

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université du 8 mai 1945-Guelma-

Faculté de Mathématiques, Informatique et Sciences de la Matière
Département d'informatique



Mémoire

Spécialité : Informatique

Option:

Sciences et technologies de l'information et de la communication

Thème

Modèle d'auto-organisation pour les protocoles de routage dans les villes intelligentes.

Présenté par:

FERHAT Aymen

Encadré par:

Dr. BENHAMIDA Nadjette

Membres du jury:

Mr. BERREHOUMA Nabil

Président

Dr. CHOHRA Chemseddine

Examineur

Juin 2023.

Remerciements

Je tiens à exprimer ma profonde gratitude envers Dr. Nadjette BENHAMIDA pour son soutien et son accompagnement précieux tout au long de mon parcours de recherche. Sa guidance experte, sa patience et son dévouement ont été d'une importance capitale pour la réalisation de ce travail de mémoire.

Grâce à Dr. Nadjette BENHAMIDA, j'ai pu développer mes compétences en recherche, affiner ma méthodologie et approfondir ma compréhension du sujet. Ses conseils éclairés, ses remarques pertinentes et sa disponibilité constante m'ont permis d'avancer dans la bonne direction et de surmonter les difficultés rencontrées.

Je suis reconnaissant de la confiance que Dr. Nadjette BENHAMIDA a placée en moi, ainsi que de sa capacité à m'encourager à repousser mes limites et à viser l'excellence académique. Ses enseignements resteront une source d'inspiration et de motivation tout au long de ma carrière.

Je tiens également à souligner l'importance de sa bienveillance et de son soutien sur le plan personnel. Dr. Nadjette BENHAMIDA a su créer un environnement propice à l'apprentissage et au développement, où je me suis senti écouté et soutenu dans mes aspirations académiques.

Ce travail de mémoire est le résultat de notre collaboration fructueuse, et je suis honoré de lui dédier cette réalisation. Je lui suis profondément reconnaissant pour son rôle déterminant dans mon parcours universitaire. Je suis extrêmement chanceux d'avoir eu l'opportunité de travailler avec vous.

Dédicaces

Je tiens à exprimer ma profonde gratitude envers mes parents et ma famille pour leur amour, soutien inconditionnel tout au long de ce parcours de maîtrise. Leur encouragement constant, leur sagesse et leurs conseils précieux ont été les piliers sur lesquels j'ai pu m'appuyer pour atteindre mes objectifs académiques. Leur présence inébranlable dans ma vie a été ma source de motivation constante.

Je souhaite également exprimer ma reconnaissance à mes enseignants et mentors qui m'ont inspiré et poussé à repousser mes limites intellectuelles. Leurs connaissances approfondies, leur passion pour l'apprentissage et leur dévouement à transmettre leur savoir m'ont permis de grandir en tant qu'étudiant et chercheur.

Mes sincères remerciements vont également à mes camarades de classe et amis qui ont partagé ce voyage avec moi. Leurs discussions enrichissantes, leur soutien mutuel et leur esprit collaboratif ont contribué à rendre cette expérience académique mémorable et gratifiante.

Cette mémoire est le fruit des efforts collectifs de toutes ces personnes exceptionnelles, et je suis honoré de leur dédier ce travail. Leur soutien m'a permis de surmonter les défis et de réaliser mes aspirations académiques.

Merci du fond du cœur à tous ceux qui ont contribué à cette réalisation.

Aymen FERHAT

Table des matières

Table des matières	1
Liste de Figures	3
Liste de tables	4
Liste des équations	5
Liste des acronymes	6
Résumé	8
Introduction générale	11
Chapitre I : Etat de l'art	13
1. Introduction	13
2. Ville intelligente	14
2.1. Définition	14
2.2. Architecture des villes Intelligentes	15
2.3. Applications des villes intelligentes	17
3. Internet des Objets (IoT)	20
3.1. Définitions	20
3.2. Les technologies IoT	21
3.3. Les réseaux IoT	21
3.4. Les défis de l'IoT	25
4. Auto-organisation	26
4.1. Définition et objectifs de l'auto-organisation	27
4.2. Décisions locales versus distribuées ?	29
4.3. Quelques clefs pour l'auto-organisation	30
5. Conclusion	30
Chapitre II : Le routage dans les villes intelligentes	31
1. Introduction	31
2. Routage de données	32
2.1. Types de routage	32
2.2. Métriques de routage	35
3. Travaux connexes	35
4. Éléments de la problématique	37
5. Contribution	38
5.1. Description de la solution proposée	38
5.2. Algorithme proposé	41

5.3. Les avantages de notre contribution	42
6. Conclusion	43
Chapitre III : Implémentation et réalisation	44
1. Introduction	44
2. Modèle du système	44
3. Organigramme d'exécution	45
4. Implémentation	48
4.1. Environnement de développement	48
4.2. Présentation d' OMNeT++	49
5.3. Description générale de l'application réalisée	51
4.4. Paramètres de simulation	54
5. Scénario d'exécution	55
6. Discussion	61
7. Conclusion	62
Conclusion générale et perspectives	63
Références bibliographiques	65

Liste de Figures

Figure 1. Villes Intelligentes [6].	15
Figure 2. Architecture des villes Intelligentes [7].	16
Figure 3. Applications des villes Intelligentes [8].	18
Figure 4. Mobilité des villes intelligentes [9].	19
Figure 5. Réseau cellulaire [12].	22
Figure 6. Réseau de capteurs [13].	23
Figure 7. Réseau MANET [14].	24
Figure 8. Réseau VANET [15].	24
Figure 9. Quelques stratégies de structures d'auto-organisation [20].	27
Figure 10. Modèle du système.	45
Figure 11. Organigramme d'exécution au niveau de l'émetteur.	46
Figure 12. Organigramme d'exécution au niveau du destinataire.	47
Figure 13. Organigramme d'exécution au niveau des nœuds intermédiaires.	48
Figure 14. Logo de l'OMNet++.	49
Figure 15. Diffusion du RREQ.	52
Figure 16. Mise à jour du RREQ.	53
Figure 17. Calcul du facteur F.	54
Figure 18. Choix de la meilleure route.	54
Figure 19. Étape 1 « diffusion du RREQ par l'émetteur.	56
Figure 20. Étape 2 « le handleRREQ au niveau du nœud 1 et nœud 3 ».	56
Figure 21. HandleRREQ dans les nœuds 1 et 3.	57
Figure 22. Étape 2 au niveau du nœud 7.	57
Figure 23. Étape 2 au niveau du nœud 6.	58
Figure 24. Étape 2 au niveau du nœud 2.	58
Figure 25. Étape 2 au niveau du nœud 4.	58
Figure 26. Étape 3 « renvoi de la réponse avec le score calculé du route courante » 1.	59
Figure 27. Étape 3 « renvoi de la réponse avec le score calculé du route courante » 2.	59
Figure 28. Mise à jour de la table du routage.	60
Figure 29. Étape finale « choix du meilleure route » 1.	60
Figure 30. Étape finale « choix du meilleure route » 2.	61

Liste de tables

Tableau 1 : Paramètre de simulation.	55
---	----

Liste des équations

Équation 1 : calcul des scores au niveau du récepteur.....	40
Équation 2 : calcul du facteur de décision au niveau de l'émetteur.....	40

Liste des acronymes

Acronyme	Désignation
AODV	Ad hoc On-Demand Distance Vector
AP	Access point
APDU	Application Protocol data unit
ARPA	Agence des projets de recherche
ARPANET	Advanced Research Projects Agency Network
ASCII	American Standard Code for Information Interchange
BS	Base station
CASCR	Context-Awareness in Sea Computing Routing Protocol
CoAP	Constrained Application Protocol
CPS	Check clocks per second
CSMA	Carrier Sense Multiple Access
DARPA	Defense Advanced Research Projects Agency
DEV	Development Network
DHCP	Dynamic Host Configuration Protocol
DNS	domain name system
DSDV	Destination Sequenced Distance Vector
DYMO	Dynamic MANET On-demand
EBCDIC	Extended Binary Coded Decimal Interchange Code
EGP	Exterior Gateway Protocol
ESS	extended service set
FANET	Flying ad hoc network
FDDI	Fiber Distributed Data Interface
FTP	File transfer protocol
GUI	Graphical user interface
HLR	Home Location Register
HPC	High Performance Computing
HTTP	Hypertext Transfer Protocol
IBSS	Independent basic service set
IDE	integrated development environment
IEEE	Institute of Electrical and Electronics Engineers
IGP	interior gateway protocol
IOT	Internet of things
IP	Internet Protocol
ISO	International Organization for Standardization
ITS	Information technology specialist
LAN	Local area network
LEACH	Low-energy adaptive clustering hierarchy
LORA	long range
LTE	Long-Term Evolution
MAN	metropolitan area network
MANET	mobile ad hoc network

MAU	Multi access station
MAV	Micro Aerial Vehicles
MSC	Mobile Switching Center
NED	Network Description
NFC	near field communication
NIST	National Institute of Standards and Technology
NOC	network operations center
NTIC	Nouvelles Technologies de l'Information et de la Communication
OLSR	Optimized Link State Routing Protocol
OMNET	Operation and Maintenance New Equipment Training.
OSI	Open systems interconnection
PDU	protocol data unit
Qos	Quality of service
REL	Relation
RFID	Radio frequency identification
RPL	Recognition of Prior Learning
RREP	Route repley
RREQ	Route request
SAN	Storage Area Network
SDN	Software-Defined Networking
SDU	Service data unit
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSID	Service set identifier
SSRA	System Safety Risk Assessment
TCP	Transmission control protocol
TIC	Technologies de l'information
UAV	Unmanned Aerial Vehicles
UDP	User Datagram Protocol
UIT	International Telecommunication Union
UNB	Ultra NarrowBand
VANET	vehicular ad hoc network
VLR	visitor location register
WAN	wide-area network
WANET	wireless ad hoc network
WIFI	Wireless Fidelity
WLAN	Wireless local area network
XML	Extensible Markup Language
ZRP	Zone Routing Protocol

Résumé

Dans les villes intelligentes, de nombreuses infrastructures et appareils sont interconnectés, créant ainsi un réseau complexe de capteurs, de dispositifs de communication et de systèmes de gestion. Ces éléments génèrent et transmettent en permanence d'énormes quantités de données, notamment des informations sur la circulation, les transports en commun, la consommation d'énergie, les déchets, la qualité de l'air, etc. Il est essentiel de collecter, traiter et distribuer ces données de manière efficace pour permettre une prise de décision rapide et informée. Cependant, ces opérations de collection, traitement et distribution nécessitent une communication entre les différents acteurs de la ville intelligentes. La communication dans un tel environnement est basée sur l'envoi et la réception de messages entre un émetteur de la donnée et son récepteur. Donc, il est important de déterminer la meilleure voie pour acheminer les données entre les différents points du réseau, en minimisant les temps de latence, les congestions et les coûts ; c'est le rôle d'un protocole de routage de données. Dans ce travail, nous proposons un nouvel algorithme de routage basé sur le principe d'auto-organisation afin d'améliorer la qualité de service dans les villes intelligentes et assurer une utilisation efficace des ressources dans les communications IoT. Enfin, pour réaliser ce projet, nous avons opté pour le simulateur OMNet++ et le langage de programmation C sous Windows.

Mots clés : Algorithme de Routage, Auto-organisation, Internet des Objets, Qualité de Service, Villes Intelligentes.

Abstract

In smart cities, numerous infrastructures and devices are interconnected, creating a complex network of sensors, communication devices and management systems. These elements constantly generate and transmit enormous quantities of data, including information on traffic, public transport, energy consumption, waste, air quality and more. Collecting, processing and distributing this data efficiently is essential for rapid, informed decision-making. However, these collection, processing and distribution operations require communication between the various players in the smart city. Communication in such an environment is based on the sending and receiving of messages between a data sender and its receiver. So, it's important to determine the best route for routing data between different points in the network, minimizing latency, congestion and costs; this is the role of a data routing protocol. In this work, we propose a new routing algorithm based on the principle of self-organization to improve quality of service in smart cities and ensure efficient use of resources in IoT communications. To carry out this project, we opted for the OMNet++ simulator and the C programming language under Windows.

Keywords: Auto-organization, Internet of things, Quality of Service, Routing algorithm, smart cities.

في المدن الذكية، ترتبط العديد من البنى التحتية والأجهزة ببعضها البعض، مما يؤدي إلى إنشاء شبكة معقدة من أجهزة الاستشعار وأجهزة الاتصال وأنظمة الإدارة. تعمل هذه العناصر بشكل مستمر على توليد ونقل كميات هائلة من البيانات، بما في ذلك معلومات عن حركة المرور والنقل العام واستهلاك الطاقة والنفايات وجودة الهواء الخ. يعد جمع هذه البيانات ومعالجتها وتوزيعها بكفاءة أمرًا ضروريًا لتمكين اتخاذ القرارات في الوقت المناسب. ومع ذلك، تتطلب عمليات الجمع والمعالجة والتوزيع هذه التواصل بين مختلف الجهات الفاعلة في المدينة الذكية. يعتمد الاتصال في مثل هذه البيئة على إرسال واستقبال الرسائل بين مرسل البيانات ومتلقيها. لذلك، من المهم تحديد أفضل طريقة لتوجيه البيانات بين نقاط الشبكة المختلفة، وتقليل أوقات الانتظار والازدحام والتكاليف؛ هذا هو دور بروتوكول توجيه البيانات. في هذا العمل، نقترح خوارزمية توجيه جديدة تستند إلى مبدأ التنظيم الذاتي من أجل تحسين جودة الخدمة في المدن الذكية وضمن الاستخدام الفعال للموارد في اتصالات إنترنت الأشياء. أخيرًا، لتنفيذ هذا المشروع، اخترنا محاكي OMNet++ ولغة البرمجة C في Windows .

الكلمات المفتاحية: إنترنت الأشياء، المدن الذكية، خوارزمية التوجيه، التنظيم الذاتي، جودة الخدمة.

Introduction générale

La majorité de la population mondiale vit désormais dans les villes, selon l'Organisation des Nations Unies (ONU), qui le rapporte depuis plus d'une décennie. De plus, elle déclare que les infrastructures et les ressources actuelles sont à peine suffisantes pour répondre à la demande progressive provoquée par cette croissance démographique et cette concentration géographique. Cette condition peut parfois entraîner d'importants déficits économiques, environnementaux et de ressources naturelles. La question qui se pose aujourd'hui est comment trouver des solutions efficaces pour les différents problèmes qui s'accroissent avec l'augmentation de la population ? La nécessité d'aborder cette question et de trouver des réponses est devenue de plus en plus évidente. En particulier, la réponse serait d'exposer pleinement comment les villes modernes se transforment pour devenir capables d'offrir des services de pointe et d'améliorer la qualité de vie de leurs habitants. Par conséquent, la communauté scientifique, les organismes gouvernementaux et les acteurs du secteur privé ont travaillé ensemble pour faire émerger un nouveau paradigme connu sous le nom de "ville intelligente", qui est basé sur l'intégration des Technologies de l'Information et de la Communication (les TIC) et afin de permettre une utilisation plus efficace des ressources et d'infrastructure existantes. Donc, l'accent est mis sur les solutions TIC en connectant et gérant en mieux les nombreux systèmes existants pour créer un système unique afin de relever les défis futurs [1].

Les villes offrent à leurs habitants une meilleure qualité de vie à mesure qu'ils deviennent plus intelligents. Cependant, la prolifération récente des dispositifs « IoT » (Internet des objets) a joué un rôle important dans le développement de ces villes intelligentes. Afin d'assurer le bon fonctionnement des applications et des services, l'IoT a provoqué une explosion du développement des nouvelles technologies et des appareils de communication connectés. Ces nouveaux appareils ont transformé de nombreux domaines en environnements intelligents, tels que les transports, l'énergie, l'environnement, la santé, la gouvernance, etc. [2].

Les réseaux dans un tel système sont souvent sans infrastructure pour des questions de coûts et de rapidité de déploiement [1]. Par ailleurs, pour les mêmes raisons, chaque nœud du réseau s'alimente à partir d'une batterie dont la capacité est limitée en termes d'énergie. Il convient donc pour ces réseaux particuliers, déployés dans un environnement contraint, de trouver de nouvelles solutions d'organisation et de communication plus adaptées [1].

Dans ce travail, nous étudions le problème de routage de données dans les villes intelligentes en se basant sur un modèle d'auto-organisation adéquat. Notre objectif consiste donc à améliorer la qualité de service dans ce domaine tout en prenant en considération particulièrement la latence des utilisateurs, qui doit être minimale dans un tel système vu ces contraintes critiques tels que l'énergie limitée et la mobilité des appareils (nous utilisant le terme nœud dans le reste du mémoire pour des raisons de simplification).

Le reste de notre mémoire est structuré comme suite :

- Chapitre I « Etat de l'art » : dans ce chapitre nous introduirons le concept de ville intelligente, son architecture et ses applications, les notions rattachées à l'internet des objets (IoT) seront également étudié ainsi que les différentes technologies émergentes et les réseaux qui le constituent.
- Chapitre II « le routage dans les environnements intelligents » : Quant au deuxième chapitre, nous abordons les algorithmes de routage de données dans les villes intelligentes. Les travaux relatifs spécifiquement sont ensuite décrits dans une section spécifique. Notre proposition à la résolution de cette problématique sera également étudié.
- Chapitre III « modélisation et implémentation » : Dans ce chapitre, nous présentons les différentes étapes de l'implémentation et la réalisation de notre travail.
- Nous achèverons avec une conclusion générale et quelques perspectives pour les futurs travaux.

Chapitre I : Etat de l'art

1. Introduction

Le concept de villes intelligentes, ou "smart cities", est de plus en plus présent dans les discussions sur l'avenir de l'urbanisation. Les villes intelligentes visent à améliorer la qualité de vie de leurs habitants en utilisant les technologies de l'information et de la communication (TIC) pour mieux gérer et optimiser divers aspects de la vie urbaine, tels que la mobilité, l'environnement, l'énergie, la sécurité, etc.

Dans une ville intelligente, les objets connectés et les dispositifs IoT (Internet of Things) sont utilisés pour surveiller et contrôler différents aspects de la ville, tels que les systèmes de transport en commun, l'éclairage public, la circulation routière, la gestion de l'énergie, la collecte des déchets, la qualité de l'air, les infrastructures de télécommunications, etc. Les données collectées par ces dispositifs doivent être traitées et analysées en temps réel, ce qui permet de prendre des décisions utiles basées sur des informations précises et actualisées.

Cependant, ces objets sont souvent mobiles et limités principalement en termes de ressources de calcul, stockage et énergie. Au fait, cela augmente les taux de partitionnement du réseau, la perte de données, les interférences et collisions, etc. L'une des solutions les plus utilisées afin d'améliorer ces quantités est l'utilisation de mécanisme d'auto-organisation. L'intervention humaine est réduite au minimum dans un réseau auto-organisé, et cela facilite considérablement son déploiement.

Toutefois, l'auto-organisation n'exclut pas complètement la nécessité d'un certain niveau de contrôle et de planification globale. Il est important donc de combiner les concepts d'auto-organisation et de contrôle centralisé pour atteindre un équilibre optimal entre la flexibilité et la stabilité du réseau.

Dans ce chapitre, nous présentons les concepts et les définitions fondamentaux nécessaires pour comprendre le contexte de notre travail de recherche. Nous commencerons par introduire le concept de ville intelligente, son architecture et ses applications, ainsi que les notions rattachées à l'internet des objets (IoT) et le concept de l'auto-organisation dans les

systèmes informatiques. En fin, nous présenterons les principaux concepts de routage de données dans les réseaux informatiques.

