

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université KASDI Merbah - Ouargla

Faculté des Nouvelles Technologies de l'Information et de la Communication

Département d'Informatique et de la Technologie de l'Information



Mémoire

Présenté en vue de l'obtention du diplôme de

Master en Informatique

Spécialité : Administration et sécurité des réseaux

Présenté par : Larouci Aicha

Ben Slimane Afaf

Thème

**Simulation et comparaison de protocoles de communication
pour l'internet des objets**

Encadreur	Membre jurée	
Ghrib Toufik	Pr	Zga. A
	Ex	Euschi. S

Année universitaire : 2022/2023

Remerciements :

Nous remercions en premier lieu ALLAH de nous avoir donné non seulement le courage mais aussi la force et la volonté nécessaire pour la réalisation de ce modeste travail.

Nous tenons à remercier toutes les personnes qui ont contribué de manière directe ou indirecte à l'aboutissement de ce travail :

En premier lieu, Nous remercions très sincèrement M. Ghrib Toufik pour son encadrement et surtout pour l'aide précieuse qu'il nous a apportée tout au long du travail.

Nous remercions également tous les enseignements de spécialité de l'informatique.

Nous terminons par un grand remerciement à nos familles pour leurs encouragements et pour leur soutien moral.

RESUME :

L'Internet des objets est un système qui communique entre les appareils informatiques, les appareils numériques, les personnes et tout ce qui a le pouvoir de transférer n'importe quelle quantité de données dans un réseau sans l'aide d'aucune interaction humaine et humaine à l'ordinateur. L'IoT combine les appareils intelligents et les technologies de communication, et permet à ces appareils d'échanger l'information et les données entre eux. Il y a tellement des protocoles utilisés dans l'environnement Internet des objets comme CoAP, RPL, 6LoWPAN, MQTT etc. Dans ce travail, nous utilisons le protocole CoAP, 6LoWPAN et RPL pour l'évaluation des performances en utilisant le simulateur Cooja dans le système d'exploitation Contiki. Nous avons analysé qu'après examen de chaque protocole est mieux sur sa route dépend de ses applications pour faire comparaison entre eux.

Mots-clés : Internet des objets (IoT), protocole 6LoWPAN, protocole RPL, protocole COAP, simulateur COOJA, comparaison.

ABSTRACT:

Internet of things is a system that communicate between computing devices, digital devices, people and any things that have a power for transferring any amount of data in a network without help of any human-to-human and human-to-computer interaction. IoT combined the smart devices and communication technologies, and it enable these devices to interchange the information and data to each other. There are so many protocols used in Internet of Things environment like CoAP, RPL, 6LoWPAN, MQTT etc. In this work, we are used CoAP, 6LoWPAN and RPL protocol for performance evaluation using Cooja Simulator in Contiki Operating System. We analyzed that after examination of every protocol is better on its route relies on its applications to make comparison between them.

Keywords: Internet of Things (IoT), 6LoWPAN protocol, RPL Protocol, COAP Protocol, COOJA Simulator, comparison.

ملخص:

إنترنت الأشياء هو نظام يتواصل بين أجهزة الكمبيوتر والأجهزة الرقمية والأشخاص وكل ما لديه القدرة على نقل أي كمية من البيانات إلى الشبكة دون مساعدة أي تفاعل بشري وبشري مع الكمبيوتر. تجمع إنترنت الأشياء بين الأجهزة الذكية وتقنيات الاتصال، مما يسمح لهذه الأجهزة بمشاركة المعلومات والبيانات مع بعضها البعض. هناك العديد من البروتوكولات المستخدمة في بيئة إنترنت الأشياء مثل CoAP و RPL و 6LoWPAN و MQTT وما إلى ذلك. البروتوكول هو مجموعة من القواعد التي تستخدم لإجراء اتصال بين نظام الكمبيوتر والأجهزة الذكية على الشبكة. في هذا العمل، نستخدم بروتوكول CoAP و 6LoWPAN و RPL لتقييم الأداء باستخدام المحاكاة Cooja في نظام التشغيل Contiki.. قمنا بتحليل كل بروتوكول بعد فحص أفضل طرق تطبيقاته من أجل إجراء مقارنة بينهم.

الكلمات الرئيسية: إنترنت الأشياء (IoT)، البروتوكول 6LoWPAN، البروتوكول RPL، البروتوكول COAP، المحاكاة COOJA، مقارنة.

LISTE DES MATIERES

REMERCIEMENTS :	III
RESUME :	IV
ABSTRACT:.....	IV
ملخص:.....	IV
LISTE DES MATIERES	V
LISTE DE FIGURES	VII
LISTE DES TABLAUX	IX
ABREVIATIONS	X
INTRODUCTION GENERALE	2
CHAPITRE 1 : PRESENTATION D'IOT (INTERNET DES OBJETS)	5
1.1 INTRODUCTION.....	5
1.2 OBJET CONNECTE (OC)	5
1.3 DEFINITION : QU'EST-CE QUE L' IOT (INTERNET DES OBJETS) ?.....	6
1.4 QUELLE EST L'HISTOIRE DE L'INTERNET DES OBJETS ?.....	7
1.5 CONCEPT DE L' IOT	8
1.6 IMPORTANCE DE L'IOT	9
1.7 ÉVOLUTION DU WEB ET DE L'INTERNET.....	9
1.8 COMMENT FONCTIONNENT L'INTERNET DES OBJETS ?.....	10
1.9 APPLICATIONS DE L'IOT.....	11
1.10 LE M2M	12
1.10.1 Définition.....	12
1.10.2 Différence entre M2M et IoT.....	13
1.11 LES DEFIS ET LES BENEFICES DE L'IOT.....	13
1.11.1 Les Défis de l'IoT.....	13
1.11.2 Les bénéfices de l'IoT.....	14
1.12 L'ARCHITECTURE DE L'IOT	15
1.13 LES PROTOCOLES DE L'IOT	16
1.14 CONCLUSION	18
CHAPITRE 2 : LES PROTOCOLES COAP, 6LOWPAN & RPL	20
2.1 INTRODUCTION.....	20
2.2 LE PROTOCOLE CONTRAINTES APPLICATIF (CoAP)	20
2.2.1 Définition.....	20
2.2.2 Caractéristiques du protocole CoAP	21

2.2.3	<i>Mode de fonctionnement</i>	22
2.2.4	<i>Segmentation d'un message CoAP</i>	24
2.3	6LoWPAN (IPV6 OVER LOW-POWER WIRELESS PERSONAL AREA NETWORK)	25
2.3.1	<i>Définition</i>	25
2.3.2	<i>Architecture 6LoWPAN</i>	26
2.3.3	<i>Fonctionnement de 6LoWPAN</i>	26
2.3.4	<i>Le routage dans le 6LoWPAN</i>	27
2.4	RPL : LE PROTOCOLE DE ROUTAGE DES RESEAUX LLN	28
2.4.1	<i>Définition</i>	28
2.4.2	<i>Fonctionnement du protocole RPL</i>	29
2.4.3	<i>Les modes d'opération du protocole RPL</i>	30
2.4.4	<i>Les paradigmes de communication</i>	31
2.5	CONCLUSION	32
CHAPITRE 3 : SIMULATION DES PROTOCOLES		33
3.1	INTRODUCTION	33
3.2	SIMULATION	33
3.3	ENVIRONNEMENT DE TRAVAIL ET OUTILS DE SIMULATION	33
3.3.1	<i>Outils matériels</i>	33
3.3.2	<i>Outils logiciels</i>	34
3.4	LES ETAPES DE LA SIMULATION	35
3.4.1	<i>Simulation du protocole CoAP</i>	37
3.4.2	<i>Simulation du protocole 6lowpan</i>	40
3.4.3	<i>Simulation du protocole RPL</i>	41
3.5	LES PARAMETRES ESTIMES ET LES UTILES UTILISEES	42
3.6	CONCLUSION	45
CHAPITRE 4 : COMPARAISON ET EVALUATION DE PERFORMANCES DES PROTOCOLES		47
4.1	INTRODUCTION	47
4.2	ANALYSE LES RESULTATS DE SIMULATION ET EVALUATION DE PERFORMANCES DES PROTOCOLES	47
4.2.1	<i>Comparaison selon la consommation d'énergie</i>	47
4.2.2	<i>Comparaison selon l'économisation d'énergie</i>	48
4.2.3	<i>Comparaison selon les paquets reçus</i>	48
4.2.4	<i>Comparaison selon le taux de transmission par paquets</i>	49
4.3	CONCLUSION	49
CONCLUSION GENERALE		52
BIBLIOGRAPHIE		53

LISTE DE FIGURES

<i>N° Figure</i>	<i>Nom de Figure</i>	<i>Page</i>
1.1	Internet des objets	14
1.2	Historique de l'internet des objets	16
1.3	Les applications d'internet des objets	19
1.4	Architectures IoT	24
2.1	En-tête du protocole COAP	28
2.2	Principe du fonctionnement du protocole CoAP	29
2.3	Les deux couches de la protocole COAP	29
2.4	La Sémantique de la Couche du Messagerie	30
2.5	<i>La Sémantique des requêtes et réponses</i>	31
2.6	<i>segmentation de Message COAP</i>	32
2.7	<i>Architecture 6LoWPAN</i>	33
2.8	<i>Les deux modes de routage Route-over Mesh-under</i>	35
2.9	<i>Les graphes DAG et DODAG</i>	36
2.10	<i>L'envoi des messages DIO et DAO et la construction de DODAG</i>	37
3.1	<i>Interface des logicielles</i>	41
3.2	<i>Exécution de Cooja</i>	41
3.3	<i>Interface du simulateur COOJA</i>	42
3.4	<i>Création d'une nouvelle simulation</i>	42
3.5	<i>Les fenêtres de l'Interface du simulateur COOJA</i>	42
3.6	<i>Création des notes</i>	43
3.7	<i>Fenêtre de compilation</i>	44
3.8	<i>Ajout des notes</i>	44
3.9	<i>Fenêtre de serial socket server</i>	45

3.10	<i>Commencement de la simulation</i>	45
3.11	<i>Simulation de 6LoWPAN</i>	47
3.12	<i>Simulation de RPL</i>	48
3.13	<i>Utilisation du PowerTracker</i>	49
3.14	<i>Utilisation de l'outil "Mot output"</i>	49
3.15	<i>Interface Wireshark</i>	51
4.1	<i>Comparaison de la consommation d'énergie pour (CoAP, 6lowpan et RPL)</i>	53
4.2	<i>Comparaison de l'économisation d'énergie</i>	53
4.3	<i>Comparaison selon les paquets reçus</i>	53
4.4	<i>Comparaison selon le taux de transmission par paquets</i>	54

LISTE DES TABLAUX

<i>N° Tableau</i>	<i>Nom de tableau</i>	<i>Page</i>
1.1	Protocoles de chaque couche	25
2.1	<i>Messages de la couche messagerie CoAP</i>	30

Abréviations

Abréviaton	Explication
6LoWPAN	IPv6 Over Low-Power Wireless Personal Area Network
ARPANET	Advanced Research Projects Agency Network
CoAP	Contraintes Applicatif Protocol
DAO	Destination Annonce Objet
DIO	DODAG Information Object
DIS	DODAG Information Sollicitation
ETSI	European Telecommunications Standards Institute
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IT	Information Technology
LBR	Lossy Border Router
M2M	Machine to Machine
MP2P	Multipoint to Point
MQTT	Message Queuing Telemetry Transport
OC	Connected Object
OCE	Connected Object Enriched
OT	Operational Technology
P2P	Point to Point
P2PM	Point to Multipoint
REST	Representational State Transfer

RFID	Radio-Frequency Identification
RPL	Routing Protocol for Low-Power and Lossy Networks
URI	Uniform Resource Identifier

Introduction générale

INTRODUCTION GENERALE

L'Internet des objets (IoT) désigne la connectivité dynamique des appareils physiques avec ressources limitées avec le soutien de l'infrastructure Internet. Il aide à développer l'interaction entre ces entités et aussi tous les objets et réseaux pris en charge par Internet. IoT peut prolonger la communication Internet à des types hétérogènes d'objets qui sont utilisés dans la technologie embarquée pour se joindre à l'environnement par le biais du soutien de la technologie Internet.

Selon les besoins et l'application régulière, l'IoT a déjà été inclus dans le plan de développement quinquennal national de la Chine. L'exploration et l'application de la technologie de l'Internet des objets est certain de la promotion et la révolution, en même temps, pour promouvoir l'économie nationale et d'améliorer constamment la puissance nationale globale.

Le montant des entités liées a déjà dépassé le montant total d'être humain il y a quelques années, et il est prévu d'atteindre 50 milliards d'ici la fin de 2025. Au fur et à mesure que de plus en plus de dispositifs sont introduits dans notre vie normale, il a créé une énorme pression sur la connectivité Internet en termes de stockage et de routage. Ces appareils communiquent constamment entre eux et avec le serveur par l'Internet. Par conséquent, les nécessités de protocoles de communication spécifiques à l'application sont apparues et se poursuivront dans les jours à venir.

Ce travail fournira une étude comparative entre trois protocoles plus couramment utilisés sur l'Internet des objets, 6LoWPAN, RPL, et CoAP. En outre, il sera également meilleure voie de communication entre les nœuds IoT. Une grande partie des données qui sont le véritable enjeu est de développer la capacité des gens de comprendre les variations et leurs déductions plus clairement, et de prendre des actions solides en conséquence.

Différents simulateurs ont été utilisés par un certain nombre de chercheurs, tels que NS-3, Tossim, et OPNET pour différentes plateformes comme TinyOS, POSIX, lwIP, etc. Cependant, COOJA a été choisi pour simuler des nœuds contiki à grande échelle. Ce simulateur est spécialement conçu pour simuler des capteurs qui consomment très peu de puissance et être très précis.

Notre étude est axée sur la recherche du meilleur protocole de communication entre les nœuds IOT, et par conséquent, l'étude peut être utile à la maison applications, soins de santé, services publics, applications d'entreprise et automatisation.

Ce travail est organisé de la manière suivante : Après une introduction générale, nous donnons des informations générales sur l'Internet des objets dans le premier chapitre. Dans le deuxième chapitre, nous présentons les trois protocoles de communication RPL, CoAP, et 6LoWPAN et nous allons expliquer leur fonctionnement. Le troisième chapitre sera consacré à la simulation de nos trois protocoles par le simulateur Cooja.

Dans le quatrième chapitre, nous présentons des résultats obtenus et évaluons les performances sur ces résultats et ferons comparaison entre les trois protocoles. Une conclusion générale et perspectives viennent à la fin de ce travail.

Chapitre 1 :

Présentation d'IoT (Internet des Objets)

Chapitre 1 : Présentation d'IoT (Internet des Objets)

1.1 Introduction

Depuis l'arrivée d'internet, la conception des appareils électroniques s'est considérablement développée. Du plus petit gadget comme les montres jusqu'aux grosses machines comme les voitures, internet les a transformés en de véritables bijoux technologiques. Depuis quelques années, ces objets peuvent s'échanger des informations entre eux sans l'aide de l'homme, grâce à internet. Ce processus de connexion est ce qu'on appelle **Internet des Objets** ou **IdO**, tiré de l'anglais **Internet of Things** ou **IoT**. Mais comment fonctionne-t-il vraiment ? A-t-il un impact considérable dans la vie quotidienne de l'humanité ? Pour mieux comprendre le concept, nous vous invitons à voir tous les détails dans ce chapitre.

1.2 Objet connecté (OC)

Avant de définir les concepts d'IoT, il est important de définir l'objet connecté qui est un dispositif dont la finalité première n'est pas d'être un système informatique ni une interface d'accès au web, exemple, un objet tel qu'une machine à café ou une serrure était conçue sans intégration de systèmes informatiques ni connexion à Internet. L'intégration d'une connexion Internet à un OC permet de l'enrichir en termes de fonctionnalité, d'interaction avec son environnement, il devient un OC Enrichi (OCE), par exemple, l'intégration d'une connexion internet à la machine à café la rendant accessible à distance.

