People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

UNIVERSITY KASDI MERBAH OUARGLA

Faculty of New Technologies of
Information and Communication

Department of computer science
and Information Technology

Mémoire

With a view to obtaining the diploma of

MASTER Informatique

(Specialty: Industrial)

Presented by:

BEN SADIA MERIEM
SID KENZA

Title

PERFORMANCE EVALUATION OF AN IoT PROTOCOL

Before the jury:

Dr. Boukhamla Akram	MCA	President	UKM Ouargla
BENMIR Abdelkader	MAA		UKM Ouargla
KHELILI Khalida	MAA	Supervisor	UKM Ouargla

College year: 2019-2020

THANKS

My Lord, I am instructed to thank your grace that bestowed upon me and my father and to work

You will be pleased with him, and enter me with your mercy on your righteous servants. "

- Ants 01-

Praise be to God, so much praise for his innumerable blessings, and from him he has succeeded us to complete

This work, where it was bounty to us, was great.

In gratitude we extend our sincere thanks, sincere praise and sincere appreciation to all of us

Help us accomplish this work and especially mention Professor Khalili.KF for her acceptance

Supervising this modest study with all effort and time with care, guidance and support.

We also do not forget the professors of the Department of Computer Science for their advice and guidance.

Finally, we are pleased to extend our sincere thanks and appreciation to all those who helped us little or much, hoping for them

From God Almighty reward, reward and praise, and praise be to God, Lord of the worlds.

Dedication1

Praise be to Allah . Alhamdulillah for everything after a study process that has brought with it many difficulties, hardship and fatigue. I dedicate my graduation to my hope in life and the joy of my eyes and the secret of my success, my dear mother, may God last and prolong her life. And my father, who taught me how to hold the pen and how to write words without regret. To everyone in my heart, just as I do not forget all those who supported us in this suffering from the Corona pandemic to the scientific and technical edifice and the medical staff, especially to the Qasdimerbah University. I dedicate this work to all of them.

BEN SADIA MERIEM

Dédicaces2

Praise be to God, who helped us with knowledge, adorned us with dreams, and honored us with piety I dedicate this humble work to the most precious thing in existence my honorable parents whose words were to pray for me for success and fulfillment. Punished by paradis as wide as the heavens and the earth, and to all my Muslim brothers

To all our relatives and loved ones and those with whom I have shared the sweet and bitter life of those whom I cherish their company, and I especially mention Maryam, the work friend. To all the classmates for whom I have the highest love

To those with whom the future of science brought me together, with whom I shared the various stages of my academic career. To all those who is at my heart just as I do not forget all those who supported us in this suffering with the Corona pandemic to the scientific and technical edifice and the medical staff, to especially to University Qasdi Merbah Ouargla to all of these I dedicate this work.

SID KENZA

Abstract:

The Internet of things is defined as making different things connected to the Internet as phone, refrigerator, traffic lights, smart home,street lights and even the person himself, where these things can communicate with each other and accomplished various tasks without human intervention.

The goal of this technology is to improve individual life, making it safer and more luxurious, and helping him to save time and effort.

IoT technologies areapplying in a lot of fields as health field, agricultural field, industrial field, self-driving cars and smart homes and cities, … etc

And by virtue of the fact that devices in the Internet of Things are small and restricted in terms of energy, processing power and battery life and depend on wireless sensor networks (WSN)that are characterized by low-bandwidth and unreliable communication, the result is that many of these devices cannot provide an efficient and acceptable communication.

In order to improve IoTconnection several protocols were proposed in application layer (MQTT,AQMP, CoAP …. where the most used protocol isCoAP.) , and because of this varity of protocols (no standarisation) making choice of protocol for application is not so easy.

For helping user in there choice of communication protocol this work will give a performance study of one the most used protocols : CoAP.

In this work we have evaluated CoAP protocol, by evaluating the parameters: energy consumption, throughput, successfully delivered packets and delays, by simulating the protocol using the COOJA simulator, and studying the results obtained from the simulation then analyzing them to find its weakness and adventages.

We have concluded, at the end of our work, that the CoAP protocol acheeve interoperability, reliability, and scalability. One of its weaknesses is the energy consumption in abundance .

Keywords: Internet of Things, CoAP, Performances Evaluation, Simulation, COOJA, WSN.

Résumé

L'Internet des objets est défini comme la création de différentes choses connectées à Internet comme le téléphone, le réfrigérateur, les feux de signalisation, la maison intelligente, les lampadaires et même la personne elle-même, où ces choses peuvent communiquer entre elles et accomplir diverses tâches sans intervention humaine.

Le but de cette technologie est d'améliorer la vie individuelle, la rendant plus sûre et plus luxueuse, et l'aidant à gagner du temps et des efforts.

Les technologies IoT s'appliquent dans de nombreux domaines tels que le domaine de la santé, le domaine agricole, le domaine industriel, les voitures autonomes et les maisons et villes intelligentes, etc.

En raison du fait que les appareils de l'Internet des objets sont petits et limités en termes d'énergie, de puissance de traitement et d'autonomie de la batterie et dépendent de réseaux de capteurs sans fil (WSN) caractérisés par une faible bande passante et une communication peu fiable, le résultat est que bon nombre de ces appareils ne peuvent pas fournir une communication efficace et acceptable.

Afin d'améliorer la connexion IoT plusieurs protocoles ont été proposés en couche application (MQTT, AQMP, CoAP…), Et en raison de cette variété de protocoles (pas de standardisation) le choix du protocole d'application n'est pas si facile.

Pour aider l'utilisateur dans le choix du protocole de communication, ce travail permettra d'étudier en permanence l'un des protocoles les plus utilisés: CoAP.

Dans ce travail nous avont évaluer le protocole COAP, en évaluant les paramètres: consommation d'énergie, débit, paquets reçus avec succès et retards, en simulant le protocole à l'aide du simulateur COOJA, et on étudiant les résultats obtenus à partir de la simulation puis on les analysant afin de découvrir les perfermances du protocole, ainsi que ses point faibles.

Nous avons conclu, a la fin de notre travail, que le protocole CoAP garantit l'interopérabilité, la fiabilité et l'évolutivité. L'une de ses faiblesses est la consommation d'énergie en abondance.

Mots clés: Internet des objets, CoAP, performances, évaluation, simulation, COOJA, WSN.

الملخص:

تعرّف تقنية إنترنت الأشياء بأنها تجعل أشياء مختلفة متصلة بالإنترنت مثل الهاتف ، والثلاجة ، وإشارات المرور ، والمنزل الذكي ، وأضواء الشوارع ، وحتى الشخص نفسه .حيث يمكن لهذه الأشياء التواصل مع بعضها البعض وإنجاز مهام مختلفة دون تدخل بشري.

الهدف من هذه التقنية هو تحسين الحياة الفردية وجعلها أكثر أمانًا ورفاهية والمساعدة على توفير الوقت والجهد.

يتم تطبيق تقنيات إنترنت الأشياء في العديد من المجالات مثل المجال الصحي ، والمجال الزراعي ، والمجال الصناعي ، والسيارات ذاتية القيادة ، المنازل والمدن الذكية ، ... إلخ.

وبحكم حقيقة أن الأجهزة في إنترنت الأشياء صغيرة ومقيدة من حيث الطاقة وقوة المعالجة وعمر البطارية وتعتمد على شبكات الاستشعار اللاسلكية (WSN) التي تتميز بنطاق ترددي منخفض واتصالات غير موثوقة ، فإن النتيجة هي أن العديد من هذه الأجهزة لا يمكنها توفير اتصال فعال ومقبول.

من أجل تحسين اتصال إنترنت الأشياء ، تم اقتراح عدة بروتوكولات في طبقة التطبيق (MQTT، AQMP، CoAP...الخ.) ، وبسبب هذا التنوع في البروتوكولات فإن اختيار بروتوكول للتطبيقات انترنت الأشياء ليسبالامر السهل .

لمساعدة المستخدم في اختيار بروتوكول الاتصال ، سيعطي هذا العمل دراسة مناسبة لأحد البروتوكولات الأكثر استخدامًا: CoAP.

في هذا العمل ، قمنا بتقييم بروتوكول COAP ، من خلال تقييم المعاملات: استهلاك الطاقة ، والإنتاجية ، والحزم المستلمة بنجاح والتأخيرات ، من خلال محاكاة البروتوكول باستخدام محاكي COOJA ، ودراسة النتائج التي تم الحصول عليها من المحاكاة ثم تحليلها لمعرفة اين تكمن قوة هذا البروتوكول.

أخيرًا ،توصلنا أن بروتوكول CoAP يحقق قابلية التشغيل البيني والموثوقية وقابلية التوسع ، ومن نقاط ضعفه هو استهلاك الطاقة بكثرة .

<u>الكلمات المفتاحية:</u> إنترنت الأشياء ، CoAP ، الأداء ، التقييم ، المحاكاة ، COOJA، WSN.

Table of contents

CHAPETR 1: INTERNET OF THINGS

CHAPETR 2: CoAP PROTOCOL

CHAPETR 3: CoAP PROTOCOL SIMULATION

CHAPETR 4: **PERFORMANCES EVALUATION**

Liste of tables

Liste Of figures

List of Abreviations

IEEE	Institute of Electrical and Electronics Engineers
IETF	International Engineering Task Force
CoAP	Constrained Application Protocol
HTTP	Hypertext Transfer Protocol
IoT	Internet Of Thing
M2M	Machine To Machine
MQTT	MQ Telemetry Transport
NFC	Near Fields Communication
RFID	Radio Fréquence Identification
IP	Internet Protocole
IPV6	Internet Protocole version 6
IPV4	Internet Protocole version 4
6LoWPAN	over Low Power Wireless Personal Area Networks
MAC	Medium Access Control
WSN	Wireless Sensor Networks
GPS	Global Positioning System
AMQP	Advanced Message Queuing Protocol
GPS	Global Positioning System
TCP/IP	Transmission Control Protocol / Internet Protocol
UDP	User Datagram Protocol
RAM	Random Access Memory
WPAN	Wireless Personal Area Network
PLC	Public limited company
CPU	central processing unit
Wifi	wireless fidelity
HTML	hyper text markup language
LLN	Low-power and Lossy Network
RPL	Routing Protocol for Low-power and Lossy Network
URL	Uniform Resource Locator
TLS	Transport Layer Security
WOT	web of thing
DTLS	Datagram Transport Layer Security
ROM	Read-Only Memory.
DR	resource discovery

INTRODUCTION:

The development of informatics and technology has gone through many stages that have changed the world for the better, from the computer to the Internet to the laptop to the smartphone, where the connection between us no longer exceeds the click of a button and is now possible for various things (refrigerators, TVs, traffic light, humans ... etc.). These things can communicate with each other to perform certain tasks and functions without human intervention, and this is what we call today the Internet of Things (IoT).

The use of IoT technology has become reliable in most fields, including the health field, the industrial field, the agricultural field, as well as what is known as smart homes, smart cities, self-driving cars, and others.

In order to use this technology, the Internet of things applications must observe all its characteristics including reliability, interoperability, scalability, and security. However, ensuring these characteristics is difficult due to the nature of the connected devices and the formation of complex networks in the Internet of things.

We can say that any object connected to the Internet that it is a smart thing, in other words, anything that can obtain an address ipv6, ipv4. Among the most important components of a smart object are the sensors devices such as motion sensors, humidity sensors, and gas sensors.

According to the task required from the smart object, and by the nature of these devices (constrained and limited Resources in terms of energy, processing power, memory capacity, and battery) it works on the wireless sensor networks(WSN), which are characterized by low-power and relatively unreliable communication, as well as intermittent connections. From here, one of the most important challenges in the Internet of things appears, which is how to support the communication from machine to machine with high efficiency and reliability.

In order to achieve the best connection, several protocols have been proposed (CoAP, MQTT, ACMP...etc) where the implementation of these protocols must respect the criteria of the Internet of Things and meet all the required characteristics. Until now there is no basic protocol that combines all standards and characteristics. Here appears for users the variation and difficulty of choosing the best communication protocol for IoT applications.

In recent years, it is widely expected that CoAP will become the standard protocol in the Internet of Things.

In this work, we will evaluate the performance of CoAP protocol through simulation and by the use of a COOJA network simulator. allows us to perform simulations under low bandwidth and high latency, and the network loss packet and this is in order to verify that the system is running efficiently and respect the required characteristics(characteristics internet of things). Through measurements obtained from the simulation, we can give insight into the effectiveness of the protocol within a constrained wireless network, through the results of the experiments.

We will organize this work as follows:

In Chapter 1, we will show some concepts of the Internet of Things. Introducing its definition, some of the fields using IoT applications, architecture, and the protocols most commonly used. Finally, we mention some of the challenges (aspects) of the Internet of Things.

In the second semester, we will conduct a theoretical study of the CoAP protocol. Presenting the definition, features, structure and characteristics, as well as the applications that use the protocol, and we study the opposite of the characteristics of the Internet of Things.

In Chapter Three, a method for simulating the CoAP protocol in a COOJA simulator, we study its effectiveness through some experiments based on smart street lighting scenarios, according to different criteria and standards.

In Chapter 4, we will assess the performance of the CoAP protocol with aspects of the Internet of Things through the results obtained from the simulations.

RELATED WORK

Due to its efficacity and degree of use in the last years, many works have studied and compared the performance of application CoAP protocol in the IoT environment. We now review some relevant of these works in chronological order.

- In [1], performance analysis of service discovery of CoAP servers in a local network was conducted using Zolertia ZI Contiki nodes and COOJA simulator.

- In this paper[2], CoAP allows these devices to communicate interactively over the internet. The integration of such tiny, ubiquitous electronic devices to the internet enables interesting real-time applications. evaluate the performance of a stack consisting of CoAP and 6LoWPAN over the IEEE 802.15.4 radio link using the Contiki OS and COOJA simulator, along with the CoAP framework Californium (Cf).
- In this paper[3], we experimentally evaluate these rate control mechanisms for untrusted CoAP communications between devices over simulated GPRS / UMTS links and in the IEEE 802.15.4 multi-storey test of constrained devices. The results show that, contrary to the observation, CoCoA performs better than, or at least similar to, the default CoAP in terms of packet delivery ratio and delay in all scenarios analyzed with the COOJA simulator.
- To achieve the vision of the Internet of Things (IoT). In [4], implemented both centralized and distributed modeling for CoAP-based service discovery technologies. He used Contiki OS and COOJA simulator to build an experimental setup of proprietary prototyping systems. Loopholes have been reached in CoAP-based public service discovery technologies to recommend further areas for improvement.
- In [5]. This paper shows the implementation of CoAP and comparison of CoAP with HTTP with regard to energy consumption and response time of both client server transaction and the results shows that CoAP is more appropriate compared to HTTP. The simulations has been carried out using Contiki Operating system and Cooja simulator which serves for networked, memory constrained systems and low power wireless IoT devices.
- To study protocol performance in this paper [6], he compared protocols such as CoAP, 6LoWPAN, and RPL using the Contiki OS COOJA simulator. This work aims to analyze these protocols based on some criteria like power consumption, radio duty cycle, average time between beams, etc. It was analyzed and concluded that each protocol should be preferably based on its application path. However, depending on the power consumption or the average time between packs, CoAP produces a slightly better result.

- In this paper [7] a comprehensive survey on RPL was conducted. The various techniques used in RPL have been studied in this paper. RPL protocol performance is examined with a COOJA simulator. RPL work tested using Contiki-enabled sky motions in a COOJA setting. Heterogeneity of network density and node type are the main factors considered for the RPL examination in terms of energy efficiency, packet delivery ratio, and latency.
- In [8], reviewed XMPP, AMQP, CoAP, MQTT, DDS, and MQTT-SN protocols that are available in the application layer of IoT and afterward they compared every protocol with knowing their execution. To assess their performance, they had picked different measurements, for example, packet transmission ratio, throughput, power consumption, and bandwidth. It is audited that the MQTT, XMPP, AMQP, and MQTT-SN protocols that keep running on TCP produces higher PDR while contrasted with CoAP and DDS protocols that keep running on UDP, which does not back retransmission of packets. Also, it is watched that CoAP has higher throughput, consistent ideal bandwidth utilization, and low power consumption differentiated with other data protocol that is an appropriate real-time environment. After that, they watched, how the gadget gets managed remotely utilizing Contiki OS with COOJA simulator
- In this paper [9], trade-offs between performance, power consumption, and level of security are explored for the most recent version of the widely accepted Contiki OS (version 3.x) when IETF-supported DTLS security is enabled for the constrained Application Protocol (CoAP). More specifically, the DTLS framework is integrated into the Contiki 3 CoAP stack for two different cipher sets,and performance is evaluated against insecure CoAP implementations through simulation, in terms of speed, overall memory, and power consumption for different WSN server network environment.
- In this paper [10], we present a study on the power consumption and network traffic of the IoT Home Application Layer, Restricted Application Protocol (CoAP), and Message Queue Remote Transfer (MQTT) version of Sensor Networks (MQTT_SN). The simulations in COOJA presentedto evaluate the performance of these protocols with different network configurations.

The following table shows what has been studied in the past and the addition that has been made to this work:

DOCUMENTS	WOTRK
DC[2]	They evaluated coap protocol performance. By calculating packet loss, throughput, delay. And by increasing the number of packets.
DC[3]	They evaluated coap protocol performance. By calculating delay. And by increasing Data Transmission Interval.
DC[4,6]	They evaluated coap protocol performance. By calculating response time, power consumption. And by increasing the number of servers.
DC[5]	They evaluated coap protocol performance compared to HTTP. By calculating response time, power consumption. And by increasing the number of servers.
DC[8,10]	They evaluated coap protocol performance compared to XMPP, AMQP, CoAP, MQTT, DDS, and MQTT-SN. By calculating packet transmission ratio, throughput, power consumption. And by increasing the number of servers.

After getting acquainted with the previous study in which the performance of the coap protocol was evaluated as shown in the table. In this work, we will add an overall assessment of the protocol in terms of throughput, power consumption, and successfully received packets by increasing the number of servers and packet size in each trial. With two improvements suggested, if possible.

Chapter1:

INTERNET OF THINGS

1 INTRODUCTION

The Internet of Things (IoT) is a network of physical objects - devices, tools, vehicles, buildings, and other elements embedded in electronics, circuits, software, sensors, and a network connection that enables these objects to collect and share data.[11]

The Internet of Things interact without human intervention, and some areas have been developed, including healthcare, transportation, and auto industries. There have also been many new developments in the integration of living organisms with sensors on the Internet. IoT development includes infrastructure, communications, interfaces, protocols, and standards.

In this chapter, we will, firstly, give some definitions of the Internet of Things, some of the fields that use IoT applications, then the architecture of the Internet of Things and the most popular protocols used, where we focus on the application protocols, in the last, we mention some challenges (aspects) of the Internet of Things.

History:

The concept of the internet of things first became popular in 1999,through the auto_id center at MIT and related maket analysis publications .

The Internet of Things emerged to connect machines to servers capable of supervising them (these machines include computers networked in what some have called the "machine internet"). Gradually objects have been modified (with RFID1 chips for example) or designed to "speak IP2 protocol," becoming "connected objects," connected to centralized servers and/or able to communicate with each other and/or with server networks and various actors, in a less and less centralized way.[12]

The explosion in the number of smartphones and connections has created a new market with almost infinite opportunities. In 2016, 5.5 million objects are connected every day worldwide. This number could quickly exceed the billion mark by 2020.

2 THE INTERNET OF THINGS DEFINITION

The Internet of Things (IoT) is "a network that connects and combines objects with the Internet, following the protocols that ensure their communication and exchange of information through a variety of devices "[13].

And among the other definitions, we quote the following: “The Internet of Things is a network of networks which allows, via a standardized and unified electronic identification system, and wireless mobile devices, to identify directly and unambiguously digital entities and physical objects and thus to be able to recover, store, transfer and process, without discontinuity between the physical and virtual worlds, the related data" [14].

3 FIELDS OF APPLICATION OF THE IoT

There are several areas and environments in which IoT can play a remarkable role and improve our lives. These applications include transportation, healthcare, environmental monitoring, smart cities, industrial automation, and agriculture ... etc.

IoT applications have a big impact on the quality of life of people as they also generate huge benefits. Figure (1) shows some areas of the application of IoT.[15]

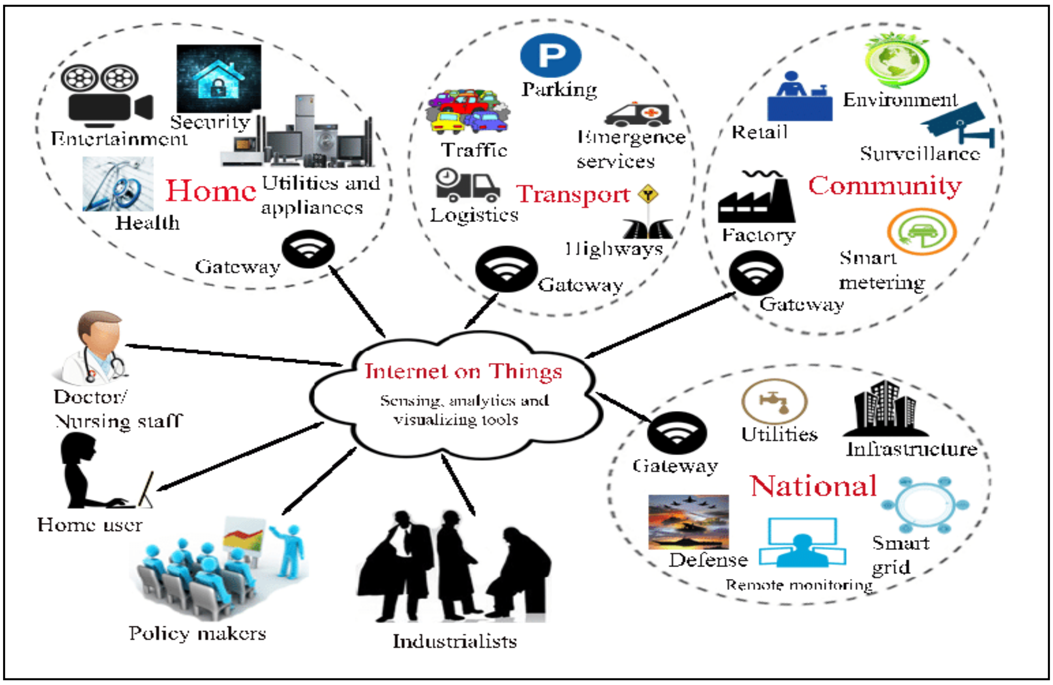


Figure 1: **Fields of applications of IoT**

4 THE ESSENTIAL ELEMENTS OF INTERNET OF THINGS

The Internet of Things consists of several basic components: Connecting objects, sensors, and sensor networks.

4.1 Connect Things

Internet of Things has many of the technologies used to connect an object to the Internet. We only cite a few major Internet of Things technologies. Including RFID, NFC, and ZigBee communications protocol, Wi-Fi, Bluetooth.

Figure 2: **IoTConnect Things**

4.2 Sensors / Device

A sensor is a device for collecting information that elaborates from a physical quantity, another physical quantity of a different nature (very often electrical). This quantity representative of the quantity taken can be used for measurement or control purposes. subject to considerable constraints, among these constraints we cite the most important: energy, processing power, and wireless exchanges. The exchange of information or even the interaction between different objects is possible, even without user intervention. The object will "capture", measure a physical characteristic of its environment, possibly apply computerized processing to it, and provide the result to other users (For example, our phone is a device that has multiple sensors such as GPS, accelerometer, camera but our phone does not simply sense things). [16]

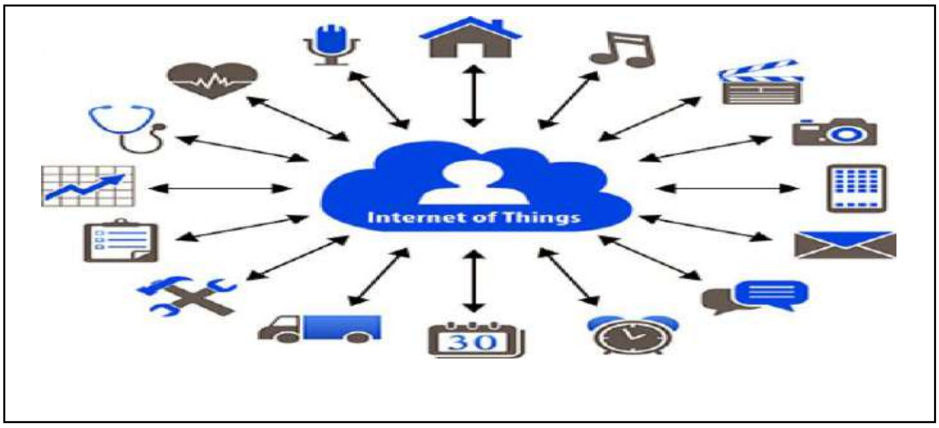


Figure 3: **IoTSensors**

4.3 Sensor Networks

A wireless sensor network is an ad hoc network with most nodes which are micro-sensors capable of collecting and transmitting environmental data in an autonomous manner. The position of these nodes is not necessarily determined. They can be randomly dispersed in a geographical area, endowed with wireless transmission capacities, autonomous in energy, and whose positioning is, most often, free [17].

4.4 User Interface

the user may sometimes also have an interface through which they can actively check into the *IoT* system. for example, a user who has a camera installed in his home, the user may also be able to perform a procedure that may be counterproductive and affect the system. for example, if the user detects some changes in the refrigerator, the user can set the temperature remotely through their phone.

5 THE INTERNET OF THINGS ARCHITECTURE

A variety of IoT architectures are proposed by standards bodies. A couple of architecture is based on a 3-layer IoT architecture shown in figure 4. and Other types of architecture are 5-layer IoT architecture shown in figure 4.[18,19,20,21]

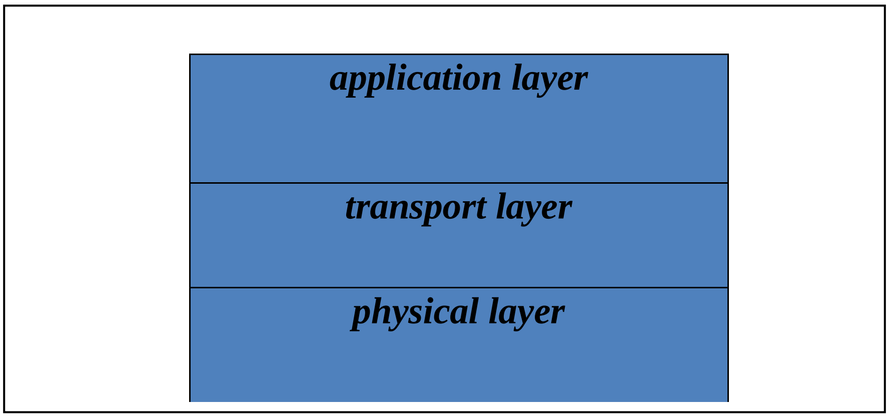


Figure 4: **Three-layer IoT architecture[21]**

Beginning of the IoT, the accepted architecture was the 3-layer architecture. It consists of three layers which are called physical, transport , and application. The purpose of the physical layer is to identify each thing in the IoT system. This is done by gathering information about each object. This layer contains RFID tags, sensors, cameras, etc. The second layer is the transport layer. It is the core of the IoT. It transmits the information gathered by the physical layer. It contains the software and hardware instrumentations of internet network in addition to the management and information centers. The third layer is the application layer. The application layer’s target is to converge between the IoT social needs and industrial technology (i.e. it can be considered as the middle tier between the industry technologies and how it can be controlled to cover the human needs).

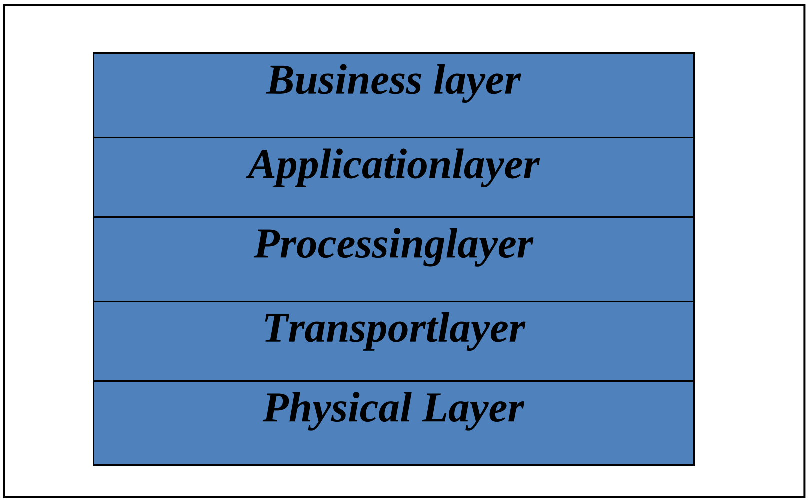
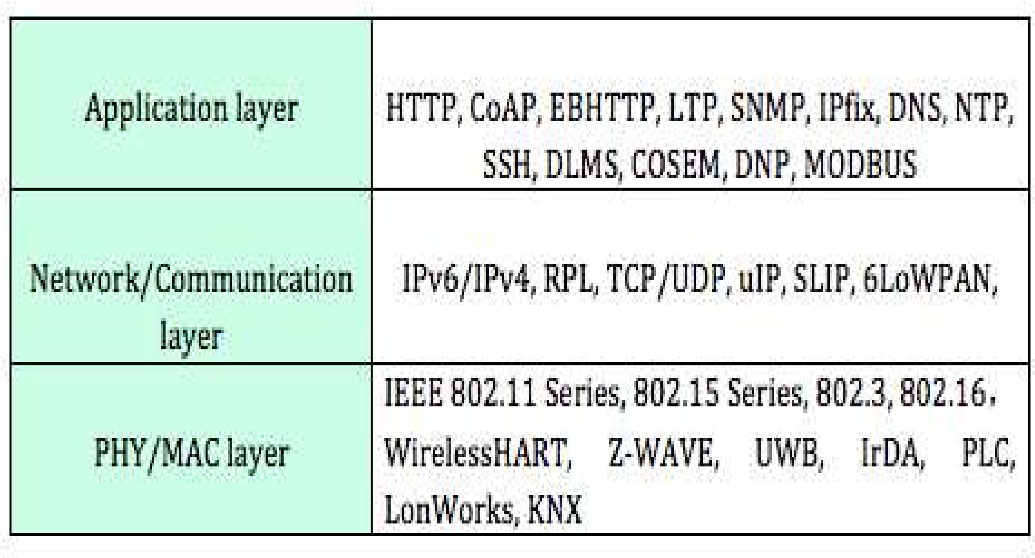


Figure 5: **Five-layer IoT architecture[21]**

The 3-layer architecture became not sufficient due to the expected IoT development. Therefore, 5-layer architecture is proposed. The first layer is called business. The purpose of this layer is to define the IoT applications charge and management. Also, it is responsible for the user's privacy and all research related to IoT applications. The second layer is called an application. The target of this layer is determining the types of applications, which will be used in the IoT. Also, it develops the IoT applications to be more intelligent, authenticated, and safe. The third layer is called processing. Its responsibility is to handle the information gathered by the Physical layer. The handling process contains two main topics; storing and analyzing. The target of this layer is extremely hard due to the huge gathered information about system things. So, it uses some techniques such as database software, cloud computing, ubiquitous computing, and intelligent processing in information processing and storing. The fourth layer is called transport. It seems like the transport layer in the 3-layer architecture. It transmits and receives the information from the Physical layer to the processing layer and visa versa. It contains many technologies such as infrared, Wi-Fi, and Bluetooth. Also, the target of this layer is to address each thing in the system using IPV6. The fifth layer is called Physical. The target of this layer is to define the physical meaning of each thing in the IoT system such as locations and temperatures. It also gathers the information about each object in the system and transforms this data into signals. In addition, it contains the technologies that are used in the IoT such as the RFID and the GPRS [18]. Figure (5) presents the 5-Layer architecture.