2. Ville intelligente

Dans cette section nous présentons quelques généralités sur les villes intelligentes.

2.1. Définition

Dans la littérature, plusieurs définitions du concept de « ville intelligente » (en anglais Smart City) ont été proposées, parmi lesquelles, nous pouvons citer :

- « Une ville intelligente est celle qui utilise les solutions TIC dans différentes sphères d'activités pour améliorer la gestion de l'infrastructure, garantir de meilleurs services et assurer une haute qualité de vie aux citoyens. L'introduction des solutions TIC permet d'interconnecter les systèmes de santé, de transport, d'énergie, d'éducation, etc., pour former un seul écosystème [3] » (Figure 1).

Aussi nous pouvons dire que :

- « Ce concept met l'emphase sur les nouvelles technologies de l'information et des communications (NTIC). En effet, ces solutions technologiques permettraient de répondre aux difficultés que rencontrent les pouvoirs publics, essentiellement en matière de gestion des infrastructures des grandes villes, pour faire face à la croissance continue de la population » [2].

De même, l'organisation internationale de normalisation (ISO, en anglais : International Organization for Standardization) définit la ville intelligente comme étant :

- « une ville qui augmente le rythme auquel elle obtient des résultats en termes de durabilité sociale, économique et environnementale et qui répond à des défis tels que le changement climatique, la croissance démographique rapide et l'instabilité politique et économique en améliorant fondamentalement la manière dont elle implique la société, applique des méthodes de leadership collaboratif, collabore entre disciplines et systèmes urbains et utilise les données et les technologies modernes pour offrir de meilleurs services et une meilleure qualité de vie aux personnes se trouvant dans la ville (résidents, entreprises, visiteurs), maintenant et dans un futur prévisible, sans désavantage injuste pour d'autres ni dégradation de l'environnement naturel» [4].

Aussi, une nouvelle définition des villes intelligentes a été proposée par « the IEEE IoT Initiative's Smart Cities Working Group » [5] :

- « Une ville intelligente est une zone urbaine qui utilise des services ou des produits technologiques ou non technologiques qui : améliorent le bien-être social et éthique de ses citoyens ; assurent la qualité, la performance et l'interactivité des services urbains afin de réduire les coûts et la consommation de ressources ; et augmentent le contact entre les citoyens et le gouvernement ».

Sachant que l'IEEE (Institute of Electrical and Electronics Engineers) a développé des normes pour les villes intelligentes et leurs différents services. Un exemple spécifique de cette norme est l'ISO 37122, qui est destiné à fournir un ensemble complet d'indicateurs permettant de mesurer les progrès vers une ville intelligente [4].

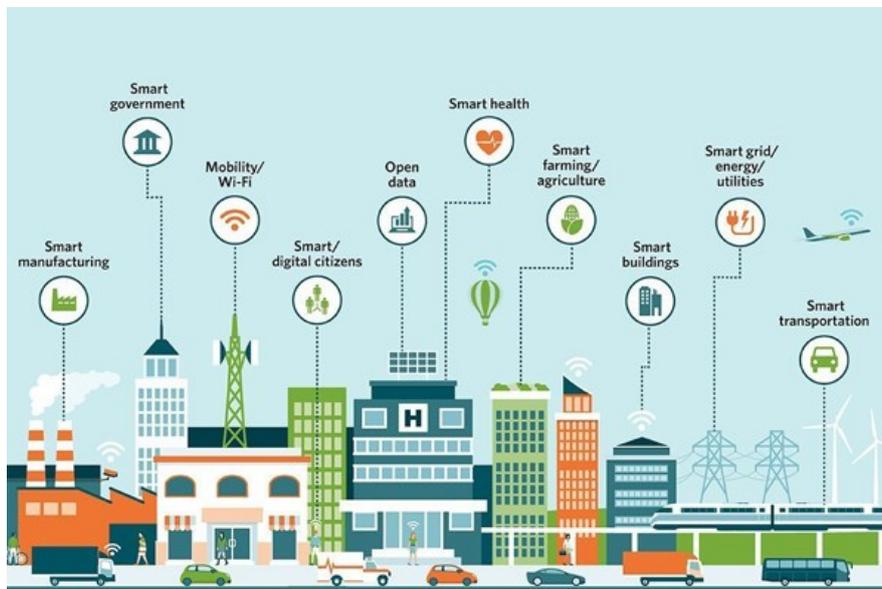


Figure 1. Villes Intelligentes [6].

Aussi, l'intégration transparente de plusieurs technologies et de protocoles différents facilitant la communication entre des équipements hétérogènes est essentielle à l'efficacité opérationnelle des villes intelligentes et à leur croissance [2].

2.2. Architecture des villes Intelligentes

Il n'existe pas une architecture universelle de villes intelligentes qui permettra de faciliter leur déploiement réel à cause des variations des fonctionnalités requises pour chaque ville [2]. Cependant, la majorité des architectures de villes intelligentes proposées dans la littérature comprennent les quatre couches suivantes (Figure 2) : (1) la couche de détection de

données, (2) la couche transmission, (3) la couche gestion de données et (4) la couche application.

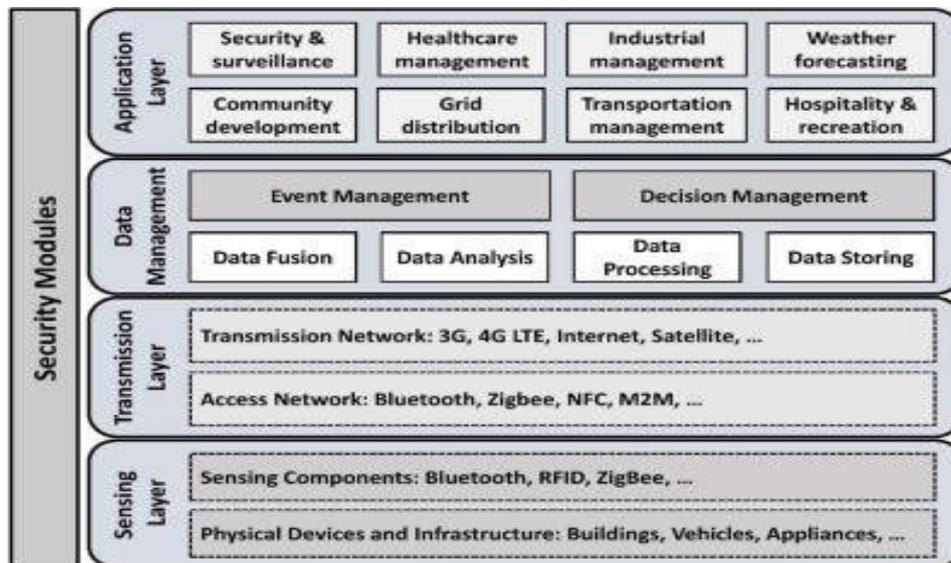


Figure 2. Architecture des villes Intelligentes [7].

- **La couche collecte de données :** L'acquisition de données à partir des appareils physiques hétérogènes est la principale fonctionnalité de cette couche [2]. D'une part, la collecte de données est considérée comme le rôle le plus important car elle contrôle le reste des opérations d'une ville intelligente. D'autre part, elle est considérée comme la tâche la plus difficile en raison de l'immense hétérogénéité des données. De plus, il a été affirmé que la mise en œuvre d'une ville intelligente repose sur toutes les formes de données et de calculs, en raison de leurs importances dans le processus de prise de décision [7].
- **La couche transmission :** Cette couche fournit les données aux couches supérieures grâce à diverses technologies et protocoles de communication [2]. Les différentes technologies de communication utilisées dans le déploiement des villes intelligentes sont illustrées à la Figure 2. Par exemple, elles utilisent les technologies de réseau d'accès : Bluetooth, Zigbee, NFC (near field communication), M2M, RFID et Zwave qui offrent une couverture relativement courte et aussi des technologies pour la transmission réseau comme 3G, 4G LTE (long-term evolution), 5G, et LP-WAN (Low-Power Wide Area Network) qui offrent une couverture plus large.
- **La couche gestion de données :** La couche gestion de données traite et stocke les informations reçues qui sont utiles pour le bon fonctionnement de divers services offerts

par la couche application [2]. En fait, l'efficacité de la couche de gestion des données est vitale pour une ville intelligente durable, car la performance des services de la ville intelligente repose sur la gestion des données. La tâche fondamentale de la couche de données est de maintenir la vitalité des données, en se concentrant sur le nettoyage, l'évolution, l'association et la maintenance des données. Cette couche exécute diverses tâches : (1) de manipulation, (2) d'organisation, (3) d'analyse, (4) de stockage et (5) de prise de décision concernant les données [7].

- **La couche application :** La couche d'application est la couche supérieure de l'architecture de ville intelligente qui sert d'intermédiaire entre les citoyens et la couche de gestion des données. Les performances de la couche d'application influencent fortement le point de vue des utilisateurs et leur satisfaction à l'égard des opérations de la ville intelligente, car elle interagit directement avec les citoyens. En fait, les citoyens sont préoccupés par le comportement intelligent de la ville qui leur offre des services intelligents, par exemple des prévisions météorologiques intelligentes. La couche d'application se compose de divers éléments dans plusieurs domaines. Les principaux services de la couche application sont : le transport intelligent, les prévisions météorologiques, la santé intelligente, le gouvernement intelligent, etc. (voir la section 2.3). Sachant que permettre le partage d'informations, par la couche application, entre différentes applications est considéré comme une approche prometteuse pour l'évolution des villes intelligentes [7].

2.3. Applications des villes intelligentes

Pour transformer une simple ville en une ville intelligente, il est essentiel de mettre en œuvre un effort considérable visant à intégrer des solutions basées sur les TIC dans les secteurs clés de la société tels que la gouvernance, l'économie, le transport, l'environnement et la santé (Figure 3) [2].



Figure 3. Applications des villes Intelligentes [8].

• **Gouvernance intelligente :**

La gouvernance intelligente, également connue sous le nom de gouvernance numérique ou gouvernance électronique, vise à rendre les systèmes de gouvernance plus efficaces, transparents, participatifs et responsables.

L'objectif de la gouvernance intelligente est de transformer les administrations publiques en utilisant les TICs pour fournir des services plus efficaces et répondre aux besoins des citoyens de manière plus rapide et personnalisée. Elle contribue à la création de villes intelligentes en favorisant une gouvernance participative, transparente et orientée vers les résultats, tout en exploitant les avantages des avancées technologiques pour améliorer la qualité de vie des citoyens.

Les technologies utilisées dans la gouvernance intelligente comprennent à titre d'exemple : les plateformes en ligne, les applications mobiles, les systèmes d'information géographique (SIG), les outils d'analyse de données, etc.

- **Économie intelligente :**

L'objectif de l'économie intelligente est de créer un environnement favorable à l'innovation, à la compétitivité et à la durabilité économique. Elle englobe divers aspects, tels que l'utilisation de l'intelligence artificielle, de l'analyse de données, de l'Internet des objets (IoT) et des plateformes numériques pour optimiser les processus commerciaux, stimuler l'entrepreneuriat et favoriser l'innovation. Elle permet également une meilleure intégration des acteurs économiques, y compris les entreprises, les startups, les travailleurs indépendants et les consommateurs, dans l'écosystème économique numérique.

- **Transport intelligent :**

Le transport intelligent joue un rôle crucial dans le développement des villes intelligentes en améliorant la mobilité, la sécurité, l'efficacité et la durabilité des systèmes de transport.

L'importance du transport intelligent réside dans sa capacité à réduire les congestions routières, les temps de déplacement et les émissions de CO₂ grâce à des systèmes avancés de gestion de la circulation, de signalisation intelligente, de gestion des parkings et de planification des transports en commun (Figure 4). Le transport intelligent permet d'optimiser l'utilisation des infrastructures existantes et d'améliorer la fluidité du trafic.



Figure 4. Mobilité des villes intelligentes [9].

- **Éléments environnementaux intelligents :**

L'intelligence est une caractéristique importante des villes intelligentes. Il comprend la qualité de l'air, la gestion de l'eau, le maintien des espaces verts, la surveillance des émissions, la collecte des déchets, l'efficacité énergétique et la préservation des ressources naturelles [2] [3].

L'objectif principal de l'environnement intelligent est de parvenir à un équilibre entre le développement urbain et la protection de l'environnement. Cela implique, à titre d'exemple, la mise en place de solutions innovantes pour la gestion des déchets, l'utilisation efficace des ressources naturelles, la réduction de la pollution, la promotion des énergies renouvelables, la préservation de la biodiversité et la création d'espaces verts et de parcs urbains.

- **Santé intelligente :**

La santé intelligente englobe un nombre important d'applications et de solutions, telles que les dossiers de santé électroniques, les dispositifs médicaux connectés, les applications mobiles de suivi de la santé, la télémédecine, l'intelligence artificielle en santé et l'analyse de données de santé. Ces technologies permettent la collecte, le partage et l'analyse des données de santé pour faciliter le diagnostic, le suivi à distance des patients, etc.

Le traitement de ces dernières entraîne des décisions en temps réel sur l'état de santé des malades [2].

3. Internet des Objets (IoT)

L'IoT est un paradigme qui considère la présence d'une variété d'objets et de dispositifs dans le même environnement et qui peuvent interagir les uns avec les autres grâce à des connexions filaires et sans fil ainsi qu'à des systèmes d'adressage uniques pour créer de nouvelles applications ou de nouveaux services et/ou atteindre des objectifs communs [2].

3.1. Définitions

Dans la littérature, il existe différentes définitions de l'IoT [10] :

- IEEE décrit l'Internet des Objets comme « un réseau d'éléments embarqués avec des capteurs se connectant à Internet ».
- L'Union Internationale des Télécommunications (UIT) définit l'IoT comme « une infrastructure globale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution » [11].

- L'Internet des objets (IoT) est décrite par l'Institut national des normes et de la technologie (NIST) comme « une technologie de systèmes cyber-physiques (CPS) pour connecter des appareils intelligents dans divers secteurs tels que les transports, les soins de santé et l'énergie ».
- Enfin, Cisco dans le secteur commercial définit l'Internet des Objets « IoT » dans le cadre de l' « IoE » (Internet of Everything) comme la technologie qui connecte les personnes, les processus, les données et les choses pour échanger l'information aux expériences, aux capacités et aux avantages économiques.

3.2. Les technologies IoT

Il existe différentes technologies utilisées dans IoT [2] :

- Le ZigBee (IEEE 802.15.4) est une technologie à faible consommation d'énergie qui assure des communications machine à machine en transmettant des informations dans de courte portée.
- L'identification radiofréquence (RFID) utilise la technologie des ondes radio pour envoyer les informations d'un objet.
- La communication en champ proche (NFC) est une connexion à courte portée comme la RFID et qui peut être considérée comme un complément au Bluetooth et 802.11 et qui est principalement utilisée pour connecter les téléphones mobiles à d'autres appareils.
- La technologie Wireless Fidelity (WiFi) utilise les ondes radio pour permettre à plusieurs appareils de communiquer localement.
- La technologie cellulaire 5G et son utilisation dans l'IoT répondent à l'évolution des environnements intelligents et offrira une expérience utilisateur optimale ainsi qu'une amélioration de la qualité, de la fluidité et de la rapidité des transmissions.
 - L'Internet des objets intègre des technologies de transmission à longue distance et à faible consommation d'énergie tels que Sigfox et LoRaWAN.

3.3. Les réseaux IoT

L'IoT est un ensemble de réseaux qui se distinguent par le type de composants, la façon dont ils sont déployés et les technologies utilisées pour transmettre les informations. Il joue un rôle important dans la création de villes intelligentes [2]. Parmi les réseaux IoT les plus connus, on peut citer :

• Réseau Cellulaire :

Afin d'atteindre une exploitation élevée du spectre qui est limité, les réseaux cellulaires déploient des cellules plus petites [2]. La figure 5 montre un schéma de l'architecture d'un réseau cellulaire.

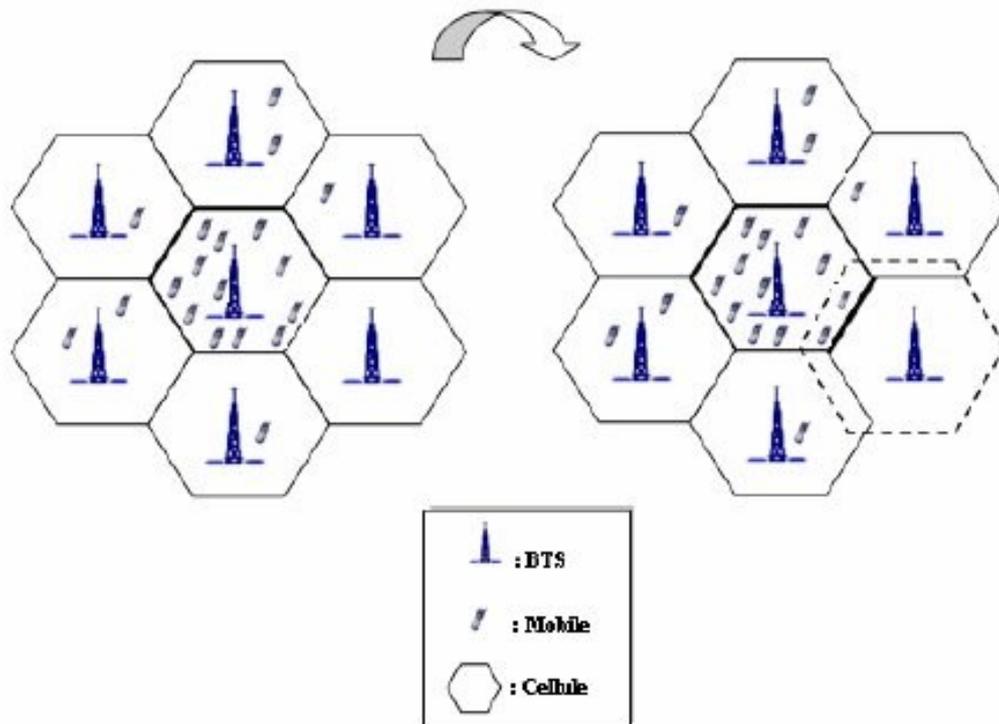


Figure 5. Réseau cellulaire [12].

Un réseau cellulaire est un système de communication sans fil qui permet aux appareils mobiles, tels que les téléphones portables, de se connecter à un réseau de télécommunication. Le réseau cellulaire est basé sur la division de la zone de couverture en petites zones appelées "cellules". Chaque cellule est desservie par une station de base (BTS) et qui sont interconnectées pour former un réseau étendu qui permet la transmission des signaux et des données entre les appareils mobiles et les infrastructures de télécommunication.

Il existe différentes générations de réseaux cellulaires, notamment la 2G, la 3G, la 4G et la 5G, qui offrent des performances et des fonctionnalités améliorées. Chaque génération apporte des améliorations en termes de vitesse de transmission des données, de capacité du réseau, de latence et de compatibilité avec les nouvelles technologies et applications.

• Réseau de Capteurs :

Un réseau de capteurs est un système composé de nombreux capteurs interconnectés via un système de communication sans fil telle que le Wifi et le Bluetooth, qui collectent et transmettent des données de manière collaborative. Chaque capteur est un petit dispositif électronique capable de détecter, mesurer et enregistrer différentes informations sur l'environnement physique, telles que la température, l'humidité, la luminosité, la pression, etc. Un capteur a une intelligence suffisante pour le traitement du signal et pour la diffusion des données [2]. La figure 6 montre un exemple de ce réseau dont les capteurs sont répartis sur une zone géographique.

