Acknowledgements

After all, I thank Allah,

I am grateful to my supervisor Mrs. Somia Sahraoui for help me more

I would like to thanks my jury members: the president Pr Cherif Foudil and the examiner Mrs Benseghir Nadia

I would like to thank everyone who helped us directly or indirectly in the production of this report. Among others:

- The computer science department of the University of Biskra through his department head Mr Babahenini Mohamed Chawki
- All our teachers Especially(Dr.Kerdoudi,Dr.Bitam,Dr.Hattab,Dr.Gasmi
- All my firend and sister's Especially(Salah,Majda,Gasmi,Taki,Kacem,Manel,Rachid,Souab)
- My dear aunt, who died shortly before, graduated And this is the work that is revealed to her,rabi yarhmik my dear aunt

Abstract

IoT a newly emerging term meant the new generation of Internet (network) that allows the understanding of interconnected devices over the Internet Protocol. These devices include sensors, sensors, sensors and various artificial intelligence tools and others. This definition goes beyond the traditional concept of connecting people with computers and smartphones over a network One global and through traditional known IP. What distinguishes Internet things is that it allows the person to be free from the place, that is, the person can control the tools without having to be in a specific place to deal with a particular device.

However, despite the advantages provided by the Internet, there is a problem that remains difficult to protect the information transmitted through the Internet Protocol and this is due to the large number of connected devices in the network, which reached in 2020 more than 20 million and this made hackers Internet make it a treasure for them In this study, we chose to study the actions and methods used to reduce the risks faced by the Restricted Application Protocol (COAP). This protocol, which relies only on the transition protocol, Serve ((UDP, because Internet things in terms of transmission of information based on the user data protocol because it is the best in terms of speed compared to the Transmission Control Protocol (TCP).

There have been numerous studies and researches on the field of protection in relation to the restricted application protocol. Among these studies is the study of the Indian researcher Raza, who relied on the technology of transport layer in the data schema (DTLS), which relies mainly on the technology of encryption between the information transmitted and this leads to the nature of confidentiality The network hackers will have great difficulty getting it

This Project aims at clarifying the restricted application protocol, clarifying some of the attacks that may be exposed, and the studies that have been carried out in this field, as well as simulating a DOS attack on a network using the Cooja program.

Key words:
IOT, IDS, Security, Cooja.

Abbreviations

DoS **D**enial **O**f **S**ervice
HIDS **H**ost **b**ased **I**ntrusion **D**etection **S**ystem
IDS **I**ntrusion **D**etection **S**ystem

IoT **I**internet **O**f **T**hings
NIDS **N**etwork **b**ased **I**ntrusion **D**etection **S**ystem
CoAP **C**onstrained **A**pplication **P**rotocol
MQTT **M**essage **Q**ueue **T**elemetry **T**ransport
UDP **U**ser **D**atagram **P**rotocol
XMPP **E**xtensible **M**essaging and **P**resence **P**rotocol
DTLS **D**atagram Transport Layer Security **T**ransport **l**ayer**S**ecurity
M2M **M**achine **To** **M**achine

TCP **T**ransmission **C**ontrol **P**rotocol
IP **I**nternet **P**rotocol
RFID **R**adio **F**requency**I**Dentification

Contents

List of Figures

List of Tables

2

General Introduction

IoT has emerged as a result of convergence of many factors, including ease of communication and in addition to reduce the problems of communication, but this is not enough because of the change of the network and composition of several things (Devices) have become vulnerable to many attacks and this is because the information has been distributed to several places (devices) .

In the information society, the security of computer systems is a crucial issue. The control of the information processed and shared within these systems is a problem that is all the more delicate as the number of users of these systems is important. With more than 24 billion devices connected by 2020 [2].

IoT has many layers (Service layer, application layer, physical layer). The application layer is very important as it contains the top-level communication protocols that generate and handle the sensitive IoT data. Consequently, the integration of efficient security mechanisms in the application layer is very important. In this work, we have addressed a detective solution against DoS threats targeting CoAP protocol.

This document is organized into four chapters. The first chapter presents generalities on IoT. The second chapter presents the protocol CoAP, while the third chapter studies the security issues in CoAP. Finally, chapter four brings the description of the solution that we have been working on, as well as the evaluation results. .

Chapter 1

Generalities about IoT

1.1 Introduction

Today, we are living in the era of smart technologies which represents a "ubiquitous computing" or “web 0.3” [3], and this leads us to the need to keep pace with these smart technologies to study or know how to use Internet of Things (IoT) has emerged strongly as a more prosperous area to express this kind of a new technology [3].
It is not the first technology in this field, but also the cloud computing technology has been used to represent the ubiquitous computing world[3].
The vision of IoT according to Kevin’s vision was to enable networked devices to propagate their information about physical world objects through the web. In recent years, the most of the IoT proposed architectures are used, web semantic to publish information through the social networks; for instance, the iPhone has innovated service is Nike + iPod to record information and published it on the social networks and the social network [4].
In this chapter, we will give the history and some definition for IoT, challenges and Researches works and many other generalities.

1.2 M2M History :

- The first telemetry system was rolled out in Chicago way back in 1912. It is said to have used telephone lines to monitor data from power plants.
- Telemetry expanded to weather monitoring in the 1930s, when a device known as a radiosonde became widely used to monitor weather conditions from balloons.
- In 1957 the Soviet Union launched Sputnik, and with it the Space Race. This has been the entry of aerospace telemetry that created the basis of our global satellite communications today.

- Broad adoption of M2M technology began in the 1980s with wired connections for SCADA (supervisory control and data acquisition) on the factory floor and in-home and business security systems.

- In the 1990s, M2M began moving toward wireless technologies. ADEMCO built their own private radio network to address intrusion and smoke detection because budding cellular connectivity was too expensive

- In 1995, Siemens introduced the first cellular module built for M2M [5].

1.3 IoT Definition

1.3.1 Cisco Definition

The Internet of Things (IoT) is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment.

1.3.2 Simple Information of IoT

- Internet of Things (IoT) comprises things that have unique identities and are connected to the Internet.

- The Internet of Things (IoT) is a novel paradigm that is rapidly gaining ground in the scenario of modern wireless telecommunications.

- The basic idea of this concept is the pervasive presence around us of a variety of things or objects – such as Radio-Frequency Identification (RFID) tags, sensors, actuators, mobile phones, etc. – which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals [5].