Un OC peut interagir avec le monde physique de manière indépendante sans intervention humaine. Il possède plusieurs contraintes telles que la mémoire, la bande passante ou la consommation d'énergie, etc. Il doit être adopté à un usage, il a une certaine forme d'intelligence, une capacité de recevoir, de transmettre des données avec des logiciels grâce aux capteurs embarqués. Un objet connecté a une valeur lorsqu'il est connecté à d'autres objets et briques logicielles, par exemple : une montre connectée n'a d'intérêt qu'au sein d'un écosystème orienté santé/bien-être, qui va bien au-delà de connaître l'heure.

Un OC à trois éléments clés :

- Les données produites ou reçues, stockées ou transmises.
- Les algorithmes pour traiter ces données.

– L'écosystème dans lequel il va réagir et s'intégrer.

Les propriétés d'usage d'un OC :

– Ergonomie (utilisabilité, maniabilité, ...).

– Esthétisme (formes/couleurs/sons/sensations, ...).

– Usage (histoire culturelle, profil, matrice sociale, ...).

– Méta-Morphisme (adaptabilité, personnalisation, modulation, ...). [1]

1.3 Définition : qu'est-ce que l'IoT (Internet des Objets) ?

L'IoT ou IdO est définie comme une « **infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution** ». Cette définition est tirée de l'Union internationale des télécommunications. Il est clair que tout le monde ne puisse pas comprendre totalement cette définition.

En d'autres termes, l'Internet des Objets désigne **tous les appareils physiques qui peuvent s'échanger des données entre eux** par le biais d'internet. Ces « **objets** » sont équipés de **capteurs** et de **logiciels** qui leur permettent de se connecter à d'autres terminaux. Ils peuvent **transférer des données sur un réseau**, sans qu'une main humaine ne les assiste.

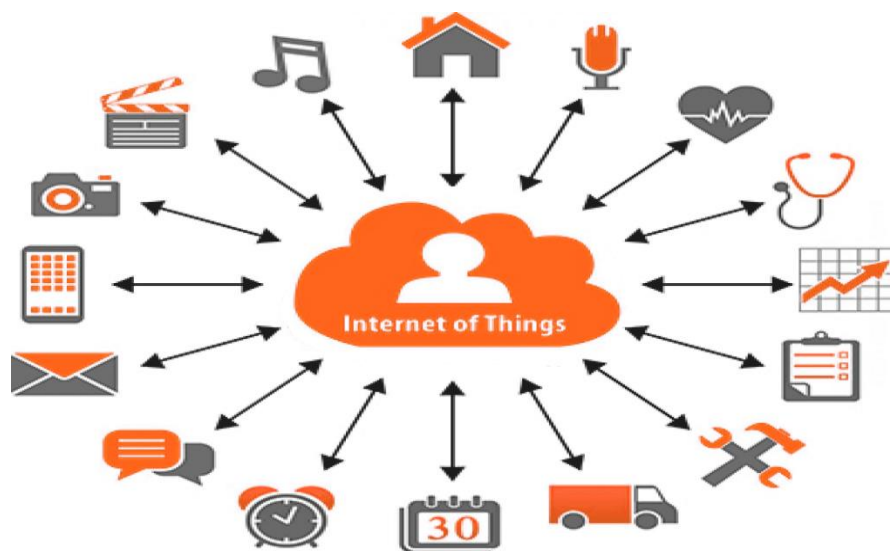


Figure 1.1 : Internet des objets.

La communication entre ces appareils se présente sous diverses formes. Tout d'abord, il y a les objets qui sont connectés directement à internet. Ensuite, la technologie **machine to machine** ou **M2M** permet la communication entre machines via un réseau et des capteurs. Et enfin, il y a les « **smart connected devices** » comme les smartphones et tablettes. Ces appareils permettent de centraliser le contrôle de différents objets connectés via des **applications de gestion**.

L'Internet des Objets est présent dans **presque tous les objets du quotidien**. Il assure les fonctionnalités sur les petits appareils comme les montres connectées ou dans des systèmes de domotique pour la maison. Il peut être aussi présent dans des grandes infrastructures comme les villes intelligentes. Et entre les deux, il y a les dispositifs dans les entreprises de production et de santé. [2]

1.4 Quelle est l'histoire de l'Internet des objets ?

Le premier objet connecté à Internet fut un distributeur de boissons fraîches à l'université Carnegie Mellon (États-Unis) en 1982. La notion d'appareil connecté existe, quant à elle, depuis les années 1970. À l'époque, l'expression « Internet des objets » n'est pas encore inventée : on parle alors d'Internet intégré ou d'informatique omniprésente.

Ce n'est qu'en 1999 que Kevin Ashton, informaticien chez Procter & Gamble, utilise pour la première fois l'expression « Internet des objets ». Cette appellation avait pour but de susciter l'intérêt des décideurs sur la technologie de radio-identification (RFID) ainsi que les autres capteurs qu'il souhaitait mettre en place sur les produits de leur chaîne d'approvisionnement.

Cette même année, le professeur à l'Institut de technologie du Massachusetts (MIT) Neil Gershenfeld publiait *When Things Start to Think*. Dans ce livre, il y décrit clairement la direction prise par l'interconnexion des machines, sans toutefois utiliser l'expression d'Ashton.

L'évolution de l'IoT s'est ensuite fortement construite sur la communication dite « machine to machine » (M2M). Elle fait référence aux appareils capables de se connecter les uns aux autres via un réseau, et ce, sans interaction humaine.

Aujourd'hui, l'IoT est un réseau de capteurs composé de milliards d'objets intelligents, connectant des personnes, des systèmes informatiques et des applications dans le but de partager et de collecter des données. L'IoT est rendu possible par la forte convergence technologique

actuelle (Internet, technologies sans fil, systèmes micro électromécaniques, micro services, etc.) qui se situe entre technologie opérationnelle (OT) et technologie de l'information (IT). [3]

Historique de la Technologie: la Connectivité des choses

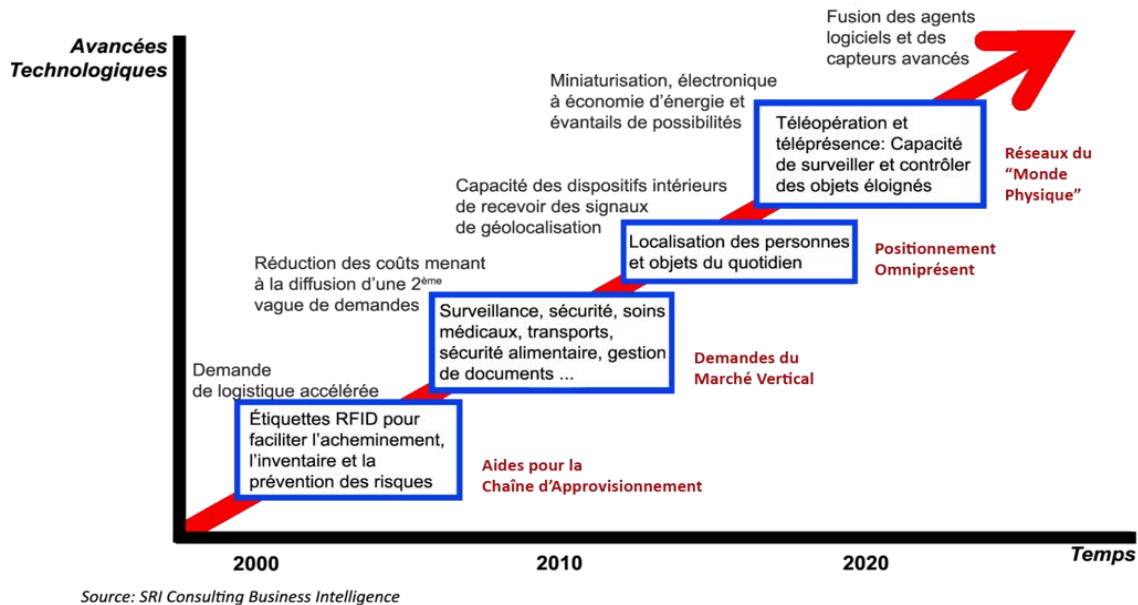


Figure 1.2 : Historique de l'internet des objets. [4]

1.5 Concept de l'IoT

L'Internet des objets repose sur l'idée que tous les objets seront connectés un jour à Internet et seront donc capables d'émettre de l'information et éventuellement de recevoir des commandes. On parle aussi d'"ubiquitous computing", c'est à dire informatique omniprésente, ambiante, ou pervasive. Ce nouveau paradigme informatique est basé non plus sur le PC, mais sur des objets quotidiens intégrant des capteurs et des capacités de communication. L'internet des objets propose de créer une continuité entre le monde réel et le monde numérique : il donne une existence aux objets physiques dans le monde numérique. Sur le plan fonctionnel, l'Internet des objets désigne une informatique qui se fonde dans notre quotidien pour nous simplifier la vie, nous faire gagner du temps, décharger notre cerveau de la mémorisation de données logistiques (itinéraires, agenda, etc.). Il permet de créer de nouveaux usages, comme, par exemple, des informations en temps réel sur la localisation de ses amis. Il permet aussi de faire des mesures exhaustives, là où on se contentait dans le passé d'un simple panel, par exemple avec la mesure du trafic automobile dans les rues de la capitale. [5]

1.6 Importance de l'IoT

Pour pouvoir mesurer l'importance de l'IoT, il faut d'abord comprendre les différences entre l'Internet et le World Wide Web (ou Web), des termes souvent confondus. L'Internet est la couche physique, c'est-à-dire le réseau composé de commutateurs, de routeurs et d'autres équipements. Sa principale fonction est de transporter les informations d'un point A à un point B de façon rapide, fiable et sécurisée. Le Web, lui, est une couche applicative qui intervient sur l'Internet. Son rôle essentiel est de fournir une interface permettant d'exploiter les informations qui circulent sur l'Internet. [5]

1.7 Évolution du Web et de l'Internet

Le Web est passé par plusieurs phases distinctes :

Étape 1 : Le Web 1.0.

Le monde des "éditeurs" Tout a commencé par une phase de recherche pendant laquelle le Web était appelé ARPANET (Advanced Research Projects Agency Network). Le Web était alors surtout utilisé par des universitaires à des fins de recherche. Au commencement du web, les contenus étaient réalisés par quelques éditeurs, qui les publiaient sur des sites web. On avait donc un nombre restreint d'émetteurs d'informations, et un grand nombre de lecteurs : c'est le web 1.0., premier stade d'évolution d'Internet.

Étape 2. : Le Web 2.0.

Et l'utilisateur se mit à produire du contenu Dans sa deuxième phase, le Web peut être qualifié de « brochure électronique ». Nous assistions alors à une véritable prise d'assaut des noms de domaine. Toutes les entreprises ressentaient le besoin de partager des informations sur Internet pour faire connaître leurs produits et leurs services. Avec l'arrivée des premiers réseaux sociaux, chaque individu s'est mis à générer lui-même des contenus. Quand quelqu'un publie un commentaire sur un article, crée un profil sur Facebook, publie une vidéo sur You tube ou écrit un Tweet, il produit un contenu (texte, image, vidéo...) qui pourra être consulté par d'autres. C'est le web dit "social", souvent appelé web 2.0.

Étape 3. : Le web 3.0.

Les machines s'y mettent Désormais, nous voyons émerger de nombreux objets physiques dits "connectés". Ces objets utilisent Internet pour envoyer des informations et en recevoir. Par exemple, des chaussures connectées envoient le nombre de foulées après chaque course à un système distant, cela permettra ainsi de consulter l'historique des courses sur un site web. [5]

1.8 Comment fonctionnent l'Internet des objets ?

Un système IoT comprend des capteurs/appareils qui communiquent avec le Cloud par le biais d'une forme de connectivité. Une fois que les données ont atteint le Cloud, un logiciel les traite et décide d'effectuer une action, comme régler les capteurs/appareils, sans que l'utilisateur ait à intervenir ni à envoyer une alerte.

Un système IoT complet se compose de quatre éléments distincts. Capteurs ou appareils, connectivité, traitement des données et interface utilisateur. Examinons-les tour à tour :

Capteurs/Appareils : Ceux-ci collectent des données à partir de leur environnement. Un appareil peut comporter plusieurs capteurs : par exemple, un smartphone contient un GPS, un appareil photo, un accéléromètre, etc. Essentiellement, le ou les capteurs collectent des données de l'environnement dans un but précis.

Connectivité : Une fois que l'appareil a collecté les données, il doit les envoyer au Cloud. Il le fait de différentes façons : Wi-Fi, Bluetooth, satellite, réseaux étendus à faible puissance (LPWAN) ou connexion directe à Internet via Ethernet. L'option de connectivité spécifique dépendra de l'application de l'IoT.

Traitement des données : Une fois que les données ont atteint le Cloud, le logiciel les traite et peut décider d'effectuer une action. Il peut s'agir d'envoyer une alerte ou de régler automatiquement les capteurs ou l'appareil sans intervention de l'utilisateur. Cependant, l'utilisateur doit parfois intervenir, et c'est là que l'interface utilisateur entre en jeu.

Interface utilisateur : Si l'utilisateur doit intervenir ou s'il veut vérifier le système, une interface utilisateur le lui permet. Toute action effectuée par l'utilisateur est envoyée en sens inverse par le système. De l'interface utilisateur au Cloud et de nouveau aux capteurs/appareils pour effectuer le changement demandé.

Les protocoles précis de connectivité, de mise en réseau et de communication utilisés par les appareils compatibles avec le Web varient en fonction de chaque application de l'IoT. De plus en plus, l'IoT utilise l'intelligence artificielle (IA) et l'apprentissage automatique pour faciliter et accélérer les processus de collecte de données. [6]

1.9 Applications de l'IoT

Il existe toute une série d'applications de l'IoT. Exemples de crypto monnaies connues :

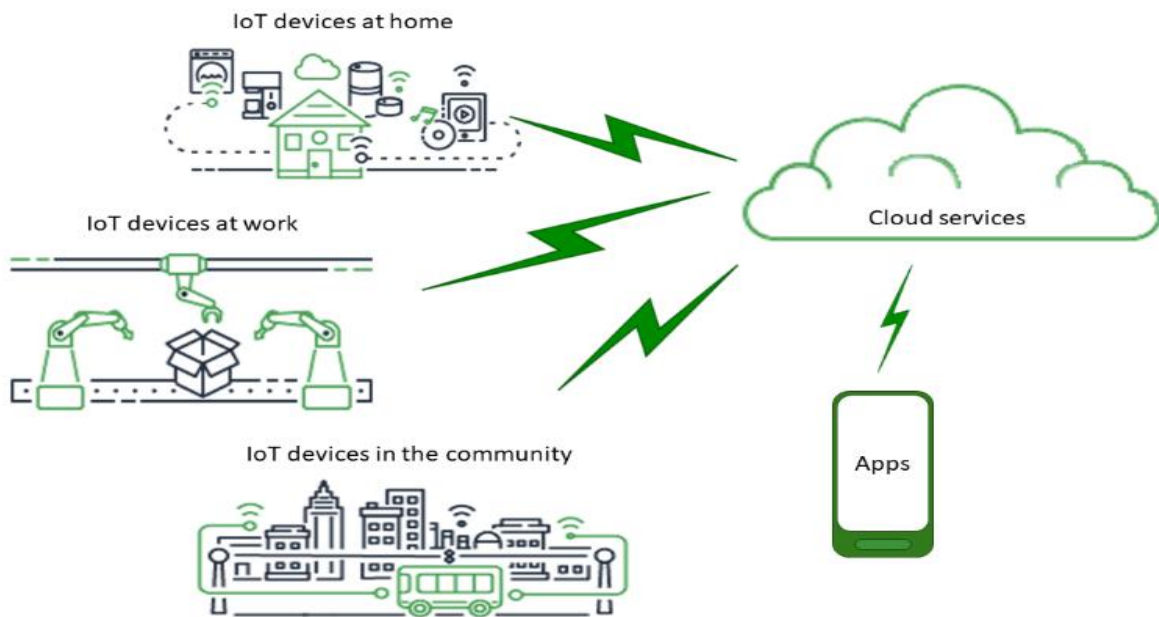


Figure 1.3 : Les applications d'internet des objets

Accessoires connectés : Les accessoires connectés sont peut-être l'aspect le plus visible de l'IoT pour l'utilisateur de tous les jours. Il s'agit notamment des moniteurs d'activité physique, des montres intelligentes, des lunettes intelligentes, des casques de réalité virtuelle, etc.

