

République Algérienne Démocratique Et Populaire

Université Saad Dahleb Blida 1

Faculté des Sciences

Département informatique



**Thème :**

---

## **Proposition d'une Approche Cross-Layer pour les Réseaux LLNs à contraintes de Délai**

---

En vue d'obtention du diplôme de Master

Domaine : MI

Filière : Informatique

Spécialité : Système informatique et réseaux

**Promoteur:**Mr.MohamedOULD-KHAOUA

**Encadreur :** Mr.NadirBOUCHAMA (CERIST)

**Rapport présenté par :**

Fetta Nadia ABD MEZIEM

Yasmine AHMED SERIR

Année universitaire : 2019/2020

# Remerciement

Nous tenons tout d'abord à remercier Dieu le tout puissant, qui nous a donné la force et la patience d'accomplir ce travail. En second lieu, nous tenons à remercier notre encadreur : **Mr. Nadir BOUCHAMA**, et notre promoteur : **Mr. Mohamed OULD-KHAWA** pour leurs disponibilité et leurs judicieux conseils qui ont contribué à alimenter notre réflexion.

Nos remerciements vont également aux membres du jury pour leur présence, pour leur lecture attentive de notre thèse ainsi que pour les remarques qu'ils nous adresseront lors de cette soutenance afin d'améliorer notre travail.

A tous les intervenants, nous présentons nos remerciements, notre respect et notre gratitude.

# Dédicaces

Nous dédions ce modeste travail:

A nos très chers parents, source de vie, d'amour et d'affection qui nous ont donné le courage et nous ont soutenu pendant toute cette période difficile,

A à nos chers frères et sœurs, Hakim, Fares, Imen et Amel pour leur soutien tout au long de notre parcours universitaire.

A mon petit-neveu Moncef et mes deux petites nièces Eline et Inès adorées.

A toute notre famille, source d'espoir et de motivation

A tous nos amis, Farah, Anes et Youcef qui étaient toujours là pour nous prêter main forte et à qui nous souhaitons plus de succès.

## Résumé

Le routage dans les réseaux LLNs peut s'avérer un énorme challenge pour les chercheurs, pour cela le groupe de travail RoLL de l'IETF a proposé un nouveau protocole de routage RPL (IPv6 routing protocol for LLNs) spécialement dédié pour les réseaux à faible consommation d'énergie et sensibles à la perte de paquets.

En revanche RPL ne peut pas satisfaire tous les critères de ces types de réseaux, d'où l'association d'une autre couche paraît cruciale.

Notre but est de proposer une nouvelle approche Cross-Layer (inter-couche) qui est la synchronisation entre la couche liaison de données (TSCH) et la couche réseau (RPL), pour minimiser la latence (le délai de bout en bout), en proposant une nouvelle métrique, cela permettra d'améliorer la performance du réseau.

**Mots clés :** réseaux LLNs, routage, approche Cross-Layer, délai de bout en bout, TSCH, RPL.

يشكل التوجيه في شبكات LLN تحديًا كبيرًا للباحثين، ولهذا السبب اقترحت مجموعة عمل IETF RoLL بروتوكول توجيه جديدًا RPL (بروتوكول توجيه IPv6 لشبكات LLN) مخصص للأجهزة ذات استهلاك منخفض للطاقة وعرضة لفقدان الحزم.

ومع ذلك، لا يمكن لـ RPL تلبية جميع معايير هذه الأنواع من الشبكات، وبالتالي يبدو ارتباط طبقة مع طبقة أخرى (تبادل المعلومات بين الطبقات) أمرًا بالغ الأهمية.

هدفنا هو اقتراح نهج جديد متعدد الطبقات يتمثل في المزامنة بين طبقة ارتباط البيانات (TSCH) وطبقة الشبكة (RPL)، لتقليل زمن الوصول (تأخير من طرف إلى طرف)، من خلال اقتراح مقياس جديد والذي سيؤدي إلى تحسين أداء الشبكة.

**الكلمات الرئيسية:** LLNs، التوجيه، نهج متعدد الطبقات، التأخير من طرف إلى طرف، RPL، TSCH .

## **Abstract**

Routing in LLN networks proves to be a huge challenge for researchers, for this reason the RoLL working group of the IETF has proposed a new routing protocol RPL (IPv6 routing protocol for LLNs) specially dedicated for low power consumption and generally susceptible to packet loss networks.

On the other hand, RPL cannot meet all the criteria for this type of network, hence the association of another layer seems crucial.

Our goal is to propose a new Cross-Layer approach which is the synchronization between the data link layer (TSCH) and the network layer (RPL), to minimize latency (end-to-end delay), by proposing a new metric , this will improve the performance of the network.

**Keywords:** LLN networks, routing, Cross-Layer approach, end to end delay, TSCH, RPL.

## Table des matières

Remerciement

Dédicaces

Résumé

ملخص

Abstract

Introduction générale .....	1
Chapitre 01 : Introduction à l'Internet des Objets .....	4
1.1 Introduction .....	4
1.2 Internet des objets .....	4
1.3 Architecture de l'internet des objets.....	6
1.3.1 La couche perception .....	7
1.3.2 La couche réseau.....	8
1.3.3 La couche application .....	8
1.4 Les éléments de l'internet des objets .....	8
1.4.1 Identification .....	9
1.4.2 Détection .....	9
1.4.3 Communication.....	9
1.4.4 Calcul .....	11
1.4.5 Services .....	12
1.4.6 Sémantiques .....	12
1.5 Les domaines d'application d'IoT .....	13
1.5.1 Ville intelligente (smart cities).....	14
1.5.2 Sécurité et urgences .....	15
1.5.3 Les transports .....	15
1.5.4 Domaine médical .....	15
1.5.5 La domotique .....	16
1.6 Qualité de service .....	16
1.7 Industrial internet of things.....	16
1.7.1 Industry 4.0 .....	17
1.8 Défis de l'IoT.....	17
1.9 Conclusion .....	19

Chapitre 02 : Le standard IEEE 802.15.4e et le routage dans les réseaux IoT industriel .....	20
2.1 Introduction .....	20
2.2 Low Power and Lossy Networks(LLN) .....	20
2.3 Standard IEEE 802.15.4 .....	21
2.3.1 LR-WPAN .....	21
2.3.2 Composants du LR-WPAN.....	22
2.3.3 Limitations de IEEE 802.15.4.....	22
2.4 Standard IEEE 802.15.4e.....	23
2.4.1 Améliorations fonctionnelles générales .....	23
2.4.2 Modes de comportement de la couche mac .....	24
2.5 Time Slotted Channel Hopping TSCH.....	25
2.5.1 Fonctionnement du TSCH .....	25
2.5.2 Structure de la slotframe .....	26
2.5.3 Saut de canal .....	26
2.5.4 Formation du réseau.....	27
2.6 IPv6 Low power Wireless Personal Area Networks (6LowPan) .....	28
2.6.1 Mécanismes dans la couche d'adaptation 6LoWPAN .....	29
2.7 Protocoles de routages .....	30
2.7.1 Protocoles de routage proactifs .....	30
2.7.2 Protocoles de routage réactifs .....	30
2.7.3 Protocole de routage Proactif vs Réactif.....	31
2.7.4 Lightweight On-demand Ad hoc Distance-vector routing protocol-next generation:(LOADng).....	31
2.7.5 Le protocole de routage ipv6 pour LLNs(RPL).....	32
2.7.5.1 RPL Routes Ascendantes (construction de la topologie dodag) .....	33
2.7.5.2 RPL Routes Descendantes .....	34
2.7.5.3 Un exemple illustrant le processus de construction du DODAG.....	35
2.7.5.4 Fonction objective (OFs).....	36
2.7.5.5 TrickleTimer.....	37
2.8 Défis de routage dans les LLN .....	38
2.9 Solution des protocoles de routage .....	40
2.10 Indépendance entre la couche MAC et la couche RÉSEAU .....	41
2.10.1 Limites de l'indépendance entre la couche MAC et la couche RÉSEAU.....	41
2.11 Conclusion .....	42
Chapitre 03 : Etat de l'art sur Cross –Layer .....	43



3.1 Introduction .....	43
3.2 La communication dans les architectures Cross-Layer .....	43
3.3 Les approches du Cross Layer dans les réseaux sans fil .....	46
3.4 Type de liaison Cross Layer .....	47
3.4.1 Interactions entre la couche Physique et Transport .....	48
3.4.2 Interactions entre la couche Réseau et Physique .....	48
3.4.3 Interactions entre la couche Physique et Mac .....	48
3.4.4 Interactions entre la couche Réseau et Mac .....	48
3.4.5 Interactions entre la couche Réseau et Application .....	48
3.5 Cross-layering entre la couche Mac et la couche Réseau.....	49
3.6 Classification de Cross-Layering entre la couche Mac et la couche Réseau.....	49
3.6.1 Approches de routage utilisant des informations du protocole mac (MAC-AWARE routing).....	49
3.6.1.1 ORW (OPPORTUNISTIC ROUTING FOR WIRELESS SENSOR NETWORKS) .....	50
3.6.2 Approches mac utilisant des informations du protocole de routage (routing-aware mac).....	51
3.6.2.1 RMAC (routing enhanced mac protocol).....	51
3.6.2.2 CL-MAC (CROSS-LAYER MAC) .....	53
3.6.2.3 AREACAST.....	53
3.7 Conclusion .....	55
Chapitre 04 : QL-ETX, une nouvelle approche cross-layer pour les LLNs.....	56
4.1 Introduction .....	56
4.2 L'approche Cross-Layer TSCH-RPL .....	57
4.2.1 Délai de bout en bout .....	57
4.2.2 Modèle du système du TSCH .....	58
4.2.3 L'approche QL-ETX.....	59
4.2.3.1 Motivation .....	59
4.2.3.2 Les étapes de l'approche QL-ETX.....	60
4.3 Implémentation de l'approche QL-ETX.....	64
4.3.1 Aperçu sur le système d'exploitation Contiki OS.....	64
4.3.2 Aperçu sur le système d'exploitation Conitki-ng .....	66
4.3.3 Simulateur Cooja .....	67
4.3.4 Détails de l'implémentation de QL-ETX.....	68
4.4 Conclusion .....	72
Chapitre 05 : Evaluation des performances de QL-ETX .....	73

5.1 Introduction .....	73
5.2 Scénario de simulation.....	73
5.3 Métriques de performance .....	76
5.4 Evaluation des performances.....	77
5.5 Conclusion .....	81
Conclusion générale .....	82
Bibliographies .....	84

## Liste des figures

<b>Figure 1.1</b> Evolution de l'IoT .....	6
<b>Figure 1.2</b> Architecture de l'IoT .....	7
<b>Figure 1.3</b> Les technologies de l'IoT .....	13
<b>Figure 1.4</b> Domaines d'application d'IoT.....	14
<b>Figure 2. 1</b> Une architecture de réseau à faible puissance et à pertes (LLN) .....	21
<b>Figure 2. 2</b> La matrice de décalage temporel et de canal utilisée dans la méthode TSCH .....	27
<b>Figure 2. 3</b> Phase de balayage et de synchronisation de la méthode TSCH. EB, balise améliorée .....	28
<b>Figure 2. 4</b> Architecture 6LoWPAN .....	29
<b>Figure 2. 5</b> Propagation des DIO et formation des DAG .....	35
<b>Figure 2. 6 a)</b> Routage des données en mode non stocké RPL <b>b)</b> Routage des données en mode stocké RPL.....	36
<b>Figure 3. 1</b> La communication dans les architectures Cross Layer .....	45
<b>Figure 3. 2</b> Les approches du Cross Layer dans les réseaux sans fil .....	47
<b>Figure 3. 3</b> Période DATA de RMAC.....	52
<b>Figure 4. 1</b> Délai liés à la mise en file d'attente .....	58
<b>Figure 4. 2</b> Modèle de système de mise en file d'attente .....	59
<b>Figure 4. 3</b> Exemple d'un réseau de capteurs.....	60
<b>Figure 4. 4</b> la structure des messages DIO .....	61
<b>Figure 4. 5</b> La méthode de fonctionnement de l'approche QL-ETX dans la couche réseau du protocole RPL.....	63
<b>Figure 4. 6</b> Architecture de Contiki.....	65
<b>Figure 4. 7</b> Pile protocolaire de Contiki .....	66
<b>Figure 4. 8</b> L'interface de simulation .....	67
<b>Figure 4. 9</b> Déclaration de la métrique RPL-DAG-MC-ETX-QUEUE.....	68
<b>Figure 4. 10</b> Insertion de deux champs dans RPL-MC.....	69

<b>Figure 4. 11</b> La fonction qui récupère Link layer address du voisin .....	69
<b>Figure 4. 12</b> La récupération Link layer adresse du voisin passé en paramètre .....	70
<b>Figure 4. 13</b> L'implémentation de la fonction qui retourne le nombre de paquets.....	70
<b>Figure 4. 14</b> Récupération de nombre de paquets.....	71
<b>Figure 4. 15</b> Ajout d'un nouveau cas dans la fonction update metric container.....	72
<b>Figure 5. 1</b> Topologie aléatoire de 40 nœuds .....	74
<b>Figure 5. 2</b> Variation de $\alpha$ pour 20 nœuds .....	77
<b>Figure 5. 3</b> Variation de $\alpha$ pour 40 nœuds .....	78
<b>Figure 5. 4</b> Délai QL-ETX vs MRHOF vs QL .....	79
<b>Figure 5. 5</b> Débit QL-ETX vs MRHOF vs QL.....	80
<b>Figure 5. 6</b> Taux de livraison QL-ETX vs MRHOF vs QL.....	81

## Liste des tableaux

<b>Tableau 3.1</b> Tableau résumant les caractéristiques des approches cross-layer existantes ainsi que leurs inconvénients.....	54
<b>Tableau 4.1</b> Les champs du conteneur métrique objet .....	61
<b>Tableau 5. 1</b> Paramètres de simulation.....	75

## **Abréviations**

<b>6LOWPAN</b>	Ipv6 Over Low Power Wireless Personal Area Networks
<b>6TISCH</b>	Ipv6 Over TSCH
<b>ACK</b>	Acknowledgement
<b>AODV</b>	Ad-Hoc On Demand Distance Vector
<b>ASN</b>	Absolute Slot Number
<b>BE</b>	Beacon Enabled
<b>BLINK</b>	Radio Frequency Identification Blink
<b>CHOFFSET</b>	Channel Offset
<b>CL-MAC</b>	Cross-Layer Mac
<b>COAP</b>	Constrained Application Protocol
<b>CSMA-CA</b>	Carrier Sense Multiple Access With Collision Avoidance
<b>CTS</b>	Clear To Send
<b>DAG</b>	Directed Acyclic Graphs
<b>DAO</b>	Destination Advertisement Object
<b>DAOACK</b>	Destination Advertisement Object Acknowledgement
<b>DIO</b>	Dodag Information Object
<b>DIS</b>	Dodag Information Solicitation
<b>DODAG</b>	Destination Oriented Directed Acyclic Graph
<b>DSME</b>	Deterministic And Synchronous Multi-Channel Extension
<b>EB</b>	Enhanced Beacons
<b>EBTT</b>	Enhanced Beacons Transmission Time
<b>EDC</b>	ExpectedDuty Cycle
<b>EPC</b>	Electronic Product Codes
<b>ETX</b>	Expected Transmission Count
<b>EXI</b>	Efficient Xml Interchange
<b>FASTA</b>	Fast Association
<b>FFD</b>	Full Function Devices
<b>FSP</b>	Flow Setup Packet
<b>GSM</b>	Global System For Mobile
<b>ICMP</b>	Internet Control Message Protocol
<b>ID</b>	Identifier

<b>IE</b>	Information Element
<b>IETF</b>	Internet Engineering Task Force
<b>IIoT</b>	Industrial Internet Of Things
<b>IoT</b>	Internet If Things
<b>IP</b>	Internet Protocol
<b>IPV4</b>	Internet Protocol Version 4
<b>IPV6</b>	Internet Protocol Version 6
<b>ITU</b>	International Telecommunication Union
<b>LBR</b>	LLN Border Router
<b>LE</b>	Low Energy
<b>LLDN</b>	Low Latency Deterministic Network
<b>LLN</b>	Low Power And Lossy Networks
<b>LOADng</b>	The Lightweight On-Demand Ad Hoc Distance-Vector Routing Protocol – Next Generation
<b>LR WPAN</b>	Low Rate Wireless Personal Area Network
<b>M2M</b>	Machine To Machine
<b>MAC</b>	Medium Access Control
<b>MANET</b>	Mobile Ad-Hoc Network
<b>MAR-WSN</b>	Mac-Aware Routing In Wireless Sensor Networks
<b>MC</b>	Metric Container
<b>MIT</b>	Massachusetts Institute Of Technology
<b>MP2P</b>	Multipoint To Point
<b>MRHOF</b>	Minimum Rank With Hysteresis Objective Function
<b>MTU</b>	Maximal Transmission Unit
<b>NBE</b>	Non-Beacon Enabled
<b>NCH</b>	Number Channel
<b>NFC</b>	Near Field Communication
<b>NSC</b>	Number Scanning Channel
<b>OF</b>	Objective Function
<b>OF0</b>	Objective Function Zero
<b>OS</b>	Operating System
<b>OSI</b>	Open Systems Interconnection
<b>OWL</b>	Web Ontology Language

<b>OWR</b>	Opportunistic Routing For Wireless Sensor Networks
<b>P2P</b>	Point To Point
<b>P2MP</b>	Point To Multipoint
<b>PAN</b>	Personal Area Network
<b>PDR</b>	Packet Delivery Ratio
<b>PHY</b>	Physique
<b>QL</b>	Queue Length
<b>QoS</b>	Quality Of Service
<b>RCSF</b>	Réseau De Capteurs Sans Fil
<b>RDF</b>	Resource Description Framework
<b>RFD</b>	Reduced Function Devices
<b>RFID</b>	Radio Frequency Identification
<b>RMAC</b>	Routing Enhanced Mac Protocol
<b>RPL</b>	Routing Protocol For Low Power And Lossy Networks
<b>RREP</b>	Route Reply
<b>RREQ</b>	Route Request
<b>RTS</b>	Request To Send
<b>SBC</b>	Single Board Computers
<b>SIFS</b>	Short Interframe Space
<b>SIG</b>	Special Interest Group
<b>SPOF</b>	Single Point Of Failure
<b>TCP</b>	Transmission Control Protocol
<b>TDMA</b>	Time-Division Multiple Access
<b>TIC</b>	Technologies De L'information Et De La Communication
<b>TSCH</b>	Time Slotted Channel Hopping
<b>UCODE</b>	Ubiquitous Codes
<b>UDGM</b>	Unit Disk Graph Medium
<b>UDP</b>	User Datagram Protocol
<b>UV</b>	Ultraviolet
<b>W3C</b>	World Wide Web Consortium
<b>WDD</b>	Weighted Distance Delay
<b>WPAN</b>	Wireless Personal Area Networks
<b>WSN</b>	Wireless Sensor Network



## Introduction générale

L'Internet des objets (Internet of Things ou IoT) est un domaine émergent qui promet une connexion omniprésente à Internet, transformant des objets communs en appareils connectés. Le paradigme de l'IoT change la façon dont les gens interagissent avec les choses qui les entourent. Il ouvre la voie à la création d'infrastructures connectées de manière permanente pour prendre en charge des services innovants et promet une flexibilité et une efficacité accrues. De tels avantages sont intéressants non seulement pour les applications grand public, mais également pour le domaine industriel. Au cours des dernières années, nous avons été témoins du paradigme de l'IoT faire son chemin dans le marché de l'industrie avec des solutions conçues [1].

L'un des éléments constitutifs de l'IoT est le réseau à faible puissance et avec perte (Low Power and Lossy Networks ou LLNs), un ensemble de périphériques embarqués interconnectés, tels que des nœuds de capteurs, généralement caractérisés par des contraintes sur les ressources des nœuds et les technologies de communication sous-jacentes. Les contraintes de nœud peuvent inclure des restrictions de puissance, de traitement et de stockage, tandis que le système de communication est soumis à une perte de paquets élevée, des limitations de taille de trame, de faibles débits de données, de courtes plages de communication et des topologies de réseau à changement dynamique. De telles limitations rendent difficile le développement de solutions de routage efficaces pour les LLNs, une tâche rendue encore plus ardue par les déploiements potentiels à grande échelle de ces réseaux, qui devraient comprendre des milliers de nœuds ou plus [2].

Le routage dans les LLNs est l'un des principaux défis de l'émergence de l'IoT. Les contraintes de ces derniers ont un impact significatif sur la conception du protocole, les protocoles de routages existants traditionnels dédiés aux réseaux ad hoc (AODV, OLSR, DSR, TBRPF, DYMO) étaient inadaptés et ne répondaient pas aux besoins des réseaux LLNs.

Pour répondre à ces exigences le groupe de travail IETF ROLL a proposé des protocoles de routages spécifiques pour les réseaux LLNs tels que le protocole de routage IPv6 pour les réseaux à faible puissance et avec perte appelé RPL et LOADng. RPL a été normalisé par l'IETF tandis que LOADng apparaît comme une solution alternative.

RPL est un protocole prometteur qui présente plusieurs avantages pour les petits appareils. L'un de ses principaux atouts repose sur sa flexibilité pour gérer les changements de topologie du réseau et les modifications d'implémentation. De plus, il est utilisé pour répondre aux exigences des LLNs telles que l'équilibrage de charge, la consommation d'énergie déséquilibrée et le trafic réseau.

La problématique est que les solutions proposées pour le routage dans les réseaux IoT sont souvent basées sur la couche réseau sans pour autant associer d'autres couches (notamment les couches basses), ce qui peut mener parfois à une mauvaise performance du réseau. L'intervention d'une seule couche dans certains cas peut s'avérer insuffisante et mener à de mauvais résultats non satisfaisants et avoir un temps de latence élevé pour les réseaux IoT industriel où le délai est plus prioritaire que l'énergie pour certaines applications. RPL en lui-même a quelques lacunes ce qui impose l'association du mode TSCH pour assurer la fiabilité des couches supérieures du réseau.

Pour permettre l'interaction entre les couches et assurer la qualité de service, plusieurs chercheurs ont proposé des approches Cross-Layer(inter-couche) dans la littérature comme ; MAC-AWARE routing, RMAC (routing enhanced mac protocol), ORW (OPPORTUNISTIC ROUTING FOR WIRELESS SENSOR NETWORKS) , etc.

dans ce travail nous avons implémenté une nouvelle approche Cross-Layer(inter-couche) entre la couche mac en adoptant le mode TSCH et la couche réseau RPL pour améliorer le contrôle et les échanges de messages entre les deux couches afin d'apporter de meilleures performances et aboutir à notre objectif.

Nous avons proposé une nouvelle fonction objective QL-ETX qui est une combinaison entre deux paramètres l'un venant de la couche réseau et l'autre de la couche liaison de données.

Pour ce faire nous avons adapté le simulateur Cooja en modifiant des fichiers et en ajoutant des fonctions qui satisferont nos besoins, pour atteindre notre but qui consiste à améliorer le délai de bout en bout.

Pour mener à bien notre travail, nous avons organisé le rapport comme suit:

**Chapitre 1:** est une introduction sur l'internet des objets, ses éléments, son architecture et ses domaines d'applications

**Chapitre 2:** discute le standard IEEE 802.15.4e et le routage dans les réseaux IOT industriel

**Chapitre 3:** est consacré à l'état de l'art sur Cross-Layer

**Chapitre 4:** est dédié à la proposition et l'implémentation de la nouvelle fonction objective QL-ETX

**Chapitre 5:** traite l'évaluation des performances de la nouvelle approche, les résultats de simulation et discussion

**Chapitre 6:** conclusion générale et perspective

# Chapitre 01 : Introduction à l'Internet des Objets

## 1.1 Introduction

L'Internet n'a jamais cessé d'influencer ou même de transformer la vie de l'être humain grâce à son évolution et le développement des besoins de ce dernier, par conséquent l'informatique assiste l'émergence d'un nouveau paradigme « l'Internet des objets » ou « internet of things IoT » [3], [4], [5], ce paradigme a attiré l'intention des chercheurs car il représente une partie de l'internet de futur qui pourra aider à l'amélioration de la qualité de vie de l'humain.

Dans ce chapitre nous présentons une introduction sur l'Internet des objets en donnant une définition de l'IoT, ses éléments, son architecture et ses domaines d'application.

## 1.2 Internet des objets

L'internet des objets (IoT) se traduit à l'heure actuelle par un nombre croissant d'objets physiques qui sont connectés à Internet, c'est-à-dire transformer ces objets de traditionnels à intelligents on les rendant capables de voir, d'entendre, de penser et d'effectuer des tâches en les faisant « parler » ensemble, partager des informations et coordonner les décisions on possédant une identité propre [3].

Techniquement l'IoT est défini comme « une infrastructure de réseau mondial dynamique avec des capacités d'auto-configuration basées sur des normes et des protocoles de communication interopérables; les choses « Things » physiques et virtuelles dans un IoT ont des identités et des attributs et sont capables d'utiliser des interfaces intelligentes et d'être intégrées en tant que réseau d'information ».

Les mots « Internet » et « Things » signifient un réseau mondial interconnecté basé sur les technologies sensorielles, de communication, de mise en réseau et de traitement de l'information, qui pourrait être la nouvelle version des technologies de l'information et des communications (TIC) [5].

Le but de l'IoT est de créer «un monde meilleur pour les êtres humains», où les objets qui nous entourent savent ce que nous aimons, ce que nous voulons et ce dont nous avons besoin et agissent en conséquence sans instructions explicites autrement dit d'interagir sans intervention humaine [6].

Au fil du temps, l'IoT devrait avoir une application professionnelle, à contribuer à la qualité de vie et à faire croître l'économie mondiale. De plus, les appareils doivent être développés pour répondre aux exigences des clients en termes de disponibilité n'importe où et n'importe quand. De nouveaux protocoles sont également requis pour compatibilité de communication entre des choses hétérogènes (êtres vivants, véhicules, téléphones, appareils électroménagers, biens, etc.) [3].

Le terme «Internet des objets» a été inventé par Kevin Ashton [7] dans une présentation en 1998. Il a mentionné: «L'Internet des objets a le potentiel de changer le monde, tout comme Internet. Peut-être encore plus ». Ensuite, le centre MIT Auto-ID a présenté sa vision de l'IoT en 2001. Plus tard, l'IoT a été officiellement introduit par l'Union internationale des télécommunications (ITU) par le rapport Internet de l'ITU en 2005 [8].