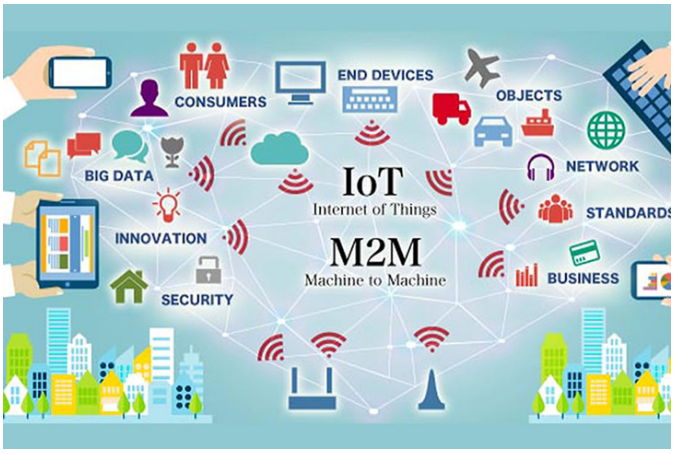

Figure 1.1: IoT[6]

1.4 IoT Domains

Potentialities offered by the IoT make possible the development of a huge number of applications, these can be grouped into the following domains:

- Transportation and logistics domain
- Healthcare domain
- Smart environment (home, office, plant) domain
- Personal and social domain [7]

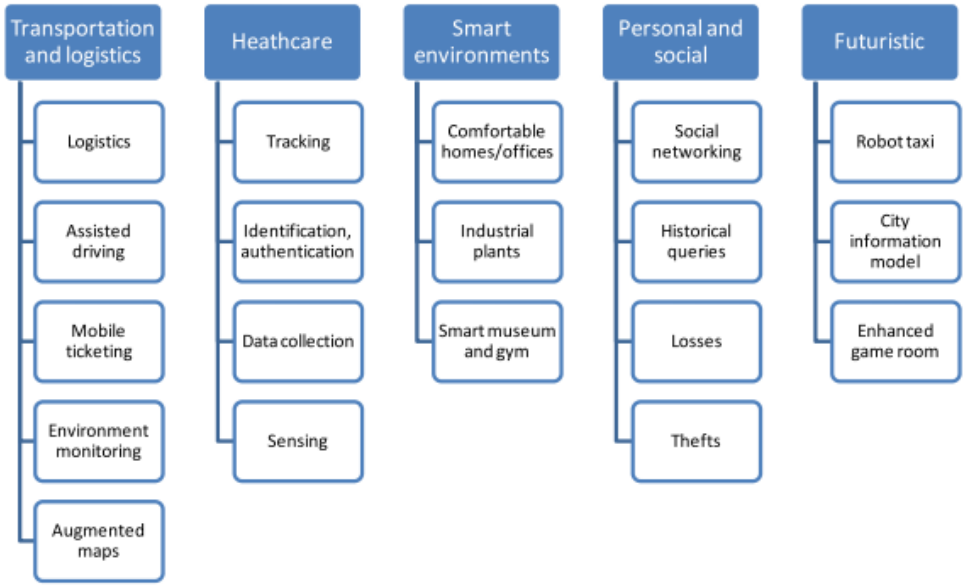

Figure 1.2: Applications Domains and Relevent Major Scenarios [7]

1.4.1 Exemple of Applications

1.4.1.1 Transportation and logistics

Advanced cars, trains, buses as well as bicycles along with roads and/or rails are becoming more instrumented with sensors, actuators, and processing power. Roads themselves and transported goods are also equipped with tags and sensors that send important information to traffic control sites and transportation vehicles to better route the traffic, help in the management of the depots, provide the tourist with appropriate transportation information, and monitor the status of the transported goods [7].

Figure 1.3: Transport Domain Application

1.4.1.2 Healthcare domain

Many benefits provided by IoT technologies to the healthcare domain and the resulting applications can be grouped mostly into the tracking of objects and people (staff and patients), identification and authentication of people, automatic data collection and sensing [11].

There is much application for this domain:

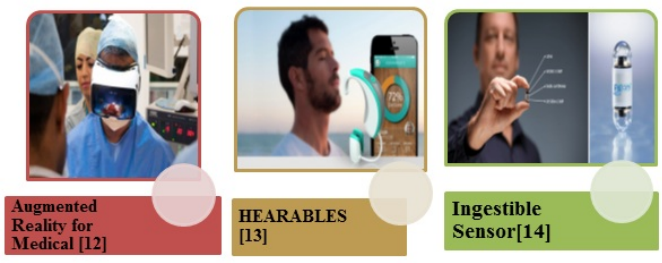

Figure 1.4: Healthcare Domain

1.4.1.3 Smart environments domain

A smart environment is that making its “employment” easy and comfortable thanks to the intelligence of contained objects, be it an office, a home, an industrial plant, or a leisure environment[7].

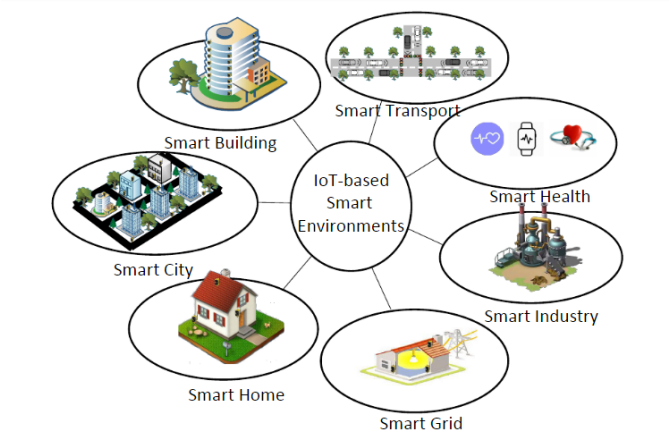

Figure 1.5: IoT Based Smart Environement [12]

1.4.1.4 Futuristic applications domain

The applications described in the previous sections are realistic as they either have been already deployed or can be implemented in a short/medium period since the required technologies are already available. Apart from these, we may envision many other applications, which we herein define futuristic since these rely on some (communications, sensing, material and/or industrial processes) technologies that either is still to come or whose implementation is still too complex [7].

