisent les mêmes schémas de modulation et de codage pour tous les utilisateurs finaux et définissons deux variables de décision telles que :

$$x_{i,j,k} = \text{the percentage of the flow } f_i \text{ assigned to the interface } j \text{ of gateway } k \quad (2.1)$$

2.2.1 Problème d'Optimisation

L'objectif d'OFAP-MIG est de maximiser la quantité de données valides acceptées par les passerelles du réseau IdO tout en respectant les contraintes d'intégrité, de délais, d'énergie et de capacité du réseau. Il est formulé comme suit :

$$\max \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} S_i x_{i,j,k} \quad (2.2)$$

2.2.2 Contraintes

La fonction d'optimisation est soumise aux contraintes suivantes :

2.2.2.1 Contrainte de type d'interface pour la connexion entre l'objet connecté et la passerelle

On doit s'assurer qu'un même type d'interface (e.g., WiFi, Zigbee, etc.) est utilisé pour la connexion entre d'un objet connecté et une passerelle.

$$B_{i,j} x_{i,j,k} = A_{j,k} x_{i,j,k}, \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \quad (2.3)$$

2.2.2.2 Intégrité du Flux

La somme des flottants de la variable de décision $y_{j,k}$ qui renvoie le pourcentage d'affectation doit être égale à "1". Cette contrainte consiste à garantir que lorsque nous divisons les flux et les envoyons sur plusieurs interfaces, ils doivent créer un seul flux du côté de la passerelle.

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} x_{i,j,k} B_{i,j} = 1, \quad \forall i \in \mathcal{I}. \quad (2.4)$$

2.2.2.3 Capacité Énergétique

Lorsque le flux f_i est transmis via l'interface j de la passerelle k , sa consommation d'énergie ne doit pas dépasser l'énergie disponible de la batterie de l'appareil C_i . La consommation énergétique de l'interface j de la passerelle k lors de l'envoi du flux f_i est le produit du délai de transmission et de la consommation énergétique par seconde $E_{j,k}$ de l'interface j de la passerelle k :

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \frac{S_i}{T_{j,k}} E_{j,k} x_{i,j,k} B_{i,j} \leq C_i, \quad \forall i \in \mathcal{I}. \quad (2.5)$$

2.2.2.4 Écheance des Flux

Chaque flux f_i doit être reçu dans son délai respectif D_i . Nous définissons trois délais pour chaque flux f_i : Délai de commutation $\theta_{i,j,k}$, Délai de transmission $\beta_{i,j,k}$ et Délai de mise en mémoire tampon $\alpha_{i,j,k}$.

1. **Délai de Commutation** : le délai de commutation est la somme du temps d'établissement d'une connexion ce qui peut aussi être le moment d'établir la connexion d'un appareil à plusieurs interfaces et du délai de commutation lorsque le module d'optimisation affecte le flux à une interface différente de l'interface à laquelle l'appareil est actuellement connecté et nous le notons $\theta_{i,j,k}$. Nous devons définir une autre variable binaire qui montre s'il y a du trafic entre l'interface j des objets connectés et l'interface j de la passerelle k . Nous désignons cette variable par $Y_{i,j,k}$ qui est "1" lorsqu'il y a du trafic et "0" lorsqu'il n'y a pas de trafic entre les interfaces des objets connectés et les interfaces des passerelles. Les équations suivantes montrent la relation entre les variables. Il convient de noter que M est un nombre constant qui est très grand :

$$Y_{i,j,k} \geq B_{i,j} x_{i,j,k} \quad (2.6)$$

$$Y_{i,j,k} \leq M * B_{i,j} x_{i,j,k} \quad (2.7)$$

2. **Délai de Transmission** : est égal à la taille du flux, S_i , divisé par le débit de transmission de l'interface de la passerelle k , $T_{j,k}$, comme suit :

$$\beta_{i,j,k} = \frac{S_i}{T_{j,k}} x_{i,j,k} B_{i,j}, \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}. \quad (2.8)$$

3. **Délai de Mise en Mémoire Tampon** : comprend deux termes différents : (a) le délai d'envoi de la quantité de données restantes dans le tampon de l'interface j de la passerelle k ($Q_{j,k}$), qui est déjà transmis au cours de la tranche horaire précédente. (b) Le deuxième terme est le délai d'envoi des flux qui sont programmés pour être envoyés en premier dans le même intervalle de temps avant f_i . Pour programmer les flux, nous supposons que les flux

avec l'échéance la plus proche sont programmés pour être transmis en premier lorsqu'ils sont affectés à la même interface. Par conséquent, $\alpha_{i,j,k}$ est défini comme suit :

$$\alpha_{i,j,k} = \frac{Q_{j,k}}{T_{j,k}} + \sum_{i' \in \mathcal{I}'} \frac{S_{i'} x_{i',j,k} B_{i,j}}{T_{j,k}}, \quad \forall i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}. \quad (2.9)$$

$\mathcal{I}'$ désigne l'ensemble d'appareils qui sont programmés pour envoyer leurs flux en premier car ils ont des échéances antérieures à f_i et on le définit comme ci-dessous :

$$\mathcal{I}' = \{i' \in \mathcal{I} \mid \{i' \neq i\} \text{ and } D_{i'} < D_i\} \quad (2.10)$$

Pour garantir la contrainte de délai qui signifie que lorsque nous envoyons le flux f_i via l'interface j de la passerelle k , la somme de tous les délais susmentionnés ne doit pas dépasser la date limite du flux D_i .

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \theta_{i,j,k} Y_{i,j,k} + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \beta_{i,j,k} x_{i,j,k} B_{i,j} + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \alpha_{i,j,k} x_{i,j,k} B_{i,j} \leq D_i, \quad \forall i \in \mathcal{I} \quad (2.11)$$

2.2.2.5 Capacité du Réseau

Chaque interface réseau a une capacité de traitement de données limitée. Il faut donc s'assurer que la quantité de données qui est allouée à l'interface j de la passerelle k ne dépasse pas la capacité de la passerelle $R_{j,k}$.

$$\sum_{i \in \mathcal{I}} S_i x_{i,j,k} B_{i,j} \leq R_{j,k}, \quad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}. \quad (2.12)$$

Nous avons listé les notations que nous avons utilisées dans le tableau 2.2.

Tableau 2.2 Liste des notations

Notation	Description
$\mathcal{F}$	Ensemble de flux
$\mathcal{I}$	Ensemble d'appareils IdO
$\mathcal{I}'$	Ensemble d'appareils programmés pour envoyer leurs flux en premier
$\mathcal{J}$	Ensemble d'interfaces
$\mathcal{K}$	Ensemble des passerelles
D_i	Échéance des flux f_i
S_i	Taille des flux f_i
C_i	Capacité de la batterie des appareils IdO
$E_{j,k}$	Consommation d'énergie par un appareil IdO pour envoyer le flux f_i
$T_{j,k}$	Taux de transmission de l'interface j
$R_{j,k}$	Capacité de l'interface j
$Q_{j,k}$	Données restantes dans le tampon de l'interface j
$\beta_{i,j,k}$	Délai de transmission
$\alpha_{i,j,k}$	Délai de mise en mémoire tampon
$\theta_{i,j,k}$	Délai de commutation
$x_{i,j,k}$	The percentage of the flow f_i assigned to the interface j of gateway k
$A_{j,k}$	Matrice binaire des interfaces des passerelles
$B_{i,j}$	Matrice binaire des interfaces dans les objets connectés
$Y_{i,j,k}$	Variable binaire de l'existence du trafic

2.3 Solutions Proposées

Dans cette section, nous présentons tout d'abord l'algorithme adopté pour résoudre le problème d'assignation de flux multiprotocole optimal dans le réseau IdO d'une maison intelligente. Ensuite, nous présenterons le travail précédent que nous avons utilisé comme base pour comparer nos résultats de la solution et qui est proposé par Amor *et al.* (2019) pour résoudre le problème d'assignation de flux pour une passerelle.

2.3.1 Complexité du Problème

Dans cette section, nous allons prouver que *l'assignation de flux optimisée de passerelle multi-interfaces (OFAP-MIG)* est un problème NP-complet.