Avec le temps l'IoT a beaucoup évolué, cette évolution peut être illustrée par plusieurs phases comme le montre la figure1.1

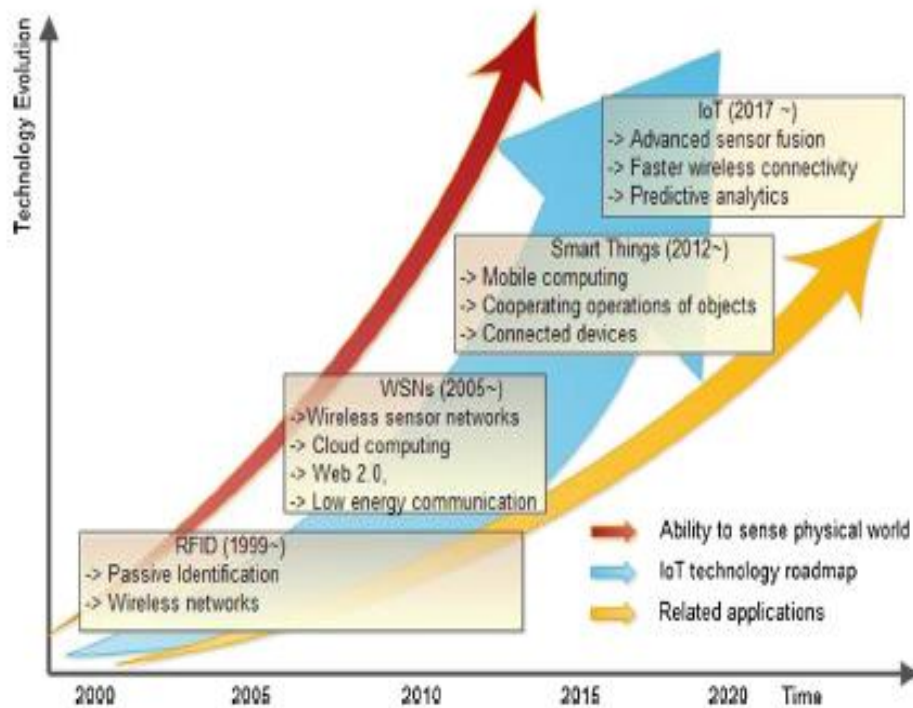


Figure 1.1 Evolution de l'IoT [5]

### 1.3 Architecture de l'internet des objets

L'architecture d'un système IoT est composée de plusieurs niveaux qui communiquent entre eux pour relier le monde tangible des objets au monde virtuel des réseaux et du cloud.

L'architecture la plus élémentaire est une architecture à trois couches, elle a été introduite aux premiers stades de la recherche dans ce domaine. Elle comprend trois couches, à savoir les couches perception, réseau et application.

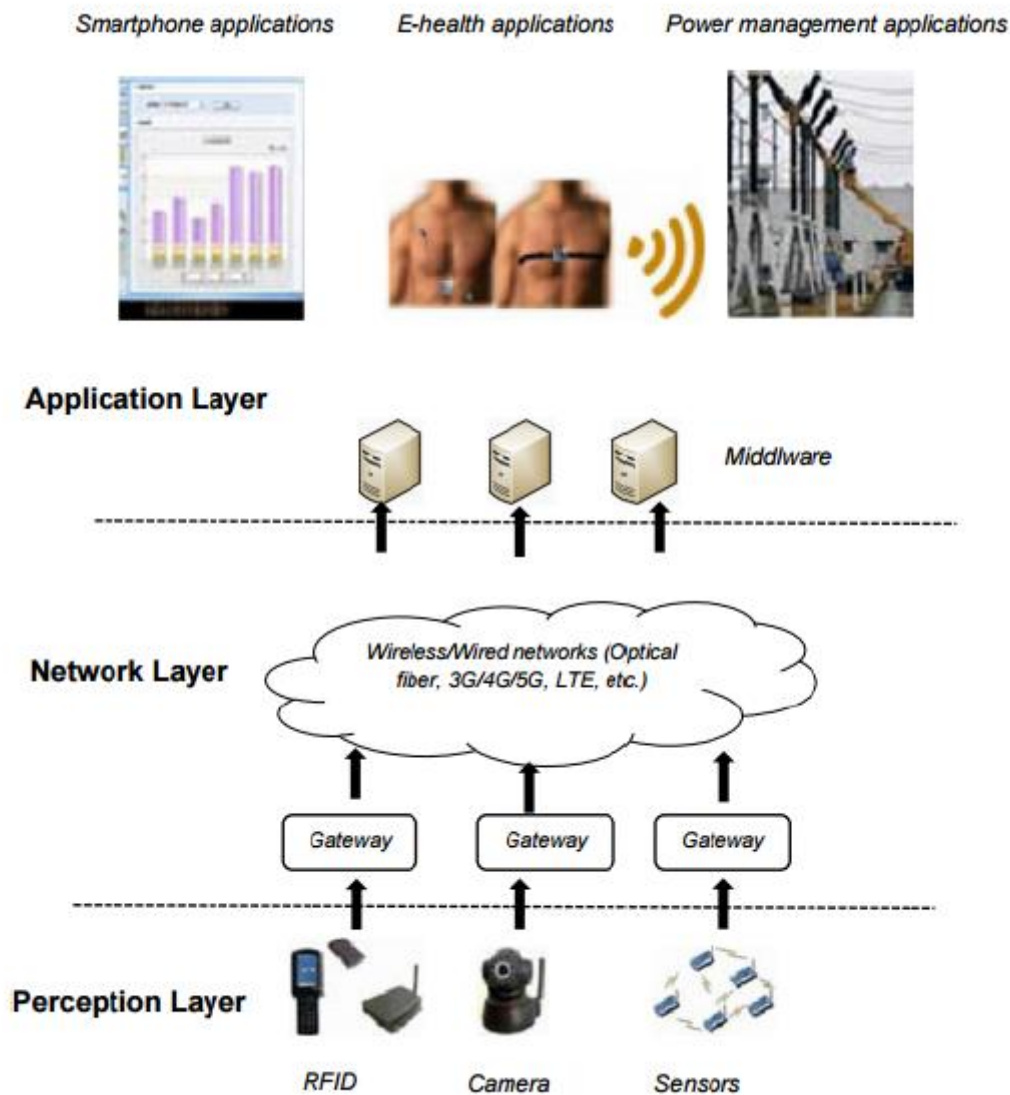


Figure 1.2 Architecture de l'IoT [4]

### 1.3.1 La couche perception

La tâche principale de la couche perception est de collecter simplement des informations sur les processus faisant partie de la couche de perception et d'extension de l'érudit. La couche d'application comprend un dispositif de reconnaissance d'étiquette de code à barres à deux gradations, une étiquette RFID, un capteur d'écart de caméra et un terminal m2m, un terminal portable, un réseau de capteurs, une passerelle de capteur. Le problème principal de cette couche doit être résolu: reconnaître et percevoir les choses: collecter et capturer l'information.

Ce que la couche de perception doit briser, c'est obtenir la compétence de perception la plus sensible et la plus compréhensible et résoudre le problème en ce qui concerne la faible puissance, la miniaturisation et l'aspect à bas prix [9].

### **1.3.2 La couche réseau**

La couche réseau IOT est établie sur la base des télécommunications mobiles et internet. Sa principale caractéristique est de transmettre l'information entre une longue distance. La couche réseau comprend divers réseaux de communication et réseaux intégrés basés sur Internet, qui est généralement considéré comme la partie la plus mature. En outre, elle comprend la partie qui traite intelligemment les informations massives. Le centre de gestion du centre d'information IOT cloudingcomputing, plateforme, expert système, etc., c'est-à-dire que la couche réseau a non seulement la compétence de fonctionnement du réseau, mais améliore également la compétence de fonctionnement de l'information. Par exemple, dans le système de paiement par téléphone portable, dès que la couche réseau a fini de confirmer les informations RFID intégrées collectées et téléchargées sur Internet via un équipement simple, le paiement sera effectué via le réseau bancaire. La couche réseau est l'infrastructure permettant à l'IOT de fournir un service commun [9].

### **1.3.3 La couche application**

La couche application fournit les services demandés par les clients. Par exemple, la couche d'application peut fournir des mesures de température et d'humidité de l'air au client qui demande ces données. L'importance de cette couche pour l'IoT est qu'elle a la capacité de fournir des services intelligents de haute qualité pour répondre aux besoins des clients. La couche d'application couvre de nombreux marchés verticaux tels que la maison intelligente, le bâtiment intelligent, le transport, l'automatisation industrielle et les soins de santé intelligents [10], [11], [9].

## **1.4 Les éléments de l'internet des objets**

L'internet des objets est composé de six éléments principaux nécessaires qui fournissent ces fonctionnalités et ces capacités qui relient entre le monde virtuel et le monde physique.

Ces éléments sont les suivants :



### 1.4.1 Identification

L'identification est cruciale pour l'IoT pour nommer et faire correspondre les services à leur demande. L'adressage des objets IoT est essentiel pour différencier l'ID d'objet et son adresse. L'ID d'objet fait référence à son nom et l'adresse de l'objet fait référence à son adresse dans un réseau de communication. De nombreuses méthodes d'identification sont disponibles pour l'IoT, telles que les codes de produits électroniques (EPC) et les codes omniprésents (uCode). De plus, les méthodes d'adressage des objets IoT incluent IPv6 et IPv4 [3].

### 1.4.2 Détection

Les capteurs intelligents collectent en continu des données de l'environnement au sein du réseau et transmettent les informations à un entrepôt de données, une base de données ou un cloud. Les données collectées sont analysées pour prendre des mesures spécifiques en fonction des services requis. Les capteurs IoT peuvent être des capteurs intelligents, des actionneurs ou des dispositifs de détection portables. Les ordinateurs à carte unique (SBC) intégrés avec des capteurs et des fonctionnalités TCP / IP et de sécurité intégrées sont généralement utilisés pour réaliser des produits IoT (par exemple, Arduino Yun, Raspberry PI, BeagleBone Black, etc.) [3].

### 1.4.3 Communication

L'Internet des objets (IoT) est basé sur la mise en réseau des objets. En un mot, IoT est défini comme «un développement proposé d'Internet dans lequel les objets du quotidien ont une connectivité réseau, leur permettant d'envoyer et de recevoir des données».

Nous pouvons résumer les protocoles de communication sans fil les plus importants dans les 6 normes suivantes [3]:

➤ **Radio Frequency Identification (RFID)**

La technologie RFID est une percée majeure dans le paradigme de la communication embarquée qui permet la conception de microprocesseurs pour la communication de données sans fil, elle est la première technologie utilisée pour réaliser le concept M2M (étiquette RFID et lecteur). L'étiquette RFID représente une simple puce ou étiquette attachée pour fournir l'identité de l'objet. Le lecteur RFID transmet un signal d'interrogation à l'étiquette et reçoit un signal réfléchi de l'étiquette, qui à son tour est transmis à la base de données. La base de

données se connecte à un centre de traitement pour identifier les objets sur la base des signaux réfléchis dans une plage (de 10 cm à 200 m) [12]. Les étiquettes RFID peuvent être actives, passives ou semi-passives / actives. Les tags actifs sont alimentés par batterie tandis que les tags passifs n'ont pas besoin de batterie. Les tags semi-passifs / actifs utilisent l'alimentation de la carte en cas de besoin.

➤ **Near Field Communication (NFC)**

Le protocole NFC fonctionne dans la bande haute fréquence à 13,56 MHz et prend en charge le débit de données jusqu'à 424 kbps. La plage applicable est jusqu'à 10 cm [13]. NFC implique un initiateur et une cible, l'initiateur génère activement un champ RF qui peut alimenter une cible passive (une puce non alimentée appelée «tag»). Cela permet aux cibles NFC de prendre des facteurs de forme très simples tels que des étiquettes, des autocollants, des porte-clés ou des cartes sans batterie. La communication NFC peer-to-peer est possible à condition que les deux appareils soient alimentés.

➤ **IEEE 802.15.4 (zigbee)**

La norme IEEE 802.15.4 spécifie à la fois une couche physique et un contrôle d'accès moyen pour les réseaux sans fil de faible puissance ciblant des communications fiables et évolutives. Zigbee est principalement utilisé en milieu industriel. Il présente des avantages importants dans les systèmes complexes offrant un fonctionnement à faible puissance, une sécurité élevée, une robustesse et une puissance élevées et est bien placé pour tirer parti des réseaux de capteurs et de contrôle sans fil dans les applications IoT [14].

➤ **Cellulaire (GSM/3G/4G/5G)**

Toute application IoT nécessitant un fonctionnement sur de plus longues distances peut tirer parti des capacités de communication cellulaire. Alors que le cellulaire est clairement capable d'envoyer de grandes quantités de données, en particulier pour la 5G, le coût et la consommation d'énergie seront trop élevés pour de nombreuses applications. Mais il peut être idéal pour les projets de données à faible bande passante basés sur des capteurs qui enverront de très faibles quantités de données sur Internet [15].

➤ **IEEE 802.11 (wifi)**

La connectivité WiFi est l'un des protocoles de communication IoT les plus populaires, il utilise des ondes radio pour échanger des données entre 100 m de portée. Il permet aux appareils intelligents de communiquer et d'échanger des informations sans utiliser de routeur dans certaines configurations ad hoc. La norme WiFi la plus courante utilisée dans les foyers et dans de nombreuses entreprises est la norme 802.11n, qui offre une plage de centaines de mégabits par seconde, ce qui est bien pour les transferts de fichiers mais peut être trop consommateur en énergie pour de nombreuses applications IoT [16].

➤ **IEEE 802.15.1(Bluetooth)**

Bluetooth présente une technologie de communication qui est utilisé pour échanger des données entre des appareils sur de courtes distances en utilisant une radio à courte longueur d'onde pour minimiser la consommation d'énergie. Récemment, le groupe d'intérêt spécial Bluetooth (SIG) a produit Bluetooth 4.1 qui fournit Bluetooth LowEnergy ainsi qu'une connectivité haut débit et IP pour prendre en charge l'IoT [17].

#### **1.4.4 Calcul**

Les unités de traitement (les microcontrôleurs, les microprocesseurs.....) et les applications logicielles représentent le «cerveau» et la capacité de calcul de l'IoT. Diverses plates-formes matérielles ont été développées pour exécuter des applications IoT (Arduino, UDOO, Intel Galileo, Raspberry PI, Z1, WiSense, Mulle et T-Mote Sky). De plus, de nombreuses plateformes logicielles sont utilisées pour fournir des fonctionnalités IoT. Parmi ces plates-formes, les systèmes d'exploitation qui fonctionnent pendant toute la durée d'activation d'un appareil. Il existe plusieurs systèmes d'exploitation en temps réel qui sont de bons candidats pour le développement d'applications IoT. Par exemple, le Contiki a été largement utilisé dans les scénarios IoT. Contiki possède un simulateur appelé Cooja qui permet aux chercheurs et aux développeurs de simuler et d'émuler des applications IoT et WSN (Wireless Sensor Network), TinyOS, LiteOS et Riot OS offrent également un système d'exploitation léger conçu pour les environnements IoT [3].

### 1.4.5 Services

Globalement, les services IoT peuvent être classés en quatre classes [3]: services liés à l'identité, services d'agrégation d'informations, services conscients de collaboration et services omniprésents. Les services liés à l'identité sont les services les plus fondamentaux et les plus importants utilisés dans d'autres types de services. Chaque application qui doit apporter des objets du monde réel au monde virtuel doit identifier ces objets. Les services d'agrégation d'informations collectent et résumement les mesures sensorielles brutes qui doivent être traitées et signalées à l'application IoT.

Les services axés sur la collaboration agissent en plus des services d'agrégation d'informations et utilisent les données obtenues pour prendre des décisions et réagir en conséquence. Les services omniprésents, cependant, visent à fournir des services axés sur la collaboration chaque fois qu'ils sont nécessaires à toute personne qui en a besoin, où que ce soit.

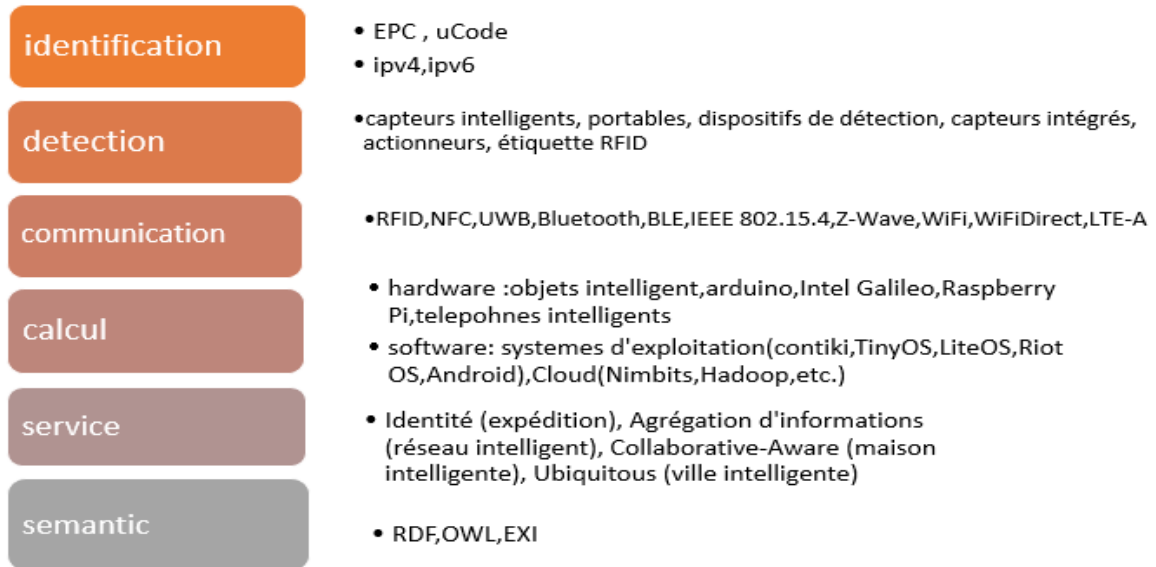
Le but ultime de toutes les applications IoT est d'atteindre le niveau des services omniprésents. Cependant, cette fin n'est pas facilement réalisable car il y a beaucoup de difficultés et de défis à relever. La plupart des applications existantes fournissent des services liés à l'identité, à l'agrégation d'informations et à la collaboration.

### 1.4.6 Sémantiques

La sémantique dans l'IoT fait référence à la capacité d'extraire intelligemment des connaissances par différentes machines pour fournir les services requis.

L'extraction des connaissances comprend la découverte et l'utilisation des ressources et la modélisation des informations. En outre, cela comprend la reconnaissance et l'analyse des données pour donner un sens à la bonne décision de fournir le service exact. Ainsi, la sémantique représente le cerveau de l'IoT en envoyant des demandes à la bonne ressource. Cette exigence est prise en charge par les technologies du Web sémantique telles que le Resource Description Framework (RDF) et le Web OntologyLanguage (OWL). En 2011, le consortium World Wide Web (W3C) a adopté le format Efficient XML Interchange (EXI) comme recommandation [18].

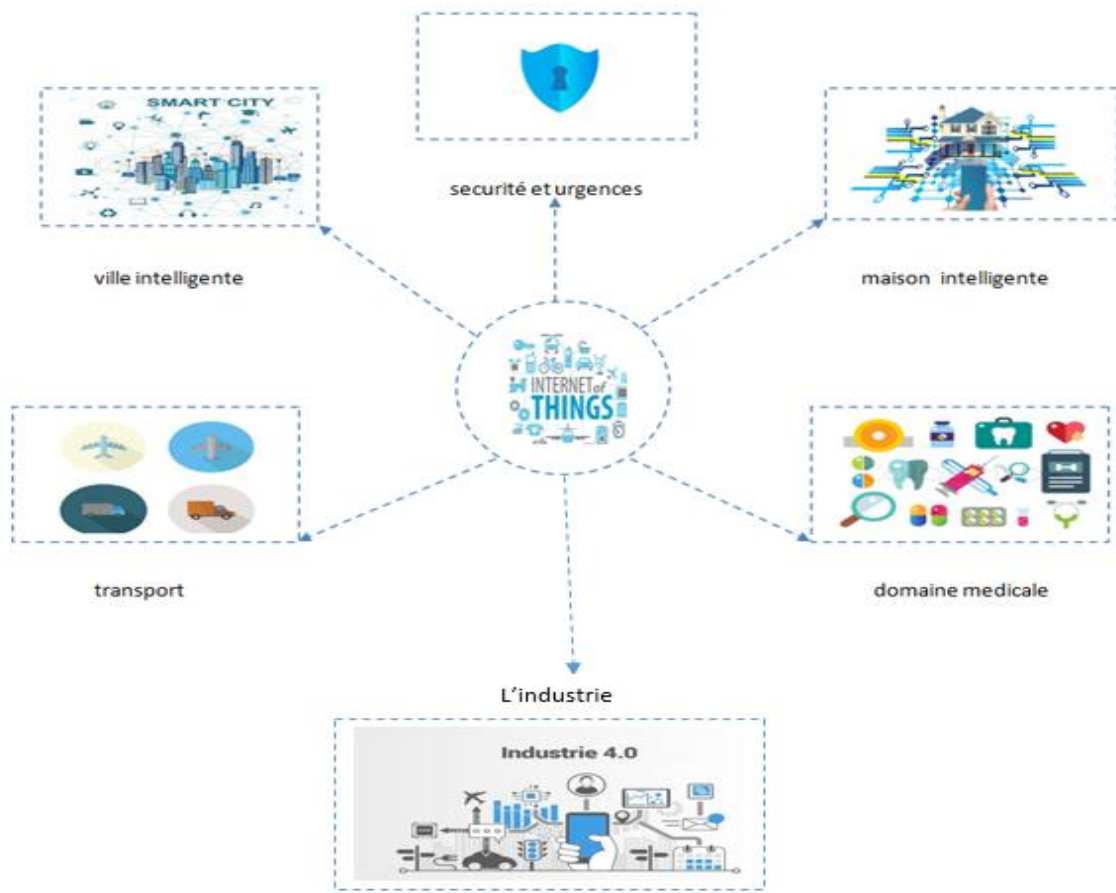
La figure 1.3 ci-dessous montre les différents éléments de l'IoT ainsi ses différentes technologies.



**Figure 1.3** Les technologies de l'IoT [3]

## 1.5 Les domaines d'application d'IoT

Plusieurs domaines d'application sont touchés par l'IoT, Parmi ces principaux domaines nous citons: le domaine de la sécurité, le domaine du transport, la ville intelligente, la domotique et le domaine de la santé [19] [20].



**Figure 1.4** Domaines d'application d'IoT

### 1.5.1 Ville intelligente (smart cities)

- Suivi de la disponibilité des parkings en ville.
- Surveillance des vibrations et des conditions matérielles dans les bâtiments, ponts et monuments historiques.
- Détecter les appareils Android, iPhone et en général tout appareil qui fonctionne avec les interfaces Bluetooth ou Wi-Fi.
- Mesure de l'énergie rayonnée par les stations cellulaires et routeurs Wi-Fi.
- Surveillance des véhicules et des niveaux piétons pour des itinéraires de conduite et de marche optimisés.

### **1.5.2 Sécurité et urgences**

- Contrôle d'accès au périmètre: détection et contrôle des personnes non autorisées et restreintes.
- Présence de liquide: détection de liquide dans les centres de données, les terrains de construction sensibles et les entrepôts pour éviter les pannes et la corrosion.
- Niveaux de rayonnement: dans les environs des centrales nucléaires, mesure distribuée des niveaux de rayonnement pour générer des alertes de fuite.
- Gaz explosifs et dangereux: détection des fuites et des niveaux de gaz dans les environnements industriels, les environs des usines chimiques et à l'intérieur des mines.

### **1.5.3 Les transports**

Depuis la création de l'IoT en 1999, le nombre des véhicules intelligents sont en croissance, presque tous les véhicules vendus aujourd'hui dans le monde renferment déjà des capteurs et de moyens de communication pour traiter la congestion du trafic, la sécurité, la pollution et le transport efficace des marchandises, etc.

L'objectif est qu'une voiture soit capable de communiquer de façon autonome avec d'autres véhicules ou une centrale de surveillance pour prévenir les accidents et réduire les coûts d'assurance.

### **1.5.4 Domaine médical**

- Assistance aux personnes âgées ou handicapées vivant de façon indépendante.
- Réfrigérateurs médicaux: surveillance et contrôle des conditions à l'intérieur des congélateurs stockant des médicaments, des vaccins et des éléments organiques.
- Soins aux sportifs: surveillance des signes vitaux dans les centres et terrains de haute performance.
- Rayonnement ultraviolet: mesure des rayons solaires UV pour avertir les personnes de ne pas être exposées à certaines heures.

### 1.5.5 La domotique

- Utilisation d'énergie et d'eau: surveillance de la consommation d'énergie et d'eau pour obtenir des conseils sur la façon d'économiser des coûts et des ressources.
- Appareils de contrôle à distance: allumer et éteindre des appareils à distance pour éviter les accidents et économiser de l'énergie.
- Systèmes de détection des intrusions: détection des fenêtres et ouvertures de portes et violations pour empêcher les intrus.

### 1.6 Qualité de service

La qualité de service (QoS) est un autre facteur important pour l'IoT. La QoS peut être définie comme une mesure pour évaluer la qualité, l'efficacité et les performances des dispositifs IoT, des systèmes et de l'architecture. Les métriques QoS importantes et requises pour les applications IoT sont la fiabilité, le coût, la consommation d'énergie, la sécurité, la disponibilité et le temps de service. Un écosystème IoT plus intelligent doit répondre aux exigences des normes QoS. De plus, pour garantir la fiabilité de tout service et appareil IoT, ses métriques QoS doivent être définies en premier. De plus, les utilisateurs peuvent également être en mesure de spécifier leurs besoins et exigences en conséquence. Plusieurs approches peuvent être déployées pour l'évaluation de la qualité de service, mais comme mentionné par (White et al). il existe un compromis entre les facteurs de qualité et les approches. Par conséquent, des modèles de bonne qualité doivent être déployés pour surmonter ce compromis. Il existe certains modèles de bonne qualité disponibles dans la littérature tels que ISO / IEC25010 et OASIS-WSQM qui peuvent être utilisés pour évaluer les approches utilisées pour l'évaluation de la QoS. Ces modèles fournissent un large éventail de facteurs de qualité qui sont tout à fait suffisants pour l'évaluation de la qualité de service des services IoT[21].

### 1.7 Industrial internet of things

IIoT signifie Internet industriel des objets ou IoT industriel qui, à l'origine, faisait principalement référence à un cadre industriel dans lequel un grand nombre d'appareils ou de machines sont connectés et synchronisés grâce à l'utilisation d'outils logiciels et de technologies de troisième plate-forme dans une machine à machine et contexte de l'internet des objets, plus tard un contexte Industrie 4.0 ou Internet industriel.



Aujourd'hui, l'IIoT est principalement utilisé dans le cadre des applications de l'Internet des objets en dehors du marché de l'espace des consommateurs et de l'IoT d'entreprise, en tant que terme générique pour les applications et les cas d'utilisation dans plusieurs secteurs industriels.

L'Internet des objets industriel ou IIoT est défini comme «des machines, des ordinateurs et des personnes permettant des opérations industrielles intelligentes utilisant des analyses de données avancées pour des résultats commerciaux transformationnels».

### **1.7.1 Industry 4.0**

L'industrie 4.0 fait référence à une nouvelle phase de la révolution industrielle qui se concentre fortement sur l'interconnectivité, l'automatisation, l'apprentissage automatique et les données en temps réel. L'industrie 4.0, parfois appelée IIoT ou fabrication intelligente, associe la production physique et les opérations à la technologie numérique intelligente, à l'apprentissage automatique et au Big Data pour créer un écosystème plus holistique et mieux connecté pour les entreprises qui se concentrent sur la fabrication et la gestion de la chaîne d'approvisionnement. Bien que chaque entreprise et organisation opérant aujourd'hui soit différente, elles sont toutes confrontées à un défi commun: le besoin de connectivité et l'accès à des informations en temps réel sur les processus, les partenaires, les produits et les personnes.