- Robot taxi
- City information model
- Enhanced game room

1.5 IoT Characteristics

They have in IoT 7 important Characteristics (see the figure 1.6)

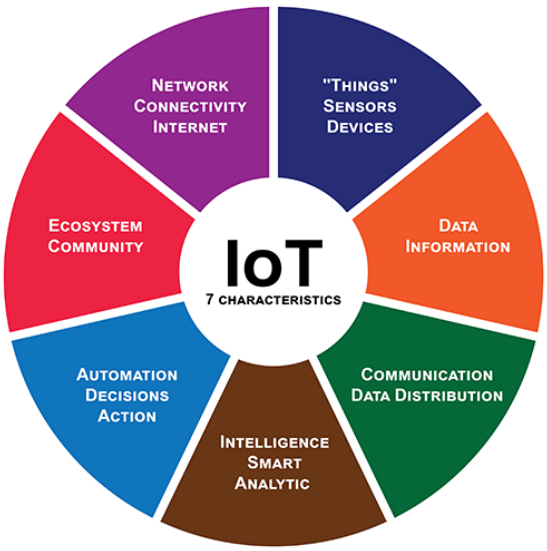

Figure 1.6: Characteristics of IoT

1. Connectivity: this does not need much further explanation. Devices, sensors, they need to be connected: to an item, to each other, actuators, a process and to 'the Internet' or another network

2. Things: anything that can be tagged or connected as such as it's designed to be connected. From sensors and household appliances to tagged livestock. Devices can contain sensors or sensing materials can be attached to devices and items

3. Data: data is the glue of the Internet of Things, the first step towards action and intelligence.

4. Communication: devices get connected so they can communicate data and this data can be analyzed

5. Intelligence: the aspect of intelligence as in the sensing capabilities in IoT devices and the intelligence gathered from data analytics (also artificial intelligence).

6. Action: the consequence of intelligence. This can be a manual action, action based upon debates regarding phenomena (for instance in climate change decisions) and automation, often the most important piece[13].

7. Ecosystem: the place of the Internet of Things from a perspective of other technologies, communities, goals and the picture in which the Internet of Things fits. The Internet of Everything dimension, the platform dimension and the need for solid partnerships[13]

1.6 IoT Architecture

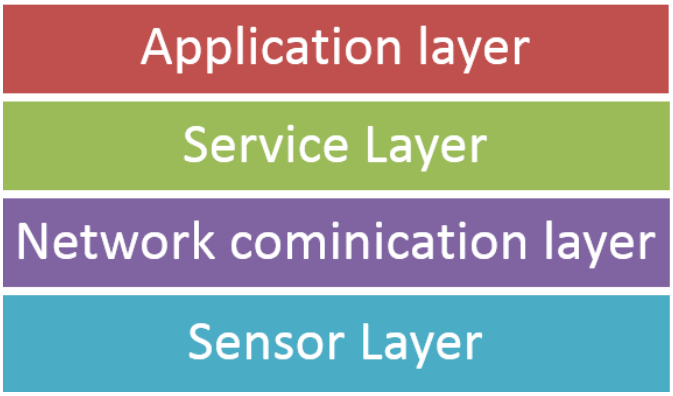

Figure 1.7: Four Layers in IoT

1. Sensing Layer: The first layer of the Internet of Things consists of Sensor-connected IoT devices, it is used to gather data from objects the characteristic of this layer
   - Lowest Abstraction Layer
   - With sensors we are creating digital nervous system
   - Incorporated to measure physical quantities
   - Interconnects the physical and digital world
   - Collects and process real-time information[2]
2. Gateway Network Layer: it provides the infrastructure to support over the wired or wireless connection among things, The various IOT devices of layer 1 need to be connected to the internet via a more powerful computing device called the IoT gateway which primarily acts as a networking device. So, similar to how a WiFi router helps us connect many laptops, phones and tablets to the internet at home, the IoT gateway aggregates data from numerous sensing devices and relays it to the cloud [14].
3. Service Layer: It is used to
   - Create and manage the services required by user application
   - Capturing of periodic sensory data
   - Data Analytics (Extracts relevant information from massive amount of raw data)
   - Streaming Analytics (Process real time data)
   - Ensures security and privacy of data[2]

1.7 Challenges and Recent Research Directories

- Networking:
  Generally, the Networking issue has great relevance on the Internet because it includes some of the important factors which use to manage networks. First of all, traffic and protocols that have a significant impact on the behaviour of the network, these points are mentioned in[15] D. Giusto et al. Sought to deal with networking challenges via mobile Ad-Hoc Network. The authors have used mobile ad hoc networks (MANET) interconnected to fixed networks by a different gateway. In IoT, can't be predicted where the object moved, and the object may be needed to transmit from network to another. The biggest problem is in dynamic gateways change and the difficulty of identifying the location of things. The MANET consists of a number of self-organized mobile nodes or objects and is considered as a way to maintain a connection, additionally Multi-homed ad-hoc is seen as an extension to the existing infrastructure in IoT.

- Routing :
  Routing process means selecting the best path between the source and the destination to complete the communication process successfully. There are various ways to determine the best path based on the communication protocol type such as a number of hops, costs, and bandwidth. Can be classified routing protocols into two main categories are i) Reactive protocols: the path is established after transmission request is made, ii) Proactive protocols: initial path before the request is made. In [12], Sudip Misra et al. proposed the protocol under the name of "fault-tolerant routing protocol" for IoT. This protocol was designed by using learning automata (LA) and cross-layer concept. LA dealing with optimization problems to choose optimal solutions, the need to cross-layer is saving energy of the items of IoT (i.e. RFID).

- Heterogeneity :
  The IoT environment is the best-known example to represent the heterogeneity issue because it contains a plethora of the different devices in their nature; the main objective of IoT is creating a common way to abstract the heterogeneity of these devices and achieving the optimal exploitation of their functionality.

- Interoperability :
  Interoperability concept can be defined as the ability to create systems or devices cooperating with each other in an efficient way. In [16] Jussi et al. sought to use the semantic level interoperability architecture for pervasive the computing and IoT; the architecture relies on the semantic information sharing solutions called "smart-M3".

- Quality of Service (QoS) :
  Ideally, QoS is defined as "the amount of time that is taken to deliver the message from the sender and the receiver" if this time is equal or less than pre-specified time requirement the QoS is achieved. ITU re-defined QoS concept as a degree of conformance of delivering service to the user by the provider with the agreement between them. For QoS assurance, must cope with service models to determine which degree of QoS for each Internet service.

- Scalability :
  Scalability is one of the most important challenges of IoT, which means how to deal with the sustainable growth of the Internet in an efficient manner. In other words, "Scalability is the ability of a system or network to handle the growing scale of any environment without an effect on performance". Currently, the Internet comprises around 9 billion devices with a next era of the Internet which known Web 0.3 or ubiquitous computing it is expected to reach 24 billion devices, the increasing of this number have a broad impact on the performance of the network.

- Virtualization :
  Virtualization is known as the ability to share hardware resources among multiple operating systems. The virtualization technology allows for the multiple operating systems and software like applications or services to run upon the same server through creating more than virtual machine inside the physical machine. The vision of this concept helps to increase the performance of the network via increasing utilization, maximizing scalability, saving cost, etc.
  Actually, there are three areas used to represent the virtualization technology, namely, i) network virtualization, storage virtualization, and server virtualization.