Le *problème de partition (PP)* revient à déterminer si un ensemble de N entiers, de somme A peut être partitionné en deux sous-ensembles ayant chacun une somme de $A/2$. Le PP est un problème NP-Complet bien connu. Nous basons la preuve sur la construction d'une instance du problème à partir de n'importe quelle instance du PP.

Supposons que nous n'avons qu'une seule passerelle disponible sur le réseau IdO. Soit chaque élément de l'ensemble du PP est un flux de l'instance du problème OFAP-MIG et la valeur de chaque élément, la taille du flux S_i du flux correspondant f_i . De plus, supposons qu'il n'y a que deux interfaces ($\mathcal{J} = 2$), avec la capacité et le délai suivants :

$$R_j = A/2, \quad \forall j \in \{1, 2\}.$$

L'instance OFAP-MIG construite est réalisable, car la disponibilité totale des ressources sur les deux interfaces suffit à répondre à la demande totale de flux.

Considérons une solution de l'instance OFAP-MIG construite où aucune division ne se produit. Si une telle solution existe, alors chaque flux est affecté à une interface donc le délai d'envoi du flux sera de D_i . De plus, étant donné que tout flux divisé en deux interfaces donne un délai différent du premier en raison des trois délais différents mentionnés dans l'équation 2.9. Par conséquent, par la construction de l'instance, il devient évident qu'aucune valeur inférieure à D_i ne peut être atteinte. Nous devons répondre s'il existe ou non une solution pour laquelle le délai est au plus de la même valeur, dans notre cas D_i .

Quelle que soit la solution du problème, la taille des flux sur chaque interface sera de $A/2$. S'il n'y a pas de division des services dans une solution, alors leur assignation aux deux interfaces est une partition des entiers dans le PP avec une somme égale. Par conséquent, si la réponse à l'instance PP d'origine est positive et que nous mappons les flux aux éléments des sous-ensembles de solution, alors en affectant les flux aux interfaces, aucune division n'existera et le délai sera le mappage D_i aux éléments des sous-ensembles de solution. Ainsi, la réponse à l'instance est, *Oui*. Inversement, si la réponse à l'assignation est oui, alors il ne peut pas y avoir de flux fractionné,

donc l'assignation des flux aux deux interfaces est une solution PP valide. Par conséquent, résoudre le problème équivaut à résoudre une instance PP arbitraire.

2.3.2 DRL-OFAP-MIG

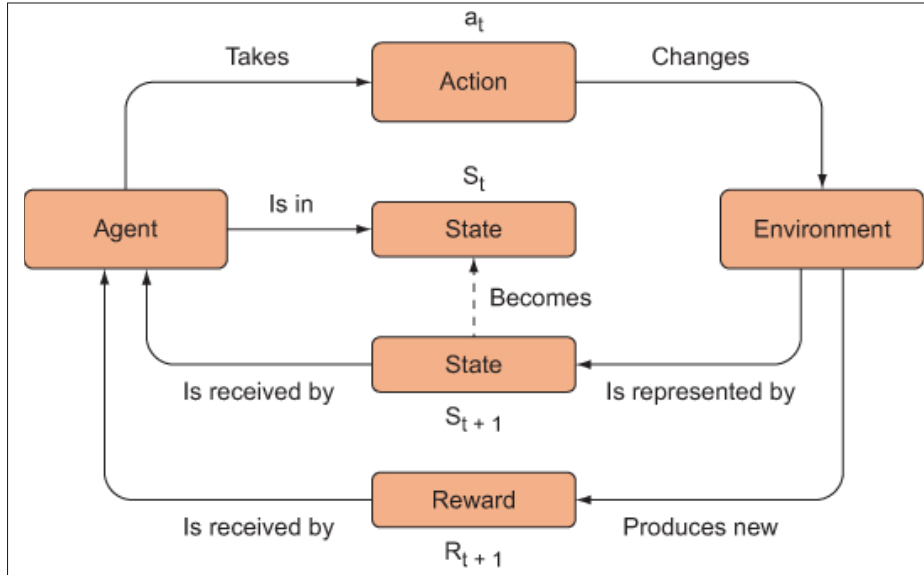


Figure 2.3 Cadre de RL

Le problème d'assignation de flux dynamique dans plusieurs passerelles IdO présente des dimensions élevées dans l'espace d'état qui est la matrice des données acceptées et la date limite des flux et un espace d'action compliqué qui est l'ensemble des flux actifs, l'ensemble des passerelles, l'ensemble des interfaces. De plus, en raison des contraintes qui ont été mentionnées dans la section 2.2.2, OFAP-MIG est une programmation mixte non entière (PMNE) avec une complexité de calcul élevée. Les raisons susmentionnées motivent l'adoption de DRL multi-agents pour notre problème, mais la question est pourquoi DRL.

En fait, certaines des premières méthodes RL qui recherchent des objectifs d'apprentissage n'impliquaient que le stockage des expériences dans une table de recherche et la mise à jour de cette table à chaque itération de l'algorithme, comme le montre la figure 2.3. L'idée était de laisser l'agent jouer dans l'environnement et voir ce qui s'est passé, et stocker ses expériences de ce qui s'est passé dans une sorte de base de données. Pour les petits états et les problèmes

d'action, ce qui signifie des environnements simples, cette méthode Q-table fonctionne en fait assez bien. Cependant, lorsque l'environnement devient plus compliqué, il est difficile d'utiliser et de trouver la stratégie optimale dans une énorme mémoire de table de valeurs Q. C'est là qu'intervient l'apprentissage en profondeur.

Un algorithme d'apprentissage en profondeur peut apprendre à faire abstraction des détails d'un environnement spécifique et à apprendre les caractéristiques importantes d'un état. Étant donné qu'un algorithme d'apprentissage en profondeur a un nombre fini de paramètres, nous pouvons l'utiliser pour compresser n'importe quel état possible en certaines données que nous pouvons traiter efficacement, puis utiliser cette nouvelle représentation pour prendre notre décision. Récemment, DNN a été introduit dans le cadre du Q-learning et le DQN est la méthode la plus adaptée pour traiter la question des grands espaces. Dans DQN, le DNN est utilisé pour représenter l'espace d'action et d'état. Le DQN utilise donc un réseau cible parallèlement au réseau en ligne pour stabiliser les performances globales du réseau.

La figure 2.4 montre l'architecture proposée, dans laquelle les agents DRL sont déployés sur des serveurs *cloud* et des passerelles IdO et leurs liens sont censés n'avoir aucune contrainte. Le but de l'agent sur l'info nuagique est de contrôler l'ensemble du réseau et de décider quelle solution globale est la meilleure. Afin de réduire la taille de l'espace d'action dans le cadre, nous proposons de manière innovante une méthode en deux étapes dans chaque tranche de temps. Plus précisément, l'agent DRL détermine d'abord l'ensemble des dispositifs ou flux IdO possibles qui sont connectés à chaque interface j , ce qui peut limiter la taille de l'espace d'action correspondant. Par la suite, l'agent DRL dérive une solution d'assignation de flux optimale basée sur l'ensemble donné d'appareils IdO connectés en résolvant un problème d'optimisation non convexe.