L'un des plus grands avantages de l'Internet des objets industriel réside dans la réduction des erreurs humaines et du travail manuel, l'augmentation de l'efficacité globale et la réduction des coûts, à la fois en termes de temps et d'argent. Nous ne pouvons pas non plus oublier les fondements possibles de l'IIoT dans le contrôle de la qualité et la maintenance[1].

## **1.8 Défis de l'IoT**

Plusieurs défis se dressent entre l'idée conceptuelle de l'IoT et le déploiement complet de ses applications dans notre vie quotidienne [22], [3], [21]:

- **Limitations des ressources**

L'architecture IoT à ressources limitées a été un obstacle majeur à la définition d'un mécanisme de sécurité robuste. Contrairement aux paradigmes conventionnels, les algorithmes cryptographiques doivent être limités pour fonctionner dans ces contraintes. Avec toutes les diffusions ou multidiffusions requises pour l'échange de

clés ou de certificats, le stockage ainsi que les besoins énergétiques doivent être pris en compte afin de fournir une mise en œuvre réussie des protocoles de sécurité et de communication pour l'IoT. Cela implique de reconcevoir ces protocoles pour qu'ils soient légers et écoénergétiques malgré des calculs complexes et l'amélioration des techniques de récupération d'énergie.

- **Mobilité et dynamique des réseaux**

Les nœuds capteurs dans les LLNs sont conçus pour être stationnaires dans la grande majorité des scénarios, cependant, il existe encore des cas dans lesquels il existe un nombre considérable de nœuds mobiles. Par exemple, dans les applications de surveillance de la santé, le mode de déploiement habituel est mobile car les nœuds de capteurs sont attachés au corps humain pour surveiller à distance les conditions de santé pendant que les sujets vontquent à leurs occupations. Par conséquent, les stratégies de routage générales doivent tenir compte de la mobilité possible des nœuds.

- **Passage à l'échelle**

Dans l'IoT, des milliards d'objets devraient faire partie du réseau. Ainsi, les systèmes et applications qui s'exécutent sur eux devront gérer cette quantité sans précédent de données générées.

- **Fiabilité**

La fiabilité se réfère au bon fonctionnement du système basé sur sa spécification. La fiabilité vise à augmenter le taux de réussite de la prestation de services IoT. Elle a une relation étroite avec la disponibilité car par la fiabilité, nous garantissons la disponibilité des informations et des services dans le temps. La fiabilité est encore plus critique et a des exigences plus strictes en matière de domaine des applications d'intervention d'urgence.

- **Sécurité**

Alors qu'un nombre croissant et une variété d'appareils connectés sont introduits dans les réseaux IoT, la menace de sécurité potentielle s'intensifie. Bien que l'IoT améliore la productivité des entreprises et améliore la qualité de vie des personnes, l'IoT augmentera également les surfaces d'attaque potentielles pour les pirates et autres

cybercriminels. Les appareils IoT présentent des vulnérabilités en raison du manque de chiffrement de transport, des interfaces Web non sécurisées, d'une protection logicielle inadéquate et d'une autorisation insuffisante. Certaines applications IoT prennent en charge des infrastructures sensibles et des services stratégiques tels que le réseau intelligent et la protection des installations. D'autres applications IoT généreront de plus en plus d'énormes quantités de données personnelles sur la santé et la situation financière que les entreprises pourront exploiter pour leurs entreprises. Le manque de sécurité et de confidentialité créera une résistance à l'adoption de l'IoT par les entreprises et les particuliers.

- **Confidentialité**

Comme les données IoT transitent par plusieurs sauts dans un réseau, un mécanisme de cryptage approprié est nécessaire pour garantir la confidentialité des données. En raison d'une intégration diversifiée des services, des appareils et du réseau, les données stockées sur un appareil sont vulnérables à la violation de la vie privée en compromettant les nœuds existant dans un réseau IoT. Les appareils IoT susceptibles d'attaques peuvent entraîner un attaquant à affecter l'intégrité des données en modifiant les données stockées à des fins malveillantes.

## **1.9 Conclusion**

L'Internet des objets offre la possibilité de connecter des milliards d'objets du quotidien à internet, leur permettant d'interagir et de partager des données. Cette perspective ouvre de nouvelles portes vers un avenir où le monde réel et virtuel fusionnera de manière transparente grâce au déploiement massif de dispositifs embarqués. L'IoT a le potentiel d'ajouter une nouvelle dimension au secteur des TIC en permettant les communications avec et entre les objets intelligents, conduisant à la vision de n'importe quand, n'importe où, n'importe quel média et tout paradigme de communication.

Dans ce premier chapitre nous avons décrit les éléments de l'IoT ainsi que leurs principaux domaines d'application, l'architecture IoT et les différents défis.

Le chapitre suivant est dédiée à une étude sur la norme IEEE 802.15.4, son amendement et le routage dans les réseaux LLNs.

# Chapitre 02 : IEEE 802.15.4e et le routage dans les réseaux IoT industriel

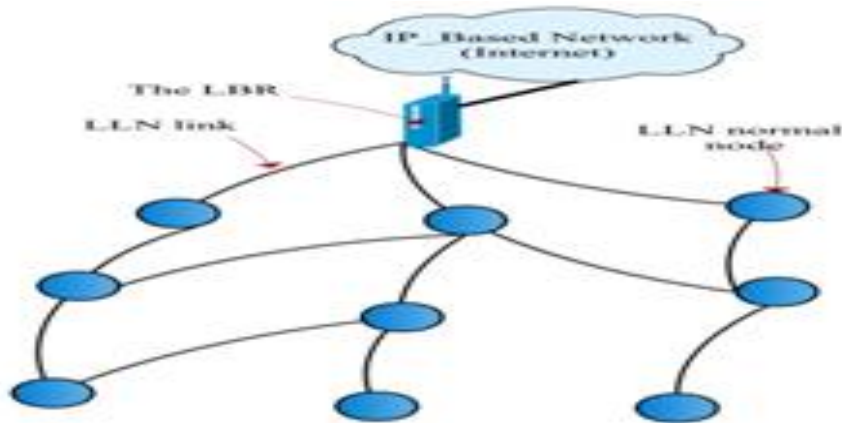
## 2.1 Introduction

En ce qui concerne IoT industriel, il existe différentes normes qui permettent la construction des réseaux LLNs. La norme IEEE 802.15.4 a réussi à se positionner comme la norme de facto pour ce type de réseau [23], ayant inspiré des systèmes industriels différents, tels que WirelessHART [24], ISA100.11a [25], WISA [26] et ZigBee [27]. De la même manière, plusieurs protocoles de routage furent proposés pour répondre aux exigences de ces réseaux telles qu'une limitation d'énergie, perte de données, puissance de calcul et capacité de mémoire très limitées, etc. Le protocole RPL a été proposé comme solution standard pour la couche de routage pour construire des chemins vers un nœud racine.

Ce chapitre est une présentation de l'IEEE 802.15.4 et les principaux protocoles de routage. Dans cette optique nous allons détailler le TSCH et le protocole RPL.

## 2.2 Low Power and Lossy Networks(LLN)

Les réseaux à faible puissance et à pertes (LLN) sont une classe de réseau dans laquelle les routeurs et leur interconnexion sont contraints. Les routeurs LLN fonctionnent généralement avec des contraintes de puissance de traitement, de mémoire et d'énergie (alimentation par batterie). Leurs interconnexions se caractérisent par des taux de perte élevés et de faibles débits de données. Les LLN comprennent de quelques dizaines à des milliers de routeurs. Les flux de trafic pris en charge incluent le point à point (entre les périphériques à l'intérieur du LLN), le point à multipoint (d'un point de contrôle central à un sous-ensemble des dispositifs à l'intérieur du LLN) et multipoint à point (des dispositifs à l'intérieur du LLN vers un point de contrôle central) [28].



**Figure 2. 1** Une architecture de réseau à faible puissance et à pertes (LLN) [29]

Pour ce type de réseau nous allons parler de standard IEEE 802.15.4 de la couche mac, 6lowpan et le protocole de routage RPL de la couche réseau.

## 2.3 Standard IEEE 802.15.4

Il existe plusieurs protocoles de communication sans fil qui supportent différents types d'applications, telles que les communications vidéo, vocales et data. Chacun de ces protocoles met un compromis entre les propriétés telles que le débit, la latence, l'efficacité énergétique, l'un de ces protocoles est le IEEE 802.15.4 [30] [31] qui a été publiée pour la première fois en 2003 pour les LR-WPAN.

### 2.3.1 LR-WPAN

Au cours de la dernière décennie, plusieurs normes visant les communications sans fil à faible puissance ont été définies pour répondre aux besoins en matière de qualité de service des communications industrielles [31],[32].Le protocole de communication IEEE 802.15.4 [30] est destiné aux réseaux sans fil de la famille des LR-WPAN du fait de leur faible consommation, de leur faible portée et du faible débit.

Le protocole ne définit que la couche physique et la couche d'accès au canal de communication alors que quelques propositions, telles que les protocoles ZigBee [27] ou RPL [33], ont été établis pour compléter la pile protocolaire de communications [34].

### 2.3.2 Composants du LR-WPAN

Dans la norme IEEE 802.15.4 [30], les périphériques peuvent être classés en dispositifs entièrement fonctionnels (FFD) et dispositifs à fonctions réduites (RFD), les nœuds routeurs FFD pour transférer des données via un routage multi-saut et les nœuds terminaux RFD qui impliquent une pile de protocoles légers et économiques.

Le coordinateur PAN est un nœud FFD qui agit en tant que contrôleur principal responsable de la synchronisation du temps, auquel d'autres périphériques peuvent être associés pour la formation d'un réseau. Un FFD peut également agir en tant que coordinateur fournissant des services de synchronisation et d'acheminement locaux à ses voisins. Chaque coordinateur doit être associé à un coordinateur PAN et celui-ci forme son propre réseau s'il ne trouve pas d'autres réseaux à proximité.

Le dispositif RFD est un nœud final d'un réseau IEEE 802.15.4, Un nœud RFD est destiné à des applications extrêmement simples, telles qu'un commutateur de lumière ou un capteur infrarouge passif, qui sont généralement synchronisées avec un coordinateur et qui n'ont pas de fonctionnalités de routage.

### 2.3.3 Limitations de IEEE802.15.4

Les performances du protocole mac IEEE 802.15.4, ont été minutieusement étudiées dans le passé [35], [36], [37]. En conséquence, un certain nombre de limitations et d'anomalies ont été identifiées, dont on peut citer :

- Délai illimité : Étant donné que le protocole mac IEEE 802.15.4, repose sur un algorithme CSMA-CA, aucun délai de temps maximal n'a été fixé afin que les données atteignent leurs destinations finales.
- Fiabilité de communication limitée : L'inefficacité de l'algorithme CSMA-CA utilisé pour l'accès au canal cause un taux de livraison très faible, même lorsque le nombre de nœuds n'est pas très élevé.
- Pas de protection contre les interférences et le brouillage multi-chemin : Le protocole mac IEEE 802.15.4 utilise un canal unique et ne possède pas de mécanisme de saut de fréquence intégré pour atténuer les effets négatifs des interférences et de brouillage multi-chemin.

Pour ces raisons, la norme IEEE 802.15.4 ne convient pas à de nombreux scénarios critiques, dans lesquels les applications ont des exigences strictes en termes de rapidité et de fiabilité.

## **2.4 Standard IEEE 802.15.4e**

Le standard IEEE 802.15.4 est la référence pour les réseaux de capteurs sans fil RCSFs. Ces réseaux sont bien connus pour leur facilité d'installation, leur coût extrêmement faible, tout en maintenant une pile de protocoles simple et flexible. Néanmoins, de nombreuses limitations et faiblesses ont été identifiées, qui rendent le standard inapproprié à de nombreuses applications. Pour surmonter ces limitations, un groupe de travail nommé 802.15 Task Group 4e a été créé en 2008 pour améliorer et ajouter des fonctionnalités au code de la norme IEEE 802.15.4. En 2012, le Conseil d'Association des Standards de l'IEEE a approuvé la norme IEEE 802.15.4e [38] en tant qu'un amendement à la norme IEEE 802.15.4 afin de mieux soutenir les divers domaines d'application industriels. La technologie de base est inspirée des technologies de réseau industrielles antérieures telles que WirelessHart et ISA 100.11.a .La norme améliorée fournit de nouvelles fonctionnalités telles qu'une faible consommation d'énergie, des éléments d'information, des balises améliorées, des métriques de performance mac et une association rapide.

### **2.4.1 Améliorations fonctionnelles générales**

La norme IEEE 802.15.4e étend de la norme IEEE 802.15.4 précédente en introduisant des modes de comportement mac, c'est-à-dire des nouveaux protocoles mac conçus pour prendre en charge des domaines d'application spécifiques. Elle apporte aussi des améliorations mac et des mécanismes fonctionnels généraux non liées à un domaine d'application en particulier. Le standard IEEE 802.15.4e introduit les améliorations fonctionnelles générales suivantes :

- ✓ Énergie de bas niveau (LE) : Ce mécanisme est destiné aux applications où l'efficacité énergétique est prioritaire à la latence. Cela permet à un nœud de fonctionner à un très faible cycle (1% ou moins). Ce mécanisme est important dans le contexte de l'Internet des Objets, car les protocoles ont été conçus en supposant que les nœuds capteurs sont toujours actifs.
- ✓ Les éléments d'information (IE) : C'est un mécanisme extensible pour échanger des informations dans la sous-couche mac.

- ✓ Les balises améliorées (EB) : Ce sont une extension des trames de balises IEEE 802.15.4 qui permettent d'offrir une plus grande flexibilité pour le protocole mac. Ils permettent de créer des trames spécifiques à l'application, en incluant les IE pertinents.
- ✓ Les trames multi-usage : Ils sont basés sur les IE et fournissent un format de trame flexible pouvant traiter un certain nombre d'opérations mac.
- ✓ La mesure de performance mac : C'est un mécanisme qui permet de fournir à la couche réseau et aux couches supérieures des informations sur la qualité des canaux afin de prendre des décisions appropriées.
- ✓ Association rapide (FastA) : La procédure d'association IEEE 802.15.4 introduit un délai important afin d'économiser de l'énergie. Pour les applications critiques, le temps de latence est prioritaire par rapport à l'efficacité énergétique. Par conséquent, le mécanisme FastA permet à un nœud de s'associer dans un laps de temps réduit.

## 2.4.2 Modes de comportement de la couche mac

Le standard IEEE 802.15.4e définit cinq nouveaux modes de comportement mac. Chacun prend en charge des domaines d'application spécifiques :

- ❖ TSCH : Il cible des domaines d'application tels que l'automatisation industrielle et le contrôle des processus et offre une prise en charge des communications multi-sauts et multicanaux via l'approche TDMA.
- ❖ DSME : Son objectif est de prendre en charge les applications industrielles et commerciales avec des exigences strictes en termes de rapidité et de fiabilité. À cette fin, DSME combine un accès au support par division et par répartition dans le temps et propose deux modes différents de diversité de canaux. Ce mode de comportement est conçu spécifiquement pour les réseaux maillés et multi-sauts.
- ❖ LLDN : Il est conçu pour les réseaux à un seul canal et à un seul saut. Il est destiné à l'automatisation dans l'industrie qui requiert une latence très faible.
- ❖ AMCA : Il est destiné aux domaines d'application nécessitant de vastes déploiements, tels que les réseaux intelligents, les réseaux de surveillance et les réseaux de contrôle



des processus. Le mode AMCA repose sur une adaptation multi-canal asynchrone et ne peut être utilisée que sur des Non-Beacon-Enabled PANs.

- ❖ BLINK : Ce mode est destiné aux domaines d'application tels que l'identification des articles ou de personnes, la localisation et le suivi. Plus précisément, il permet à un nœud de communiquer son ID à d'autres nœuds sans association préalable ni accusé de réception. Les paquets BLINK sont généralement envoyés par des périphériques "transmit only" via le protocole Aloha.

## 2.5 Time Slotted Channel Hopping TSCH

L'amendement IEEE802.15.4e de cette norme définit un mécanisme d'accès moyen appelé Time Slotted Channel Hopping (TSCH), qui cible les scénarios agressifs, tels que les environnements industriels. TSCH nous permet d'effectuer des sauts de canal de fréquence entre différentes transmissions pour atténuer les fortes interférences de ce type de scénario. Le mécanisme utilisé par TSCH soit très efficace dans des situations de fonctionnement en régime permanent, réduisant considérablement les pertes dues aux interférences, il entrave également les tâches de connexion au réseau et de première synchronisation [39].

### 2.5.1 Fonctionnement du TSCH

Le mode TSCH est principalement conçu pour ordonnancer la communication de données aux nœuds du réseau ainsi que leurs liens respectifs. La communication entre les nœuds se déroule en suivant un ordonnancement (Schedule), ainsi les nœuds voisins, dont les transmissions peuvent s'interférer, ne seront pas planifiés à transmettre sur le même slot de temps (slot offset) et décalage de canal (Channel Offset). L'ordonnancement est projeté sous forme de matrice composée du slot offset et du channel offset, où chaque cellule représente un lien spécifique et peut être réservée pour un seul lien ou être partagée entre de multiples liens. En cas de collision de ces dernières, un système appelé backoff est défini par le protocole mac IEEE 802.15.4e dans le but de résoudre ce conflit. De plus, pour les slots partagés, l'algorithme CSMA/CA slotté est utilisé par les nœuds du PAN-TSCH [40], [38]. Le standard IEEE 802.15.4e à mode TSCH décrit seulement le mode d'exécution de la couche mac, mais ne spécifie pas la façon de maintenir, construire et mettre à jour l'ordonnancement ni de s'adapter aux contraintes de trafic du réseau [39].

## 2.5.2 Structure de la slotframe

En mode TSCH, les nœuds se synchronisent sur une trame périodique (slotframe) composée d'un certain nombre d'intervalles de temps (timeslots). Une slotframe est un enchaînement d'un ensemble de timeslots. La durée de chaque timeslot permet de transmettre un paquet et de recevoir un accusé de réception. La taille d'une slotframe est définie par le nombre de timeslots dans la slotframe. Chaque slotframe se répète de façon cyclique, formant ainsi un ordonnancement de communication. Chaque nœud obtient des informations sur la synchronisation, les sauts de canal, le timeslot et la slotframe à partir des EB qui sont envoyées périodiquement par d'autres nœuds afin d'introduire le réseau. Lorsqu'un nœud reçoit un EB valide, il se synchronise sur le réseau, initialise la slotframe et envoie ses propres balises. À partir de ce moment, la slotframe se répète automatiquement selon la notion du temps partagée par les nœuds et ne requiert pas de balises pour initier les communications.

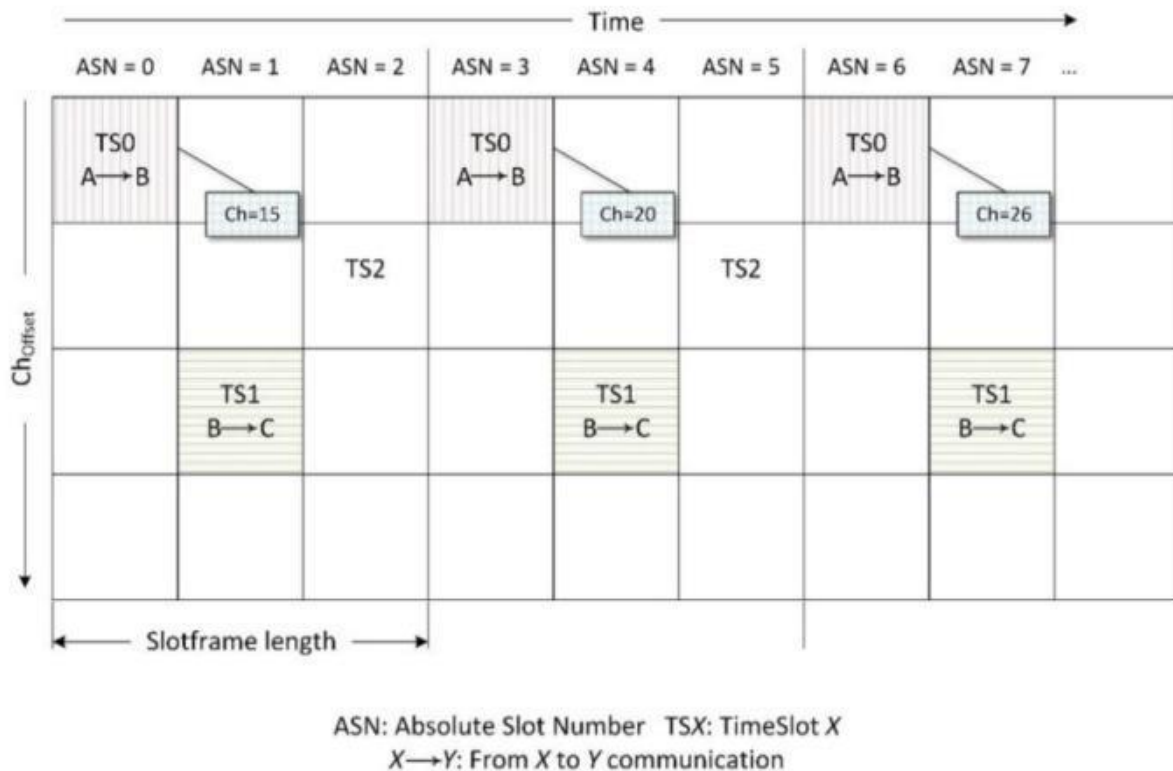
## 2.5.3 Saut de canal

La communication multi-canal de mode TSCH repose entièrement sur les sauts de canal. Le mode TSCH peut utiliser jusqu'à 16 canaux de communication définis par un décalage de canal (appelé channel offset). Dans le mode TSCH, le lien de communication entre une paire de nœuds est défini par le couple  $[n, \text{channel offset}]$ . Il s'agit d'une affectation de la communication par paire des timeslots «n» et de channel offset consécutifs. La fréquence utilisée pour la communication peut être définie par la fonction  $f$ [39].

$$f = F \{(\text{choffset} + \text{ASN}) \bmod (\text{Nch})\} \quad (2.1)$$

Dans l'équation (2.1),  $\text{choffset}$  est un paramètre qui permet d'utiliser différents canaux dans le même intervalle de temps,  $\text{ASN}$  aide à déterminer le nombre de timeslots qui se sont écoulés depuis que le nœud a rejoint le réseau,  $\text{Nch}$  est défini comme étant le nombre de canaux utilisés pour le réseau actuel, vu qu'utiliser tous les 16 canaux n'est pas obligatoire. Certains canaux peuvent ne pas être utilisés si on cherche à améliorer l'efficacité énergétique ou si leur qualité s'est détériorée. Comme mentionné précédemment, la fonction  $F$  peut être définie comme une table de correspondance. À partir de l'équation (2.1), il est à noter qu'un canal différent ( $\text{Ncanal}$ ) peut être implémenté avec le même offset pour un ASN incrémenté, c'est-à-dire le mécanisme de saut de canal peut être utilisé avec une fréquence différente sur le même lien.

La figure 2.2 illustre un exemple de programme d'intervalle de temps et de décalage de canal. La communication entre une paire de périphériques est planifiée pendant le même intervalle de temps et le même décalage de canal, et est répétée chaque trame de slot. Le décalage de canal offre la possibilité d'utiliser le même créneau horaire avec une fréquence différente à chaque itération du cadre de créneau le canal utilisé pour établir une connexion dans chacun de ces emplacements change selon l'équation (2.1).



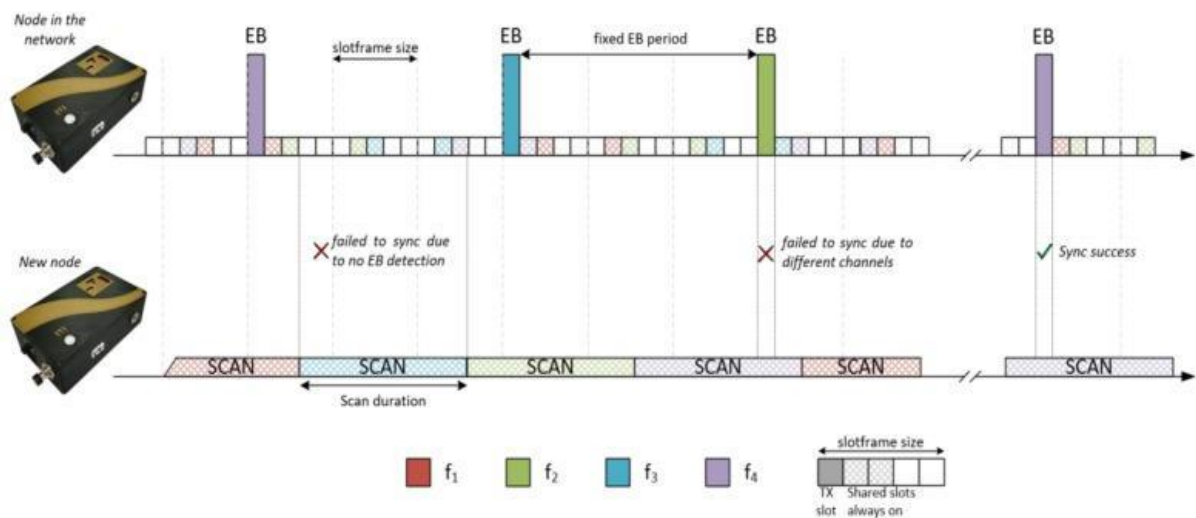
**Figure 2. 2** La matrice de décalage temporel et de canal utilisée dans la méthode TSCH [39]

### 2.5.4 Formation du réseau

Dans la première étape, les nœuds qui sont déjà connectés au réseau envoient périodiquement des EB en fonction du temps de transmission EB (EBTT), tandis que les nœuds qui attendent de se synchroniser écoutent, balayant les canaux définis, pendant un temps de balayage appelé TSCAN. Le nombre total de canaux est défini par le nombre de canaux (NC); cependant, il est possible d'avoir un nombre total de canaux différents pour la phase de balayage, qui est déterminé par le nombre de canaux de balayage (NSC).