Figure 1.8: Challenges in IoT

1.8 IoT Disadvantages

- Security: IoT creates an ecosystem of constantly connected devices communicating over networks. The system offers little control despite any security measures. This leaves users exposed to various kinds of attackers.
- Privacy The sophistication of IoT provides substantial personal data in extreme detail without the user's active participation
- Complexity: Some find IoT systems complicated in terms of design, deployment, and maintenance given their use of multiple technologies and a large set of new enabling technologies.
- Flexibility: Many are concerned about the flexibility of an IoT system to integrate easily with another. They worry about finding themselves with several conflicting or locking systems[16].
- Compliance: IoT, like any other technology in the realm of business, must comply with regulations. Its complexity makes the issue of compliance seem incredibly challenging when many consider standard software compliance a battle [16].

1.9 IoT Protocols

There are many protocols in IoT:

- Application Layer Protocol:
  1. Constrained Application Protocol (CoAP): We will talk about him in the next chapter.
  2. Message Queue Telemetry Transport (MQTT): It is suitable to connect embedded devices, network with application
  3. Extensible Messaging and Presence Protocol (XMPP): It is suitable for multi-party chatting, voice and video calling and tells the community
  4. Advanced Message Queuing Protocol(AMQP): It is wire-level protocol
  5. Data Distribution Service (DDS): It is used for multicasting application [17]
- Infrastructure Protocols:
  1. Routing Protocol for Low Power and Lossy Networks (RPL): This protocol is standardized by IETF. This routing protocol supports simple and complex traffic models like i. Point to point ii. Point to multi-point iii. Multi-point to point [17]

2. 6LoWPAN (IPV6 Low power wireless personal area network): This standard provides header compression to reduce the data transmission overhead, forwarding to link- layer to support multi-hop delivery.
3. IEEE 802.15.4 (Zigbee): This protocol was created to specify a sub-layer as Medium Access Control (MAC) and Physical Layer (PHY) for low rate wireless private area networks (LR-WPAN).
   It provides reliable communication, operability on different platforms and a large number of nodes [17].

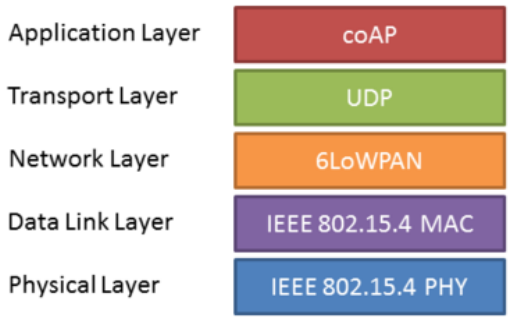

Figure 1.9: Protocols in IoT

1.10 Conclusion

Throughout this chapter, we have presented some relevant generalities on the internet of Things. The next chapter will be devoted to the study of CoAP protocol on which our solution is focused.

Chapter 2

The Constrained Application Protocol

2.1 Introduction

IoT protocols are divided into several parts Application protocol, Infrastructure protocol and influential protocol. In the application layer, there is a protocol that is similar to the HTTP Protocol. That protocol is called CoAP(Constrained Application Protocol). The Internet Engineering Task Force (IETF) working group has designed this protocol to be used for M2M applications, IoT objects and suitable for constrained devices that have limited resources [18].
In this chapter, we will give a presentation of the CoAP protocol and its main features. Then, we will define structure protocol CoAP, after that we will study the main difference between protocol CoAP and other protocol.

2.2 What's Protocol CoAP ?

The Constrained Application Protocol (CoAP) is a web transfer protocol at the application layer intended to be used with constrained devices (e.g., low-power node, sensors, switches or actuators) and constrained (e.g., low-power, lossy) networks [19].
The Constrained Application Protocol (CoAP) was designed for the Internet of Things (IoT) deployments, assuming that UDP [RFC0768] can be used unimpeded [20]. It enables those nodes to be able to talk with other constrained nodes over the Internet [21]. The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation [22].
CoAP is based on REST architecture, the protocol considers the various objects in the network as resources [22].

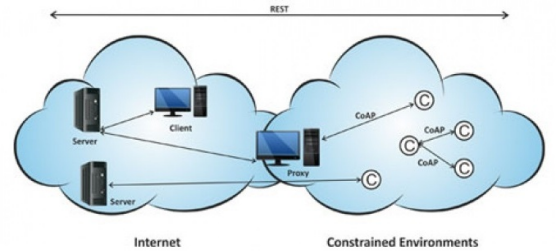

Figure 2.1: Overview of CoAP Protocol for Constraint Devices [22]

2.2.1 REST Architecture

Representational State Transfer (REST) is a style of architecture based on a set of principles that describe how networked resources are defined and addressed. REST is an alternative to SOAP and JavaScript Object Notation (JSON).
It is important to note that REST is a style of software architecture as opposed to a set of standards. As a result, such applications or architectures are sometimes referred to as RESTful or REST-style applications or architectures.
An application or architecture considered RESTful or REST-style is characterized by:

- State and functionality are divided into distributed resources

- Every resource is uniquely addressable using a uniform and minimal set of commands (typically using HTTP commands of GET, POST, PUT, or DELETE over the Internet)

- The protocol is client/server, stateless, layered, and supports caching [23].

In REST architecture, a REST Server simply provides access to resources and REST client accesses and presents the resources. Here each resource is identified by URIs/ global IDs. REST uses various representations to represent a resource like text, JSON and XML. Recently, JSON is the most popular format being used in web services [24].

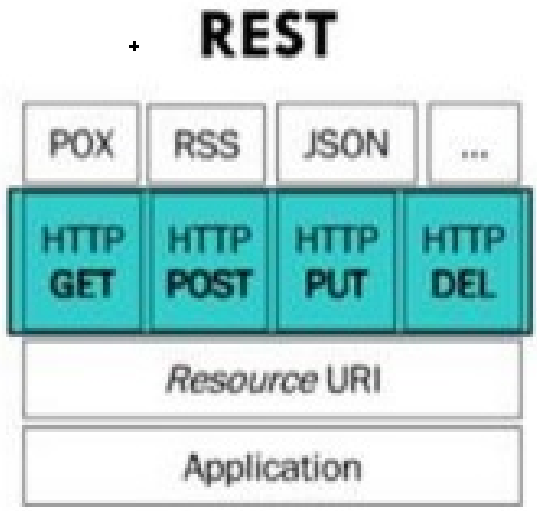

Figure 2.2: Rest Architecture [23]

2.3 The main features of CoAP

CoAP has the following main features:

- Web protocol fulfilling M2M requirements in constrained environments [24].
- UDP binding with optional reliability supporting unicast and multicast requests [24].
- Low header overhead and parsing complexity.
- Asynchronous message exchanges.
- URI and Content-type support.
- Simple proxy and caching capabilities [25,26].