La forme générale de la fonction de valeur $Q(s, a)$ de notre architecture DRL qui se compose d'une phase de construction de Deep Neural Network (DNN) hors ligne et d'une phase d'apprentissage

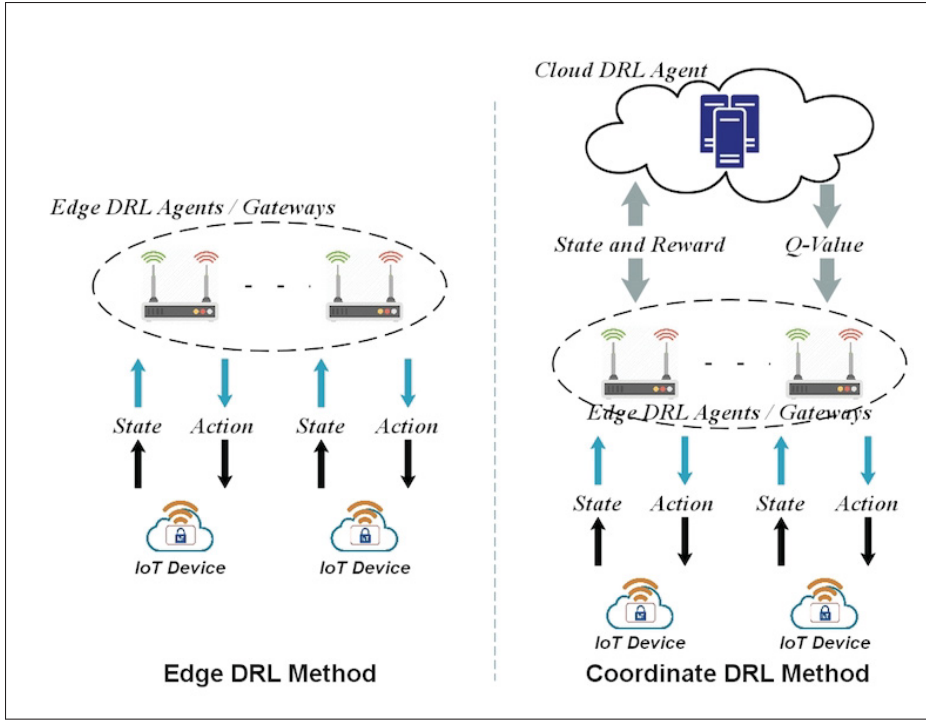


Figure 2.4 Architecture DRL

Q profond en ligne (Liu, Yang, Jiang, Xie & Zhang, 2019), est donné comme :

$$Q(s, a) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r_k(s_k, a_k) | s_0 = s, a_0 = a \right]. \quad (2.13)$$

Où $r_k(.)$ est la récompense obtenue dans l'intervalle de temps t , et γ est le facteur de remise qui se situe entre $[0, 1]$. Pour construire un DNN approximativement précis, la construction DNN hors ligne doit collecter des échantillons de valeurs estimées et ses correspondants (s, a) .

Pour construire un réseau DNN hors ligne suffisamment précis, nous utilisons un réseau de neurones à anticipation qui comporte deux couches cachées d'unités softplus entièrement connectées avec 64 et 32 neurones, respectivement. Nous utilisons la même méthode d'entraînement que (Mnih et al., 2015). Ce réseau peut accumuler suffisamment d'échantillons d'estimations de valeur et les (s, a) correspondants. L'espace d'état, l'espace d'action et la récompense sont décrits comme suit :

Algorithme 2.1 Le cadre Basé sur les DRL

Offline :

- 1 Chargez les profils de transition d'état historiques et les $Q(s, a)$ estimés correspondants dans la mémoire d'expérience $\mathcal{D}$ avec une capacité de $N_{\mathcal{D}}$; Pré-former le DNN avec les paires d'entrées (s, a) et les $Q(s, a)$ estimés correspondants ; **Online :**
- 2 **for** *chaque tranche horaire* t_z **do**
- 3 Sélectionnez une action aléatoire avec une probabilité ϵ ;
- 4 **if** $(a_k \notin Q(s_k, a_k))$ **then**
- 5 | prendre des mesures a_k
- 6 **else**
- 7 | $a_k = \operatorname{argmax}_a Q(s_k, a_k; \theta)$ Où $Q(\cdot)$ est estimé par le DNN,
- 8 **end if**
- 9 Exécuter l'action a_k ,
- 10 Dériver l'ensemble des flux possibles $\mathcal{F}_p$,
- 11 Sur la base de l'ensemble donné de flux, obtenez la solution optimale en résolvant OFAP-MIG,
- 12 Reconfigurer le réseau avec la nouvelle solution,
- 13 Observez la récompense r_k et le nouvel état s_{k+1} ,
- 14 $\mathcal{D} \leftarrow (s_k, a_k, r_k, s_{k+1})$,
- 15 Échantillonnez au hasard un mini-lot de transitions d'état Où (s_i, a_i, r_i, s_{i+1}) avec une taille de $N_{\mathcal{B}}$,
- 16 Cible $y_i = r_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta)$,
- 17 Mettre à jour les poids du DNN avec une fonction de perte de $(y_i - Q(s_{i+1}, a'; \theta))^2$,
- 18 **end for**

1. **Territoire de l'État :** L'état de l'agent DRL se compose de l'état binaire (accepter ou rejeter) de chaque flux d'utilisateur $x_{i,j} \in [0, 1]$ et sa demande respective qui est ici son échéance D_i . Ainsi, le vecteur d'état est représenté par :

$$[x_{1,j,k}, x_{2,j,k}, \dots, x_{I,j,k}, D_1, D_2, \dots, D_I] \quad (2.14)$$

2. **Espace Action :** L'agent décide quels flux sont acceptés ou rejetés par les interfaces des passerelles. L'action est donc représentée par :

$$[x_{1,j,k}, x_{2,j,k}, \dots, x_{n,j,k}] \quad (2.15)$$

Avec l'augmentation des flux et des interfaces et passerelles, l'espace d'action augmente de façon exponentielle et ce sera un problème difficile à traiter. De plus, chaque action a un effet sur les états, ce qui signifie que le nombre d'espaces d'états est également très grand.

3. **Récompense :** En effectuant une action a et en observant l'état s , l'agent obtiendra une récompense immédiate pour représenter l'objectif du cadre qui maximise la validité des données acceptées par les passerelles du réseau IdO tout en satisfaisant les exigences de délai des appareils IdO. Dans l'ensemble, nous définissons cette récompense comme :

$$R_{real} - R_{max} \quad (2.16)$$

Où R_{max} est la capacité maximale possible des interfaces dans le réseau IdO et R_{real} est la capacité réelle utilisée de la passerelle à chaque créneau horaire.

De plus, après la formation DNN hors ligne, nous adoptons le Q-learning profond pour le contrôle dynamique en ligne à l'agent DRL de périphérie. Le processus d'apprentissage profond en ligne sur la passerelle Edge peut être décrit comme suit :

- **étape 1 :** Au début de chaque intervalle de temps, l'agent DRL de passerelle déduit la valeur Q estimée par le DNN à partir de l'agent DRL à l'info nuagique avec l'entrée de la paire état-action (s_k, a_k) pour chaque a_k . Ensuite, pour sélectionner l'action d'exécution, la politique ϵ -greedy est toujours utilisée, ce qui signifie que l'agent DRL choisit l'action avec la valeur Q la plus élevée avec une probabilité $(1 - \epsilon)$.
- **étape 2 :** Sur la base de l'ensemble donné de flux possibles $\mathcal{F}_p$, déterminé dans la première étape, l'agent DRL dérive en outre une solution d'assignation de flux optimale en résolvant l'équation 2.2, problème d'optimisation soumis aux contraintes capacité énergétique, échéance des flux et capacité du réseau.
- **étape 3 :** Après avoir observé la récompense immédiate et l'état suivant des appareils IdO, les informations de transition d'état (s_k, a_k, rk, s_{k+1}) sont sauvées dans la mémoire d'expérience $\mathcal{D}$.

- **étape 4 :** A la fin du créneau temporel, l'agent DRL edge met à jour les paramètres θ du DNN avec des échantillons donnés $N_{\mathcal{B}}$ de la mémoire d'expérience $\mathcal{D}$.

2.3.3 GA-OFAP-MIG

Comme indiqué dans la section 2.3.1, OFAP-MIG est une programmation mixte non entière (PMNE) avec une complexité de calcul élevée, pour laquelle il est extrêmement difficile d'obtenir une solution globalement optimale avec une faible complexité (Li et al., 2019).

De nombreuses études ont tenté de développer des approches heuristiques pour approximer la solution à un problème d'allocation optimale des ressources. L'utilisation d'un algorithme génétique est l'une de ces approches heuristiques couramment utilisées pour résoudre ces problèmes. Les algorithmes génétiques (AG) sont des algorithmes heuristiques qui s'inspirent de la sélection naturelle et de la génétique évolutive. Avec un bon équilibre entre la complexité et l'efficacité, les AG fournissent une solution presque optimale au problème NP-difficile (Kim & Ko, 2015).