La figure 2.3 représente les trames à deux nœuds dans cette phase de synchronisation TSCH en utilisant quatre fréquences (f1 à f4). Il existe des circonstances dans lesquelles un nœud en

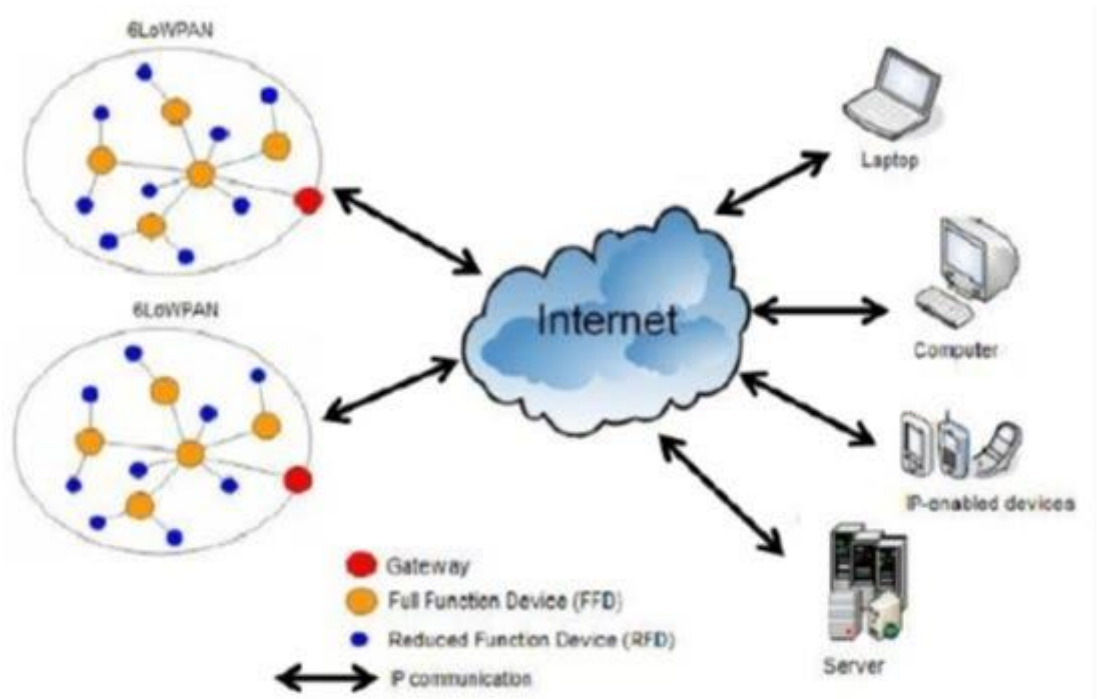
cours d'analyse ne trouve aucun message de signalisation. Selon la configuration des paramètres EBTT, TSCAN, NC et NSC, il peut être observé que le processus de synchronisation peut être plus facile ou plus difficile, car il est influencé par la quantité de messages de synchronisation qui sont transmis et par le temps d'écoute dans chaque canal [39].



**Figure 2. 3** Phase de balayage et de synchronisation de la méthode TSCH. EB [39]

## 2.6 IPv6 Low power Wireless Personal Area Networks (6LoWPan)

6LoWPAN est un réseau de communication simple à faible coût qui permet la connectivité sans fil dans des applications avec une puissance limitée et des exigences de débit réduites car il fournit un réseau IPv6 sur les réseaux IEEE 802.15.4 [41]. Il est formé par des appareils compatibles avec la norme IEEE 802.15.4 et caractérisés par une courte portée, un faible débit binaire, une faible puissance, une faible utilisation de la mémoire et un faible coût, où son architecture est illustrée dans la figure 2.4 [42]. Lorsqu'un nœud de capteur de capacité de traitement inférieure dans un 6LoWPAN ou un dispositif dit à fonction réduite (RFD) souhaite envoyer son paquet de données à un périphérique compatible IP en dehors du 6LoWPAN, il envoie d'abord le paquet au nœud de capteur de capacité de traitement supérieure ou ainsi- appelé dispositif de fonction complète (FFD) dans le même PAN. Les FFD qui réagissent comme un routeur dans 6LoWPAN transmettront le paquet de données saut par saut à la passerelle 6LoWPAN. La passerelle 6LoWPAN qui se connecte au 6LoWPAN avec le domaine IPv6 transmet ensuite le paquet au périphérique compatible IP de destination à l'aide de l'adresse IP.



**Figure 2. 4** Architecture 6LoWPAN [43]

### 2.6.1 Mécanismes dans la couche d'adaptation 6LoWPAN

L'unité de transmission maximale (MTU) pour un paquet IPv6 sur IEEE 802.15.4 est de 1280 octets. Cependant, la taille de trame mac maximale définie par IEEE 802.15.4 est de 127 octets, où 25 octets sont réservés pour la surcharge de trame et ne laissent que 102 octets pour la charge utile. La situation empire si la couche liaison impose une surcharge supplémentaire à des fins de sécurité en ajoutant un en-tête de sécurité auxiliaire dans la trame mac, ce qui dans le cas maximum ne laisse que 81 octets pour le paquet IPv6. Ainsi, un paquet IPv6 complet ne tient pas dans une trame IEEE 802.15.4. De plus, étant donné que l'en-tête IPv6 dans un paquet IPv6 fait 40 octets, il ne reste que 41 octets pour les couches supérieures. En réservant l'en-tête UDP (User Datagram Protocol) 8 octets ou l'en-tête TCP (Transmission Control Protocol) 20 octets ajouté au niveau de la couche de transport, le paquet IPv6 ne laisse pratiquement que quelques octets d'espace pour l'utilisation des données d'application. Par conséquent, afin de mettre en œuvre la connexion transparente de la couche mac et de la couche réseau IPv6, le groupe de travail 6LoWPAN a suggéré d'ajouter une couche d'adaptation entre la couche mac et la couche réseau pour obtenir la compression d'en-tête, la fragmentation et le transfert de couche deux [43], [44], [45].

## 2.7 Protocoles de routages

### 2.7.1 Protocoles de routage proactifs

Dans les réseaux utilisant un protocole de routage proactif, chaque nœud conserve une ou plusieurs tables représentant la topologie entière du réseau. Ces tableaux sont mis à jour régulièrement afin de maintenir à jour les informations de routage de chaque nœud vers tous les autres nœuds. Pour conserver les informations de routage à jour, les informations de topologie doivent être échangées régulièrement entre les nœuds, ce qui entraîne une surcharge relativement élevée sur le réseau. D'un autre côté, les routes seront toujours disponibles sur demande. Une approche proactive du routage MANET vise à maintenir une compréhension de la topologie constamment mise à jour. L'ensemble du réseau devrait, en théorie, être connu par tous les nœuds. Il en résulte un surcoût constant du trafic de routage, mais pas de retard initial dans la communication [46], [47]. Voici quelques exemples sur les protocoles de routage proactifs:

- IPv6 Routing Protocol for Low Power and Lossy Networks(RPL)
- Wireless Routing Protocol (WRP)
- Global State Routing (GSR)
- Fisheye State Routing Protocol (FSP)

### 2.7.2 Protocoles de routage réactifs

Les protocoles de routage réactifs suivent une procédure de détermination de route. Si un nœud source doit envoyer un paquet au nœud de destination, tout d'abord la route vers le nœud de destination est déterminée, puis une connexion est établie entre ces nœuds. Pour la procédure de détermination de l'itinéraire, les paquets de demande d'itinéraire sont inondés sur tout le réseau. L'inondation est une méthode fiable de diffusion des informations sur le réseau, mais elle utilise la bande passante et crée une surcharge réseau, le routage réactif diffuse les demandes de routage chaque fois qu'un paquet a besoin d'être routé, cela peut entraîner des retards dans la transmission des paquets lors du calcul des itinéraires, mais avec très peu de trafic de contrôle surcharge et à une utilisation en mémoire généralement plus faible que le protocole de routage proactif, cela augmente l'évolutivité du protocole[46],[47].Quelques exemples sur les protocoles de routage réactifs:

- The Lightweight On-demand ad hoc distance vector routing protocol, next generation

- Dynamic Source Routing (DSR)

### 2.7.3 Protocole de routage Proactif vs Réactif

#### 1. Protocole de routage Proactif:

- Temps de latence minimale en raison de la maintenance des itinéraires en tout temps.
- Peut entraîner une surcharge beaucoup plus élevés en raison des mises à jour fréquentes des itinéraires.
- Lorsqu'un paquet doit être transmis, l'itinéraire est déjà connu.

#### 2. Protocole de routage Réactif:

- Latence plus élevée car les routes doivent être découvertes lorsque le nœud source lance une demande de route.
- Surcharge réduite car les itinéraires ne sont maintenus que sur demande.
- Déterminer un itinéraire uniquement lorsqu'il y a des données à envoyer.

### 2.7.4 Lightweight On-demand Ad hoc Distance-vector routing protocol-next generation (LOADng)

Le protocole LOADng [48],[49], [50]utilise une approche réactive basée sur l'idée que les LLN sont inactifs la plupart du temps, donc une approche proactive générerait une surcharge inutile. Ainsi, LOADng établit un itinéraire vers une destination donnée uniquement à la demande lorsqu'il y a des données à envoyer. Comme l'IETF a beaucoup travaillé sur la conception d'AODV, un protocole réactif pour les MANET, une conséquence logique a été de l'adapter aux LLN afin de le rendre implémentable sur des périphériques à mémoire limitée.

Lorsqu'un périphérique a un paquet à envoyer vers une destination donnée, il consulte une table de routage et invoque LOADng en cas d'entrée invalide. Le protocole inonde un message Route-Request (RREQ) à travers le réseau pour atteindre tous les nœuds. Un nœud recevant un RREQ vérifie s'il s'agit de la destination du message. Sinon, il transmet RREQ à ses voisins. Le nœud apprend également le chemin inverse vers l'expéditeur du message RREQ et l'ajoute à la table de routage. Finalement, le nœud de destination reçoit RREQ et répond en unicast d'un message RouteReply (RREP) vers l'expéditeur de la demande. RREP suit l'itinéraire inverse stocké. Dans le même temps, les nœuds intermédiaires apprennent

l'itinéraire vers la destination. Lorsque RREP atteint l'expéditeur de la demande, la route bidirectionnelle est installée aux nœuds intermédiaires. L'un des principaux inconvénients de LOADng est le délai de découverte d'itinéraire. Pendant le processus de découverte, les paquets sortants sont mis en mémoire tampon, ce qui peut provoquer des pertes dans les périphériques à mémoire limitée. De plus, les inondations sont très inefficaces en énergie, de sorte que les nœuds peuvent souffrir d'un épuisement énergétique. Un autre problème est lié aux collisions de messages de contrôle suite à des inondations, ce qui peut conduire à des retransmissions inutiles [40].

### **2.7.5 Le protocole de routage ipv6 pour LLNs(RPL)**

RPL est un protocole de routage proactif à vecteur de distance basé sur IPv6 conçu par la communauté IETF pour répondre aux exigences de routage d'une large gamme d'applications LLN. Le RPL est optimisé en particulier pour les applications de collecte de données (c'est-à-dire le modèle de trafic MP2P), et il fournit également un support raisonnable pour le modèle de trafic P2MP, tout en fournissant un support indirect pour le modèle P2P [51], [33].

RPL organise son réseau physique en une forme de graphiques acycliques dirigés (DAG) où chaque DAG est enraciné dans une seule destination et est appelé DAG orienté vers la destination (DODAG) en termes de RPL. Le DODAG représente la destination finale du trafic au sein du domaine réseau reliant la topologie à d'autres domaines IPv6 tels qu'Internet. Il est appelé routeur LLN Border Router (LBR) dans le contexte des LLN. RPL utilise le terme routes ascendantes pour faire référence aux routes qui acheminent le trafic des nœuds normaux vers le LBR (c'est-à-dire MP2P) tandis que les routes qui transportent le trafic de la racine DODAG vers les autres nœuds (c'est-à-dire P2MP) sont appelés les routes descendantes. Pour créer les routes ascendantes, chaque nœud du réseau doit sélectionner l'un de ses voisins comme parent préféré (saut suivant) vers la racine. De même, chaque nœud désireux de participer au routage descendant doit s'annoncer à l'un de ses parents, de préférence le parent préféré. RPL utilise le terme instance pour faire référence à plusieurs DODAG qui partagent les mêmes politiques et mécanismes de routage. Plusieurs instances RPL peuvent coexister simultanément dans une topologie physique spécifique et un nœud peut rejoindre plusieurs instances à la fois. Cependant, dans chaque instance, un nœud est autorisé à s'associer à une seule racine (DODAG).



Pour échanger les informations de routage nécessaires à la construction de la topologie du réseau et des chemins de routage, RPL introduit quatre messages de contrôle de type ICMPv6 comme détaillé ci-dessous [33]:

- DODAG Information Object (DIO): les DIO sont utilisés pour transporter les informations pertinentes et les paramètres de configuration qui permettent à un nœud de découvrir une instance RPL, de rejoindre un DODAG spécifique, de sélectionner un ensemble de parents candidats et de maintenir le DODAG.
- Destination Advertisement Object (DAO): ce message de contrôle permet à un nœud de propager ses informations de destination vers le haut le long du DODAG jusqu'à la racine DODAG afin que les routes descendantes de la racine DODAG vers ses nœuds associés puissent être construites.
- DODAG Information Solicitation (DIS): ce message est utilisé par un nœud RPL pour solliciter un DIO auprès des nœuds voisins afin de rejoindre le DODAG.
- Destination Advertisement Object Acknowledgement (DAOACK) : le DAO-ACK peut être unicast par un destinataire DAO à l'expéditeur DAO pour accuser réception de ce dernier.

### 2.7.5.1 RPL Routes Ascendantes (construction de la topologie dodag)

Le processus de construction du DODAG et des routes ascendantes est contrôlé par les DIO [52]. En plus d'autres informations de routage, les DIO portent le rang, la position relative d'un nœud RPL par rapport à la racine DODAG, et une politique de routage appelée la fonction objective (OF) qui spécifie comment un nœud RPL calcule son rang et sélectionne son parent préféré. Plus précisément, la construction du DODAG est initiée en ayant la racine DODAG multidiffuser des messages DIO vers ses nœuds voisins annonçant son rang et l'OF qui doit être utilisé. Lors de la réception d'un DIO, un nœud RPL (a) ajoute l'adresse de l'expéditeur à son ensemble de parents candidats, (b) calcule son propre rang, (c) sélectionne son parent préféré parmi les parents candidats, et enfin, (d) met à jour le reçu DIO avec son propre rang, puis multidiffuse le rang calculé vers d'autres nœuds voisins.

### 2.7.5.2 RPL Routes Descendantes

Afin de faciliter les modèles de communication P2MP et P2P, des routes descendantes doivent également être établies et maintenues. RPL utilise les messages DAO ICMPv6 à cet effet. Un nœud RPL désireux de s'annoncer comme destination accessible du point de vue racine, unicast un DAO à son parent préféré annonçant son propre préfixe de destination [52].

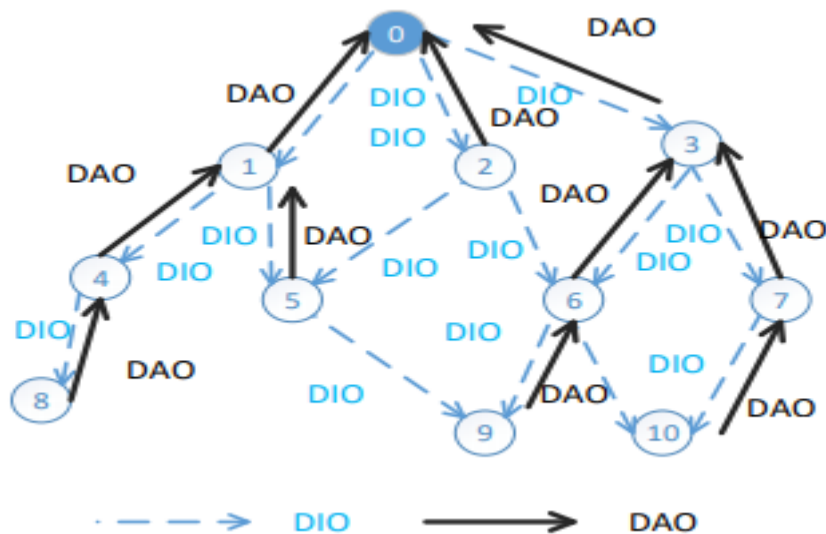
Le traitement du DAO reçu par le parent repose sur le mode de fonctionnement actuel annoncé dans les messages DIO. À cette fin, RPL a spécifié deux modes de création et de maintenance des itinéraires descendants, à savoir le stockage (piloté par table) et le non-stockage (routage source). En mode de stockage, lorsqu'un parent reçoit un DAO de l'un de ses enfants, il: (a) stocke le préfixe de destination annoncé localement dans sa table de routage avec l'adresse de l'expéditeur DAO, comme le prochain saut pour atteindre cette destination; et (b) transmet le DAO reçu, à son tour, à son propre parent préféré pour assurer la propagation de la destination annoncée vers le haut jusqu'à la racine DODAG.

Dans le mode sans stockage, le DAO annoncé porte également l'adresse du parent de la destination en plus du préfixe de destination annoncé. Ici, cependant, un parent recevant un DAO le transmet simplement à son propre parent préféré sans conserver aucun état de routage, jusqu'à ce qu'il soit finalement reçu par la racine DODAG. Une fois que la racine DODAG reçoit le DAO transmis, elle conserve les informations reçues dans sa table de routage sous la forme d'une relation parent-enfant, utilisée plus tard par le plan de données pour construire une route source pour la destination prévue. Par conséquent, lorsque la racine doit communiquer avec une destination spécifique, elle attache la route source de cette destination à l'en-tête du paquet et transmet le paquet au saut suivant. Un nœud de transmission recevant ce paquet inspectera simplement l'en-tête de routage source pour déterminer sur quelle interface il doit envoyer le paquet ensuite.

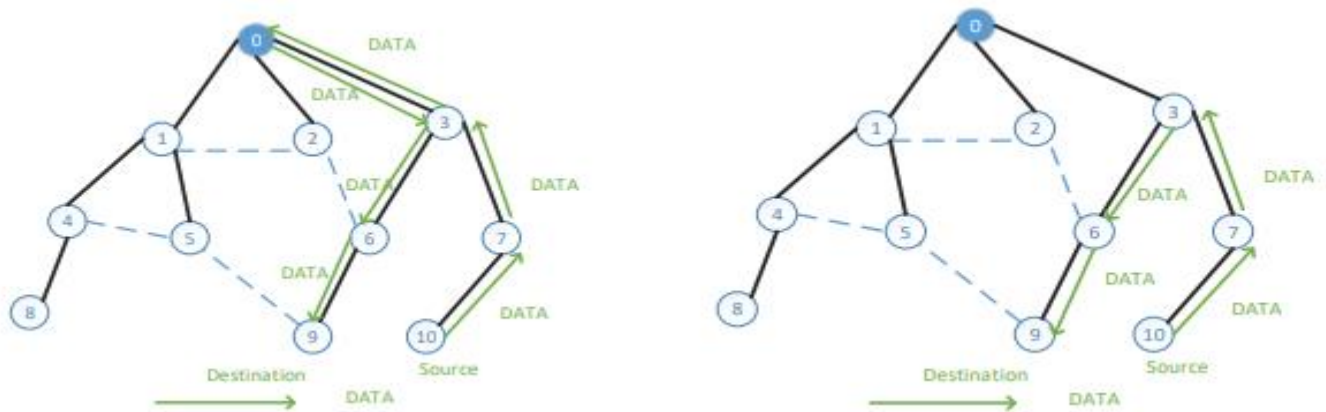
RPL fournit également une prise en charge du modèle de trafic P2P dans lequel un nœud communique avec un autre nœud du réseau. Par conséquent, lorsqu'un nœud doit envoyer un paquet à un autre nœud dans le DODAG, le paquet est d'abord transmis vers le haut du DODAG jusqu'à ce qu'il arrive à un ancêtre qui a un chemin connu vers le nœud de destination. Ensuite, le paquet est transmis vers le bas du DODAG par cet ancêtre via les routeurs intermédiaires et enfin vers le nœud de destination.

### 2.7.5.3 Un exemple illustrant le processus de construction du DODAG

La figure 2.5 montre un exemple simple de propagation DIO et de formation DAG. La racine lance la formation du DAG en publiant des informations sur la racine à l'aide de l'option d'information DAG (DIO), qui contient plusieurs informations, y compris la distance entre le nœud émetteur et le routeur frontalier. Les nœuds ont choisi un nœud comme parent qui fournit le coût le plus bas pour atteindre le routeur de frontière. Les lignes pleines de la figure représentent la relation parent-enfant dans le DODAG, tandis que les lignes pointillées représentent d'autres liens disponibles. RPL a été normalisé pour fonctionner sur deux modes, "non-stockage" et "stockage". En mode de stockage, un nœud du LLN est capable de stocker des tables de routage et des informations de saut suivant pour tous les nœuds de sa sous-arborescence. En mode sans stockage, les nœuds ne stockent pas les routes vers une destination autre que la racine DAG. Le routage des données en mode de non-stockage RPL et en mode de stockage RPL sont montrés sur la figure 2.6 [29].



**Figure 2. 5** Propagation des DIO et formation des DAG [29]



**Figure 2.** 6a) Routage des données en mode non stocké RPL b) Routage des données en mode stocké RPL [29]

### 2.7.5.4 Fonction objective (OFs)

Le terme fonction objective (OF) est utilisé pour décrire l'ensemble de règles et de politiques qui régissent le processus de sélection et d'optimisation des itinéraires de manière à répondre aux différentes exigences des différentes applications. En termes techniques, l'OF est utilisé pour deux objectifs principaux; premièrement, il spécifie comment le classement peut être dérivé d'une ou d'un ensemble de métriques de routage (par exemple énergie, nombre de sauts, latence, débit, fiabilité de la liaison et couleur de la liaison 2), deuxièmement, il définit comment le classement doit être utilisé pour sélectionner la préférence du parent [53]. Actuellement, deux OFs ont été normalisés pour le RPL, à savoir la fonction objective Zéro (OF0) [59] et le Minimum Rank with Hysteresis Objective Function (MRHOF) [54].

### Fonction objective zéro (OF0)

L'OF0 est conçu pour sélectionner le nœud le plus proche du DODAG root en tant que parent préféré sans tentative d'exécution de la charge équilibrage [55]. Le rang d'un nœud ( $R_n$ ) est calculé en ajoutant une valeur scalaire strictement positive ( $rank\_increase$ ) au rang parent préféré sélectionné ( $R_p$ ) selon l'eq2.2 et eq 2.3 comme suit:

$$R_n = R_p + rank\_increase \quad (2.2)$$

$$Rank\_increase = (R_f * S_p + S_r) * MinHopRankIncrease \quad (2.3)$$

Sp : représente une métrique comme ETX ou hop-count.

Rf et Sr: facteurs de normalisation.

### **Minimum Rank with Hysteresis Objective Function (MRHOF)**

Le MRHOF [53] est conçu dans le but d'empêcher un taux de désabonnement excessif dans la topologie du réseau (c'est-à-dire un changement fréquent du parent préféré). Dans le MRHOF, un nœud calcule le coût du trajet à travers chaque voisin en additionnant deux composants; la valeur de la métrique du nœud ou du lien voisin candidat et la valeur de la métrique sélectionnée publiée dans le conteneur de métriques. Après avoir calculé les coûts de trajet de tous les parents candidats, un nœud sélectionne le parent avec le coût de trajet le plus bas comme parent préféré.

Cependant, contrairement à OF0, MRHOF passe à un nouveau parent uniquement si le nouveau coût de trajet calculé minimum est inférieur au coût de trajet du parent préféré d'au moins PARENT\_SWITCH\_THRESHOLD, qui est la partie hystérésis de MRHOF [56]. Si plusieurs parents candidats partagent le même coût de parcours, d'autres critères de départage peuvent être utilisés [56], [53]. ETX sur un lien peut être calculé comme suit:

$$ETX = 1 / (DF * DR) \quad (2.4)$$

DF représente la probabilité de recevoir un paquet du nœud voisin et DR est la probabilité de recevoir un accusé de réception avec succès. Le nœud (n) peut calculer son rang en utilisant cette formule:

$$R(n) = ETX + R(p) \quad (2.5)$$

Où R(p) représente le rang du parent nœuds.

#### **2.7.5.5 Trickle Timer**

L'un des principes de conception clés du RPL est de minimiser les frais généraux de contrôle de routage et les coûts de signalisation afin de réduire la consommation d'énergie et d'améliorer la fiabilité. À cet égard, RPL utilise l'algorithme Trickle [57] [58] pour régir la transmission du trafic de signalisation utilisé pour construire et maintenir le DODAG. L'idée de base derrière Trickle est d'ajuster la fréquence de transmission des messages en fonction des conditions du réseau. Trickle s'appuie sur deux mécanismes simples pour diffuser efficacement les informations de routage. La première consiste à modifier de manière

adaptative le débit de signalisation en fonction des conditions actuellement présentes dans le réseau. Plus précisément, Trickle augmente le taux de transmission lorsqu'un changement dans les informations de routage est découvert (c'est-à-dire qu'une incohérence est détectée) comme moyen de remplir rapidement le réseau avec des informations à jour. Alors que le réseau s'approche de sa phase stable, Trickle réduit de façon exponentielle le débit de transmission pour limiter le nombre de transmissions lorsqu'il n'y a pas de mise à jour à propager. Le deuxième mécanisme utilisé par Trickle est le mécanisme de suppression dans lequel un nœud supprime la transmission de son paquet de contrôle s'il détecte qu'un nombre suffisant de ses voisins ont transmis la même information, limitant ainsi les transmissions redondantes [59].

## 2.8 Défis de routage dans les LLN

La conception de protocoles de routage efficaces pour les LLN repose sur les caractéristiques uniques de ces réseaux. Les ressources de mémoire et de traitements limités, les faibles débits de données et l'alimentation électrique limitée de la majorité des appareils, ainsi que la nature à perte des interconnexions (liaisons), doivent toutes être prises en compte. Dans ce qui suit, nous avons mis en lumière certains des problèmes de conception du processus de routage qui se posent dans le contexte des LLN.

1. **Diversité des applications :** Plusieurs applications sont envisagées pour fonctionner sous l'égide des LLN, notamment l'automatisation de la maison / du bâtiment, les applications industrielles, la surveillance de l'environnement, les applications militaires, etc. Ces diverses applications présentent des caractéristiques et, par conséquent, des exigences différentes en termes de consommation d'énergie, temps de convergence, surcharge de trafic, fiabilité, latence ou autres mesures de performances. Par conséquent, un grand défi pour un protocole de routage LLN est de répondre à toutes ces exigences diverses et conflictuelles dans le budget de ressources de l'application [61], [59].
2. **Modèles de communication:** Le modèle de communication dominant dans les applications LLN est le multipoint à point (MP2P) [33], dans lequel les données sont collectées par un groupe de capteurs et transmises à une destination commune appelée LBR ou puits. D'autres modèles de communication existent également, y compris le Point-to MultiPoint (P2MP), où le récepteur envoie des données aux nœuds de capteur

associés et le Point-to-Point (P2P) dans lequel un nœud de capteur communique entre eux dans le réseau [22] [62]. Cette diversité dans les modèles de communication représente un autre défi lors de la conception des protocoles de routage des LLN.

- 3. Passage à l'échelle:** Il est prévu que les LLN fonctionneront dans des déploiements de densités différentes, allant de quelques voisins par nœud à des centaines[22][63] .Un protocole devrait donc être capable de gérer tous les cas dans la plage viable et ses paramètres devraient être ajustés dynamiquement en fonction de ce qu'il rencontre dans la pratique[33] . En d'autres termes, l'évolutivité est un problème de conception qui devrait être satisfait par un protocole LLN.
- 4. Le manque de ressources:** La nature limitée par les ressources des LLN impose un nouvel ensemble de restrictions au développement de protocoles et de primitives de routage efficaces. De manière générale, la capacité des petites batteries d'un nœud de capteur est le facteur le plus restrictif et doit être soigneusement pris en compte [64]. Ainsi, un protocole de routage devrait choisir d'envoyer juste assez de mises à jour pour assurer la fraîcheur des routes construites tout en conservant un profil de faible puissance. Les mises à jour «justes» peuvent varier de la transmission d'une mise à jour toutes les secondes à une transmission groupée toutes les quelques minutes, selon les conditions actuelles du réseau et le type d'application afin de s'assurer que le budget énergétique de l'application est respecté [57].
- 5. Liens non fiables :**Les LLN sont caractérisés par des liaisons avec perte et peu fiables, et une mise à jour n'est pas garantie d'atteindre sa destination dès sa première transmission[65][66] .Dans certains cas, le taux de perte de liaison dans un réseau ne peut pas être prévu à l'avance et, pire encore, la même liaison peut présenter différents taux de perte au fil du temps en raison de facteurs tels que les collisions au niveau du récepteur, le problème de terminal caché et les interférences avec les émetteurs radio des nœuds voisins Cependant, il existe encore des cas où un taux de perte a priori peut être grossièrement prévu en fonction, par exemple, des statistiques des déploiements précédents. Par conséquent, un routage le protocole devrait avoir la capacité de fonctionner efficacement dans des conditions aussi peu fiables.