2.4 CoAP Architecture

The CoAP protocol is closely aligned to the traditional web stack based on HTTP. However, UDP is used instead of TCP at the transport layer. CoAP uses binary encoding unlike the textual encoding of HTTP but otherwise, both are based on RESTful APIs and request-response method. This one-to-one mapping between the two stacks makes it possible to interwork the two protocols via a proxy. Within the constrained network, CoAP is used but elsewhere CoAP messages can be translated into HTTP. Such proxies

may reside in the edge of a cloud. It's also possible to bypass such a proxy and deploy CoAP end to end if cloud platforms support this [27,28,29].

* CoAP architecture has two layers, message layer and request/response layer.

2.4.1 Message layer

Is responsible for controlling the message exchange over UDP between two endpoints [30].

2.4.1.1 What's a typical message flow in CoAP

It may be tempting to think of sensor nodes as CoAP clients but in fact, they are typically CoAP servers. But they could also be CoAP clients consuming information available at other devices on the network [31].
In a typical CoAP flow, clients discover resources available at the server by asking for URI path GET /.well-known/core. The server will reply with the content of type application/link-format. Now that the client knows what resources are available, it can request the content of a specific resource (such as temperature) by calling GET /temperature. The server will respond with the value of that resource [31].

2.4.1.2 Message Types

Message Layer supports 4 types message: CON (confirmable), NON (non-confirmable), ACK (Acknowledgement), RST (Reset) [31].

- **The Confirmable CON message** The client then waits for an acknowledgement response from the destination server. If at the end of a time T, the client has not had an answer, he considers that the packet is lost and retransmits the message in a random time interval to control the congestion of the network [31].

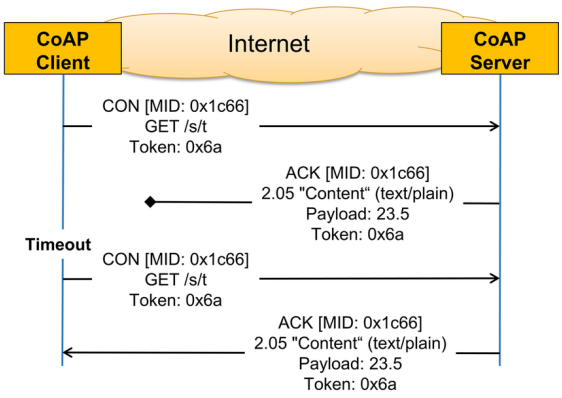

Figure 2.3: Example of CoAP Message (CON) exchange

Figure 2.3 explains the case when the client does not receive an ACK for its CON within a certain time, it retransmits the same CON again until it gets acknowledged or until the client runs out of retransmission attempts.

- The message Non-Confirmable (NON)[31]:
  The client sends the message without waiting for an acknowledgement. In the case where the recipient does not have the ability to perform the request, he can respond with an RST message with the Message.

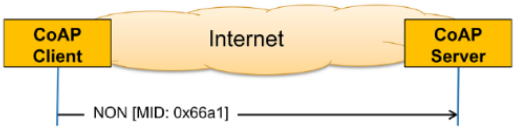

Figure 2.4: Example of CoAP Non-confirmable Message (NON) exchange

- Acknowledgment ACK: This type of message acknowledges the customer and thus ensures the good reception of the message [32].
  An Acknowledgement message acknowledges that a specific Confirmable message arrived. By itself, an acknowledgement message does not indicate success or failure of any request encapsulated in the Confirmable message, but the Acknowledgement message may also carry a Piggybacked Response [33].

- Reset RST message: Allows the server to reject the request sent by the client [32]. A Reset message indicates that a specific message (Confirmable or Non-confirmable) was received, but some context is missing to properly process it. This condition is usually caused when the receiving node has rebooted and has forgotten some state that would be required to interpret the message.
  Provoking a Reset message (e.g., by sending an Empty Confirmable message) is also useful as an inexpensive check of the liveness of an endpoint ("CoAP ping"). Message [32].

2.4.1.3 Message transport

There are 2 types: Reliable and Unreliable message transport

1. Reliable message transport: Keeps retransmission until get ACK with the same message ID. Using default time out and decreasing counting time exponentially when transmitting CON. If recipient fail to process message, it responses by replacing ACK with RST [33].

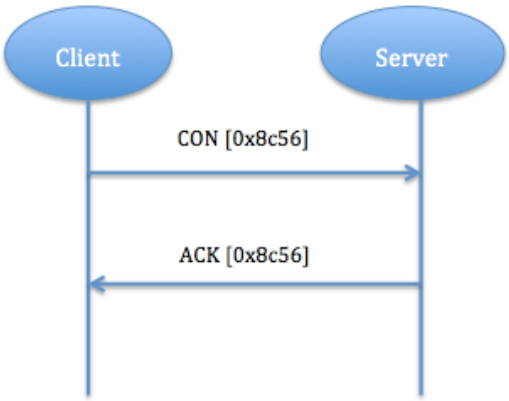

Figure 2.5: Reliable Message Transport [33]

2. Unreliable message transport: transporting with NON type message. It doesn't need to be Acknowledged, but has to contain message ID for supervising in case of retransmission. If recipient fail to process message, server replies RST [33].

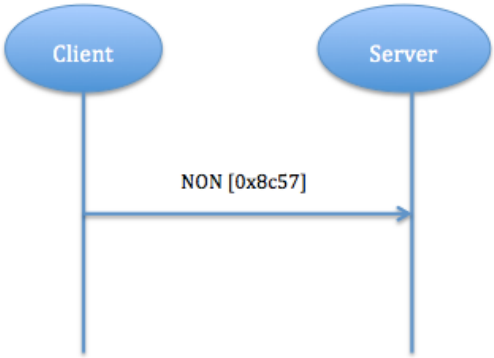

Figure 2.6: Unreliable Message Transport [33].

2.4.2 Request/Response Layer

2.4.2.1 Model

1. Piggy-backed: Client sends request using CON type or NON type message and receives response ACK with confirmable message immediately [34].