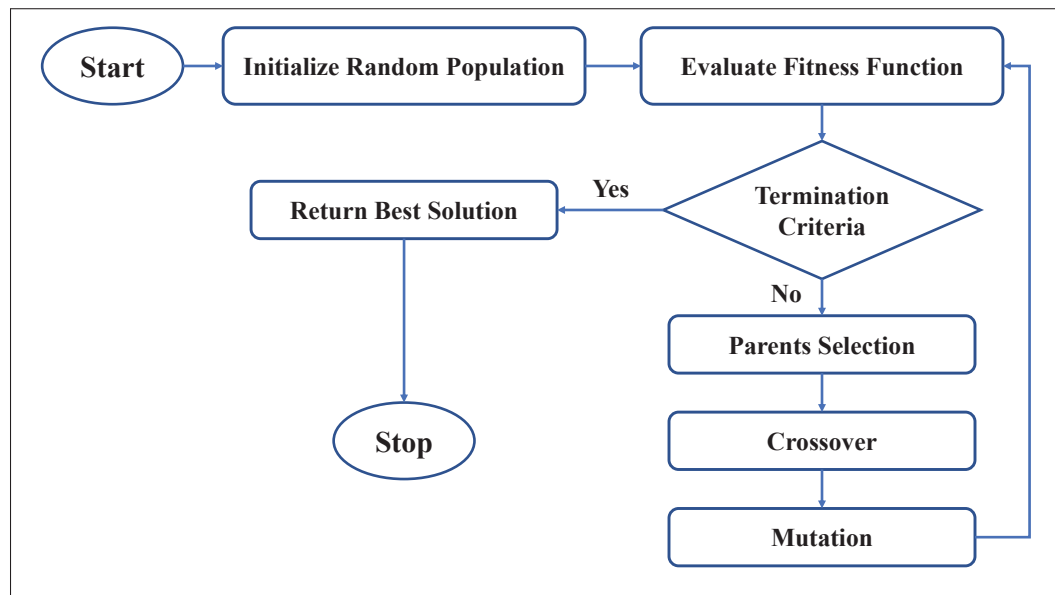


Figure 2.5 Organigramme de la Méthode GA

Les raisons du choix d'un algorithme génétique pour résoudre notre problème d'assignation de flux sont que notre problème est un problème multi-objectif. En d'autres termes, nous devons trouver un état d'assignation de flux qui :

1. Soit faisable lorsque l'on considère la connectivité des appareils et des passerelles IdO ;
2. Maximise les données valides acceptées par les passerelles de notre réseau IdO tout en satisfaisant leur QoS ;
3. Évite toute violation de la contrainte de capacité de passerelle pour toutes les passerelles ;

Ces objectifs multiples peuvent être représentés de manière simplifiée dans un algorithme génétique en définissant une fonction de fitness. Pour faciliter la mise en œuvre pratique, nous proposons une méthode de solution proche de l'optimum avec une complexité moindre en tirant parti de l'algorithme génétique. La figure 2.5 montre le processus de l'algorithme GA.

Chromosome :

Chaque solution possible doit être codée sous la forme d'un chromosome constitué de gènes, et elle peut être modifiée par un ensemble d'opérateurs génétiques. Habituellement, la représentation génétique d'une solution est un tableau de bits ou d'entiers. Par conséquent, nous proposons un schéma de codage qui a une localité et une héritabilité élevées et qui garantit que l'état d'assignation de chaque flux répond aux exigences et aux contraintes des passerelles et des interfaces. Notre schéma de codage indique quels appareils IdO sont affectés à quelle interface des passerelles. L'encodage du chromosome est comme ci-dessous :

$Chromosome \ C = Gene \ [i]$ $i = \text{Nombre de Flux}$

La solution représentée utilisant ce schéma de codage peut être facilement décodée dans un état d'assignation de flux. De plus, l'héritabilité de ce schéma de codage est élevée de sorte que la descendance résultant du croisement entre deux parents consiste en des solutions partielles des deux parents. Bien que ce schéma de codage puisse générer une solution infaisable, un tel cas peut facilement être traité par la fonction de fitness définie dans la sous-section suivante.

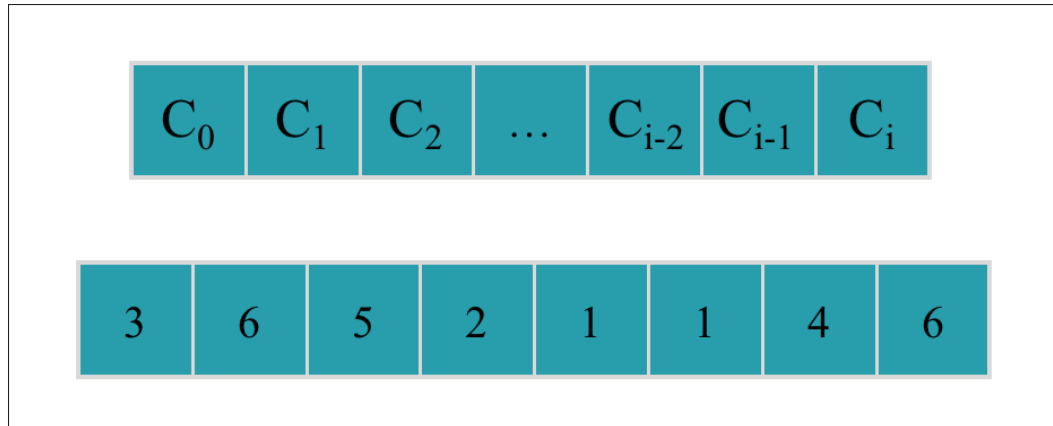


Figure 2.6 Codage de Chromosome

Ensuite, nous donnons les processus pour résoudre le problème OFAP-MIG avec le GA :

1. Initialisation :

La population est un sous-ensemble de solutions de la génération actuelle. Il peut également être défini comme un ensemble de chromosomes. La population initiale doit fournir des solutions possibles qui sont suffisamment diverses pour permettre à terme de trouver la solution optimale. La population est généralement définie comme un tableau bidimensionnel de – taille de la population, taille x , taille des chromosomes. Il existe deux méthodes principales pour initialiser une population dans un GA :

- initialisation aléatoire : Remplir la population initiale avec des solutions complètement aléatoires.
- initialisation heuristique : Remplir la population initiale en utilisant une heuristique connue pour le problème.

Deux modèles de population sont largement utilisés :

- état Stationnaire : Dans l'état stationnaire GA, nous générons un ou deux descendants à chaque itération et ils remplacent un ou deux individus de la population. Un GA en régime permanent est également appelé GA incrémental.

- générationnel : dans un modèle générationnel, nous générons « n » descendants, où n est la taille de la population, et la population entière est remplacée par la nouvelle à la fin de l'itération.

2. **Évaluation de la Condition Physique :**

La fonction de fitness simplement définie est une fonction qui prend une solution candidate au problème en entrée et produit en sortie comment « convient » à notre « bonne » solution par rapport au problème considéré. Après les initialisations, les valeurs de fitness sont dérivées pour chaque chromosome de la population actuelle en fonction du problème d'optimisation et de sa contrainte. Son but est de sélectionner de meilleurs chromosomes pour reproduire la prochaine génération en tant que parents. Juger si un chromosome va bien ou non est basé sur sa propre aptitude.

Plus la valeur de fitness est faible, meilleur est le chromosome. Cependant, dans notre cas, tous les chromosomes possibles ne sont pas des solutions réalisables, en raison des exigences des contraintes de l'équation 2.9. En faisant en sorte que les individus ayant une meilleure forme physique reproduisent des descendants, GA pourrait rechercher à la fois des régions faisables et infaisables, puis obtenir plus facilement une solution globale sous-optimale.

3. **Sélection de Parent :**

La sélection des parents est le processus de sélection des parents qui s'accouplent et se recombinent pour créer des descendants pour la prochaine génération. La sélection des parents est très cruciale pour le taux de convergence de l'AG, car de bons parents conduisent les individus vers des solutions meilleures et plus adaptées. Cependant, il convient de veiller à éviter qu'une solution extrêmement adaptée ne s'empare de l'ensemble de la population en quelques générations, car cela conduit à des solutions proches les unes des autres dans l'espace des solutions, entraînant ainsi une perte de diversité. Le maintien d'une bonne diversité dans la population est extrêmement crucial pour le succès d'une AG. Cette prise de la population entière par une solution extrêmement ajustée est connue sous le nom de convergence prématurée et est une condition indésirable dans un AG.