## 2.9 Solution des protocoles de routage

Comme nous avons mentionné dans ce chapitre et le chapitre précédent IoT et les réseaux LLNs confrontent beaucoup de défis, et pour cette raison plusieurs protocoles de routages ont été conçu pour répondre à ces exigences.

- ✓ RPL supporte la mobilité en permettant à un nœud mobile de se connecter comme feuille à n'importe quel nœud appartenant à la topologie de routage, il est également fiable car il s'appuie sur l'algorithme Trickle pour maintenir efficacement la topologie de routage, permettant une réaction rapide aux changements de connectivité tout en minimisant la surcharge dans des conditions stables. RPL a la capacité de fonctionner efficacement dans des conditions peu fiables, car Les LLN sont caractérisés par des liaisons avec perte et peu fiables, et une mise à jour n'est pas garantie d'atteindre sa destination dès sa première transmission, il est utilisé pour répondre aux exigences des LLN telles que l'équilibrage de charge, la consommation d'énergie déséquilibrée, le trafic réseau et la mémoire[2].
- ✓ LOADng-IoT améliore la qualité de service et la fiabilité du réseau en augmentant le taux de livraison des paquets et en réduisant la latence de bout en bout pour les différents types de messages échangés par les nœuds dans les scénarios IoT denses et dispersés. Il réduit le nombre de messages de contrôle requis pour construire des routes entre les nœuds, contribuant à un réseau plus efficace avec une surcharge réduite, il réduit la quantité d'énergie requise pour construire des chemins et acheminer des messages de données, ce qui rend le réseau plus économe en énergie. Il dispense de l'utilisation de passerelles Internet prédéfinies car les nœuds connectés à Internet sont recherchés à la demande et peuvent évoluer en fonction de leur disponibilité de connexion. Cette fonctionnalité supprime également l'existence d'un point de défaillance unique (SPOF) pour la connexion des appareils IoT avec des services Internet externes. Il présente une solution flexible, grâce à laquelle des parties de la proposition peuvent être adoptées en fonction des capacités matérielles des nœuds [48].



## **2.10 Indépendance entre la couche MAC et la couche RÉSEAU**

Pour assurer l'interconnexion entre les éléments d'un réseau, les couches qui compose la pile de protocoles a été proposée pour modéliser le découpage fonctionnel des réseaux de capteurs sans fil, notamment qu'une couche correspond à un ensemble de fonctions ou de processus cohérents entre eux et assurant des fonctionnalités précises. Ces fonctionnalités sont indépendantes c'est à dire que chaque couche fonctionne indépendamment de l'autre.

### **2.10.1 Limites de l'indépendance entre la couche MAC et la couche RÉSEAU**

Bien que l'indépendance des couches soit très répandue dans le domaine des réseaux filaires grâce à sa modularité, le principe de séparation n'est pas très efficace pour les réseaux de capteurs sans fil.

En effet, cette indépendance ne garantit pas des performances optimales pour les réseaux LLNs car ces derniers présentent certaines caractéristiques différentes des autres types de réseaux telles que les ressources limitées (énergie, mémoire, calcul) et la présence d'interférences radio. Ces caractéristiques génèrent de nouvelles contraintes lors de la conception des protocoles dans le modèle en couches pour les réseaux de capteurs sans fil qui doivent être adaptatifs.

Parmi les limites du modèle en couches, nous citons [67]:

- Les frontières entre les couches sont strictes.
- Chaque couche doit assurer des fonctionnalités très particulières et ne peut ajouter des fonctionnalités d'autres couches.
- Combiner des protocoles de couches différentes et les utiliser ensemble ne garantit pas de meilleures performances du réseau.
- Les décisions sur le transfert de données d'un nœud vers un autre sont prises indépendamment dans les couches, ces décisions ne sont pas corrélées dans une approche en couches, ce qui ne permet pas une optimisation simultanée de la même métrique.

- Les trois plans transversaux de gestion permettent une certaine communication entre les couches ; par exemple, les protocoles de routage qui s'appuient sur l'énergie extraient cette information depuis la couche physique. Mais cette communication reste très limitée et ne permet pas de surmonter les limites de chaque couche.

## **2.11 Conclusion**

Dans ce chapitre nous avons introduit la norme IEEE 802.15.4 ainsi que ses limitations par la suite nous avons parlé sur son amendement et les différentes améliorations apportés .Nous avons également discuté les protocoles de routage proactifs et réactifs existants tels que RPL et LOADng et pour finir nous avons cité les limites de l'indépendance entre la couche liaison de données et la couche réseau.

Dans le prochain chapitre nous allons présenter les travaux connexes existants sur Cross-Layer.

# Chapitre 03 : Etat de l'art sur Cross –Layer

## 3.1 Introduction

La plupart des recherches menées dans le cadre de l'étude et la conception de protocoles d'amélioration du délai de bout en bout dans les LLNs convergent, la plupart du temps, vers des solutions de routage souvent basées sur la couche réseau sans pour autant associer d'autres couches (notamment les couches basses), cependant ces solutions ne garantissent pas des performances optimales pour les réseaux IoT car ces derniers présentent certaines caractéristiques différentes des autres types de réseaux telles que les ressources limitées (énergie, mémoire, calcul) et la présence d'interférences radio. C'est dans ce contexte que le Cross-Layer est apparu comme un nouveau domaine de recherche. Son principe de base est de rendre l'information disponible à tous les niveaux de la pile des protocoles. C'est-à-dire, qu'il permet la définition de protocoles ou de mécanismes qui ne respectent pas l'isolation des couches du modèle OSI. Ainsi, il autorise la communication entre deux, ou plusieurs, couches adjacentes, ou non adjacentes, dans le but d'améliorer les performances globales du système tout en minimisant la dépense d'énergie.

Ce chapitre est consacré pour se familiariser avec le Cross-Layer en détail en présentant les différentes recherches menées dans ce contexte.

## 3.2 La communication dans les architectures Cross-Layer

Le principe de base du concept Cross-Layer est de permettre l'échange d'informations entre les couches adjacentes et non adjacentes afin d'améliorer les performances de transmission. Cet échange d'informations peut être mis en œuvre suivant différents schémas. Parmi toutes les architectures Cross-Layer proposées dans la littérature, trois modèles de communication peuvent être distingués [68] : La communication directe entre les couches, une base de données partagée entre les couches et tube de signalisation inter couche.

### **a) Communication directe entre les couches**

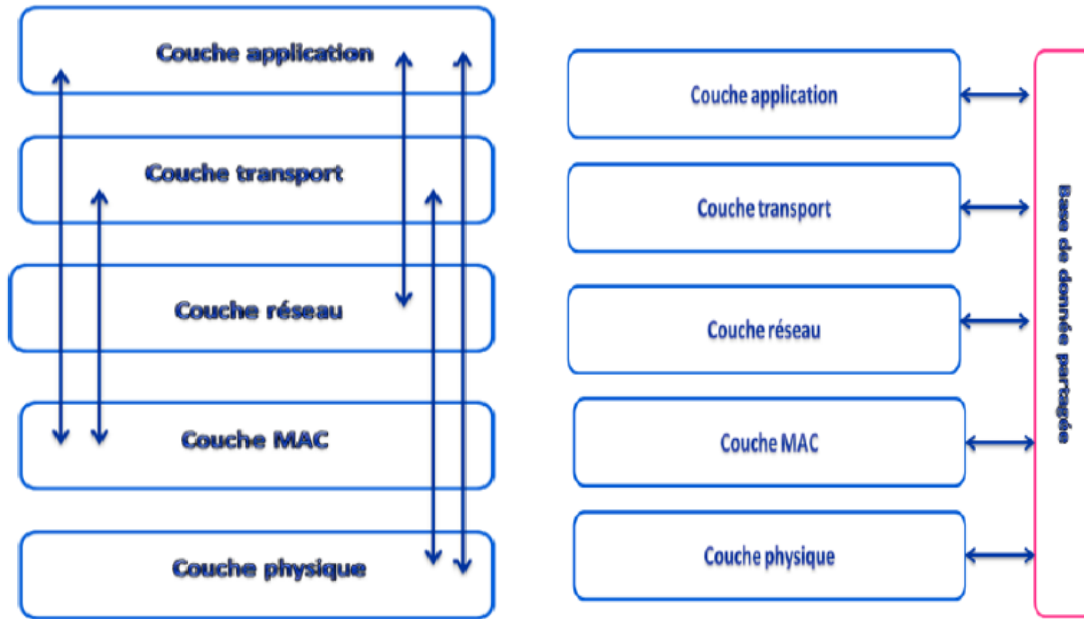
La communication directe entre les couches est le modèle le plus utilisé par les architectures Cross Layer. Il permet à une couche d'accéder directement aux paramètres et aux variables d'une autre couche sans passer par un intermédiaire.

### **b) Communication via une base de données partagée**

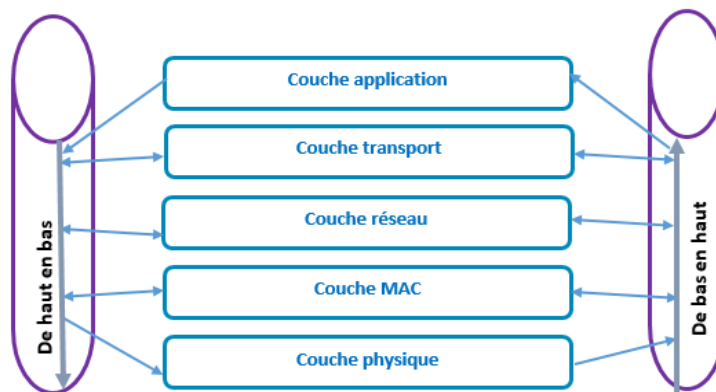
Plusieurs architectures Cross Layer [68] [61] proposent l'utilisation d'une base de données partagée afin de stocker et de récupérer des paramètres. Celle-ci est accessible par toutes les couches qui peuvent, ainsi, s'informer de l'état des autres couches ou récupérer des paramètres de configuration nécessaire à leur fonctionnement interne. La base de données est considérée comme une nouvelle couche en parallèle à toutes les autres.

### **c) Tube de signalisation inter-couche**

L'une des approches utilisées pour la mise en œuvre de la signalisation inter-couche est révélée par [69] un canal de signalisation inter-couche, qui permet la propagation de messages de signalisation couche à couche avec le flux de données par paquets à l'intérieur de la pile de protocoles de manière ascendante ou descendante. Une propriété importante de cette méthode de signalisation est que les informations de signalisation se propagent avec le flux de données à l'intérieur de la pile de protocoles et peuvent être associées à un paquet particulier entrant ou sortant de la pile de protocoles.



a) Communication directe entre les couches    b) Communication via une base de donnée partagée



c) Tube de signalisation inter-couche

**Figure 3. 1** La communication dans les architectures Cross-Layer [61]

### **3.3 Les approches du Cross Layer dans les réseaux sans fil**

Dans la littérature, plusieurs techniques Cross Layer ont été proposées pour améliorer les performances des transmissions sans fil. Au début, ces mécanismes étaient limités à l'interaction entre la couche physique et la couche liaison de données. De plus, les mécanismes proposés étaient indépendants et visaient l'amélioration d'une imperfection précise. Par la suite, nous avons assisté à l'apparition de plusieurs travaux proposant des interactions avec les couches supérieures et aussi à l'apparition d'architectures Cross Layer qui tentent de faire collaborer plusieurs couches, prenant en charge plusieurs paramètres, pour une optimisation globale. Ainsi, les auteurs dans [61], ont classifié les approches Cross Layer en quatre classes :

#### **a) L'approche ascendante**

Les couches supérieures optimisent leurs mécanismes en fonction des paramètres (conditions) des couches inférieures.

#### **b) L'approche descendante**

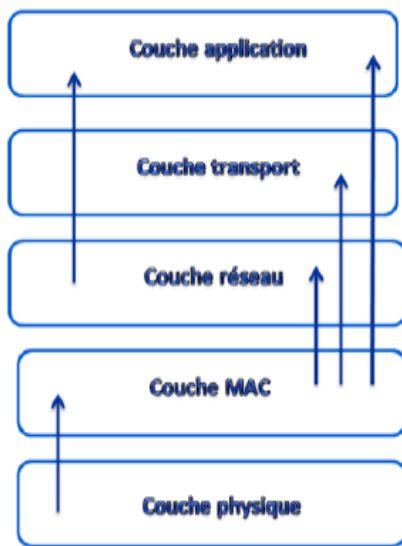
Les couches supérieures décident des paramètres de configuration des couches inférieures. Ou bien, les couches inférieures considèrent certaines spécificités du niveau applicatif pour exécuter leurs traitements.

#### **c) L'approche mixte**

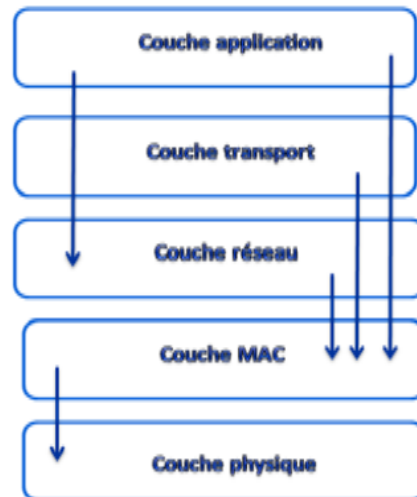
Cette approche exploite les deux approches précédentes dans une même architecture afin de trouver la meilleure configuration Cross Layer pour un fonctionnement optimal du système.

#### **d) Approche fusion de couches adjacentes**

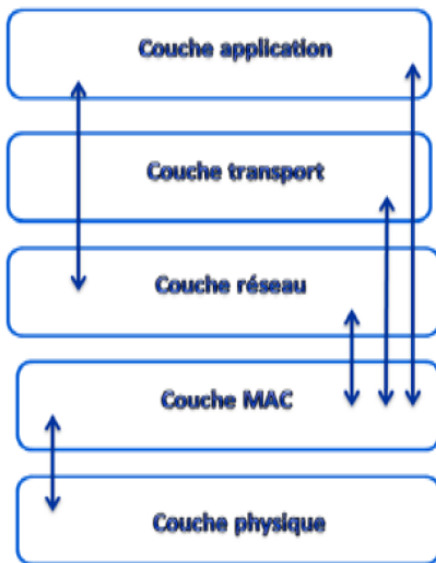
Ces approches combinent des informations et des fonctionnalités de protocoles de couches différentes en une seule nouvelle couche.



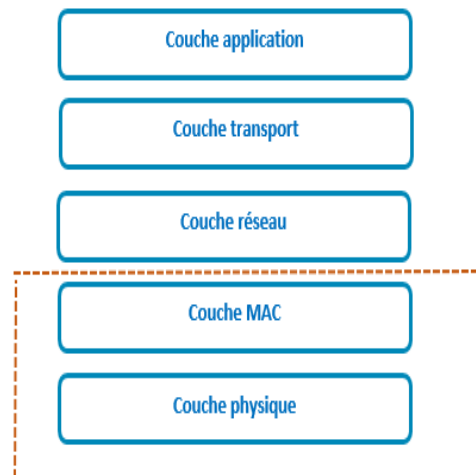
a) L'approche ascendante



b) L'approche descendante



c) L'approche mixte



d) Approche fusion de couches adjacentes

**Figure 3. 2** Les approches du Cross Layer dans les réseaux sans fil.

### 3.4 Type de liaison Cross Layer [70]

Dans cette partie, on s'intéresse au Cross Layer entre deux couches.

### **3.4.1 Interactions entre la couche Physique et Transport**

Les solutions se basent sur l'optimisation de la puissance de l'émission d'un nœud capteur et le contrôle de la congestion, c'est-à-dire trouver une puissance d'émission de tel sorte qu'il ne créera pas d'interférences.

### **3.4.2 Interactions entre la couche Réseau et Physique**

Les protocoles de routage sont également touchés par la transmission à cause de la couche PHY, car l'augmentation de la puissance de transmission peut améliorer la capacité d'un lien, mais peut aussi dégrader la capacité de l'ensemble du réseau en raison de l'augmentation des interférences. Pour cela, la plupart des solutions se basent sur la qualité du canal (lien de routage) pour assurer le bon acheminement en diminuant le temps de latence et le nombre de saut et bien sûr la consommation d'énergie.

### **3.4.3 Interactions entre la couche Physique et Mac**

La majorité des solutions Cross Layer MAC-PHY se basent sur les protocoles mac.

En plus des caractéristiques du canal sans fil et la radio, les circuits, le contenu de l'information qui sera envoyé par le nœud capteur est également important dans la conception de la couche mac. Le contenu de ces informations est étroitement lié à des propriétés physiques. Par exemple, la qualité de liaison fournie par la couche mac, dépend de la puissance de transmission.

### **3.4.4 Interactions entre la couche Réseau et Mac**

Dans cette approche, le prochain saut est choisi par rapport au voisinage, car le trafic est périodique dans les RCSFs, c'est-à-dire que les nœuds sont en état actif-inactif.

D'autre part, les retards de paquets et/ou la faiblesse de la bande passante peut se produire, mais en forçant la couche réseau à modifier son itinéraire de décisions, ces derniers problèmes seront réglés. La modification des décisions de routage prévues influence les performances de la couche mac.

### **3.4.5 Interactions entre la couche Réseau et Application**

L'un des rôles de la couche application est de définir le délai de tolérance, c'est-à-dire le RCSF ne tolère aucun retard de transmission ou un retard qui ne dépasse pas le temps



prédéfini. Pour cela, l'interaction entre la couche Application et Réseau peut gérer ce problème en définissant la tolérance de retard à la couche Réseau qui peut agir en conséquence, gardant en vue par exemple la capacité de batterie (pile) restante ou un autre paramètre potentiel.

### **3.5 Cross-Layering entre la couche Mac et la couche Réseau**

L'ITU (International Telecommunication Union) considère que le cross-layering consiste à permettre des interactions entre différentes couches (incluant les couches non adjacentes) dans le but d'atteindre des gains dans les performances du réseau.

Comme notre objectif principal est de minimiser la latence des communications dans les réseaux de capteurs sans fil en adoptant des approches Cross-Layer, nous nous intéressons aux approches Cross-Layer concernant la couche Réseau et la couche Liaison de données car leurs décisions ont un impact important sur la latence des communications.

Cette approche assure des communications entre le protocole de routage et le protocole mac en connectant la sortie d'un des deux protocoles en entrée au deuxième protocole. Par exemple, le protocole de routage fournit son résultat (la structure de routage) au protocole mac pour qu'il l'utilise afin d'améliorer ses décisions temporelles ou vice versa.

### **3.6 Classification de Cross-Layering entre la couche Mac et la couche Réseau**

Comme nous l'avons mentionné ci-dessus un nombre important d'approches Cross-Layer existe dans la littérature. Nous nous intéressons dans notre étude aux approches traitant la couche Liaison de données (plus particulièrement la sous-couche Mac) et la couche Réseau, qui comporte deux catégories :

- a.** les protocoles de routage utilisant des informations du protocole mac,
- b.** les protocoles mac utilisant des informations du protocole de routage.

#### **3.6.1 Approches de routage utilisant des informations du protocole mac (MAC-AWARE routing)**

Dans cette catégorie, le protocole mac, une fois déterminé, doit envoyer des informations temporelles au protocole de routage pour que ce dernier puisse trouver le prochain saut pour chaque nœud.

### 3.6.1.1 ORW (OPPORTUNISTIC ROUTING FOR WIRELESS SENSOR NETWORKS)

ORW [71] est un protocole de routage basé sur des informations du protocole mac et qui vise à diminuer le délai et la consommation d'énergie en utilisant tous les voisins comme nœuds relais potentiels. Il propose un routage opportuniste qui se base sur un protocole mac utilisant le concept de cycles sommeil/réveil pour chaque nœud. Avec ce type de protocole mac, les données dans ORW sont envoyées au premier voisin réveillé.

Les concepts de ORW étant génériques, le type du protocole mac utilisé n'est pas essentiel (avec ou sans contention) ; la transmission se fait toujours vers le premier voisin qui se réveille et qui assure un rapprochement vers le sink.

#### Fonctionnement :

— ORW utilise une métrique de routage appelée EDC (ExpectedDuty Cycle) qui reflète, pour un nœud donné  $i$ , le nombre de réveils prévus jusqu'à ce qu'un paquet arrive à la destination.

$$EDC_i (S_i) = \frac{1}{\sum_{j \in S_i} P_{ij}} + \frac{\sum_{j \in S_i} P_{ij} EDC_j}{\sum_{j \in S_i} P_{ij}} + w \quad (3.1)$$

Où :

$S_i$  est un sous ensemble des voisins de  $i$ ,  $P_{ij}$  est la qualité de lien entre les deux nœuds voisins  $i$  et  $j$  qui reflète la probabilité du succès de la transmission des données, et  $w$  est une constante.

La métrique EDC est calculée de manière itérative à partir du sink et jusqu'à ce qu'elle devienne stable, par des échanges de paquets dans le réseau.

Quand un nœud a des données à transmettre au sink, il choisit comme prochain saut le premier voisin qui se réveille et dont l'EDC est inférieur au sien. La procédure est répétée chez chaque nœud, jusqu'à ce que les données arrivent au sink. La métrique EDC permet de garantir la transmission des données dans la direction du sink, puisque son calcul commence

au sink et se propage dans le réseau tout en s'incrémentant, d'où le choix d'un voisin avec un EDC inférieur à celui de l'émetteur.

ORW est un protocole Cross-Layer car, pour chaque émetteur, il choisit comme prochain saut le voisin qui se réveille le premier et cette information est extraite du protocole mac.

### **3.6.2 Approches mac utilisant des informations du protocole de routage (routing-aware mac)**

Dans cette catégorie, la couche Liaison de données, et plus particulièrement le protocole mac, attend des informations du protocole de routage afin d'établir un ordonnancement des communications pour les nœuds du réseau.

Comme pour les protocoles mac classiques, les approches mac Cross-Layer présentent deux formes : les approches basées sur la contention et les approches sans contention.

Les approches mac Cross-Layer basées sur la contention, trouvées dans la littérature, sont RMAC [72], CL-MAC [73] et AreaCast [74].

#### **3.6.2.1 RMAC (routing enhanced mac protocol)**

RMAC [72] est aussi un protocole mac cross-layer avec contention. Il vise à réduire la latence de communication de bout en bout, et ce, en utilisant un seul paquet de contrôle appelé Pion et qui joue le rôle de RTS et CTS en même temps. En plus des champs trouvés dans un RTS classique (adresse de l'émetteur, adresse du prochain saut, durée de la transmission, ....), le Pion contient l'adresse de la destination finale et le nombre de sauts effectués.

#### **Fonctionnement :**

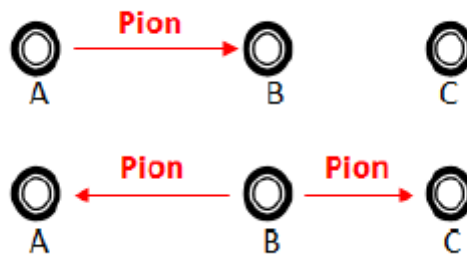
Dans RMAC, deux périodes sont utilisées, la période DATA et la période SLEEP. La première sert à trouver les nœuds qui participeront dans la transmission des données. Dans la deuxième période se fait la transmission effective des données.

##### **1. Période DATA :**

Quand un nœud A a des données à envoyer, il choisit une période aléatoire et attend que le canal de communication soit libre. Dès que le canal est libre, le nœud A envoie un Pion à son prochain saut (par exemple le nœud B) trouvé dans sa table de routage.

Le nœud B vérifie s'il est la destination finale. S'il ne l'est pas, il cherche dans sa table de routage le prochain saut qui permet d'atteindre la destination finale, le nœud B attend une

période (SIFS) avant de transmettre son Pion. Le nouveau Pion contient : l'adresse du saut précédent A, l'adresse de l'émetteur B, l'adresse du prochain saut C et l'adresse de la destination finale, le compteur de sauts est incrémenté dans ce Pion, ce dernier servira aussi comme CTS pour le nœud A et comme RTS pour le nœud C. Le nœud A reçoit le Pion du nœud B et attend que la période SLEEP commence pour démarrer la transmission, le nœud C reçoit le Pion du nœud B et refait la même procédure. La procédure continue jusqu'à ce qu'un Pion arrive à la destination finale, ou la période de transmission de données se termine.



**Figure 3. 3** Période DATA de RMAC

**2. Période SLEEP :** Après la fin de la période DATA qui détermine les nœuds participants dans l'acheminement des données, la période SLEEP commence et s'effectue comme suit :

Quand la période SLEEP commence, le nœud A envoie les données au nœud B par la suite le nœud B renvoie un accusé de réception ACK à l'émetteur A, dès réception de l'accusé ACK, le nœud A passe à l'état de sommeil. Le nœud B transmet les données au nœud C, attend l'accusé ACK puis passe à l'état sommeil. La procédure continue jusqu'à ce que les données arrivent à la destination finale. Pendant la transmission entre les nœuds A et B, les autres nœuds (comme le nœud C) peuvent s'endormir et se réveiller à la fin de cette transmission. La durée de sommeil est égale à la somme de la durée de transmission + SIFS + durée de l'ACK.

À chaque fois qu'un nœud a des données à transmettre à la station de base, les deux périodes, DATA et SLEEP, sont utilisées successivement.

### 3.6.2.2 CL-MAC (CROSS-LAYER MAC)

CL-MAC [73] propose de gérer efficacement les trafics multi-flux et de réduire la latence des communications. La communication dans CL-MAC est faite à l'aide du paquet de contrôle FSP (Flow Setup Packet) contenant l'adresse de l'émetteur, les adresses des destinations finales et les adresses des prochains sauts trouvés dans les tables de routage.

Le FSP sert comme un message RTS pour le nœud récepteur et en même temps comme un message CTS pour le nœud émetteur de données. Il a aussi la particularité de pouvoir s'adresser à plusieurs destinataires en même temps.

#### **Fonctionnement :**

Un nœud A prépare le FSP<sub>i</sub> en incluant les adresses des prochains sauts (de sa table de routage) et les adresses des destinations finales, *i* étant le nombre de destinations finales. L'ordre dans lequel les destinations finales apparaissent dans le FSP<sub>i</sub> est important car il reflète la priorité dans l'ordonnancement. Selon sa position dans le FSP<sub>i</sub>, le voisin qui reçoit le FSP saura s'il participe à la communication ou non ; dans l'affirmative, il connaîtra la durée pour laquelle il peut s'endormir. Après la lecture du FSP<sub>i</sub>, le voisin cherche ses prochains sauts (dans sa table de routage), met à jour le FSP<sub>i</sub> et le diffuse. À la fin de la transmission des FSPs, chaque nœud connaît l'émetteur des données et le moment d'envoi. La transmission des données commence.