Maisons intelligentes : Un système de maison intelligente se connecte aux appareils pour automatiser des tâches particulières et est généralement contrôlé à distance. Parmi les exemples d'appareils de l'IoT dans les maisons intelligentes, on peut citer les appareils de cuisine sans fil, les systèmes de musique à détection d'humeur, l'éclairage intelligent, les stores motorisés, les portes et fenêtres automatisées, les compteurs électriques intelligents, etc.

Villes intelligentes : Les villes intelligentes utilisent des appareils de l'IoT, comme des capteurs et des compteurs connectés, pour collecter et analyser les données. Ces données peuvent être utilisées pour améliorer les infrastructures, les services publics ainsi que d'autres services.

Voitures à conduite autonome : Les voitures à conduite autonome sont généralement équipées d'un système technologique basé sur l'IoT qui partage des informations concernant le véhicule lui-même et la route qu'il emprunte. Les données relatives au trafic, à la navigation, à l'environnement extérieur et bien plus encore sont collectées et analysées par les systèmes informatiques de la voiture pour lui permettre de conduire de façon autonome.

Vente au détail : L'Internet des objets est de plus en plus utilisé dans la vente au détail. Pensez, par exemple, aux caisses automatisées, aux remises personnalisées, aux étagères intelligentes (qui avertissent le détaillant lorsqu'elles sont en rupture de stock), aux employés robotisés et à la gestion optimisée de la chaîne d'approvisionnement. Est un exemple de l'IoT qui comble le fossé entre le commerce en ligne et le commerce traditionnel. Les magasins sont sans argent liquide puisque l'argent a déduit des portefeuilles Amazon des clients. Ils ajoutent également des articles aux paniers des clients en temps réel lorsque ceux-ci choisissent des produits dans les rayons.

Télésanté : La télésanté, parfois appelée télémédecine, désigne la prestation de services de santé au moyen de la technologie. L'IoT est un aspect essentiel de la télésanté (l'acronyme IdOM est parfois utilisé pour désigner l'Internet des objets médicaux). Par exemple, le diagnostic médical à distance, la communication numérique de l'imagerie médicale, les consultations vidéo avec des spécialistes, etc.

L'agriculture intelligente : L'agriculture intelligente implique l'utilisation de la technologie numérique pour améliorer les efforts agricoles. Par exemple, les agriculteurs peuvent utiliser des capteurs, des caméras ainsi que d'autres appareils connectés pour améliorer la vue d'ensemble de leur exploitation et adapter les opérations pour améliorer leur rendement.

Cette liste n'est pas exhaustive. Dans de nombreux secteurs, l'IoT transforme les opérations et les méthodes de travail. Parmi les appareils de l'IoT, citons les téléphones mobiles intelligents, les réfrigérateurs intelligents, les montres intelligentes, les moniteurs d'activité physique, les alarmes d'incendie intelligentes, les serrures de porte intelligentes, les vélos intelligents, les capteurs médicaux, les systèmes de sécurité intelligents ainsi que les assistants virtuels, comme Alexa et Google Home, et la liste est encore longue. [6]

1.10 Le M2M

1.10.1 Définition

Le M2M (Machine-to-Machine) constitue un ensemble de technologies réseaux sans fil ou filaires rendant des systèmes communiquant et leur permettant de s'échanger automatiquement des informations, sans intervention humaine. Il existe deux grandes familles de technologies, à savoir le sans-fil et les filaires. La plus connue des technologies sans-fil est celle du réseau Internet mobile de type 2G, 3G, 4G mais il en existe d'autres, basées sur des radiofréquences

différentes allant de la plus courte portée avec le NFC, à la plus longue, avec le Wireless M-bus par exemple. Du côté des technologies M2M filaires, on trouve le courant porteur en ligne. [5]

1.10.2 Différence entre M2M et IoT

Souvent désignés par leurs initiales la machine to machine (M2M) et l'Internet des Objets (IoT) sont deux concepts très proches. Il existe néanmoins une différence essentielle à retenir. Nous vous l'expliquons dans les lignes qui suivent. Le M2M et l'IoT sont deux concepts, deux phénomènes très importants et assez proches. Ce sont deux solutions qui proposent des accès à distances à des objets ou des capteurs. Le M2M se définit traditionnellement par un réseau de télécommunication point à point utilisant un module cellulaire ou WiFi intégré, pour connecter des machines ou des objets à un réseau. L'intervention humaine n'est pas nécessaire, les informations circulent d'un endroit à un autre et peuvent être relayées via un serveur vers un logiciel. L'IoT repose quant à lui sur l'identification de chaque objet, souvent connecté les uns aux autres, afin qu'il puisse envoyer des données sur une plateforme Cloud ou à une application à plus large échelle. [5]

1.11 Les Défis et les bénéfices de l'IoT

1.11.1 Les Défis de l'IoT

Il y a des défis à l'application de l'Internet des objets de point de vue coût de la mise en œuvre du fait que l'attente de la technologie doit être disponible à faible coût avec un grand nombre d'objets. De plus, l'IOT est également face à de nombreux autres défis qui se résument dans cinq sous sections suivantes :

1- La Découverte Automatique : Dans les environnements dynamiques les services appropriés pour les objets doivent être identifiés automatiquement. Ceci nécessite des moyens sémantiques appropriés pour décrire leur fonctionnalité afin de les exploiter d'une manière efficace.

2- L'Interopérabilité : Chaque type d'objets intelligents dans l'Internet des objets possède différentes capacités de stockage d'information, de traitement et de communication. Différents objets intelligents seraient également soumis à des conditions différentes telles que la disponibilité de l'énergie et les besoins en bande passante de communications. A cet effet, pour faciliter la communication et coopération de ces objets, des normes communes sont requises.

3- La Sécurité et La Confidentialité : Les aspects de sécurité et de protection d'Internet tel que la confidentialité des communications, l'intégrité des messages ainsi que l'authenticité et la fiabilité de partenaires de communication doivent être assurés dans un environnement IoT. A

titre d'exemple dans certain cas il y a un besoin d'accéder à certains services pour accomplir une tâche ou d'empêcher de communiquer avec d'autres objets du système IoT pour des raisons préventives de sécurité. De plus, d'autres exigences selon la particularité du domaine d'application, doivent être aussi considérées comme par exemple dans les transactions commerciales des objets intelligents doivent être protégés des yeux des concurrents.

4- La Tolérance Aux Pannes : Les objets dans IoT sont beaucoup plus dynamiques et mobiles que les ordinateurs Internet, et dans un changement rapide et de manière inattendue. Pour structurer Internet des objets d'une manière robuste et digne de confiance on utilise la redondance en plusieurs niveaux pour l'adaptation automatique au changement de conditions.

5- L'Auto-Organisation : Les objets intelligents ne devraient pas être gérés comme des ordinateurs qui nécessitent leurs utilisateurs pour les configurer et les adapter à des situations particulières. Les objets mobiles, qui sont souvent utilisées de façon sporadique, doivent établir spontanément des liens et pouvoir être organisé et configurer par eux-mêmes en fonction de leur environnement d'exécution [6].

1.11.2 Les bénéfices de l'IoT

L'Internet des objets promet ainsi une forte plus-value à chaque organisme. En connectant objets, personnes et environnements, il devient possible de développer des améliorations qui ne pourront être que bénéfiques.

Les principaux bénéfices que peut avoir une entreprise dotée d'un système IoT sont :

- **Amélioration de la productivité :** l'IoT permet la surveillance, le monitoring et le contrôle des différents process, ce qui optimise les différentes opérations qui augmentent la productivité et l'efficacité.
- **Analyses prédictives :** grâce à la collecte de nombreuses données, les nouvelles technologies de l'IoT permettent d'examiner les patrons récurrents et contribuent à l'analyse prédictive qui peut être principalement utilisée en maintenance. Ces informations précises vont servir à améliorer les process et les services existants.
- **Rapidité d'action :** les données permettent de suivre en temps réel et même à distance les systèmes mis en place. Elles facilitent l'optimisation des interventions de maintenance, mais aussi donnent un avantage stratégique à l'entreprise dans le suivi de l'évolution des marchés.
- **Diminution des erreurs humaines :** grâce à la complémentarité des technologies comme l'intelligence artificielle, l'IoT permet de minorer les erreurs humaines dues à des tâches mondaines ou répétitives.

L'IoT constitue ainsi un atout dans l'aide à la prise de décision et au rendement de l'entreprise. Il représente un avantage stratégique face au marché concurrentiel. C'est une tendance qui ne cessera de croître, puisqu'on estime qu'il y aura plus de 20 milliards d'objets connectés dans le monde d'ici 2020, Face à leur futur prometteur, il serait temps d'en profiter et commencer à transformer son business [6].

1.12 L'architecture de l'IoT

Une architecture technique est un cadre créé pour permettre aux concepteurs et aux développeurs de considérer le système dans son ensemble et de le décomposer en sections. Selon les normes mondiales 1 (GS1), « une architecture de référence est une fondation pour permettre l'intégration des diverses technologies dans les applications IoT »

Il existe de nombreuses propositions d'architectures pour l'IoT parmi les architectures qui ont été proposées, il y a l'architecture à cinq couches et à trois couches proposées par IEEE. En ce moment y'a pas une architecture universellement acceptée. [7]

A. L'architecture à trois couches : Cette architecture se compose de :

La couche application (application layer) : est chargée de fournir des services spécifiques à l'application à l'utilisateur. Il définit diverses applications dans lesquelles l'Internet des objets peut être déployé, par exemple, les maisons intelligentes, les villes intelligentes et la santé intelligente.

La couche réseau (abstract layer) : est responsable de la connexion à d'autres objets intelligents, périphériques réseau et serveurs. Ses fonctionnalités sont également utilisées pour la transmission et le traitement des données des capteurs.

La couche perception (things layer) : est la couche physique, qui possède des capteurs pour détecter et recueillir des informations sur l'environnement. Il détecte certains paramètres physiques ou identifie d'autres objets intelligents dans l'environnement.

B. L'architecture à cinq couches : Cette architecture à cinq couches se compose de :

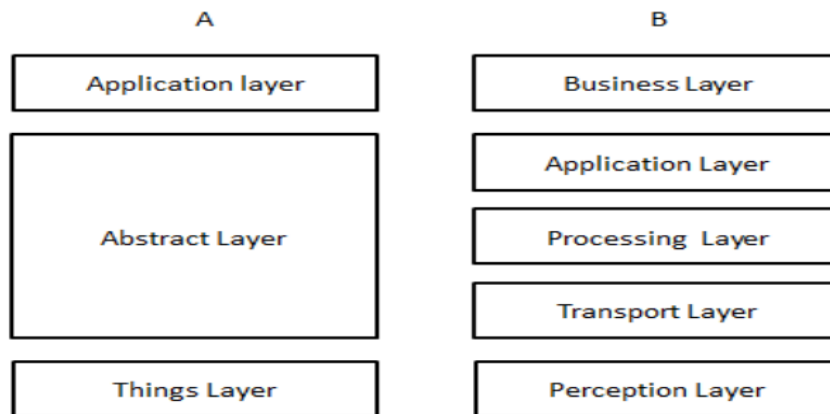
La couches de perception et d'application : leur rôle est le même que celui de l'architecture à trois couches.

La couche de transport (transport layer) : transfère les données du capteur de la couche de perception à la couche de traitement et vice versa via des réseaux tels que sans fil, 3G, LAN, Bluetooth, RFID et NFC.

La couche de traitement (processing layer) : est également appelée couche middleware. Il stocke, analyse et traite d'énormes quantités de données provenant de la couche transport. Il

peut gérer et fournir un ensemble diversifié de services aux couches inférieures. Il utilise de nombreuses technologies telles que les bases de données, le cloud computing et les modules de traitement des méga données.

La couche métier (business layer) : gère l'ensemble du système IoT, y compris les applications, les modèles commerciaux et de profit, et la confidentialité des utilisateurs. [7]



(A) Architecture à trois couches (B) Architecture à cinq couches

Figure 1.4 : Architectures IoT

1.13 Les protocoles de l'IoT

De nombreuses normes IOT sont proposées pour faciliter et simplifier les tâches des programmeurs d'applications et des fournisseurs de services. Différents groupes ont été créés pour fournir des protocoles, y compris les efforts menés par le W3C, IETF, EPC global, IEEE et l'ETSI. L'IoT ambitionne de faire communiquer chaque système avec tous autres au moyen de protocoles communs. La mise en application à une large échelle du concept d'IOT apparaît largement tributaire d'une standardisation de la communication entre objets dite M2M.

Au niveau de la couche de liaison : le standard IEEE 802.15.4 est plus adapté que l'Ethernet aux environnements industriels difficiles.

Au niveau réseau : le standard 6LoWPan a réussi à adapter le protocole IPV6 aux communications sans fil entre nœud à très faible consommation.

Au niveau routage : l'IETF a publié en 2011 le standard RPL.

Au niveau de la couche application : le protocole CoAP qui tente d'adapter HTTP, beaucoup trop gourmand aux contraintes des communications entre nœuds à faible consommation.

Le tableau suivant montre les différents protocoles IoT selon l'architecture à cinq couches : [7]

Tableau 1.1 Protocoles de chaque couche

Couches	Protocoles
Couche métier (buisness layer)	Gère l'ensemble du système IoT
Couche Application (Application layer)	<ul style="list-style-type: none"> • CoAP • MQTT • XMPP • AMQP • WEBSOCKET • DDS
La couche de traitement (processing layer/ middleware layer)	La couche middleware (traitement) est une couche logicielle interposée entre la couche de transmission (transport) et couche d'application, qui peut associer directement les services aux demandeurs correspondants et a un lien vers la base de données. La base de données, l'informatique omniprésente, le cloud computing et la prise de décision peuvent avoir lieu dans cette couche.
Couche Transport / Couche réseau (Transport layer/network layer)	<ul style="list-style-type: none"> • UDP : protocole de couche de transport simple pour les applications réseau client / serveur basé sur le protocole Internet (IP). UDP est souvent utilisé dans des applications spécialement conçues pour des performances en temps réel. • TCP : Protocole utilisé pour la majorité des connexions Internet. • DTLS : Le protocole DTLS assure la confidentialité des communications pour les protocoles de datagramme. Le protocole permet aux applications client / serveur de communiquer d'une manière conçue pour empêcher l'écoute clandestine. Le protocole DTLS est basé sur le protocole de transport Protocole Layer Security (TLS) et offre des garanties de sécurité équivalentes. <ul style="list-style-type: none"> • RPL : RPL (routage IPv6 pour les réseaux à faible puissance / perte) • 6LOWPAN : 6LoWPAN est un acronyme d'IPv6 sur les réseaux personnels sans fil à faible puissance. Il s'agit d'une couche d'adaptation pour IPv6 sur les liaisons IEEE802.15.4. Ce protocole fonctionne uniquement dans la gamme de fréquences 2,4 GHz avec un taux de transfert de 250 kbps.

Couche Perception	<ul style="list-style-type: none">• IEEE 802.15.4 : une norme qui spécifie la couche physique et le contrôle d'accès au support pour les réseaux personnels sans fil à bas débit (LR-WPAN). Il est géré par le groupe de travail IEEE 802.15.
-------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1.14 Conclusion

Dans ce chapitre nous avons définis l'internet des objets (L'IoT), Nous avons cité brièvement sur les technologies et leur architecture, par la suit nous avons parlé de la motivation et son fonctionnement et les différents protocoles de L'IoT.

Dans le prochain chapitre nous présentons les trois protocoles de communication de notre étude RPL, CoAP, et 6LoWPAN et nous allons expliquer leur fonctionnements, caractéristiques, architectures, ... ect.

Chapitre 2 :

Les protocoles CoAP, 6LoWPAN & RPL

Chapitre 2 : Les protocoles CoAP, 6LoWPAN & RPL

2.1 Introduction

Un certain nombre de travaux de recherche sont en cours sur les protocoles de communication les plus utilisés dans l'Internet des objets, qu'ils sont souvent utilisés par les périphériques à ressources limitées. Actuellement, trois protocoles de communication sont devenus populaires, et c'est pourquoi nous les avons choisis pour notre étude.