Dans cette section, nous utilisons la méthode de sélection par tournoi qui a été proposée pour la première fois par Brindle en 1983. Dans la sélection par tournoi K-Way, nous

sélectionnons K individus de la population au hasard et sélectionnons les meilleurs d'entre eux pour devenir un parent. Le même processus est répété pour sélectionner le parent suivant. La sélection de tournoi est également extrêmement populaire dans la littérature car elle peut même fonctionner avec des valeurs de fitness négatives.

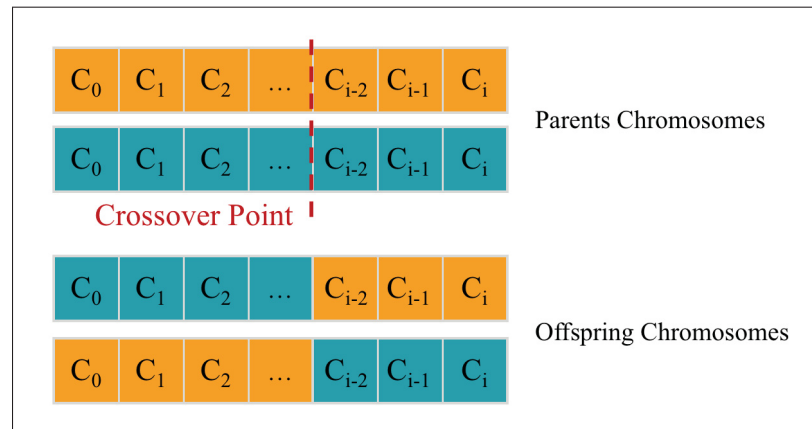


Figure 2.7 Croisement en un Point

4. **Crossover :**

L'opérateur de croisement est analogue à la reproduction et au croisement biologique. Dans ce cas, plus d'un parent est sélectionné et une ou plusieurs progénitures sont produites en utilisant le matériel génétique des parents. Le crossover est généralement appliqué dans un GA avec une forte probabilité. Les opérateurs de croisement bien connus sont un point unique, deux points, un point k, uniforme, partiellement apparié, ordre, croisement préservant la priorité, shuffle, substitut réduit et cycle.

Dans ce croisement à un point, un point de croisement aléatoire est sélectionné et les queues de ses deux parents sont échangées pour obtenir de nouveaux descendants. La figure 2.7 montre les informations génétiques après échange. Il a remplacé les bits du tableau de queue des deux parents pour obtenir la nouvelle progéniture.

5. **Mutation :**

En termes simples, la mutation peut être définie comme un petit ajustement aléatoire dans le chromosome, pour obtenir une nouvelle solution. Il est utilisé pour maintenir et introduire la diversité dans la population génétique et est généralement appliqué avec une faible

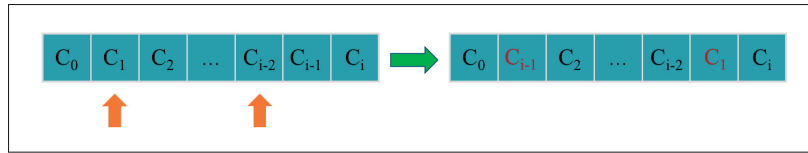


Figure 2.8 Mutation

probabilité. Si la probabilité est très élevée, le GA se réduit à une recherche aléatoire.

La mutation est la partie de l'AG qui est liée à « l'exploration » de l'espace de recherche. Il a été observé que la mutation est essentielle à la convergence de l'AG alors que le croisement ne l'est pas. Les opérateurs de mutation bien connus sont le déplacement, l'inversion simple et la mutation brouillée. Dans la mutation d'échange comme indiqué dans la figure 2.8, nous sélectionnons au hasard deux positions sur le chromosome et échangeons les valeurs. Ceci est courant dans les codages basés sur la permutation.

2.4 Conclusion

Dans ce chapitre, nous avons présenté notre modèle de système et son modèle mathématique ainsi que la fonction objectif dont le but est de maximiser la quantité de données valides acceptées par les interfaces des passerelles du réseau IdO de la maison intelligente, tout en satisfaisant les exigences de délais des flux et services. Nous avons proposé des approches basées sur les DRL et génétiques pour résoudre le problème d'optimisation et atteindre la solution optimale. Dans le chapitre suivant, nous mettrons en œuvre les solutions mentionnées et évaluerons leurs performances.

CHAPITRE 3

RÉSULTATS EXPÉRIMENTAUX

Ce chapitre sera consacré sur la partie pratique de notre travail. Cela consiste à simuler notre système avec des données proches du réel. En effet, c'est ici que nous allons effectuer l'implémentation des modèles précédemment créés. Cela nous permettra de vérifier l'efficacité de ces modèles ainsi que de valider nos hypothèses. Notre implémentation permettra également la vérification des méthodes utilisées ainsi que leurs validation avec les résultats obtenus.

Nous allons entamer cette partie par la conversion de nos algorithmes en code de programme qui seront exécutés dans la simulation. L'implémentation est suivie par la définition du protocole de la simulation. La suite consiste à la spécification des configurations nécessaires pour faire tourner la simulation. Ces configurations sont basées sur des valeurs existantes ainsi que des valeurs expérimentales. Les résultats de la simulation seront interprétés après avoir défini le montage de la simulation, suivis de discussions et commentaires sur ces résultats.

3.1 Configuration de la simulation

Dans cette section, nous présentons les détails de la simulation et de l'implémentation de l'algorithme proposé dans le chapitre précédent. D'abord, nous présentons l'architecture de simulation et ses paramètres. Ensuite, nous décrivons plusieurs scénarios pour tester les performances des modèles proposés en les comparant avec la solution optimale qui est calculée par le solveur CPLEX. Enfin, nous présentons et discutons les résultats des scénarios.

3.1.1 Environnement de simulation

Nous considérons un réseau IdO avec trois passerelles. Ensuite, nous faisons varier le nombre d'appareils IdO de 10 à 80. Ces appareils IdO sont équipés de trois types de technologies de communication différents : le WiFi, le Bluetooth et le ZigBee avec des débits de transmission de 2 Mbps, 500 Kbps et 200 Kbps respectivement. Les tailles des flux sont générées de manière

aléatoire et contiennent des données pour deux types d'applications IdO de ville intelligente (Amor *et al.*, 2019) (a) applications à faible débit de données avec un délai maximum de 2 minutes (par exemple, parking intelligent, surveillance de la qualité de l'air) et (b) Applications critiques qui peuvent tolérer des délais allant jusqu'à 20 secondes (par exemple, alarmes pour les situations énergétiques, applications de soins de santé).

Dans notre implémentation de la construction DNN, la capacité de la mémoire d'expérience $N_{\mathcal{D}}$ et la capacité du minibatch $N_{\mathcal{B}}$ sont fixés respectivement à 500 et 32, et la remise de récompense $\gamma = 0.5$.

Dans la mise en œuvre de l'AG, nous avons défini la taille de la population à 40 et le taux de mutation à 0,005. La taille du tournoi est fixée à 15 et le nombre de points croisés est 2.

Tableau 3.1 Caractéristiques des interfaces pour les simulations

Interface	Taux de Transmission	Capacité	Tampon
Interface 1	2 Mb/s	2000 B	500 B
Interface 2	500 Kb/s	1000 B	0
Interface 3	200 Kb/s	1500 B	0
Interface 4	2 Mb/s	2000 B	500 B
Interface 5	500 Kb/s	1000 B	0
Interface 6	200 Kb/s	1500 B	0

3.1.2 Scénarios de simulation

Nous considérons deux scénarios différents en faisant varier différents paramètres :

1. **Quantité de données générées (Scénario 1) :** Ce scénario est conçu pour montrer les performances de notre solution proposée. Dans ce scénario, le nombre de flux générés est fixé à 80 et le nombre d'interfaces par appareil est fixé à 3. On fait varier la quantité totale de données générées. Pour chaque ensemble de flux généré, nous calculons le taux

d'acceptation de données valides qui est égal à la quantité totale de données acceptées divisée par la quantité totale de données générées.

Tableau 3.2 Quantité de données générées

Nombre de Passerelles	Nombre de Flux	Nombre d'Interfaces
3	80	6

2. **Taille du réseau (Scénario 2) :** Nous étudions l'impact de la variation de la taille du réseau. Le nombre d'interfaces par appareil est fixé à trois interfaces et on fait varier le nombre d'appareils qui envoient des flux vers la passerelle. Les appareils IdO génèrent des flux dont la somme est égale à 40 Mo.