6 THE INTERNET OF THINGS COMMUNICATION PROTOCOLS

According to the three-layer structure of the Internet of Things in the table (1) consisting of the physical layer concerned with the transfer of packets between parts of the network and determines its communication methods (PLC, MAC, IEEE, ... etc.), and the transport layer that creates communication channels for data transmission and is used by the application layer (such as IPV6, IPV4, 6LOWPAN, TCP, UDP ... etc.), and the application layer it is the most important layer, which includes the protocols used by the applications; to provide services to users or exchange data between applications, including the protocol that is studied in this note (the CoAP protocol).



Application layer	HTTP, CoAP, EBHTTP, LTP, SNMP, IPfix, DNS, NTP, SSH, DLMS, COSEM, DNP, MODBUS
Network/Communication layer	IPv6/IPv4, RPL, TCP/UDP, uIP, SLIP, 6LoWPAN,
PHY/MAC layer	IEEE 802.11 Series, 802.15 Series, 802.3, 802.16, WirelessHART, Z-WAVE, UWB, IrDA, PLC, LonWorks, KNX

Table 1: **Protocols in IOT**

6.1 Application layer protocols:

because we are interested in the application protocols, we will give a brief definition of the most famous :

6.1.1 AMQP

AMQP protocol is used in IoT environment which focuses on message exchange, and communication. AMQP uses different message delivery guarantees; at most once, at least once, and exactly once to ensure reliability. This protocol also uses a TCP transport layer to ensure reliability. Publish/subscribe approach of AMQP consists of two components: exchange queue and message queue, the exchange queue is responsible for message routing to the suitable order in queue. Message queue keeps storing messages until they are sent to the receiver.

There is a specific process with a set of rules to exchange messages between exchange components and message queues. [22]

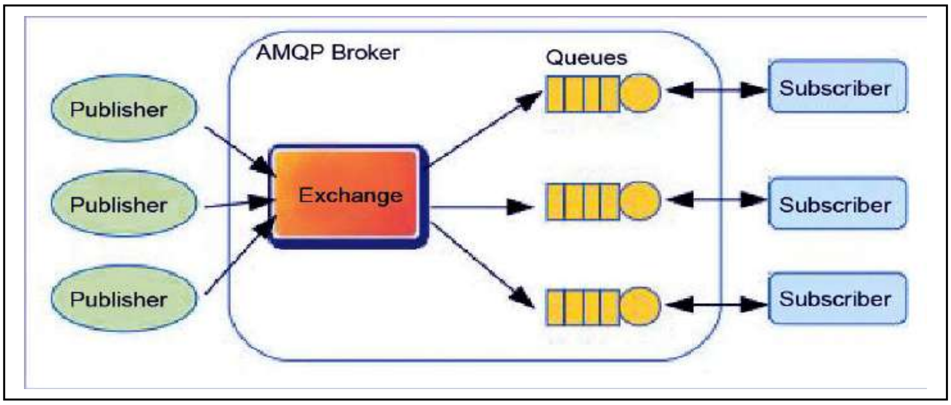


Figure 6: **AMQP protocol**

6.1.2 CoAP

Constrained application protocol (CoAP) is a request/response protocol; it is similar to the client-server model. Nevertheless, this protocol is only sufficient in a constrained environments such as: constrained node with low capability in RAM or CPU, and constrained network, such as lower power using wireless personal area network (WPAN). This constrained environment led to bad packet delivery and high overhead. CoAP was designed by Internet Engineering Task Force (IETF) which is mainly interested in machine to machine (M2M) applications and the automation of systems to reduce overhead, enhance packet delivery, and to increase the simplicity of work, by using a simple interface with HTTP [23].

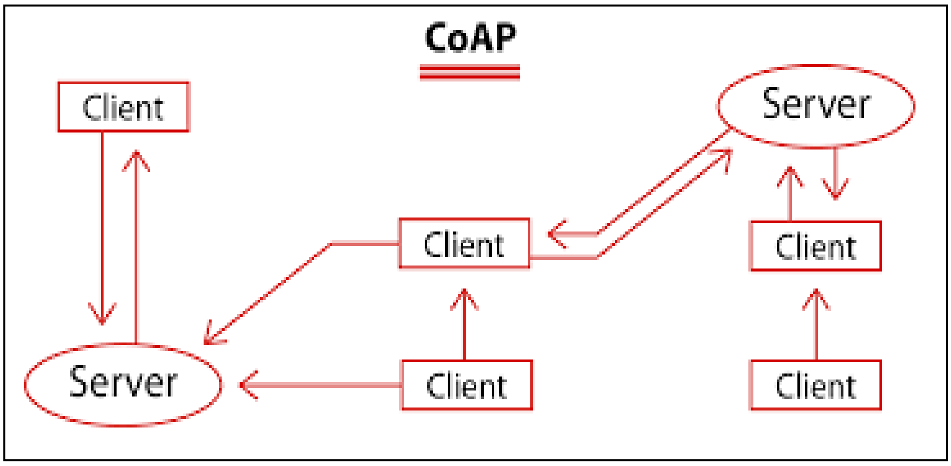


Figure 7: **CoAP protocol**

6.1.3 HTTP

The HTTP protocol (HyperText Transfer Protocol) is the most used protocol on the Internet, its “request/response” architecture the purpose of the HTTP protocol is to allow the transfer of files (essentially in HTML format) localized thanks to a character string called URL between a browser (the client) and a web server [9].

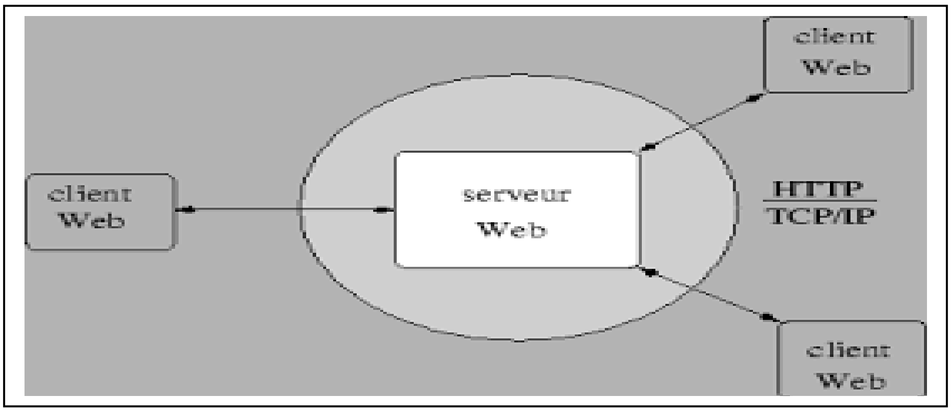


Figure 8: **HTTP protocol**

6.1.4 MQTT

MQTT Represents an ideal messaging protocol for IoT and M2M communications. It aims to connect devices and integrated networks to applications and middleware. MQTT uses the subscription publication model to provide transition flexibility and ease of implementation. It is suitable for limited resource devices that use unreliable or low bandwidth links. MQTT is built on top of the TCP protocol. It consists of three components, Subscribers, Publishers and Brokers. Many apps use MQTT such as Healthcare, Monitoring, Energy Meter, and Facebook Notification. Therefore, the MQTT protocol enables small, low power, low memory devices to be routed in vulnerable areas and low bandwidth networks.[24]

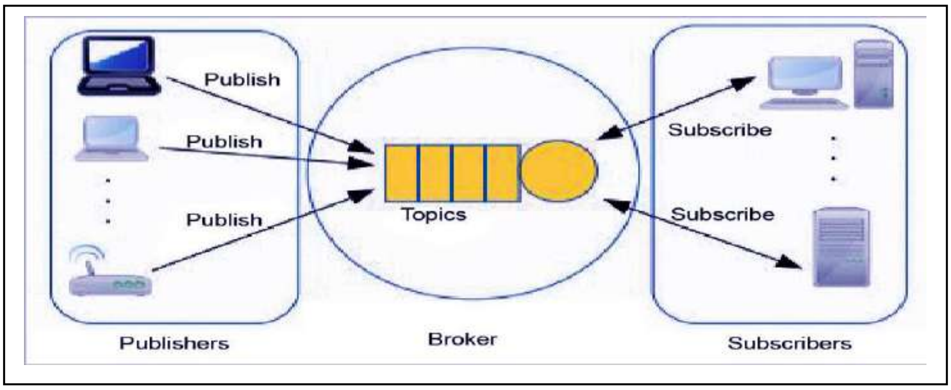


Figure 9: **MQTT protocol**

Comparison between IoT protocols [25]:

Protocol	MQTT	AMQP	CoAP	HTTP
Type Protocol	Messaging	Messaging	transfer Web	transfer Web
Model Communication	Publish / Subscribe	Producr /Consumr	Request/ Response	Request/Response
Transport	TCP/IP	TCP/IP	UDP/IPv6 /6LowPAN	TCP/IP
Security	TLS/SSL	TLS/ SSL	DTLS	TLS/SSL
Format	Binary, Text (json, xml, csv)	Binary, Text	Binary, Text	Binary, Text
Constraints on objects connected	Strong	Medium	Strong	Weak
Header size (bytes)	2	8	4	4
Mainframeworks	Emqtt, HiveMQ, Mosquitto, Eclipse Paho	RabbitMQ, StormMQ	Eclipse Californium, nCoAP	Django REST, Apache Tomcat, Node.js, Ruby on Rail

Table 2: Comparison between IoT protocols

7 CHALANGES OF THE INTERNET OF THINGS

IoT devices with limited functionality have been around for at least a decade. What has changed recently is the ubiquity of connectivity options (WIFI, 3G, and Bluetooth, etc.), cloud services, and analytics, which are great enablers for IoT. The Cloud provides a platform for hosting intelligent software, networking a large number of IoT devices, and provisioning them with a large amount of data. This enables smart decisions to be made without human intervention. However, there are still some current challenges limiting the adoption of IoT [25]:

A. Privacy and Security

As the IoT becomes a key element of the Future Internet and the usage of the Internet of Things for large-scale, partially mission-critical systems create the need to address trust and security functions adequately. New challenges identified for privacy, trust, and reliability are:

- providing trust and quality of-information in shared information models to enable re-use across many applications.
- Providing secure exchange of data between IoT devices and consumers of their information.
- Providing protection mechanisms for vulnerable devices.

B. Cost versus Usability

IoT uses technology to connect physical objects to the Internet. For IoT adoption to grow, the cost of components that are needed to support capabilities such as sensing, tracking and control mechanisms need to be relatively inexpensive in the coming years.

C. Interoperability

In the traditional Internet, interoperability is the most basic core value; the first requirement of Internet connectivity is that “connected” systems be able to “talk the same language” of protocols and encodings. Different industries today use different standards to support their applications. With numerous sources of data and heterogeneous devices, the use of standard interfaces between these diverse entities becomes important. This is especially so for applications that support cross organizational and various system boundaries. Thus the IoT systems need to handle a high degree of interoperability.

D. Data Management

Data management is a crucial aspect of the Internet of Things. When considering a world of objects interconnected and constantly exchanging all types of information, the volume of the generated data and the processes involved in the handling of those data become critical.

E. Device Level Energy Issues

One of the essential challenges in IoT is how to interconnect “things” in an interoperable way while taking into account the energy constraints, knowing that communication is the most energy-consuming task on devices.

8 THE INTERNET OF THINGS CHARACTERISTICS

Internet of things consists of several aspects, including[26,27,28,29,30,31]:

8.1 Interoperability:

Since networks consist of many heterogeneous devices and standards, the ability to communicate proportionately to their differences is essential for the Internet of Things. You can see the problems of IoT interoperability from different points of view due to the heterogeneity, one of which is the interoperability of the device with sufficient computing

resources and capabilities such as the Raspberry Pi and smartphones. Low-level IoT devices are also limited in terms of resource power, processing power, and communication capabilities of typical hosts such as RFID tags and low-cost micro-sensors. The interoperability of the network that deals with mechanisms to enable the seamless exchange of messages between systems through different networks must therefore address issues of addressing, routing, resource optimization, security, service quality, and mobility support. Grammar interoperability This level of interoperability is important to enable migration Seamless messaging between different Internet of Things (IoT) systems such as (WoT) is proposed to provide greater interoperability. Semantic interoperability that allows contact parties to share information and the interoperability of the platform that allows applications to access IoT platforms and integrate data from different platforms. It is therefore imperative that the devices and protocols that are part of the Internet infrastructure be able to accurately interpret message exchange. Lack of system interoperability can increase complexity and cost.

8.2 Scalability:

Due to the increase in the world population, the number of devices has outpaced the world population and continues to grow. It is therefore imperative that the effective routing protocols of Wireless Sensor Networks (WSNs) that will participate in the Internet of Things be scalable and adaptive to changes in the network topology. Hence, the scalable protocol should perform well as the network grows or the workload increases. The scalability of sensor networks also supports the expansion of the network to more nodes that might not be expected during the initial network design phase.

8.3 Network environment:

The network type and the type of nodes that constitutes the network can greatly influence on the success or failure of a IoT protocols. Whenever one is setting up a network choosing the right protocol that can serve the needs of the devices in it makes a relevant difference in the environment.

8.4 Performance:

IoT protocols effectiveness and methods used to handle situations on the network and devices can determine the speed and response time capability of the protocols.

8.5 Reliability:

The main purpose of collecting big data for IoT system is real situation awareness with accurate information. Therefore, reliability of data exchange between sensors is required if those want to improve application usage. For example in computer networks, a trusted protocol is a communication protocol that tells the sender whether or not data delivery to the intended recipients was successful. For example, RDP (Trusted Data Protocol) downloads and corrects errors using an efficient and reliable data transfer service. The main goal of RDP is to remain effective in environments where there may be non-sequential message segment delivery or lengthy delays and loss of transmission.

Energy Efficiency: The focus of this project is on small IoT devices. One of the limitations that occurs in such devices is the short life span of the power supply. So it is vital that IoT protocols avoid rapidly depleting battery life.

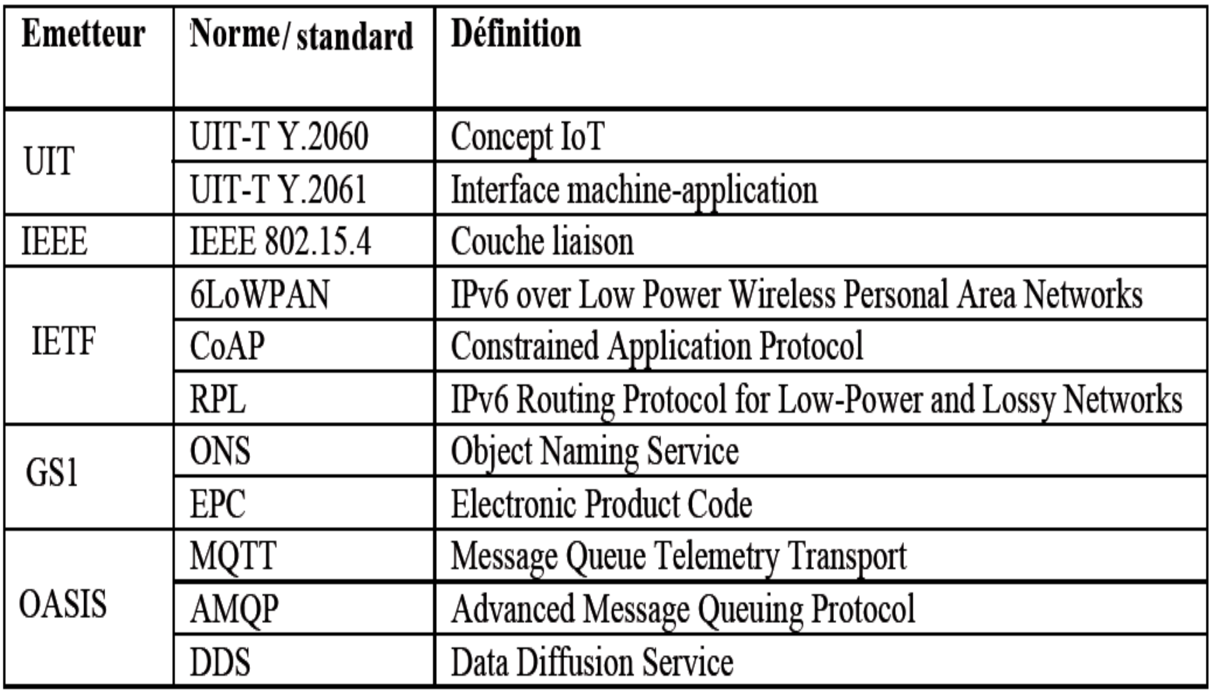
8.6 Security:

Some data exchanges by devices from the Internet of Things may contain sensitive information that should not be easily accessed, for example web cameras that allow the homeowner to remotely monitor his home from his smartphone or a sensor that acts as keys to access your home or car must Access is granted only to the owner. Increasing the number of wearable devices and sensors. One of the characteristics of the IoT application layer is data sharing. The latter faces issues related to data privacy and access control. Among the most common application tier security issues are data access and authentication, phishing attacks, malware active, layer security requirements and Malwares attack.

9 Standards And Standardization

As IoT devices continue to saturate society, standardization is essential to achieve universally accepted specifications and protocols for true interoperability between IoT devices and applications. Today's standards mark a major milestone for the Internet of Things by offering the unique value proposition of a single interoperability platform for all activated devices, [32]

Nearly 140 organizations around the world are now directly or indirectly affected by the standardization of M2M communication. This phase of standardization is indeed one of the crucial factors in the evolution of the mobile Internet towards the Internet of Things. There are thousands of standards "specific" to particular IoT contexts, among them and in particular those already used by industry finding those offered by the Internet Engineering Task Force (IETF), Institute of Electrical and Electronic Engineers (IEEE), International Telecommunications Union (ITU) and Global Standard1 (GS1), Organization for the Advancement of Structured Information Standards (OASIS) , as illustrated in the following table (3)[33]:



Emetteur	Norme/ standard	Définition
UIT	UIT-T Y.2060	Concept IoT
	UIT-T Y.2061	Interface machine-application
IEEE	IEEE 802.15.4	Couche liaison
IETF	6LoWPAN	IPv6 over Low Power Wireless Personal Area Networks
	CoAP	Constrained Application Protocol
	RPL	IPv6 Routing Protocol for Low-Power and Lossy Networks
GS1	ONS	Object Naming Service
	EPC	Electronic Product Code
OASIS	MQTT	Message Queue Telemetry Transport
	AMQP	Advanced Message Queuing Protocol
	DDS	Data Diffusion Service

Table 3: Organization Standards de protocole IoT

- ITU: the two recommendations, itU-T Y.2060 which provides a general view of the IoT concept and the ITU-T Y.2061 that describes the conditions for the machine interface oriented to communications applications in the NGN environment (next generation networks).

- IEEE and IETF in the field of IP protocol sensor networks. These efforts were first realized by the proposal of a layer model on the OSI model as well as protocols more suited to industrial networks than the TCP/IP model on Ethernet.

- GS1: has proposed the EPC (Electronic Product Code) system, which is a unique individual identifier for identifying an electronic product, as well as the EPC global Network architecture that defines the organization of information systems to ensure the exchange of EPC information at the global level. One of its main components, the ONS (Object Naming Service), is directly based on the DNS (Domain Name System).

- OASIS [34]: a not-for-profit consortium that guides development and adoption of "open" standards for the information society. The consortium's work on the Internet of Things focuses on standardized network and messaging technologies such as Message Queue Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP), and the Data Distribution Service (DDS). These protocols are at the application layer

10 CONCLUSION

In the purpose of introducing and explaining our research field, we have given in this chapter a brief presentation of the Internet of Things, we have first begun by some IoT definitions, and some fields which use IoT applications, then we passed to present some technical issues as the essential IoT elements, the IoT architecture and some most famous used protocols where we focus in the application protocols. We finish by giving challenges (aspects) of IoT.

In the next chapter, we will present in detail the IoT application Protocol CoAP.

Chapter2:

COAP PROTOCOL

1 INTRODUCTION

Small devices are unable to communicate with limited resources. In addition to the Internet of Things (IoT), you have to pay attention to contrast. Because billions of different sensors, computers, and other communication elements need to communicate together, which may operate on different protocols. Therefore, to address this issue, the Internet Engineering Task Force (IETF) developed the constrained Application Protocol (CoAP).

In this chapter, We will get acquainted with the CoAP protocol theoretically, as we will deal with the definition, features, structure, and characteristics of the protocol, as well as the applications that use the protocol, and we will study it with IoT characteristics.

2 DEFFINITION OF CoAP (CONSTRAINED APPLICATION PROTOCOL)

CoAP is an application layer protocol developed by the IETF CoRE Working Group. It is designed for constrained environments. Based on a REST style architecture, the protocol considers the various objects in the network as resources. A Unique Universal Resource Identifier (URI) is assigned to each resource. The protocol uses the corresponding URI to operate the different resources.[35] In 2010, the IETF CoRE working group start on the development of CoAP that focus on environments of low power IP network enabling interoperability between constrained devices and the general device communication over the Internet[36].

Constrained Application Protocol (CoAP) is a fairly new web transfer protocol that was designed to be used for constrained devices (e.g. simple electronic devices with limited capabilities) and constrained networks (e.g. LLNs, 6LoWPAN)[36].

3 FEATURES CoAP HAS THE FOLLOWING MAIN FEATURES:[37][38]

- Web protocol fulfilling M2M requirements in constrained environments.
- User Datagram Protocol (UDP) binding with the optional reliability, supporting unicast and multicast requests.
- Asynchronous message exchanges.
- Low header overhead and parsing complexity.
- URI and Content-type support.
- Simple proxy and caching capabilities.
- A stateless HTTP mapping, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way or for HTTP simple interface to be realized alternatively over CoAP.

- Security binding to Datagram Transport Layer Security (DTLS).

4 CoAP ARCHITECTURE

On design, the CoAP protocol structure is the most important the key was to avoid message fragmentation so that the CoAP packets could be fit in one single frame at the Ethernet or IEEE 802.15.4 layer. As is shown in Figure 10, the CoAP Message Layer is designed to deal with UDP and asynchronous switching, and the request/response layer handles the communication method.

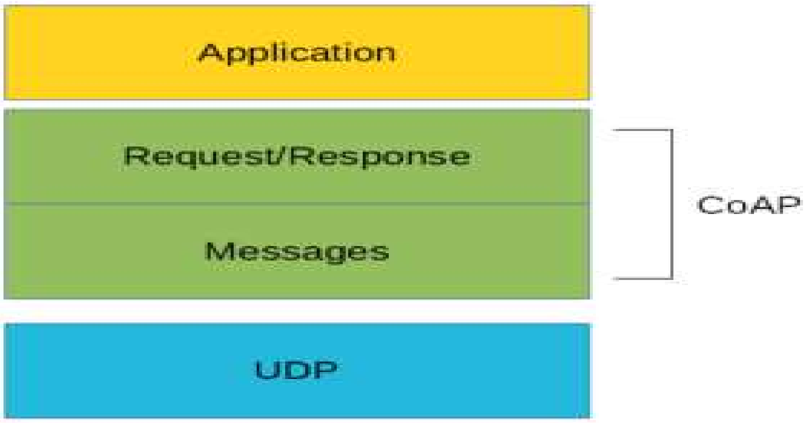


Figure 10: **CoAP architecture.**

4.1 Message Layer

This is the lowest layer of CoAP. This layer deals with UDP exchanging messages between endpoints. Each CoAP message has a unique ID; this is useful to detect message duplicates. A CoAP message is built by these parts: A binary header, A compact options, Payload.[39]

Message Layer supports 4 types message: CON (confirmable),NON (non-confirmable), ACK (Acknowledgement),RST (Reset).

A confirmable message is a reliable message. When exchanging messages between two endpoints, these messages can be reliable. In CoAP, a reliable message is obtained using a Confirmable message (CON). Using this kind of message, the client can be sure that the message will arrive at the server. A Confirmable message is sent again and again until the other party sends an acknowledge message (ACK). The ACK message contains the same ID of the confirmable message (CON).[36]

The picture below shows the message exchange process:

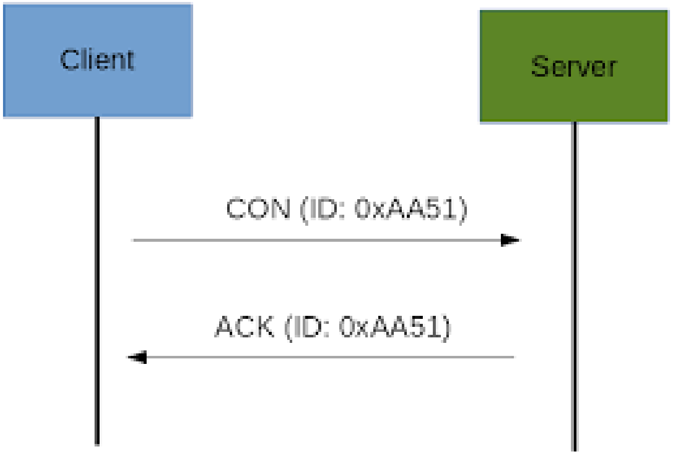


Figure 11: **Reliable message transmission (ACK).**

If the server has troubles managing the incoming request, it can send back a Rest message (RST) instead of the Acknowledge message (ACK):

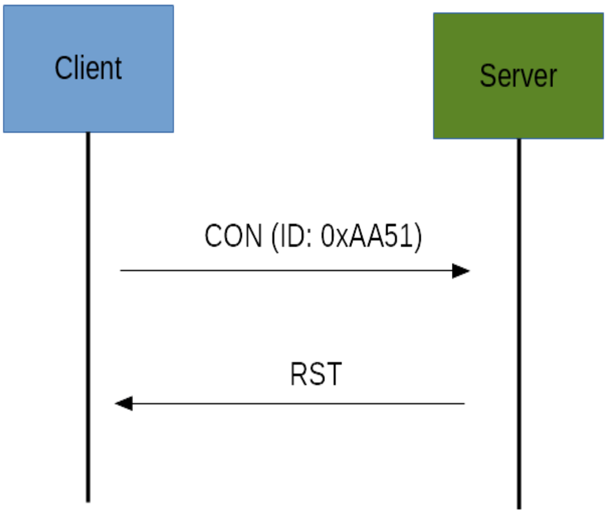


Figure 12: **Reliable message transmission (RST).**

4.2 Request/Response Layer

The CoAP Request/Response is the second layer in the CoAP abstraction layer. The request is sent using a Confirmable (CON) or Non-Confirmable (NON) message. There are several scenarios depending on if the server can answer immediately to the client request or the answer if not available.[37]

4.2.1 Scenario1: Piggybacked response

if the server can answer immediately to the client request, then if the request is carried using a Confirmable message (CON), the server sends back to the client an Acknowledge message containing the response or the error code:[36]

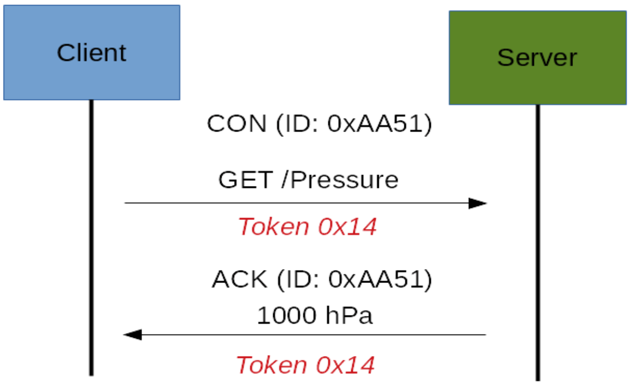


Figure 13: **Piggy-backed request/response transmission**

As you can notice in the CoAP message, there is a Token. The Token is different from the Message-ID and it is used to match the request and the response.

4.2.2 Scenario 2: Separate response

If the server can't answer to the request coming from the client immediately, then it sends an Acknowledge message with an empty response. As soon as the response is available, then the server sends a new Confirmable message to the client containing the response. At this point, the client sends back an Acknowledge message:[37]

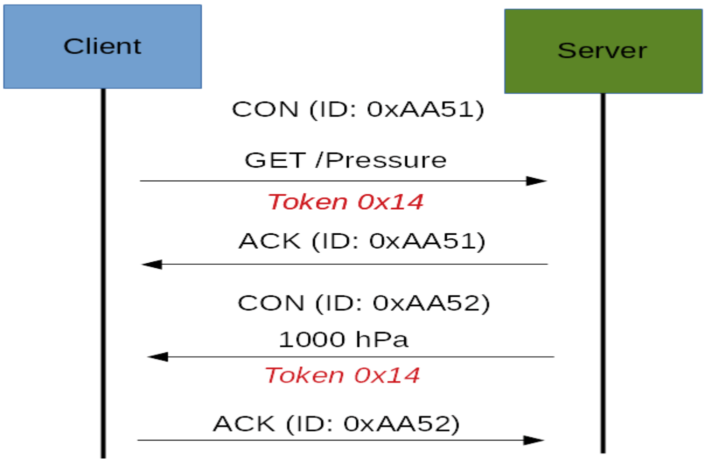


Figure 14: **Separate request/response transmission**

4.2.3 Scenario 3: NON-confirmable response

If the request coming from the client is carried using a NON-confirmable message, then the server answer using a NON-confirmable message.

4.2.4 CoAP request methods:

CoAP supports the basic methods of GET, POST, PUT, DELETE, which is easily mapped to HTTP.

As CoAP methods manipulate resources, they have the same properties of safe (only retrieval) and idempotent (you can invoke it multiple times with the same effects) . The GET method is safe, therefore it MUST NOT take any other action on a resource other than retrieval. The GET, PUT and DELETE methods MUST be performed in such a way that they are idempotent. Unlike PUT, POST is not idempotent because the URI in the request indicates the resource that will handle the enclosed body. This resource indicated by the POST may be used for data processing, a gateway to other protocols and it may create a new resource as a result of the POST.