### 3.6.2.3 AREACAST

AreaCast [74] vise à améliorer la robustesse des protocoles de routage et le taux de livraison des paquets, tout en assurant une bonne consommation énergétique. Bien qu'il fonctionne de la même façon que les autres approches distribuées de sa catégorie, il corrige en plus le problème de nœuds défaillants i.e. dans le cas où le prochain saut est défaillant, un autre nœud le remplacera.

#### **Fonctionnement :**

Les nœuds s'échangent des messages Hello entre eux pour connaître leurs voisinages, par la suite l'étape de préparation enchainera :

Un nœud émetteur diffuse un message RTS contenant l'adresse du prochain saut. Les voisins de ce nœud reçoivent le message RTS et chaque voisin vérifie, s'il est le prochain saut, alors il devient Relai Explicite, sinon, il applique un algorithme d'auto-élection qui permet d'élire les nœuds qui sont voisins à la fois avec l'émetteur, avec le Relai Explicite et avec le prochain saut. Si les conditions sont vérifiées, il devient Relai Implicite, sinon, il s'endort le temps de la transmission. Le Relai Explicite renvoie un message CTS ; s'il ne le fait pas, le premier Relai Implicite le remplace et ainsi de suite.

Transmission de données : dès la réception du CTS, l'émetteur commence la transmission des données, après réception des données, le Relai renvoie un accusé de réception ACK à l'émetteur et refait la même procédure.

Le tableau suivant résume les caractéristiques des approches cross-layer existantes ainsi que leurs inconvénients.

**Tableau 3.1** Tableau résumant les caractéristiques des approches cross-layer existantes ainsi que leurs inconvénients

Approche	Dist/cent	MAC	ROUTAGE	Objectifs	Limites
<b>ORW</b> [71]	Distribuée	Cycle sommeil/ Réveil	Opportuniste	Latence Energie	processus de calcul itératif / vue globale du réseau
<b>RMAC</b> [72]	Distribuée	Avec contention	quelconque	Latence	cycle sommeil/réveil seulement pour les voisins à un saut et absence des transmissions simultanées
<b>CL-MAC</b> [73]	Distribuée	Avec contention	quelconque	Latence	
<b>AreaCast</b> [74]	Distribuée	Avec contention	quelconque	énergie, taux de livraison des paquets	non orientée latence

### **3.7 Conclusion**

Dans ce chapitre nous avons bien détaillé le Cross-Layering en présentant ses types, les différentes communications existantes et ses approches, puis nous avons discuté également ses travaux connexes pour avoir une image plus claire afin de l'utiliser dans notre étude pour proposer et implémenter une nouvelle approche Cross-Layer pour atteindre notre but.

Dans le chapitre qui suit nous nous intéressons à la proposition et l'implémentation de l'approche QL-ETX comme une nouvelle fonction objective du protocole RPL.

# Chapitre 04 : QL-ETX, une nouvelle approche Cross-Layer pour les LLNs

## 4.1 Introduction

Le modèle en couches, inspiré du modèle OSI utilisé pour assurer les communications dans les réseaux de capteurs sans fil, fonctionne sur le principe de séparation des couches. Bien que ce principe soit efficace pour les réseaux filaires, il n'est pas une solution optimale pour les réseaux IoT (LLNs). Par conséquent, l'exploitation de nouvelles techniques et approches dans le modèle en couches s'est avérée nécessaire. Dans cette optique nous avons tenté de concevoir une nouvelle approche Cross-Layer.

Une panoplie d'approches Cross-Layer entre les différents protocoles de la couche mac et la couche réseau existe, à l'opposé aux approches basée sur TSCH et RPL qui sont inexistantes, puisque le TSCH est un nouveau mode issu de l'amendement de la norme IEEE802.15.4, alors il attire l'attention des chercheurs pour contribuer à son développement.

Le routage dans l'IoT confronte d'énormes défis comme par exemple l'énergie, la mobilité, le délai, la sécurité ,etc. Pour ces raisons, nous avons constaté qu'il est nécessaire de créer une Cross-Layer basée sur la communication entre TSCH et RPL en apportant des modifications aux deux couches (mac, réseau) pour le but de diminuer le délai de bout en bout.

Plusieurs Cross-Layer ont été proposé mais chacunes ont leurs limites; comme AreaCast (k,heurtefeux et al) et CoLaNet (C. Chou and K. Chuang) leurs objectifs sont l'énergie et taux de livraison de paquets(PDR) mais ne sont pas orienté latence , ORW (L.Louail et al) est orienté latence mais il est basé sur un processus de calcule itératif /vue globale du réseau.

Dans les fonctions objectives existantes telles MRHOF qui est basée seulement sur la qualité du lien ,OF0 qui est basée seulement sur le rang (nombre de sauts ) et le problème de collision dû à la charge du réseau ,le délai augmente, c'est pourquoi nous avons créé une nouvelle fonction objective QL-ETX qui prend en paramètres deux métriques ,ETX et le nombre des



paquets en files d'attente ,ce nombre est récupéré à partir de la couche liaison de données ce qui introduit une communication entre les deux couches .

Ce chapitre est dédié à la conception et l'implémentation de la nouvelle approche proposée.

## **4.2 L'approche Cross-LayerTSCH-RPL**

Pour le but de minimiser le délai de bout en bout nous avons étudié en détails les deux couches, la couche mac (TSCH) et la couche réseau (RPL) afin de trouver une solution Cross-Layer qui permet l'interaction entre ces dernières.

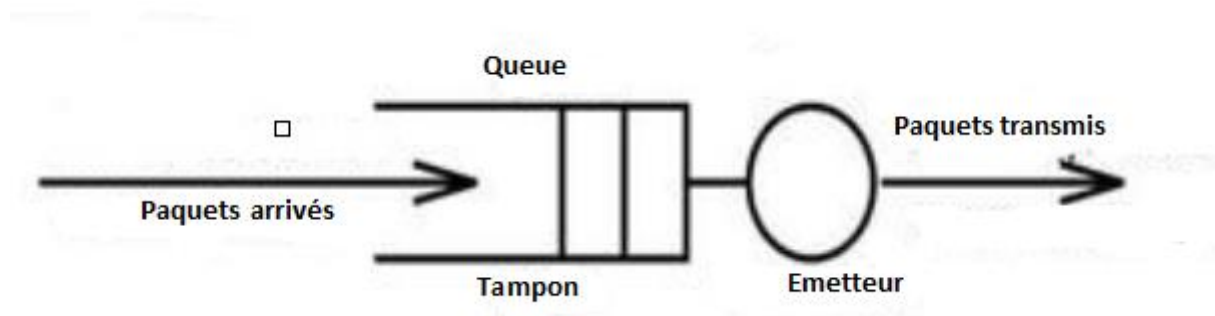
### **4.2.1 Délai de bout en bout**

Le délai de bout en bout fait référence au temps nécessaire pour qu'un paquet soit transmis sur un réseau de la source à la destination. C'est un terme courant dans la surveillance de réseau IP, et diffère du temps d'aller-retour en ce que seul le chemin dans la direction de la source à la destination est mesuré. On trouve plusieurs causes des délais de bout en bout, par exemple :

- Délais de traitement
- Délais liés à la mise en tampon
- Délais de transmission
- Délais de propagation

Dans notre étude nous nous intéressons aux délais liés à la mise en tampon qui est le temps pendant lequel un paquet attend dans une file d'attente jusqu'à ce qu'il puisse être transmis. Ce terme est le plus souvent utilisé en référence aux routeurs. Lorsque des paquets arrivent sur un routeur, ils doivent être traités et transmis. Un routeur ne peut traiter qu'un seul paquet à la fois. Si les paquets arrivent plus rapidement que le routeur ne peut les traiter (comme dans une transmission en rafale), le routeur les place dans la file d'attente (également appelée tampon) jusqu'à ce qu'il puisse les transmettre. Le délai peut également varier d'un paquet à l'autre, de sorte que des moyennes et des statistiques sont généralement générées lors de la mesure et de l'évaluation du délai de mise en file d'attente.

Le délai de mise en file d'attente maximal est proportionnel à la taille de la mémoire tampon. Plus la ligne de paquets en attente de transmission est longue, plus le temps d'attente moyen est long comme le montre la figure 4.1.



**Figure 4. 1** Délai liés à la mise en file d’attente

Dans ce contexte nous avons essayé de réduire le temps d’attente dans les tampons, en se basant sur le protocole de routage RPL qui choisit le prochain saut qui a moins de paquets dans son tampon, pour cela nous nous sommes familiarisés avec le modèle du système de chaque protocole pour pouvoir proposer une nouvelle approche.

#### 4.2.2 Modèle du système du TSCH

Nous avons une file d'attente de transmission distincte pour chaque voisin, remplie par les couches supérieures et consommée par TSCH. De plus, il existe une file d'attente pour les paquets qui nécessitent un traitement post-transmission; c'est-à-dire des rappels de couche supérieure. Cette file d'attente est remplie par TSCH et consommée par les couches supérieures dans un contexte de processus. Enfin, TSCH dispose d'une file d'attente d'entrée, où il place toutes les trames de données entrantes valides, traitées ultérieurement par un processus de couche supérieure. La figure 4.2 illustre les modules qui lisent et écrivent dans les différentes files d'attente.

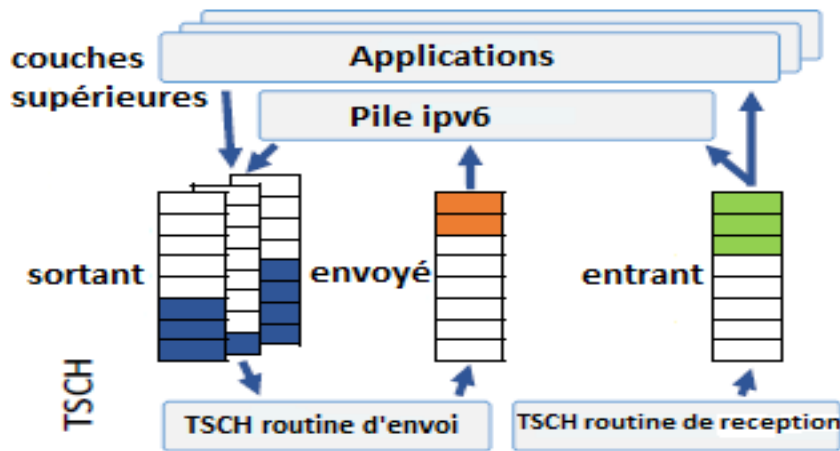


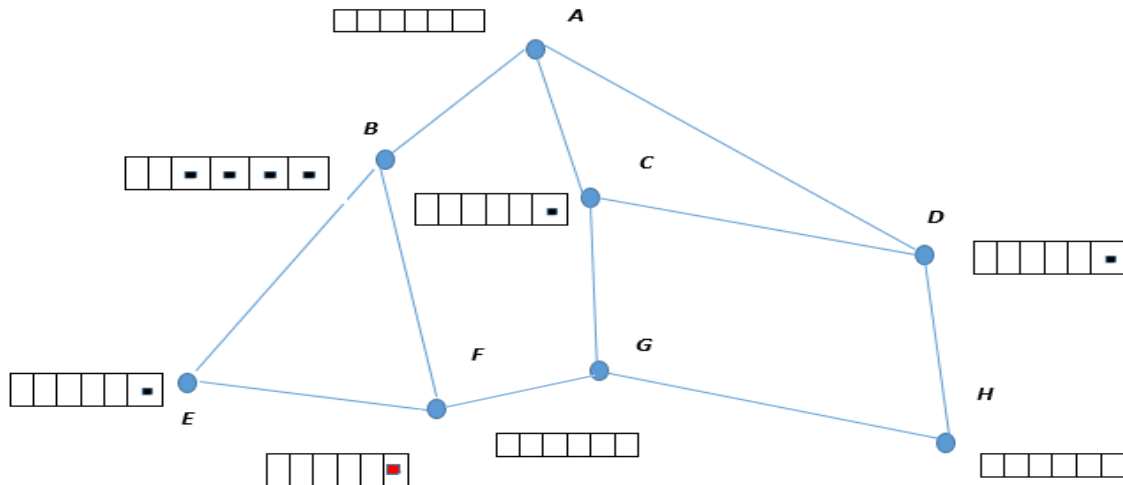
Figure 4. 2 Modèle du système de mise en file d'attente

### 4.2.3 L'approche QL-ETX

La plupart des solutions proposées pour le routage sont souvent basées sur la couche réseau sans associer d'autres couches, pour cela nous avons proposé une solution Cross-Layer qui permet la synchronisation entre le TSCH et le RPL, afin d'obtenir une meilleure performance du réseau.

#### 4.2.3.1 Motivation

La figure 4.3 présente un réseau de huit nœuds (de A à H). Pour chaque nœud nous avons une file d'attente de transmission avec le nombre de paquets qui sont en attente. Supposons que nous voulions envoyer des données du nœud F vers le nœud A.



**Figure 4. 3** Exemple d'un réseau de capteurs

Un protocole de routage qui utilise par exemple OF0 permet de trouver un chemin court (F-B-A) comportant 3 sauts malgré que la file d'attente du nœud père (B) soit surchargé. Ce type de routage n'est pas capable de trouver le chemin F-G-C-A comportant 4 sauts mais avec une latence inférieure (les files d'attentes de G et C sont presque vides). Ainsi, un chemin le plus court en termes de nombre de sauts ne garantit pas la meilleure latence. Ce comportement observé avec la métrique de nombre de sauts apparaît aussi avec la métrique de distance géographique, la même chose avec la métrique ETX si le protocole de routage qui utilise MRHOF choisit une route avec une meilleur ETX mais les files d'attentes soit surchargés alors la latence augmente.

Par conséquent, le protocole de routage doit profiter des informations temporelles venant des files d'attentes afin de minimiser la latence des communications.

#### 4.2.3.2 Les étapes de l'approche QL-ETX

L'approche QL-ETX passe par les 4 étapes suivantes :

##### 1. Phase de découverte du voisinage

Dans cette phase, les nœuds échangent des messages DIO et DAO pour connaître la topologie et découvrir les nœuds voisin, Jusque-là rien de différent, une nouvelle métrique 'nombre de

paquets’ est proposée. Cette métrique est échangée entre les nœuds voisins dans le contexte quand un nœud a plusieurs voisins candidats pour être le père, il récupère le nombre de paquets dans la file d’attente de chaque voisin, pour pouvoir calculer le cout.

## 2. Récupération de nombres des paquets et fonction objective

La récupération des nombres des paquets est faite à partir des files d’attente de transmission de chaque nœud, par une fonction qui se trouve dans la sous couche mac, ce nombre est échangé par les messages DIO entre les nœuds.

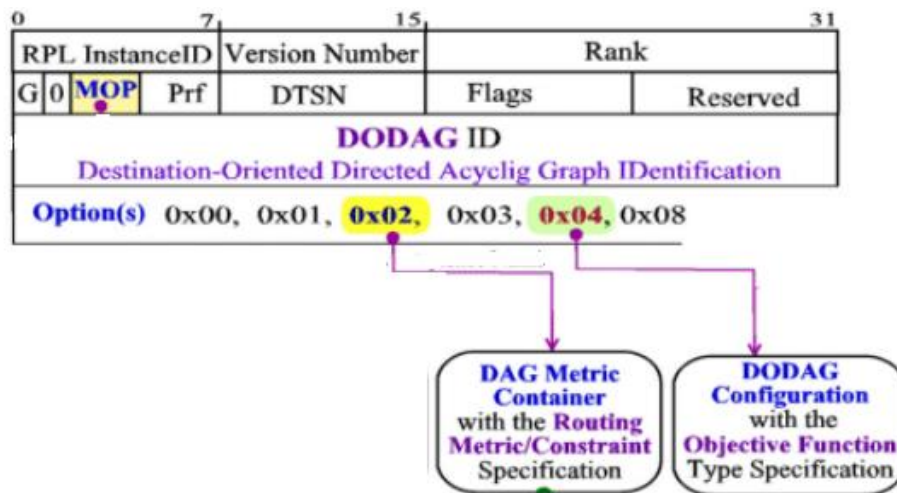


Figure 4. 4 la structure des messages DIO

Nous avons modifié la structure `rpl_metric container`, `metric_object` en ajoutant des nouveaux champs qui sont `QUEUE` et `cout` comme le montre le tableau ci-dessous.

Tableau 4.1 Les champs du conteneur métrique objet

Champ	Taille (bit)	Description
ETX	16	le nombre de retransmission
QUEUE	32	Le nombre des paquets dans la file d’attente
Cout	16	Combinaison d’ETX et QUEUE en utilisant la FO QL-ETX

### 3. Estimation du cout de QL-ETX

Après avoir récupéré le nombre de paquets du voisin  $i$  et calculer ETX, notre nouvelle fonction objective que nous avons introduite calcule le cout selon l'équation 4.1 suivante :

$$\text{Cout}_i = \alpha * \text{QUEUE}_i + (1-\alpha) * \text{ETX}_i \quad (4.1)$$

- $\alpha$ : détermine la pondération entre QUEUE et ETX
- $\text{QUEUE}_i$  : le nombre de paquets récupéré pour le voisin  $i$
- $\text{ETX}_i$  : le nombre de retransmission pour le voisin  $i$
- $\text{Cout}_i$  : résultat de combinaison pour le voisin  $i$

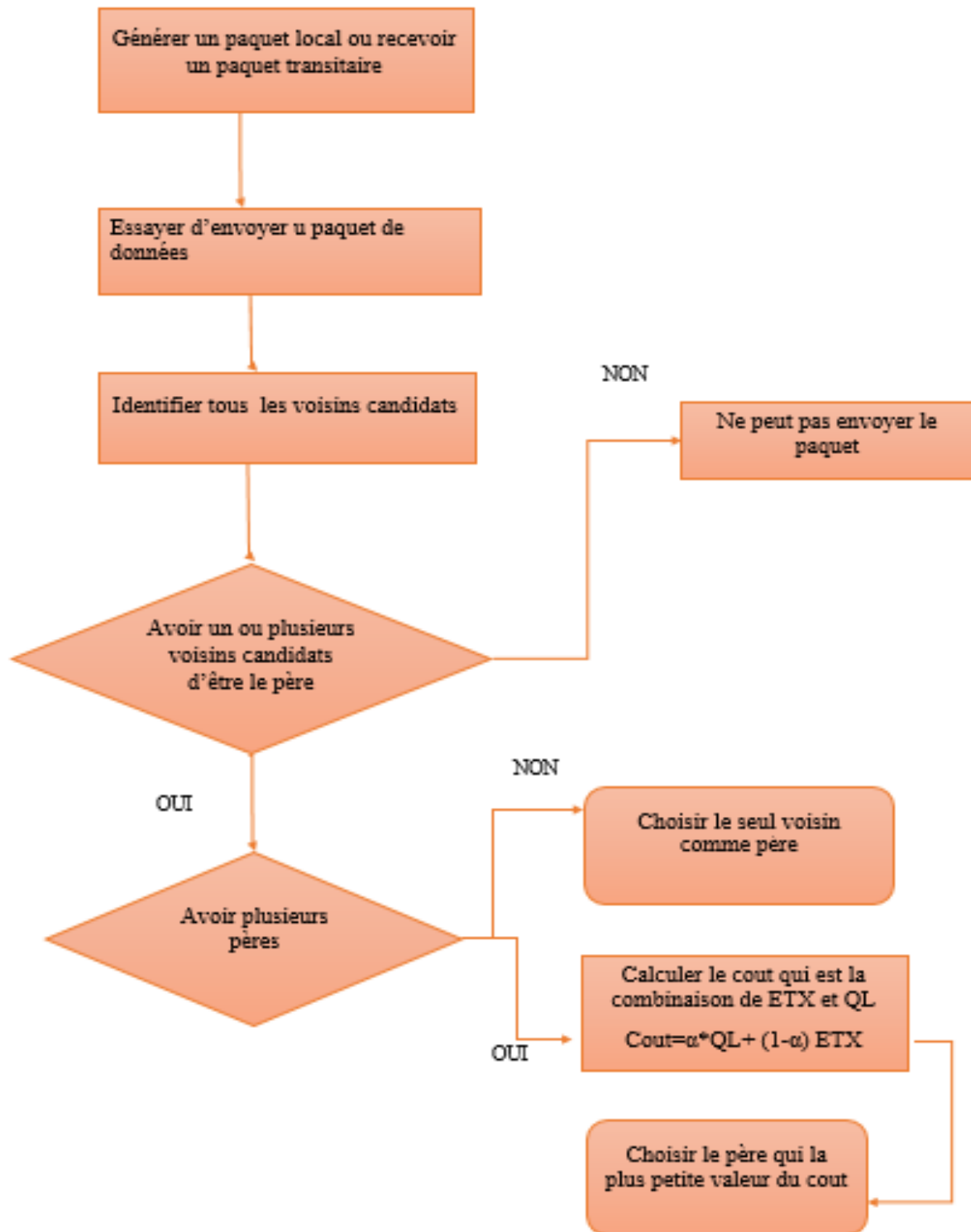
La valeur de  $\alpha$  qui varie de 0 à 1 aide à estimer la meilleure combinaison entre les deux paramètres pour avoir un coût optimal.

### 4. Choisir le parent

Contrairement aux autres fonctions objectives, notre proposition sélectionne à chaque fois le parent qui a le plus petit cout pour construire les routes vers le haut en utilisant deux métriques, une des deux est une métrique de la couche mac, par conséquent, notre proposition assure l'interaction entre les deux couches lors du routage des paquets.

Lorsqu'un nœud veut envoyer un paquet au SINK, il existe trois cas :

Si le nœud a un seul voisin, ce dernier est considéré comme son parent préféré et le paquet est transmis au SINK via ce parent. Si le nœud a plusieurs parents candidats, alors il choisit le voisin qui a le plus petit cout, sinon le paquet ne peut pas être envoyé. Cette procédure est faite par tous les nœuds pour créer la meilleure route entre un nœud  $i$  et le SINK.



**Figure 4. 5** La méthode de fonctionnement de l’approche QL-ETX dans la couche réseau du protocole RPL

Le pseudo code suivant est une explication d'organigramme qui présente la méthode de fonctionnement de l'approche QL-ETX dans la couche du protocole de routage RPL.

```

Nœud (i);
If (recevoir_paquet_transitaire(p) OU générer_paquet(p))
{
  Liste_v = Identification_des_voisins (); // une fonction que retourne une liste des voisins
  If (est_vide (liste_v))
  {
    Afficher (" erreur aucun voisin le paquet ne peut pas être acheminé ");
  }
  Else
  {
    If (longueur (liste_v) = 1)
    {
      Parent = liste_v [0]; // choisir le seule voisin comme père
    }
    Else
    {
      i=0;
      While (i < longueur (liste_v))
      {
        Cout [i] =  $\alpha$ *QL [liste_v[i]] + (1- $\alpha$ )* ETX[liste_v[i]]; // calculer le cout de chaque
                                                                    voisin
      }
      Parent = min (Cout[n]); //choisir le parent qui le plus petit cout
    }
  }
}

```

### 4.3 Implémentation de l'approche QL-ETX

Cette partie est consacré aux détails de l'implémentation de l'approche QL-ETX sous le simulateur Cooja et le système d'exploitation Contiki-ng qui nous ont permis de simuler notre approche.

#### 4.3.1 Aperçu sur le système d'exploitation Contiki OS

Contiki [75] est une plate-forme WSN qui fournit des logiciels et des matériels. Contiki a été créé par Adam Dunkels en 2002. C'est un système d'exploitation léger et open source conçu



spécifiquement pour les dispositifs IoT à faible consommation de ressources. Contiki dispose d'une pile de mise en réseau hautement optimisée comprenant plusieurs normes IoT telles que 6LoWPAN et IPv6.

### a) Architecture

Contiki est développé en langage C et il est constitué d'un noyau, de bibliothèques, d'un ordonnanceur et d'un jeu de processus. Comme tout système d'exploitation, son rôle est de gérer les ressources physiques telles que le processeur, la mémoire et les périphériques informatiques (d'entrées/sorties) comme montre la figure 4.5. Il fournit ensuite aux applications des interfaces permettant d'utiliser ces ressources. Conçu pour les modules de capteurs sans fil il occupe peu d'espace en mémoire et permet une consommation énergétique très faible.

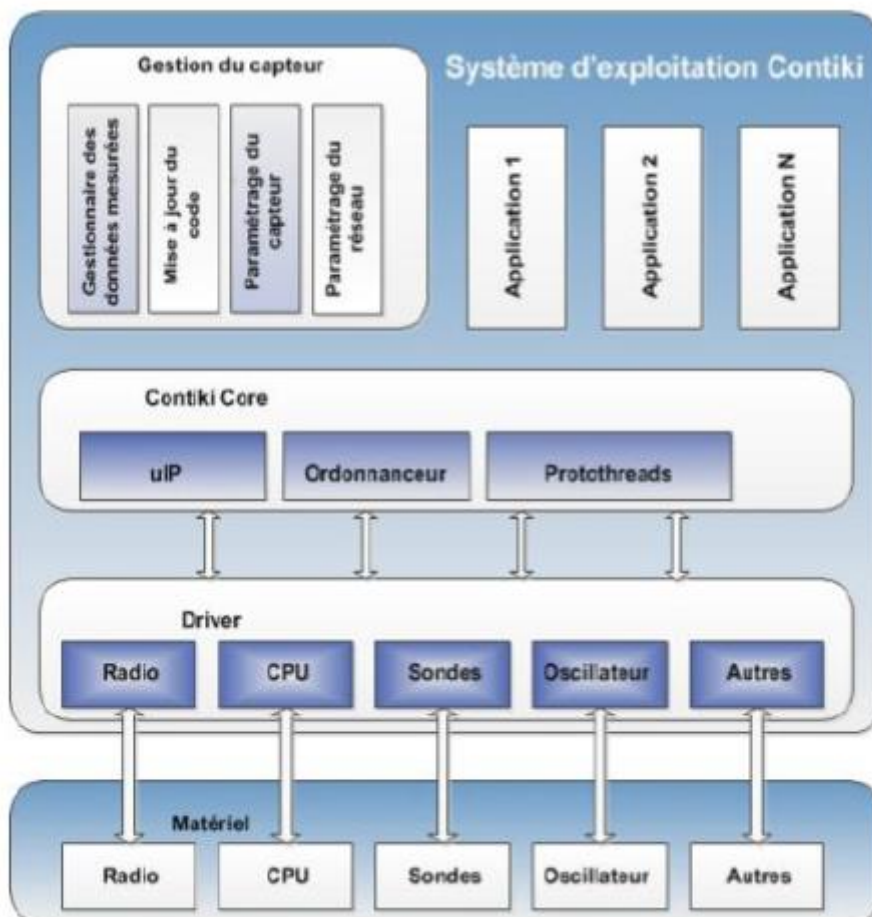


Figure 4. 6 Architecture de Contiki

## b) Connectivité

Contiki offre deux types de connectivité comme montre la figure 4.6 :

- **La couche Rime** : elle permet un dialogue avec les capteurs voisins ainsi que le routage.
- **La couche uIP**: orientée Internet, elle offre les services essentiels du protocole IP mais nécessite plus de ressources que Rime. Contiki gère les standards 6LoWPAN, RPL, CoAP.