Tableau 3.3 Taille du réseau

Nombre de Passerelles	Somme des Gonnées Générées	Nombre d'Interfaces
3	40 MB	6

3. **Durée du processus (Scénario 3) :** Nous étudions l'impact de la variation de la taille du réseau sur le temps d'exécution requis pour différentes solutions. On suppose qu'il y a trois passerelles dans notre réseau IdO. Le nombre d'interfaces par appareil est fixé à deux interfaces et on fait varier le nombre d'appareils qui envoient des flux vers les passerelles. Les appareils IdO génèrent des flux dont la somme est égale à 40 Mo. Nous essayons également de faire varier le nombre des flux sur les deux solutions proposés et de voir comment ce paramètre affectera le temps et les performances des solutions.

Tableau 3.4 Durée du processus

Somme des Gonnées Générées	Nombre de Passerelles	Nombre d'Interfaces
40 MB	3	6

3.2 Résultats de la simulation

1. **Quantité de données générées (Scénario 1) :** La figure 3.1 montre la somme des données acceptées et la figure 3.2 montre le taux d'acceptation d'une donnée valide tout en faisant varier la quantité de données provenant des appareils IdO vers la passerelle. Au début, lorsque la quantité de données générées est comprise entre 0 Mo et 10 Mo, le taux d'acceptation des algorithmes de la solution varient entre 0,9 et 1. Nous pouvons voir que les deux solutions proposées sont proches de la solution optimale. Par exemple, lorsque les appareils IdO génèrent 20 Mo de données, le taux d'acceptation de la solution optimale est d'environ 47 pour cent tandis que le taux d'acceptation de GA et DRL est respectivement de 45 et 44. On observe que la somme des données acceptées augmentent en fonction de la charge du réseau jusqu'à atteindre la quasi-totalité de la capacité du réseau. La raison de cette augmentation est que lorsque la taille des flux augmente, l'agent répartit les flux entre les interfaces disponibles en respectant leurs impératifs de délais.

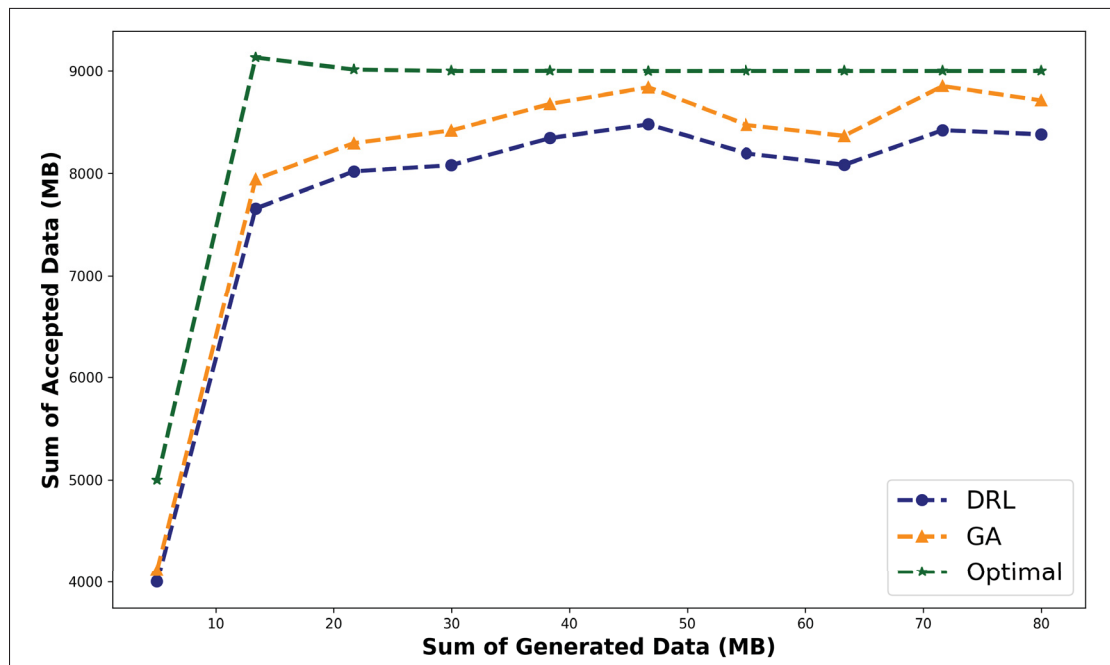


Figure 3.1 Somme des Données Acceptées

2. **Taille du réseau (Scénario 2) :** Dans ce scénario, nous pouvons voir dans les figures 3.3 et 3.4 qu'au début, lorsque nous augmentons le nombre de flux, la somme des données

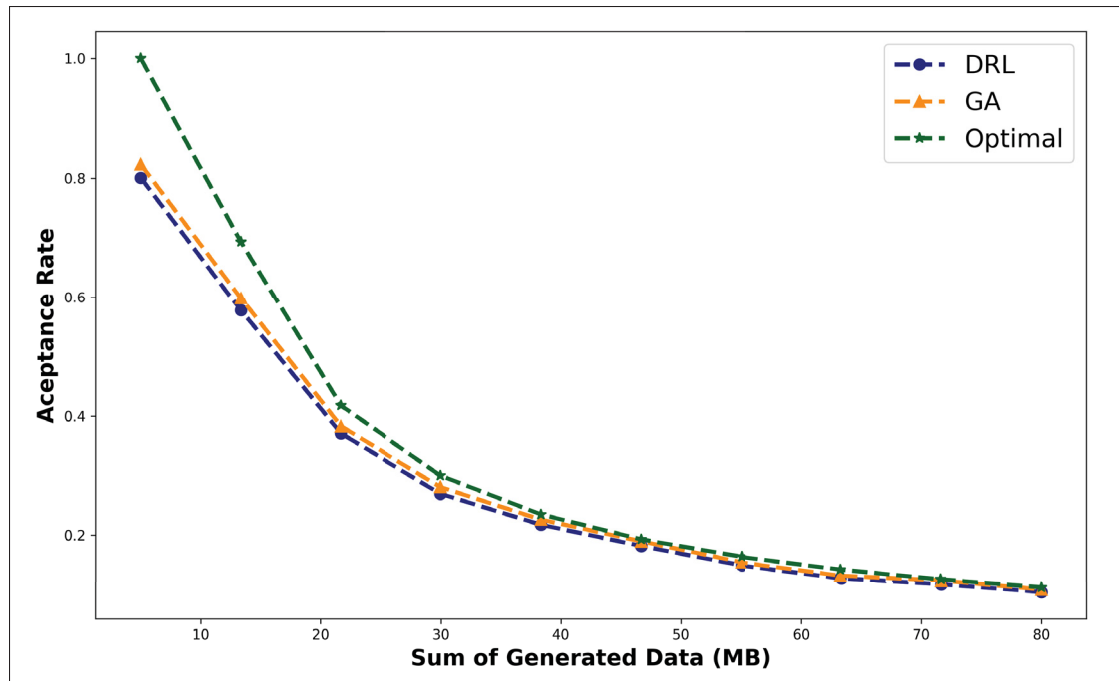


Figure 3.2 Taux d'Acceptation

acceptées et également le taux d'acceptation augmentent. Lorsqu'on atteint 50 flux, on observe que la somme des données acceptées fluctue autour de la capacité maximale des interfaces. La raison en est la taille des flux. Cela signifie que lorsque l'agent peut répartir les flux entre les interfaces avec succès, nous voyons que le taux d'acceptation augmente mais lorsqu'il ne peut pas le faire en raison des contraintes, le taux d'acceptation diminue.

3. **Durée du processus (Scenario 3) :** Dans la figure 3.5, on peut voir que le temps d'exécution des deux méthodes augmente lorsque l'on augmente le nombre de flux. Il est également bon de mentionner que la méthode DRL est très dépendante de la formation.