GET: The GET method retrieves the information of the resource identified by the request URI. Upon success a 200 (OK) response SHOULD be sent. The response to a GET is cacheable if it meets the requirements in caching.

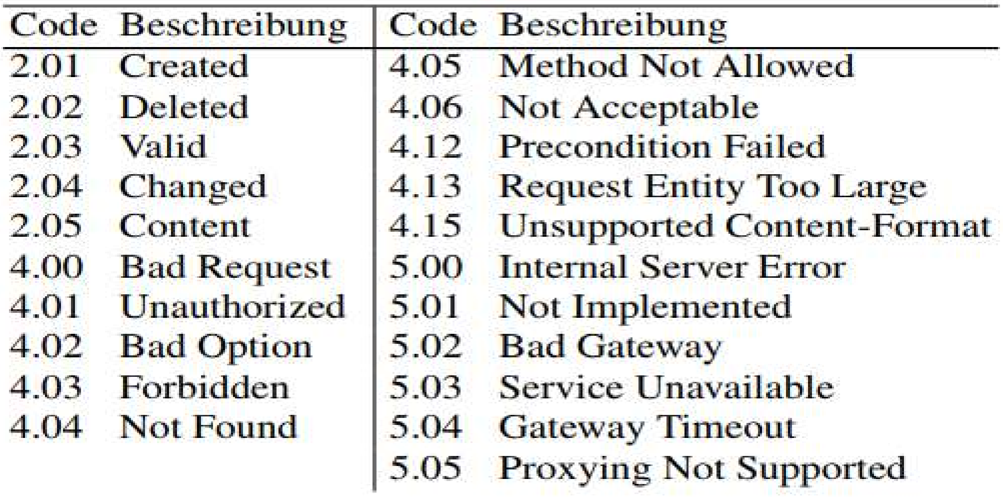
POST: The POST method is used to request the server to create a new subordinate resource under the requested parent URI. If a resource has been created on the server, the response SHOULD be 201 (Created) including the URI of the new resource in a Location Option with any possible status in the message body. If the POST succeeds but does not result in a new resource being created on the server, a 200 (OK) response code SHOULD be returned. Responses to this method are not cacheable.

PUT: The PUT method requests that the resource identified by the request URI be updated or created with the enclosed message body. If a resource exists at that URI the message-body SHOULD be considered a modified version of that resource and a 200 (OK) response SHOULD be returned. If no resource exists then the server MAY creates a new resource with that URI, resulting in a 201 (Created) response. If the resource could not be created or modified, then an appropriate error response code SHOULD be sent. Responses to this method are not cacheable.

DELETE: The DELETE method requests that the resource identified by the request URI be deleted.The response 200 (OK) SHOULD be sent on success.Responses to this method are not cacheable.

4.2.5 CoAP Response codes

Response codes are similar to HTTP. For instance, 2.xx indicates success, 4.xx indicates client error and 5.xx indicates a server error. Although some response codes match the HTTP status codes (e.g., 4.04 and 404” Not Found”), others have different codes (2.05 ”Content” is equivalent to 200 ”OK”, but 2.05 is only used in response to GET) or are not represented at all.



Code	Beschreibung	Code	Beschreibung
2.01	Created	4.05	Method Not Allowed
2.02	Deleted	4.06	Not Acceptable
2.03	Valid	4.12	Precondition Failed
2.04	Changed	4.13	Request Entity Too Large
2.05	Content	4.15	Unsupported Content-Format
4.00	Bad Request	5.00	Internal Server Error
4.01	Unauthorized	5.01	Not Implemented
4.02	Bad Option	5.02	Bad Gateway
4.03	Forbidden	5.03	Service Unavailable
4.04	Not Found	5.04	Gateway Timeout
		5.05	Proxying Not Supported

Table 4: CoAP response codes

5 MESSAGES FORMAT

in this paragraph, we present the CoAP message format. The constrained application protocol is fundamental for restricted environments, and for this reason, it uses built-in messaging. To shun fragmentation, the message occupies the data section in the UDP datagram.

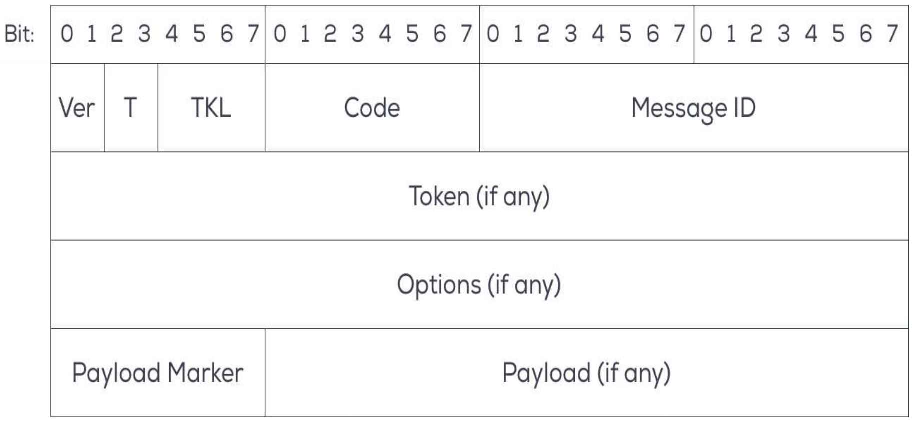


Figure 15: **CoAP message format**

The CoAP header has been designed to be easy to analyze by programs running on small devices such as sensors (Figure 15). , the CoAP header begins with a fixed four-byte portion, which includes

Protocol version (**Ver**): this is the version of the CoAP protocol, namely 1

The type of message (**T**): Indicates that the message is of type Confirmable (0), Non-confirmable (1), Acknowledgment (2), or Reset (3).

Token length (TKL): Indicates the length (variable) of the Token field (0-8 bytes)

Code: 8-bit integer, separated into 3 class bits (the most significant bits) and 5 detail bits (the least significant bits), documented in the form c.dd or "c" is a digit between 0 and 7 (3 bits) and "dd" represents two digits between 00 and 31 (5 bits). The class can indicate a request (0), a success response (2), a client error response (4), or a server error response (5). All other class values are reserved. A special case is that of an empty message whose code is 0.00. In the case of a request, the Code field indicates the method. 0.01 = GET; 0.02 = POST; 0.03 = PUT; 0.04 = DELETE.

Message ID : a 16-bit integer used for reliable transmission, duplicate detection and to match ACK/RST to corresponding CON/NON-messages [36] Note that with these two bytes, we are limited to approximately 250 messages per second (due to the EXCHANGE_LIFETIME parameter which is 247 seconds by default). This is not a very serious limit: the resources of CoAP machines do not allow them to be too talkative in any case.

Token: 4-bit unsigned integer. Indicates the length of the variable-length Token field (0-8 bytes)[37].The token value is used to correlate a request and its response.

Option: In the options field, if any, it must contain the option number, option value length, and the value itself. The option number is not declared specifically, rather is calculated by the equation: option number = option delta + previous option number.

Option delta is used to establish the difference between current option number from its previous one.

Option length simply indicate the size of the option value.

Option value is a sequence of ‘L’ length, define by the length field and can contain the following formats: empty (zero), opaque, uint (option length) or string(UTF-8).

The options field has two different classes that define the way unrecognized options are to be handled by endpoints. Those classes are either critical or elective .

CoAP defines several options used on both request and responses:

Option	Role
Content Format	determines the representation format of the message payload.
ETag	or entity-tag defines a resource-local identifier that can distinguish between similar representation of the same resource that vary over time.
Location-Path and Location-Query:	defines an absolute path, a query string or both that specifies the new location and/or query argument of a resource created with a POST request.
Max-Age	indicates a maximum time in which the response cached can be considered fresh.
Proxy-Uri	it refers a request to a forward-proxy. The forward-proxy is requested to forward the request or service it from a valid cache and return the response
Proxy-Scheme	it is used to assemble an absolute-URI in a proxy request. The absolute-URI is constructed from Uri-Host, Uri-Port, Uri-Path, Uri-Query options
Uri-Host	determines the Internet host of the resource being requested.
Uri-Path	determines one segment of the absolute path to the resource
Uri-Port	defines transport-layer port number of the resource.
Uri-Query	specifies one argument to describe the resource.
Accept	clients can use this option to indicate the acceptable contentformat to get from a response.
If-Match	it may be used to make a request conditional on the current existence or value of an ETag for one or more representations of the target resource.
If-None-Match	opposite to If-Match, If-None-Match may be used to make a request conditional on the non-existing target resource.Auxiliary to PUT request to avoid any accidental resource overwrite,especially if the same resource is being used by multiple clients.
Size1	mainly used in block-wise transfer, it determines the resource representation size in a request.

Table 5: CoAP defines several options used on both request and responses

the payload: The payload has a set of one byte that is called the Payload Maker, which marks the start of the payload data. If the Payload Maker has a value of all ones (0xFF16), there is data present, otherwise, the payload is empty.

6 AN EXAMPLE OF COMMUNICATIONS IN CoAP

step1: The client sends a Confirmable GET request for the resource CoAP://server/temperature to the server with a Message ID of 0x7d34. The request includes one Uri-Path Option (Delta 0 + 11 = 11, Length 11, Value "temperature").

Step2: A 2.05 (Content) response is returned in the Acknowledgement message that acknowledges the Confirmable request,echoing both the Message ID 0x7d34. The response includes a Payload of "22.3 C" and is 11 bytes long.

Step3: The client sends a GET request with a message ID 0x7d35, of confirmable type CON,and with a uri-query = / humidity.

Step4: the Confirmable GET request is lost. After ACK TIMEOUT second, The client retransmits the request.

Step5: the server acknowledges the Confirmable request and sends a 2.05 (Content) response of type ACK acknowledgment, and with the response to the request as payload .

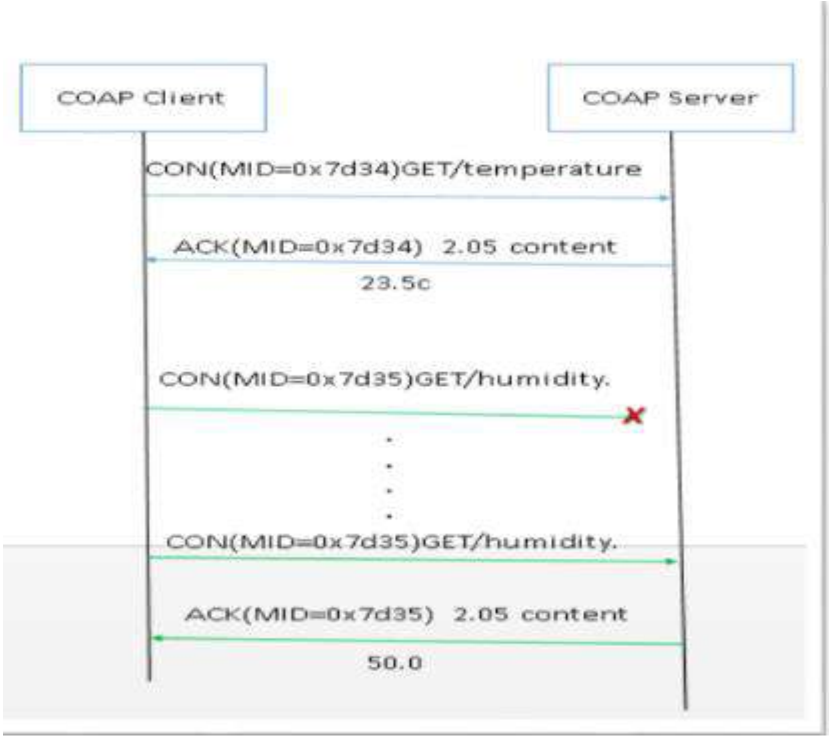

COAP Client
COAP Server
CON(MID=0x7d34)GET/temperature
ACK(MID=0x7d34) 2.05 content
23.5c
CON(MID=0x7d35)GET/humidity.
CON(MID=0x7d35)GET/humidity.
ACK(MID=0x7d35) 2.05 content
50.0

Figure 16: **CoAP example communication**

7 WHEN TO USE CoAP PROTOCOL

Some of the specific situations in which CoAP is useful are[40]:

The device cannot run HTTP or TLS: If this is the case, then turning on CoAP and DTLS can practically do the same as HTTP. If one is an expert with HTTP APIs, the migration is simple. Receive read GET, POST, PUT and DELETE mutations and security turns on DTLS.

Use battery in device: If this is a problem, running CoAP will improve battery performance when compared to HTTP over TCP / IP. UDP saves some bandwidth and makes the protocol more efficient.

Subscription required: If no one can run MQTT and HTTP Polling is impossible, CoAP is a solution.

8 PRACTICAL APPLICATIONS OF CoAP PROTOCOL

CoAP has been successfully implemented in many areas including[41]:

Domainindustrial	logistics and product life management	Purchases
		Fastpayment
		Identify the equipment
Domainhealth and well-being	medical health care	Altalmatmedical
		Monitor medicalequipment
		Smart hospital services
domain smart city	public safety and environmental monitoring	Video surveillance
		emergency plan
		Monitor employees
	smart homes and buildings	Lighting
		Energy management
		child protection

Table 6: Practical Applications Of CoAP Protocol

9 PROTOCOL CHARACTERISTICS

9.1 Proxying

The COAP proxy is designed to brace applications that need to interact with WSN nodes, such as smart city development.

figures 17 and 18 illustrate the network architecture and the protocol stack and the role of the proxy.

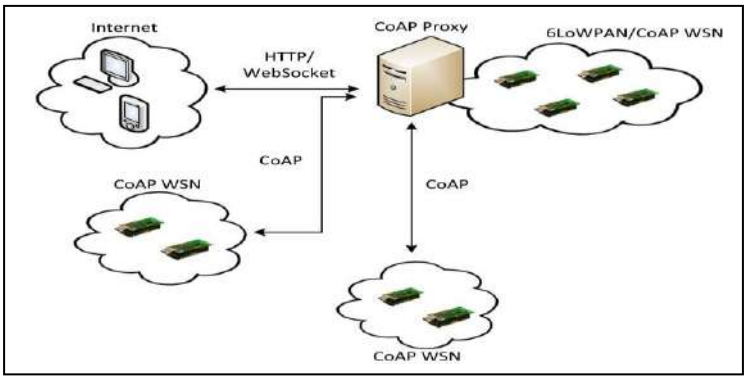


Figure 17: **Network Architecture Using Proxy**

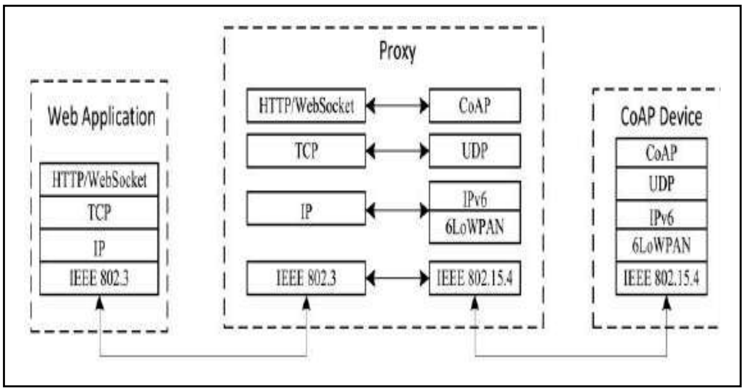


Figure 18: **Protcol Stack**

The Constrained Application Protocol (CoAP) proxy allows for adapting the protocol stacks of Web applications and CoAP devices.

The CoAP proxy also has the IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) edge router and gateway functions to interconnect disjointed CoAP networks.

A proxy serves as an intermediary that relays and/or forwards information between endpoints. Proxy is quite essential, especially in a constrained environment, by tasking another CoAP node to act in their behalf and performing either their request or their response it can consequently improve network performance, grant access to sleeping devices and limits energy consumption, bandwidth, and network traffic.[36]

There are two types of proxies on a CoRE structure: forward-proxy, selected by a client, and reverse-proxy, selected by the origin server, the similarity between CoAP and HTTP protocol offer a convenient way to implement proxy between this two protocols.That is how to devices implements the two protocols can easily connected throw the Proxy. There are two types of cross-protocol proxying :

ProxyingCoAP-HTTP: Since CoAP methods are equivalent to HTTP methods, it allows access to HTTP server resources for CoAP clients, HTTP, TCP and optionally TLS can be created easily.[37]

HTTP-CoAPProxying: allows access to the resources of a CoAP server for HTTP clients. the client must specify the absolute path to the resource including the schema (CoAP / CoAPs) in the method invocation for To send an HTTP request to the proxy. Once the proxy has delivered the message, it will request the specified CoAPresource[37].

9.2 Caching

The goal of caching in CoAP is to reuse a prior response message to satisfy a current request. In some cases, a stored response can be reused without the need for a network request, reducing latency and network round-trips; a "freshness" mechanism is used for this purpose When a response is "fresh" in the cache, it can be used to satisfy subsequent requests without contacting the origin server, thereby improving efficiency.[37]

Even when a new request is required, it is often possible to reuse the payload of a prior response to satisfy the request, thereby reducing network bandwidth usage; a "validation" mechanism is used for this purpose, When an endpoint has one or more stored responses for a GET request, but cannot use any of them (e.g., because they are not fresh), it can use the ETag Option in the GET request to give the origin server an opportunity both to select a stored response to be used, and to update its freshness. [37]

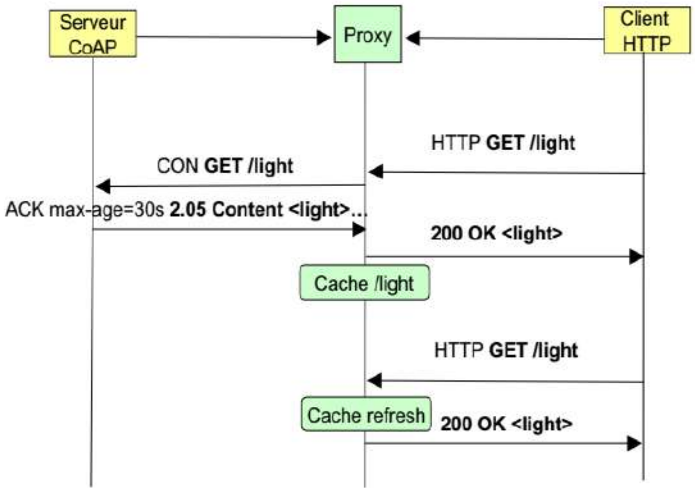


Figure 19: **Caching In CoAP**

9.3 Resource Discovery

In machine-to-machine (M2M) applications where there are no humans in the loop, it is important to find a way to discover resources of a constrained server.

The Resource Directory stores descriptions of the resources held by the servers. So customers can discover all the necessary resources in a single request. To use the DR, either for recording or for searching, the device must know how to reach it. Endpoint Note 8 can locate the DR in several ways. Either the terminal point has the address of the RD in static in its firmware and discovers it at startup, or by the Edge Router which transmits the information during the router advertisement (for example the default route if the RD is installed in the router) or by using the CoRE Link Format by doing a GET / .well-known / core? rt = core.rd *

The example in Figure 3 present a client requesting the list of the obtainable resources of the server (GET /.well-known/core). The returned list (in CoRE Link Format) shows that the server has, among others, a resource called /s/t that, when queried, returns the temperature in degrees Celsius. The client then requests the value of this resource (GET /s/t) and receives a plain text reply from the server with the value of the actual temperature as payload of the message[42].

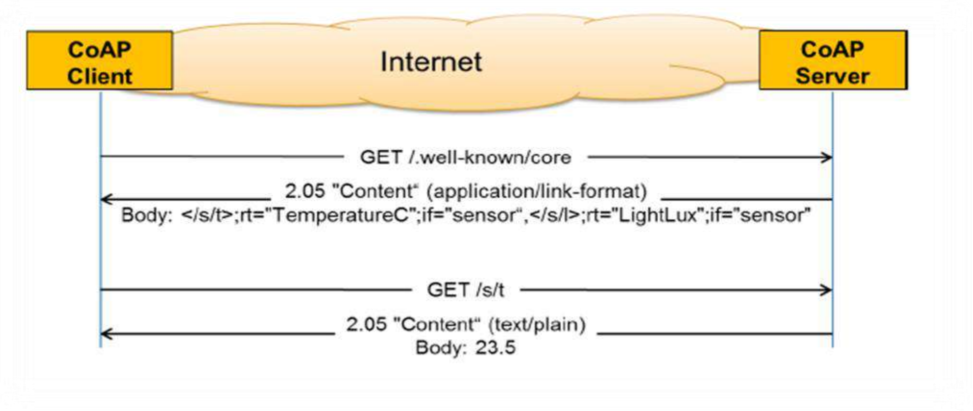

CoAP Client
Internet
CoAP Server
GET /.well-known/core
2.05 "Content" (application/link-format)
Body: </s/t>;rt="TemperatureC";if="sensor",</s/l>;rt="LightLux";if="sensor"
GET /s/t
2.05 "Content" (text/plain)
Body: 23.5

Figure 20: **An example direct**

However, direct resource discovery is impractical in many M2M scenarios in which nodes may have prolonged sleep periods. To solve this problem, you can use CoRE Resource Directories (RD) that host descriptions for resources on other servers. This way the CoAP server can register its resources with one or more RDs. Clients in turn can discover these resources by performing searches against RD.[43][44].

For example the same resource discovery that was performed by using direct communication between the client and the server in Figure 20 can now be performed by using an RD as illustrated in Figure 21.

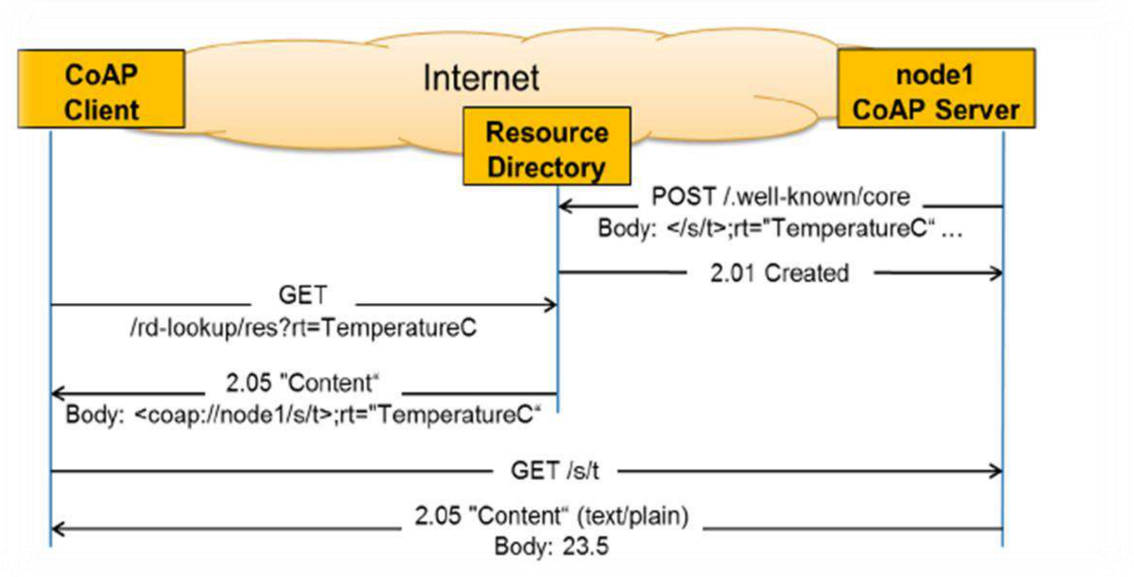

CoAP Client
Internet
Resource Directory
node1 CoAP Server
POST /.well-known/core
Body: </s/t>;rt="TemperatureC" ...
2.01 Created
GET
/rd-lookup/res?rt=TemperatureC
2.05 "Content"
Body: <coap://node1/s/t>;rt="TemperatureC"
GET /s/t
2.05 "Content" (text/plain)
Body: 23.5

Figure 21: **An example resource discovery by using a Resource Directory.**

10 SECURING CoAP

This section defines the DTLS binding for CoAP.

There are three main elements when considering security, namely integrity, authentication, and confidentiality.constrained environment brings a few challenges and implementations that need to be taken into consideration due to the unreliable transport protocol and device limitations. Consequently, not all cipher suites are suitable for some nodes, adding complexity and initial handshake overhead, which can take 4 to 6 messages exchange before establishing the connection[39].

CoAP is secured using Datagram TLS (DTLS) over UDP.DTLS in the application layer protect end-to-end communication. DTLS also avoids cryptography overhead problems that occur in lower-layer security protocols. DTLS solves two problems reordering and packet loss.[37].

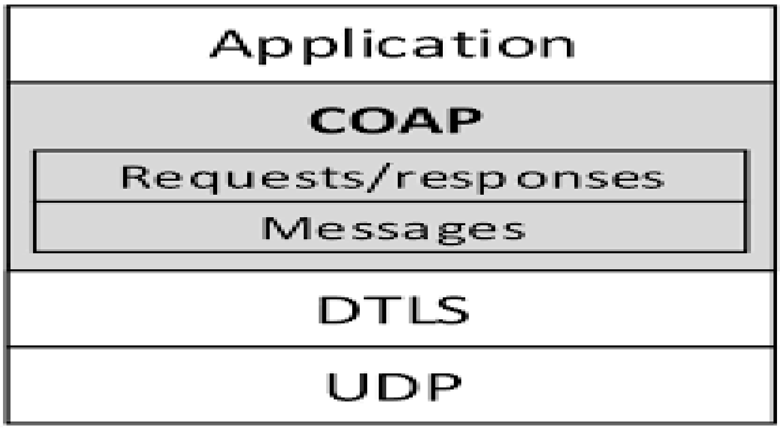


Figure 22: **Abstract Layering of DTLS-Secured CoAP**

CoAP device is provided with the security information that it needs, including keying materials and access control lists [36]. CoAP defines four security modes in which a CoAP the device operates, with NoSec and RawPublicKey mandatory to implement:

- **NoSec:** DTLS is not used. Alternatively, CoAP can be used with IPsec.
- **PreSharedKey**: DTLS is enabled, there is a list of pre-shared keys, and each key includes a list of which nodes it can be used to communicate with as described.
- **RawPublicKey (RPK**): DTLS is enabled, An asymmetric key pair is used, but without a certificate. The public keys.of a device are stored, for example, in the firmware ”raw” (without X.509v3 structure) and can be updated. The device.also has an identity calculated from the public key and a list of identities of the nodes it can communicate with.[36][38].

- **Certificate:** DTLS is enabled and the device has an asymmetric key pair with an X.509 certificate that binds it to its subject and is signed by some common trust root The device also has a list of root trust anchors that can be used for validating a certificate[36].

It is also important that during a client and server interaction using DTLS a client request, of any type, should have a matching session, timestamp and Message ID, and the same goes for the response to the request.It is also important that during a client and server interaction using DTLS a client request, of any type, should have a matching session, timestamp and Message ID, and the same goes for the response to the request.

11 CoAP AND IOTASPACT

There are many factors that allow us to analyze and evaluate the performance of the protocol (evaluate) and we will take some of them. That are going to be discussed here are:

Network Environment ,Performance , Energy Consumption ,Cost Efficiency ,Interoperability ,Scalability , Reliability ,Security and Alternative Application Protocols[36] .

11.1 Network Environment

The suitability of a protocol is highly dependable on the network environments and nodes present in said network. Logically, analyzing the type of network, its conditions, limitation and requirements to be reached, can influence the overall performance of the protocol, especially, if all or most needs are correctly met. CoAP, as mentioned above, is a standard that thrives in low-power, lossy networks types, where devices are 8-bit microcontrollers with minimum ROM and RAM, with high packet rate loss and throughput of 10 Kbit/s. Thus, CoAP is suitable for both 6LoWPAN, LLN and appears to be appropriate to be used on short distance (less than a mile) networks .

Because CoAP is based on REST architecture and share method and response codes, integration to the existing Web infrastructure is much simpler with CoAP. Because of both HTTP based applications and CoAP based applications are REST-based, IoT and Internet devices can simply use cross-proxy that easily maps the request/response model from one standard to another and thus avoiding the complexity from application gateway implementations.

Although CoAP optimizes RESTful architecture for M2M applications and supports and adjusts some HTTP features for constrained devices, this protocol lacks in maturity. CoAP is not fully yet mature as a standard and the protocol is still relatively new which means that other protocols, more well-known and widely deployed in the Internet, might take precedence over CoAP. However, for a relatively new protocol, CoAP is gaining rapid visibility, with companies, with success interoperability tests and important add-ons ratifications, CoAP seems to have a future as a key protocol in the development of IoT.

11.2 Performances

The traditional request/response model, that is implemented by both CoAP and HTTP, is based on regular resource polling, where clients request resources and the server responds to those requests. However, polling can be detrimental to the constrained network if a client demands, periodically, to have current resource representation. In HTTP this is solved with repeated polling, or long polling, the clients send periodically request messages to the server in order to receive an updated resource. The server, in turn, only responds when the updated version is available. This mechanism does solve a few latenciesissues and optimizes processing and network resources for long-lived devices. For constrained devices, however, this is not the most optimal solution as devices have limited construct. This can be an issue, especially for a CoAP proxy that communicates with CoAP devices and Web applications. CoAP proxy may support HTTP long polling, although with multiple Web applications trying to get data from the constrained network can cause an increase in overhead, latency, and increase in network traffic. Overall the extra complexity and added overhead can slightly decrease performance.

Just like any other web protocol, CoAP adheres to congestion control a mechanism to maintain the network stable and regulate the number of messages flowing on network and nodes. Despite this, the CoAP congestion control definition is quite simple and only takes into consideration confirmable messages. Consequently, non-confirmable messages transmission can overrun the network. This protocol drawback is yet to be adjusted with future consideration, such as drafts for Advance congestion control, that aims to optimize mechanisms that will have a higher performance.

11.3 Energy Consumption

As many of the devices that composes IoT environments are deprived of constant sources of energy and relies on battery or some kind of energy harvesting, one of the many challenges faced by IoT devices is to provide an efficient system of protocols that minimizes power consumption while keeping the same level of performance and capabilities as any other Internet node.

On a constrained node network the availability of power or energy of a device varies. Nodes on constrained environments will have at least one of the following energy limitation type:

- **Event energy-limited:** delimits an amount of power that is solely used for a determined event, such as a button being pressed.

- **Period energy-limited:** have either accessibility to maintenance, where their main battery is replaced, or have means to harvest energy.

- **Lifetime energy-limited:** have a certain amount of power available along its lifetime, as there is no other way to recharge or change its main battery.