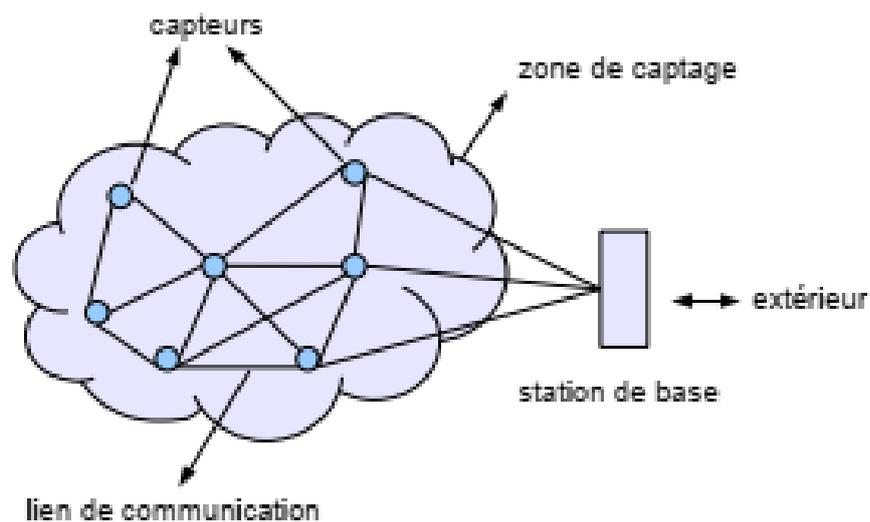


Figure 6. Réseau de capteurs [13].

• Réseau de Mobiles (MANET) :

Un réseau MANET (Mobile Ad hoc Network) est un réseau sans fil autonome et décentralisé, composé de nœuds mobiles interconnectés (figure 7). Dans un MANET, les nœuds, tels que des appareils mobiles, des ordinateurs portables ou des capteurs, peuvent se déplacer librement sans dépendre d'une infrastructure de communication préexistante.

Dans un MANET, chaque nœud peut jouer le rôle de source, de relais ou de destination des données, et tous les nœuds coopèrent pour acheminer les paquets de données à travers le réseau. Les nœuds communiquent directement entre eux ou en utilisant des nœuds voisins comme relais pour étendre la portée du réseau.

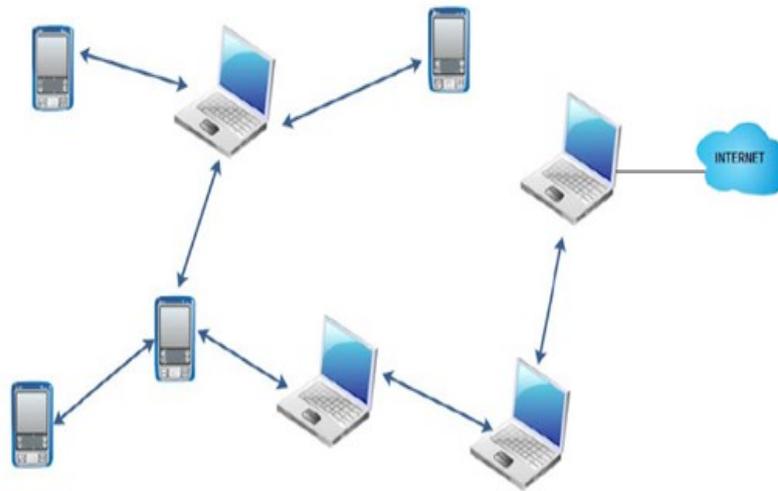


Figure 7. Réseau MANET [14].

Ces réseaux présentent plusieurs caractéristiques tels que : (1) le dynamisme, (2) l'auto-organisation, (3) la décentralisation et (4) la limitation de ressources.

• Réseau Véhiculaire (VANET) :

Un VANET (Vehicular Ad hoc Network) est un type de réseau ad hoc constitué de véhicules équipés de dispositifs de communication sans fil, tels que des capteurs, des antennes et des unités de contrôle (figure 8). Les VANET sont conçus pour permettre la communication entre les véhicules (V2V : vehicle to vehicle) et entre les véhicules et les infrastructures routières (V2I : vehicle to infrastructure), créant ainsi un environnement de communication véhiculaire.

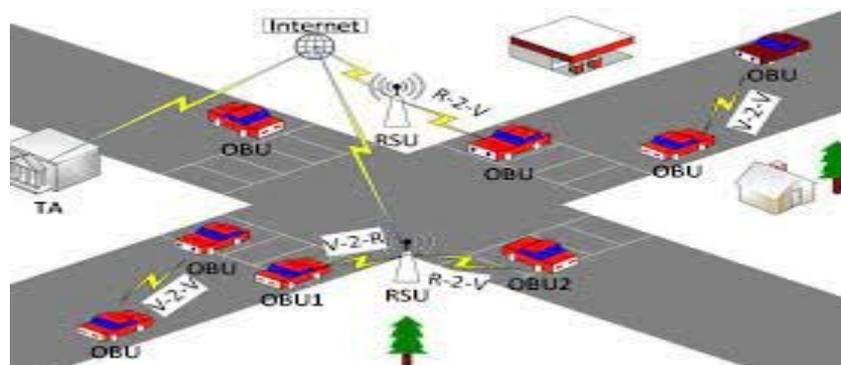


Figure 8. Réseau VANET [15].

Dans un VANET, les véhicules peuvent échanger des informations, telles que des données de localisation, des alertes de sécurité, des informations sur les conditions routières, les feux de signalisation, les accidents, etc. Ces informations sont transmises via des connexions

sans fil à courte portée, généralement basées sur les normes de communication telles que le standard IEEE 802.11p (Wi-Fi dédié aux véhicules) ou les réseaux cellulaires.

Les VANET sont considérés comme une technologie clé pour le développement des véhicules autonomes et de la conduite coopérative, car ils permettent une communication efficace et en temps réel entre les véhicules et l'environnement routier, contribuant ainsi à améliorer la sécurité, l'efficacité et le confort de la conduite.

3.4. Les défis de l'IoT

L'augmentation exponentielle du nombre d'objets connectés a causé également une augmentation exponentielle du volume des données générées par ces objets. Cela a créé de nouveaux défis tels que [16] :

- **L'hétérogénéité des dispositifs et d'environnement:** L'hétérogénéité des dispositifs fait référence à la diversité des appareils connectés, des protocoles et des technologies utilisées au sein d'un réseau IoT. Dans un tel environnement, il peut exister une variété de dispositifs qui sont différents en termes de capacités, de fonctionnalités, de protocoles de communication, de puissance de traitement, de mémoire, de connectivité réseau, etc.
- **Les ressources limitées :** L'IoT s'est principalement développé à l'aide de dispositifs intelligents avec des ressources limitées pour réduire les coûts et conserver des objets compacts. Cela signifie que ces appareils ont des limites en termes de capacité de calcul, de mémoire, d'énergie, de bande passante réseau et de stockage. Pour cela, les dispositifs IoT sont généralement conçus pour être économes en ressources afin de prolonger leur durée de vie opérationnelle et de réduire leur impact sur les réseaux et l'infrastructure.
- **La mobilité :** Jour après jour, les dispositifs intelligents dans l'IoT sont maintenant capables de se déplacer librement. Cela fait référence à la capacité des dispositifs IoT à se déplacer ou à être déployés dans des environnements mobiles et hétérogènes. Les dispositifs IoT mobiles peuvent être intégrés dans des véhicules, des drones, des robots ou même être portés par des individus. Le processus de gestion de la mobilité concerne la gestion des transitions de connectivité lorsque les dispositifs IoT se déplacent d'un réseau à un autre ou d'un point d'accès à un autre. Ce processus vise à assurer une connectivité continue et fiable pour les dispositifs mobiles, en minimisant les interruptions de service et les pertes de données.
- **La sécurité :** Les objets intelligents sont généralement utilisés dans des environnements ouverts sans surveillance. [16]. Aussi, la nature ouverte des réseaux signifie que l'infrastructure est accessible depuis de nombreux points de connexion externes via des

opérateurs mobiles concurrents, Internet et des technologies tierces. Tous ces facteurs risquent d'introduire des failles de sécurité et des vulnérabilités. De plus, étant donné que différents fournisseurs de services partagent l'infrastructure du réseau central, la défaillance d'un seul fournisseur aurait un impact sur l'ensemble de l'infrastructure du réseau. La sécurité dans l'IoT vise à protéger les dispositifs, les réseaux et les données contre les menaces potentielles.

4. Auto-organisation

Un système, composé de plusieurs entités, est dit auto-organisé s'il possède une certaine structure et une certaine fonctionnalité. La structure signifie que les entités de ce système sont organisées d'une manière particulière et qu'elles interagissent ou communiquent entre elles d'une certaine façon. La fonctionnalité signifie que le système, dans son ensemble, remplit une certaine fonction ou assure une certaine tâche [19].

La notion d'auto-organisation joue, un rôle important dans le domaine des réseaux de communication et des réseaux informatiques. Les principaux objectifs sont (1) de développer des protocoles qui facilitent le fonctionnement du réseau, (2) de minimiser le besoin de configuration et (3) de permettre de nouveaux types de réseaux de communication, tels que les réseaux ad hoc et les réseaux de capteurs entièrement décentralisés [19].

Un exemple typique de l'émergence de fonctions auto-organisées se trouve dans le domaine de l'attribution des adresses IP [19]. Traditionnellement, un administrateur configure manuellement chaque ordinateur avec une adresse IP provenant d'un espace d'adressage spécifique qui lui a été attribué par une autorité supérieure. Cette approche traditionnelle ne comporte aucun élément d'auto-organisation ; au contraire, elle nécessite une intervention humaine importante et crée une structure d'adresses très rigide. Un premier pas vers l'auto-organisation a été fait avec l'introduction du protocole de configuration dynamique des hôtes (DHCP : Dynamic Host Configuration Protocol). Ce protocole permet aux ordinateurs d'obtenir automatiquement une adresse IP à partir d'un serveur installé par l'administrateur dans son domaine. Cela permet aux ordinateurs de s'adapter aux changements de leur environnement (par exemple, d'obtenir une nouvelle adresse IP lorsqu'ils changent de réseau) [19].

4.1. Définition et objectifs de l'auto-organisation

4.1.1. Définition et propriétés

Dans la littérature, nous trouvons plusieurs définitions de la notion d'auto-organisation dans les réseaux informatiques. Cependant, la définition la plus courante est la suivante :

« La notion de l'auto-organisation fait référence à l'organisation d'un système sans interaction avec une entité extérieure et sans contrôle centralisé. Ainsi, l'auto-organisation doit nécessairement être basée sur des interactions locales, de façon complètement distribuée » [20].

Par conséquent, un système est dit auto-organisé « s'il est organisé sans aucune entité externe et sans contrôle centralisé. Ainsi, les entités, qui constituent le système, doivent interagir localement avec d'autres entités d'une façon distribuée » [17] (figure 9).

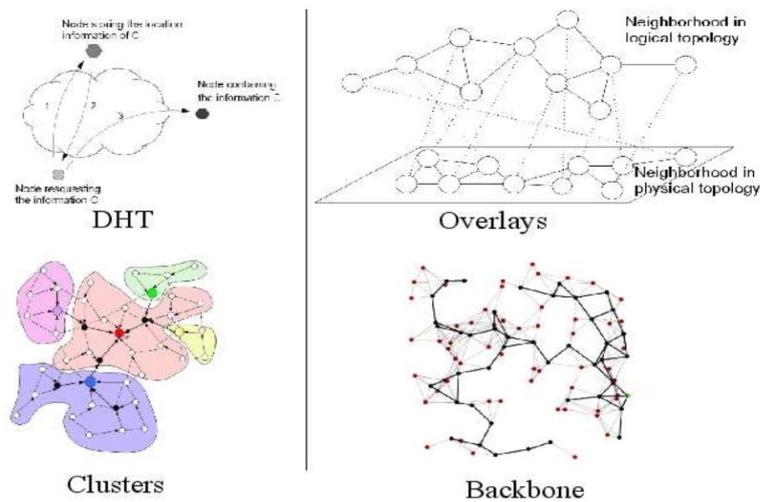


Figure 9. Quelques stratégies de structures d'auto-organisation [20].

Cependant, l'auto-organisation ne se limite pas seulement au contrôle de type distribué ou localisé. Elle concerne la relation entre le comportement des entités individuelles et la structure et la fonctionnalité du système global qui en résultent. Dans les systèmes auto-organisés, l'application d'un comportement plutôt simple au niveau de l'un des nœuds de ce système peut conduire à une réorganisation occasionnelle du système global. Ce phénomène est appelé comportement émergent.

Une caractéristique importante des systèmes auto-organisés est leur capacité d'adaptation aux changements du système ou de l'environnement. En fait, les entités s'adaptent continuellement aux changements de manière coordonnée, de sorte que le système se réorganise

toujours en réaction à différents déclencheurs de changement internes et externes. Ce faisant, il tente de converger vers les structures bénéfiques souhaitées tout en évitant d'autres structures.

Les principaux avantages des systèmes auto-organisées sont :

- La robustesse face aux défaillances et aux pannes.
- Un système auto-organisé continu toujours à fonctionner efficacement même si le nombre de nœuds est très élevé grâce son niveau élevé de scalabilité.
- Aucune congestion due à un nombre important de nœuds dans le réseau ne doit se produire [20].

Les propriétés que doivent posséder un schéma d'auto- organisation sont donc [20] :

- Interactions locales uniquement,
- Emergence d'une structure globale à partir d'informations locales,
- Réactivité aux changements locaux et robustesse,
- Structure d'auto-organisation non orientée source,
- Passage à l'échelle.

Enfin, nous pouvons dire qu'un réseau auto-organisé est souvent connu comme un réseau dont les nœuds s'adaptent à l'environnement en prenant des décisions locales ou distribuées dans le but d'atteindre un objectif global tel que dans le cas des protocoles de routage des données.

4.1.2. Principes et objectifs

L'objectif d'un système auto-organisé est plus que de simples interactions locales au niveau des nœuds sans contrôle extérieur. En particulier il est essentiel, en général, de faire émerger un comportement global à partir des différentes interactions locales [20].

Aujourd'hui, la notion d'auto-organisation joue un rôle important dans le cadre des tous les réseaux informatiques, particulièrement les réseaux mobiles et hétérogènes pour les raisons suivantes [20] :

- Elle permet principalement de faciliter la configuration des nœuds et la mise en place de protocoles de communications.
- Dans un réseau auto-organisé, l'intervention humaine est réduite au minimum ce qui facilite son déploiement.

Dans le cas de protocoles de routage, à partir d'échanges d'informations locales nous s'intéressons à faire émerger un lien virtuel, dit aussi logique, permettant d'assurer les communications inter-processus. Autrement dit, ce lien de communication servira à garantir la transmission de messages entre un émetteur et un récepteur dans le réseau.

Pour atteindre cet objectif, nous aurons besoin de construire deux vues du réseau : une vue locale, représentant l'état du chaque nœud et une vue globale, possédant toutes les vues locales. Bien évidemment, cette vue globale doit être construite en temps borné (en plus, la borne doit être connue) et de façon cohérente [20].

Sachant que, cette structure émergente doit s'adapter à l'environnement et réagir aux changements locaux (par exemple batterie faible, mobilité des nœuds, partitionnement du réseau, etc.). Plus précisément, un changement local ne doit entraîner qu'une modification locale du lien de communication et ne pas impacter sur son intégralité. C'est-à-dire changement d'une partie du lien de communication (la route) et non pas tout le lien (ce qui peut nécessiter une réexécution du protocole de routage). Cette réaction dynamique aux changements locaux en un temps suffisamment court et borné entraîne la propriété d'adaptabilité [20].

Dans un système auto-organisé, le mécanisme d'auto-organisation va permettre de réagir aux cas de défaillance d'un lien ou d'un nœud du réseau (tels que dans le cas de : panne, déconnexion, mobilité, manque d'énergie, ...). Donc, le lien de communication, dans ce cas, va s'adapter et se reconstruire localement [20]. Ce lien disons « logique » sera utilisé tout au long la communication courante et il peut s'adapter afin de garantir la communication en cours.

Nous pouvons considérer que la plupart de réseaux disponibles aujourd'hui dans les villes intelligentes sont auto-organisés car tous les protocoles utilisés sont basés sur des interactions locales entre les nœuds et ils sont répartis sur l'ensemble des membres du réseau.

4.2. Décisions locales versus distribuées ?

Dans les réseaux informatiques, la plupart des protocoles de routage sont basés sur des activités de décision locales et/ou distribuées. Cependant, pour prendre une décision, il est nécessaire d'avoir à disposition les informations adéquates. La notion locale dans un tel système fait référence principalement à la vision partielle qu'il a un nœud du réseau. Sinon, elle est dite distribuée dans le cas où elle est basée sur la vision globale du réseau.

Dans une structure auto-organisée, il est préférable que les décisions soient prises via un processus local basé sur des informations et des interactions locales et non pas de façon distribuée [20] malgré qu'un tel choix ait sûrement plusieurs inconvénients.

Sachant que [20]:

- ✓ Un processus est dit locale si sa décision est prise localement en fonction des informations locales qu'il possède et donc de la vue partielle du réseau qu'il possède.
- ✓ Une décision est dite distribuée si plusieurs nœuds peuvent interagir et converger vers une seule décision. La décision prise n'est donc plus issue de seulement des informations locales mais bien aussi d'un ensemble d'informations distribuées obtenues à l'aide des actions internes et externes.

Notons que choisir une stratégie locale c'est faire le choix de minimiser les informations qui circulent sur le réseau, d'éviter un cout supplémentaire pour synchroniser les décisions, de réagir plus rapidement aux changements locaux, etc. [20] mais cela risque d'augmenter les taux d'erreurs.

4.3. Quelques clefs pour l'auto-organisation

Les principes clefs de l'auto-organisation sont cité dans [19] en spécifiant qu'il est nécessaire de définir quelles sont les règles et les protocoles nécessaires pour tirer parti de l'interaction entre les nœuds du réseau [20]. Quatre paradigmes sont à discutés [20] :

- Emergence d'un comportement global à partir de règles locales.
- Interactions locales et coordination des nœuds.
- Reduction des informations sur l'état du réseau.
- Adaptation dynamique à l'environnement.

5. Conclusion

Dans ce chapitre, nous avons présenté quelques définitions et concepts de base qui sont en relation avec notre travail tels que les villes intelligentes, l'IoT ainsi que le mécanisme de l'auto-organisation.

Dans le prochain chapitre, nous abordons le problème de routages de données dans les environnements intelligents.

Chapitre II : Le routage dans les villes intelligentes

1. Introduction

Dans les villes intelligentes, de nombreuses infrastructures et appareils sont interconnectés, créant ainsi un réseau hétérogène et complexe de capteurs, de dispositifs de communication, de systèmes de gestion, etc. Ces éléments génèrent et transmettent en permanence d'énormes quantités de données, notamment des informations sur la circulation, les transports en commun de citoyens, la consommation d'énergie, les déchets, la qualité de l'air, etc. Il est essentiel de collecter, traiter et distribuer ces données de manière efficace pour permettre une prise de décision rapide et informée.