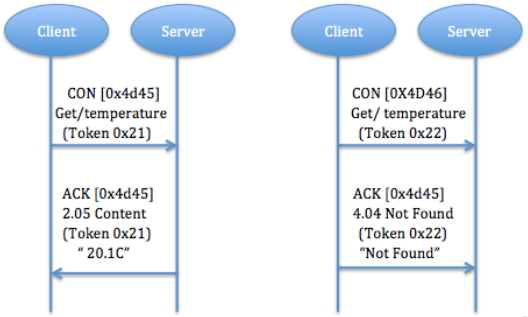

Figure 2.7: The successful and Failure Response Results of GET [35]

2. Separate response: If server receive a CON type message but not able to response this request immediately, it will send an empty ACK in case of client resend this message. When server ready to response this request, it will send a new CON to client and client reply a confirmable message with acknowledgment. ACK is just to confirm CON message, no matter CON message carry request or response [18]. (above figure 2.8)

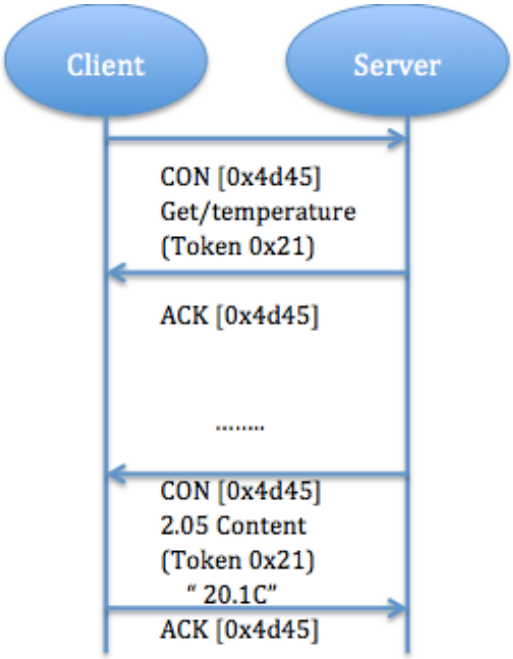

Figure 2.8: A Get request with a separate response [35]

3. Non confirmable request and response: unlike Piggy-backed response carry confirmable message, in Non confirmable request client send NON type message indicate that Server don't need to confirm. Server will resend a NON type message with response [36]

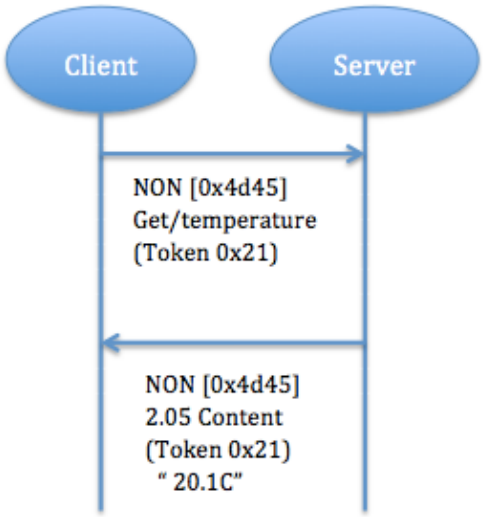

Figure 2.9: Non confirmable request and response [36]

2.4.2.2 Request Method Definitions :

The client requests an action using a Method code on a resource which identified by Uniform Resource Identifier (URI) on a server [37].
The CoAP defines four different method codes:

- GET: operation to retrieve representation in resource identified by the URI request.
- POST: Submits information to be processed to a specified resource. The output result depends on the target resource, usually, results in the target resource being created or updated.
- PUT: requests that the resource identified by the URI be created or updated with the carried information representation
- DELET: requests that the identified resource be deleted [38].

2.4.2.3 Response Code Definitions:

The server sends back to the client a response code indicates the outcome of the request process. There are three classes of Response code

- Success 2.xx : This class indicates that the request has been successfully received and processed. Here are some examples of response codes under this class: 2.01 Created, 2.02 Deleted
- Client Error 4.xx : This class indicates that the request from the client was not valid or has an error. For example, 4.00 for a Bad request or 4.04 Not found this

response code is like the common HTTP 404 which indicate that the request is correct but the server could not find the resource identified by the URI.

- Server Error 5.xx : This class indicates that the server has an error or incapable of processing the request. For example, 5.03 Service unavailable [32].

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

Table 2.1: CoAP Code Status [32]

2.4.3 Message Format:

CoAP messages are encoded in a binary format. The message format is shown in Figure 2.10.
CoAP messages are encoded in a simple binary format. The message format starts with a fixed-size 4-byte header. This is followed by a variable-length token value, which can be between 0 and 8 bytes long.

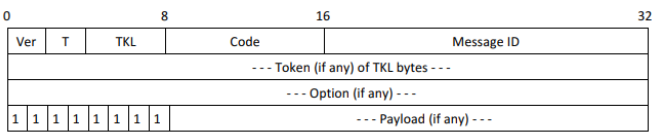

Figure 2.10: Structure/format of CoAP Message [38]

The message format starts with a fixed-size 4 bytes header that contains five fields and defined as follows:

- Version (Ver): 2-bit unsigned integer Indicates the protocol version number
- Type (T): 2-bit unsigned integer Indicates the message type CON (0), NON (1), ACK (2), RST (3).
- Token Length (TKL): 4-bit unsigned integer Indicates the number of bytes of the Token field.
- Code: 8-bit unsigned integer Indicates the status of a client and server request and response.
- Message ID: 16-bit unsigned integer Contains the Message ID that used to detect message duplication as well as to match ACK/RST to CON/NON message types.

Message contents	Description	Bytes
Version	• Protocol version	2 bits
Type	• Confirmable (CON) - Must be acknowledged by the receiver with an ACK packet. • Non-Confirmable (NON) - messages that do not require acknowledgement • Acknowledgement (ACK) - Acknowledge a confirmable message • Reset (RST) - Reject a confirmable or remove an observer	2 bits
Token Length	• Specifies the length of the token as 0 to 8 bytes	4 bits
Code	• Response code analogous to HTTP response codes, can be a success message, client error, or server error	8 bits
Message ID	• Identifier for each message sent, which in most implementations are likely sequentially assigned; not very effective for security purposes	16 bits
Options	• Can be set to include one or more options, including a subset of what's available via HTTP headers	-
Payload	• Body of message or specific format, if any	-

Table 2.2: Details of CoAP Message Components

2.5 CoAP vs Other Protocol

2.5.1 CoAP vs HTTP

Both protocols share same structure of REST and use the same methods. CoAP is defined as the compressed version of HTTP. However, HTTP is an old protocol that is stable and well used, while CoAP is very new and research is still ongoing. HTTP is a heavy weight protocol on the internet as it required heavy code space implementation and network usage [39].
In contrast CoAP is specially designed to be light in both implementation and in network usage which make it more suitable for the small or class-1 devices. HTTP is supported

with TCP while CoAP uses UDP (asynchronous request/repose) that can simply carry message and transmit lost packets [39].