Power management, generally, is more of an issue on Physical and MAC layers rather than the Network, Transport and Application layer. The increase of smart objects on the Internet made that some changes were needed to better reduce energy consumption. CoAP, for instance, does take into account the lack of an unlimited source of energy supply by constrained devices. In that matter, CoAP, provides a few implementations that can help reduce the energy used in both transmission and processing of data showing good results in comparisons to HTTP, The use of proxies and caching can be used to both reduce the response time and energy consumption. Moreover, the simplicity of UDP and the descriptive but short URIs lower the processing time, thus also diminish the energy usage.

11.4 Cost Efficiency

The main advantages to reduce cost on a network is to deploy standardized protocol solutions, just like CoAP, as a non-standardized solution would breach the end-to-end Internet principle. Taking away the end-to-end principle would result in the need to translate from standardized Internet protocol to proprietary protocol in the last few meters. Thus introducing application gateways, which not only adds to the complexity of the network but is time-consuming and costly to maintain, install and operate ,Another advantage is that CoAP makes use of the CoRE link service/resource discovery. This is beneficial as it facilitates the incorporation of new devices on the network, replacements, and expansion at a minimum cost as devices can use the resource and service discovery to find all the needed information to attach itself to the network, which means maintenance of devices is cheaper as there is no need to installation and manual configuration.

11.5 Interoperability

The standardization of Internet protocols is still an ongoing process, especially for IoT devices. IETF CoRE working group has done and still does major efforts to standardize CoAP. The standardization process of Internet protocols plays a critical role in IoT ecosystems, enabling interoperability between many heterogeneous IoT devices, applications, and networks. Interoperability is a vital functionality to secure the survival of a protocol on the WoT.

CoAP is an open-source web protocol, therefore its source code is made available to anyone to copy, modify and redistribute the protocol. The main advantage of making CoAP open-source is that it can solve its maturity issue as it encourages both IoT developers and industry to experiment with the protocol, thus leading to its adoption. However, a possible downside to the open-source is that it can root to many different implementations of the protocol hence not being able to guarantee interoperability. In a similar manner, an open-source protocol has the main leverage of code transparency, supporting changes to be made in the code to support interoperability.

11.6 Scalability

In terms of scalability, CoAP does provide many features that set it apart from other protocols and does make it a better fit for IoT. Particularly, CoAP supports device, service and resource discovery. The discovery system provided by the CoRE link format enables a flexible mechanism that allows device mobility and scalability. Because nodes are able to perform an automated task that enables them to find other devices and resources, it facilitates the replacement or inclusion of nodes on the network.

11.7 Reliability

The reliable protocol provides notification on data delivery status to the intended recipients and with the adoption of UDP as the main transfer protocol, reliability in CoAP is achieved through Confirmable messages that expect an Acknowledge in return. However, this mechanism verifies that a message has arrived at the right recipient but it does not give any indication that the message delivered was successful and without errors. Therefore reliability in CoAP is minimum and optional. As reliability is optional, some message might be marked as Non-confirmable and does not need to be acknowledged, but CoAP does use Message ID in both types of message to avoid duplicates.

11.8 Security

Similar to any other web protocol, CoAP is vulnerable to Internet attacks:

Denial of Service (DoS), eavesdropping, spoofing, and many others.

Nevertheless, CoAP is not depleted of security solutions, rather the opposite. This protocol does offer multiple security modes through its DTLS binding. One of the main issues with security in constrained devices is that because of device limitation and strain for resources, security might not be a top priority, thus not being implemented in all situations as the lightness of messages is given preference. This rather than adding the additional overhead from security implementation to the already strained network and device .

A more related security issue with CoAP is that it adheres security protocol:

DTLS. This protocol does provide reasonable transport layer security but it does also have a few issues that still have not been dealt with completely. First, some DTLS cipher suite implements initial handshakes procedures that add to both complexity and overhead that constrained devices to try to avoid.

The second issue is that there is no concrete definition of DTLS for multicast communication or any other security measure at all that can provide security for group communication.

IETF’s Smart Object Lifecycle Architecture tackles many of the security issues in constrained environments for Constrained Environments (SOLACE). There are drafts and proposals that aim to specify better security implementation in CoAP. Some of those ongoing proposals make a recommendation for security in multicast communication, while another addresses the possible reduction of DTLS overhead and some deal with alternative security protocols, such as IPsec.

12 CONCLUSION

The purpose of this chapter is to present a theoretical study of the CoAP protocol, where we presented the most important elements, including its features, structure, and characteristics, as well as we made a theoric comparison of the protocol with the characteristics of the Internet of things.

In the next chapter, we will present the simulation of CoAP Protocol.

Chapter3 :

COAP PROTOCOL SIMULATION

1 INTRODUCTION

The simulation consists of modeling the whole of the studied system and to simulate it numerically using environments resulting from measurements on a real system or probabilistic models [46]. The advantage of simulation is to be able to work on unavailable systems. For example, during the design stage, it is much less expensive to carry out a preliminary simulation of the considered alternatives. In addition, simulation is a very flexible way to study a problem. This technique allows program reruns with parameter changes and execution trace taking without the unpredictable disturbances of a real environment.

In this chapter, we will present some of the evaluation methods, then we explaine the simulation of the CoAP protocol according to the proposed scenario (Smart Street lighting), and the tools used. We will do different experiment simulations for CoAP by changing and increasing the number of servers, packet size, and transmission data interval.

2 VALIDATION METHODS

There are different techniques for evaluating the performance of a system on a WSN. Among them, we can cite analytical modeling, measurements obtained from real experiences, or simulation.

2.1 Analytical methods

It proposes analytical methods to study the behavior of a system by solving the mathematical equations on which its mathematical model is based. The importance of analytical methods is mainly in solving equations that are generally inexpensive at computing time. Moreover, the analytical methods make it possible to obtain a good understanding of the functioning of a system, since one is more able to analyze some of its imbalances by solving his model, and thus proposing modifications to solve them such as Formal methods.[45]The analytical method in scientific literature requires a proof, exemplary examination, and quantitative measures.

2.2 Real Experience

The validation of protocols and applications from real tests is complex to implement. In fact, a problem is difficult to study from real experiences for several reasons: Experiments are difficult to reproduce. , It can be disturbed by external factors and the experimenter cannot control it, the study of increase, decrease and variation of speed and pattern of movement is a complex process.[46]The disadvantage of real experience methods is that it is generally necessary to make restrictive assumptions about the real system in order to be able to obtain workable models, and since our subjects do not require real application because it is a study of performance only, this method was not chosen.

2.3 Simulation

A discrete event simulation consists in reproducing the behavior of a system by studying a specific perception of its model. The advantage of simulation is to offer a very general approach that makes it possible to study any model, as long as the simulation tool adapts to the model under study. On the other hand, its disadvantage is that it requires a lot of machine calculation time.[47]

There are several separate event simulators. Among them, let us quote the NS-2 and NS-3 network simulators and OMNeT, which allow simulation of different types of networks including wait networks, OPNET and COOJA is a tool for performance studies.

COOJA, being the default network simulator for Contiki came natively bundled along with Contiki 3.0. COOJA has a good GUI environment and allows for quick simulation setups and analysis. While our topic requires a study of performance, COOJA, therefore, was found as one of the best to simulate the protocol, due to its flexibility, extensibility, and quick prototyping.

3 METHODOLOGY

In order to assess the performance of the CoAP protocol, we will base our evaluation on a scenario of smart street lighting. We simulate different examples by controlling variables (data tansmission interval, number of nodes, packet size) using the COOJA simulator and Eclipse Californium. In order to obtain measures results we changed some files in the Contiki system (rpl-icmp6.c, mrhof.c , ….), and we created a client in Eclipse called CoAPClient.java , and by using Python, we will get the results and make them curves to facilitate evaluation and analysis.

4 TOOLS USED IN THIS SIMULATION

In order to achieve a simulation of the COAP protocol we need to know about the used tools

4.1 Software

- ✓ **UBUNTU 19.01**
- ✓ **Eclipse Californium**

Californium is a powerful CoAP framework targeting back-end services communicating with smaller Internet of Things devices. Stronger Internet of Things devices may use Californium as well. It provides a convenient API for RESTful Web services that support all of CoAP’sfeatures[48].

- ✓ **ContikiOS [49]**

  Contiki is an open-source operating system that runs on tiny low-power microcontrollers and makes it possible to develop applications that make efficient use of the hardware while providing standardized low-power wireless communication for a range of hardware platforms.

  Contiki is used in numerous commercial and non-commercial systems, such as city sound monitoring, street lights, networked electrical power meters, industrial monitoring, radiation monitoring, construction site monitoring, alarm systems, remote house monitoring, and so on.

- ✓ **Wireshark**

  Wireshark is a network packet analyzer that tries to observe the messages exchanged between executing protocol entities which is an open-source software project and is released under the GNU General Public License (GPL).[49]

- ✓ **COOJA**

Is a wireless sensors network simulator depend on Contiki operating system. It is a flexible Java-based simulator that supports using C language to develop application software by Java Native Interface. One of the great advantages of this COOJA simulator is that it can simulate the application software simultaneously in high-level algorithm development and low-level hard driver development. The COOJA simulator has great extensibility. Application developers can alter parts of the simulation environment without changing any COOJA main code. It means that the system can be added to new parts such as interfaces, plugins, and radio mediums or reconfigured existing parts. With these advantages of COOJA, we can implement a variant simulation with different conditions and system settings such as different packet generation rates, different MAC protocols, and different network topology [49].

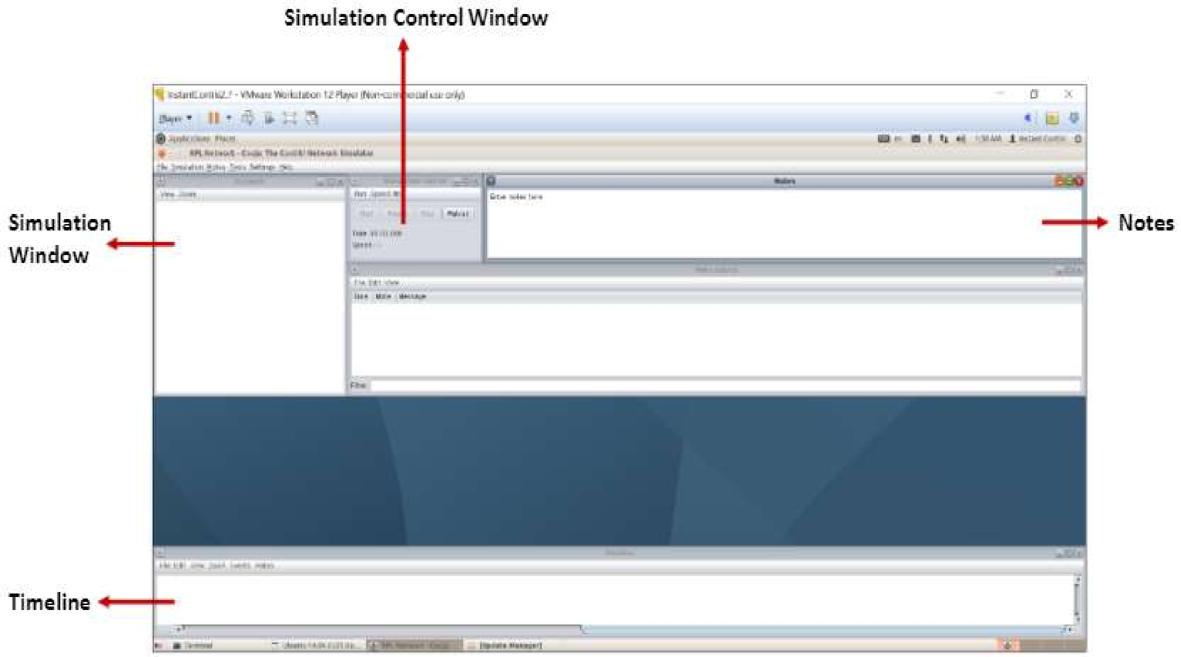


Figure 23: **COOJA simulator Interface**

Simulation window:

In a simulation we have several windows:

- The Timeline window: at the bottom of the screen, displays all the events of communication in the simulation over time, very convenient to understand what is happening in the network.

- The Network window: at the top left of the screen, shows us all the nodes in the simulated network.

- The Timeline window: at the bottom of the screen, displays all the events of communication in time simulation, very convenient to understand what is happening in the network.

- The Mote Output window, on the right side of the screen, shows us all the serial port prints from all nodes.

- The Notes window at the top right is where we can put notes for our simulation.

- The Simulation control window: is where we can start, pause and load our simulation.

- ✓ **C Language**

Is a compiled language (as opposed to interpreted languages). This means that a C program is written as a text file, called a source file. This file is obviously not executable by the microprocessor, it must be translated into machine language. This operation is performed by a program called a compiler [50]. It was used in this work in order to modify the files of the COOJA emulator according to the requirements of the study

- ✓ **Java language**

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere[51].

- ✓ **Python language**

Python is a programming language (like C, C ++, FORTRAN, Java ...), It was developed in 1989. Its main characteristics are as follows: Open source is free to use, source files are available and editable; Equipped with a very extensive base library and a large amount of libraries available for scientific computing, statistics, databases and visualization ... etc and Dynamic writing is done automatically during execution of the program, which allows great flexibility and speed of programming, but it is motivated by excessive memory consumption and loss of performance, providing support for "integrating other languages" [53] . It was used in this work to draw curves resulting from the study.

4.2 Hardware

The characteristics of the computer in which the study is simulated are:

- ✓ Computer Hp 650;
- ✓ Processor: intel(R) core(TM) i3-4005 CPU 64 bait;
- ✓ RAM:6.00GO.

5 SIMULATION ENVIRONMENT

The table (7) summarizes the main parameters of the simulation environment. We are studying the simulation of the protocol. The simulation is adopted by a network of 15 servers. Initially, the nodes are placed in random at 100 x 100 m. The speed of movement of the node varies by a value of 1 m / s. The maximum travel speed and pause time determine how much the model moves. To create a moderate simulation, the pause time is set to 5 seconds. We will simulate different scenarios. Each simulation will take one (1) hour .

The default values for some environment parameters are expressed below:

Parameters	Value
Surface	100x100 m
Phase initialization	1 minit
Time simulation	1 hour
Randomseed (Random speed)	123,456
Mote startup delay	5s
Speed of nodes	1s
Break time	5s
Number of servers	15

Table 7: Simulation Environment

6 SIMULATION SCENARIO: SMART STREET LIGHTING

To evaluate protocol performance, we create a Simplified Smart Street that uses CoAP to communicate with captive devices:

Street lights around the world currently consume a lot of electrical energy, which is automatically turned on in the dark and turns off during the day. So some companies and universities are developing smart street lighting systems with control to reduce energy consumption. Smart streets operate with a self-controlled distributed optical system, whereby the lights turn on when there is grass and turn off or dim in their absence with the distributor's sensor, and provide safety to warn them when there is a danger [54,55].

Figure (24) shows an example of our smart street lighting system. Street lights turn on before pedestrians come and turn off or reduce the energy when there is no one around through a distributed sensor network,

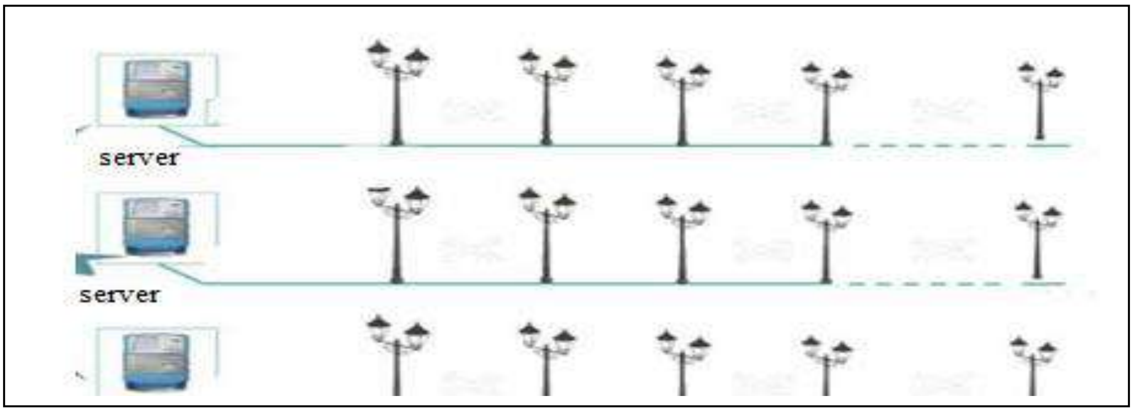


Figure 24: **Smart street lighting**

The smart street contains a set of electrically adjustable LED matrix light poles, a communications device, a controller, surveillance cameras, and Wi-Fi communications. It is switched on for several minutes when movement is detected in the designated area by sensors placed in various locations, such as electric poles, house gates, to ensure that every street lamp is working before pedestrians notice it. Then, it sends the message using COAP, so that each group of converging columns sends data to its nearest server, and the server, in turn, communicates with GETAWAY. In this scenario, COAP performance is studied by criteria (power consumption, throughput, packet loss, and delay). Figure (25) shows the diagram sequence for a smart street lighting system.

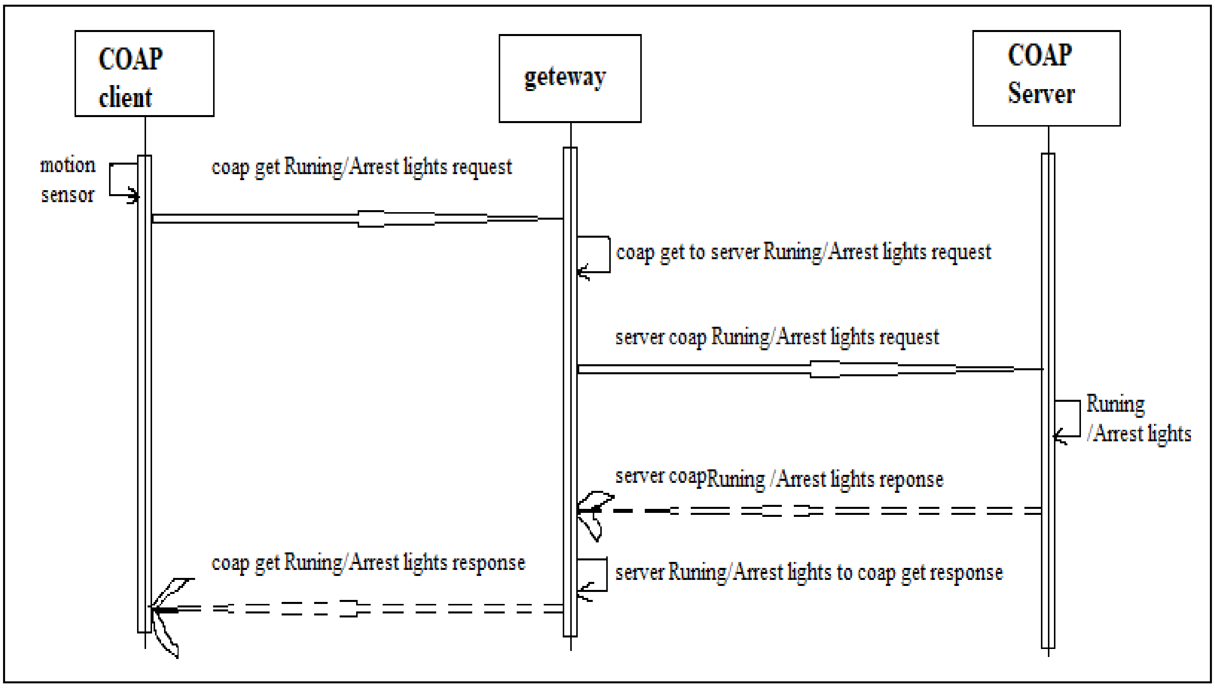

COAP client
geteway
COAP Server
motion sensor
coap get Runing/Arrest lights request
coap get to server Runing/Arrest lights request
server coap Runing/Arrest lights request
Runing /Arrest lights
server coapRuning /Arrest lights reponse
coap get Runing/Arrest lights response
server Runing/Arrest lights to coap get response

Figure 25: **Sequence diagram for RUNING /ARREST from the SMART STREET LIGHTS.**

7 CoAP SIMULATION USING COOJA

In this section, we will run the protocol CoAP in COOJA and make a connection between the Border router and the server, as well as using the Wireshark. Observed The steps are as follows:

- ✓ **Step 1**: We open a new simulation, Open motes menu >> add motes >> create new motes type>>skyThree files are necessary to run CoAP applications. In order to create the motes :border-router.cer-example-server.c, er-example-client.c

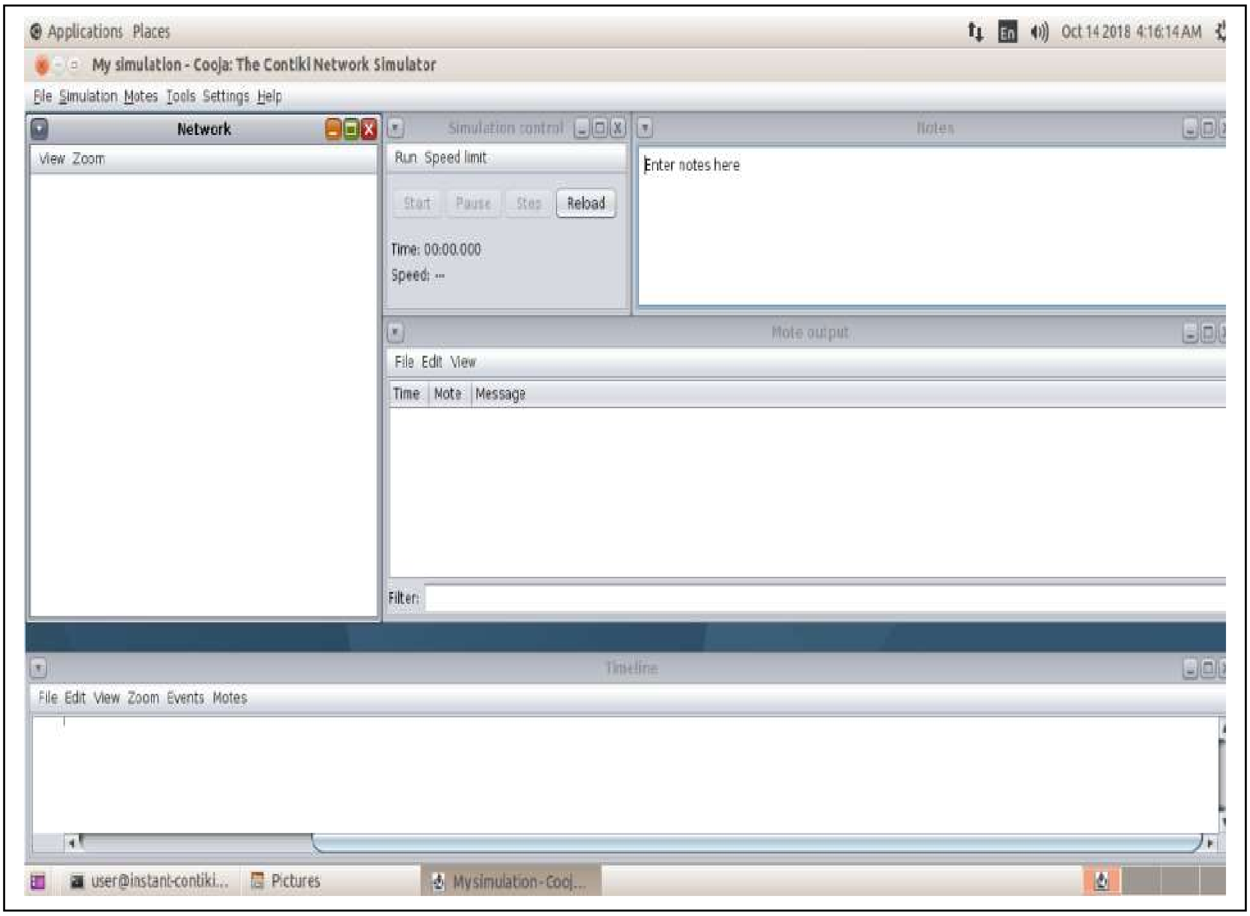
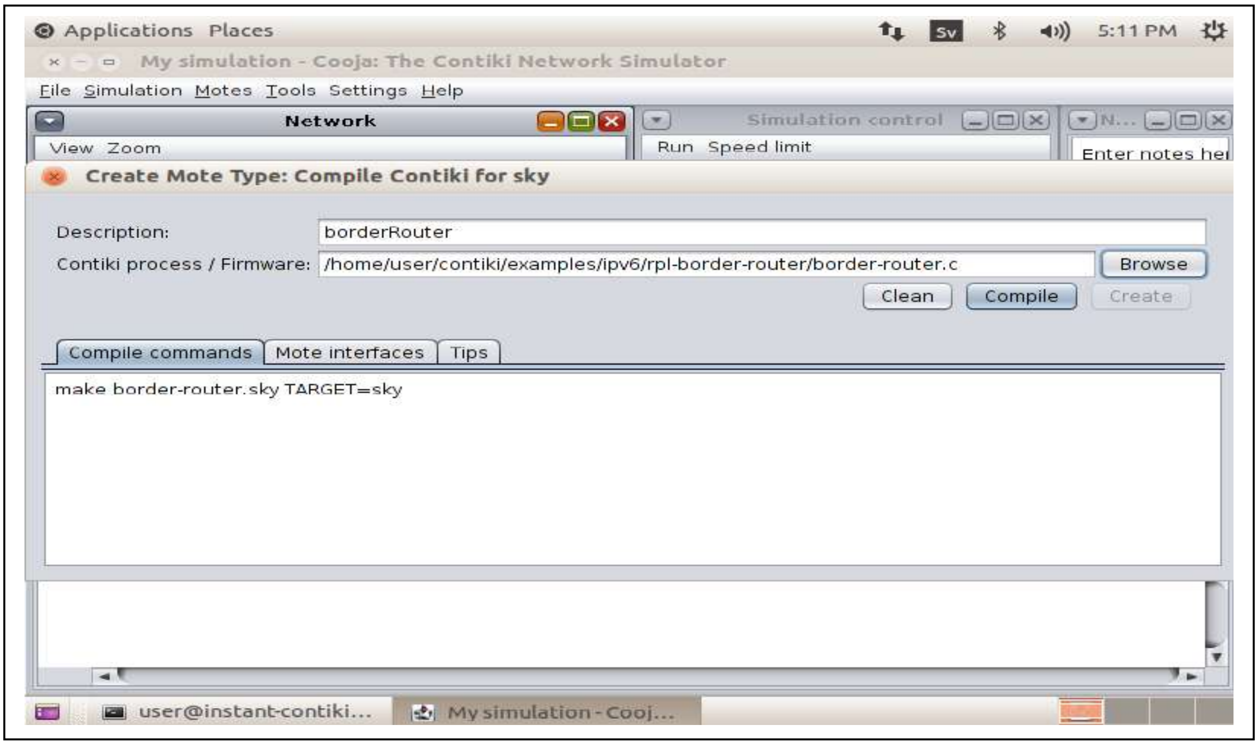


Figure 26: **COOJA window**

- ✓ **Step 2**: To create border router motes: home/user/contiki/examples/ipv6/rpl-border-router/border-router.c, Choose the file in location >> compile >> create >> Add motes.



Create Mote Type: Compile Contiki for sky

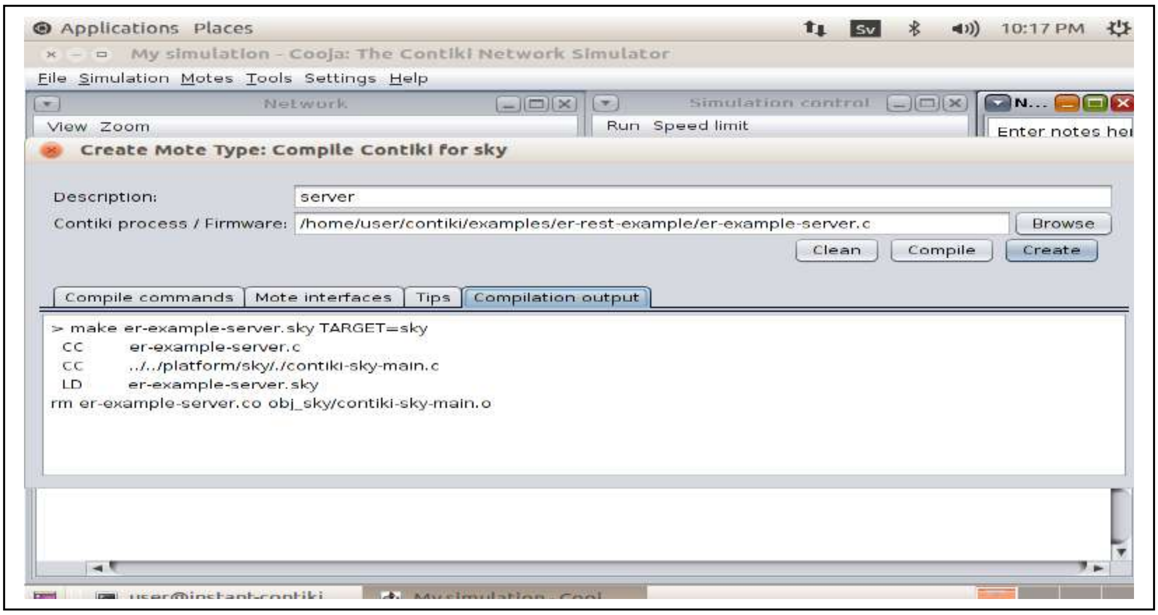
Description: borderRouter

Contiki process / Firmware: /home/user/contiki/examples/ipv6/rpl-border-router/border-router.c

```
make border-router.sky TARGET=sky
```

Figure 27: **Border Router Mote**

- ✓ **Step 3**: To create server motes : /home/user/contiki/examples/er-rest-example/er-example-server.c, choose a file in location >> compile >> create >> choose server >> Add motes.