Cependant, ces opérations de collecte, traitement et distribution nécessitent une communication entre les différents acteurs de la ville intelligentes tels que les appareils utilisateurs les stations de traitement et de stockage de données. La communication dans un tel environnement est basée sur l'envoi et la réception de messages entre un émetteur de la donnée et son récepteur. Donc, il est important de déterminer la meilleure voie pour acheminer les données entre les différents points du réseau, en minimisant les temps de latence, les congestions et les coûts ; c'est le rôle d'un protocole de routage de données. Ce processus est essentiel pour garantir une communication fluide et fiable entre les différents acteurs du système urbain, tels que les autorités municipales, les entreprises, les résidents et les services publics.

La problématique du routage de données dans les villes intelligentes présente plusieurs défis. Tout d'abord, en raison de la complexité et de la diversité des infrastructures et des appareils connectés, il est nécessaire de développer des algorithmes de routage adaptés à chaque contexte spécifique. De plus, les contraintes de temps réel et la demande croissante en termes de débit de données nécessitent des solutions de routage à faible latence et à haute capacité. Aussi, la mobilité des acteurs des villes intelligentes est une contrainte importante à ne pas négliger lors le choix d'un algorithme de routage de données. En effet, un simple déplacement de l'un des composants de la ville peut affecter les performances de l'algorithme en augmentant la latence par exemple ou encore la perte de données suite à un partitionnement du réseau à cause de ce déplacement.

Dans ce chapitre, nous abordons les éléments liés à la problématique du routage des données dans les villes intelligentes en discutant le rôle des algorithmes de routage, leurs types et métriques. Nous présentons aussi les principaux travaux connexes qui ont été réalisés dans ce domaine. Puis nous discutons notre contribution et nous concluons par une conclusion.

2. Routage de données

Les réseaux informatiques actuels d'une façon générale et les villes intelligentes particulièrement disposent d'énormes et multiples sources d'informations transmises par un grand nombre d'équipements interconnectés par des systèmes de communication en général hétérogènes [2].

Le protocole de routage est central pour le bon fonctionnement de tout système de communication multi-sauts[2]. Le premier objectif d'un protocole de routage est : (1) d'établir une route entre une source et une destination, (2) de maintenir une trace sur l'existence de cette route et (3) de faciliter le bon acheminement des données le long de la route retenue [2].

Nous pouvons distinguer plusieurs défis que rencontre un protocole de routage dans les réseaux sans fil et plus particulièrement dans les villes intelligentes, tels que, comme dans le cas de l'IoT : (1) la mobilité des nœuds, (2) la limitation de ressources (3) l'hétérogénéité de dispositifs et réseaux et (4) la sécurité.

La problématique du routage de données dans les villes intelligentes est complexe. En effet, les réseaux dans un tel environnement sont souvent peu stables [2] parce que :

- Les nœuds sont mobiles ;
- Les nœuds peuvent entrer et sortir du réseau à tout moment, soit parce qu'ils s'éteignent (batterie faible, à cause d'une défaillance de type hardware ou software, etc.) soit parce qu'ils sortent de la portée radio de nœuds du réseau;
- Les ressources des nœuds sont souvent limitées (capacité de calcul, énergie...) car ce sont des équipements embarqués légers et mobiles ;
- L'hétérogénéité des nœuds et réseaux.
- La sécurité, etc.

2.1. Types de routage

La disposition des nœuds, les nœuds du voisinage et la façon de transmettre l'information sont les principales notions qui servent à classer les protocoles de routage dans les réseaux. Parmi les classes de protocoles de routage les plus connues, nous citons [2] :

- **Basé sur la topologie** : Ce type de protocole utilise des informations du lien stockées dans une table dédiée au routage. Il existe deux types de routage basé sur la topologie : le routage proactif et le routage réactif.
 - **Les protocoles de routage proactifs** : sont basés sur l'existence de tables de routage au niveau de chaque nœud. Lors l'envoi d'un message, l'émetteur consulte sa table pour connaître la route à suivre jusqu'au destinataire de ce message. Donc, les nœuds connaissent à tout moment comment d'atteindre les autres nœuds du réseau. Cependant, il faut être capable de mettre à jour les tables de routage périodiquement à cause de la mobilité des nœuds, cela nécessite la diffusion de beaucoup de messages de contrôle et engendre du trafic supplémentaire réduisant ainsi la bande passante disponible, consommant plus d'énergie et augmentant la latence [18].
 - **Les protocoles de routage réactifs** : ne maintiennent pas en permanence des tables de routage de l'ensemble du réseau. Lorsqu'un nœud a besoin d'envoyer un message vers un autre élément du réseau, il commence par déterminer une route lui permettant d'atteindre le destinataire du message. Les nœuds du réseau ont une vision partielle du réseau. Les routes étant déterminées à la demande, ce mode de fonctionnement permet de réduire le nombre des messages de contrôle, mais engendre des temps de latence lorsqu'un nœud veut envoyer pour la première fois un message vers un autre nœud ou si la route en mémoire n'est plus valide [18].
 - **Protocoles de routage hybrides** : Ces protocoles combinent des aspects des protocoles proactifs et réactifs pour offrir une flexibilité et une optimisation des performances. Ils utilisent à la fois des tables de routage préétablies et des calculs de routage à la demande. Exemple de protocole hybride : ZRP (Zone Routing Protocol).
- **Basé sur la géographie** : Cette classe d'algorithme utilise les informations de localisation géographique des nœuds dans un réseau pour déterminer les chemins de communication optimaux. Dans ce type d'algorithme, chaque nœud est identifié par ses coordonnées géographiques, telles que sa latitude et sa longitude. Son principe permet de choisir les voisins les plus proches comme prochains sauts ce qui réduit les délais de transmission et la consommation d'énergie nécessaire pour les communications. Les nœuds peuvent donc évaluer la distance physique entre eux et prendre des décisions de routage en fonction de ces mesures.

Ainsi, ce type de protocole est considéré comme le plus adapté aux réseaux à grande échelle [2].

- **Basé sur les groupes ou clusters :** les nœuds d'un réseau dans ce cas sont regroupés en clusters ou en groupes logiques. Chaque cluster est composé d'un nœud central appelé le nœud chef de cluster, qui agit comme un point central de coordination pour les nœuds du cluster. Les décisions de routage sont prises au niveau du nœud chef et il est responsable de la collecte des informations de routage à partir des nœuds du cluster, de la prise de décisions de routage en fonction de ces informations et de l'acheminement des paquets vers les destinations finales. Ce type de protocole a un avantage majeur dans la diminution de la charge du réseau et les temps de livraison des paquets durant leur transmission dans le réseau.
- **Basé sur la diffusion :**
Ce type de protocole est simple à concevoir et il se base sur la diffusion des messages de proche en proche [2]. Donc, les nœuds d'un réseau diffusent les paquets de données à tous les nœuds voisins, qui les diffusent à leur tour jusqu'à ce que le paquet atteigne sa destination. Cependant, l'inconvénient majeur de cette famille d'algorithmes est qu'il risque d'entraîner des boucles de diffusion ce qui peut augmenter aussi la congestion réseau.
- **Basé sur la qualité de service (QoS) :**
Les villes intelligentes prennent en charge un ensemble d'applications de sûreté, sécurité et de confort dans les différents systèmes existants. Ces applications ont des exigences variées de qualité de services telles que le délai de transmission et la perte de données. Par conséquent, elles imposent des communications de données par des protocoles de routage efficaces basés sur la qualité de service et dans les limites requises afin de satisfaire le fonctionnement optimal de cette diversification d'attentes. En cas de non-conformité, les conséquences peuvent être périlleuses.
- **Basé sur l'optimisation par des algorithmes heuristiques :**
Dans le contexte du routage, les algorithmes heuristiques cherchent à trouver des chemins de routage qui optimisent certains critères, tels que la latence, la consommation d'énergie, la bande passante, etc. Ces algorithmes utilisent des techniques d'exploration de l'espace de recherche pour évaluer différentes combinaisons de chemins et sélectionner la meilleure solution possible, même si elle n'est pas nécessairement la solution optimale absolue.

2.2. Métriques de routage

Le calcul de la métrique de routage repose sur une multitude d'informations qui sont soit locales ou distribuées (collectées sur le réseau). Ces informations peuvent inclure [2] :

- La topologie du réseau, en considérant le nombre de voisins, le nombre de sauts ;
- La puissance du signal ;
- La qualité de lien en termes de durée, stabilité ;
- La bande passante disponible ;
- La distance séparant les nœuds ;
- Les paramètres de mobilité des nœuds.
- En plus des paramètres de la QoS, telle que la latence.

3. Travaux connexes

Plusieurs travaux de recherche ont proposé, adapté et/ou évalué plusieurs algorithmes de routage de données dans les environnements mobiles et intelligents. Parmi lesquels nous pouvons citer :

1. **RPL (Routing Protocol for Low-Power and Lossy Networks)** : RPL est un protocole de routage spécialement conçu pour les réseaux à faible consommation d'énergie et à perte de paquets, tels que les réseaux de capteurs sans fil. Il utilise une topologie en arbre pour acheminer les données vers une passerelle ou un nœud central. RPL est efficace en termes de consommation d'énergie et prend en compte les contraintes des dispositifs IoT [21] [22] [23].
2. **AODV (Ad hoc On-Demand Distance Vector)** : AODV est un protocole réactif couramment utilisé dans les réseaux ad hoc, y compris l'IoT. Il établit des routes à la demande lorsque des nœuds souhaitent communiquer. AODV est adapté à la mobilité des nœuds et permet de réduire la surcharge de routage en établissant des chemins plus courts [24] [25] [26].
3. **CoAP (Constrained Application Protocol)** : CoAP est un protocole de communication léger pour les dispositifs à ressources limitées dans l'IoT. Il est basé sur les protocoles Web (HTTP) et permet aux dispositifs de communiquer avec des serveurs en utilisant des requêtes GET, POST, PUT et DELETE. CoAP est souvent utilisé pour la communication entre dispositifs IoT et serveurs cloud [27].
4. **MQTT (Message Queuing Telemetry Transport)** : MQTT est un protocole de messagerie léger utilisé pour la communication machine à machine (M2M) dans l'IoT.

Il suit un modèle de publication/abonnement, où les dispositifs publient des messages sur des sujets spécifiques et les autres dispositifs s'abonnent à ces sujets pour recevoir les messages. MQTT est efficace en termes de consommation d'énergie et de bande passante [28].

5. **OLSR (Optimized Link State Routing)** : OLSR est un protocole de routage proactif qui utilise des informations sur l'état des liens pour calculer les routes. Il est adapté aux réseaux ad hoc, y compris l'IoT, et prend en compte la mobilité des nœuds. OLSR réduit la surcharge de routage en établissant des liens multipoints et en évitant les boucles [29] [30] [31] [26].
6. **DSDV (Destination-Sequenced Distance Vector)** : DSDV est un protocole de routage proactif basé sur un vecteur de distance séquentiel de destination. Il est utilisé pour les réseaux sans fil, y compris l'IoT, et maintient une table de routage pour chaque nœud, qui est mise à jour périodiquement. DSDV est adapté aux environnements avec des nœuds statiques et des routes stables [32] [26].
7. **LORA (Long Range)** : LoRa est une technologie de communication sans fil à longue portée conçue pour les applications IoT. Elle utilise des protocoles de routage spécifiques, tels que LoraWAN, pour permettre la connectivité à bas débit et à longue distance. LoRa est efficace en termes de consommation d'énergie et permet une large couverture géographique [33][34][35].
8. **ZRP (Zone Routing Protocol)** : ZRP est un protocole hybride qui combine des aspects des protocoles proactifs et réactifs. Il divise le réseau en zones et utilise des protocoles proactifs pour les nœuds statiques à l'intérieur des zones et des protocoles réactifs pour les nœuds mobiles et les routes interzones. ZRP est adapté aux réseaux hétérogènes avec des nœuds statiques et mobiles [36].
9. **LEACH (Low Energy Adaptive Clustering Hierarchy)** : LEACH est un protocole de routage adapté aux réseaux de capteurs sans fil dans l'IoT. Il utilise une approche de clustering, où les capteurs sont regroupés en clusters avec des nœuds leaders rotatifs. LEACH permet d'économiser l'énergie en activant périodiquement les nœuds leaders pour agréger les données et les transmettre à la passerelle [37][38][39][40].
10. **DYMO (Dynamic MANET On-demand)** : DYMO est un protocole de routage réactif conçu pour les réseaux ad hoc mobiles, y compris l'IoT. Il établit des routes à la demande en utilisant des messages de routage diffusés dans le réseau. DYMO est adapté à la mobilité des nœuds et permet d'économiser la bande passante en utilisant des métriques de routage simplifiées [31].

11. **L'algorithme SSRA (Smart and self-organised routing algorithm)** : est proposé dans [41]. Son objectif est assuré une communication efficace dans les villes intelligentes. Cet algorithme utilise le principe de l'auto-organisation et il capable d'utiliser les expériences des routes déjà retenues afin d'améliorer ses expériences et améliorer ses performances. Un travail similaire dans [42], consiste à l'amélioration de la couverture dans les réseaux d'objets mobiles connectés.

4. Éléments de la problématique

D'après les discussions des travaux connexes présentés dans la section précédente, nous remarquons que la décision ou le choix de la route et en général basée sur nombre important de paramètres ou nécessite une quantité importante de données tels que dans [41][42] et/ou ressources de calcul, ce qui augmenter le temps de calcul et donc la prise de décision tels que dans les travaux [37][38][39][40]. Cela peut affecter négativement les performances des protocoles de routage particulièrement en termes de qualité de service. Toutefois, en mettant l'accent sur la QoS, les villes intelligentes peuvent offrir une meilleure qualité de vie aux résidents, optimiser les services urbains et promouvoir le développement durable.

Aussi, d'autres algorithmes prennent en considération un nombre réduit de paramètres tels que dans [21] [22] [23] 12, dont le choix de la route est basé principalement sur les deux facteurs d'énergie et la perte de paquets. Cependant, dans un tel domaine il est important de réduire également la latence en choisissant par exemple les routes les plus courtes et les moins surchargées. Aussi, nous s'intéressons à maximiser la durée de validité ou de vie de la route retenue ; il est essentiel donc de choisir la route la plus stable par exemple, etc.

De même, dans les travaux [24] [25] [26], le choix est basé principalement sur la route la plus courte. Cela peut conduit à une surcharge des liens les plus courtes ; ce qui risque aussi augmenter leurs latences et donc la latence moyenne du réseau.

Les travaux [27] et [28] présentent de algorithmes de routage très simple mais aussi qui ont montré leurs efficacités dans un nombre restreint de réseaux et qui ne peuvent pas être utilisés dans n'importe quel réseau, spécialement dans les réseaux homogènes dans les villes intelligentes, ce qui présente la limite majeure de ces algorithmes.

En plus, le protocole OSLR et ces versions dans [29][30][31] et [32] sont des protocoles proactifs qui sont basés sur l'existence de tables de routage au niveau de chaque nœud. La mise à jour les tables de routage périodiquement nécessite la diffusion de beaucoup de messages de

contrôle et engendre du trafic et des traitements supplémentaires. Ce qui conduit à une dégradation de performances du réseau principalement en termes de la bande passante disponible, la consommant d'énergie et la latence.

Aussi, l'algorithme DYMO [31] peut rencontrer des défis lorsqu'il s'agit de gérer efficacement la mobilité des nœuds. Lorsque les nœuds se déplacent fréquemment, il peut y avoir des perturbations dans la connectivité et des déconnexions temporaires. Étant donné que DYMO est un protocole de routage à la demande, il nécessite une découverte de route lorsqu'un nœud souhaite communiquer avec un autre nœud. Cela augmente le trafic réseaux du routage, la consommation d'énergie et la latence.

Ce travail vise à améliorer la qualité de service dans les protocoles de routage de données dans les villes intelligentes, tout en prenant en considération les principales et les simples caractéristiques et contraintes d'un tel environnement comme la mobilité continue des nœuds, l'énergie limitée, etc. Cette amélioration de la qualité de service est basée principalement sur :

- La réduction du nombre de message de control et du Traffic réseau.
- La réduction de la latence des clients.
- La minimisation de la consommation d'énergie.
- L'optimisation de l'algorithme de routage en termes de nombre de critères de choix et temps de calcul sans dégradation de son efficacité ni fiabilité.
- Augmentation des durées de vies des routes retenues.
- Equilibrage de charge entre les différentes routes possibles dans le même réseau.

5. Contribution

Dans cette section nous présentons les principes de base de notre solution.

5.1. Description de la solution proposée

Notre proposition consiste à l'optimisation des performances de routage et de la fiabilité de la connectivité dans les réseaux dynamiques et hétérogènes dans les villes intelligentes. Donc, nous donnons la priorité à un routage hybride, intelligent et auto-organisé afin de réduire la surcharge de contrôle et la dépense énergétique. L'intelligence fait référence à la décision locale et collaborative tout en assurant une coordination entre les différents nœuds participants au processus de communication ce qui permet une optimisation continue du routage en réponse aux différentes conditions du réseau.

Notre proposition est basée principalement sur les points suivants :

1. Dans l'étape de la découverte de route, nous avons utilisé le principe des algorithmes de vague centralisé de « Écho de Segall, 1983 » connus dans les systèmes distribués (voir algorithme 1) avec un mécanisme de prévention contre les boucles afin de réduire le nombre de message générés par notre algorithme.

Algorithme1 : algorithmes de vague centralisé de « Écho de Segall, 1983

```
Var recp : entier init 0
Var father : processus init udef
Pour l'initiateur :
    For all q ∈ Neighp Do émettre (jeton) vers q
    WHILE recp < #Neighp Do recevoir (jeton) : recp := recp + 1
    décider()
Pour les non-initiateurs :
    recevoir (jeton) de q : fatherp := q ; recp := recp + 1
    For all q ∈ Neighp ∧ q ≠ fatherp do émettre (jeton) vers q
    While recp < #Neighp do recevoir (jeton) : recp := recp + 1
    émettre (jeton) vers fatherp
```

En effet, après la diffusion d'un paquet de découverte par le nœud émetteur, et sur réception du premier message, les nœuds voisins commencent par enregistrer les informations concernant l'émetteur de ce paquet (appelé Father dans l'algorithme de Écho de Segall), puis le faire suivre vers tous leurs voisins excepte vers celui duquel il vient (le Father). Puis lorsqu'un nœud reçoit un message de tous ses voisins, le message est renvoyé en écho au processus Father.

Cependant, dans notre algorithme cette opération est répétée pour chaque réception d'un message de découverte. Lorsque le nœud émetteur reçoit un message de tous ses voisins, il calcul le score de la route courante en fonction des informations locales et distribuées, il l'insère dans le message puis il renvoie le message le score en écho à son processus Father.

Quand le message qui contient le score arrive chez l'émetteur du message de la découverte, il décide sur le chemin à utiliser pour sa communication en fonction de tous les scores reçus.

2. De plus, dans notre algorithme nous avons utilisé une table de routage partielle, au niveau de chaque nœud du système, contenant toutes les informations concernant les chemins découverts. Cette table est utilisée afin de (1) maintenir une trace sur l'existence de cette route et (2) faciliter le bon acheminement des données le long de la route retenue. Les routes dans la table partielle possèdent une durée de vie afin de pouvoir les utiliser pour les prochaines communications tant que cette durée est supérieure à 0.
3. De même, et pour augmenter la durée de validité de la route retenue, nous avons pris en considération des informations de mobilité des nœuds. Plus précisément, le sens de la mobilité de chaque nœud ainsi que sa vitesse.