Dans ce chapitre on va expliquer en détail la définition, la structure, les caractéristiques de base et le fonctionnement de chaque protocole.

2.2 Le Protocole Contraintes Applicatif (CoAP)

2.2.1 Définition

CoAP (Constrained Application Protocol) est un protocole de transfert Web optimisé pour les périphériques et réseaux contraints utilisés dans les réseaux de capteurs sans fil pour former l'Internet des objets. Basé sur le style architectural REST, il permet de manipuler au travers d'un modèle d'interaction client-serveur les ressources des objets communicants et capteurs identifiées par des URI en s'appuyant sur l'échange de requêtes-réponses et méthodes similaires au protocole HTTP. L'utilisation des services web est courante sur les applications Internet. CoAP étend ce paradigme à l'Internet des objets et aux applications M2M qui peuvent ainsi être développés avec des services web RESTful partagés et réutilisables. Tout en prenant en compte les contraintes et besoins de l'Internet des objets tel que le support de l'asynchrone ou du multicast. CoAP est prévu pour devenir un protocole d'application omniprésent dans le futur Internet des objets. Le protocole CoAP se situe au niveau applicatif de la couche OSI et s'appuie sur UDP pour la communication. Il met en œuvre une méthode d'observation des ressources et fournit des fonctions de découverte des périphériques pour minimiser l'intervention humaine. Implémenté avec différents langages, ce protocole peut être utilisé dans des domaines tels que la santé ou la gestion énergétique. Il offre des performances adaptées aux objets à faibles ressources ainsi que la sécurité pour les données sensibles. [8]

2.2.2 Caractéristiques du protocole CoAP

Avec l'achèvement de la spécification CoAP, il est prévu que des millions de périphériques seront déployés dans différents domaines d'application. Ces applications vont de l'énergie intelligente, du réseau intelligent, de la gestion technique des bâtiments, de la gestion intelligente de l'éclairage, des systèmes de contrôle industriels, du suivi des actifs à la surveillance de l'environnement. CoAP deviendrait le protocole standard permettant l'interaction entre les périphériques et la prise en charge de l'IoT applications. Les environnements RESTful restreints (CoRE) sont le groupe de travail dans IETF qui conçoit le protocole CoAP. Les principales caractéristiques de COAP sont les suivantes : [8]

Un en-tête concis : Le protocole a un en-tête de base de seulement 4Ko et une taille totale de 10 à 20 Ko selon les requêtes (en ajoutant les options et les données)

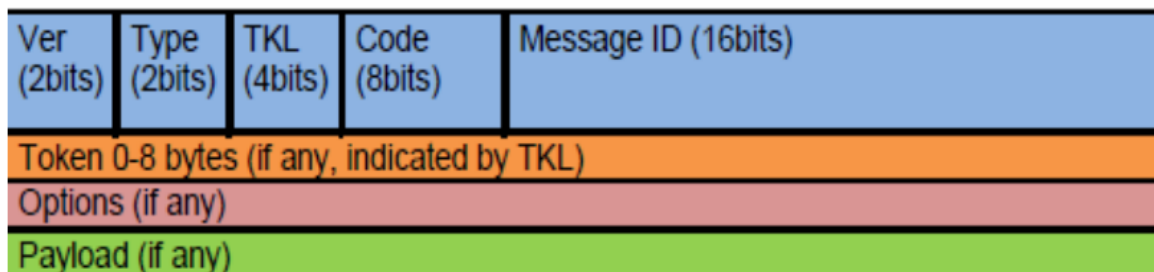


Figure 2.1 : En-tête du protocole COAP

Gestion des méthodes : COAP met à disposition les requêtes GET, PUT, POST et DELETE. Ces méthodes vont pouvoir être utilisées par le client pour accéder aux données. Leur utilisation est similaire à celle de HTTP.

Les URIs : Le protocole supporte les Uniform Resource Identifier ce qui va permettre de spécifier la cible grâce à un nom d'hôte, un port, un chemin et un paramètre de requête.

Le type de contenu : Le protocole permet de transporter des données d'usages différents en mettant dans l'entête le type de données de la charge utile.

La découverte de ressources : La découverte de ressources est un besoin clé dans une architecture web et notamment pour les applications M2M, cette recherche doit être autonome

et ne doit pas avoir besoin d'une interaction humaine. Les serveurs COAP doivent pouvoir fournir une URI à n que le serveur puisse être découvert.

2.2.3 Mode de fonctionnement

CoAP utilise un modèle client-serveur qui ressemble à celui de HTTP, où les clients envoient des requêtes sur des ressources REST pour récupérer de l'information d'un capteur ou contrôler un périphérique et son environnement. Il faut noter que CoAP traite les échanges de manière asynchrone au travers de datagrammes UDP. [7]

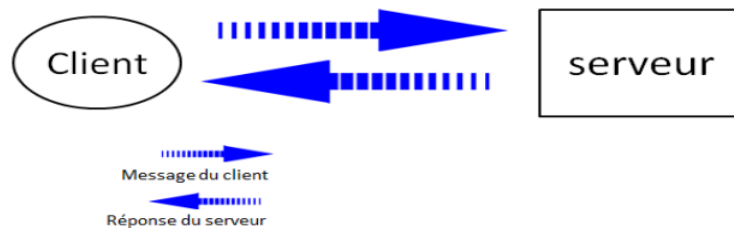


Figure 2.2 : Principe du fonctionnement du protocole CoAP

2.2.3.1 Composants

a. client : le client envoie une requête qui est similaire à une requête http pour demander une action.

b. serveur : répond au client par un code réponse.

c. message : les informations échangées entre le client et le serveur.

2.2.3.2 Fonctionnement du protocole COAP

CoAP s'appuie sur une approche à deux couches, une couche de messagerie CoAP où 4 messages sont définis et une couche d'interaction sous forme de requête/réponse héritée du protocole http.

1 Les Couches de La Protocole COAP : CoAP s'appuie sur une approche à deux couches.

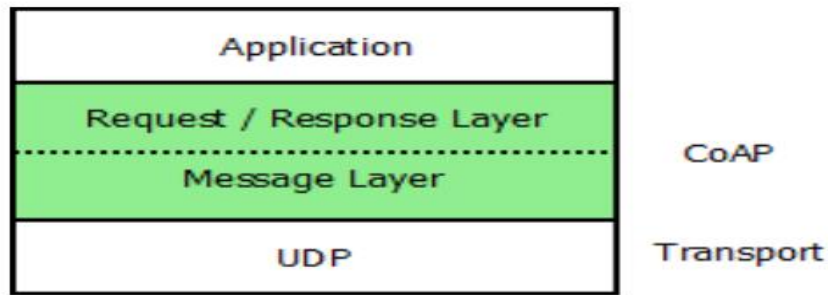


Figure 2.3 : Les deux couches de la protocole COAP

a- **La Couche de Message** : CoAP utilisée afin de traiter la non fiabilité d'UDP ainsi que la nature asynchrone des interactions (4 messages sont définis CON, ACK, NON, RST).

Tableau 2.1 : Messages de la couche messagerie CoAP

Message	Description
CON	Confirmable: Le message requiert une réponse qui peut être véhiculée par un message d'acknowledgment ou envoyé de façon asynchrone si la demande ne peut être traitée immédiatement. Peut-être aussi bien une requête qu'une réponse qui doit être acquittée.
ACK	<i>Acknowledgment</i> : Le message confirme la réception d'un message <i>Confirmable</i> .
NON	<i>Non-confirmable</i> : Le message ne requiert aucune réponse ou acquittement.
RST	<i>Reset</i> : Dans le cas où le message n'a pu être traité.

- **La sémantique de la couche message** : une requête est transportée dans un message CON (Confirmable) ou NON (Non-confirmable) et si la réponse est disponible immédiatement dans le cas d'un message CON, elle est transportée dans un message Acknowledgment (ACK). Si le message ACK est perdu, l'émetteur du message CON le retransmettra.

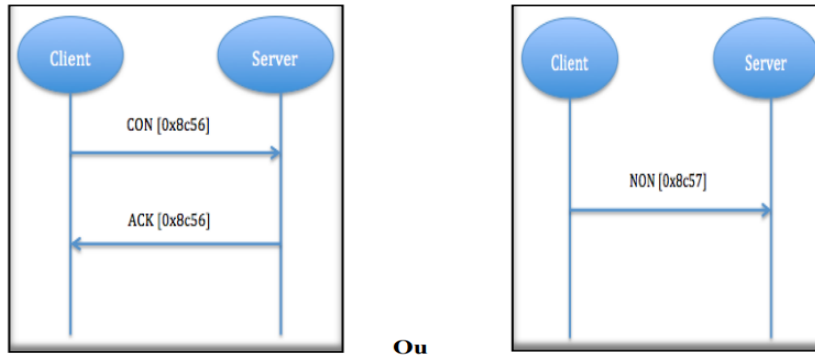


Figure 2.4 : La Sémantique de la Couche du Messagerie

b- La Couche requête et réponse (d'interaction) : sous forme de requête/réponse héritée du protocole HTTP (Requêtes GET, POST, PUT, DELETE et catégories de réponses).

- **La sémantique des requêtes et réponses COAP** est transportée dans des messages COAP qui incluent soit un code de méthode (GET, POST, DELETE, PUT) ou un code de réponse (e.g., 404, 205, etc.). Un jeton (token) est utilisé pour associer une requête à une réponse indépendamment des messages qui les transportent. Le Token est indépendant de l'identificateur de message.

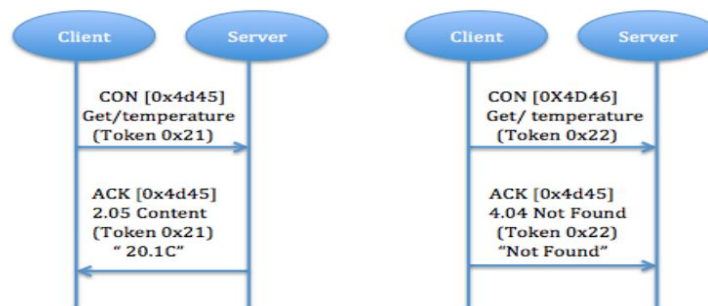


Figure 2.5 : La Sémantique des requêtes et réponses

2.2.4 Segmentation d'un message CoAP

CoAP s'appuie sur UDP ou DTLS, ce qui limite la taille maximum des représentations des ressources qui peuvent être transférées sans fragmentation. Même si UDP supporte des tailles de segments de données pouvant atteindre 64000 octets en considérant qu'IP assurera la fragmentation, cela ne fonctionne pas réellement pour des applications contraintes comme celles relatives à M2M. Plutôt que de s'appuyer sur la fragmentation IP, CoAP propose ses

propres mécanismes de fragmentation via le transfert des représentations des ressources en utilisant plusieurs blocs de requête/réponse. Cette option permet au serveur de fonctionner sans état, sans établissement de connexion et sans avoir à mémoriser les précédents transferts de bloc. L'option bloc permet donc de manière minimaliste un transfert de représentation de ressource important.

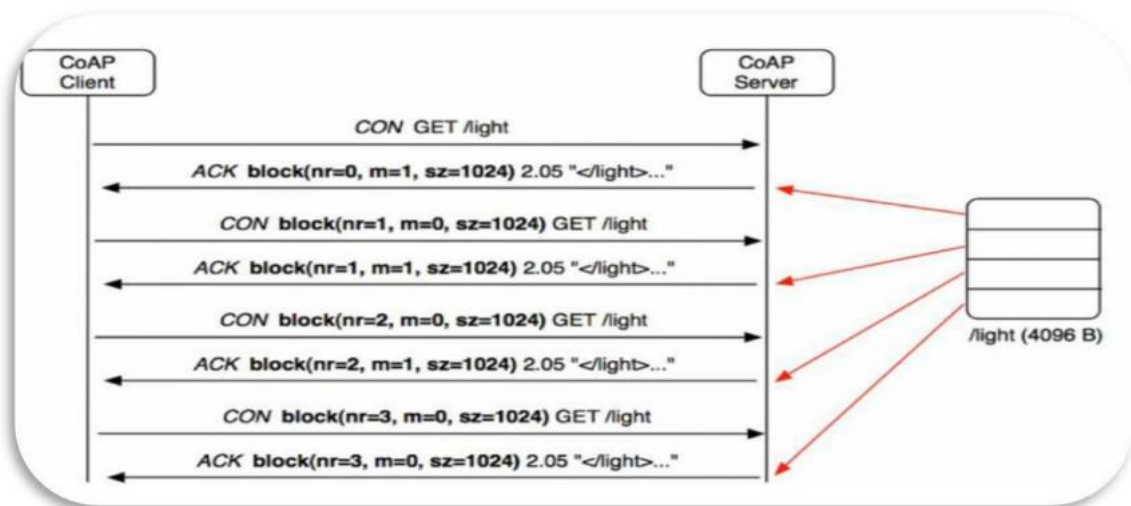


Figure 2.6: segmentation de Message COAP

2.3 6LoWPAN (IPv6 Over Low-Power Wireless Personal Area Network)

2.3.1 Définition

Les réseaux IPv6 personnels sans fil de faible puissance (6LoWPAN) est un groupe de travail créé par l'IETF afin de permettre l'adoption de IPv6 par les réseaux LLN. En effet, puisque l'adoption de l'IPv6 par les réseaux LLN a fait très vite l'unanimité à cause des milliards d'objets potentiels à connecter à internet, un autre problème s'est posé, celui de la taille des trames très petites définies par les protocoles de communications des couches sous-jacentes. Si on prend pour exemple la norme IEEE802.15.4, la taille des trames est uniquement de 127 octets alors que les paquets IPv6 ont une taille minimale de 1280 octet. Il est donc nécessaire

de définir une couche d'adaptation afin de permettre au paquet IPv6 de transiter à travers des trames de taille beaucoup plus réduites. C'est l'IETF qui a défini la couche adaptative 6LoWPAN afin de résoudre cette incompatibilité. 6LoWPAN se base principalement sur deux mécanismes afin de réduire la taille des datagrammes IPv6 à savoir la fragmentation et la compression des entêtes afin de permettre aux paquets IPv6 d'être envoyés ou reçus via les réseaux LLN. [9]

2.3.2 Architecture 6LoWPAN

Le réseau 6LoWPAN est composé par des nœuds partageant le même préfixe IPv6 et de routeur de bord qui s'occupe de la gestion de la compression et la fragmentation d'en-têtes IPv6. On distingue trois familles d'architecture dans un réseau 6LoWPAN, réseau 6LoWPAN simple avec un seul routeur de bord (Edge router) connecte à d'autre réseau IP, réseau 6LoWPAN étendu avec plusieurs routeurs de bord reliant par une dorsale et réseau ad hoc 6LoWPAN non connecté. [9]

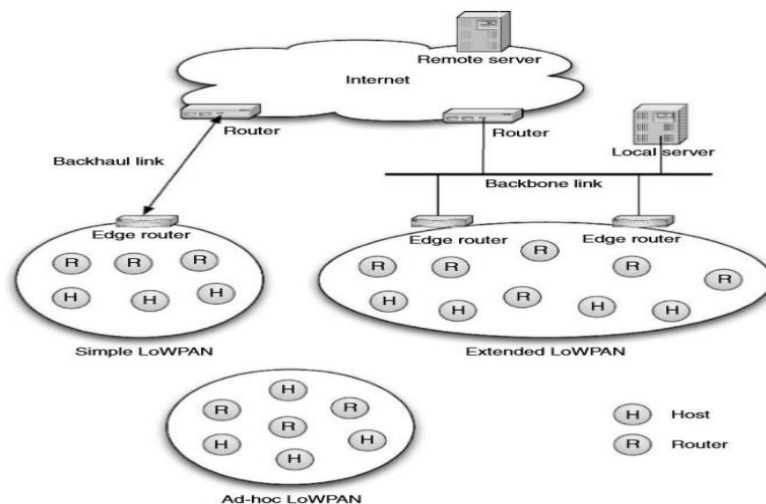


Figure 2.7 : Architecture 6LoWPAN

2.3.3 Fonctionnement de 6LoWPAN

6LoWAPN c'est une couche d'adaptation situé entre la couche réseau et la couche liaison de données charge de réduire la taille des paquets IPv6 en utilisant la fragmentation et la compression.