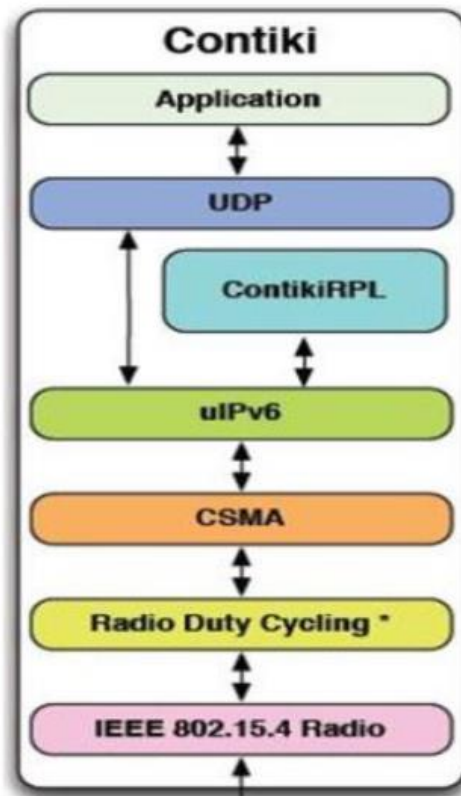


Figure 4. 7 Pile protocolaire de Contiki [75]

En 2017, les auteurs de Contiki ont lancé un nouveau fork du système d'exploitation Contiki nommé Contiki-ng

### 4.3.2 Aperçu sur le système d'exploitation Conitki-ng

Contiki-ng est un système d'exploitation multiplateforme open source pour les appareils IoT de nouvelle génération. Il se concentre sur les communications fiables (sécurisées) à faible consommation d'énergie et les protocoles standard, tels que IPv6 / 6LoWPAN, 6TiSCH, RPL et CoAP.

Contiki-ng est une nouvelle version du projet Contiki, il a été créé comme un fork du Contiki OS et conserve certaines de ses fonctionnalités d'origine, ce système propose deux implémentations différentes de RPL: RPL-classic et RPL-lite. RPL-classic a une taille de code de 227 Ko tandis que RPL-lite a une empreinte de code relativement plus petite de 204 Ko. La principale différence entre les deux implémentations est que RPL-lite n'implémente pas certaines fonctionnalités qui semblent inutiles selon l'analyse de RPL, comme le mode de stockage et l'existence de plusieurs instances (par exemple, une seule instance a été prise en charge qui utilise la métrique MRHOF et ETX). Cependant, toutes les implémentations de RPL basées sur Contiki n'incluent aucune de ses fonctionnalités de sécurité.

### 4.3.3 Simulateur Cooja

Contiki propose un simulateur de réseau appelé Cooja. Ce simulateur permet l'émulation de différents capteurs sur lesquels seront chargés un système d'exploitation et des applications. Cooja permet ensuite de simuler les connexions réseaux et d'interagir avec les capteurs. Cet outil permet aux développeurs de tester les applications à moindre coût.

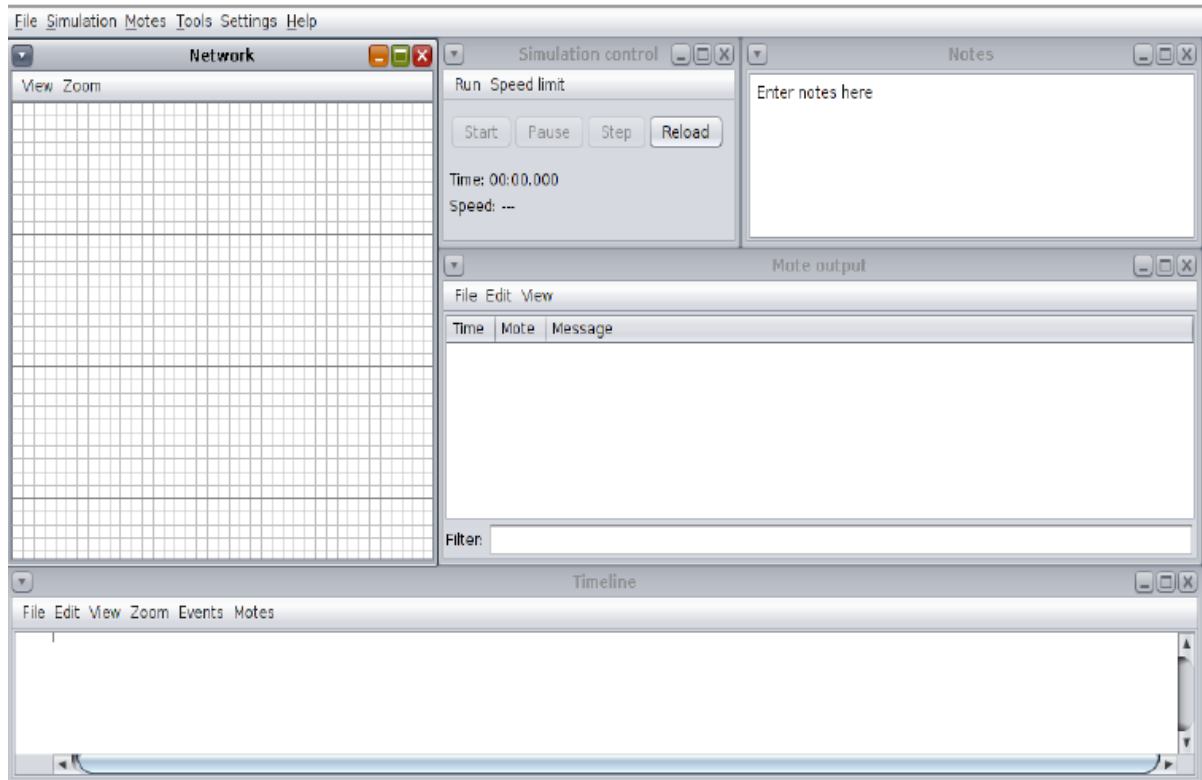


Figure 4. 8 L'interface de simulation

L'interface de simulation, montrée dans la Figure 4.7, se compose de cinq Fenêtres. La fenêtre Network montre la disposition physique du réseau, c'est-à-dire vous serez capables de placer physiquement des nœuds et les déplacer, selon le besoin, pour former la topologie et la disposition à laquelle vous vous intéressez.

La fenêtre Simulation Control vous permet de commencer, arrêter et recharger la simulation. Il vous permet aussi de contrôler le taux auquel la simulation procède.

La fenêtre Mote Output montre n'importe quelle production série produite par tous les nœuds, c'est-à-dire la production de la commande de printf. Vous pouvez filtrer la production montrée basée sur une chaîne que vous entrez dans le champ de "Filter".

La fenêtre Timeline, sert à la visualisation des communications radio (transmission, réception, collision) ainsi que les états réveillé et endormis des nœuds capteurs.

La fenêtre Note est l'endroit où nous pouvons mettre des notes pour notre simulation.

Cooja inclut d'autres fenêtres pas moins importantes et peut même être étendu pour en avoir plus.

#### 4.3.4 Détails de l'implémentation de QL-ETX

Afin d'améliorer le protocole de routage RPL nous avons proposé une nouvelle fonction objective, nommée QL-ETX, qui prend en paramètre ETX et QUEUE :

- Tout d'abord nous avons défini une nouvelle métrique qui est **RPL-DAG-MC-ETX-QUEUE** dans le fichier **contiki-ng/os/net/routing/rpl-lite/rpl-conf.h** qui représente la combinaison entre ETX et QUEUE pour l'utiliser dans le fichier **contiki-ng/os/net/routing/rpl-lite/mrhof.c** pour qu'un nœud puisse choisir son parent à partir de cette métrique.

```
/* The MC advertised in DIOs and propagating from the root */
#ifdef RPL_CONF_DAG_MC
#define RPL_DAG_MC RPL_CONF_DAG_MC
#else
#define RPL_DAG_MC RPL_DAG_MC_ETX_QUEUE
#endif/* RPL_CONF_DAG_MC */
```

**Figure 4. 9** Déclaration de la métrique RPL-DAG-MC-ETX-QUEUE

- Nous avons ajouté deux champs COUT et QUEUE dans le fichier **contiki-ng/os/net/routing/rpl-lite/rpl-types** tel que QUEUE représente le nombre de paquets dans la file d'attente et le cout représente la métrique utilisée pour choisir le parent

```

/** \brief Logical representation of a DAG Metric Container. */
struct rpl_metric_container {
    uint8_t type;
    uint8_t flags;
    uint8_t aggr;
    uint8_t prec;
    uint8_t length;
    union metric_object {
        struct rpl_metric_object_energy energy;
        uint16_t etx;
        uint64_t cout;
        int queue;
    } obj;
};
typedef struct rpl_metric_container rpl_metric_container_t;

```

**Figure 4. 10** Insertion de deux champs dans RPL-MC

- Nous avons récupéré Link-layer adresse du voisin passé dans les paramètres de la fonction **nbr\_path\_costen** utilisant la fonction **rpl\_neighbor\_get\_lladr**, qui existe dans **contiki-ng/os/net/routing/rpl-lite/rpl-neighbor.h**

```

/**
 * Returns a neighbors's link-layer address
 *
 * \param nbr The neighbor
 * \return The link-layer address if any, NULL otherwise
 */
const linkaddr_t *rpl_neighbor_get_lladdr(rpl_nbr_t *nbr);

```

**Figure 4. 11** La fonction qui récupère Link layer address du voisin

```

static uint16_t
nbr_path_cost(rpl_nbr_t *nbr)
{
    uint16_t base;
    const linkaddr_t *linkadr = rpl_neighbor_get_lladdr(nbr);

    if(nbr == NULL) {
        return 0xffff;
    }

#if RPL_WITH_MC
    /* Handle the different MC types */
    switch(curr_instance.mc.type) {
        case RPL_DAG_MC_ETX:
            base = nbr->mc.obj.etx;
            LOG_INFO(" metric est etx \n");
            break;
        case RPL_DAG_MC_ENERGY:
            base = nbr->mc.obj.energy.energy_est << 8;
            LOG_INFO(" metric est enrg \n");
            break;
        case RPL_DAG_MC_ETX_QUEUE:
            /*recuperer link layer address du voisin et calculer le nombre des paquets dans sa fille d'attente */
            nbr->mc.obj.queue= tsch_queue_packet_count(linkadr);

```

**Figure 4. 12** La récupération Link layer adresse du voisin passé en paramètre

- après avoir récupérer l'adresse, nous avons récupérer le nombre des paquets dans la file d'attente de ce voisin en utilisant la fonction `tsch_queue_packet_count` qui se trouve dans le fichier `contiki-ng/os/net/mac/tsch/tsch-queue`.

```

/* Returns the number of packets currently in the queue */
int
tsch_queue_packet_count(const linkaddr_t *addr)
{
    struct tsch_neighbor *n = NULL;
    if(!tsch_is_locked()) {
        n = tsch_queue_add_nbr(addr);
        if(n != NULL) {
            return ringbufindex_elements(&n->tx_ringbuf);
        }
    }
    return -1;
}

```

**Figure 4. 13** L'implémentation de la fonction qui retourne le nombre de paquets

```

/*recupérer link layer address du voisin et calculer le nombre des paquets dans sa fille d'attente */
if (node_id != 1){
    nbr->mc.obj.queue= tsch_queue_packet_count(linkadr);}

    afficher link layer address et ipv6 address du voisin
    LOG_INFO("link layer adr neighbor : ");
    LOG_INFO_LLADDR(linkadr);
    LOG_INFO_("\n");
    LOG_INFO("ipv6 adr neighbor : ");
    LOG_INFO_6ADDR(rpl_neighbor_get_ipaddr(nbr));
    LOG_INFO_("\n");
    LOG_INFO(" metric est cout \n");

/* affecter le nombre des paquets + ETX dans la variable base */
base =0.5* nbr->mc.obj.etx +0.5* (uint32_t)nbr->mc.obj.queue;
}

```

**Figure 4. 14**Récupération de nombre de paquets

Nous avons modifié aussi la fonction `update_metric_conatiner` qui se trouve dans **contiki-ng/os/net/routing/rpl-lite/mrhof.c**

```

/* Handle the different MC types */
switch(curr_instance.mc.type) {
case RPL_DAG_MC_NONE:
    break;
case RPL_DAG_MC_ETX:
    curr_instance.mc.length = sizeof(curr_instance.mc.obj.etx);
    curr_instance.mc.obj.etx = path_cost;
    break;
case RPL_DAG_MC_ENERGY:
    curr_instance.mc.length = sizeof(curr_instance.mc.obj.energy);
    if(curr_instance.dag.rank == ROOT_RANK) {
        type = RPL_DAG_MC_ENERGY_TYPE_MAINS;
    } else {
        type = RPL_DAG_MC_ENERGY_TYPE_BATTERY;
    }
    curr_instance.mc.obj.energy.flags = type << RPL_DAG_MC_ENERGY_TYPE;
    /* Energy_est is only one byte, use the least significant byte of the path metric. */
    curr_instance.mc.obj.energy.energy_est = path_cost >> 8;
    break;
case RPL_DAG_MC_ETX_QUEUE:
    curr_instance.mc.length = sizeof(curr_instance.mc.obj.cout);
    curr_instance.mc.obj.cout = 0.5*curr_instance.mc.obj.etx +0.5*curr_instance.mc.obj.queue;
    break;
}

```

**Figure 4. 15** Ajout d'un nouveau cas dans la fonction update metric container

## 4.4 Conclusion

Dans ce chapitre nous avons présenté et expliqué les modifications apporté dans les différentes fonctions des protocoles RPL et TSCH pour permettre l'interaction entre ces deux derniers. Des captures sont jointes pour illustrer le travail réalisé.

Le chapitre suivant montre les résultats obtenus ainsi qu'une étude comparative visant à évaluer les performances de l'approche QL-ETX.



# Chapitre 05 : Evaluation de performances de QL-ETX

## 5.1 Introduction

Dans le chapitre précédent, nous avons présenté et implémenté l'approche QL-ETX qui a été conçu pour le but de minimiser la latence dans les réseaux LLNs, il s'agit d'une nouvelle fonction objective qui sert à trouver le prochain saut selon les deux métriques ETX et QUEUE.

Dans ce chapitre, nous présentons les résultats obtenus lors de la simulation de notre approche Cross-Layer (QL-ETX), et nous allons la comparer avec les résultats du protocole inter-couche (MRHOF).

## 5.2 Scénario de simulation

Afin de tester le comportement général de QL-ETX par rapport à MRHOF, plusieurs simulations ont été déroulées sous le simulateur Cooja, un réseau de 40 nœuds qui tentent d'envoyer des paquets de données à un nœud racine (sink).

Lorsqu'un nœud du réseau envoie un paquet de données, la procédure de sélection de père est appelée pour obtenir le meilleur parent, cette procédure est exécutée au niveau de chaque nœud qui reçoit le paquet pour le délivrer au sink.

Afin d'introduire une perte dans le support sans fil, nous utilisons l'Unit Disk Graph Medium de Cooja (UDGM) Distance\_Loss qui est un modèle dans lequel la plage de transmission est modélisée comme un disque idéal où tous les nœuds situés dans la distance de transmission peuvent recevoir les paquets tandis que les nœuds situés derrière ce disque ne reçoivent pas de paquets.

Le scénario du réseau est illustré dans la figure 5.1. Le cercle vert indique la zone de transmission. Le cercle gris indique la zone d'interférence. Le tableau 5.1 contient les paramètres de simulation.

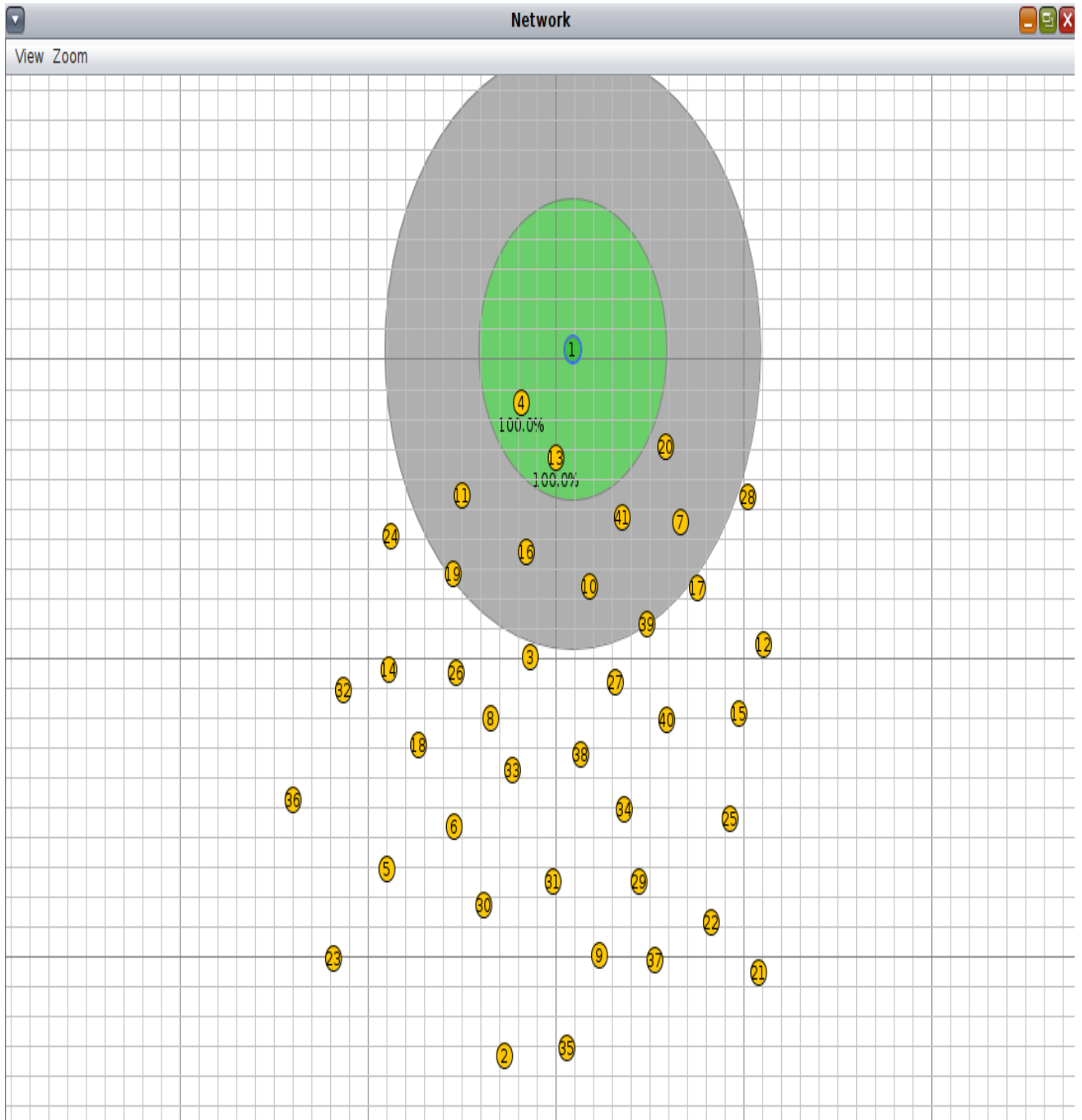


Figure 5. 1 Topologie aléatoire de 40 nœuds

Le scénario de simulation consiste à un ensemble de nœuds répartis d'une manière aléatoire sur une surface de 900 m<sup>2</sup> avec une zone de transmission de 50m et une zone d'interférence de 100m.

Nous exécutons 10 simulations pour chaque fonction objective. Chaque simulation est exécutée pendant 15 min pour 10, 20, 30,40 nœuds pour garantir une topologie a multi sauts, cette durée est suffisante pour avoir des résultats nécessaires.

Le protocole TSCH est utilisé pour améliorer à la fois l'efficacité énergétique et la fiabilité en combinant le TDMA et le saut de canal.

On suppose que :

- La batterie est toujours pleine (toujours chargée).
- Aucun nœud ne tombe en panne.

**Tableau 5. 1** Paramètres de simulation

Paramètre	Valeur
Système d'exploitation	Contiki-ng
Simulateur	Cooja
Temps de simulation	900 s = 15 mn
Nombre de simulation	10
Taille de la grille	300*300 m
Nombre de nœuds	10, 20, 30, 40,50
Fonction objective	MRHOF, QL-ETX
Plateforme	Cooja Mote
Model radio	UDGM
Zone de transmission	50 m
Zone d'interférence	100m
Couche mac	TSCH
Couche réseau	RPL
Intervalle d'envoi de paquets de données	3 paquets/ s
Taux de réussite TX /RX	100%/100%

### 5.3 Métriques de performance

Différentes métriques sont définies pour évaluer les performances de MRHOF , QL-ETX et QL selon le scénario mis en jeu :

- ✓ **La latence moyenne** : est le temps moyen requis par le paquet de données pour parcourir la distance de la source à la destination, ce temps est calculée en utilisant l'équation suivante:

$$Latence\ moyenne = \frac{\sum_{k=1}^n \text{temps de reception} - \text{temps d'envoi}}{n} \quad (5.1)$$

Ou  $n$  est le nombre total de paquets reçus avec succès. Les informations de synchronisation sont fournies par le simulateur Cooja.

- ✓ **PDR (taux de livraison de paquets)** : Pour calculer le taux de livraison de paquets, nous mesurons le nombre de paquets envoyés de tous les nœuds à la racine et le divisons par le nombre de paquets reçus avec succès dans la racine. Comme le montre l'équation suivante :

$$PDR = \frac{\sum \text{des paquets recus par le sink}}{\sum \text{des paquets envoyer par un noeud } i} \quad (5.2)$$

- ✓ **Débit** : est la quantité de paquets transmis avec succès d'un nœud à un autre au cours d'une période donnée. Il est calculé selon l'équation suivante :

$$Debit = \frac{\sum \text{paquets recus par lesink}}{\text{temps desimulation}} \quad (5.3)$$

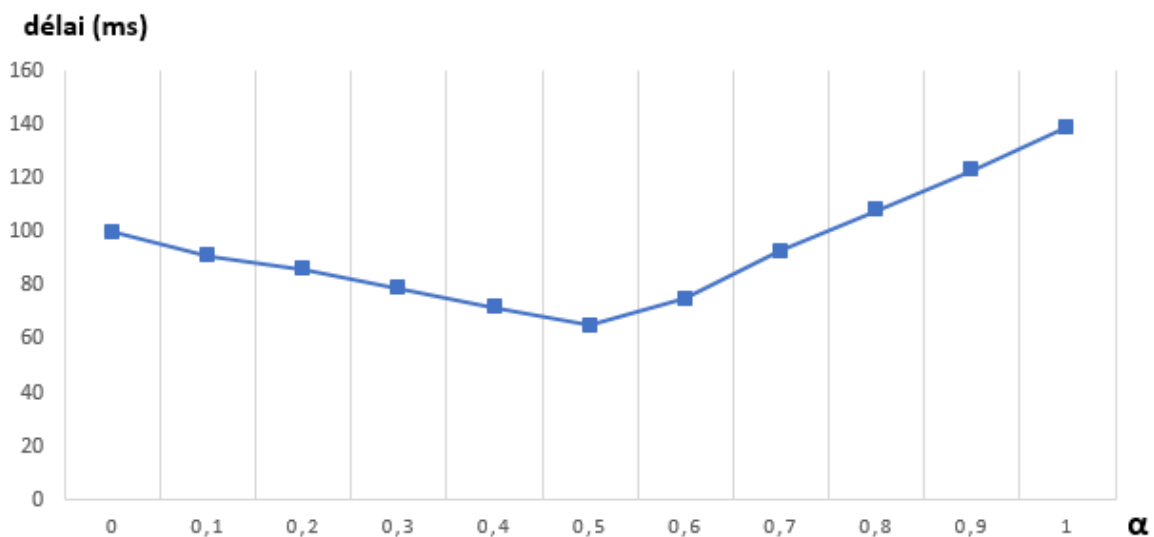
## 5.4 Evaluation des performances

La nouvelle fonction objective QL-ETX qui a été implémenté pour permettre l'interaction entre les deux couches liaison de données et réseau , a été testé et évalué afin de la comparer avec la fonction objective MRHOF qui est basé seulement sur la couche réseau et QL .

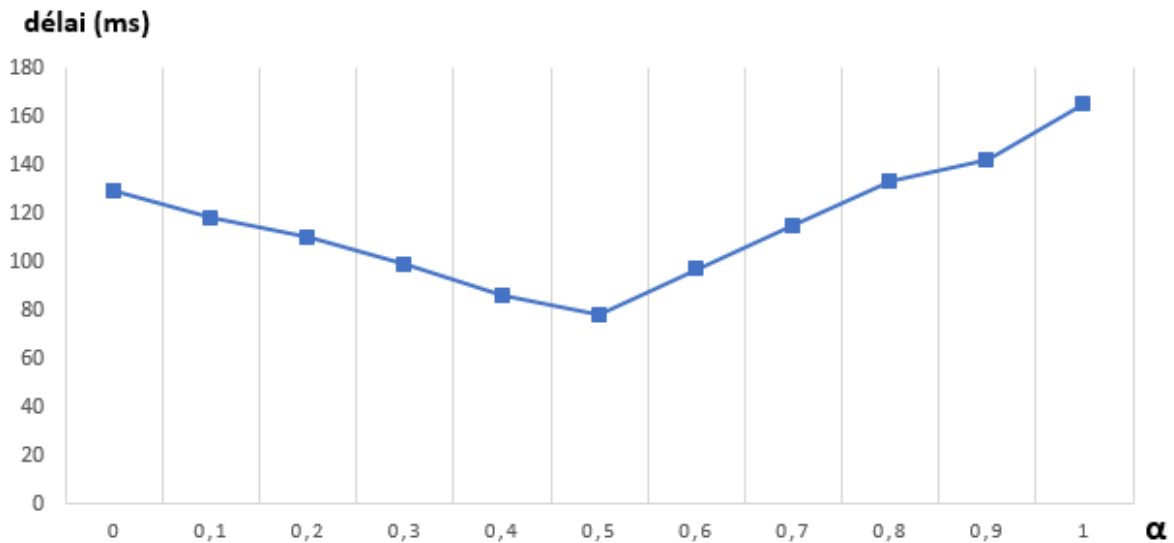
Dans un premier temps nous avons varié la valeur de  $\alpha$  de 0 à 1 pour avoir une meilleure combinaison entre les deux paramètres dans le but d'obtenir la meilleure performance selon l'équation qui suit :

$$\text{Cout} = \alpha * \text{QUEUE} + (1 - \alpha) * \text{ETX} \quad (5.4)$$

Les figures 5.2 et 5.3 illustrent les résultats obtenus lors de la variation de  $\alpha$  respectivement pour un nombre de nœuds égale à 20 et 40, qui montre que la valeur 0.5 est la meilleure pour obtenir une meilleure latence , c'est-à-dire une combinaison de 50% de ETX et 50% de QUEUE.