En effet, les nœuds participant à la communication courante, y compris l'émetteur et le récepteur, s'adaptent automatiquement au changement de leurs environnement et s'organisent automatiquement de la façon suivante :

- Chaque nœud contrôle la vitesse et le sens de son déplacement, et s'il trouve qu'il s'éloigne de la route retenue, il ralenti sa vitesse comme suit : si (1) le sens de déplacement est l'inverse de la route retenue et aussi (2) l'angle formé de l'axe de l'émetteur et le récepteur ainsi que l'axe de l'émetteur et ce nœud courant est supérieur à 60° ; alors ralentir la vitesse pour une période temps connue. Les valeurs de ralentissement de vitesse ainsi que sa période sont générés aléatoirement dans ce travail mais ils seront sujet d'autre futurs travaux afin d'étudier leurs impacts sur les performances de notre protocole de routage.
4. Aussi, nous avons inséré la latence, l'énergie disponible, le nombre de saut, la distance et la stabilité de chaque route dans le score de décision afin d'améliorer la qualité de service dans un tels domaine. Nous utilisons une formule très simple dans l'objectif et de réduire le nombre de calcul. Cette formule est similaire à la formule utilisée par SSRA [41] sauf que dans notre cas nous rajoutons la latence de chaque route et aussi nous avons travaillé avec le nombre de sauts au lieu du nombre des nœuds dans comme suit :

$$\text{Score} = (\text{stabilité} * \text{distance} * \text{énergie} * \text{latence}) / \text{nombre de sauts}$$

Équation 1 : calcul des scores au niveau du récepteur.

Pareillement que dans [41], l'émetteur choisi la route avec la valeur maximum du facteur « F » calculé comme suit :

$$F = \text{score}_i / \sum_1^{\text{nombre de noeuds}} \text{score}_j$$

Équation 2 : calcul du facteur du décision au niveau de l'émetteur.

Il est important de noter que dans notre travail, nous favorisons une décision locale vu son importance discutée dans le chapitre I. Notons aussi que cette décision peut être prise au niveau de n'importe quel nœud du réseau ; ça peut être un émetteur, un récepteur, un nœud intermédiaire, etc. ou même un nœud non concerné par la communication courante (il n'appartient à aucun chemin menant à la destination). Toutefois, nous avons opté pour l'émetteur pour des raisons de simplification et d'optimisation de notre algorithme. L'émetteur est supposé avoir toutes informations (locales) qui sont déjà récupérées du réseau en se basant sur le principe de collaboration suite à une étape de découverte tel qu'il est expliqué dans la section suivante. Ces informations sont jugées importantes pour atteindre notre objectif (voir la section suivante pour plus de détails).

Cependant, le calcul des scores se fait au niveau de récepteur. Cela aidera à alléger la charge de l'émetteur en réduisant les calculs centralisés et en les distribuant sur le réseau.

5.2. Algorithme proposé

L'algorithme que nous proposons pour le routage de données dans les villes intelligentes est le suivant :

Algorithme2 : Algorithme proposé

```

Var recp : entier init 0
Var father : processus init undef
/**Pour l'émetteur :
insérer ( (Position, vitesse, énergie, séquence ...), jeton) ;
WHILE recp = 0 Do
Begin
    For all q ∈ Neighp Do émettre (jeton);
    WHILE temps < temps-attente Do
        Recevoir (score, jeton) :
        recp := recp + 1
End
Calculer(latence) ;
Calcul-facteur-F () ; /**avec l'équation 2
Meilleure-route = routemax(F) ;
Mettre-à-jour-table-de-routagep() ;
Communication () ; /**en utilisant la meilleure route

```

Algorithme2 : Algorithme proposé (suite)

```
/***/Pour les nœuds intermédiaires
IF Recevoir (jeton) de q THEN
fatherp := q ;
if temps....
Mis-à-jour-de-la-table-de-routagep(),
insérer ( (distance, energie, stabilité ...), jeton) ;
For all q ∈ Neighp ∧ q ≠ fatherp do
    emmettre (jeton) vers q
IF Recevoir (score, jeton) de q THEN
    Mis-à-jour-de-la-table-de-routagep() ;
    IF p existe(speedcontrol) THEN
        Ralentir-Vitesse () ;
    Envoyer (score, noeud) au fatherp ;
/***/pour le nœud destinataire
Recevoir ( jeton) de q ;
fatherp := q ;
Calcul (distance, Energie, stabilité),
Insérer ( (distance, energie, stabilité ...), jeton) ;
IF angle (direction de déplacement, direction de communication) > 60 Then
    Insérer (id, speedcontrol) ;
Calcul-score-route-courante () ;
Envoyer (score, jeton) vers le fatherp ;
```

5.3. Les avantages de notre contribution

Parmi les avantages de notre solution, nous pouvons citer :

- Augmenter la durée de validité des chemins de routage retenus.
- Tolérance aux pannes.

- Tolérance aux partitionnements vue la dégradation de la vitesse de déplacement des nœuds.
- Scalabilité.
- Amélioration de la qualité de service en réduisant de la latence moyenne, la consommation d'énergie, etc.

6. Conclusion

Dans ce chapitre, nous avons discuté le routage de données dans les environnements intelligents en mettant l'accent sur les principaux types des algorithmes de routage et leurs métriques. Ensuite nous avons présenté plusieurs travaux connexes et une discussion des éléments de notre problématique. Puis, nous avons présenté notre contribution et une conclusion.

Dans le prochain chapitre, nous présenterons les différentes étapes de l'implémentation et la réalisation de notre application.

Chapitre III : Implémentation et réalisation

1. Introduction

La phase d'implémentation et de réalisation joue un rôle clé dans la concrétisation des systèmes informatiques. Elle permet de transformer les idées en réalité, de détecter les erreurs, de résoudre les problèmes et de garantir la qualité et la performance du système final.

Dans ce chapitre, nous présentons un modèle général du système implémenté, trois organigrammes d'exécution pour les nœuds émetteur, récepteur et nœuds intermédiaires, ainsi qu'une description générale du simulateur utilisé. Ensuite, nous présentons une description détaillée de notre application réalisée et les paramètres de notre simulation. Puis nous montrons un scénario d'exécution et toutes les interfaces graphiques en relation et enfin nous terminerons par une conclusion.

2. Modèle du système

Le modèle de notre système est composé des éléments suivants (figure 10) :

- Un ou plusieurs réseaux homogènes ou hétérogènes (les réseaux dans les villes intelligentes sont généralement hétérogènes). Chaque réseau est caractérisé par : une capacité qui définit le nombre maximum des nœuds qui peuvent se connecter en parallèles à ce réseau, le nombre des nœuds connecté, etc.
- N nœuds mobiles, dont l'un des nœuds est un émetteur et un autre nœud récepteur. Chaque nœud possède quelques caractéristiques tels que : l'énergie disponible, position, etc.
- Des liens de communication (logiques ou physiques) caractérisés par : une distance qui sépare deux nœuds voisins, une latence, etc.
- Un ou bien plusieurs sauts ou disons nœuds intermédiaires existe sur le lien de communication logique entre un émetteur donné et un récepteur.

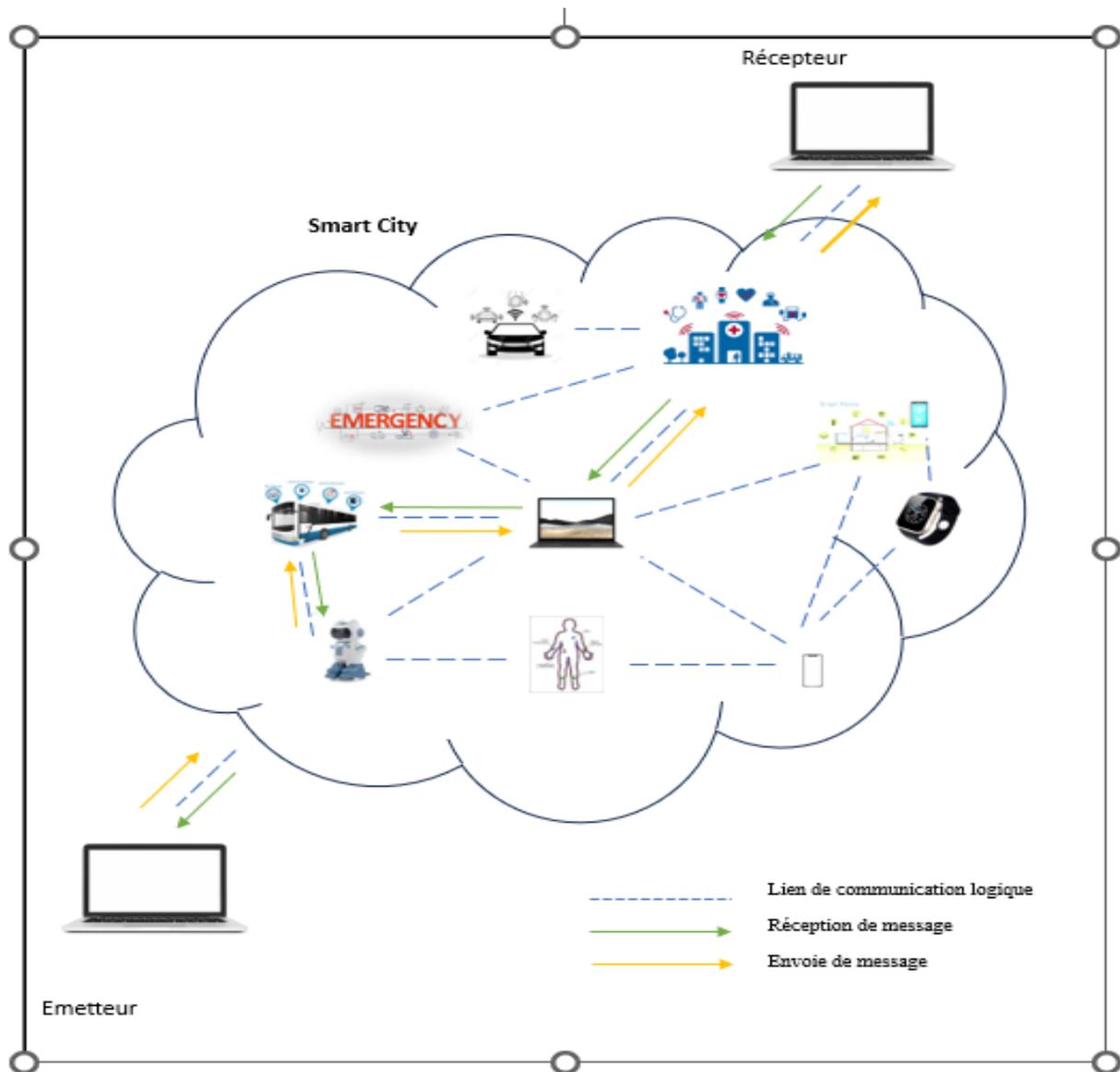


Figure 10. Modèle du système.

3. Organigramme d'exécution

Les organigrammes suivants détaillent le processus d'exécution complet de notre algorithme, ils expliquent en détail l'ensemble des processus exécutés au niveau de notre système sur le nœud émetteur, le nœud récepteur et les nœuds intermédiaire (figures 11, 12, 13). Sachant que dans un tel environnement, un nœud donné peut être à la fois un émetteur pour une communication donnée, un récepteur pour une deuxième communication et intermédiaire pour une troisième. Ainsi, les trois diagrammes existent sur tous les nœuds et appartiennent au même algorithme discuté dans la section précédente. La division donc est pour faite pour des raisons de simplification.

- Au niveau du nœud émetteur :

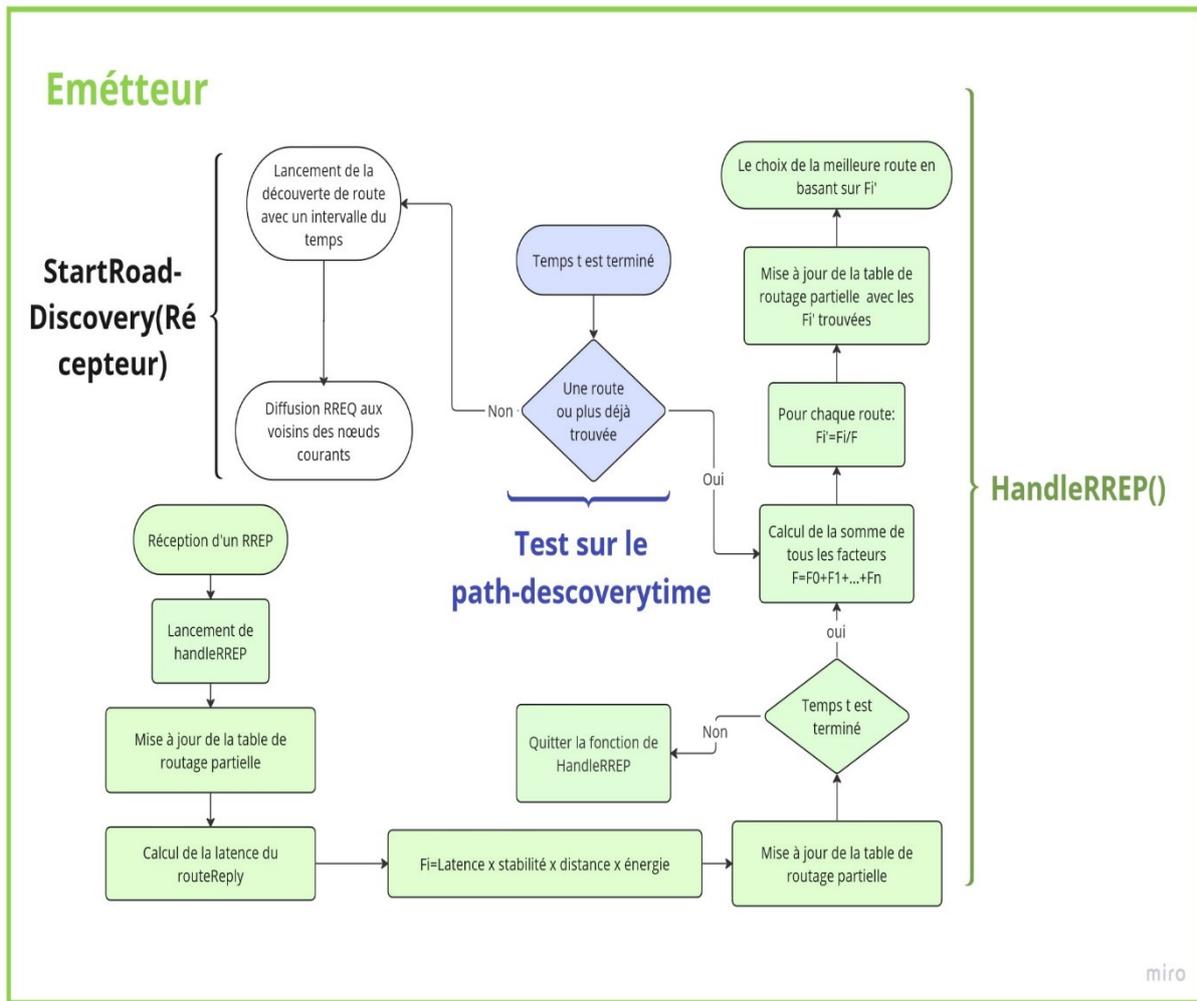


Figure 11. Organigramme d'exécution au niveau de l'émetteur.

t : temps d'attente de réponse.

- Au niveau du nœud récepteur :

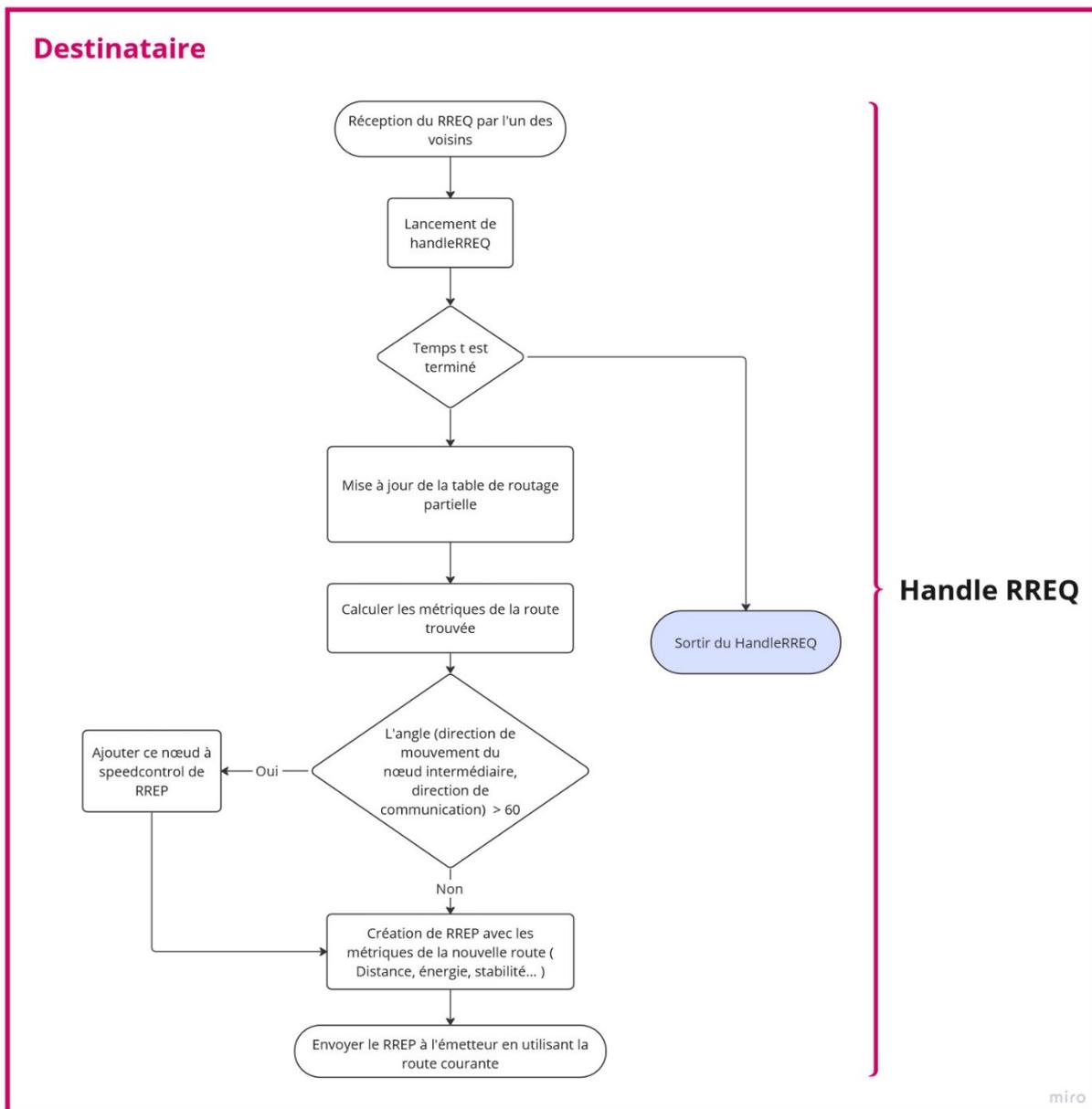


Figure 12. Organigramme d'exécution au niveau du destinataire.