1) Compression d'en-tête : La couche d'adaptation 6LoWPAN se charge de la compression des en-tête IPv6 dont la taille est de 40 octets et des en-têtes UDP de 8 octets venus de la couche réseau et les envoie vers la couche de liaison.

On peut distinguer trois scénarios de communication possibles afin d'illustrer l'intérêt d'un tel mécanisme :

Premier scénario : La communication entre deux périphériques à l'intérieur du même réseau 6LoWPAN, en utilisant des adresses locales de liaison, l'en-tête IPv6 peut être compressée à seulement 2 octets. C'est le scénario du meilleur cas.

Second scénario : La communication destinée à un périphérique en dehors du réseau 6LoWPAN et le préfixe pour le réseau externe est connu, où l'en-tête IPv6 peut être compressé à 12 octets.

Troisième scénario : Similaire au second scénario, mais cette fois sans connaître le préfixe du périphérique externe, ce qui donne un en-tête IPv6 de 20 octets scénario du pire cas.

On constate que même dans le pire cas, on arrive quand même à réduire la taille de l'en-tête IPv6 de moitié. Si malgré ce mécanisme la taille du datagramme reste supérieure à celui des trames un autre mécanisme est proposé celui de la fragmentation.

2) Fragmentation et réassemblage : Afin de réduire la longueur des datagramme IPv6, la couche 6LoWPAN fragmente les paquets venus de la couche réseau et les envoie sous forme des trames de petites tailles (trames 802.15.4 dans le cas du standard IEEE802.15.4) à son équivalent sur l'équipement distant qui se charge de les réassembler.

Chaque fragment est précédé d'un en-tête de fragmentation. Bien que nécessaire, un impact plus ou moins important selon le mode de routage utilisé. [9]

2.3.4 Le routage dans le 6LoWPAN

Le routage dans le 6LoWPAN est réalisé selon deux modes : mesh-under et route-over.

1) Mesh-under : Dans le Mesh-under, la couche réseau n'effectue aucun routage IP à l'intérieur d'un LoWPAN. La couche d'adaptation exécute le relayage de nœud en nœud sans solliciter la couche IP et achemine les paquets vers la destination sur plusieurs sauts radio donc la décision

de routage se fait au niveau de la couche d'adaptation. Si un paquet IP est fragmenté par la couche d'adaptation les fragments sont livrés au saut suivant par routage maillé. Différents fragments d'un paquet IP peuvent passer par différents chemins et ils sont rassemblés à la destination. Dans ce cas, le paquet IPv6 n'est reconstitué que sur l'équipement destinataire. Si tous les fragments sont atteints avec succès, alors la couche d'adaptation du nœud de destination rassemble tous les fragments et crée un paquet IP. Dans le cas où un fragment quelconque manque, tous les fragments pour ce paquet IP sont retransmis vers la destination.

2) Route-over : Dans le mode route-over, toutes les décisions de routage sont prises dans la couche réseau où chaque nœud agit en tant que routeur IP. Lorsqu'un paquet IP est fragmenté par la couche d'adaptation, les fragments sont envoyés au saut suivant sur la base des informations de la table de routage. Si tous les fragments sont reçus avec succès, la couche d'adaptation crée un paquet IP à partir de fragments et l'envoie à la couche réseau. Si le paquet est destiné à lui-même, la couche réseau envoie le paquet IP à la couche de transport, sinon envoie le paquet au saut suivant sur la base des informations de la table de routage. Dans ce mode de routage, les paquets sont reconstitués sur chaque équipement intermédiaire afin de prendre la décision de routage s'il y a un ou plusieurs fragments manquants, tous les fragments sont retransmis à une distance de houblon. Route-over est plus efficace dans des conditions dégradées (perte de paquets). [9]

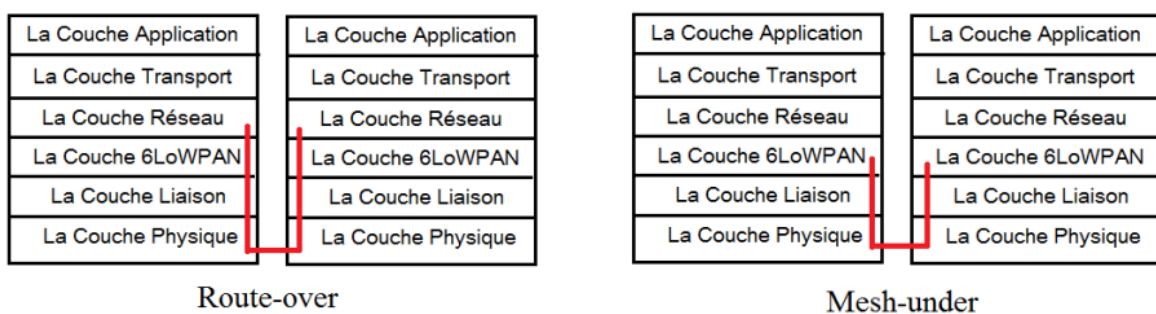


Figure 2.8 : les deux modes de routage Route-over Mesh-under

2.4 RPL : le protocole de routage des réseaux LLN

2.4.1 Définition

RPL est sans doute l'un des protocoles de routage IPv6 les plus connus pour les réseaux à faible consommation et à perte (LLN) développé par ROLL dans la RFC 6550 pour répondre aux

limites des réseaux LLN telles que la faible puissance de traitement, de batterie et de mémoire. RPL vise principalement les réseaux de collecte, où les nœuds envoient périodiquement des mesures à un point de collecte. Le protocole a été conçu pour être très adapté aux conditions du réseau et pour fournir des itinéraires de rechange, chaque fois que les itinéraires par défaut sont inaccessibles. RPL fournit un mécanisme pour diffuser l'information sur la nouvelle topologie de réseau formée dynamiquement.

2.4.2 Fonctionnement du protocole RPL

1) **Les graphes DAG et DODAG** : DAG (graphe orienté acyclique) est un graphe orienté qui ne possède pas de circuit. Il décrit les liens orientés entre les nœuds, se terminant à un ou plusieurs nœuds racines. RPL s'appuie sur la notion de DODAG (graphe acyclique orienté vers la destination), DODAG est un DAG à une seule destination à la racine c'est-à-dire à une seule racine DAG.

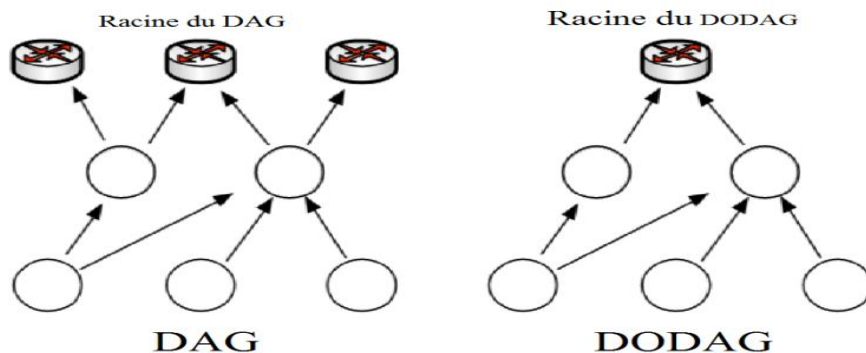


Figure 2.9 : les graphes DAG et DODAG

2) **Le modèle réseau** : RPL définit trois types de nœuds :

- **DODAG root ou Low power and lossy Border Router (LBR)** : Il se réfère à la racine d'un DODAG qui représente un point de collecte dans le réseau et a la capacité de construire un DAG. Le LBR agit également comme une passerelle (routeur edge) entre l'internet et le LLN, il peut avoir des racines multiples configurées dans le réseau.
- **Nœud Routeur (ou statefull node)** : Il se réfère à un périphérique qui peut transmettre et génère du trafic des voisins. Un tel routeur n'a pas la capacité de créer un nouveau DAG.

- Nœud host (ou stateless node) : Il se réfère à un périphérique terminal capable de générer des données, mais qui n'est pas capable de router le trafic.

3) Messages de contrôle dans RPL : Il existe principalement quatre types de message de contrôle décrits comme suit :

1. DIO (Objet d'Information DODAG) : contient des informations permettant à un nœud de découvrir une instance RPL apprendre ses paramètres de configuration de calculé son rang et de choisir des parents qui minimisent le coût sur la route vers la racine du DODAG.

2. DIS (Sollicitation Information DODAG) : pour solliciter un DIO de leurs voisins.

3. DAO (Destination Annonce Objet) : pour propager les informations de destination en remontant le DODAG le message DAO est unicast de l'enfant au parent.

4. DAO-ACK : Le message DAO- ACK est envoyé en tant que paquet unicast par un destinataire DAO en réponse à un message DAO unicast.

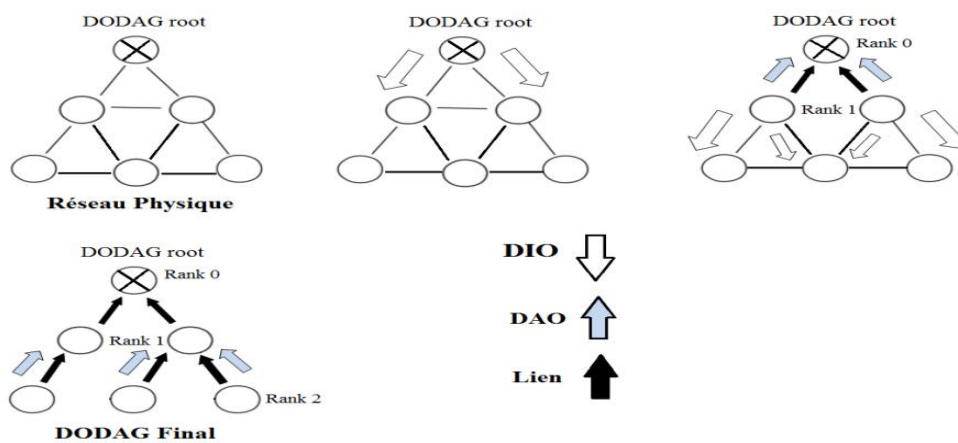


Figure 2.10 : L'envoi des messages DIO et DAO et la construction de DODAG

2.4.3 Les modes d'opération du protocole RPL

En fonction de la capacité des nœuds en termes de mémoire et de la taille éventuelle du réseau, le protocole RPL offre deux modes de fonctionnement, le fonctionnement en « storing mode » et celui en « Non-Storing mode ».

- **Le fonctionnement en « Non-Storing mode »** : Dans ce mode, seul la racine est en mesure de stocker des informations de routage. Les autres nœuds du réseau conservent uniquement les adresses de leur parent direct. Toutes les informations sur la structuration du DODAG sont transmises à la racine dans les messages DAO. En cas de besoin de router des données vers une destination quelconque, les nœuds transmettent ces données à la racine en passant par leur parent. La racine effectuera un routage à la source vers la bonne destination.

- **Le fonctionnement en « Storing mode »** : Dans ce mode, les nœuds intermédiaires sont en mesure de garder en mémoire des informations de routage puis de rediriger les données reçues vers la bonne destination en consultant les informations du routage. Contrairement au « Non Storing mode », dans le fonctionnement en « Storing mode », les messages DAO ne sont pas tous transmis à la racine. Chaque nœud transmet son message à son parent direct (parent à un saut) qui maintient une table de routage à son niveau. [9]

2.4.4 Les paradigmes de communication

RPL prend en charge trois paradigmes de communication (MP2P), (P2MP) et (P2P). Dans ce qui suit, nous détaillons le fonctionnement de ces modèles de communication.

- **Multipoint à point (MP2P)** : RPL a été conçu principalement pour optimiser le type de flux de trafic multipoint à point (MP2P), cette communication MP2P a été fournie par les constructions des routes à partir de chaque nœud vers la racine DODAG à l'aide de DIO du parent préféré d'un nœud il s'agit de « Routes ascendantes » (Upward Routes). Les destinations des flux MP2P sont des nœuds désignés qui ont une certaine importance pour l'application, tels que la fourniture de connectivité à l'Internet ou au réseau IP privé principal.

- **Point à multipoint (P2PM)** : Il s'agit de Routes descendantes (Downward Routes), RPL prend en charge le trafic P2MP il utilise un mécanisme de publicité de destination qui prévoit des itinéraires descendants de la racine vers d'autres nœuds (préfixes, adresses ou groupes de multidiffusion). Par exemple les messages DIO, P2PM est le modèle de trafic requis par plusieurs applications LLN ([RFC5867], [RFC5826], [RFC5673] et [RFC5548]).

- **Point à point (P2P)** : Pour le trafic P2P, la construction des routes ça dépend de mode de fonctionnement du protocole RPL. Si le cas de mode Non-Storing mode le paquet dirige vers une racine, ensuite la racine effectuera le routage vers la destination, si le cas de mode Storing mode, le paquet s'écoule vers la racine jusqu'à ce qu'il atteigne un ancêtre qui a une route connue

vers la destination. Cet ancêtre commun peut être la racine DODAG. Dans d'autres cas, il peut s'agir d'un nœud plus proche de la source ou de la destination. [9]

2.5 Conclusion

Dans ce chapitre on a montré les différentes caractéristiques de protocoles CoAP, 6LoWPAN et RPL. Nous avons expliqué comment ces trois protocoles fonctionnent. Dans le prochain chapitre nous allons opter à la simulation des trois protocoles pour comparer et évaluer ces performances.

Chapitre 3 :

Simulation des protocoles

Chapitre 3 : Simulation des protocoles

3.1 Introduction

Dans ce chapitre nous allons décrire le processus de la réalisation de notre simulation de chaque protocole, ainsi que les outils qui ont été utilisés, et une présentation de quelques interfaces de notre simulation.

3.2 Simulation

Une simulation d'événements discrets implique l'examen d'une impression spécifique du modèle d'un système et la duplication de son comportement. L'avantage de la simulation est qu'elle fournit une méthode très large qui permet d'étudier n'importe quel modèle tant que l'outil de simulation s'adapte au modèle étudié. En revanche, il présente l'inconvénient de nécessiter un temps de calcul machine important. Il existe différents simulateurs d'événements disponibles. Citons les simulateurs de réseau NS-3, ainsi que OMNeT, qui permettent de simuler différents types de réseaux, dont les réseaux d'attente, OPNET, et COOJA, un outil d'analyse des performances.

Le simulateur de réseau par défaut pour Contiki, COOJA, a été préinstallé avec Contiki 3.0. COOJA fournit une interface conviviale qui permet une configuration et une analyse rapides de la simulation. Parce que notre sujet nécessite une analyse des performances, COOJA a été identifié comme l'un des meilleurs outils pour simuler le protocole en raison de sa flexibilité, de son extensibilité et de sa rapidité de prototypage.

3.3 Environnement de travail et outils de simulation

Pour notre simulation on a choisi le simulateur COOJA car non seulement fournit des outils qui permettent de diffuser des données de chaque test très facilement et dans un format lisible et simple mais aussi il fournit une grande quantité de code d'exemple qui peut être utilisé.

3.3.1 Outils matériels

Nous avons utilisé :

- Un laptop HP.
- RAM : 8, 00 Go.