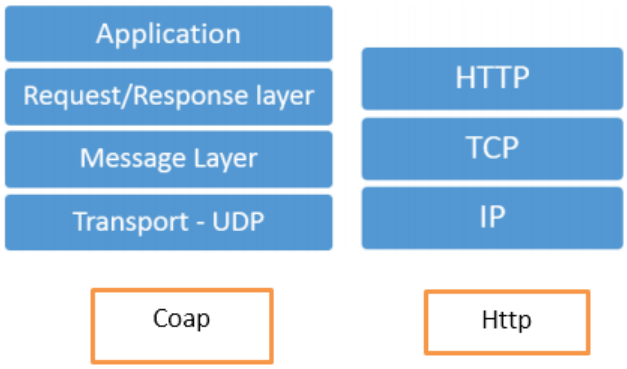

Figure 2.11: CoAP layer vs HTTP layer

2.5.2 CoAP vs Mqtt

The MQTT and CoAP, both are designed to consider the long term vision that they need to be used in lightweight environments. Both works well with low power and network constrained devices. So the choice really depends on the application. If a M2M network has to be created where messages must publish from one node to multiple interested nodes, the best choice will be to use MQTT protocol. But if a M2M network has to be created where commands must be sent to among nodes, CoAP will be the best choice for that. For example to control an AC from the smart phone, CoAP would be the smarter choice [39].

	MQTT	CoAP
Messaging transformation	Multipoint-to-Multipoint commination	Point-to-Point communication
	Topics	URI
Transport	TCP-Based	UDP-Based
Architecture	Publish/Subscribe model	Request/Response model
Layers	Single layered	Two sublayer: Request/Response and Transaction
Security	SSL/TLS	SSL/DTLS
Reliability	3 QoS levels	4 QoS levels
Performance analysis	Lower delays	Lower overhead and lower packet loss

Table 2.3: MQTT VS CoAP

Protocol	*Transport*	*Messaging*	*2G,3G,4G (1000's)*	*LowPower and Lossy (1000's)*	*Compute Resources*	*Security*	*Success Stories*	*Arch*
CoAP	UDP	Rqst/Rspnse	Excellent	Excellent	10Ks/RAM Flash	Medium - Optional	Utility field area ntwks	Tree
Continua HDP	UDP	Pub/Subsrb Rqst/Rspnse	Fair	Fair	10Ks/RAM Flash	None	Medical	Star
DDS	UDP	Pub/Subsrb Rqst/Rspnse	Fair	Poor	100Ks/RAM Flash +++	High-Optional	Military	Bus
DPWS	TCP		Good	Fair	100Ks/RAM Flash ++	High-Optional	Web Servers	Client Server
HTTP/ REST	TCP	Rqst/Rspnse	Excellent	Fair	10Ks/RAM Flash	Low-Optional	Smart Energy Phase 2	Client Server
MQTT	TCP	Pub/Subsrb Rqst/Rspnse	Excellent	Good	10Ks/RAM Flash	Medium - Optional	IoT Msging	Tree
SNMP	UDP	Rqst/Response	Excellent	Fair	10Ks/RAM Flash	High-Optional	Network Monitoring	Client-Server
UPnP		Pub/Subscrb Rqst/Rspnse	Excellent	Good	10Ks/RAM Flash	None	Consumer	P2P Client Server
XMPP	TCP	Pub/Subsrb Rqst/Rspnse	Excellent	Fair	10Ks/RAM Flash	High-Manditory	Rmt Mgmt White Gds	Client Server
ZeroMQ	UDP	Pub/Subscrb Rqst/Rspnse	Fair	Fair	10Ks/RAM Flash	High-Optional	CERN	P2P

Table 2.4: CoAP VS Other protocol

2.6 When used Coap

CoAP should be on priority for the following three factors:

- Quality of service with confirmable message .
- When multicast support is needed.
- Very low overhead and simplicity

2.7 Disadvantages of CoAP:

There are also some disadvantages of CoAP:

- Message unreliability: UDP does not guarantee the delivery of datagrams. CoAP adds a method to request a confirmation acknowledgement to confirm the message was received. This does not verify that it was received in its entirety and decoded properly.
- Standards are still maturing: CoAP is still evolving, although there is a lot of market momentum behind it. It is likely to mature quickly as use becomes more widespread.
- NAT issues: Network Address Translation (NAT) devices are commonly used in enterprise networks and cloud environments. CoAP can have problems communicating with devices behind a NAT since the IP can be dynamic over time.

2.8 CoAP Application

We can use this protocol Smart Grid and Building Automations, Legacy protocols like BACnet may be mapped as CoAP resources and respective communication data may be mapped to CoAP messages to have automated building applications. CoAP multicasts may be used for effective group communications like every sensor of a type in a room.

2.8.1 CoAP Application for Smart Homes

Information application, control equipment and communication equipment in Smart home networks have the characters of low-cost and unimportant.
CoAP could be seen as the best protocol choice for home communication networks. Smart home network provide controlling and monitoring power of home devices.
Energy control systems employ smart socket management and monitor power consuming equipment to provide voltage, current and other energy information [40].
It could understand accident warning, remote control and dynamic energy saving. The system structure is shown in below fig. Every data collection node with CoAP client could exchange information with other nodes. CoAP could both be installed in LAN or Internet [35].
In this system, CoAP-HTTP proxies are employed to provide HTTP client connection to CoAP resources and vice versa.
In system networking, date collection nodes consist of one proxy, smart socket and wireless data collection module. Energy information and environment information of equipment is collected by the smart socket and transported to data collection module through wireless channel, then send serial data to proxy to process and pack data. Control server analyzes all the data and stores them in database. The system integrates home network and Internet, users can access system webpage to remotely control switch, manage configuration, query energy consumption, etc. (Figure 2.12)

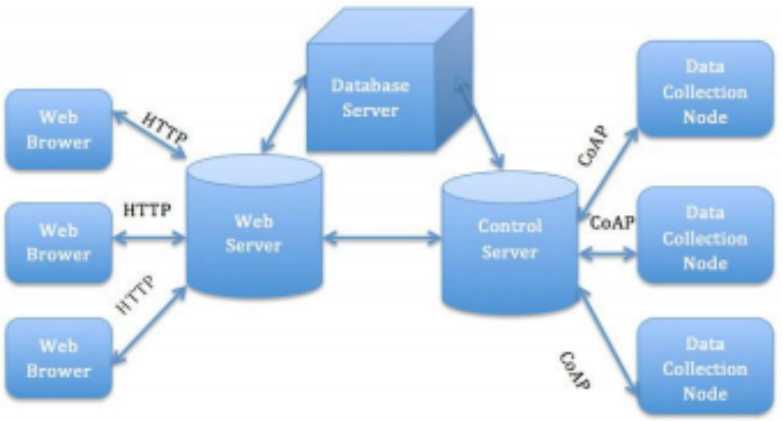

Figure 2.12: Energy Control System [40]

2.9 Conclusion

In this chapter, we have deeply presented CoAP protocol that will make a platform for our realized work. The next chapter will be dedicated to the presentation of the security problems of CoAP along with a review on the existing countermeasure solutions.