- **Au niveau nœud intermédiaire :**

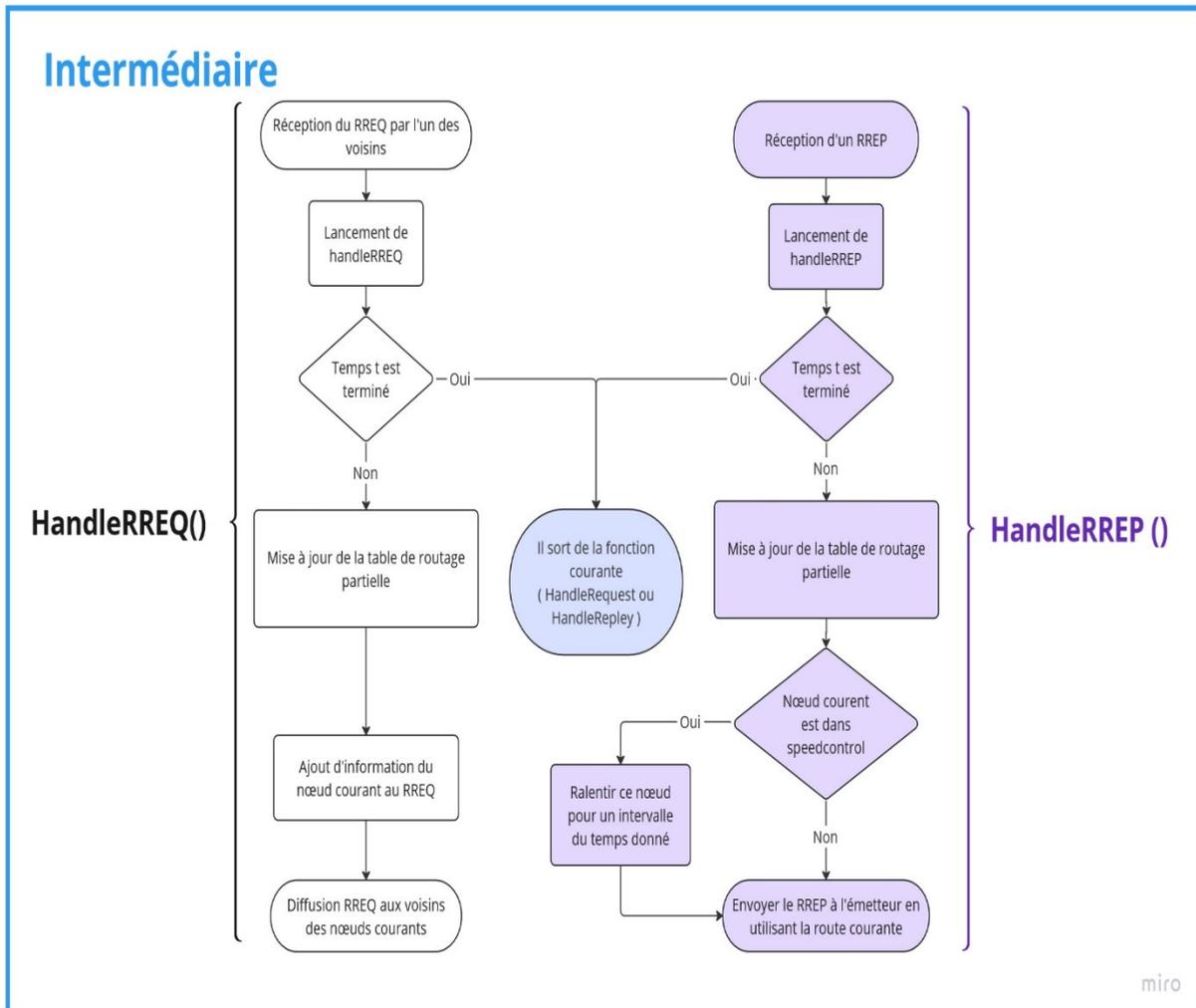


Figure 13. Organigramme d'exécution au niveau des nœuds intermédiaires.

4. Implémentation

Dans cette section nous présentons l'environnement de développement de notre application l'OMNet++ ainsi que les étapes de son installation et en fin une description général de l'application réalisée.

4.1. Environnement de développement

Nous avons utilisé le simulateur OMNeT++ pour l'implémentation de notre système, précisément la version omnetpp-6.0.1. Ce choix est basé spécialement sur la stabilité de OMNeT++ ainsi que la disponibilité des documentations requise pour le développement des objectifs visés.

Nous avons utilisé également une machine HP i5 avec 8 Go de RAM et qui procède comme système d'exploitation « Windows 10 ».

4.2. Présentation d' OMNeT++

OMNeT++ (figure 14) est une bibliothèque et un Framework de simulation écrits en C++, qui est extensible et modulaire. Il est principalement conçu dans le but de construire des simulateurs de réseau tels que les réseaux de communication filaires et sans fil, les réseaux sur puce, les réseaux de file d'attente et d'autres types similaires.

Les cadres de modèles qui sont créés en tant que projets indépendants, fournissent des fonctionnalités spécifiques au domaine, telles que la prise en charge des réseaux de capteurs, des réseaux ad hoc sans fil, des protocoles Internet, de la modélisation des performances et des réseaux photoniques. OMNeT++ est équipé d'une gamme d'outils, dont un IDE basé sur Eclipse, un environnement d'exécution graphique et de nombreux autres.

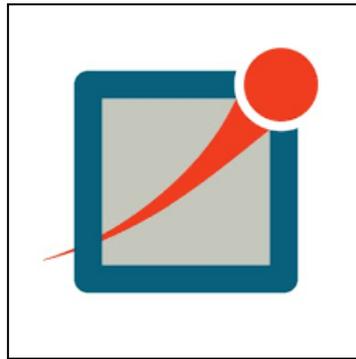


Figure 14. Logo de l'OMNeT++.

OMNeT++ comprend les composants fondamentaux suivants [43]:

- Une bibliothèque de noyau de simulation construite en C++ ;
- Le langage de description de topologie NED qui permet de modéliser des topologies de réseau ;
- Un environnement de développement intégré (IDE) basé sur la plateforme Eclipse ;
- Une interface utilisateur graphique (GUI) d'exécution de simulation interactive connue sous le nom de Qtenv ;
- Une interface de ligne de commande qui facilite l'exécution de la simulation, appelée Cmdenv ;
- Une gamme d'utilitaires, dont un outil de création de makefile ;
- Et divers autres des ressources telles que la documentation et exemples de simulations.

Au cours de sa disponibilité, OMNeT++ a été utilisé dans divers domaines, ce qui a donné lieu au développement de nombreux modèles de simulation et cadres de modèles. Ces

modèles et infrastructures couvrent un large éventail de domaines, notamment les files d'attente, la modélisation des ressources, les protocoles Internet, les réseaux sans fil, les réseaux locaux commutés, les réseaux peer-to-peer, le streaming multimédia, les réseaux mobiles ad hoc, les réseaux maillés, les réseaux de capteurs sans fil, les réseaux véhiculaires, les réseaux NOC, les réseaux optiques, les systèmes HPC, le cloud computing, les SAN, etc.

La majorité de ces frameworks sont open source, créés en tant que projets indépendants et respectent leurs propres cycles de publication.

Installation d'OMNeT++ sous windows :

La procédure d'installation d'OMNeT++ se diffère d'un système d'exploitation à l'autre. Dans notre travail, nous avons installé et utilisé l'OMNeT++ sous Windows comme suit [43]:

- **Téléchargement et préparation :** les étapes de téléchargement et préparation de l'environnement sont les suivantes :
 1. Téléchargement de code source d'OMNeT++ depuis <https://omnetpp.org>. En assurant de sélectionner l'archive spécifique à Windows, nommée **Omnetpp-6.0-windows-x86_64.zip**. Le package est autonome, en plus des fichiers OMNeT++, il inclut un compilateur C++, un environnement de construction en ligne de commande, ainsi que toutes les bibliothèques et programmes nécessaires.
 2. On copie l'archive d'OMNeT++ dans le répertoire où nous souhaitons l'installer. Nous choisissons un répertoire dont le chemin complet ne contient aucun espace. Par exemple, le placement ne doit pas être sous « Program Files ».
 3. **Extraction** de fichier zip. Le nouveau répertoire omnetpp-6.0 doit contenir des répertoires nommés doc, images, include, tools, etc., ainsi que des fichiers nommés mingwenv.cmd, configure, Makefile et d'autres.
- **Configuration :**
 1. Pour la configuration, il faut démarrer **mingwenv.cmd** dans le répertoire **omnetpp-6.0** en double-cliquant dessus dans l'Explorateur Windows. Cela ouvrira une console avec l'interpréteur de commandes **MSYS bash**, où le chemin est déjà configuré pour inclure le répertoire **omnetpp-6.0/bin**. Lors du premier démarrage de la console, il devra peut-être attendre l'extraction du répertoire tools.
 2. Tout d'abord, il faut vérifier le contenu du fichier **configure.user** pour assurer qu'il contient les paramètres nécessaires.

```
notepad configure.user
```

3. Ensuite les commandes suivantes :

```
$ ./configure
```

```
$ make
```

Puis

Le processus de construction créera à la fois des binaires de débogage et des binaires de version finale (release).

4. Pour tester tous les exemples et vérifier qu'ils s'exécutent correctement. Il est recommandé d'utiliser les commandes suivantes.

```
$ cd samples/aloha
```

```
$ ./aloha
```

Puis :

5. Pour lancer l'IDE, il suffit de taper :

```
$ omnetpp
```

5.3. Description générale de l'application réalisée

Dans ce travail, notre objectif principal est de proposer un algorithme de routage hybride (réactif et proactif à la fois), intelligent et auto-organisé, particulièrement adapté aux réseaux hétérogènes et plus précisément pour les villes intelligentes.

Notre approche est basée sur l'intelligence distribuée (via des décisions locales) et la coordination entre les nœuds du réseau, ce qui permet une optimisation continue du processus de routage et réduit la congestion réseaux en minimisant le nombre de messages de contrôle et la consommation d'énergie.

Nous présentons dans ce qui suit les étapes clés de notre algorithme ainsi que leurs méthodes d'implémentation :

1. Au début, nous spécifions les entrées de notre algorithme (les deux nœud émetteur et récepteur). Puis, nous lançons l'opération de découverte de route en spécifiant comme paramètre le temps d'attente de réponses.
 - a. Le nœud émetteur prépare un message (RREQ) qui contient ses informations comme (Position, vitesse, énergie, séquence ...)
 - b. Il le diffuse aux voisins comme suit (figure 15) :

```
SenderRREQ(energy, position, speed){
RREQ rreq ;
rreq.addEnergy(energy) ;
rreq.addPosition(position) ;
rreq.addSpeed(speed);
rreq.setHopcount(0) ;
rreqid++ ;
rreq.setRREQid(rreq.id) ;
rreq.addAddress(current_IP()) ;
Broadcastneighbors();
}
```

Figure 15. Diffusion du RREQ.

- c. Si l'émetteur ne reçoit pas son RREP dans un intervalle du temps limité, une nouvelle session sera lancée (réexécution de cette étape).
2. Lorsqu'un nœud voisin reçoit un paquet RREQ, il lance l'opération handleRREQ comme suit :
 - a. Mis à jour de la table de routage avec la fonction updateRoute : ajouter la route vers nœud précédent (father) et la route vers le nœud émetteur dans sa table de routage partielle.
 - b. Si le nœud courant est la destination, donc il calcule les métriques de la route trouvée (stabilité, énergie, distance...).
 - 1) Si le l'angle entre l'axe de direction de déplacement de ce nœud et l'axe de la direction de communication est supérieure à 60°, il insère les identifiants des nœuds concernés au champs « speedcontrol » de RREP et aussi les informations de cette route.
 - 2) Il envoie un RREP avec le score de la nouvelle route trouvée à l'émetteur en utilisant cette route.
 - c. Sinon, il rajoute ses information (Position, vitesse, énergie, ...) au RREQ reçu et le renvoie vers ses voisins (figure 16).

```

HandleRREQ(rreq){
  ReceiveRREQ(rreq);
  if(currentIPaddress==rreq->getAddr(::))return ;
  updateRoute(PreviousHop);
  updateRoute(PreviousHop,rreq.getoriginatorAddr());
  if(rreq.getDestAddr()==current) {
    RREP rrep ;
    rrep.setHopcount(0) ;
    rrep.setDistance(calculate(rreq))
    for n in rreq.nodesAll {
      if (angle(n, destination)>60°){
        rrep.addSpeedControl(n.Addr);
      }
    }
    rrep.calculStability(rreq);
    rrep.calculEnergy(rreq);
    rrep.send(rreq) using same route that the rreq has taken ;
  }
  else{
    rreq.addEnergy(current.energy) ;
    rreq.addSpeed(current.speed) ;
    rreq.addAddress(currentIPaddress) ;
    rreq.setHopcount(rreq.getHopcount()+1) ;
    broadcast rreq to all neighbours ;
  }
}

```

Figure 16. Mise à jour du RREQ.

3. Lorsqu'un nœud reçoit un RREP, Il va déclencher la fonction handleRREP comme suit :
 - a. Mis à jour de la table de routage du nœud courant :
 - 1) S'il y a une route du nœud émetteur de ce RREP, il la met à jour avec les nouvelles métriques.
 - 2) Sinon il l'ajoute dans la table.
 - b. Mis à jour de la route vers le destinataire :
 - 1) S'il existe déjà une route, il va la mettre à jour avec les nouvelles métriques.
 - 2) Sinon il va créer une nouvelle route.
 - c. Si le nœud courant se trouve dans « speedcontrol » de RREP, alors il ralentit sa vitesse.
 - d. Si le nœud courant est le nœud émetteur :
 - 1) Il calcule la latence de cette route,
 - 2) Il calcule son facteur de décision selon l'équation 1.
 - 3) Il met à jour la table de routage du nœud courant avec la nouvelle route en ajoutant le facteur calculé.
 - 4) Si le temps de roadDiscovery est échoué :

1. Il calcule le facteur F de tous les scores des routes trouvées selon l'équation 2.
2. Puis, il divise le score de chaque route (calculé par le récepteur, équation 1) sur ce facteur global F (équation 2). Nous rajoutons ce nouveau facteur à la route.
3. Il établit une table temporaire qui contient toute la route trouver avec ce nouveau facteur pour qu'il l'utilise lors le choix de la meilleure route.
4. Il termine par décider et il choisit la route avec le meilleur facteur F et lance la communication en utilisant la route choisie (figure 17).

```

Completeroutediscovery(target){
  For all routes in routing table {
    Add the sum of all of their factors ;
    And divide the factor of each route by the sum of the factors if all routes;
  }
}

```

Figure 17. Calcul du facteur F.

- 5) Sinon il lance une nouvelle découverte de route.
- e. Sinon un RREP sera envoyé au nœud suivant (figure 18).

```

HandleRREP(rrep){
  updateRoute(previoushop) ;
  updateRoute(rrep.destaddr, previoushop) ;
  if (rrep.orginatoraddr == currentIPaddress){
    latency=calculatelatency(rreq) ;
    factor = latency * distance* stability * energy ;
    updateRoute(rrep.getdestaddress() ,previoushop,factor);
    if(path_discovery_time_is_finished())
      completeroutediscovery(rrep.getdestaddr) ;
  }
  else{
    if(currentipaddress ∈ rrep.speedcontrole())
      Slowdown(current_node) ;
  }
}

```

Figure 18. Choix de la meilleure route.

4.4. Paramètres de simulation

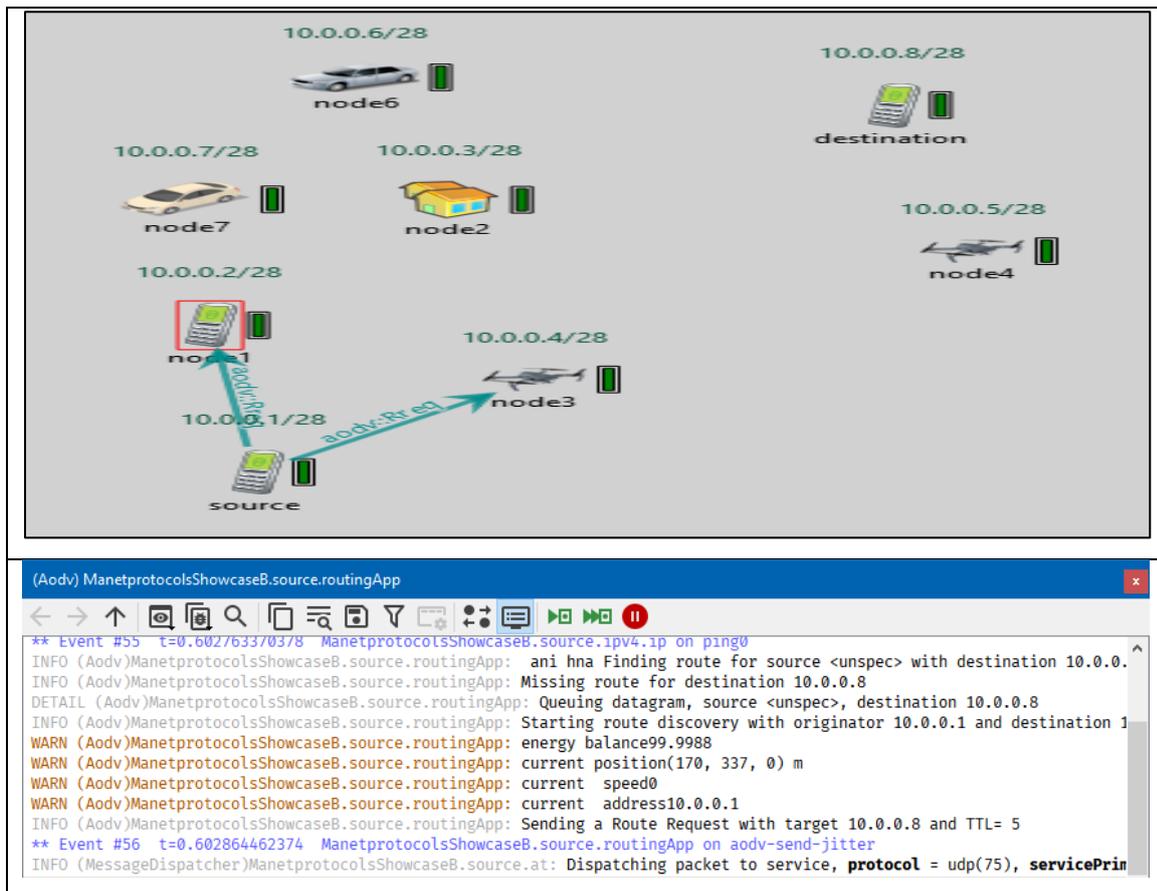
Les paramètres de simulation sont représentés dans le tableau 1.

PARAMETER	VALEUR
NOMBRE DES NOEUDS	variable (0→50)
TAILLE DU RÉSEAU	600 m × 600 m
VITESSE	25 m/s
MODÈLE DE LA MOBILITÉ	Linear Mobility
TEMPS DE SIMULATION	100 s
COUCHE PHYSIQUE	AckingWirelessInterface
INTERVAL ENTRE PACKET	2 s
TOPOLOGIE	uniform

Tableau 1 : Paramètre de simulation.

5. Scenario d'exécution

1. Requêtes pour la découverte de la route (RREQ) : Tout d'abord, le nœud source crée un RREQ avec ses informations (position, vitesse, énergie...), ensuite il l'envoie vers ses voisins (figure 19).