- Type de system : 64 bits.
- System d'exploitation : Windows 10

3.3.2 Outils logiciels

- **VMware Workstation** : VMware Infrastructure est un package complet de virtualisation d'infrastructure qui combine la virtualisation, la gestion, l'optimisation des ressources, la disponibilité des applications et l'automatisation opérationnelle. VMware Infrastructure virtualise et combine les ressources matérielles physiques sous-jacentes sur plusieurs systèmes, fournissant des pools de ressources virtuelles au centre de données dans un environnement virtuel.
- **Contiki-OS** : Contiki est un système d'exploitation open source pour de minuscules microcontrôleurs basse consommation qui permet aux développeurs de créer des programmes qui utilisent efficacement le matériel tout en fournissant une connectivité sans fil basse consommation standardisée pour une variété de plates-formes matérielles. Il est utilisé dans une large gamme de systèmes commerciaux et non commerciaux, y compris la surveillance sonore de la ville, les lampadaires, les compteurs électriques en réseau, la surveillance industrielle, la surveillance des rayonnements, la surveillance des chantiers, les systèmes d'alarme et la surveillance à distance des maisons.
- **COOJA** : C'est un simulateur polyvalent basé sur Java qui vous permet d'écrire des logiciels d'application en langage C en utilisant l'interface native Java. L'un des principaux avantages de notre simulateur COOJA est qu'il peut simuler simultanément des logiciels d'application dans le développement d'algorithmes de haut niveau et le développement de pilotes durs de bas niveau. Le simulateur COOJA est extrêmement adaptable, ou les pièces actuelles peuvent être modifiées. Nous pouvons exécuter une simulation de variation avec des conditions et des paramètres système variés, tels que différents taux de création de paquets, différents protocoles MAC et différentes topologies de réseau, en utilisant les avantages de COOJA.
- **Wireshark** : Wireshark est un outil open source c'est un analyseur de réseau et de protocole de réseau. Il est anciennement connu sous le nom d'Ethereal, peut être utilisé pour examiner les détails du trafic à divers niveaux allant des informations au niveau de la connexion aux bits qui composent un seul paquet. La capture de paquets peut fournir à un administrateur réseau des informations sur des paquets individuels.

3.4 Les étapes de la simulation

Étape 1 : Installation logicielle : Téléchargement et extraction du package contiki-os à l'emplacement : home/user/contiki sur le poste de travail VMware workstation

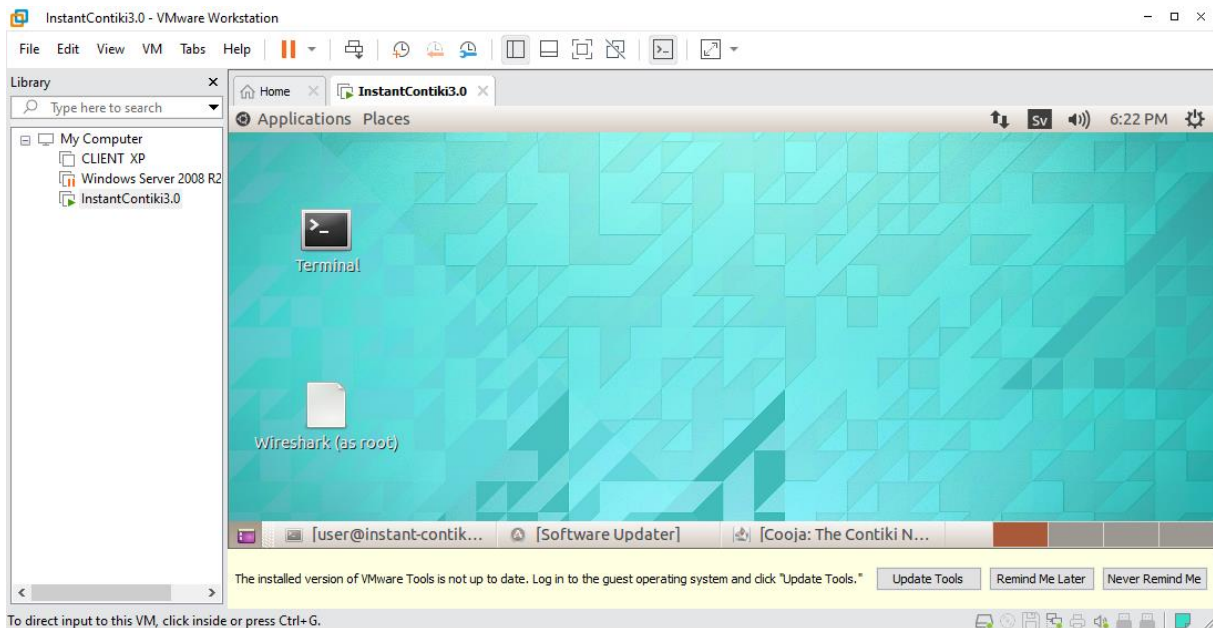


Figure 3.1 : Interface des logicielles

Étape 2 : Exécution de simulateur Cooja :

- La manière simple d'exécuter Cooja est de l'exécuter dans son propre répertoire :

```
cd contiki/tools/cooja >> ant run
```

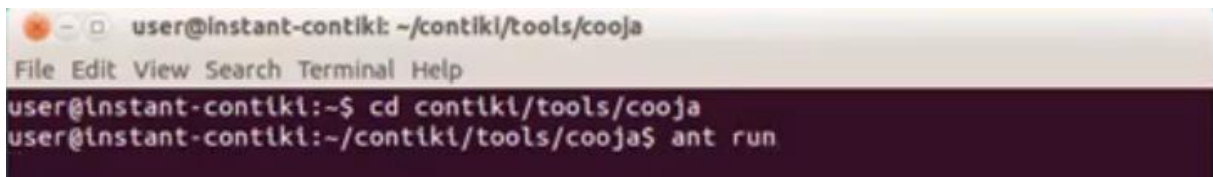


Figure 3.2 : Exécution de Cooja

- Après l'exécution de Cooja, la fenêtre suivante apparaît :

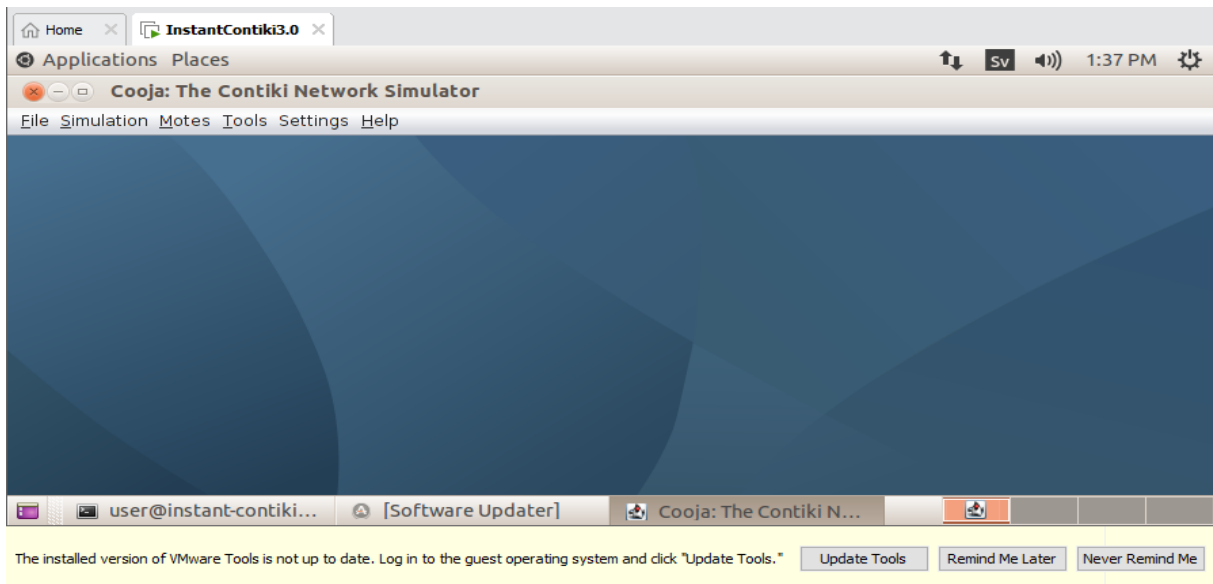


Figure 3.3 : Interface du simulateur COOJA

Étape 3 : Création d'une nouvelle simulation : Dans le menu il faut choisir :

File> New simulation. Après il faut choisir un nom de la simulation ensuite Diverses fenêtres apparaîtront pour la simulation, comme la fenêtre du réseau, fenêtre de contrôle de simulation, sortie de mote et chronologie.

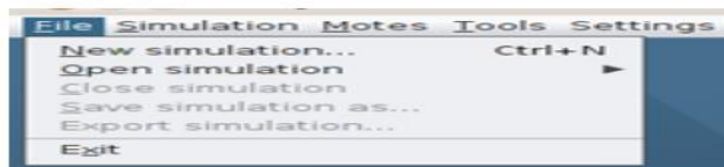


Figure 3.4 : Création d'une nouvelle simulation



Figure 3.5 : Les fenêtres de l'Interface du simulateur COOJA

Fenêtres de l'interface du simulateur COOJA : Nous avons de nombreuses fenêtres dans une simulation : [7]

- La fenêtre Réseau : en haut à gauche de l'écran, nous montre tous les nœuds du réseau simulé.
- La fenêtre de contrôle Simulation : c'est là que nous pouvons démarrer, mettre en pause et charger notre simulation.
- La fenêtre Mote Output : sur le côté droit de l'écran, nous montre toutes les impressions du port série de tous les nœuds et tous les échanges de messages entrent dans les motes.
- La fenêtre Notes : en haut à droite est l'endroit où nous pouvons mettre des notes pour notre simulation.
- La fenêtre Timeline : en bas de l'écran, affiche tous les événements de communication dans la simulation au fil du temps, très pratique pour comprendre ce qui se passe dans le réseau. [8]

3.4.1 Simulation du protocole CoAP

- Une fois que la nouvelle simulation est créée et nommée, il faut créer les motes nous on a choisi des motes de type « Sky Mote » pour le routeur, le serveur et les clients. [7]

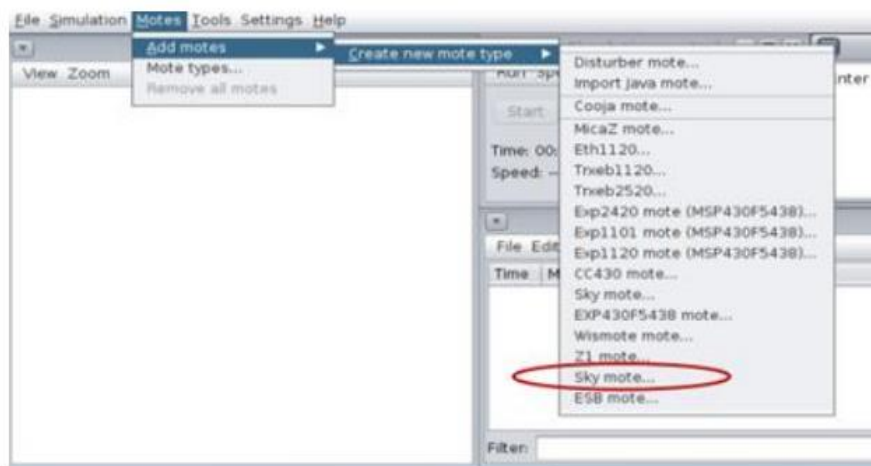


Figure 3.6 : Création des motes

- Après avoir choisi le type de mote une fenêtre apparaît qui permet de donner un ID aux motes et de choisir le fichier .C (en cliquant sur Browse) qui sera compilé comme la compilation d'un programme C et cela va permettre de simuler les motes.

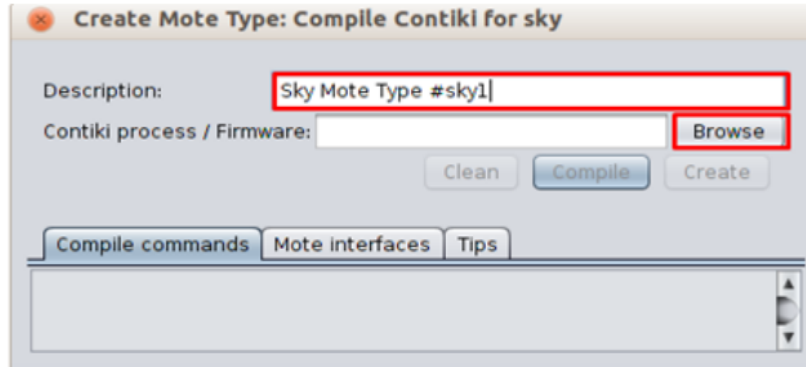


Figure 3.7 : Fenêtre de compilation

A. Le routeur : Nous lui avons donné l’ID « border router » et on a sélectionné le fichier border-router.c puis on a compilé (en cliquant sur Compile) une fois que la compilation est finie on a choisi de créer notre mote (Create) et une autre fenêtre pour donner le nombre de mote qu’on souhaite créer et leur position est apparue on a choisi 1 et enfin on a ajouté notre routeur.

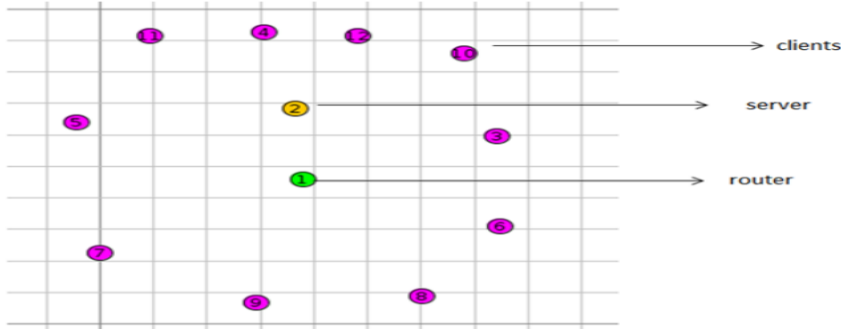


Figure 3.8 : Ajout des notes

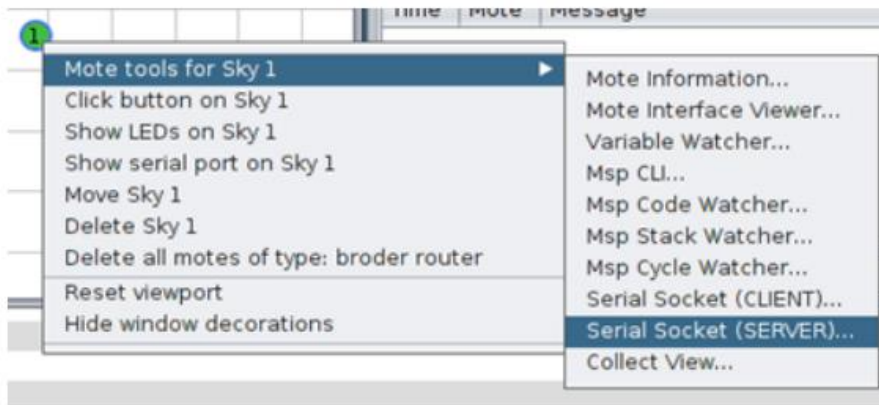
B. Le serveur : Même chose que le routeur sauf que on lui a donné l’ID « Server » et on a sélectionné le fichier er-example-server.c

C. Les clients : On a nommé le sky mote pour les clients « client » et on a choisi le fichier er-example-clients.c

➤ Voilà le résultat de création des notes :



- Une fois les nœuds créés il faut allumer le routeur comme suit :



- Après une fenêtre avec l'option de démarrage s'affiche :



Figure 3.9 : Fenêtre de serial socket server

- Pour démarrer la simulation et permettre au client de se connecter il faut cliquer sur (start) dans la fenêtre de serial socket server et la fenêtre de contrôle de simulation



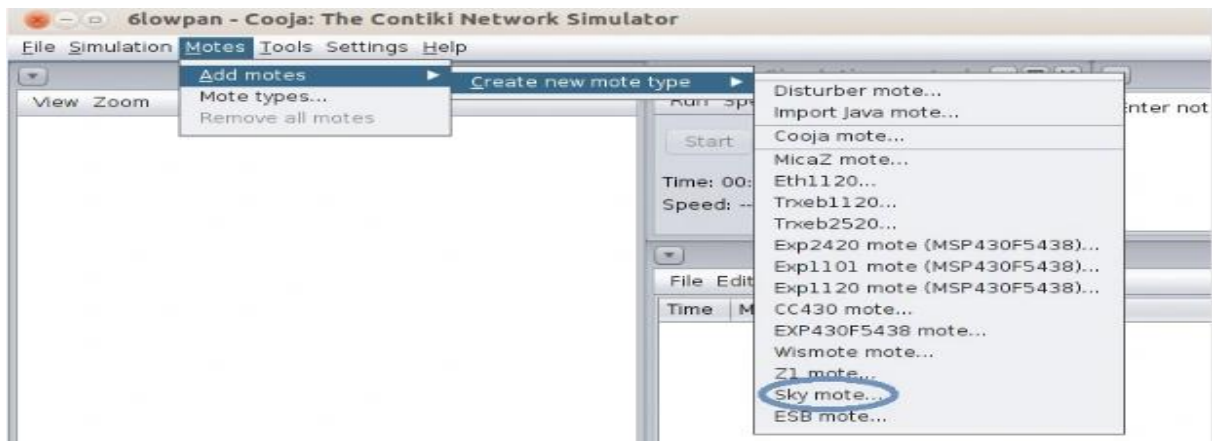
Figure 3.10 : Commencement de la simulation