3.3 Discussion

En résumé, en comparant les 3 scénarios ensemble, on peut voir que la méthode DRL donne des résultats presque optimaux en temps opportun.. Il est à noter que le taux d'acceptation de la solution basée sur les DRL dépend fortement de la formation. Les résultats présentés dans ces

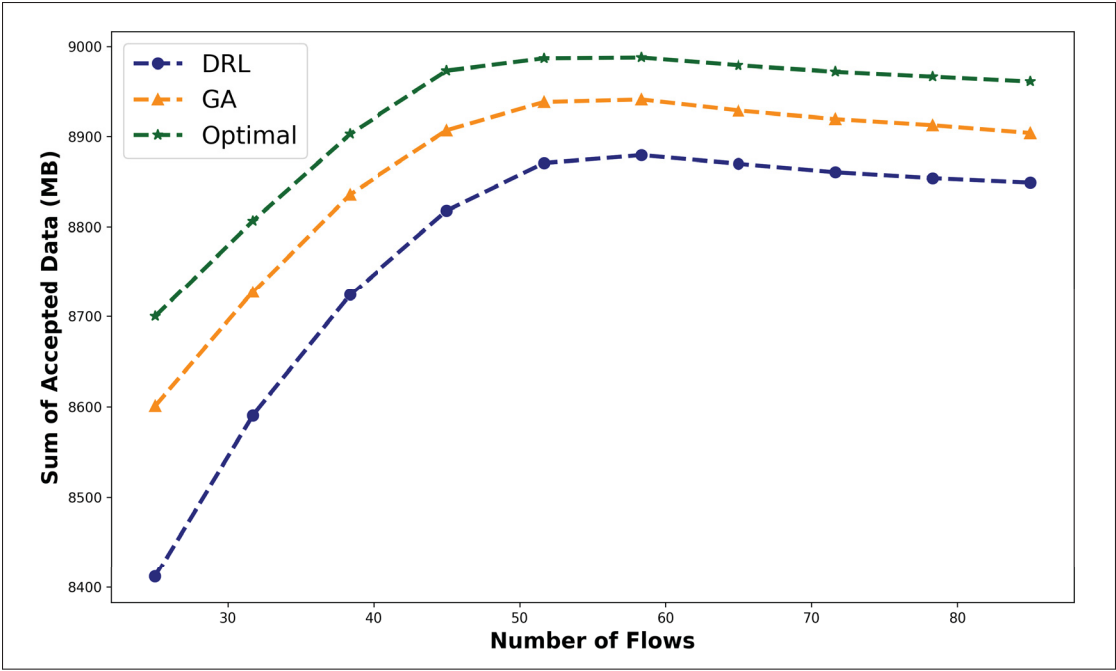


Figure 3.3 Somme des Données Acceptées

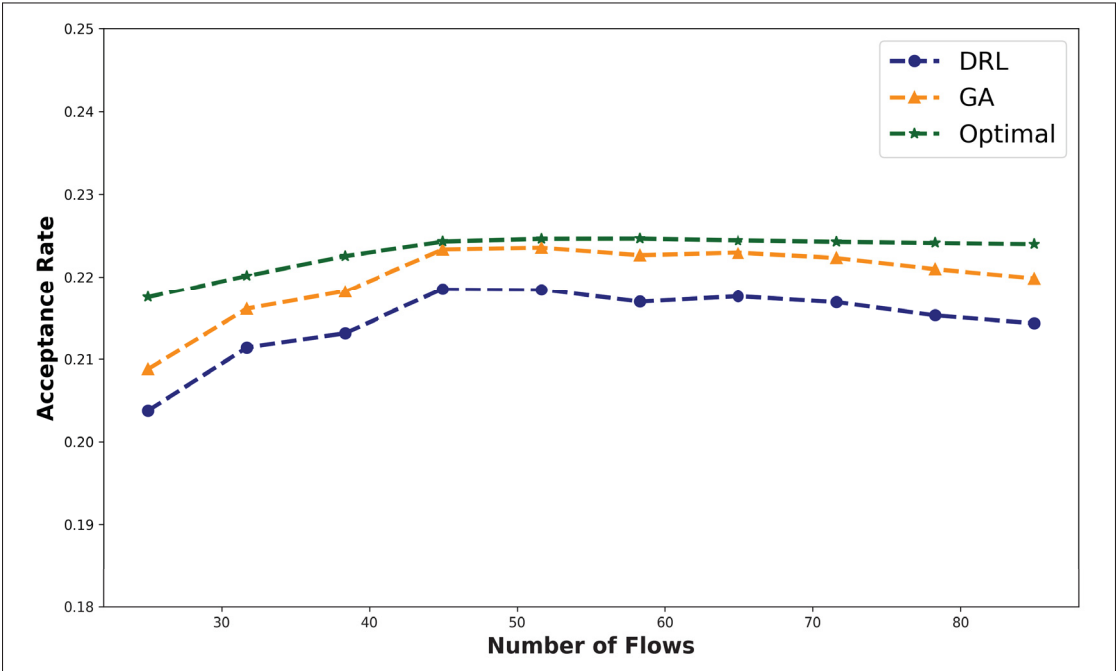


Figure 3.4 Taux d'Acceptation

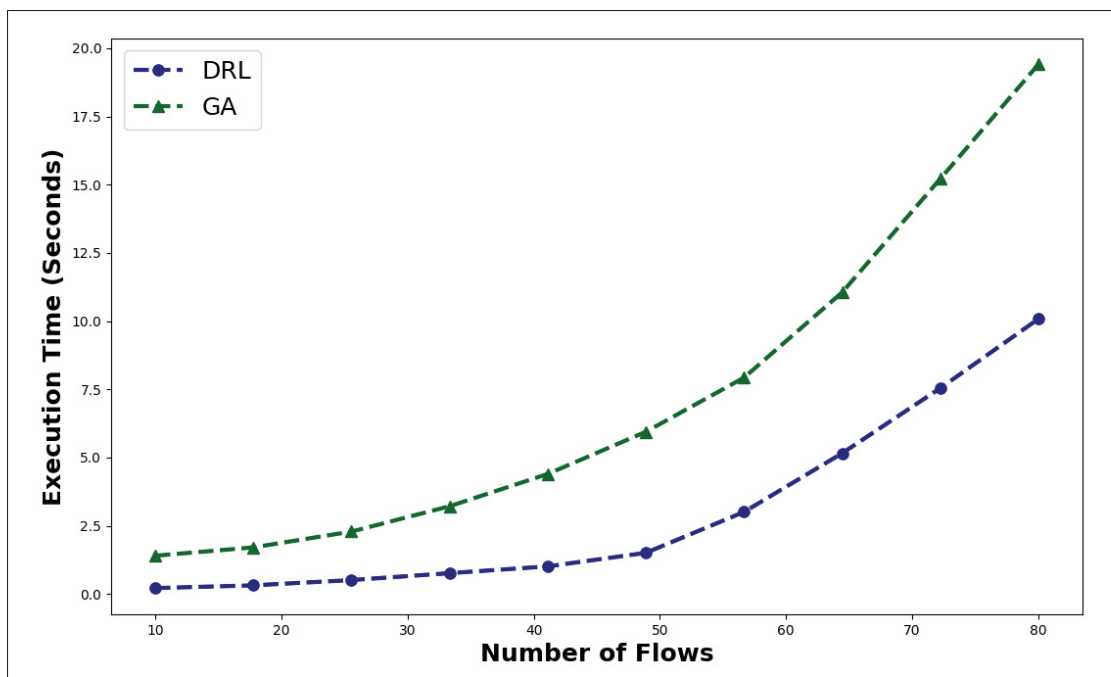


Figure 3.5 Durée du Processus

scénarios sont obtenus à partir d'agents formés afin qu'une solution presque optimale puisse être calculée en temps opportun. Cependant, dans le cas des agents sans formation, un temps supplémentaire sera nécessaire pour former les agents à comprendre les états et les actions et leurs récompenses appropriées.

CONCLUSION ET RECOMMANDATIONS

Dans ce mémoire, nous avons étudié l'assignation optimale des flux d'appareils IdO dans un réseau IdO de maison intelligente avec plusieurs passerelles équipées de multiples interfaces hétérogènes. Nous avons formulé le problème mathématiquement, et défini le problème d'assignation de flux optimal pour les passerelles multi-interfaces nommé OFAP-MIG, qui est une programmation mixte non entière (PMNE). Son objectif est de maximiser la quantité de données valides acceptées par les passerelles du réseau IdO tout en respectant le temps limite des flux et en tenant compte également de la capacité de chaque interface, du délai de commutation entre ces interfaces et de la ressource d'alimentation restante dans l'appareil pour envoyer le flux.