2. Lorsque les voisins reçoivent le paquet RREQ, ils déclenchent l'opération handleRREQ comme suit (figures 20 et 21):

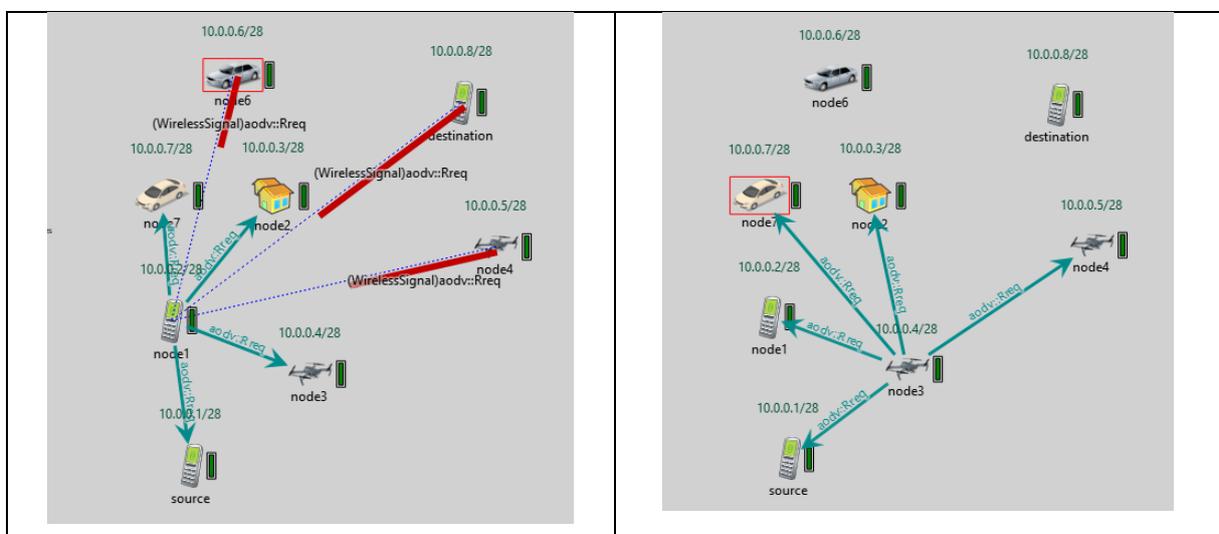


Figure 20. Étape 2 « le handleRREQ au niveau du nœud 1 et nœud 3 ».

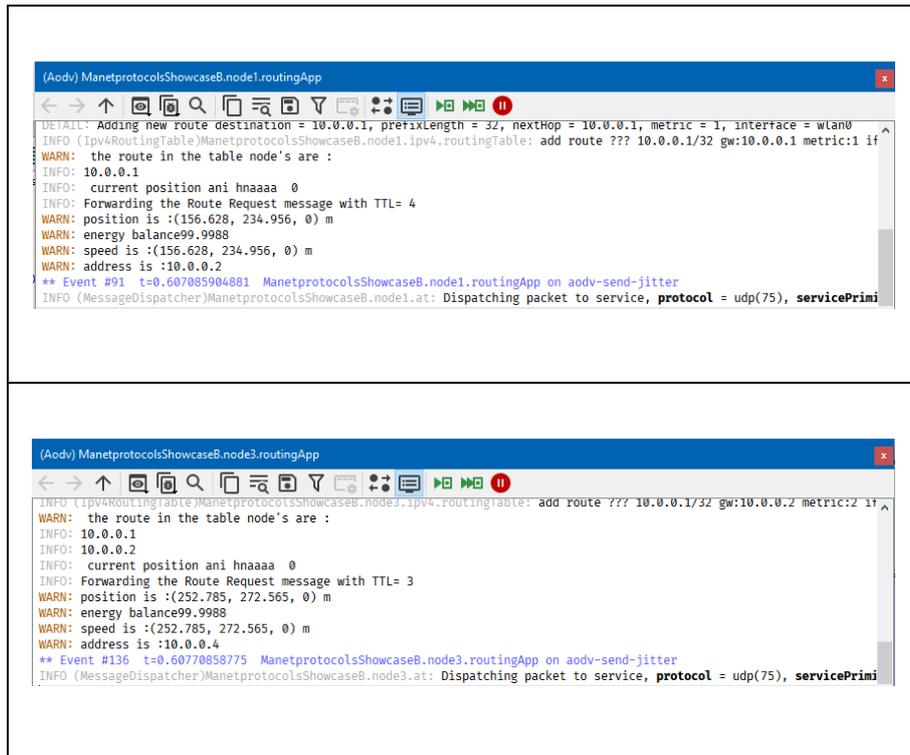


Figure 21. HandleRREQ dans les nœuds 1 et 3.

La même chose pour tous les nœuds intermédiaires comme suite (figures 22, 23, 24 et 25) :

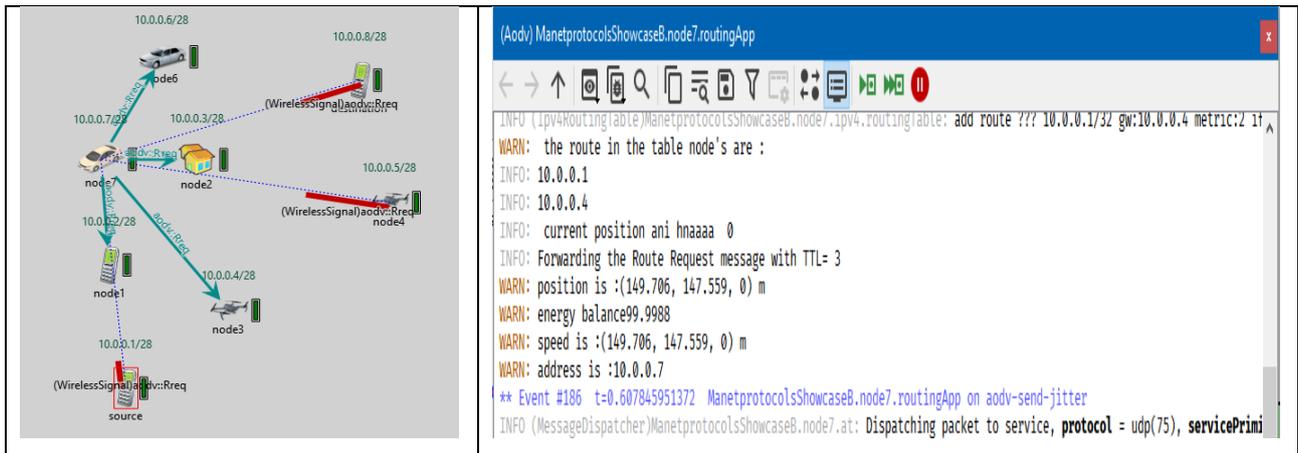


Figure 22. Étape 2 au niveau du nœud 7.

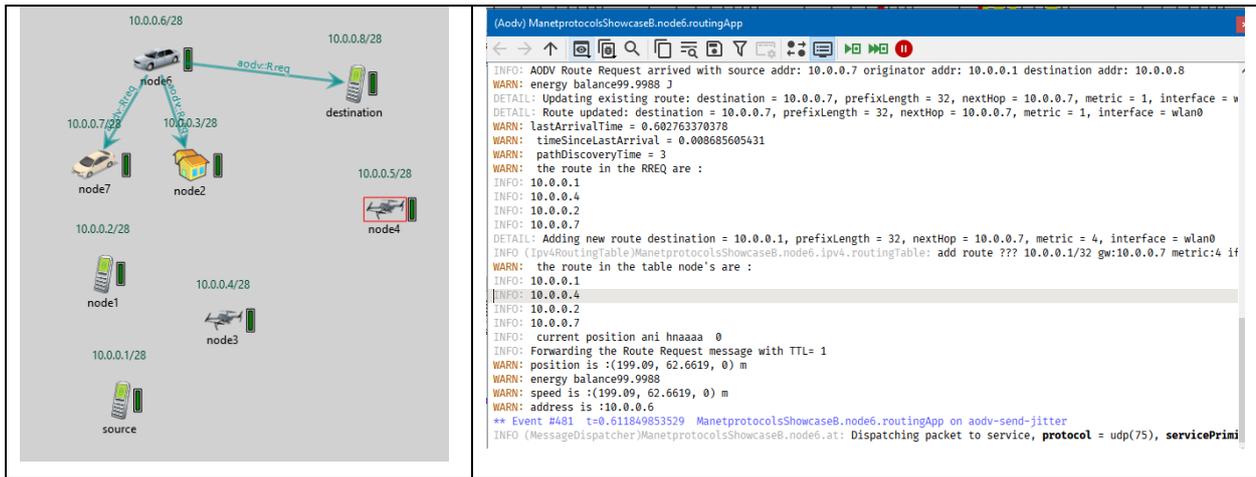


Figure 23. Étape 2 au niveau du nœud 6.

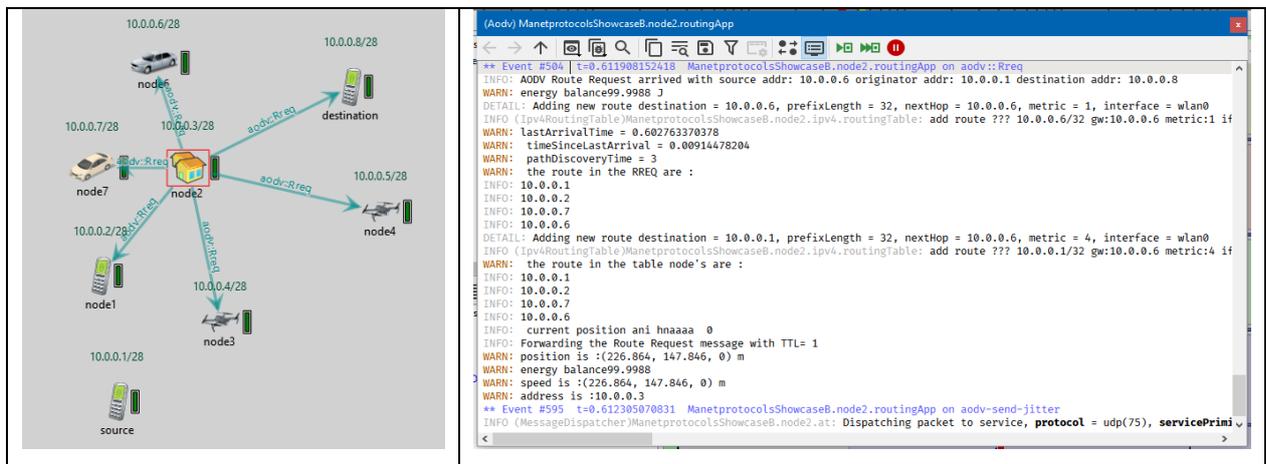


Figure 24. Étape 2 au niveau du nœud 2.

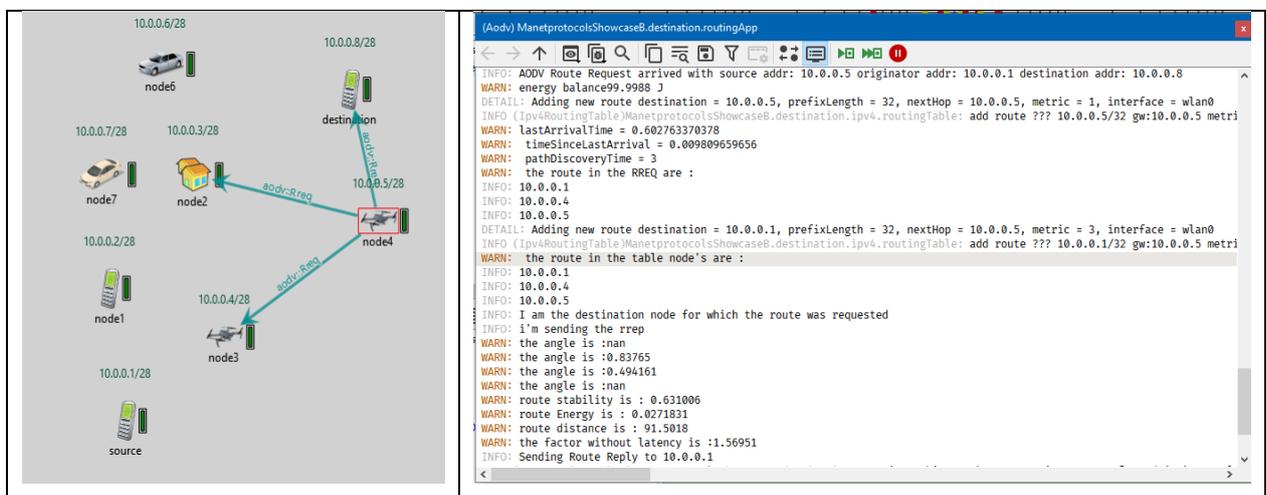


Figure 25. Étape 2 au niveau du nœud 4.

3. Réponses de la route (RREP) :

Lorsque le nœud courant est la destination du paquet, il calcule les métriques de la route trouvée (stabilité, énergie, distance...), puis il renvoie une réponse avec les métriques de la nouvelle route à l'émetteur au travers la dernière route de réception (figures 26 et 27).

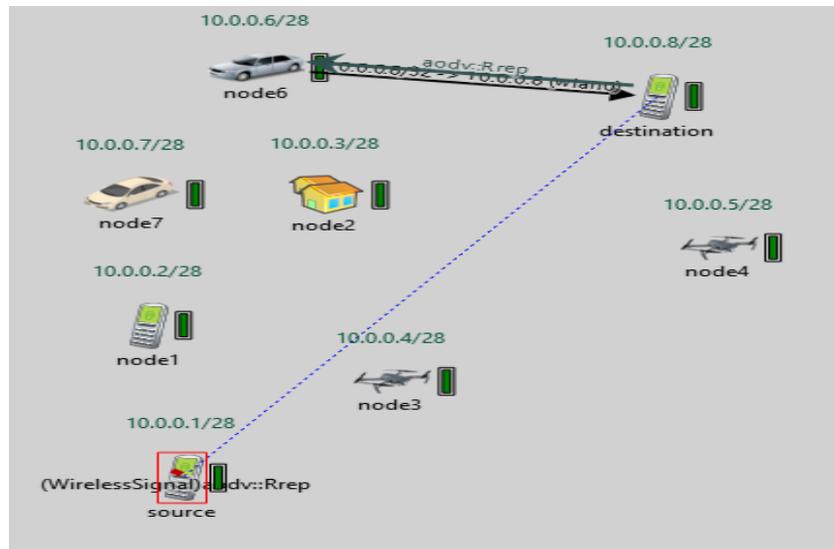


Figure 26. Étape 3 « renvoi de la réponse avec le score calculé du route courante » 1.

```
(Aodv) ManetprotocolsShowcaseB.node6.routingApp
INFO: 10.0.0.1 255.255.255.255 10.0.0.7 wlan0 (10.0.0.6) 3
INFO: 10.0.0.1 255.255.255.255 10.0.0.7 wlan0 (10.0.0.6) 3
INFO: 10.0.0.1 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 3
INFO: 10.0.0.1 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 3
INFO: 10.0.0.1 255.255.255.255 10.0.0.7 wlan0 (10.0.0.6) 4
INFO: 10.0.0.1 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 4
INFO: 10.0.0.1 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 5
INFO: 10.0.0.1 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 5
INFO: 10.0.0.3 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 1
INFO: 10.0.0.7 255.255.255.255 10.0.0.7 wlan0 (10.0.0.6) 1
INFO: 10.0.0.8 255.255.255.255 10.0.0.8 wlan0 (10.0.0.6) 1
INFO:
WARN: the route in the RREP are :
INFO: 10.0.0.8
WARN: the route in the table node's are :
DETAIL: Adding new route destination = 10.0.0.8, prefixLength = 32, nextHop = 10.0.0.8, metric = 1, interface = wlan0
INFO (Ipv4RoutingTable)ManetprotocolsShowcaseB.node6.ipv4.routingTable: add route ??? 10.0.0.8/32 gw:10.0.0.8 metric:1 if
WARN: after update/create route to rrep->getdestAddr()
INFO: -- Routing table --
INFO: Destination Netmask Gateway Iface Metric
INFO: 10.0.0.1 255.255.255.255 10.0.0.7 wlan0 (10.0.0.6) 3
INFO: 10.0.0.1 255.255.255.255 10.0.0.7 wlan0 (10.0.0.6) 3
INFO: 10.0.0.1 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 3
INFO: 10.0.0.1 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 3
INFO: 10.0.0.1 255.255.255.255 10.0.0.7 wlan0 (10.0.0.6) 4
INFO: 10.0.0.1 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 4
INFO: 10.0.0.1 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 5
INFO: 10.0.0.1 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 5
INFO: 10.0.0.3 255.255.255.255 10.0.0.3 wlan0 (10.0.0.6) 1
INFO: 10.0.0.7 255.255.255.255 10.0.0.7 wlan0 (10.0.0.6) 1
INFO: 10.0.0.8 255.255.255.255 10.0.0.8 wlan0 (10.0.0.6) 1
INFO: 10.0.0.8 255.255.255.255 10.0.0.8 wlan0 (10.0.0.6) 1
INFO:
WARN: i executed setmax to 5
INFO: Forwarding the Route Reply to the node 10.0.0.1 which originated the Route Request
```

Figure 27. Étape 3 « renvoi de la réponse avec le score calculé du route courante » 2.

- Mis à jour de la table de routage (figure 28, 29 et 30):

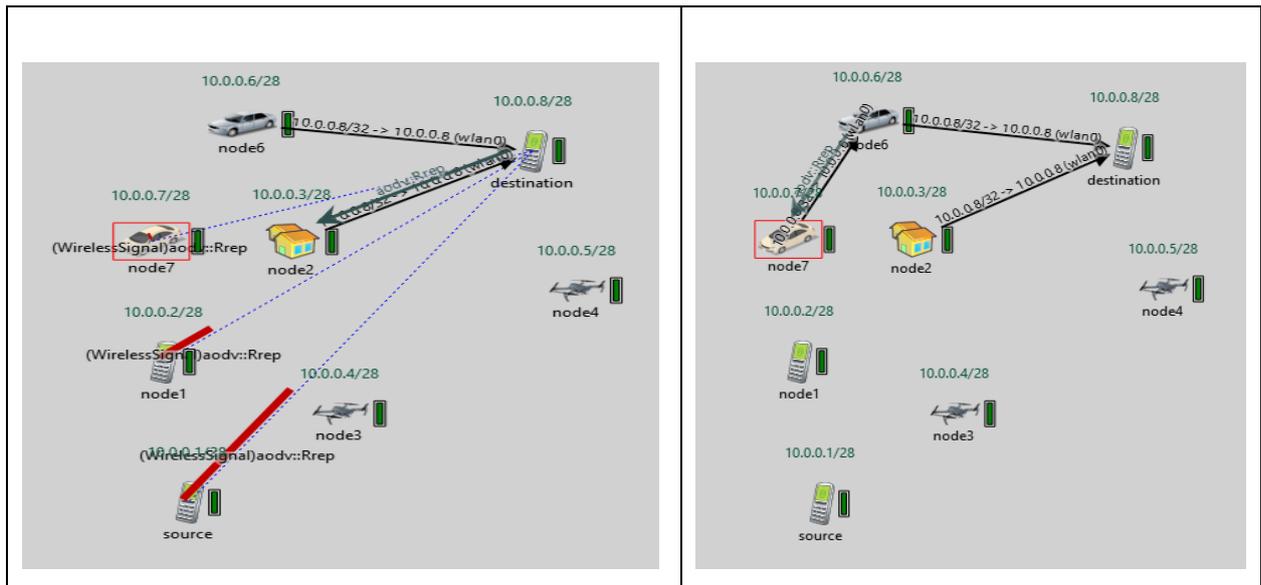


Figure 28. Mise à jour de la table du routage.