- Il faut aussi exécuter la commande "make connect-router-cooja" :

```
user@instant-contiki:~/contiki$ cd examples/ipv6/rpl-border-router
user@instant-contiki:~/contiki/examples/ipv6/rpl-border-router$ make connect-router-cooja
```

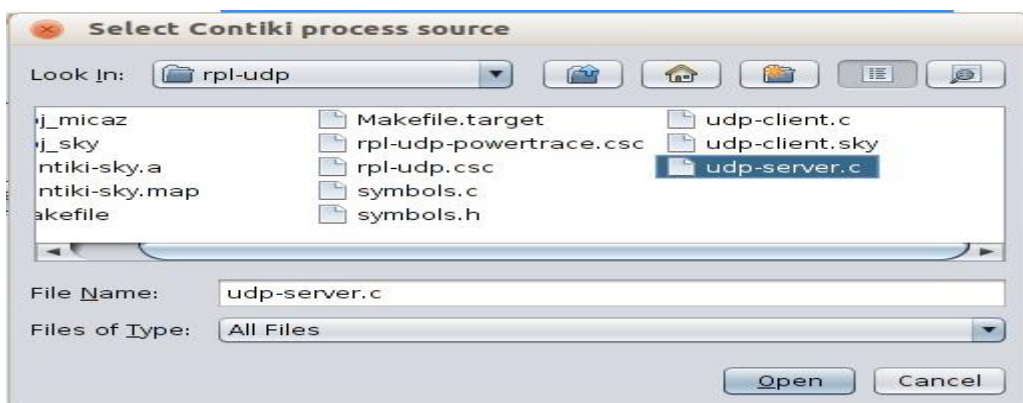
3.4.2 Simulation du protocole 6lowpan

Pour créer l'infrastructure 6LowPAN, il est nécessaire de construire un routeur frontalier réseau qui est capable de se connecter à l'hommage de lui-même. Pour simuler ce protocole et trouver le protocole d'efficacité, le sky mote est utilisé.

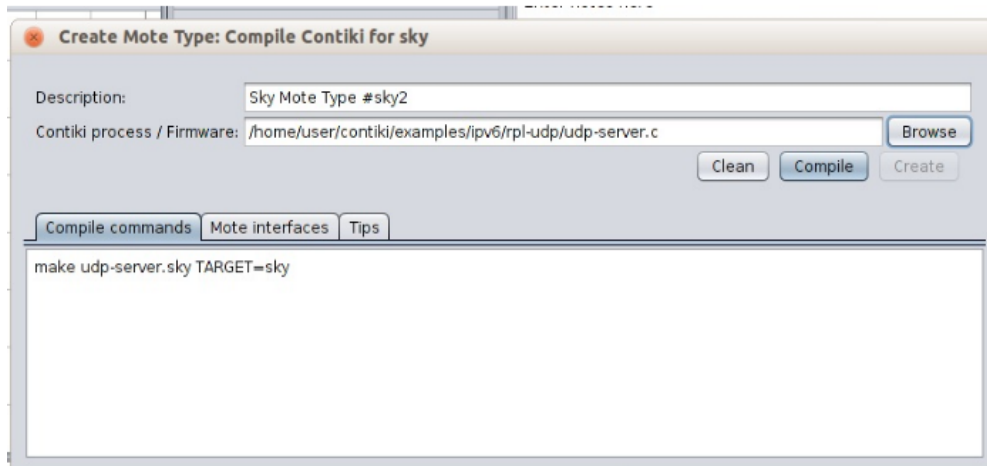


Avant la simulation de Cooja, un pont entre la simulation et l'ordinateur réel socket pour avoir un tunnel de communication vers l'extérieur est créé. Le programme tunslip6 disponible dans le stockage contiki est exécuté. Une configuration locale de Cooja est nécessaire pour créer le tunnel de paquets pour envoyer des paquets 6LowPAN. Pour cela, une socket série (SERVER) et un port qui l'écoute sur 60001 est créé. [10]

Fenêtre suivante Apparaît Cliquez sur Parcourir et localiser **examples/IPv6/rpl-udp/udp-server.c**



Appuyer sur **compile**. Attendre la fin, et appuyer sur **create**.



Appuyez sur le bouton **start** dans la fenêtre de contrôle de simulation.

Et la fenêtre de **radio message** affiche le trafic. Cela signifie clairement que notre réseau prend en charge 6LoWPAN dans sa couche réseau.

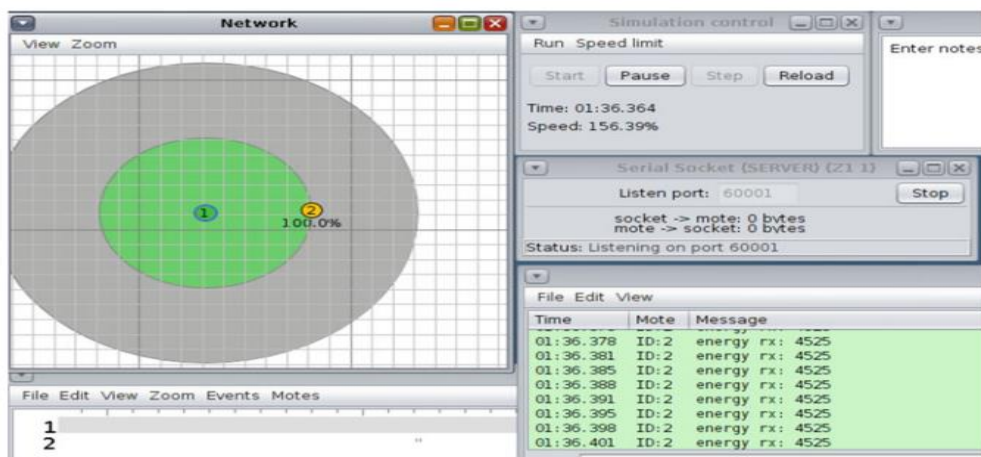


Figure 3.11 : Simulation de 6LoWPAN

3.4.3 Simulation du protocole RPL

- Pour démarrer l'application, il suffit de lancer la commande suivante : `ant run`
- Puis, on va créer une nouvelle simulation à partir du menu File / New simulation
- **Ajouter un nœud racine**

Motes / Add Motes / Create New Motes

Type / Choisir le type de mote

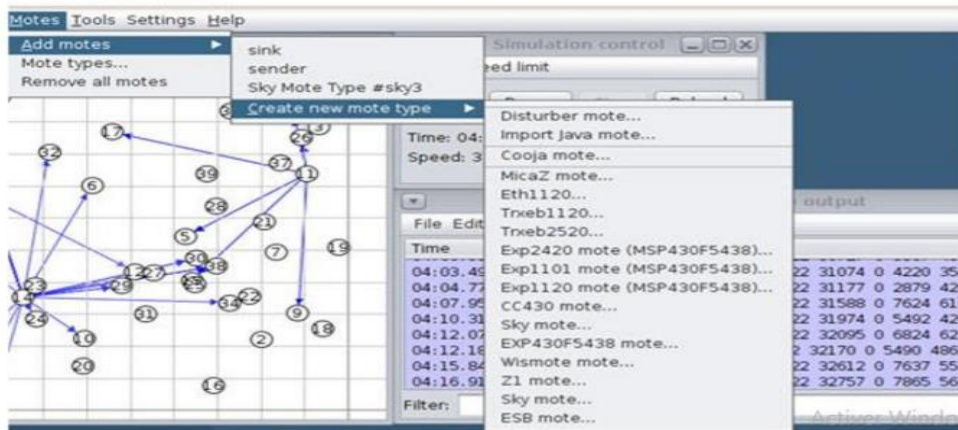


Figure 3.12 : Simulation de RPL

On ajoute une mote de type « Skymote », on implémente le code source udpsink qui est implémenté dans ... (ajouter le chemin), on a essayé de compiler le code pour voir s'il génère des erreurs, on peut faire Clear pour générer le code de nouveau parce que si le code est déjà compilé on remarque que le simulateur ne fait pas la compilation du code et surtout si on le modifie, par la suite on ajoute le nombre de nœuds souhaité, dans notre cas on a ajouté un seul nœud puisqu'il représente la racine.

- **Ajouter des nœuds Sender :** On ajoute une mote de type « Skymote » (même type avec le nœud de la racine), puis on ajoute le code source udpsender de rplcollect, on a suivi les mêmes étapes comme on a fait pour la création du nœud de la racine, mais cette fois un nombre de nœuds différents pour qu'on pourrait faire une interprétation pour étudier la consommation énergétique et le nombre de message envoyés et le nombre de message reçus. [11]

3.5 Les paramètres estimés et les utiles utilisés

- **L'énergie :** Nous étudions ici la consommation énergétique et les performances des objets par rapport à l'énergie utilisée et consommée tout au long de la simulation. Pour faire cela nous avons un outil qui PowerTracker, permettre de connaître le taux et la durée d'utilisation des fonctions radio des objets.



Figure 3.13 : Utilisation du PowerTracker

- **Transmission et réception** : On peut calculer le nombre total des messages envoyés/réceptionnés grâce à l'outil : "Mote output" du simulateur Cooja.

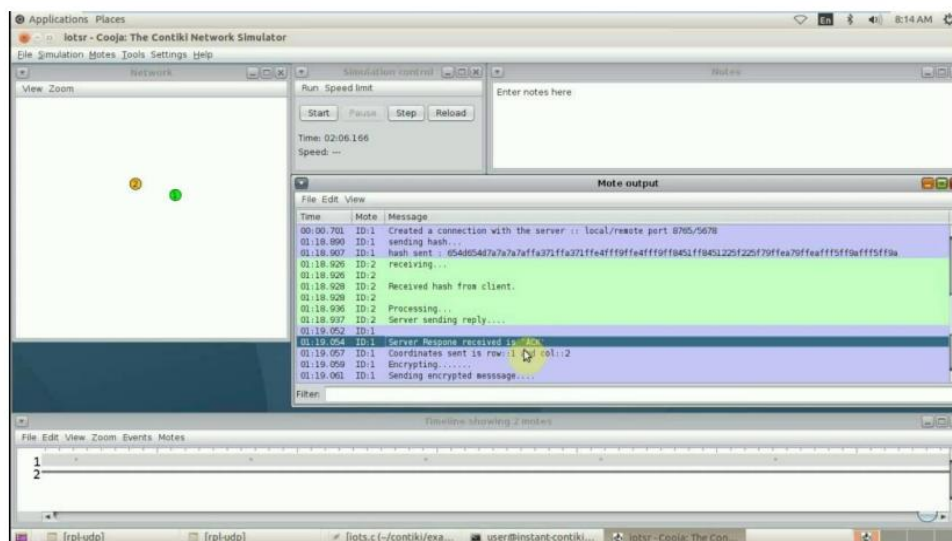


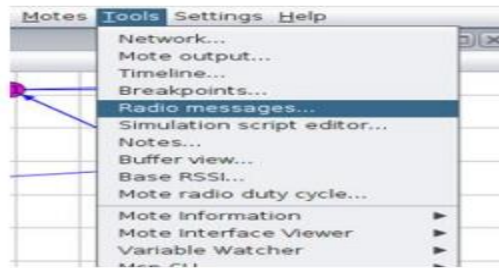
Figure 3.14 : Utilisation de l'outil "Mote output"

- **Les paquets** : Nous devons démarrer le logiciel wireshark qui est disponible par défaut dans VMplayer qui rend la simulation est beaucoup plus facile à exécuter car le wireshark capture directement les paquets entre les motes à travers le fichier PCAP généré pendant la simulation. [12]

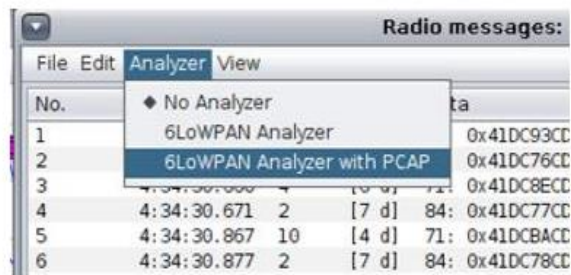
Il possible d'utiliser Wireshark comme suite : [7]

1. **Installer Wireshark** : Enregistrer le trafic réseau sous forme de fichier à l'aide de Cooja

Simulator : Sélectionner « Radio message »



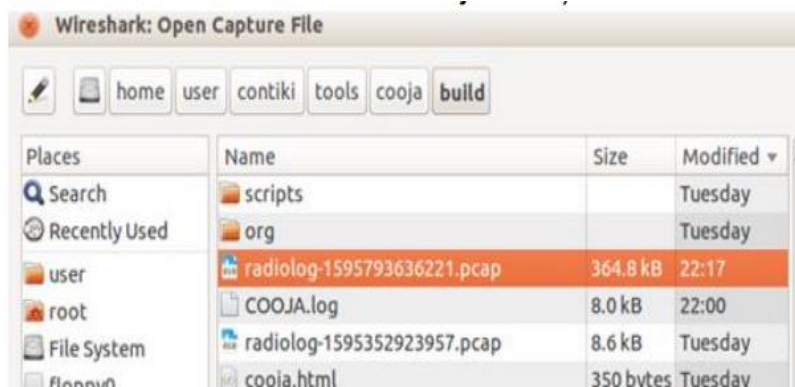
2. Sélectionner : Analyser >6LoWPAN Analyzer with PCAP



3. Ouvrir, numériser et exporter des fichiers PCAP au format CSV en utilisant le Programme Wireshark : Dans le menu « file » il faut sélectionner : File>Open



4. Sélectionner le fichier PCAP : (/home/user/contiki/tools/cooja/build)



Voici un exemple de l'interface de Wireshark :

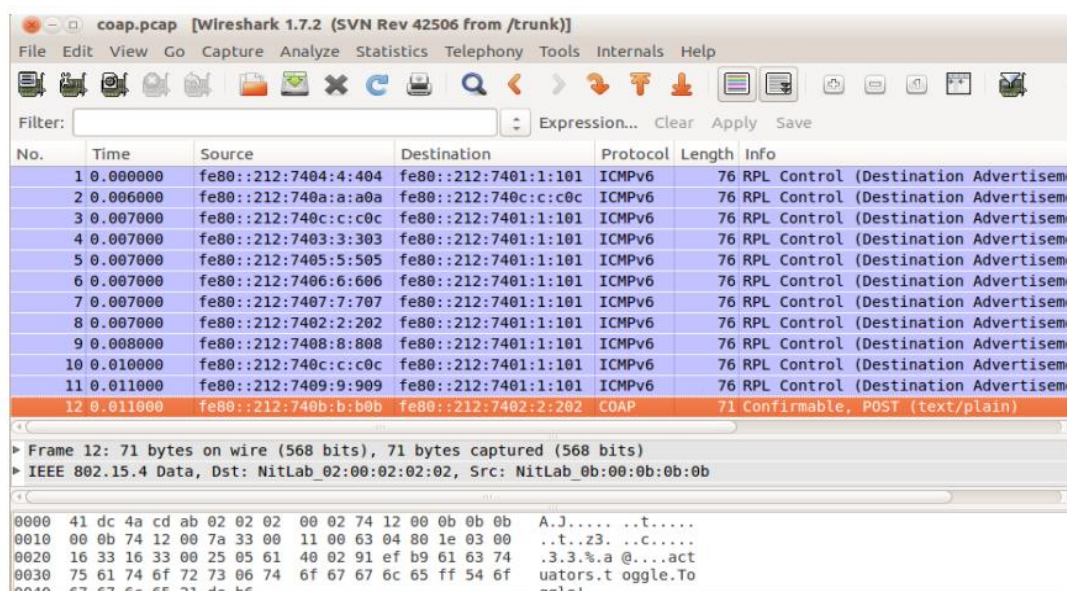


Figure 3.15 : Interface Wireshark

3.6 Conclusion

Ce chapitre a été consacré à la simulation de protocoles. Pour cette simulation nous avons utilisé l'outil de simulation COOJA qui contient différentes méthodes et outils qui nous ont permis d'obtenir à des résultats concernant les performances des protocoles : en termes de la consommation d'énergie, le taux de messages transférées et cela va nous indiquer l'utilisation de la bande passante, et le retard dans les protocoles. On a fini dans le prochain chapitre par évaluer ses performances selon les résultats de simulation.