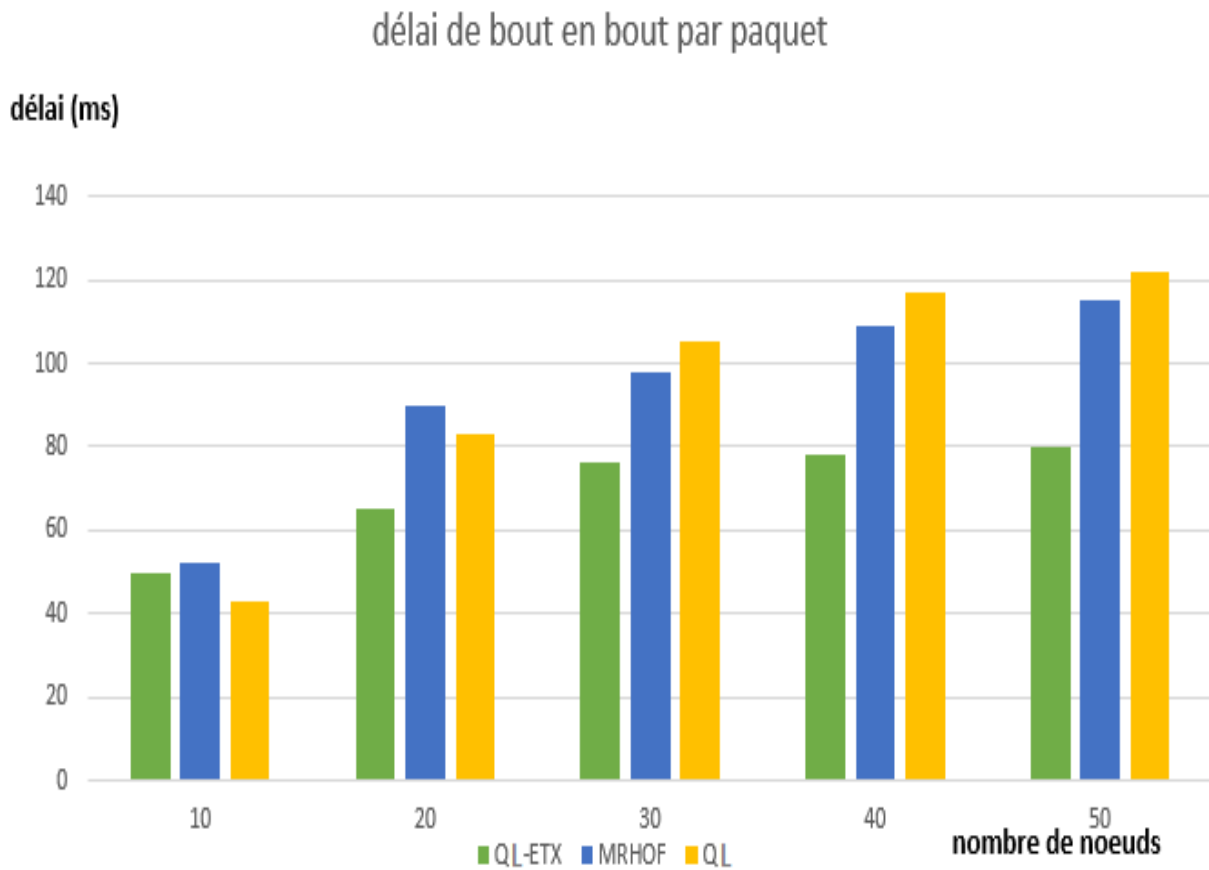


**Figure 5. 2** Variation de  $\alpha$  pour 20 nœuds



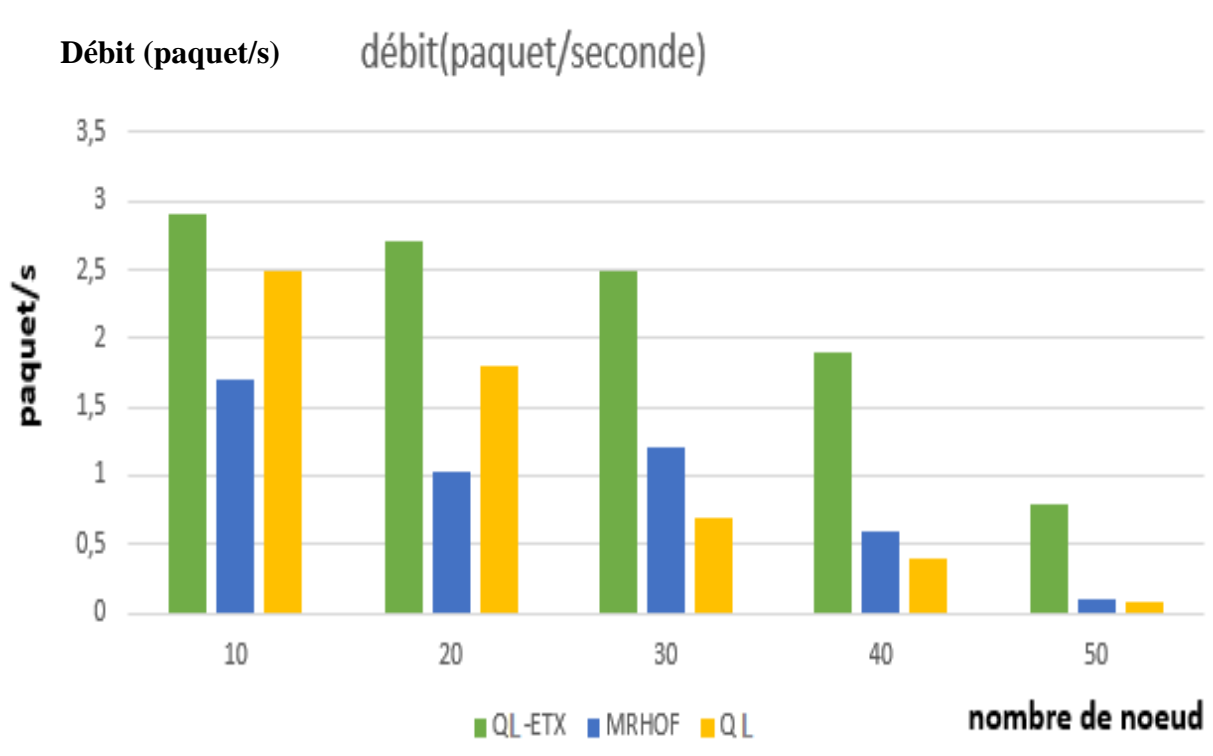
**Figure 5. 3** Variation de  $\alpha$  pour 40 nœuds

Après avoir fixé la valeur de  $\alpha$  à 0.5 nous avons calculé le délai par paquet pour un ensemble de nœuds pour QL-ETX( nombre de noeuds>10) et nous l'avons comparé à la fonction objective MRHOF et QL, comme le montre la figure 5.4. Les résultats montrent que QL-ETX a un meilleur délai pour le réseau par rapport à MRHOF, car la fonction objective QL-ETX se base sur deux métriques qui aident le nœud à choisir le parent moins congestionné avec une qualité de lien fiable par conséquent le délai de réception diminue contrairement au QL qui utilise le nombre de paquet seulement qui mène à choisir une qualité de lien mauvaise ce qui implique un délai élevés, comme le montre la figure 5.4.



**Figure 5. 4** Délai QL-ETX vs MRHOF vs QL.

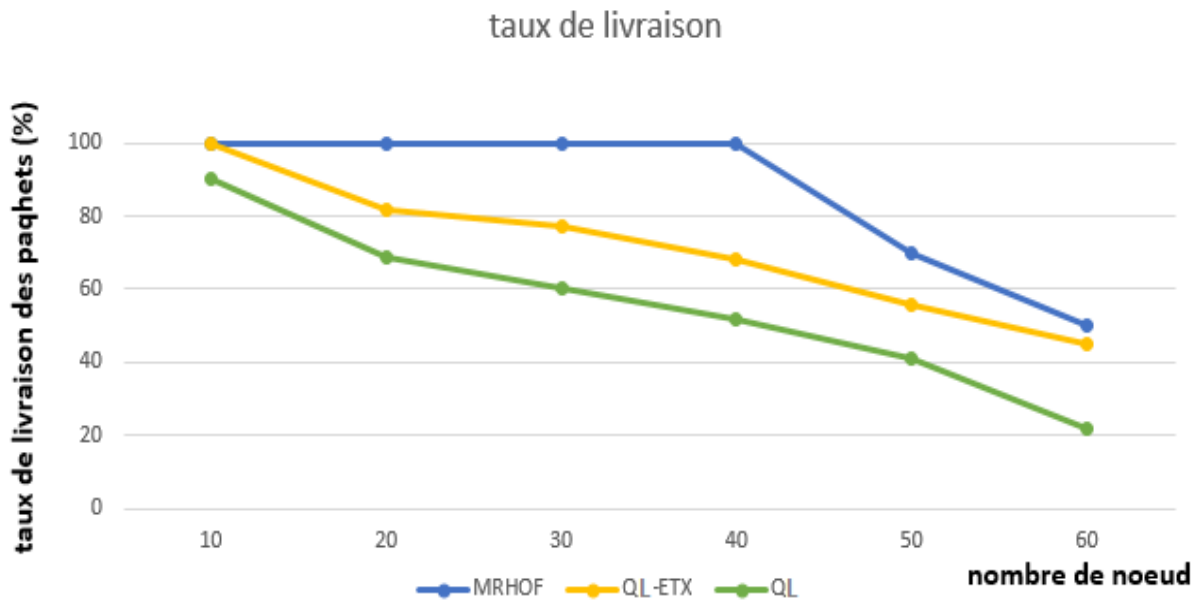
Nous avons calculé le débit (paquet/seconde) en fonction du nombre de nœuds pour notre fonction objective QL-ETX et nous l'avons comparé aux deux autres fonctions objectives, nous remarquons que le débit de QL-ETX est meilleur que celui des deux autres, comme le montre la figure 5.5.



**Figure 5. 5** Débit QL-ETX vs MRHOF vs QL.

Nous avons également calculé le taux de livraison PDR pour la fonction objective QL-ETX et QL et nous l'avons comparé avec la fonction objective MRHOF, nous avons remarqué que MRHOF a un taux de livraison mieux que QL dans la majorité des cas comme le montre la figure 5.6 ,et cela montre l'importance de notre combinaison QL-ETX ,qui donne un résultat moyen comparé à MRHOF et QL .





**Figure 5. 6** Taux de livraison QL-ETX vs MRHOF vs QL

### 5.5 Conclusion

Dans ce chapitre nous avons présenté les différents résultats obtenus lors de l'évaluation des performances de la nouvelle fonction objective proposé QL-ETX cette solution est testé sous Cooja. Nous avons fait une analyse comparative entre notre proposition et MRHOF et QL, les résultats ont montré que notre solution est meilleure lorsqu'il s'agit du délai de bout en bout, par contre le taux de livraison est mieux dans MRHOF par rapport à QL-ETX, car compter sur ETX à 100% est plus fiable que 50 %.

Par conséquent nous avons atteint notre but qui est la minimisation de la latence du réseau.

## Conclusion générale

Les réseaux LLNs (Low Power and Lossy Networks) constituent un axe de recherche très fertile avec un large potentiel et diverses applications pratiques et utiles (industrie, la communication dans un champ de bataille, télémédecine, la gestion des catastrophes naturelles, etc.). Cependant, il y a encore beaucoup de défis qui doivent être abordés pour garantir un fonctionnement efficace de ces réseaux dans des applications réelles (passage à l'échelle, manque de ressources, liens non fiables, etc.). Pour cela plusieurs protocoles sont proposés pour effectuer le routage de données dans les réseaux IoT comme le protocole RPL.

Dans ce mémoire, nous avons traité les décisions non corrélées entre les protocoles de routage et les protocoles mac, qui affectent les métriques liées à plusieurs couches telles que la latence des communications, qui est dû à l'indépendance entre ces deux couches, par conséquent on a exploré qu'on peut utiliser les informations de la couche mac pour améliorer le routage de la couche réseau. Alors le travail est centré sur la communication entre la couche mac et la couche réseau (Cross-Layer) afin d'améliorer les performances des réseaux LLNs.

Nous avons présenté d'abord des généralités sur les LLNs, la norme IEEE802.15.4e, les protocoles de routage en particulier RPL et le modèle en couches assurant les communications dans de tels réseaux. Puis, nous avons proposé une classification des approches Cross-Layer entre le routage et mac et nous avons présenté les travaux existants dans cet axe de recherche. Cette classification comporte deux classes principales : la classe des approches de routage utilisant des informations du TDMA, la classe des approches mac utilisant des informations du routage, enfin nous avons donné une taxonomie de différentes approches Cross-Layer proposée.

Pour améliorer le protocole de routage RPL nous avons utilisé les informations de la couche mac, pour ce faire nous avons proposé une nouvelle approche QL-ETX qui sert à combiner deux métriques, la première de la couche mac et la deuxième de la couche réseau, pour créer une nouvelle fonction objective afin de construire l'arbre RPL pour acheminer les paquets en plus court délai.

Cette approche est implémentée sous le simulateur Cooja, ou on a ajouté des codes dans le Contiki-ng pour atteindre notre but.

Les résultats de la simulation montrent que notre approche améliore le délai de bout en bout par paquet, le débit, le taux de perte. Ces améliorations sont dues au fait que notre solution a permis de réduire la surcharge au niveau d'un nœud.

Ce travail nous a permis de se familiariser avec les réseaux LLNs, il nous a aussi fait découvrir une nouvelle plateforme de simulation adéquate qui est Cooja. Nous avons réussi à déployer notre approche. En outre, dans ce projet on a rencontré certaines difficultés dans la réalisation de ce projet tel que la limitation de la documentation sur la norme TSCH et 6Tisch, sur la simulation sous Cooja et le nouveau système d'exploitation Contiki-ng, Surtout dans ces circonstances exceptionnelles du covid19.

Le travail réalisé dans le cadre de ce projet peut être étendu de plusieurs façons.

Nous pouvons citer à titre d'exemple :

- Il est également nécessaire de considérer l'aspect de mobilité dans les réseaux IoT, où la majorité des nœuds sont mobiles, alors il serait très intéressant de tester l'impact de la mobilité sur les performances de notre approche.
- Tester la communication multi-sink car ces approches permettent de corriger les problèmes de la consommation énergétique.
- Enfin, il serait intéressant de mener une étude expérimentale pour voir les performances de l'approche sur une plateforme IoT telle que IoT-LAB.

## Bibliographies

- [1] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," vol. 10, no. 4, pp. 2233–2243
- [2] Ghaleb, B., Al-Dubai, A., Ekonomou, E., Alsarhan, A., Nasser, Y., Mackenzie, L., & Boukerche, A. (2018). A Survey of Limitations and Enhancements of the IPv6 Routing Protocol for Low-power and Lossy Networks: A Focus on Core Operations. *IEEE Communications Surveys & Tutorials*, 1–1.
- [3] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376.
- [4] Debasis Bandyopadhyay · Jaydip Sen, , "Internet of Things - Applications and Challenges in Technology and Standardization ”.
- [5] Li, S., Xu, L.D. & Zhao, S. The internet of things: a survey. *InfSyst Front***17**, 243–259 (2015).
- [6] Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context Aware Computing for The Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, 16(1), 414–454.
- [7] K. Ashton, "That 'internet of things' thing in the real world, things matter more than ideas," *RFID Journal*, June 2009, <http://www.rfidjournal.com/article/print/4986> [Accessed on: 2012-07-30].
- [8] International Telecommunication Union, "Itu internet reports 2005: The internet of things," International Telecommunication Union, Workshop Report, November 2005, <http://www.itu.int/dms pub/itu-s/opb/pol/SPOL- IR.IT-2005-SUM-PDF-E.pdf> [Accessed on: 2011-12-12]
- [9] Z. Yang et al., "Study and application on the architecture and key technologies for IOT," in *Proc. ICMT*, 2011, pp. 747–751.

- [10] L. Tan and N. Wang, "Future Internet: The Internet of Things," in Proc. 3rd ICACTE, 2010, pp. V5-376–V5-380.
- [11] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things architecture, possible applications and key challenges," in Proc. 10th Int. Conf. FIT, 2012, pp. 257–260
- [12] R. Want, "An introduction to RFID technology," IEEE Pervasive Comput., vol. 5, no. 1, pp. 25–33, Jan.–Mar. 2006.
- [13] R. want "Near field communication," IEEE Pervasive Comput., vol. 10, no. 3, pp. 4–7, Jul./Sep. 2011
- [14]. Yeh, L.-W. & Pan, M.-S. (2014). Beacon scheduling for broadcast and convergecast in ZigBee wireless sensor networks. Computer Communications, 38, 1-12
- [15] J. Huang, F. Qian, Z. M Mao, S sen, and O. Spatscheck, "screen-off traffic characterization and optimization in 3G /4G networks" proc ACM SIGCOMM Internet Meas.Conf.IMC,pp.357-363,2012
- [16] E. Ferro and F. Potorti, "Bluetooth and Wi-Fi wireless protocols: A survey and a comparison," IEEE Wireless Commun., vol. 12, no. 1, pp. 12–26, Feb. 2005.
- [17] P. McDermott-Wells, "What is Bluetooth?" IEEE Potentials, vol. 23, no. 5, pp. 33–35, Jan. 2005.
- [18] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the Internet of Things: Early progress and back to the future," Proc. IJSWIS, vol. 8, no. 1, pp. 1–21, Jan. 2012.
- [19] L. Atzori, A. Lera, and G. Morabito. The Internet of Things: A Survey. Computer Networks 54(15), 2787-2805. (2010).
- [20] S. Krco, B. Pokric, and F. Carrez, "Designing IoT architecture(s): A European perspective," in Proc. IEEE WF-IoT, 2014, pp. 79–84

- [21] Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82, 395–411. doi:10.1016/j.future.2017.11.022
- [22] A. Brandt, J. Buron, and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, Apr. 2010
- [23] Khanafer M., Guennoun M., Mouflah H.T. A Survey of Beacon-Enabled IEEE 802.15.4 MAC Protocols in Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* 2014:16856–16876. doi: 10.1109/SURV.2013.112613.00094.
- [24] IEC 62591:2016—Industrial Networks—Wireless Communication Network and Communication Profiles—WirelessHART. International Electrotechnical Commission; Geneva, Switzerland: 2016.
- [25] IEC 62734:2014—Industrial Networks—Wireless Communication Network and Communication Profiles—ISA 100.11a.: 2014. International Electrotechnical Commission; Geneva, Switzerland: 2014.
- [26] IEC 62601:2015—Industrial Networks—Wireless Communication Network and Communication Profiles—WIA-PA. International Electrotechnical Commission; Geneva, Switzerland: 2015.
- [27] ZigBee Alliance, ZigBee Specification, 2012. [(Accessed on 19 October 2018)].
- [28] C. Bormann, M. Ersue, and A. Keranen, "Terminology for Constrained-Node Networks" , RFC 7228, May 2014.
- [29] Analysis of Energy Consumption and Topology of Routing Protocol for Low-Power and Lossy Networks, 2018.
- [30] IEEE. (2003). IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control

(MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Std 802.15.4-2003.

- [31] Lu, C., Blum, B. M., Abdelzaher, T. F., Stankovic, J. A. & He, T. (2002, Sep.). RAP: real-time communication architecture for large-scale wireless sensor networks. Proceedings. Eighth IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 55-66.
- [32] T. Van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in Proceedings of the 1st international conference on Embedded networked sensor systems. ACM, 2003, pp. 171–180.
- [33] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur and R. 73Alexander, "RPL: IPv6 Routing Protocol for LowPower and Lossy Networks", RFC 6550, Mar. 2012.
- [34] Hui, J.W.; Culler, D.E. Extending IP to Low-Power, Wireless Personal Area Networks, IEEE Internet Computing, 2008, vol. 12, no. 4, pp. 37 - 45.
- [35] Daidone, R., Dini, G. & Anastasi, G. (2014). On evaluating the performance impact of the IEEE 802.15.4 security sub-layer. Computer Communications, 47, 65-76.
- [36] Guglielmo, D. D., Restuccia, F., Anastasi, G., Conti, M. & Das, S. K. (2016). Accurate and Efficient Modeling of 802.15.4 Unslotted CSMA/CA through Event Chains Computation. IEEE Transactions on Mobile Computing, 15(12), 2954-2968.
- [37] Yazdi, E. T., Moravejosharieh, A., Willig, A. & Pawlikowski, K. (2014, Nov). Coupling power and frequency adaptation for interference mitigation in IEEE 802.15.4-based mobile body sensor networks: Part II. Australasian Telecommunication Networks and Applications Conference (ATNAC), pp. 105-110.
- [38] IEEE802.15.4e. (2012). IEEE Standard for Local and Metropolitan Area Networks. Part 15.4: Low-Rate Wireless Personal Area Networks Amendment 1 : MAC Sublayer.
- [39] Jose Vera-Pérez, David Todolí-Ferrandis, Salvador Santonja-Climent, Javier Silvestre-Blanes, and Víctor Sempere-Payá, A Joining Procedure and Synchronization for TSCH-RPL Wireless Sensor Networks , Sensors, volume 18(10); 2018.

- [40] Chang, T., Watteyne, T., Pister, K. & Wang, Q. (2015). Adaptive synchronization in multi-hop TSCH networks. *Computer Networks*, 76, 165-176.
- [41] Kushalnagaret, N. al. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals, IETF RFC 4919, 2007.
- [42] Shin, M.K.; Kim, H.J. L3 Mobility Support in Large-scale IP-based Sensor Networks (6LoWPAN), 11th International Conference Advanced Communication Technology (ICACT), 2009, vol. 2, pp. 941 – 945
- [43] A. B, K., Ng, C. K., Noordin, N. K., & Ali, B. M. (2010). A Review of 6LoWPAN Routing Protocols. *Proceedings of the Asia-Pacific Advanced Network*, 30(0), 71. doi:10.7125/apan.30.11
- [44] Kushalnagar, N.; Montenegro, G.; Hui, J.; Culler, D. Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC4944, 2007.
- [45] Kim, H.G. “Protection against Packet Fragmentation Attacks at 6LoWPAN Adaptation Layer”, *International Conference on Convergence and Hybrid Information Technology*, 2008, pp. 796 - 801.
- [46] José V. V. Sobral 1,2 , Joel J. P. C. Rodrigues 1,3,4,5,6,\* ,† , Ricardo A. L. Rabêlo 4 , Jalal Al-Muhtadi 5 and Valery Korotaev, *Routing Protocols for Low Power and Lossy Networks in Internet of Things Applications*, 2019.
- [47] S. Saranya1 , Dr. R. Manika Chezian2 ,”Comparison of Proactive, Reactive and Hybrid Routing Protocol in MANET” Vol. 5, Issue 7, July 2016
- [48] Clausen, T.; Yi, J.; Lavenu, C.; Lys, A.; Niktash, A.; Igarashi, Y.; Satoh, H. The LLN On-demand Ad hoc Distance-vector Routing Protocol-Next Generation (LOADng). In *Internet-Draft Draft-Clausen-Iln-Loadng-00.txt*; IETF Secretariat: Fremont, CA, USA, 2011.
- [49] Clausen, T.; Yi, J.; Niktash, A.; Igarashi, Y.; Satoh, H.; Herberg, U.; Lavenu, C.; Lys, T.; Dean, J. The lightweight on-demand ad hoc distance-vector routing protocol-next generation (LOADng). In *Internet-Draft draft-clausen-lln-loadng-15.txt*; IETF Secretariat: Fremont, CA, USA, 2016



- [50] T. Clausen, A. C. de Verdiere, J. Yi, A. Niktash, Y. Igarashi, and U. Herberg, “The Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng),” IETF, Draft, Oct 2012.
- [51] O. Gaddour, A. Koubâa, “RPL in a nutshell: A survey”, in *Computer Networks*, Vol. 56, no. 14, pp. 3163-3178, 2012.
- [52] J. Vasseur et al., “RPL: The IP routing protocol designed for low power and lossy networks,” Internet Protocol for Smart Objects (IPSO) Alliance, San Jose, CA, USA, 2011.
- [53] O. Gnawali and P. Levis, “The Minimum Rank with Hysteresis Objective Function”, IETF RFC 6719, Sep. 2012
- [54] JP. Vasseur, M. Kim, K. Pister, N. Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, Mar. 2012.
- [55] P. Thubert, “Objective Function Zero for the Routing Protocol for LowPower and Lossy Networks (RPL),” IETF RFC 6552, Mar. 2012.
- [56] J. Tripathi, J. de Oliveira and JP. Vasseur, Ed., "Performance Evaluation of the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6687, Oct. 2012
- [57] P. Levis, N. Patel, D. Culler, S. Shenker, "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks", Proc. 1st USENIX/ACM Symp. Networked Systems Design and Implementation (NSDI'04), pp. 15-28, Mar. 2004.
- [58] P. Levis, N. Patel, D. Culler, S. Shenker, "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless SensorNetworks", Proc. 1st USENIX/ACM Symp, pp. 34-53, Mar. 2007.
- [59] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm", RFC 6206, Mar. 2011

- [60] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", in *Computer Networks*, vol. 38, no. 4, pp. 393-422, ISSN 1389-1286, Mar. 2002."
- [61] P.Sabatier : Architecture de Communication pour les Applications Multimédia Interactive dans les Réseaux Sans fil, mémoire de doctorant, Université de Toulouse, juillet 2009.
- [62] J. Martocci, P. De Mil, N. Riou, and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.
- [63] F. Mansourkiaie and M. H. Ahmed, "Cooperative Routing in Wireless Networks: A Comprehensive Survey," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 604-626, Second quarter 2015.
- [64] Q. Tang, L. Yang, G. B. Giannakis and T. Qin, "Battery Power Efficiency of PPM and FSK in Wireless Sensor Networks," in *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1308-1319, Apr. 2007
- [65] A. Meier, T. Rein, J. Beutel and L. Thiele, "Coping with unreliable channels: Efficient link estimation for low-power wireless sensor networks," 5th International Conference on Networked Sensing Systems, Kanazawa, 2008, pp. 19-26.
- [66] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. 1st international conference on Embedded networked sensor systems*, 2003, pp. 1–13.
- [67] LemiaLouail. Approches cross-layer pour l'optimisation de la latence des communications dans les réseaux de capteurs sans fil. Réseaux et télécommunications [cs.NI]. Université de Franche-Comté, 2016.
- [68] E.R.Albu : Architecture de communication pour les réseaux d'instrumentation sans fil, mémoire de doctorant, Institut National Polytechnique de Toulouse, juin 2011.
- [69] Q. Wang and M. A. Abu-Rgheff, "Cross-layer signaling for next-generation wireless systems," *IEEE Wireless Communications and Networking (WCNC)*, pp. 1084 – 1089, 2003.

- [70] D.Pompili, M.C.Vuran and T.Melodia : Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta.
- [71] L. Louail, V. Felea, J. Bernard and H. Guyennet. MAC-aware routing in wireless sensor networks. In International Black Sea Conference on Communications and Networking (BlackSeaCom), pages 225–229. IEEE, 2015
- [72] S. Du, A. Kumar Saha and D. B. Johnson. RMAC : A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks. In INFOCOM, pages 1478–1486, 2007.
- [73] M. Hefeida, T. Canli and A. A. Khokhar. CL-MAC : A Cross-Layer MAC protocol For heterogeneous Wireless Sensor Networks. Ad Hoc Networks, 11(1) :213–225, 2013.
- [74] K. Heurtefeux, F. Maraninchi and F. Valois. AreaCast : A cross-layer approach for a communication by area in Wireless Sensor Networks. In ICON, pages 112–117, 2011.
- [75] Cooja Simulator – Contiki, [http://anrg.usc.edu/contiki/index.php/Cooja\\_Simulator](http://anrg.usc.edu/contiki/index.php/Cooja_Simulator), dernier accès Juillet 2017