Create Mote Type: Compile Contiki for sky

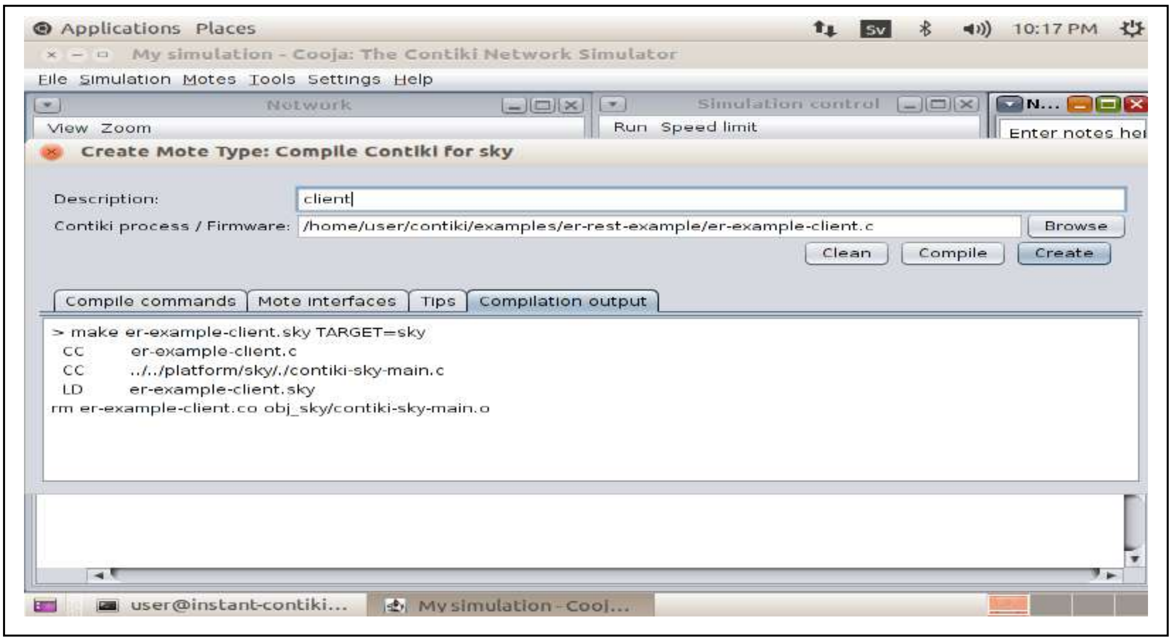
Description: server

Contiki process / Firmware: /home/user/contiki/examples/er-rest-example/er-example-server.c

```
> make er-example-server.sky TARGET=sky
  CC        er-example-server.c
  CC        ../../platform/sky/./contiki-sky-main.c
  LD        er-example-server.sky
rm er-example-server.co obj_sky/contiki-sky-main.o
```

Figure 28: **CoAP server mote**

- ✓ **Step 4**: To create client motes : /home/user/contiki/examples/er-rest-example/er-example-client.c, choose a file in location >> compile >> create >> choose server >> Add motes


Applications Places
10:17 PM
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Network
View Zoom
Simulation control
Run Speed limit
N...
Enter notes her
Create Mote Type: Compile Contiki for sky

Description: client

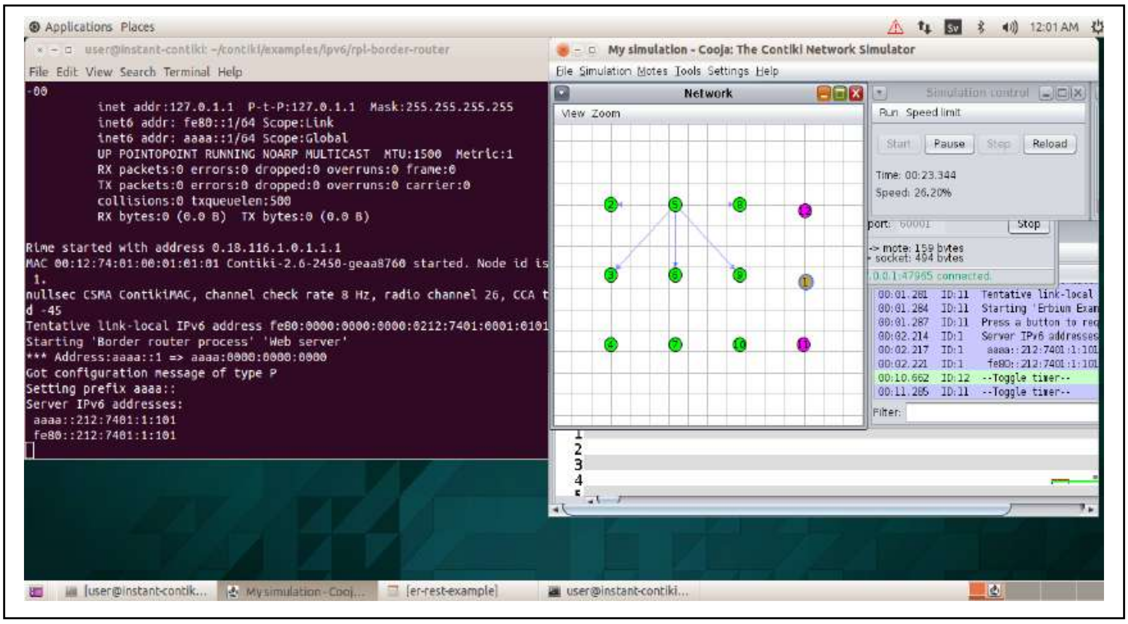
Contiki process / Firmware: /home/user/contiki/examples/er-rest-example/er-example-client.c

```
> make er-example-client.sky TARGET=sky
  CC       er-example-client.c
  CC       ../../platform/sky/./contiki-sky-main.c
  LD       er-example-client.sky
rm er-example-client.co obj_sky/contiki-sky-main.o
```

Figure 29: **CoAP client mote**

- ✓ **Step 5**: And in order to make a connection between the border router and other nodes, we must enable a bridge by Right Click Border Router Node -> Mote tools for Sky 1 -> Serial Socket (SERVER) -> start.

On the other hand, we open a new terminal, and in the following path /home/contiki/ examples/ipv6/rpl-border-router/ we do the command: make connect-router-COOJA.


Applications Places
12:01 AM
user@instant-contiki: ~/contiki/examples/ipv6/rpl-border-router
File Edit View Search Terminal Help

```
-00
          inet addr:127.0.1.1  P-t-P:127.0.1.1  Mask:255.255.255.255
          inet6 addr: fe80::1/64 Scope:Link
          inet6 addr: aaaa::1/64 Scope:Global
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

Rime started with address 0.18.116.1.0.1.1.1
MAC 00:12:74:01:00:01:01:01 Contiki-2.6-2450-geaa8760 started. Node id is
 1.
nullsec CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26, CCA t
d -45
Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7401:0001:0101
Starting 'Border router process' 'Web server'
*** Address:aaaa::1 => aaaa:0000:0000:0000
Got configuration message of type P
Setting prefix aaaa::
Server IPv6 addresses:
 aaaa::212:7401:1:101
 fe80::212:7401:1:101
```

My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Network
View Zoom
Simulation control
Run Speed limit
Start
Pause
Step
Reload
Stop
Filter:

Time: 00:23.344
Speed: 26.20%

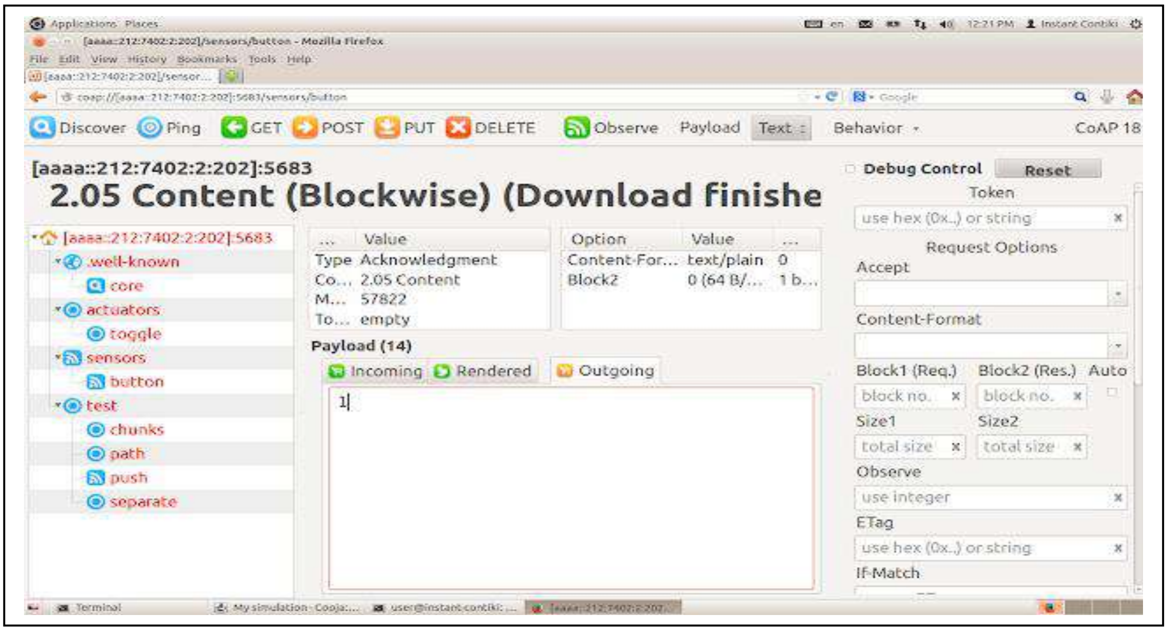
port: 60001

```
-> mote: 159 bytes
- socket: 494 bytes
.0.0.1:47965 connected.
```

```
00:01.281   ID:11   Tentative link-local
00:01.284   ID:11   Starting 'Erbium Exam
00:01.287   ID:11   Press a button to req
00:02.214   ID:1    Server IPv6 addresses
00:02.217   ID:1     aaaa::212:7401:1:101
00:02.221   ID:1     fe80::212:7401:1:101
00:10.662   ID:12   --Toggle timer--
00:11.285   ID:11   --Toggle timer--
```

Figure 30: **Bordr router and CoAP server**

- ✓ **Step 6**: Sensors can be read using ipv6 addresses by opening the Firefox browser. Open the browser and enter the following addresses in a new tab CoAP://[aaaa::212:7401:1:101] or any other sensor mote figure(31).



[aaaa::212:7402:2:202]/sensors/button - Mozilla Firefox

coap://[aaaa::212:7402:2:202]:5683/sensors/button

[aaaa::212:7402:2:202]:5683

2.05 Content (Blockwise) (Download finishe

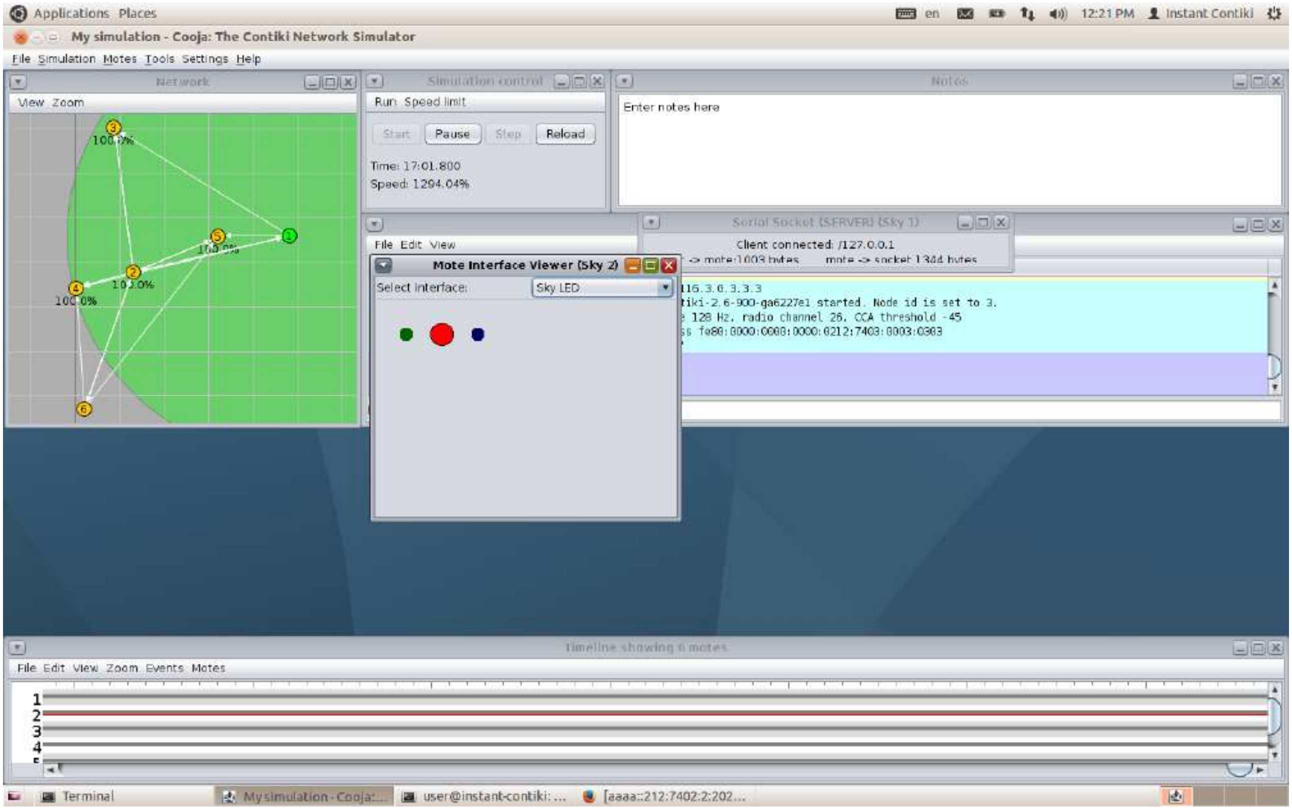
...	Value
Type	Acknowledgment
Co...	2.05 Content
M...	57822
To...	empty

Option	Value	...
Content-For...	text/plain	0
Block2	0 (64 B/...	1 b...

Payload (14)

1

Figure 31: **CoAP server**



My simulation - Cooja: The Contiki Network Simulator

Time: 17:01.800
Speed: 1294.04%

Enter notes here

Serial Socket (SERVER) (Sky 1)

Client connected: /127.0.0.1
-> mote:1003 bytes mote -> socket 1344 bytes

```
116.3.0.3.3.3
tiki-2.6-900-ga6227e1 started. Node id is set to 3.
 128 Hz. radio channel 26. CCA threshold -45
ss fe80:0000:0000:0000:0212:7403:0003:0303
```

Mote Interface Viewer (Sky 2)

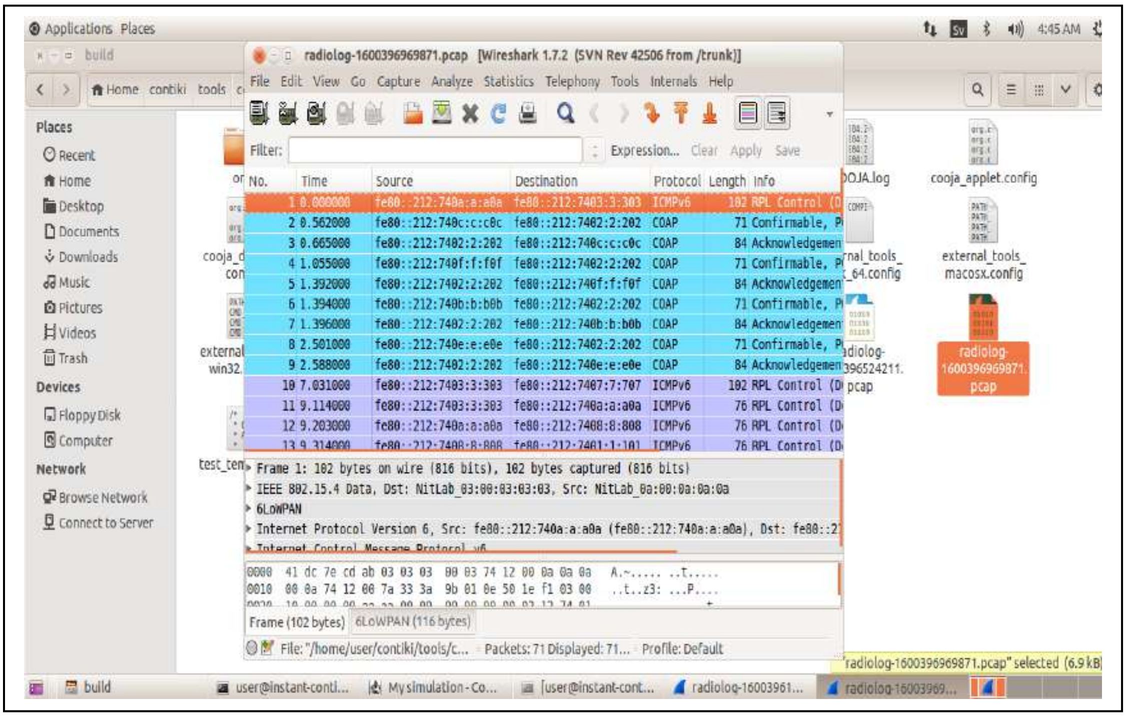
Select interface: Sky LED

Figure 32: **Cominucation CoAP Server**

The above two images show a firefox browser with a copper CU plugin to open the ipv6 address and read the sensor values. In the first picture, the toggle value 1 for Red LED is sent from the browser by selecting the POST button (OutGoing), Upon receiving the RED LED is glowing in the Mote that indicates that the node is accepting the inputs remotely.

- ✓ **Step 7**: network packet analyzer will try to capture network packets and tries to display that packet data details. It's a measuring device used to examine inside a network cable. we need to capture the radio message packets, In COOJA menu bar tools option is there and click enable radio messages: terminal and follow these steps.

Tools >> Radio messages New terminal will appear, Open menu and enable: Analyzer>> 6lowpan Analyzer with PCAP Figure(33).


Applications Places
4:45 AM
radiolog-1600396969871.pcap [Wireshark 1.7.2 (SVN Rev 42506 from /trunk)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
Places
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Devices
Floppy Disk
Computer
Network
Browse Network
Connect to Server
cooja_applet.config
external_tools_macosx.config
radiolog-1600396969871.pcap
"radiolog-1600396969871.pcap" selected (6.9 kB)
build
user@instant-conti...
My simulation - Co...
[user@instant-cont...
radiolog-16003961...

```
No.  Time      Source                 Destination            Protocol Length Info
  1 0.000000  fe80::212:740a:a:a0a   fe80::212:7403:3:303   ICMPv6     102 RPL Control (D
  2 0.562000  fe80::212:740c:c:c0c   fe80::212:7402:2:202   COAP        71 Confirmable, P
  3 0.665000  fe80::212:7402:2:202   fe80::212:740c:c:c0c   COAP        84 Acknowledgemen
  4 1.055000  fe80::212:740f:f:f0f   fe80::212:7402:2:202   COAP        71 Confirmable, P
  5 1.392000  fe80::212:7402:2:202   fe80::212:740f:f:f0f   COAP        84 Acknowledgemen
  6 1.394000  fe80::212:740b:b:b0b   fe80::212:7402:2:202   COAP        71 Confirmable, P
  7 1.396000  fe80::212:7402:2:202   fe80::212:740b:b:b0b   COAP        84 Acknowledgemen
  8 2.501000  fe80::212:740e:e:e0e   fe80::212:7402:2:202   COAP        71 Confirmable, P
  9 2.588000  fe80::212:7402:2:202   fe80::212:740e:e:e0e   COAP        84 Acknowledgemen
 10 7.031000  fe80::212:7403:3:303   fe80::212:7407:7:707   ICMPv6     102 RPL Control (D
 11 9.114000  fe80::212:7403:3:303   fe80::212:740a:a:a0a   ICMPv6      76 RPL Control (D
 12 9.203000  fe80::212:740a:a:a0a   fe80::212:7408:8:808   ICMPv6      76 RPL Control (D
 13 9.314000  fe80::212:7408:8:808   fe80::212:7401:1:101   ICMPv6      76 RPL Control (D

▸ Frame 1: 102 bytes on wire (816 bits), 102 bytes captured (816 bits)
▸ IEEE 802.15.4 Data, Dst: NitLab_03:00:03:03:03, Src: NitLab_0a:00:0a:0a:0a
▸ 6LoWPAN
▸ Internet Protocol Version 6, Src: fe80::212:740a:a:a0a (fe80::212:740a:a:a0a), Dst: fe80::2
▸ Internet Control Message Protocol v6

0000  41 dc 7e cd ab 03 03 03  00 03 74 12 00 0a 0a 0a   A.~..... ..t.....
0010  00 0a 74 12 00 7a 33 3a  9b 01 0e 50 1e f1 03 00   ..t..z3: ...P....

Frame (102 bytes)  6LoWPAN (116 bytes)
File: "/home/user/contiki/tools/c...  Packets: 71 Displayed: 71...  Profile: Default
```

Figure 33:   **Packet Analyses Using Wirechark**

8 PERFORMANCES EVALUATION EXPERIMENTS

In order to evaluate CoAP protocol performances, we use in our study three parameters (number of nodes, the interval of data transmission, packet size) where we study each parameter with power consumption, delay, packet delivery, and throughput. We did this by simulating the protocol and modifying the files provided in the system by adding some codes.

In each time we will change the number of nodes or the interval of data transaction, or the packet size, take results and transform them in curves in order to analyze the protocol.

8.1 Experiment 1: Number of servers

In this experiment we will study the throughput, delay, energy consumed and successfully delivered packets according to the number of servers. The first is simulated by 15, then 25, 35, 45 servers, Figures 34 illustrate the simulation according to the number of the server. With packets size 10 bit and data transmission interval 5 sec.

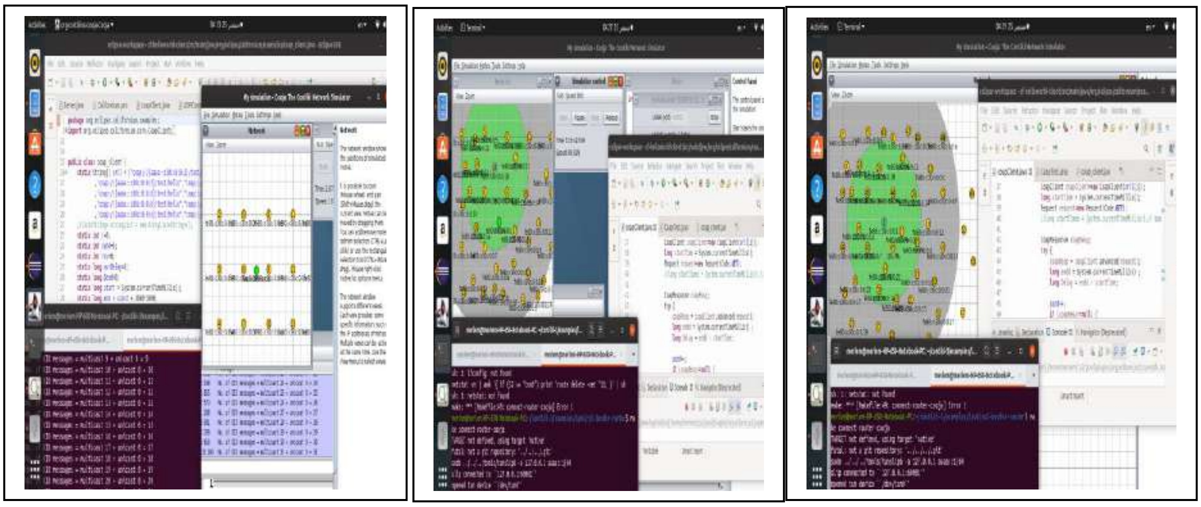
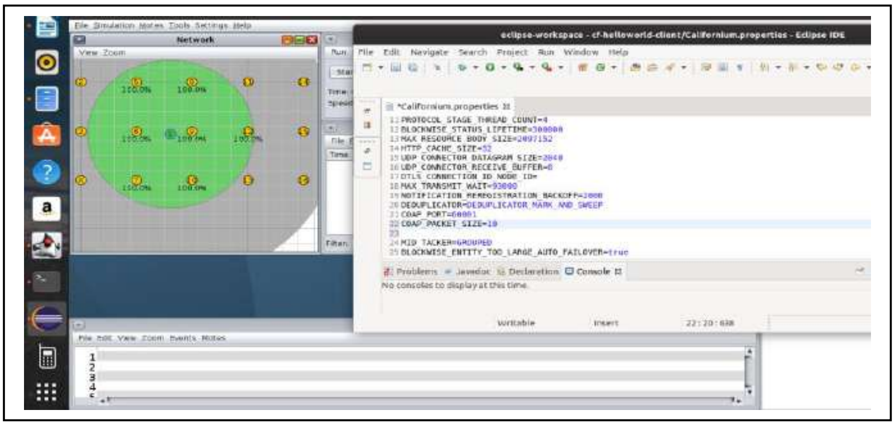


Figure 34: **Simulation For 15, Then 25, 35 Servers**

8.2 Experiment 2:Packets Size

In this experiment we will study beam output, delay, power consumed, and successfully delivered packets according to different packet sizes (10,30,50,70,90). With number of servers 15 servers and data transmission interval 5 sec .



eclipse-workspace - cf-helloworld-client/Californium.properties - Eclipse IDE
File Edit Navigate Search Project Run Window Help
*Californium.properties
Problems Javadoc Declaration Console
Writable
Insert
22 : 20 : 638
Network
View Zoom
100.0%

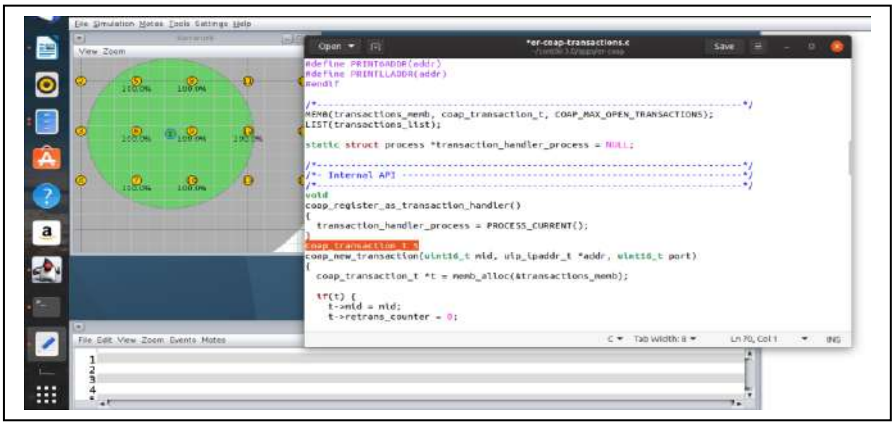
```
PROTOCOL_STAGE_THREAD_COUNT=4
BLOCKWISE_STATUS_LIFETIME=300000
MAX_RESOURCE_BODY_SIZE=2097152
HTTP_CACHE_SIZE=32
UDP_CONNECTOR_DATAGRAM_SIZE=2048
UDP_CONNECTOR_RECEIVE_BUFFER=0
DTLS_CONNECTION_ID_NODE_ID=
MAX_TRANSMIT_WAIT=93000
NOTIFICATION_REREGISTRATION_BACKOFF=2000
DEDUPLICATOR=DEDUPLICATOR_MARK_AND_SWEEP
COAP_PORT=60001
COAP_PACKET_SIZE=10

MID_TACKER=GROUPED
BLOCKWISE_ENTITY_TOO_LARGE_AUTO_FAILOVER=true
```

No consoles to display at this time.

Figure 35: **Simulation of 10 Packet Size**

8.3 Experiment 3 : Data Transmission Interval

In this experiment, we will measure what we mentioned earlier according to the field of data transmission interval. We will first take 5 seconds, then 15, 20 seconds, with the number of servers : 15 servers and packets size 10bit .

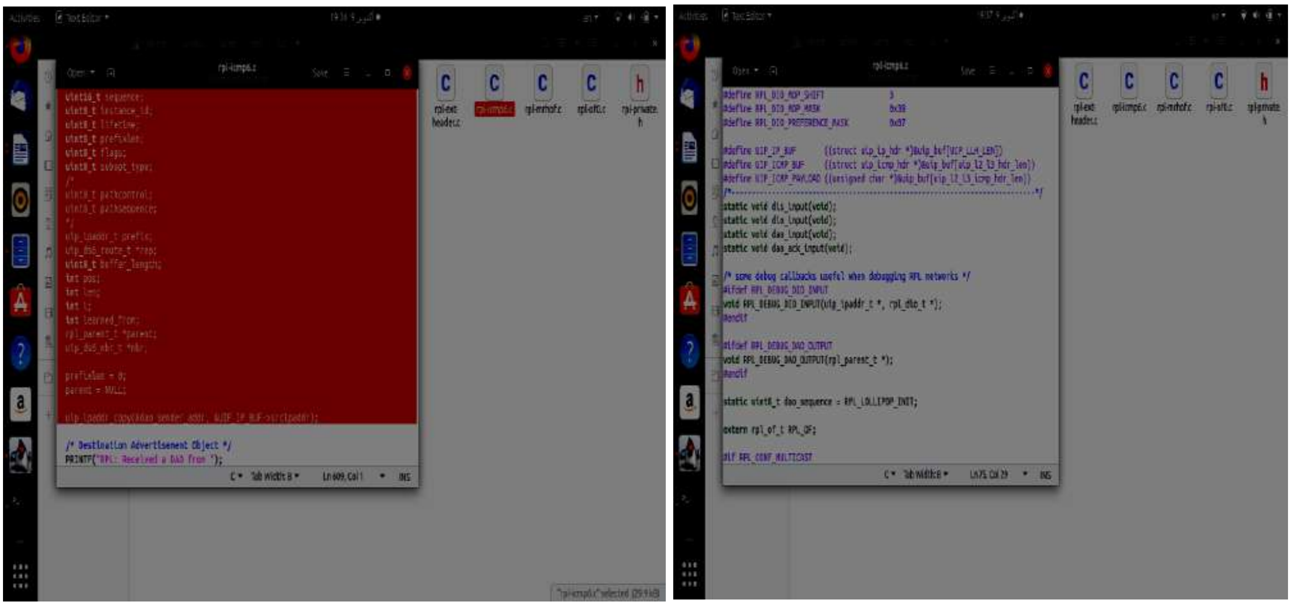


```
#define PRINT6ADDR(addr)
#define PRINTLLADDR(addr)
#endif

/*---------------------------------------------------------------------------*/
MEMB(transactions_memb, coap_transaction_t, COAP_MAX_OPEN_TRANSACTIONS);
LIST(transactions_list);

static struct process *transaction_handler_process = NULL;

/*---------------------------------------------------------------------------*/
/*- Internal API ------------------------------------------------------------*/
/*---------------------------------------------------------------------------*/
void
coap_register_as_transaction_handler()
{
  transaction_handler_process = PROCESS_CURRENT();
}
coap_transaction_t *
coap_new_transaction(uint16_t mid, uip_ipaddr_t *addr, uint16_t port)
{
  coap_transaction_t *t = memb_alloc(&transactions_memb);

  if(t) {
    t->mid = mid;
    t->retrans_counter = 0;
```

Figure 36: **Simulation Data Tansmission Interval of 5 seconds**

In order to get the required measurements (delay, throughput, power consumption, packets delivered successfully).here are some changes we have added to the files in Contiki OS ,It is not easy to change the files in this operating system, but after a long and deep search that took us effort and time we were able to introduce changes in the files by adding some instructions, as well as adding new files in the system that contain new methodes, And through these updates in the Contiki system, we were able to configure the simulator COOJA in order to complete our work seccesvely, And from some of these files we mention:

- ✓ The rpl_icmp6.h file is responsible for routing, which means it is responsible for message exchange between servers and their connection to the border router.