- Choix de la meilleure route (figures 29 et 30) :

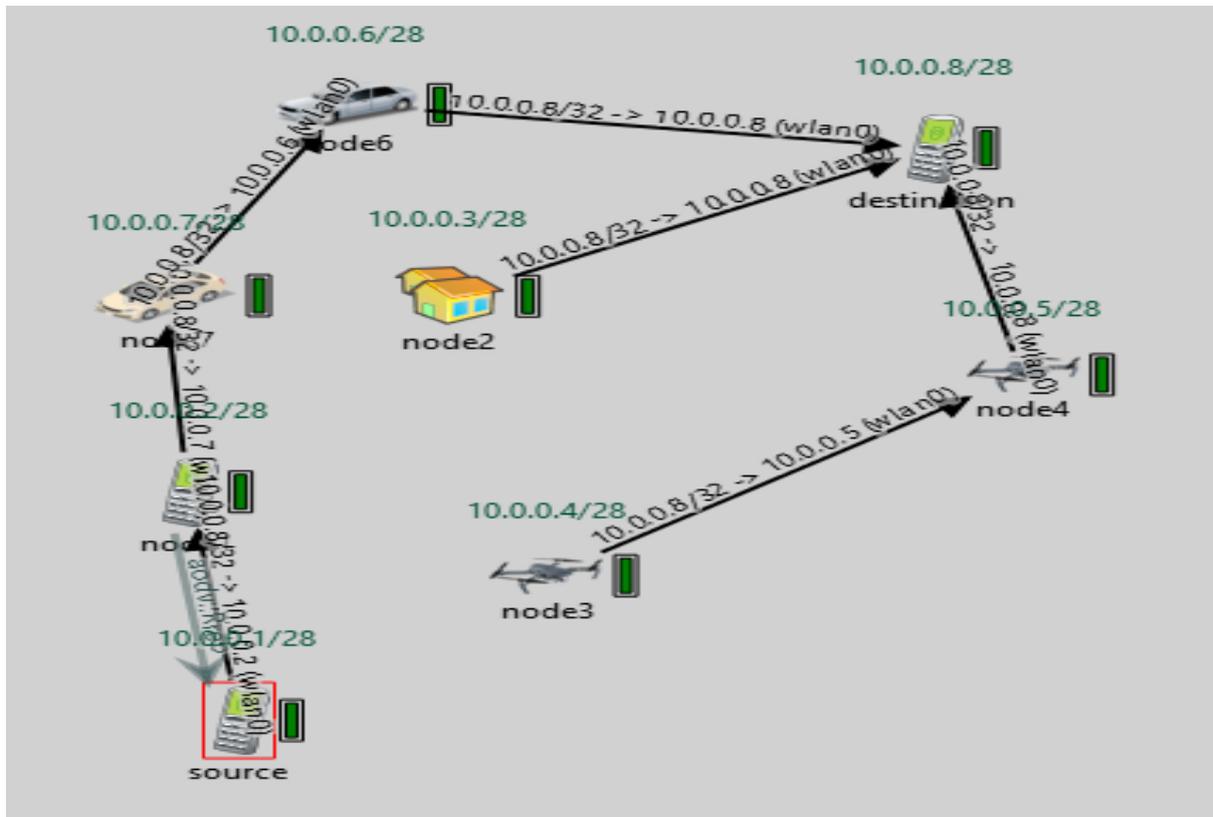


Figure 29. Étape finale « choix du meilleure route » 1.

```

(Aodv) ManetprotocolsShowcaseB.source.routingApp
INFO: -- Routing table --
INFO: Destination      Netmask      Gateway      Iface      Metric
INFO: 10.0.0.2           255.255.255.255  10.0.0.2     wlan0 (10.0.0.1)  1
INFO:
WARN:  the route in the RREP are :
INFO: 10.0.0.8
INFO: 10.0.0.6
INFO: 10.0.0.7
INFO: 10.0.0.2
DETAIL: Adding new route destination = 10.0.0.8, prefixLength = 32, nextHop = 10.0.0.2, metric = 4, interface = wlan0
INFO (Ipv4RoutingTable)ManetprotocolsShowcaseB.source.ipv4.routingTable: add route ??? 10.0.0.8/32 gw:10.0.0.2 metric:4 if
WARN:  after update/create route to rrep→getdestAddr()
INFO: -- Routing table --
INFO: Destination      Netmask      Gateway      Iface      Metric
INFO: 10.0.0.2           255.255.255.255  10.0.0.2     wlan0 (10.0.0.1)  1
INFO: 10.0.0.8           255.255.255.255  10.0.0.2     wlan0 (10.0.0.1)  4
INFO:
INFO: The Route Reply has arrived for our Route Request to node 10.0.0.8
DETAIL: Updating existing route: destination = 10.0.0.8, prefixLength = 32, nextHop = 10.0.0.2, metric = 4, interface = wl.
DETAIL: Route updated: destination = 10.0.0.8, prefixLength = 32, nextHop = 10.0.0.2, metric = 4, interface = wlan0
WARN:  the factor for this route 0.00500042
WARN:  ani hna fel completeroutediscovery 1 lifiha scheduleafter, et lw9t li b9a: 0.018210992762
** Event #2905 t=0.62097436314 ManetprotocolsShowcaseB.source.routingApp on completeDiscovery
WARN:  bizzard
WARN:  Completing route discovery, originator 10.0.0.1, target 10.0.0.8
WARN (Ipv4RoutingTable)ManetprotocolsShowcaseB.source.ipv4.routingTable: find best matching route with aodv
WARN:  route :10.0.0.2
WARN:  route :10.0.0.7
WARN:  route :10.0.0.6
WARN:  route :10.0.0.8
WARN:  reverse route :10.0.0.6
WARN:  reverse route :10.0.0.7
WARN:  reverse route :10.0.0.2
WARN:  reverse route :10.0.0.1
:
DETAIL: Sending queued datagram: source <unspec>, destination 10.0.0.8
DETAIL (Ipv4)ManetprotocolsShowcaseB.source.ipv4.ip: Sending datagram 'ping0' with destination = 10.0.0.8

```

Figure 30. Étape finale « choix du meilleure route » 2.

6. Discussion

Notre algorithme est caractérisé par un certain niveau d'intelligence et la capacité de s'auto-organiser afin de permettre aux nœuds du réseau de prendre des décisions de routage sur la base d'informations distribuées disponibles localement et de mécanismes de prise de décision locale. Notre algorithme est capable d'effectuer une analyse simple et en temps réel de l'état du réseau, puis déterminer les routes. Cela permettra principalement d'améliorer la tolérance aux pannes, équilibrer la surcharge réseau et une meilleure utilisation des ressources.

Cet algorithme prend en charge la possibilité d'avoir plusieurs routes disponibles et sélectionne la meilleure entre elles. Chaque route est évaluée en fonction de diverses métriques telles que la latence, la distance, l'énergie et la stabilité.

La stabilité d'une route est déterminée en prenant en compte la vitesse du nœud actuel, l'angle de direction de chaque nœud et la direction de communication. L'intelligence de cet algorithme réside dans la prise de plusieurs décisions : choix de la meilleure route, ralentissement de vitesse, ...etc. Ainsi, cet algorithme de routage présente des caractéristiques spécifiques qui le rendent adapté aux environnements des villes intelligentes tels que la mobilité des nœuds, les ressources limitées, etc.

7. Conclusion

Dans ce chapitre, nous avons présenté les principales étapes de réalisation et d'implémentation de notre travail tout en commençant par la modélisation du système, une description de l'environnement de développements, un scénario d'exécution et en fin les principales interfaces de l'applications.

Conclusion générale et perspectives

Pendant les dernières années, le développement des villes intelligentes a été facilité par l'apparition de plusieurs types de technologies et des appareils intelligents. Ces villes utilisent les TIC pour améliorer les processus urbains et la qualité de vie des habitants. Des réseaux de communication efficaces et fiables sont essentiels pour que les villes intelligentes puissent échanger facilement des données entre les appareils et les systèmes. Le routage devient de plus en plus difficile et critique dans un tel environnement, où de nombreux appareils et composants d'infrastructures hétérogènes sont interconnectés via des réseaux mobiles qui sont aussi hétérogènes.

Un protocole de routage efficace est essentiel pour fournir des données fiables dans les villes intelligentes. Ces protocoles doivent prendre en compte plusieurs facteurs, notamment des ressources limitées, la mobilité, l'adaptabilité et la gestion de technologies de communication hétérogènes. Notre objectif est de proposer un nouvel algorithme de routage basé sur le principe de l'auto-organisation et permet une meilleure gestion de ressources disponibles.

Pour atteindre notre objectif, nous avons présenté dans le premier chapitre un état de l'art dont nous avons mis l'accent sur les concepts de base en relation avec notre problématique tels que les villes intelligentes, l'IoT et l'auto organisation. Dans le deuxième chapitre nous avons introduit le problème de routage de données dans les environnements intelligents tout en spécifiant les types et les métriques de routage utilisés dans ce domaine. Ensuite nous avons présenté plusieurs algorithmes de routage de données en mettant l'accent sur leurs idées et objectifs principales.

Puis nous avons dégagé une étude critique de ces algorithmes qui a mené à la proposition de notre algorithme et le choix de critères à prendre en considération ainsi que l'objectif visé par un tel algorithme. Notre algorithme proposé est basé sur le principe de l'auto-organisation et la qualité de service comportant plusieurs critères à savoir la latence, la distance, l'énergie et la stabilité de routes. Pour mieux interpréter les résultats, nous avons utilisé le simulateur de réseau OMNet++.

Dans le dernier chapitre, nous avons montré les principales étapes d'implémentation et réalisation de notre travail en commençant par le modèle du système, les diagrammes d'exécution pour les différents nœuds participant au processus de routage à savoir l'émetteur, le récepteur et les nœuds intermédiaires. Aussi, nous avons donné une description générale du

simulateur utilisé, l'environnement de développement de notre projet, une description générale de l'application réalisé ainsi que les paramètres de simulation.

Comme dernière étape, nous avons présenté les différentes interfaces de notre simulation suivie d'une interprétation des différents résultats.

Perspectives

Les perspectives de recherches de ce travail sont nombreuses. Nous les présentons en suivant :

- Utiliser plus de critères d'évaluation de la qualité de service tels que principalement le degré de sécurité des réseaux disponibles, le taux d'erreur de transmission ; le taux de perte de paquets.
- Affecter des priorités aux nœuds participant au routage de données afin de bien gérer le ralentissement de la vitesse de ces derniers.
- Etudier l'impact du choix de la valeur des temps d'attente sur les performances de notre solution.
- Pour améliorer l'auto-organisation, il est important de développer des mécanismes de gestion de la topologie qui peuvent rapidement détecter et réagir aux changements topologiques. Pour cela, il serait intéressant d'intégrer les mécanismes de prédiction de la mobilité, afin d'anticiper les mouvements des nœuds. L'intégration de tels mécanismes dans notre algorithme pourrait faciliter les décisions de routage, en minimisant ainsi les perturbations causées par la mobilité fréquente des nœuds et les changements topologiques.

Références bibliographiques

- [1]. Ducrocq, T. (2013). *Auto-organisation des réseaux sans-fil multi-sauts dans les villes intelligentes* (Doctoral dissertation, Université de Lille 1)
- [2]. El Garoui, L. (2021). *Approches de Routage basées Sur les Technologies émergentes SDN et IoT Pour les Environnements Intelligents* (Doctoral dissertation, Ecole Polytechnique, Montreal (Canada)).
- [3]. Joëlle, S. I. M. A. R. D. (2015). La ville intelligente comme vecteur pour le développement durable: le cas de la ville de Montréal, Essai présenté au Centre universitaire de formation en environnement et en développement durable en vue de l'obtention du grade de maître en environnement, université de sherbrooke. *M. Env.*
- [4]. <https://www.iso.org/obp/ui/#iso:std:iso:37122:ed-1:v1:fr>
- [5]. Hammons, R., & Myers, J. (2019). Smart cities. *IEEE Internet of Things Magazine*, 2(2), 8-9. doi: 10.1109/MIOT.2019.8892761.
- [6]. <https://constrofacilitator.com/smart-city-elements-features-technology-and-govt-approach/>
- [7]. Silva, B. N., Khan, M., & Han, K. (2018). Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable cities and society*, 38, 697-713.
- [8]. <https://spcleantech.com/smart-city-sustainable-city-development/>
- [9]. <https://www.iberdrola.com/innovation/smart-mobility>
- [10]. Kassaa, P. (2019). L'internet des Objets (IoT) et les réseaux haut et bas débit, est-ce un outil essentiel pour une ville intelligente et moins Polluée ? Thèse Professionnelle. Ecole Nationale des Ponts et Chaussées.
- [11]. UIT-T, Recommandation Y.2060 - Présentation générale de l'Internet des objets, UIT-T Std., 2012.
- [12]. MEZIOUD, M. C., & KHOLLADI, M. K. (2009). Une architecture dynamique pour la résolution des problèmes d'optimisation des opérateurs de la téléphonie mobile. Conference: International Conference on Applied Informatics (ICAI'09)
- [13]. Zhang, M. (2015). *Optimisation de la couverture de communication et de mesure dans les réseaux de capteurs* (Doctoral dissertation, Reims).
- [14]. <https://geekflare.com/fr/mobile-ad-hoc-network/>.
- [15]. Aghabagherloo, A., Delavar, M., Mohajeri, J., Salmasizadeh, M., & Preneel, B. (2022). An efficient and physically secure privacy-preserving authentication scheme for Vehicular Ad-hoc NETWORKS (VANETs). *Ieee Access*, 10, 93831-93844.

- [16]. Merabet, F. (2021). *Solutions de sécurité pour l'internet des objets dans le cadre de l'assistance à l'autonomie à domicile* (Doctoral dissertation, Université de Limoges; Université Mouloud Mammeri (Tizi-Ouzou, Algérie)).
- [17]. Romdhani, B. (2012). *Exploitation de l'hétérogénéité des réseaux de capteurs et d'actionneurs dans la conception des protocoles d'auto-organisation et de routage* (Doctoral dissertation, INSA de Lyon).
- [18]. Chanet, J. P. (2007). *Algorithme de routage coopératif à qualité de service pour des réseaux ad hoc agri-environnementaux* (Doctoral dissertation, Université Blaise Pascal-Clermont-Ferrand II).
- [19]. Prehofer, C., & Bettstetter, C. (2005). Self-organization in communication networks: principles and design paradigms. *IEEE Communications magazine*, 43(7), 78-85.
- [20]. Valois, F. (2007). *Auto-organisation de réseaux radio multi-saut* (Doctoral dissertation, Université Claude Bernard-Lyon I).
- [21]. Tripathi, J., de Oliveira, J. C., & Vasseur, J. P. (2010, March). A performance evaluation study of rpl: Routing protocol for low power and lossy networks. In *2010 44th Annual Conference on Information Sciences and Systems (CISS)* (pp. 1-6). IEEE.
- [22]. Kim, H. S., Cho, H., Kim, H., & Bahk, S. (2017). DT-RPL: Diverse bidirectional traffic delivery through RPL routing protocol in low power and lossy networks. *Computer Networks*, 126, 150-161.
- [23]. Almusaylim, Z. A., Alhumam, A., & Jhanjhi, N. Z. (2020). Proposing a secure RPL based internet of things routing protocol: A review. *Ad Hoc Networks*, 101, 102096.
- [24]. Royer, E. M., & Perkins, C. E. (1999). Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications* (Vol. 2, No. 1999, pp. 90-100).
- [25]. Pathak, C., Shrivastava, A., & Jain, A. (2016). Ad-hoc on demand distance vector routing protocol using Dijkstra's algorithm (AODV-D) for high throughput in VANET (Vehicular Ad-hoc Network). In *2016 11th International Conference on Industrial and Information Systems (ICIIS)* (pp. 355-359). IEEE.
- [26]. Gupta, V., & Seth, D. (2023). Unmanned Aerial Vehicles (UAVs): Evaluation of OLSR, DSDV, AODV, and DSR Dynamic Routing Protocols. In *2023 International Conference on Device Intelligence, Computing and Communication Technologies, (DICCT)* (pp. 63-68). IEEE.
- [27]. Bormann, C., Castellani, A. P., & Shelby, Z. (2012). Coap: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(2), 62-67.

- [28]. Masykur, F., Prasetyo, A., Widaningrum, I., Cobantoro, A. F., & Setyawan, M. B. (2020, September). Application of message queuing telemetry transport (mqtt) protocol in the internet of things to monitor mushroom cultivation. In *2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)* (pp. 135-139). IEEE.
- [29]. Zhang, D. G., Zhang, T., Dong, Y., Liu, X. H., Cui, Y. Y., & Zhao, D. X. (2018). Novel optimized link state routing protocol based on quantum genetic strategy for mobile learning. *Journal of Network and Computer Applications*, *122*, 37-49.
- [30]. Yi, J., Adnane, A., David, S., & Parrein, B. (2011). Multipath optimized link state routing for mobile ad hoc networks. *Ad hoc networks*, *9*(1), 28-47.
- [31]. Yavuz, U., Dael, F. A., & Jabbar, W. A. J. S. A. (2020). Performance evaluation of DYMO and OLSRv2 routing protocols in VANET. *International Journal of Integrated Engineering*, *12*(1), 50-58.
- [32]. Kanani, P., Patil, N., Shelke, V., Salot, K., Nanavati, A., Damodaran, N., & Desai, S. (2023). Improving QoS of DSDV protocol to deliver a successful collision avoidance message in case of an emergency in VANET. *Soft Computing*, 1-11.
- [33]. Augustin, A., Yi, J., Clausen, T., & Townsley, W. M. (2016). A study of LoRa: Long range & low power networks for the internet of things. *Sensors*, *16*(9), 1466.
- [34]. Prasanna, K. R., Mathana, J. M., Ramya, T. A., & Nirmala, R. (2023). LoRa network based high performance forest fire detection system. *Materials Today: Proceedings*, *80*, 1951-1955.
- [35]. Leonardi, L., Bello, L. L., & Patti, G. (2023). MRT-LoRa: A multi-hop real-time communication protocol for industrial IoT applications over LoRa networks. *Computer Communications*, *199*, 72-86.
- [36]. Nurwarsito, H., & Umam, M. Y. (2020). Performance Analysis of Temporally Ordered Routing Algorithm Protocol and Zone Routing Protocol on Vehicular Ad-Hoc Network in Urban Environment. In *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)* (pp. 176-181). IEEE.
- [37]. Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences* (pp. 10-pp). IEEE.
- [38]. Ghazy, A. S., Kaddoum, G., & Singh, S. (2023). Low-Latency Low-Energy Adaptive Clustering Hierarchy Protocols for Underwater Acoustic Networks. *IEEE Access*.

- [39]. Jukuntla, A., & Dondeti, V. (2022). A Comparative Analysis on Variants of LEACH based routing protocols for prolonging network lifetime: Survey. In *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (pp. 351-355). IEEE.
- [40]. Varshney, S., & Kuma, R. (2018). Variants of LEACH routing protocol in WSN: A comparative analysis. In *2018 8th International conference on cloud computing, data science & engineering (confluence)* (pp. 199-204). IEEE.
- [41]. Hamrioui, S., Hamrioui, C. A. M., Lioret, J., & Lorenz, P. (2018). Smart and self-organised routing algorithm for efficient IoT communications in smart cities. *IET Wireless Sensor Systems*, 8(6), 305-312.
- [42]. Hamrioui, S., Lorenz, P., & Lloret, J. (2018). Self-organizing technique for improving coverage in connected mobile objects networks. *Telecommunication Systems*, 67, 179-193.
- [43]. <https://omnetpp.org/documentation/>