Chapitre 4 :

Comparaison et évaluation de performances des protocoles

Chapitre 4 : Comparaison et évaluation de performances des protocoles

4.1 Introduction

Ce chapitre présente, l'évaluation de performances et analyse de protocoles basé sur des valeurs obtenues à partir du logiciel contiki. Pour notre cas nous allons évaluer les protocoles en termes de la consommation d'énergie, le taux de messages transférées et cela va nous indiquer l'utilisation de la bande passante. Influence de la perte de paquets sur le transfert de données, afin de calculer et suivre ces différentes caractéristiques nous avons utilisé quelques méthodes que cooja propose.

4.2 Analyse Les résultats de simulation et évaluation de performances des protocoles

Les performances des protocoles 6LoWPAN, CoAP et RPL seront comparées en fonction de différents critères comme le paquet reçu, le nombre de nœuds, le temps et le nombre de nœuds sur la plate-forme cooja. Cooja est utilisé pour simuler les nœuds contiki qui consomment peu de ressources comme l'énergie et la bande passante pour communiquer entre eux sur le système d'exploitation Ubuntu. Actuellement, ce simulateur est très populaire parmi les chercheurs du réseau puisque le comportement précis du système peut être examiné, et des simulations de grands réseaux sont possibles avant leur intégration dans des dispositifs. [13]

4.2.1 Comparaison selon la consommation d'énergie

Le protocole CoAP est considéré comme l'un des plus protocoles importants conçus spécifiquement pour les capteurs appareils avec des ressources limitées, en particulier en puissance, de ses fonctionnalités est utilisé (UDP) au lieu de (TCP) et petite taille de tête et plus, technologie 6lowpan a fait possible de connecter les appareils et les petites choses à l'Internet, la technologie IPv6 à travers 6LoWPAN et le Protocole de routage IPv6 pour les réseaux de faible puissance et de perte (RPL) ont fait des réseaux de capteurs sans fil (WSNs) intégré avec des objets intelligents sur Internet.

Nous avons réalisé une simulation de huit nœuds. La Fig. 4.1 montre une comparaison de la consommation moyenne d'énergie pour les trois protocoles (CoAP, 6lowpan et RPL), montrant la supériorité du protocole CoAP sur le reste des protocoles. [14]

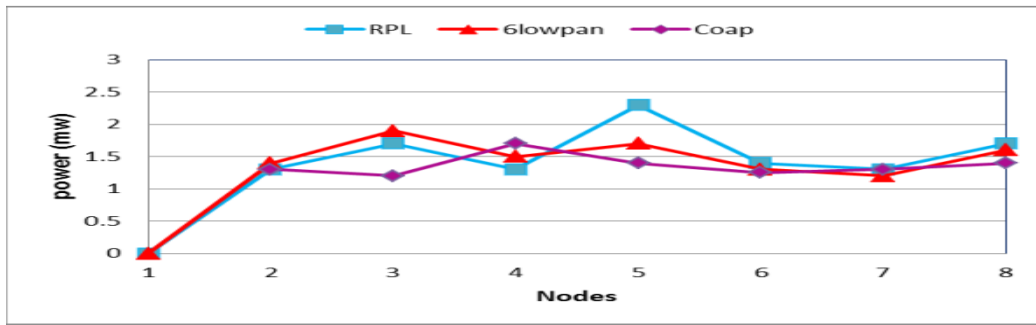


Figure 4.1 : Comparaison de la consommation d'énergie pour (CoAP, 6lowpan et RPL)

4.2.2 Comparaison selon l'économisation d'énergie

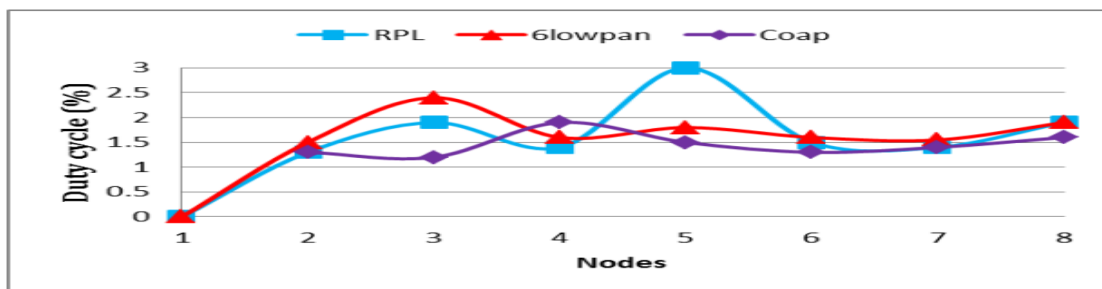


Figure 4.2 : Cycle d'utilisation moyen pour (CoAP, 6lowpan et RPL)

En ce qui concerne le cycle d'utilisation moyen, la Fig. 4.2 montre une supériorité du protocole CoAP. Cette technique (duty cycle) fait dormir les nœuds quand ils n'ont pas besoin d'être données envoyées ou reçues et réveil pour toute nouvelle opération, le but est d'économiser l'énergie. [14]

4.2.3 Comparaison selon les paquets reçus

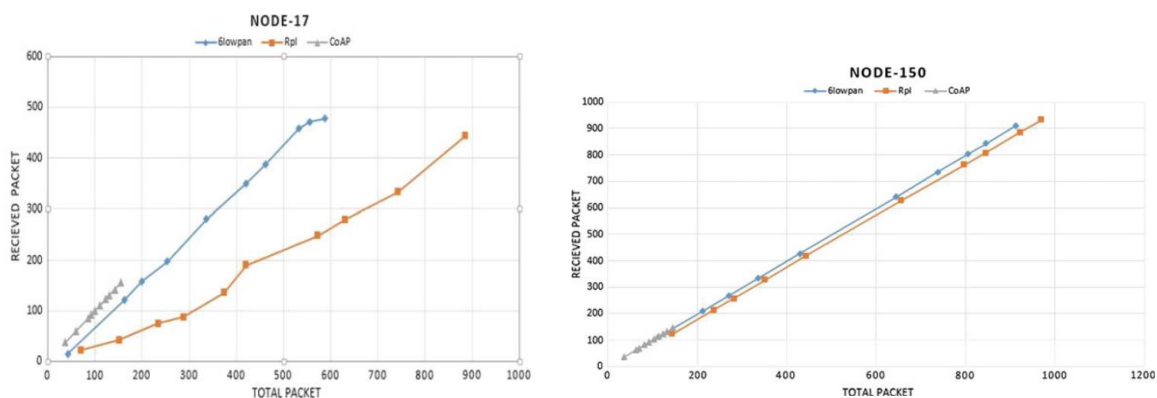


Figure 4.3 : Comparaison selon les paquets reçus

Comme la montre la Fig 4.3, l'axe X indique le nombre de paquets totaux, et l'axe Y indique le nombre de paquets reçus.

- Avec 17 nœuds. Le 6LoWPAN reçu plus de paquets que RPL et CoAP. Par exemple, 6LoWPAN a reçu 425 et RPL reçu 220 paquets sur 500. Cependant, CoAP a cessé de recevoir des paquets après un certain point.

- Le nombre de nœuds a été augmenté à 150 de 17 et a des résultats similaires ont été observés, la performance de RPL améliorée cependant. On peut remarquer que le 6LoWPAN reçu presque tous les paquets envoyés. [13]

4.2.4 Comparaison selon le taux de transmission par paquets

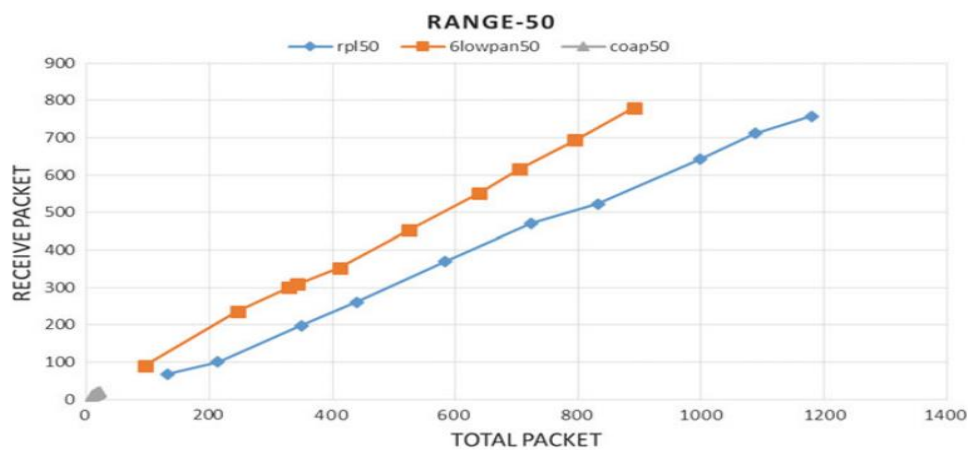


Figure 4.4 : Comparaison selon le taux de transmission par paquets

Le taux de transmission a été augmentée de 10 à 50 m. On peut le voir que le taux de transmission de CoAP est très lent. Mais le taux de transmission par paquets de 6LoWPAN est meilleur que le taux de transmission par paquets de RPL. [13]

4.3 Conclusion

Dans ce chapitre, nous avons estimé et analysé performances de RPL, 6LoWPAN et CoAP dans l'Internet des objets dans diverses situations. Les résultats de la démontrer les performances de RPL, 6LoWPAN et CoAP sont :

- Sur la base de la consommation d'énergie ou du temps moyen entre les paquets, COAP produit un résultat légèrement supérieur à 6LOWPAN et RPL.
- Le ratio de paquet reçu et transmis est très bon dans CoAP mais le problème est que le temps pris par la transmission et la réception de paquets est très élevé.
- Démarrer de recevoir un paquet dans un scénario différent est bon dans RPL mais après un certain temps lorsque la charge augmente, la performance du protocole RPL est en baisse.
- Globalement la performance du protocole 6LoWPAN est inégale dans ce scénario.

Ce résultat montre que le taux de transfert de paquets CoAP est juste mais lent. Mais sur l'autre main, le taux de perte de paquets RPL est très élevé avec une communication rapide. Après l'analyse, on peut dire que 6LoWPAN fonctionne mieux que RPL et CoAP.

Conclusion générale

Conclusion générale

IoT est l'un des thèmes clés dans le domaine actuel de la recherche en réseau depuis petites appareils qui sont intégrés dans IoT sont connectés via l'Internet.

La communication constante entre les objets se fait par de multiples protocoles qui offrent des performances qui permettent à l'IoT d'avoir plus d'impact et d'être plus performant. Parmi les protocoles les plus utilisés pour la communication entre objets dans l'IoT, les protocoles 6LoWPAN, RPL et CoAP qui sont parmi les protocoles les plus connus de l'IoT et c'est pour cela que nous avons voulu évaluer leurs performances et comprendre leur fonctionnement, afin de réaliser une comparaison entre eux.

L'objectif principal de notre travail est d'analyser et montrer l'efficacité de ces trois protocoles, cela à l'aide d'une étude par simulation en utilisant le simulateur COOJA, pour l'objectif d'aider les utilisateurs à mieux choisir le protocole qui convient le mieux à leurs projets et qui répond à leurs besoins.

Nous avons montré les performances de chaque protocole selon la consommation d'énergie, le taux de messages transférées et l'influence de la perte de paquets sur le transfert de données. Ce travail nous a permis de :

- Comprendre quelque aspect de l'IoT et les différents domaines de cette dernière.
- Acquérir des connaissances sur la simulation et le fonctionnement du simulateur COOJA.
- Connaitre les différents protocoles de l'IoT et avoir une idée sur les différences qui existent entre eux.
- Mieux comprendre le fonctionnement de 6LoWPAN, RPL et CoAP.
- Réalisé une comparaison entre les trois protocoles.

Comme résultat de ce travail, nous avons conclu que le protocole 6LoWPAN est le plus performant grâce aux résultats de comparaison, mais cela est dans la seule condition qui est la non influence, dans l'application IoT, de la grande consommation d'énergie. On peut dire aussi que le CoAP est le mieux utilisé si la consommation d'énergie est un critère important dans une application IoT.

Comme perspective nous espérons pouvoir vérifier les autres caractéristiques par simulateur et faire une vérification formelle aux trois protocoles et proposer une amélioration pour qu'ils soient plus performants et nous espérons que nous continuerons de cette manière pour l'étude des autres protocoles de l'IoT.

Bibliographie

- [1] Saleh, I. (2018). Internet des Objets (IdO) : Concepts, Enjeux, Défis et Perspectives. *Internet Des Objets*, 2(1). <https://doi.org/10.21494/iste.op.2018.0229>.
- [2] portices.fr, (2023), *Internet of Things (IoT) : Qu'est-ce que l'internet des objets ?*, URL: <https://www.portices.fr/internet-of-thing-iot>, visité le : 20/02/2023.
- [3] Ovhcloud, (2023). *Qu'est-ce que l'internet des objets?*, URL: <https://www.ovhcloud.com/fr/learn/what-is-iot/>, visité le : 22/02/2023.
- [4] wikipedia, (2023). *Internet des objets*. URL: https://fr.wikipedia.org/wiki/Internet_des_objets/, visité le : 24/02/2023.
- [5] Ramarosaona, M., (2017). *CONCEPTION ET CREATION D'UNE PLATEFORME IOT AVEC LE PROTOCOLE MQTT*. Mémoire de fin d'étude, université d'Antananarivo.
- [6] Gheribi, A., Kermezli, A., & Rebhi, R. (2022). Conception et réalisation d'un système de contrôle et supervision de la température d'un four géré par ESP8266 et Protocol MQTT, mémoire de fin d'étude, université de Médéa.
- [7] BENCHOUA, R., & HANACHI, N. E. *Etude comparative de protocoles de communication dans l'IOT*, Doctoral dissertation, université de Ouargla.
- [8] OUADAH, A. (2019). Modélisation et vérification formelle d'un protocole CoAP pour l'internet des objets. Mémoire de fin d'étude, université de M'sila.
- [9] Mahmud, A., Hossain, F., Choity, T.A., Juhin, F. (2020). Simulation and Comparison of RPL, 6LoWPAN, and CoAP Protocols Using Cooja Simulator. In: Uddin, M., Bansal, J. (eds) Proceedings of International Joint Conference on Computational Intelligence. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-13-7564-4_28.
- [10] Rana, M. Rapport Sur Les RCSF, <Http://e-biblio.univ-mosta.dz/bitstream/handle/123456789/6254/MINF189.pdf>.
- [11] DHAHOUA, OUMAIMA, (2019). Modélisation et vérification d'un protocole de communication pour l'internet des objets. mémoire de fin d'étude, université de Biskra.
- [12] BHOSALE, GITE, (Dec.-2019). Comparison of MQTT & COAP Application Layer Protocols of IoT. International Journal of Electrical, Electronics and Data Communication, ISSN(p): 2320-2084, ISSN(e): 2321-2950 Volume-7, Issue-12. <http://iraj.in>.

[13] Bansal, J. C., Deep, K., & Nagar, A. K. (2021, August 3). Special issue on swarm intelligence and its applications to engineering. Journal: evolutionary intelligence. *Evolutionary Intelligence*, 14(3), 1187–1188. <https://doi.org>.

[14] Al-muqarm, A. M. A., & Rabee, F. (2019). IoT Technologies for Mobile Crowd Sensing in Smart Cities. *Journal of Communications*, 745–757. <https://doi.org>.