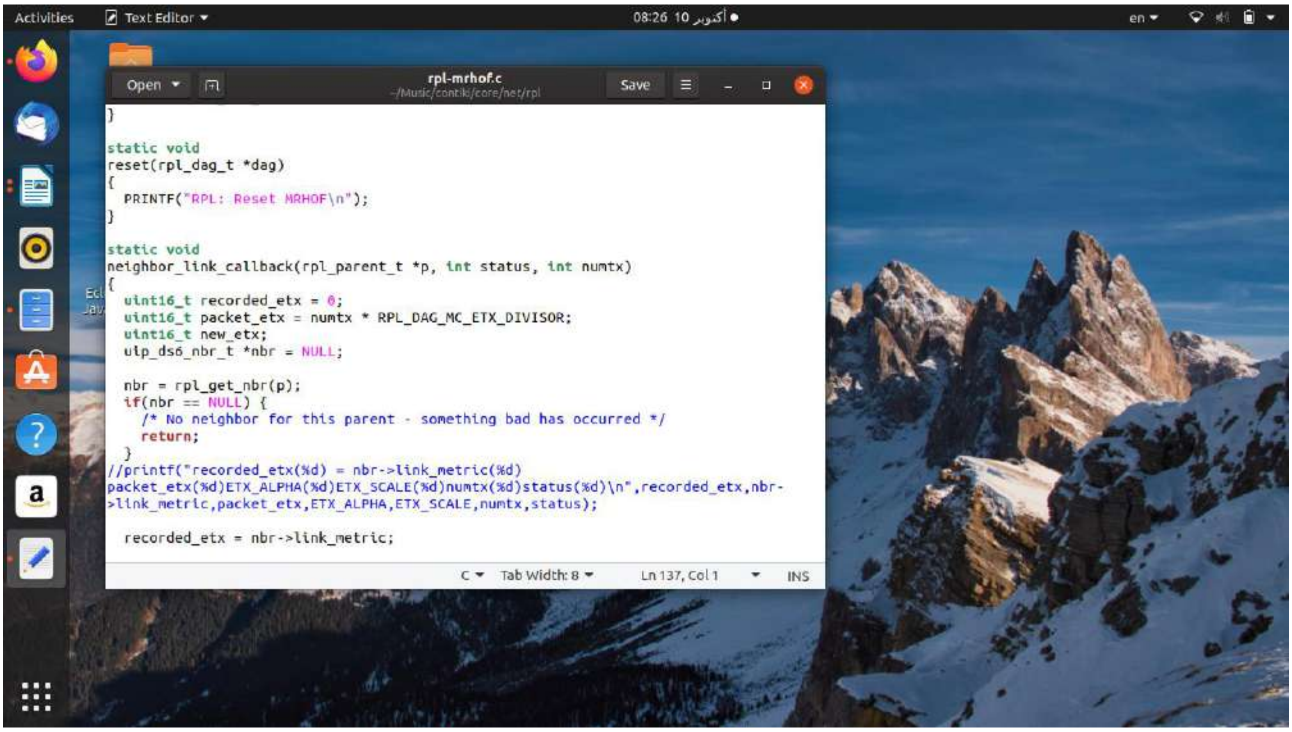


```
uint16_t sequence;
uint8_t instance_id;
uint8_t lifetime;
uint8_t prefixlen;
uint8_t flags;
uint8_t subopt_type;
/*
uint8_t pathcontrol;
uint8_t pathsequence;
*/
uip_ipaddr_t prefix;
uip_ds6_route_t *rep;
uint8_t buffer_length;
int pos;
int len;
int i;
int learned_from;
rpl_parent_t *parent;
uip_ds6_nbr_t *nbr;

prefixlen = 0;
parent = NULL;

uip_ipaddr_copy(&dao_sender_addr, &UIP_IP_BUF->srcipaddr);

/* Destination Advertisement Object */
PRINTF("RPL: Received a DAO from ");
```

```
#define RPL_DIO_MOP_SHIFT              3
#define RPL_DIO_MOP_MASK               0x38
#define RPL_DIO_PREFERENCE_MASK        0x07

#define UIP_IP_BUF       ((struct uip_ip_hdr *)&uip_buf[UIP_LLH_LEN])
#define UIP_ICMP_BUF     ((struct uip_icmp_hdr *)&uip_buf[uip_l2_l3_hdr_len])
#define UIP_ICMP_PAYLOAD ((unsigned char *)&uip_buf[uip_l2_l3_icmp_hdr_len])
/*---------------------------------------------------------------------------*/
static void dis_input(void);
static void dio_input(void);
static void dao_input(void);
static void dao_ack_input(void);

/* some debug callbacks useful when debugging RPL networks */
#ifdef RPL_DEBUG_DIO_INPUT
void RPL_DEBUG_DIO_INPUT(uip_ipaddr_t *, rpl_dio_t *);
#endif

#ifdef RPL_DEBUG_DAO_OUTPUT
void RPL_DEBUG_DAO_OUTPUT(rpl_parent_t *);
#endif

static uint8_t dao_sequence = RPL_LOLLIPOP_INIT;

extern rpl_of_t RPL_OF;

#if RPL_CONF_MULTICAST
```

Figure 37: **Add Modifications To The Rpl_Icmp6.h File**

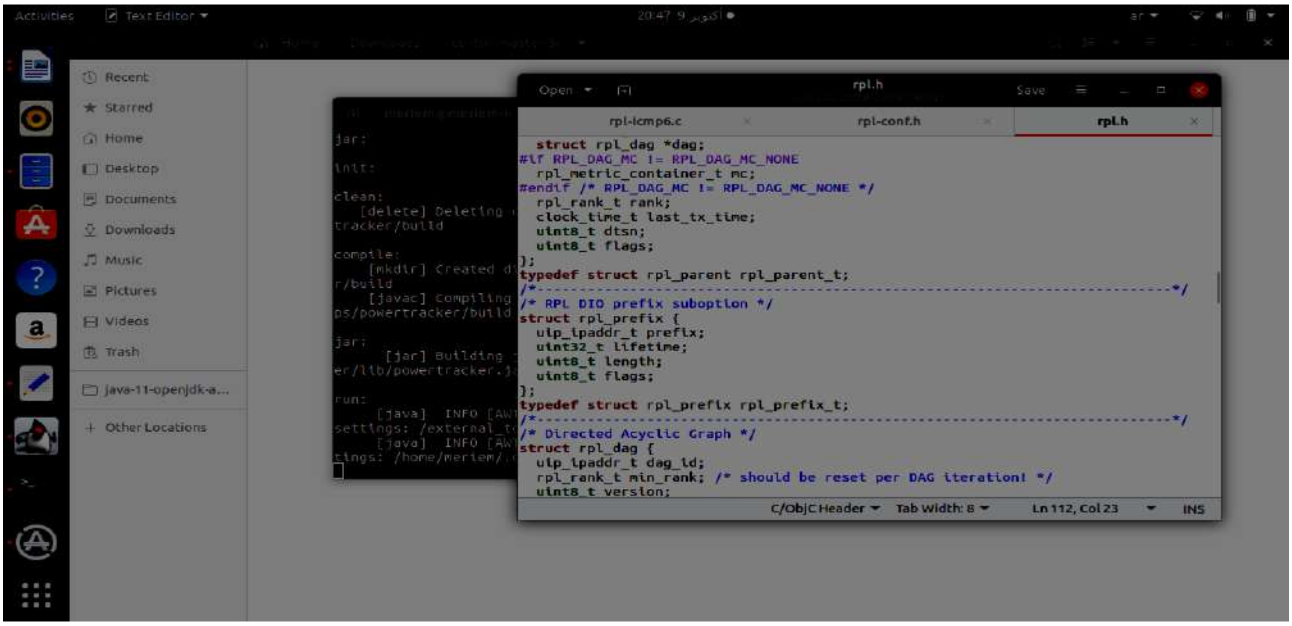
- ✓ Rpl_mrhof.c file This file is responsible for function object that the rpl protocol relies on how it communicates between servers there are two technique (hope-count and ETX) and in our simulation, we depend on the ETX technique.



```
}

static void
reset(rpl_dag_t *dag)
{
  PRINTF("RPL: Reset MRHOF\n");
}

static void
neighbor_link_callback(rpl_parent_t *p, int status, int numtx)
{
  uint16_t recorded_etx = 0;
  uint16_t packet_etx = numtx * RPL_DAG_MC_ETX_DIVISOR;
  uint16_t new_etx;
  uip_ds6_nbr_t *nbr = NULL;

  nbr = rpl_get_nbr(p);
  if(nbr == NULL) {
    /* No neighbor for this parent - something bad has occurred */
    return;
  }
//printf("recorded_etx(%d) = nbr->link_metric(%d)
packet_etx(%d)ETX_ALPHA(%d)ETX_SCALE(%d)numtx(%d)status(%d)\n",recorded_etx,nbr-
>link_metric,packet_etx,ETX_ALPHA,ETX_SCALE,numtx,status);

  recorded_etx = nbr->link_metric;
```

Figure 38: **Add Modifications To The Rpl_mrhof.c File**

- ✓ For the file rpl.h, through which we can calculate energy remaining for each nod.

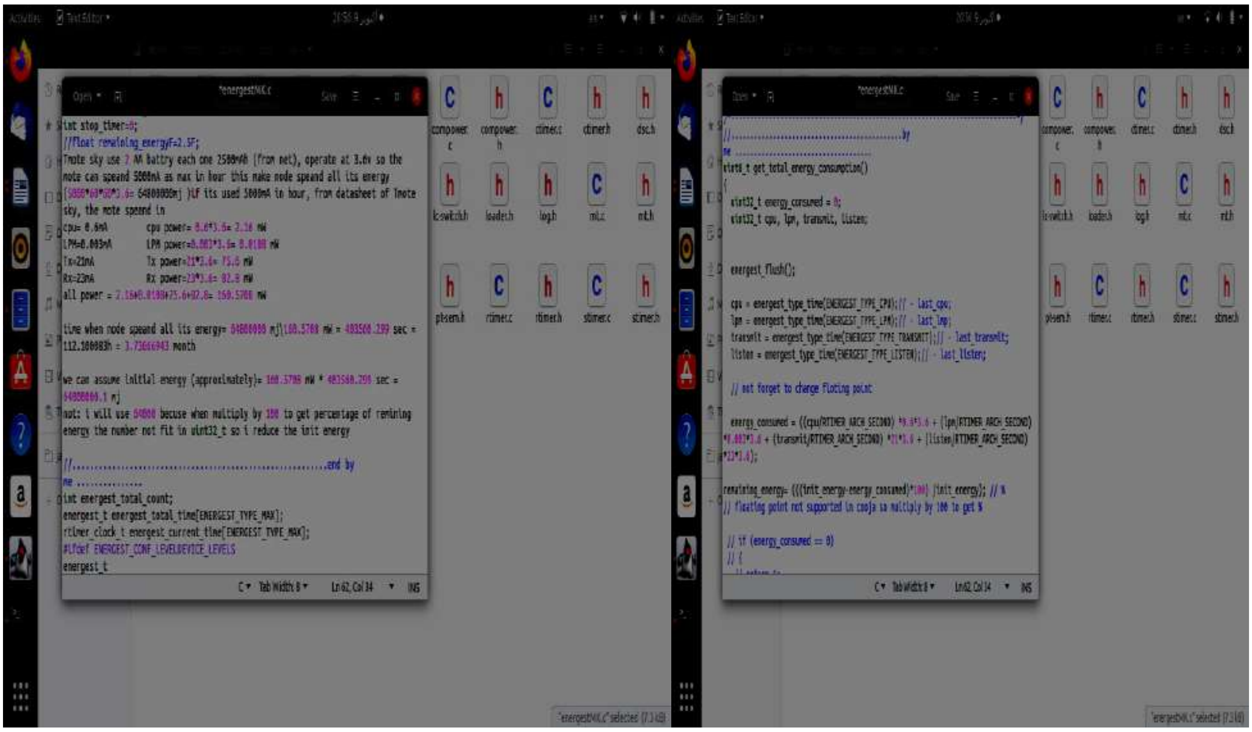


```
jar:

init:

clean:
   [delete] Deleting
tracker/build

compile:
    [mkdir] Created d
r/build
    [javac] Compiling
ps/powertracker/build

jar:
      [jar] Building
er/lib/powertracker.j

run:
     [java]  INFO [AW
settings: /external_t
     [java]  INFO [AW
tings: /home/meriem/.
```

```
  struct rpl_dag *dag;
#if RPL_DAG_MC != RPL_DAG_MC_NONE
  rpl_metric_container_t mc;
#endif /* RPL_DAG_MC != RPL_DAG_MC_NONE */
  rpl_rank_t rank;
  clock_time_t last_tx_time;
  uint8_t dtsn;
  uint8_t flags;
};
typedef struct rpl_parent rpl_parent_t;
/*---------------------------------------------------------------------------*/
/* RPL DIO prefix suboption */
struct rpl_prefix {
  uip_ipaddr_t prefix;
  uint32_t lifetime;
  uint8_t length;
  uint8_t flags;
};
typedef struct rpl_prefix rpl_prefix_t;
/*---------------------------------------------------------------------------*/
/* Directed Acyclic Graph */
struct rpl_dag {
  uip_ipaddr_t dag_id;
  rpl_rank_t min_rank; /* should be reset per DAG iteration! */
  uint8_t version;
```

Figure 39: **Add Modifications To Rpl.Conf.h File**

- ✓ file EnergestMK.c this file that we have added in order to calculate the power consumption of servers by CPU, sleep nodes, transmission nodes, and listening nodes.

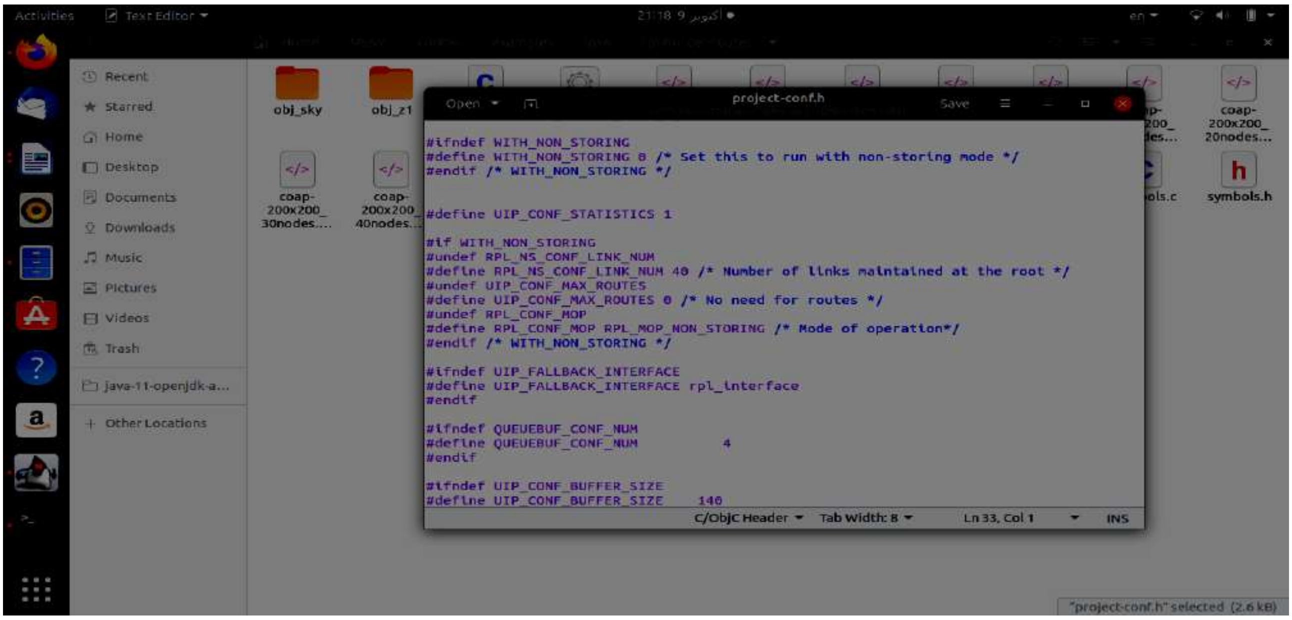


```
int stop_timer=0;
//float remaining_energyF=2.5F;
Tmote sky use 2 AA battry each one 2500mAh (from net), operate at 3.6v so the
mote can speand 5000mA as max in hour this make node speand all its energy
(5000*60*60*3.6= 64800000mj )if its used 5000mA in hour, from datasheet of Tmote
sky, the mote speand in
cpu= 0.6mA        cpu power= 0.6*3.6= 2.16 mW
LPM=0.003mA       LPM power=0.003*3.6= 0.0108 mW
Tx=21mA           Tx power=21*3.6= 75.6 mW
Rx=23mA           RX power=23*3.6= 82.8 mW
all power = 2.16+0.0108+75.6+82.8= 160.5708 mW

time when node speand all its energy= 64800000 mj\160.5708 mW = 403560.299 sec =
112.100083h = 3.73666943 month

we can assume initial energy (approximately)= 160.5708 mW * 403560.299 sec =
64800000.1 mj
not: i will use 64000 becuse when multiply by 100 to get percentage of remining
energy the number not fit in uint32_t so i reduce the init energy

//.........................................................end by
me ...............
int energest_total_count;
energest_t energest_total_time[ENERGEST_TYPE_MAX];
rtimer_clock_t energest_current_time[ENERGEST_TYPE_MAX];
#ifdef ENERGEST_CONF_LEVELDEVICE_LEVELS
energest_t
```

```
//.............................................by
me ...........................................
uint8_t get_total_energy_consumption()
{
  uint32_t energy_consumed = 0;
  uint32_t cpu, lpm, transmit, listen;


  energest_flush();

  cpu = energest_type_time(ENERGEST_TYPE_CPU);// - last_cpu;
  lpm = energest_type_time(ENERGEST_TYPE_LPM);// - last_lpm;
  transmit = energest_type_time(ENERGEST_TYPE_TRANSMIT);// - last_transmit;
  listen = energest_type_time(ENERGEST_TYPE_LISTEN);// - last_listen;

  // not forget to change floting point

  energy_consumed = ((cpu/RTIMER_ARCH_SECOND) *0.6*3.6 + (lpm/RTIMER_ARCH_SECOND)
*0.003*3.6 + (transmit/RTIMER_ARCH_SECOND) *21*3.6 + (listen/RTIMER_ARCH_SECOND)
*23*3.6);

remaining_energy= (((init_energy-energy_consumed)*100) /init_energy); // %
// floating point not supported in cooja so multiply by 100 to get %

 // if (energy_consumed == 0)
 // {
```

Figure 40: **Add Modifications To Energestmk.C File**

- ✓ the er-example-server .c file that we use to create the server node in the COOJA emulator.
- ✓ The rpl-border-router.c file that we also use to create a router node in the COOJA emulator.
- ✓ project-conf.h file in rpl-border-router folder We use this file to view the stats for the border router.



```
#ifndef WITH_NON_STORING
#define WITH_NON_STORING 0 /* Set this to run with non-storing mode */
#endif /* WITH_NON_STORING */


#define UIP_CONF_STATISTICS 1

#if WITH_NON_STORING
#undef RPL_NS_CONF_LINK_NUM
#define RPL_NS_CONF_LINK_NUM 40 /* Number of links maintained at the root */
#undef UIP_CONF_MAX_ROUTES
#define UIP_CONF_MAX_ROUTES 0 /* No need for routes */
#undef RPL_CONF_MOP
#define RPL_CONF_MOP RPL_MOP_NON_STORING /* Mode of operation*/
#endif /* WITH_NON_STORING */

#ifndef UIP_FALLBACK_INTERFACE
#define UIP_FALLBACK_INTERFACE rpl_interface
#endif

#ifndef QUEUEBUF_CONF_NUM
#define QUEUEBUF_CONF_NUM          4
#endif

#ifndef UIP_CONF_BUFFER_SIZE
#define UIP_CONF_BUFFER_SIZE    140
```

Figure 41:   **Add Modifications To Project-Conf.h File**

9 CONCLUSION

For the purpose of the simulation of CoAP protocol, we have given in this chapter a brief presentation of the steps of simulation, we have first begun by define the validation methods, and some fields areas in which they are used, then we passed to present the programs used, the simulation environment. We finish by giving providing different simulations by increasing time in eatch one (number of nodes, data transmission interval, packets size).

In the next chapter, we will present an evaluation of the protocol performance through this simulation.

Chapter4:

PERFORMANCES EVALUATION

1 INTRODUCTION

The evaluation of the performance of a system via a simulation consists of the choice of a model, the evaluation by a simulation technique, and the interpretation of the measurements collected. A large number of simulation models have been developed for the study of architectures and protocols under various network scenarios (number of nodes, mobility, etc.). They have been widely used for the evaluation of routing protocols.[45].

In this chapter, we will discuss the results obtained through simulations in the previous part, and through them, we conclud the effectiveness of the CoAP protocol with regard to the Internet of things.

2 CRETERIAS OF EVALUATION

To evaluate the performance of the protocol, we study some Measures, including [2],[6],[9]:

2.1 Network Delay

This performance metric is used to measure the average end-to-end delay of data packet transmission. The end-to-end delay implies the average time taken between a packet initially sent by the source, and the time for successfully receiving the message at the destination. Measuring this delay takes into account the queuing and the propagation delay of the packets. It is the sum of

2 * max latency and the processing delay

2.2 Network Throughput

The end-to-end network throughput measures the number of packets per second delivered at the destination. It is considered here as an external measure of the effectiveness of a protocol.is calculated as

(in Kbps) = (No of successful CoAP request/response pairs * (length of request + length of response in bits)) / total time of simulation.

2.3 Packet Delivered Successfully

The total number of packets delivered at the destinations versus the total number of packets sent from the source.

2.4 Latency

The average message latency is defined as the average amount of time between the start of distributing data and its arrival at a node interested in receiving the data. Hence the latency measures time performance for the individual message.

2.5 Energy Consumption

The energy consumption is the sum of used energy of all the nodes in the network, where the used energy of a node is the sum of the energy used for communication, including transmitting (pt), receiving (pr), and idling (pi). assuming each transmission consumes an energy unit, the total energy consumption is equivalent to the total number of packets sent in the network.Power Consumption= (Transmit/19.5 mA + Listen /21.5 mA +CPU power/1.8 mA +LPM/0.0545 mA)/3v/(32768).

2.6 Network Lifetime

It is considered as the time until the message loss rate is above a given threshold. the more complete definition for the lifetime of the network is “time to network partition” network partition occurs when there is a cut-set in the network. it will be introduced as a new metric, which will use energy variance:

$$\text{network lifetime} = e - (u + \sigma), \quad \text{where } u = \sigma ui/n$$

e is the total initial energy at each node (full battery charge),

ui is the average used energy,

n is the total number of nodes in the network,

σ is expressed as

$$\sigma 2 = \frac{(ui - u)2}{n}$$

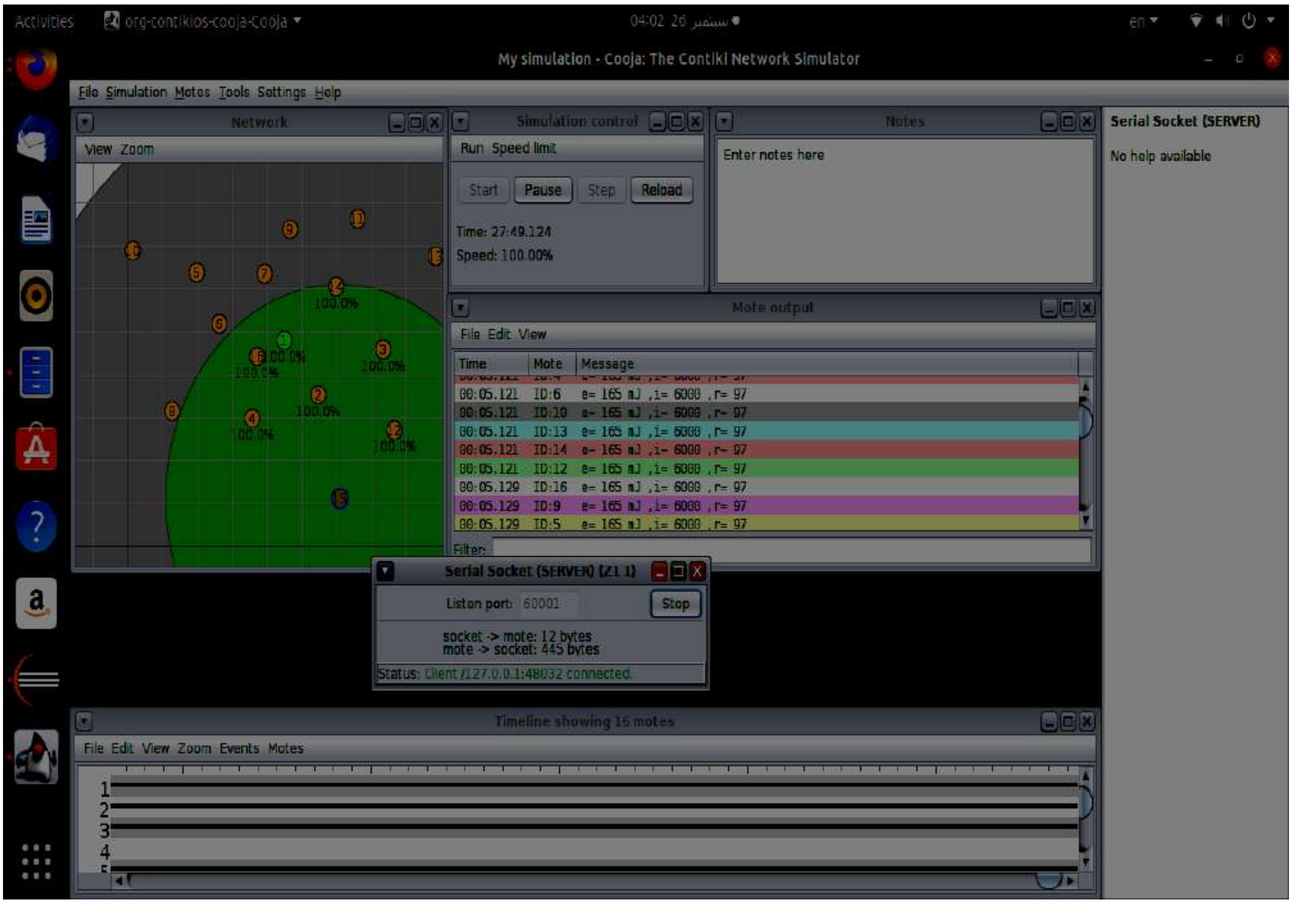
All these metrics are calculated using their cumulative average values, that is, at time t, the performance value is the average from 0 to t (seconds).

2.7 Packet Generation Rate

It is the number of packets that the sensor node transmits in one time period which is usually one second.

3 PERFORMANCE EVALUATION METHODE

With the modifications we added in the Contiki OS, each server is ready to give the values we use to get the final results and appear through a "Mote output " window in COOJA.



Activities
org-contikios-cooja-Cooja
04:02 26 سبتمبر
en
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Network
View Zoom
Simulation control
Run Speed limit
Start
Pause
Step
Reload
Notes
Serial Socket (SERVER)
No help available
Mote output
File Edit View
Filter:
Serial Socket (SERVER) (Z1 1)
Stop
Timeline showing 16 motes
File Edit View Zoom Events Motes

Time: 27:49.124
Speed: 100.00%

Enter notes here

```
Time        Mote    Message
00:05.121   ID:6    e= 165 mJ ,i= 6000 ,r= 97
00:05.121   ID:10   e= 165 mJ ,i= 6000 ,r= 97
00:05.121   ID:13   e= 165 mJ ,i= 6000 ,r= 97
00:05.121   ID:14   e= 165 mJ ,i= 6000 ,r= 97
00:05.121   ID:12   e= 165 mJ ,i= 6000 ,r= 97
00:05.129   ID:16   e= 165 mJ ,i= 6000 ,r= 97
00:05.129   ID:9    e= 165 mJ ,i= 6000 ,r= 97
00:05.129   ID:5    e= 165 mJ ,i= 6000 ,r= 97
```

Listen port: 60001

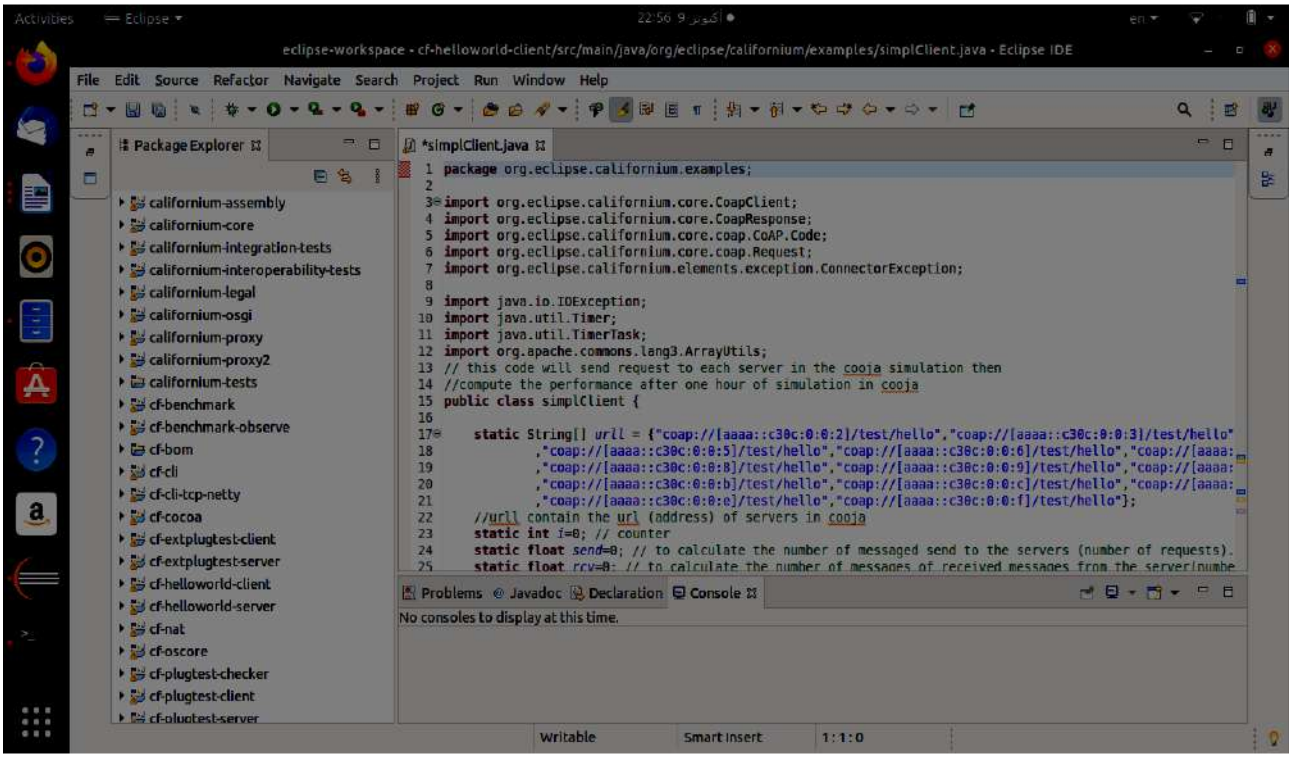
```
socket -> mote: 12 bytes
mote -> socket: 445 bytes
Status: Client /127.0.0.1:48032 connected.
```

Figure 42: **The final results are in Mote output**

On the client side, we have created a new class in the editor Eclipse, we called it “simplClient” . This client will contact the servers in COOJA every time through a local connection between Border Router and Californium Eclipse platform and after an hour of time the simulation ends and in the client file we get the average delay ,ratio of packets Successfully received ,throughput and energy consumption.