Le système conçu se compose de deux modules : Le module de surveillance et d'optimisation. Le module de surveillance collecte les informations du réseau et des appareils connectés qui sont nécessaires pour prendre une décision dans le module d'optimisation. Le module d'optimisation utilise les données collectées dans le module de surveillance comme état d'environnement et actions pour exécuter l'algorithme qui permet l'assignation optimale des flux aux interfaces réseau des passerelles. Pour résoudre l'OFAP-MIG, nous avons développé deux algorithmes qui exploitent efficacement les interfaces disponibles des passerelles tout en respectant les délais des flux.

Le premier algorithme est basé sur l'algorithme génétique et le second utilise le "Deep Reinforcement Learning (DRL)". Nous avons ensuite effectué l'évaluation des performances de nos solutions en les comparant avec la solution optimale. Les résultats expérimentaux montrent que notre solution est prometteuse et surpasse les travaux antérieurs. L'efficacité de nos algorithmes ont été démontrée selon les paramètres suivants :

1. Le taux d'acceptation des données valides ;
2. Le nombre d'appareils IdO connectés aux passerelles ;

À l'avenir, nous avons l'intention d'étendre notre modèle pour intégrer une fonction multi-objectifs et prendre en compte d'autres paramètres tels que les préférences des utilisateurs, la capacité de prévision du trafic, la capacité de batterie et de mémoire des passerelles. Ces paramètres permettront une meilleure assignation des flux aux interfaces en termes de coût et de qualité d'expérience (QoE).

BIBLIOGRAPHIE

- 3GPP. (2016). Standardization of NB-IOT. Repéré le 2016-06-22 à https://www.3gpp.org/news-events/1785-nb_iot_complete.
- Al-Saedi, S. B. & Azim, M. M. A. (2017). Radio Frequency Near Communication(RFNC) Technology : An Integrated RFID-NFC System for Objects' Localization. *IEEE-GCC Conference and Exhibition (GCCCE)*.
- Aloi, G., Caliciuri, G., Fortino, G., Gravina, R., Pace, P., Russo, W. & Savaglio, C. (2017). Enabling IoT Interoperability Through Opportunistic Smartphone-Based Mobile Gateways. *Journal of Network and Computer Applications*, 81, 74-84.
- Amor, M. G., Nguyen, K. K., Pham, C. & Cheriet, M. (2019). Optimized Flow Assignment in a Multi-Interface IoT Gateway. *International Wireless Communications and Mobile Computing Conference(IWCMC)*, pp. 1379-1384.
- Baert, M., Camerlynck, P., Crombez, P. & Hoebeke, J. (2019). A BLE-Based Multi-Gateway Network Infrastructure with Handover Support for Mobile BLE Peripherals. *IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 91-99.
- Chen, Y. et al. (2021). Deep Reinforcement Learning-Based Dynamic Resource Management for Mobile Edge Computing in Industrial Internet of Things. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 17(17), 4925-4934.
- Desai, P., Sheth, A. & Anantharam, P. (2015). Semantic Gateway as a Service Architecture for IoT Interoperability. *IEEE International Conference on Mobile Services*, pp. 313-319.
- Dian, F. J., Yousefi, A. & Lim, S. (2018). A practical study on Bluetooth Low Energy (BLE) throughput. *IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 768-771.
- Farhan, L., Kharel, R., Kaiwartya, O., Hammoudeh, M. & Adebisi, B. (2018). Towards Green Computing for Internet of Things : Energy Oriented Path and Message Scheduling Approach. *Sustainable Cities and Society*, 38, 195-204.
- Gaur, P. & Tahiliani, M. P. (2015). Operating Systems for IoT Devices : A Critical Survey. *IEEE Region 10 Symposium*, pp. 33-36.
- Guoqiang, S., Yanming, C., Chao, Z. & Yanxu, Z. (2013). Design and Implementation of a Smart IoT Gateway. *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pp. 720-723.

- HIR. (2019). Internet of Things (IoT). Dans *Heterogeneous Integration Roadmap*. IEEE.
- Honggang, Z., Chen, S. & Leyu, Z. (2018). Design and Implementation of Lightweight 6LoWPAN Gateway Based on Contiki. *IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*.
- Kang, B. & Choo, H. (2018). An Experimental Study of a Reliable IoT Gateway. *ICT Express*, 4(3), 130-133.
- Kang, B., Kim, D. & Choo, H. (2017a). Internet of Everything : A Large-Scale Autonomic IoT Gateway. *IEEE Transactions on Multi-Scale Computing Systems*, 3(3), 206-214.
- Kang, B., Kim, D. & Choo, H. (2017b). Internet of Everything : A Large-Scale Autonomic IoT Gateway. *IEEE Transactions on Multi-Scale Computing Systems*, 3(3), 206-214.
- Kim, M. & Ko, I. (2015). An Efficient Resource Allocation Approach based on a Genetic Algorithm for Composite Services in IoT Environments. *IEEE International Conference on Web Services*, pp. 543-550.
- Lehong, C., Isong, B., Lugayizi, F. & Abu-Mahfouz, A. M. (2020). A Survey of LoRaWAN Adaptive Data Rate Algorithms for Possible Optimization. *IEEE International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*.
- Lei, L., Xu, H., Xiong, X., Zheng, K. & Xiang, W. (2019). Joint Computation Offloading and Multiuser Scheduling Using Approximate Dynamic Programming in NB-IoT Edge Computing System. *IEEE Internet of Things Journal*, 6(3), 5345-5362.
- Li, X. et al. (2019). Optimizing Resources Allocation for Fog Computing-Based Internet of Things Networks. *IEEE Access*, 7, 64907-64922.
- Liu, Y., Yang, C., Jiang, L., Xie, S. & Zhang, Y. (2019). Intelligent Edge Computing for IoT-Based Energy Management in Smart Cities. *IEEE Network*, 33(2), 111-117.
- Liya, M. & Aswathy, M. (2020). LoRa Technology for Internet of Things(IoT) : A Brief Survey. *IEEE Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 8-13.
- Mnih, V. et al. (2015). Human-Level Control Through Deep Reinforcement Learning. *Nature*, 518(7540), 529-533.
- Osman, N. I. & Abbas, E. B. (2018). Simulation and Modelling of LoRa and Sigfox Low Power Wide Area Network Technologies. *IEEE International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1-5.

- Pan, G., He, J., Wu, Q., Fang, R., Cao, J. & Liao, D. (2018). Automatic stabilization of Zigbee network. *IEEE International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 224-227.
- Qin, H., Chen, W., Cao, B., Zeng, M., Li, J. & Peng, Y. (2018). DIPS : Dual-Interface Dual-Pipeline Scheduling for Energy-Efficient Multihop Communications in IoT. *IEEE Internet of Things Journal*, 6(1), 718-733.
- Ray, P. P. (2018). A Survey on Internet of Things Architectures. 30(3), 291-319.
- Samie, F., Bauer, L. & Henkel, J. (2020). Hierarchical Classification for Constrained IoT Devices : A Case Study on Human Activity Recognition. *IEEE Internet of Things Journal*, 7(9), 8287-8295.
- Sobin, C. C. (2020). A Survey on Architecture, Protocols and Challenges in IoT. *Wireless Personal Communications*, 112, 1383-1429.
- Tsai, C. & Liu, S. (2018). An Effective IoT Service-to-Interface Assignment Algorithm via Search Economics. *IEEE Internet of Things Journal*, 5(3), 1708-1718.
- Uddin, M., Mukherjee, S., Chang, H. & Lakshman, T. V. (2018). SDN-Based Multi-Protocol Edge Switching for IoT Service Automation. *IEEE Journal on Selected Areas in Communications*, 36(12), 2775-2786.
- Wang, W., Xie, T., Liu, X. & Zhu, T. (2018). ECT : Exploiting Cross-Technology Concurrent Transmission for Reducing Packet Delivery Delay in IoT Networks. *IEEE INFOCOM - IEEE Conference on Computer Communications*, pp. 369-377.