And every time we change the parameters to obtain different results in order to compare them. each time, we change the number of servers or increase the packates size Or increase in the interval of data conversion.

The figures below show the client file with comments on each instruction:

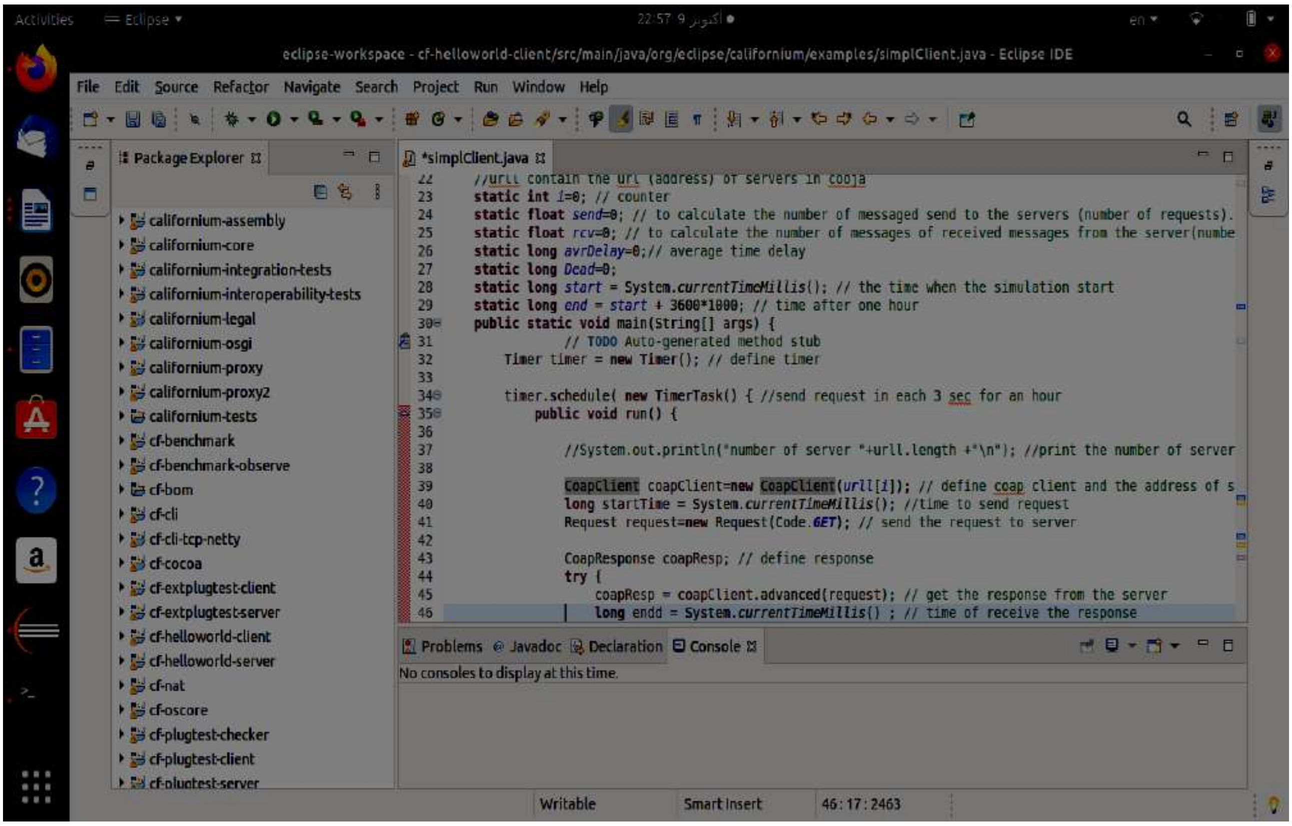


eclipse-workspace - cf-helloworld-client/src/main/java/org/eclipse/californium/examples/simplClient.java - Eclipse IDE

```java
package org.eclipse.californium.examples;

import org.eclipse.californium.core.CoapClient;
import org.eclipse.californium.core.CoapResponse;
import org.eclipse.californium.core.coap.CoAP.Code;
import org.eclipse.californium.core.coap.Request;
import org.eclipse.californium.elements.exception.ConnectorException;

import java.io.IOException;
import java.util.Timer;
import java.util.TimerTask;
import org.apache.commons.lang3.ArrayUtils;
// this code will send request to each server in the cooja simulation then
//compute the performance after one hour of simulation in cooja
public class simplClient {

    static String[] url1 = {"coap://[aaaa::c30c:0:0:2]/test/hello","coap://[aaaa::c30c:0:0:3]/test/hello"
            ,"coap://[aaaa::c30c:0:0:5]/test/hello","coap://[aaaa::c30c:0:0:6]/test/hello","coap://[aaaa:
            ,"coap://[aaaa::c30c:0:0:8]/test/hello","coap://[aaaa::c30c:0:0:9]/test/hello","coap://[aaaa:
            ,"coap://[aaaa::c30c:0:0:b]/test/hello","coap://[aaaa::c30c:0:0:c]/test/hello","coap://[aaaa:
            ,"coap://[aaaa::c30c:0:0:e]/test/hello","coap://[aaaa::c30c:0:0:f]/test/hello"};
    //url1 contain the url (address) of servers in cooja
    static int i=0; // counter
    static float send=0; // to calculate the number of messaged send to the servers (number of requests).
    static float rcv=0; // to calculate the number of messages of received messages from the server(numbe
```

No consoles to display at this time.

Figure 43: **The first part of the code**

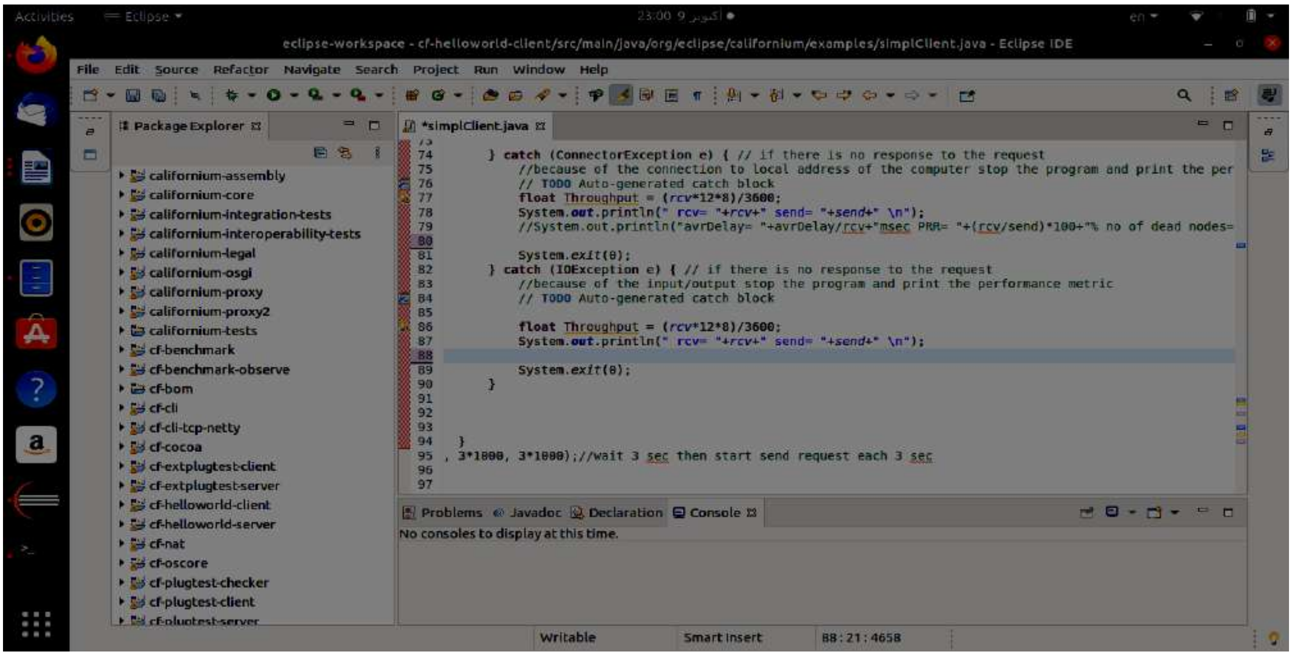


eclipse-workspace - cf-helloworld-client/src/main/java/org/eclipse/californium/examples/simplClient.java - Eclipse IDE

```java
    //url1 contain the url (address) of servers in cooja
    static int i=0; // counter
    static float send=0; // to calculate the number of messaged send to the servers (number of requests).
    static float rcv=0; // to calculate the number of messages of received messages from the server(numbe
    static long avrDelay=0;// average time delay
    static long Dead=0;
    static long start = System.currentTimeMillis(); // the time when the simulation start
    static long end = start + 3600*1000; // time after one hour
    public static void main(String[] args) {
            // TODO Auto-generated method stub
        Timer timer = new Timer(); // define timer

        timer.schedule( new TimerTask() { //send request in each 3 sec for an hour
            public void run() {

                //System.out.println("number of server "+url1.length +"\n"); //print the number of server

                CoapClient coapClient=new CoapClient(url1[i]); // define coap client and the address of s
                long startTime = System.currentTimeMillis(); //time to send request
                Request request=new Request(Code.GET); // send the request to server

                CoapResponse coapResp; // define response
                try {
                    coapResp = coapClient.advanced(request); // get the response from the server
                    long endd = System.currentTimeMillis() ; // time of receive the response
```

No consoles to display at this time.

Figure 44: **The second part of the code**



```
        } catch (ConnectorException e) { // if there is no response to the request
            //because of the connection to local address of the computer stop the program and print the per
            // TODO Auto-generated catch block
            float Throughput = (rcv*12*8)/3600;
            System.out.println(" rcv= "+rcv+" send= "+send+" \n");
            //System.out.println("avrDelay= "+avrDelay/rcv+"msec PRR= "+(rcv/send)*100+"% no of dead nodes=

            System.exit(0);
        } catch (IOException e) { // if there is no response to the request
            //because of the input/output stop the program and print the performance metric
            // TODO Auto-generated catch block

            float Throughput = (rcv*12*8)/3600;
            System.out.println(" rcv= "+rcv+" send= "+send+" \n");

            System.exit(0);
        }




    }
, 3*1000, 3*1000);//wait 3 sec then start send request each 3 sec
```

```
No consoles to display at this time.
```

Figure 45: **The last part of the code**

To evaluate the protocol's performance, we perform the following experiments:

3.1 Results Simulation Experiment1:Number Of Server

The following table presents the results extracted from the simulation of the first experiment which is based on the parameter: the number of servers. We calculated the measures of the characteristics (packet delivered, delay, throughput, energy consumption) in each time (15,25,35,45 server), By keeping the same simulation surface, and Transmission interval which is 5sec, and the packet size which is 10 bit :

Number of servers	Packetdelivery	delay	throughput	Energyconsumption
15	99.3%	1145.6 msec	35.7 bps	5537.4 mj
25	97.5%	2155.3 msec	31.1 bps	5723.6 mj
35	96.1%	3546.1 msec	27.7 bps	5842.2 mj
45	95.2%	4166.2msec	23.8 bps	5973.7 mj

Table 8: Criteria Values Depend Servers Number

From the results of the above table, the following Graphic shapes were constructed:

- ✓ Figure 46 shows the percentage of successfully delivered packets in relation to the number of servers

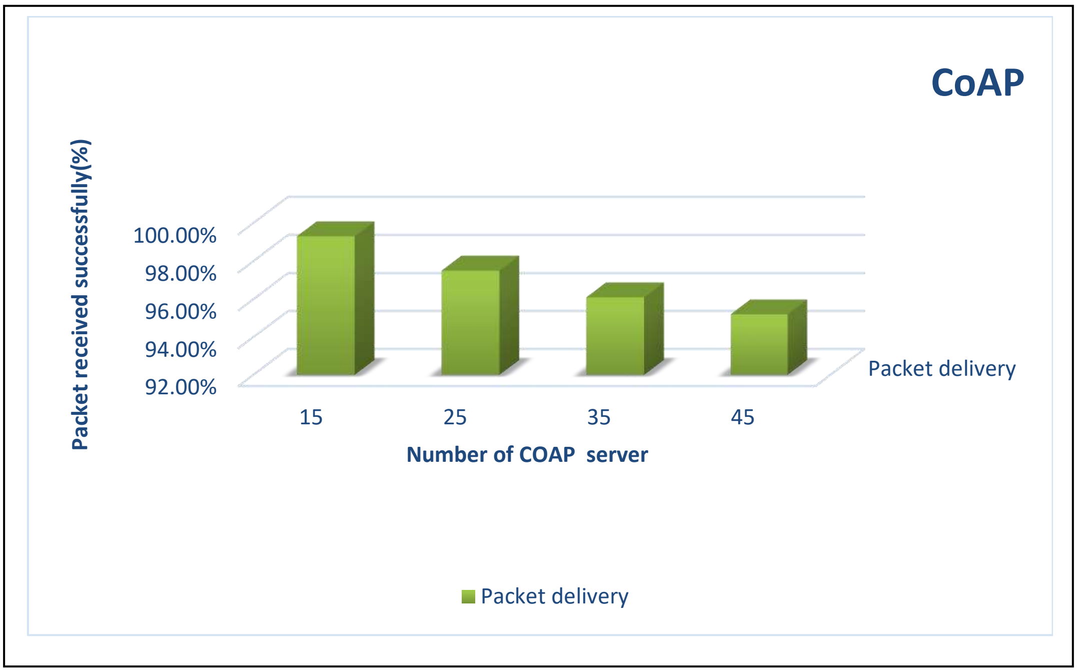


Figure 46: **Curves of Packet delivered successfully against the Number of COAP server**

We notice in figure (46) when 15 servers took the percentage of successfully delivered packets 99% acceptable, then when increasing the number of servers respectively 25, 35, 45 we notice a gradual decrease in the percentage of successfully delivered packets.

- Figure 47 shows the period of delay in relation to the number of servers

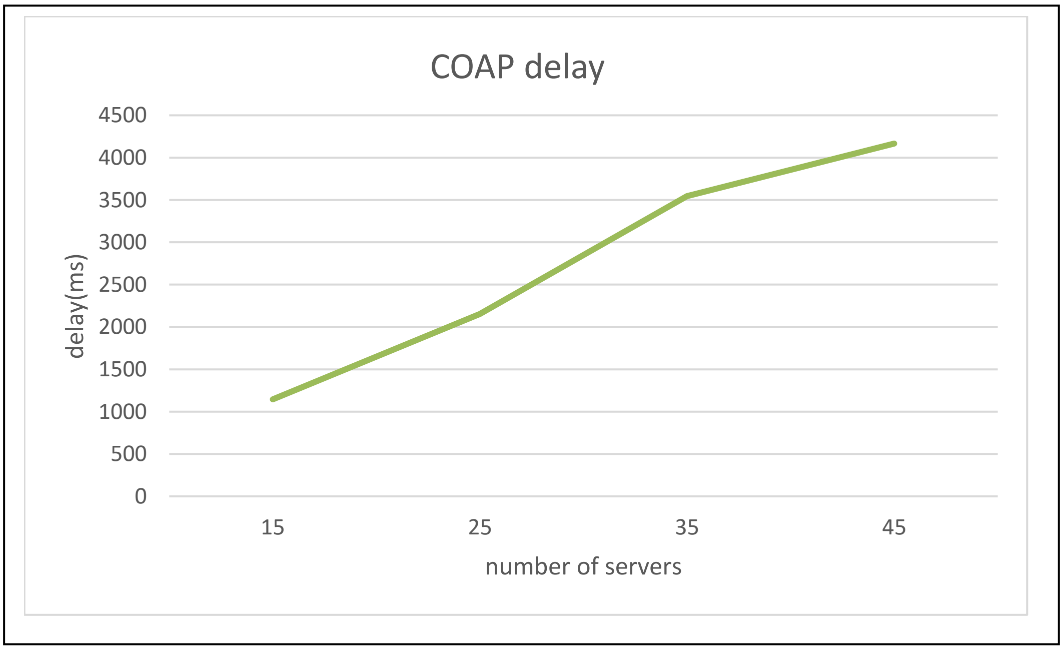


Figure 47: **Curves of delay(ms) against the Number of COAP server**

Note in Figure (47) when 15 servers take the delay time value, it does not affect the delivered packets, but when the number of servers increases, the delay time value increases respectively from 1000MS to 4000MS.

- ✓ Figure 48 shows the throughput values in relation to the number of servers

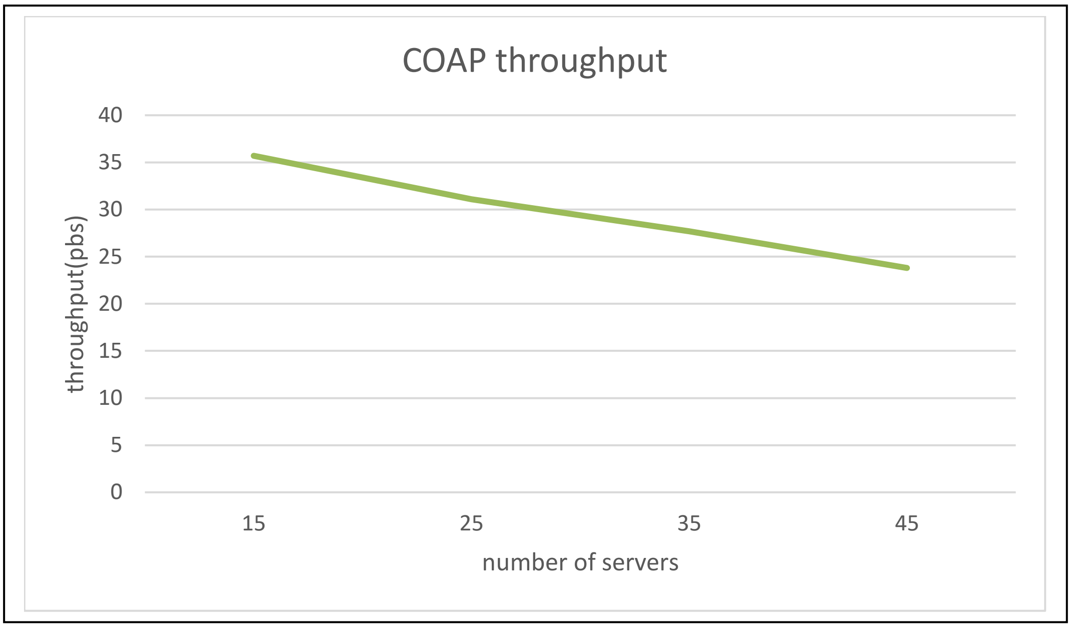


Figure 48: **Curves of throughput(pbs)against theNumber of COAP server**

We notice in figure 48 an acceptable throughput value when taking 15 servers, then when the number of servers gradually increases, the throughput decreases, respectively, from 35bps to 24 bps.

- ✓ Shows figure 49 the energy consumption values in relation to the number of servers

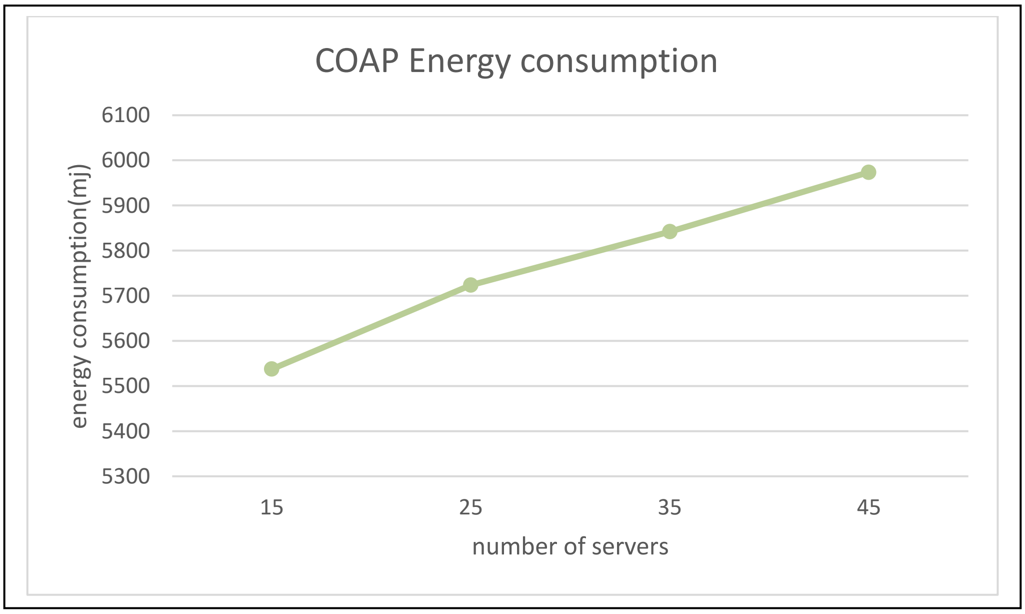


Figure 49: **Curves of energy consumption(mj)against theNumber of COAP server**

In figure (49), we notice the value of energy consumption when consuming 15 servers a few, then it increases with the number of servers gradually increasing until it reaches 6000 MJ.

Analyze

When studying packets delivered, delay, throughput, as well as energy consumption, we note with a gradual increase in the number of servers:

- a number of servers increases, the radio traffic increases due to the difficulty of accessing the Border Router explained by the presence of packets loss.
- the number of servers increases, the distance from the Border Router increases which explain the delay.
- Since throughput depends on the percentage of packets delivered successfully by the client increasing the number of servers causes a decrease in throughput.

3.2 Results Simulation Experiment 2 : Data Transmission Interval

The following table presents the results extracted from the simulation of the second experiment which is based on the parameter: the data transmission interval. We calculated the measures of the characteristics (packet delivered, delay, throughput, energy consumption) in each time (5sec,15sec,20sec), By keeping the same simulation surface, and a number of servers are 15 servers, and the packet size which is 10 bit:

Data transmission interval	Packetdelivery	delay	throughput	Energyconsumption
5 sec	99.3%	1145.6 msec	35.7 bps	5537.4 mj
15 sec	99.3%	1112.7 msec	35.2 bps	5686.8 mj
20 sec	99.2%	1134.1 msec	35.5 bps	5755.2mj

Table 9: Criteria Values Depend Data Transmission Interval

From the results of the above tables, the following Graphic shapes were constructed:

✓ figure 50 shows the percentage of successfully delivered packets in relation to the data transmission interval

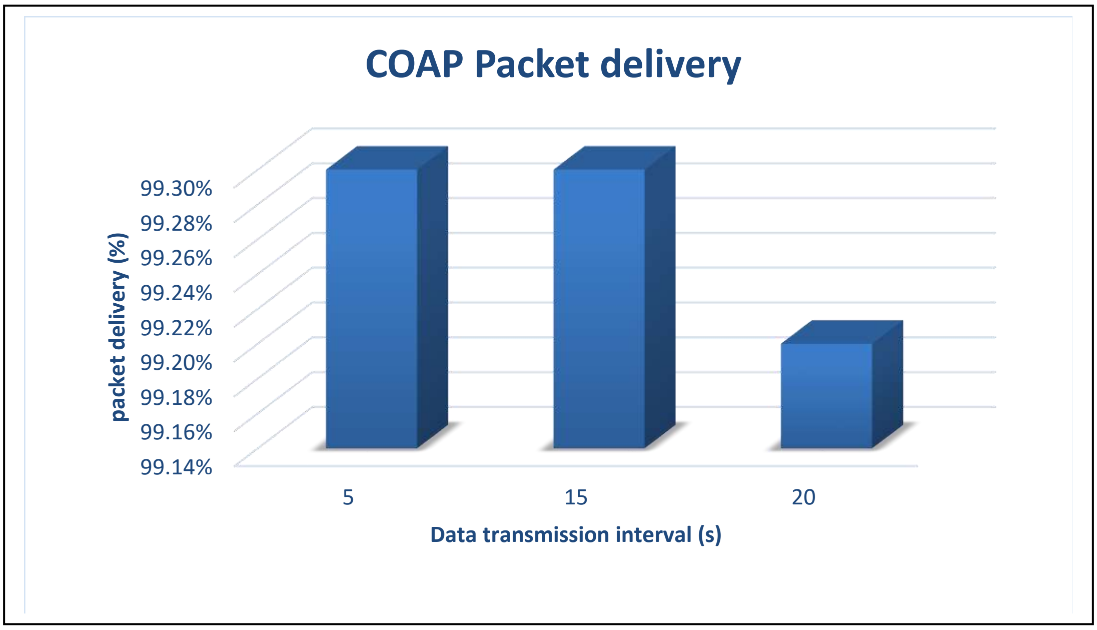


Figure 50: **Curves of packet deliveryagainst theData transmission interval (s)**

We can see in Figure 50 when the arrival interval is 5 and 15 S, the percentage of successfully delivered packets is 99%, then the packet percentage decreases by a large percentage.

✓ figure 50 shows the delay values in relation to the data transmission interval

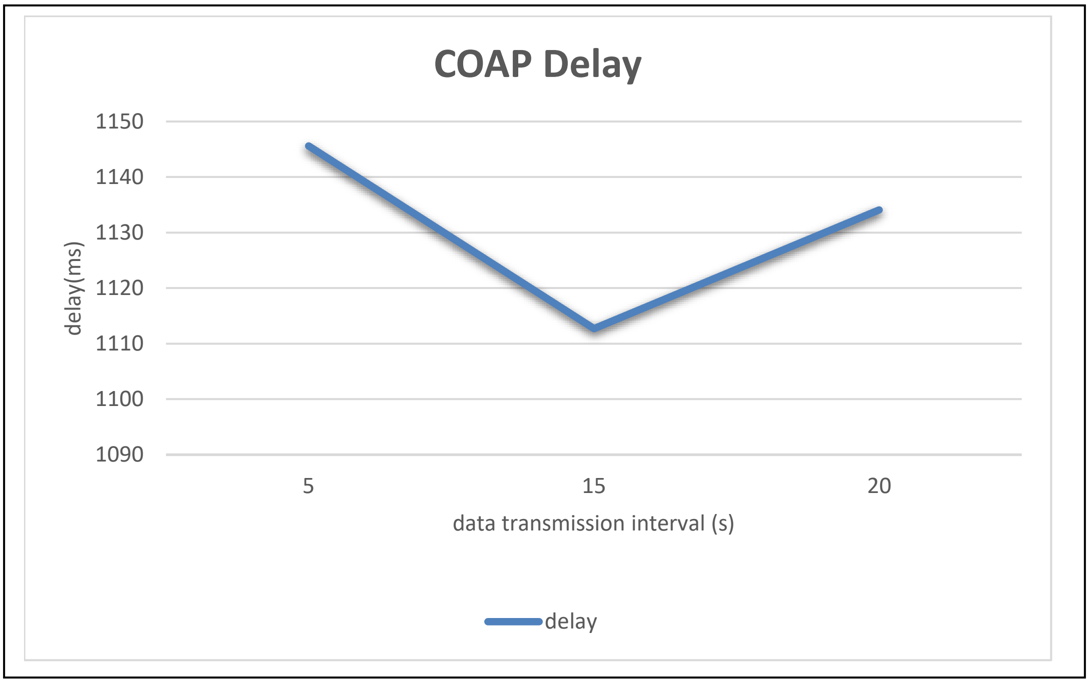


Figure 51: **Curves of delay(ms) against the Data transmission interval (s)**

We notice in Fig. 51 that when the arrival interval exceeds 15 seconds, the delay time increases slightly gradually until it reaches 1135 milliseconds.

- figure 52 shows the throughput values in relation to the data transmission interval

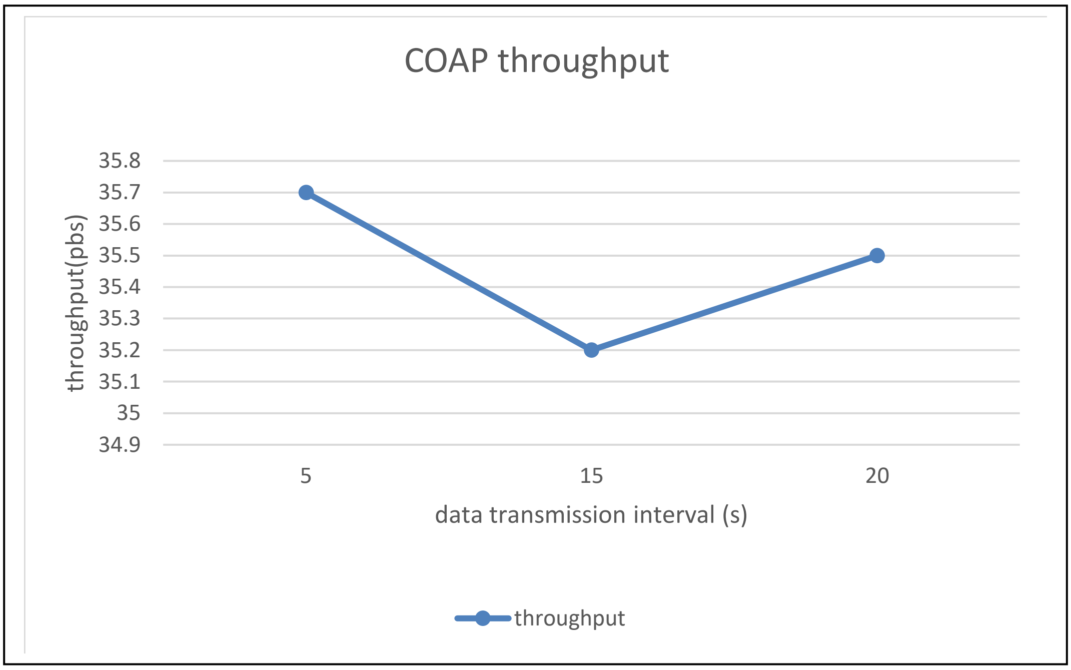


Figure 52: **Curves of throughput(pbs) against the Data transmission interval (s)**

We notice in figure 52 that when the arrival interval exceeds 15s, the throughput period gradually decreases.

✓ The figure 53 shows the energy consumption values in relation to the data transmission interval .

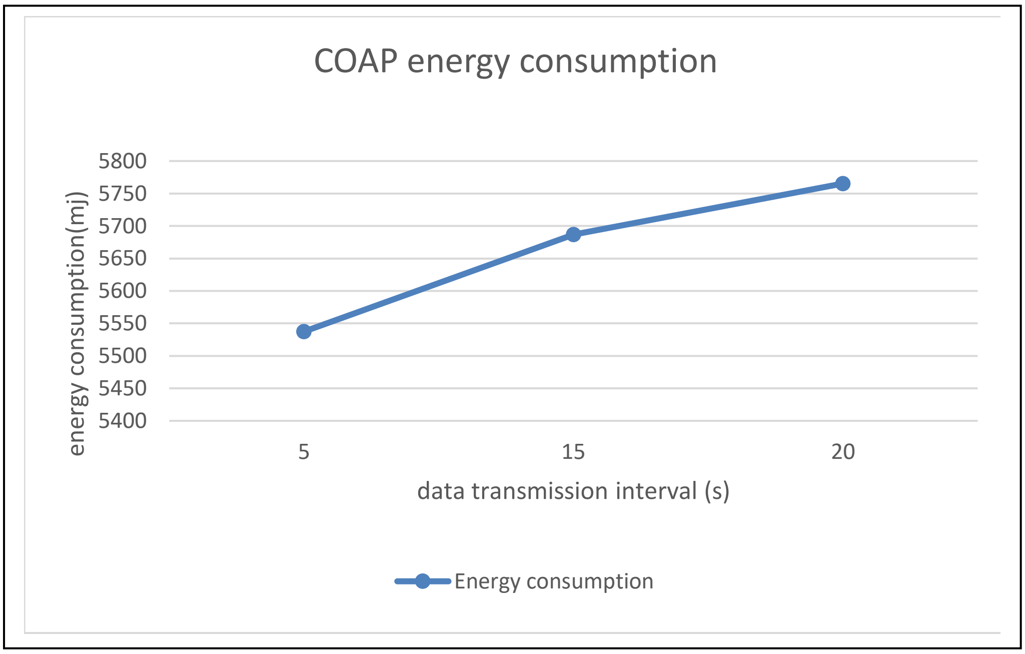


Figure 53: **Curves of energy consumption(mj) against the Data transmission interval (s)**

In Figure 53, we note the value of power consumption when the arrival interval exceeds 15s gradually increasing.

Analyze

When studying packets delivered, delay, throughput, as well as energy consumption, we note with a gradual increase in data transmission interval :

- transmission data interval is increased, there is an exchange of radio traffic, which leads to frequent communication between nodes and the increased packet loss
- radio traffic increases ample in which explains energy consumption.

3.3 Results Simulation Experiment 3 : Packets Size

The following table presents the results extracted from the simulation of the third experiment in which we relied on packets size parameter and calculated the characteristics (packet delivered, delay, throughput, energy consumption) each time, By keeping the same simulation surface, and a number of servers 15, data transmission interval 5 sec:

Packet size	Packetdelivered	Delay	Throughput	Energyconsumption
10	99.3%	1145.6 msec	35.7 bps	5537.4 mj
30	99.3%	1198.6 msec	35.9 bps	5592.8 mj
50	99.2%	1170.6 msec	36.5 bps	5648.7 mj
70	98.8%	1201.3 msec	37.3 bps	5703.4 mj
90	98.6%	1241.1 msec	38.8 bps	5767.5 mj

Table 10: Criteria Values Depend :Packet Size

From the results of the above tables, the following curves were constructed:

✓ Figure 54 shows the packets delivered in relation to the packet size

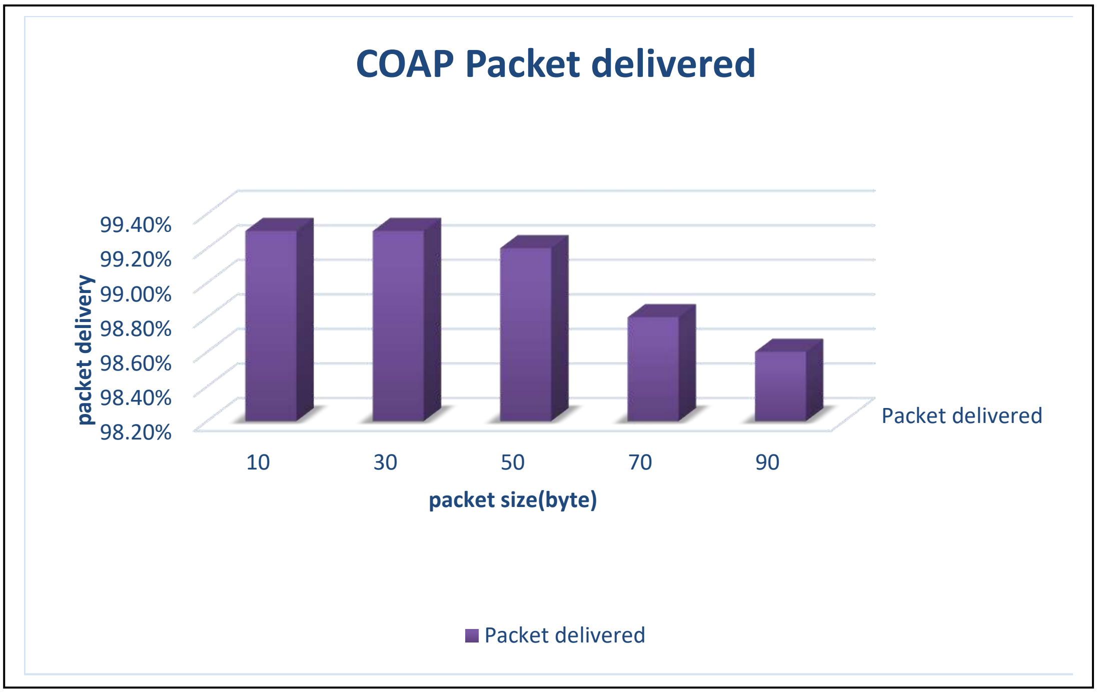


Figure 54: **Curves of packet delivery against the packet size(byte)**

We notice in figure 54 when the packet size is taken less than 50 bytes, the percentage of successfully delivered packets increases, then gradually decreases until it reaches 60%.

✓ Figure 55 shows the delay values in relation to the size of the packet.

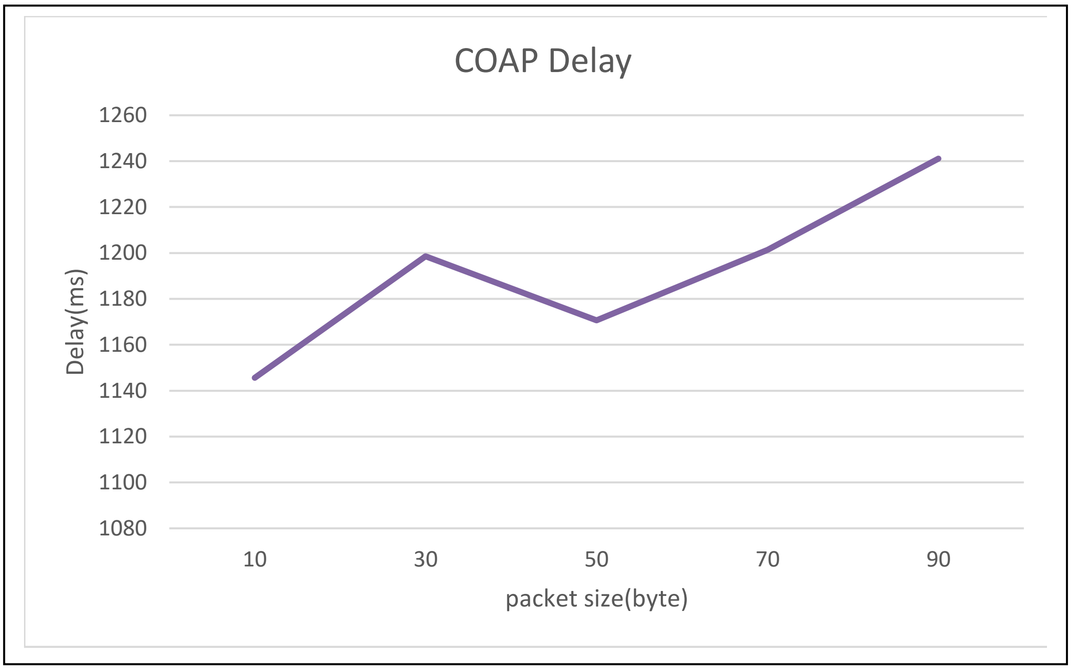


Figure 55: **Curves of Delay(ms) against the packet size(byte)**

We notice in figure 55 when the packet size is taken from 10 to 30 bit, the delay time increases, then it decreases at 30 to 50 bytes, after which it remains gradually increasing.

- ✓ Figure 56 shows throughput(pbs) values in relation to the size of the packet.

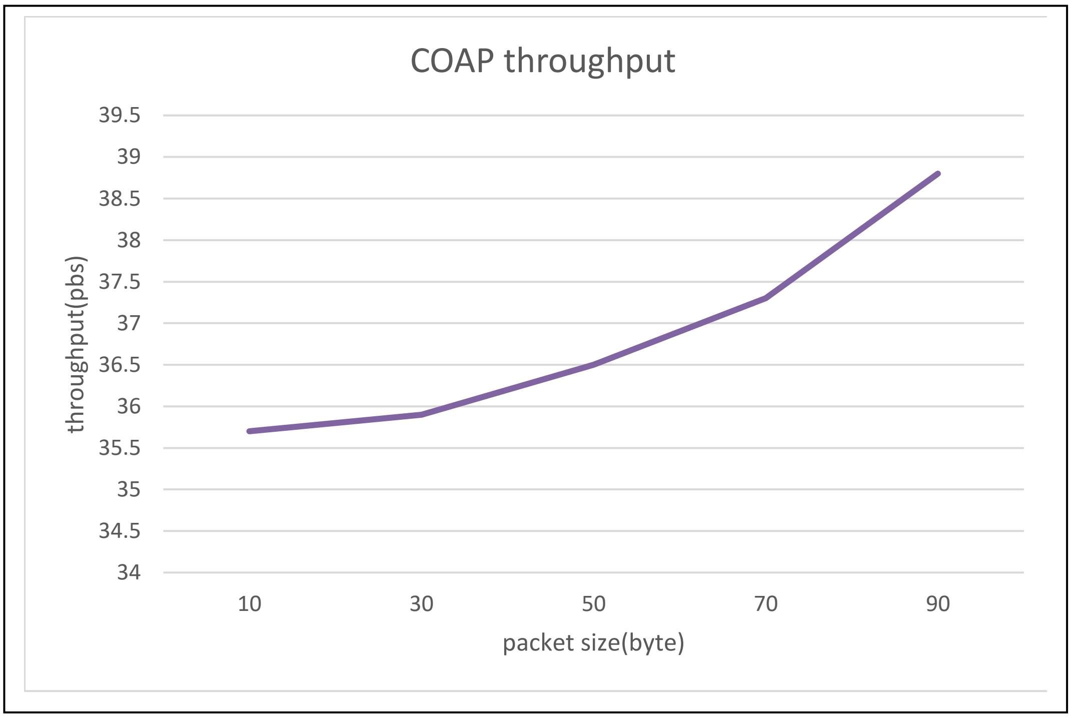


Figure 56: **Curves of throughput(pbs) against the packet size(byte)**

We notice in figure 56 that the packet size increases every time by 20 bytes, the throughput value increases gradually.

- ✓ Figure 57 shows energy consumption(mj) values in relation to the size of the packet.

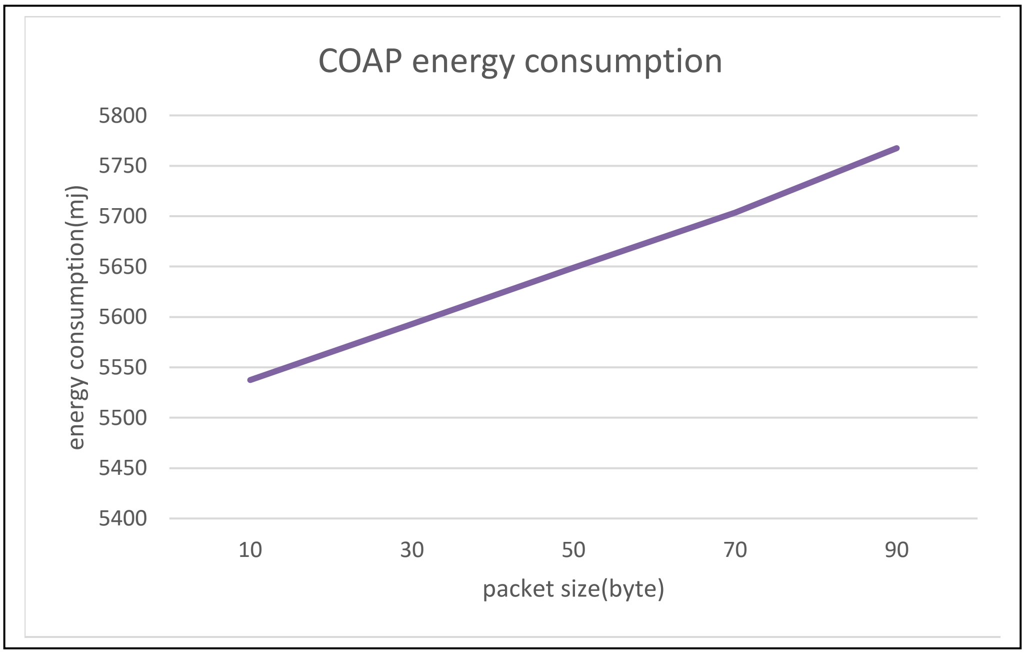


Figure 57: **Curves of energy consumption(mj) against the packet size(byte)**

In Figure 57, we note that for every packetssize auction, the value of energy consumption increases gradually.

Analyze :

When studying packets delivered, delay, throughput, as well as energy consumption with a gradual increase in packets size, we note:

- Increasing the size of the packet does not affect the delay time.
- increase in packets size does not affect the percentage of successful packets delivered in a large percentage meaning that packets loss is very little.

4 EVALUATION RESULTS

After simulations performed with COOJA and analyzing the results shown in the form of a graphical curve, the results were arrived at for performance evaluation:

✗	✓
Not affecte	affected

	Number of server	Packets size	Data transimission interval
Delay	✓	✓	✗
Throughput	✓	✗	✓
Packets deliverd	✓	✗	✗
Energy consumption	✓	✓	✗

Table 11: Compare standards by ratio for different tansmissions

through the table 11, we have reached:

- The delay becomes affected when the number of servers and packet size increases.
- Throughput decreases when the number of servers increases and Data transmission interval increases.
- When the number of servers increases, the successfully delivered packets decrease.
- The increasing number of servers and packets size affects energy consumption.

Through table 12, we have reached:

- Reliability is a burden on the application layer protocol, CoAP in this case, since it bypasses unreliable UDP at the transport layer. The results demonstrate the reliability of the protocol in terms of successful packet reception. Since the retransmission factor allows for increased reliability, the incorporation of the retransmission process of the CoAP protocol guarantees a significant increase in its reliability.
- The CoAP design over UDP has a great impact on power consumption, this is measured through different simulation scenarios which prove the continuous increase in power consumption. The throughput of applications also guarantees interoperability. In real time, the packet size was important.
- The CoAP protcol is not particularly sensitive to increasing the number of servers, despite the low rate of message delivery and without creating abnormal operating conditions (such as stopping the request delay). This ensures the scalability of the bound devices.

	Delay	Throughput	Packets deliverd
Scalabilty	✓	✓	✗
Reliability	✓	✗	✗
Intropabilty	✗	✗	✗
Energy consumption	✗	✓	✓

Table 12: CoAP and IoT Aspect Results

Finally, regarding the scenario that we have relied on in our work , it was concluded that CoAP achieves interoperability, reliability and scalability, and one of its weaknesses is its high energy consumption.

CoAP achieves low throughput and loss of packets, and is a good protocol for connecting to IoT applications.

5 CONCLUSION

In this chapter, we presented an illustrative study to evaluate the performance of the protocol through the measurements extracted from the third chapter, and through the results we concluded that the protocol is compatible with most of the characteristics (scalability, reliability, interpretation) of the Internet of Things, with the exception of energy consumption.

Conclusion:

Conclusion

The Internet of things is one of the biggest changes in innovation today, and it is very important that its products must be reliable and efficient, and respect all the required characteristics.

In this work, we focused on the protocols part due to the multiplicity and difficulty of choosing the optimal protocol for the user, and specifically, we paid attention to evaluating the performance of the CoAP protocol through simulations in the COOJA network simulator under very complex conditions to see how effective this protocol .

Among the programs and tools, we used in this research is the COOJA emulator based on the Contiki-os operating system and the californium eclipse .in order to establish a connection between the clients and the servers to study and evaluate the protocol through some criteria: energy consumption, delay, throughput and successfully delivered packets in order to find out how effective they are with characteristics Internet of Things This work was carried out in four phases:

• phase 1: a general study on the field of the Internet of things, its advantages, its use, characteristics, and protocols

• phase 2: a theoretical understanding of CoAP and a theoretical study of it with the characteristics of the Internet of things

• phase 3: How to work with the COOJA and californium eclipse emulator and get the results

• phase 4: Efficacy and outcome study and protocol evaluation

We encountered many hurdles in our work, but the previous theoretical description of the protocol was rich for us.

Finally, regarding the scenario that we have relied on in our work , it was concluded that CoAP achieves interoperability, reliability and scalability, and one of its weaknesses is its high energy consumption.

CoAP achieves low throughput and loss of packets, and is a good protocol for connecting to IoT applications.

And through the end of our study, we have to think about some suggestions that can be added in the future, including:

•Evaluate the performance of CoAP with the characteristics that we did not use

•Validation by mathematical analysis method formal

•Improvement in the CoAP protocol so that it becomes the basic protocol in the Internet of things application layer(energy consumption) This is done by reducing the size of CoAP protocol's algorithms for the reduce processing time or adding a sleep feature to the node when it is not active.
At the end of this note, we have added several skills in evaluating the performance of the protocol by simulation method.

Bibliography:

Bibliography Documents

[1] Cirani, S.; Davoli, L.; Ferrari, G.; Medagliani, P.; Picone, M.&Veltri, L. "A Scalable And Self-Configuring Architecture For Service Discovery In The Internet Of Things". Ieee Internet Of Things Journal, Ieee, 1, 508-521.2014

[2] SumeetThombre , RaihanUl Islam , Karl Andersson , Mohammad Shahadat Hossain, "Performance Analysis Of An Ip Based Protocol Stack For Wsns", Pervasive And Mobile Computing Laboratory, Lulea University Of Technology, Se-931 87 Skellefte ˚ A, Sweden, 978-1-4673-9955-5/16/$31.00 ©2016.

[3] August Betzler , Javier Isern , Carles Gomez , IlkerDemirkol , JosepParadells , "Experimental Evaluation Of Congestion Control For CoAP Communications Without End-To-End Reliability", Department Of Network Engineering, UniversitatPolitecnica De Barcelona, Barcelona, Spain, Preprint Submitted To Elsevier, May 18, 2016

[4] AyshaAkther, KaziMasudulAlam, ""An Empirical Study Of CoAP Based Service Discovery Methods For Constrained IoT Networks Using COOJA Simulator",ArticleIeee, 22-24 December, 2017

[5] Kanchana P. Naik, Rakesh Joshi U, "Performance Analysis of Constrained Application Protocol Using Cooja Simulator in Contiki OS" . International Conference on Intelligent Computing,Instrumentation and Control Technologies (ICICICT) 2017.

[6] MonishankerHalder, Mohammad Nowsin Amin Sheikh, Md. Saidur Rahman, Md. Amanur Rahman, "Performance Analysis Of CoAP, 6lowpan And Rpl Routing Protocols Of IoT Using COOJA Simulator", International Journal Of Scientific & Engineering Research Volume 9, Issue 6, June-2018.

[7] Nasir Hussain, RamyaHr, Dalvin Vinoth Kumar, Dr. Senthil, "Performance Evaluation Routing Protocol For Low Power Lossy Network Using COOJA Simulator ".Ms Student, Assistant Professor, Professor,School Of Csa,Reva University, Bangalore.2018

[8] Z. Shelby, Sensinode, K. Hartke, "Constrained Application Protocol (CoAP)," Ijser © 2018

[9] Apostolos P. Fournaris , SpiliosGiannoulis, Christos Koulamas, "Evaluating CoAP End To End Security For Constrained Wireless Sensor Networks" ,Department Of Information Technology-Idlab Ghent University - Imec, Ghent, Belgium, 978-1-7281-1542-9/19/$31.00 ©2019 .

[10] MónicaMartí , Carlos Garcia-Rubio ,Celeste Campo , "Performance Evaluation Of CoAP And Mqtt_Sn In An IoT Environment ", Department Of Telematic Engineering, University Carlos Iii Of Madrid, Avda. Universidad 30 , Proceedings, 31, 49 .2019.

[11]PradyumnaGokhale, OmkarBhat, SagarBhat," Introduction To IoT",International Advanced Research Journal In Science, Engineering And Technology,January 2018

[13] M. Han And H. Zhang, "Business Intelligence Architecture Based On Internet Of Things " Journal Of Theoretical & Applied Information Technology, Vol. 50, No. 1, Pp. 90-95, 2013.

[14] Pierre-Jean Benghozi, Sylvain Bureau, Françoise Massit-Follea, "L’internet Des Objets : Quelles Enjeux Pour L’europe", Éditions De La Maison Des Sciences De L’homme, 2009.

[15] Amina Bounab , Maroua Siakhene, "Implémentation Et Evaluation De Performances D’un Protocole De Routage Multi Chemin Dans L’internet Des Objets", Diplôme De Master, Université Saad Dahlab Blida1,2018/2019.

[16] Sylvain Cherrier, "Architecture Et Protocoles Applicatifs Pour La Chorégraphie De Services Dans L'internet Des Objets", Thèse De Doctorat, Université Paris Est, Soutenue Le 25 Novembre, 2013.

[18]. Miao W., Ting L., Fei L., Ling S., Hui D., ".Research On The Architecture Of Internet Of Things".Ieee International Conference On Advanced Computer Theory And Engineering (Icacte), Sichuan Province, China, Pages: 484-487. 2010.

[19]. Jinxin Z., Mangui L.,. "New Architecture For Converged Internet Of Things". International Conference On Internet Technology And Applications, Beijing, China, Pages: 1 - 4.2010.

[20]. Zhang J, Liang M.,. "New Architecture For Converged Internet Of Things" .Ieee International Conference On Internet Technology And Applications, Wuhan, China, Pages: 1- 4.2010

[21]. Inge G.,. "Architecture For The Internet Of Things (IoT): Api And Interconnect". 2nd International Journal Of Computer Networks (Ijcn),) : International Conference On Sensor Technologies And Applications, Cap Esterel, France, Pages: 802-807.2015.

[22] Standard, O. A. S. I. S. "Oasis Advanced Message Queuing Protocol (Amqp) ".Url Http://Docs. Oasis-Open. Org/Amqp/Core/V1. 0/Os/Amqp-Core-Complete-V1. 0-Os. Pdf. Version 1.0., 2012.

[23] Shelby, Z., Hartke, K., & Bormann, C . "The Constrained Application Protocol" .(2014).

[24] Ouadah Abdenour, " Modélisation Et Vérification Formelle D’un Protocole CoAP Pour L’internet Des Objets", Master, Université Mohamed Boudiaf De M’sila, Soutenue 2018/2019.

[25] Keyur K Patel , Sunil M Patel.PG Scholar1 Assistant Professor. Internet of Things-IOT: "Definition, Characteristics, Architecture", Enabling Technologies, Application and Future Challenges Department of Electrical Engineering. ISSN 2321 3361 © 2016 .

[27] Kiljander J, D’elia A, Morandi F, Hyttinen P, Takalo-Mattila J, Ylisaukko-Oja A, SoininenJp, "Semantic Interoperability Architecture For Pervasive Computing And Internet Of Things". Ieee Access 2:856–873 .CinottiTs (2014).

[28]. T. Salman And R. Jain, “Networking Protocols For Internet Of Things,” Pp. 1–28, 2013.

[29]. Bello O, Zeadally S, BadraM ."Network Layer Inter-Operation Of Device-To-Device Communication Technologies In Internet Of Things (IoT) ". Ad Hoc Networks 0:1–11.2016.

[30] Bauer M, Davies J, Girod-Genet M, Underwood M " Semantic Interoperability For The Web Of Things".2016.

[31]. T. N. Jagatic, N. A. Johnson, M. Jakobsson, And F. Menczer, "Social Phishing." Communications Of The Acm 50, No. 10 : 94-100.2007.

[34] B. Babovic Et Al.: " Web Performance Evaluation For IoT Applications",Ieee Access, Page 6989-6990. Volume 4-Nov 2016.

[35] AlabbasAlhajAli . "Constrained Application Protocol (CoAP) For TheIoT ". Delivred May 2018

[36] LudmillaLucioSilva . "Internet Of Things Pros And Cons Of CoAP Protocol Solution For Small Devices ". Delivred 15/02/2015

[37] Z. Shelby, K. Hartke And C. Bormann . "Constrained Application Protocol For Internet Of Things "Delivred 5/05/2014

[39]Amélie Gyrard, "Concevoir Des Applications Internet Des Objets Sémantiques Interdomaine", T H È S E, Telecom Paristech, Ecole De L’institut Télécom - Membre De Paristech, 24 Avril 2015

[40] Cirani, S.; Davoli, L.; Ferrari, G.; Medagliani, P.; Picone, M.&Veltri, L. "A Scalable And Self-Configuring Architecture For Service Discovery In The Internet Of Things Ieee Internet Of Things "Journal, Ieee, 2014, 1, 508-521.

[42] AlessandroLudoviciAnd AnnaCalveras . "A Proxy Design To Leverage The Interconnection Of CoAPwireless Sensor Networks With Web Applications ".Dilvred 09/01/2015

[43] IsamIshaq, Floris Van Den Abeele, JeroenHoebeke And Jen Rossey . ".Flexible Unicast-Based Group Communication For CoAP-Enabled Devices". Dilvred 15/04/2014

[44] Zabi Abdelhamid And HammiMessaoud."Abstraction De La Couche Hardware Par Un Serveur Web Embarqué Pour Le Web Des Objets" .Dilvred 07/06/2017

[45] Majda Moussa. "Vérification Et Configuration Automatiques De Pare-Feux Par Model Checking Et Synthèse De Contrôleur", Université De Montréal,Mémoire De Maitrisées Sciences Appliquées, 2014, P 6.

[46]WassilaKorichi, "Partage De Données En Environnements Mobiles Ad Hoc", Magistère En Informatique, UniversiteKasdi Merbah Ouargla,P92

[47] Gérard Fleury, Philippe Lacomme - Alain Tanguy, "Simulation A Evénements Discrets", Collection *Algorithmes* Dirigée Par Gérard Dreyfus, Groupe Eyrolles, 2007,

[49] L. B. Saad, C. Chauvenet, And B. Tourancheau, “Simulation Of The Rpl Routing Protocol For Ipv6 Sensor Networks: Two Cases Studies,” In Proc. Of The 2011 International Conference On Sensor Technologies And Applications (Sensorcomm’11), Nice, France. Iaria, September 2011.

[50] Anne Canteaut , Cours Pdf ,Programmation En Langage C

[52]Jean-Christophe Routier, "Cours Javascript ,Licence 1", SesiUniversit´E Lille 1

[53] "Introduction A La Programmation En Langage Python" , Université Paris-Sud ,Méthodologie Licence Mpi S2 - Année 2015-2016

[54] Bruno, A., Di Franco, F. And Rasconà, G. Smart Street Lighting. Ee Times. 2012

[55] Velaga, R. And Kumar, A. 2012. "Techno-Economic Evaluation Of The Feasibility Of A Smart Street System: A Case Study Of Rural India". Procedia Social And Behavioral Sciences. 62, 1220-1224.

[56] Sylvain Cherrier, "Architecture Et Protocoles Applicatifs Pour La Chorégraphie De Services Dans L'internet Des Objets, Thèse De Doctorat", Université Paris Est, Soutenue Le 25 Novembre, 2013.

[59]. Bill Burke: “Restful Java WithJax-Rs”, O'reilly, Copyright .Usa© 2010,

[60] Ludmilla Lucio .Silva,CompusHamosamd . "Internet Of Things – Pros And Cons Of CoAP Protocol Solution For Small Devices". UnivaerstelSbacken . .02-15.2016

[61] Roy Thomas Fielding : "Architectural Styles And The Design Of Network-Based Software Architectures", These De Doctorat, University Of California, Irvine, 2000.

Web Bibliography

[12]Https://Www.Google.Com/Search?QHistoryIn nternet Of Things,21:44,18/08/2020

[17]Http://Www.Composelec.Com/Reseau_De_Capteurs_Sans_Fil.Php,21:08,17/03/2020.

[26]Https://Whatis5g.Info/Energy-Consumption/11:23;16/07/2020.

[32] Dr Omar Elloumi, Président De La Plénière Technique Onem2m,Url : Http://Www.Epdtonthenet.Net/Article/124327/IoT-Ecosystem-Expands-Significantly-With-New-Global-Standards.Aspx.

[33] Sécurité Des Objets Connectés - Travaux Des Auditeurs, P 16, Url : Http://Images.Cigref.Fr/Publication/2014-Inhesj-Cigref-Securite_Objets_Connectes. .06/03/2020 8:00 Am

[38]Https://Dzone.Com/Articles/CoAP-Protocol-Step-By-Step-Guide 06/03/2020 8:00 Am

[41]https://fr.wikipedia.org/wiki/CoAP#Applications_pratiques; 21:57,31/08/2020.

[48]www.**eclipse**.org ; 21:57,31/08/2020.

[51]www.**java**.org ; 21:57,31/08/2020.

[57] Https://Www.Engineersgarage.Com 20/03/2020 19:00 Pm

[58] Https://Medium.Com/@Harshhvm/What-Is-CoAP-Protocol-CoAP-Protocol-Introduction-Overview-3e8bac4d7f8e,21:57,31/08/2020.