Chapter 3

Study of Security Aspects of Protocol CoAP

3.1 Introduction

In the previous chapter, we introduced some important parts in CoAP protocol that are related to its security problems such as the potential vulnerabilities. The current chapter, will studies some security aspects in that protocol and highlight some relevant related works especially those that addresse the protection of CoAP against DoS threats.

3.2 Some Conventional Concepts

3.2.1 Security

Security is a key element in enabling widespread adaptation of IoT technologies and applications. Since IoT is the integration of multiple heterogeneous networks, it should address compatibility issues between different networks that are subject to security issues [41].

3.2.2 Securing a Network

involves taking into account all possible risks, such as malicious attacks, accidents, software or hardware defects, or human errors and reducing them as much as possible [42].

3.2.3 Security Services

Computer security is a set of means implemented to reduce the vulnerability of a system against threats to ensure: authentication, confidentiality, identification and non-repudiation [43][44][45].

- Authentication: to enable secure communication between devices in the IoT, authentication must first be ensured.

- Confidentiality: regarding confidentiality, this service is to guarantee the only entities that can access and modify the data are only authorized entities and as it ensures that the information is read only by authorized persons.

- Non-repudiation: The integrity of the data is to determine if the data has not been altered or modified during the communication .

- Identification: Whose role is to define the identities of the users, IoT is known by the inclusion of various objects so they must be identified in a way to ensure secure communication between them.
  There are two ways to identify these objects, the first is to physically repair them by using RFID, QR code and the second is to equip each one of these objects with a means of wireless communication [45].

3.3 Security attacks in IoT

3.3.1 Dos and DDOS attack

Denial of service or distributed denial service of service attack is an attempt to make a machine or network resource unavailable to its intended users. Basically sending more requests than a capacity of web server [43].
Its simple When the attack is performed from one source to one destination its DOS attack (see the figure 3.1). When attack is performed from multiple source to one Destination its DDOS.(see the figure 3.2)

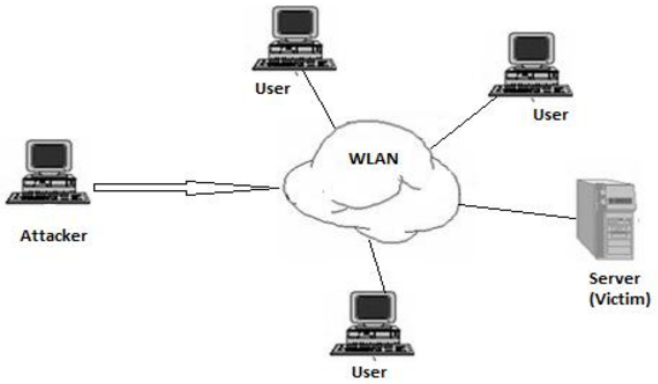

Figure 3.1: Dos attack

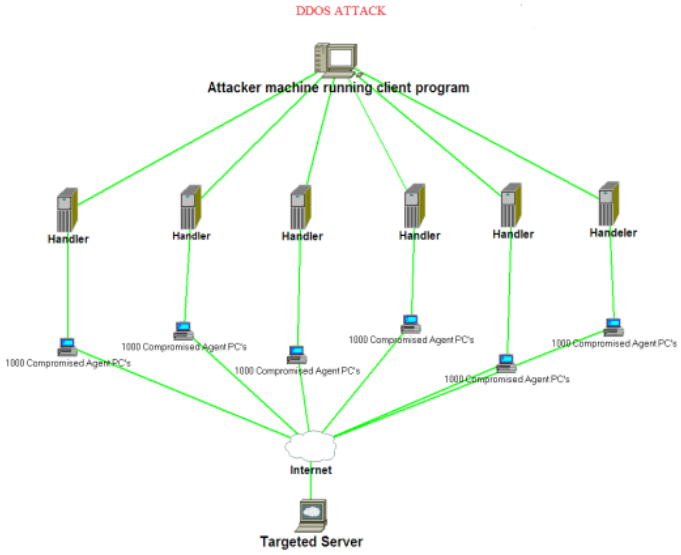

Figure 3.2: DDos attack

3.3.1.1 Types of DoS Attack

We can see from figure 3.3 what is a DoS attack for each layer : Physical, MAC, network and application layer

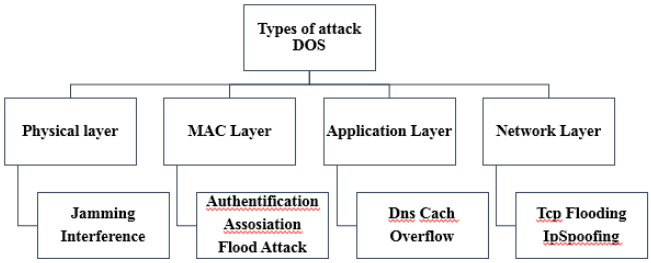

Figure 3.3: Types of DoS attacks

3.3.1.2 Common DDoS attack types

They have many attacks commently used ddos attacks like [45]

a- UDP Flood : A UDP flood, by definition, is any DDoS attack that floods a target with User Datagram Protocol (UDP) packets. The goal of the attack is to flood random ports on a remote host. This causes the host to repeatedly check for the application listening at that port, and (when no application is found) reply with an ICMP 'Destination Unreachable' packet[46].

b- Ping of Death: a ping of death ("POD") attack involves the attacker sending multiple malformed or malicious pings to a computer. The maximum packet length of an IP packet (including header) is 65,535 bytes. However, the Data Link Layer usually poses limits to the maximum frame size – for example 1500 bytes over an Ethernet network. In this case, a large IP packet is split across multiple IP packets (known as fragments), and the recipient host reassembles the IP fragments into the complete packet.

c- HTTP Flood : In an HTTP flood DDoS attack, the attacker exploits seemingly-legitimate HTTP GET or POST requests to attack a web server or application. HTTP floods do not use malformed packets, spoofing or reflection techniques, and require less bandwidth than other attacks to bring down the targeted site or server. The attack is most effective when it forces the server or application to allocate the maximum resources possible in response to every single request.

d- Botnet : A botnet refers to a group of computers which have been infected by malware and have come under the control of a malicious actor. The term botnet is a portmanteau from the words robot and network and each infected device is called a bot. Botnets can be designed to accomplish illegal or malicious tasks including sending spam, stealing data, ransomware, fraudulently clicking on ads or distributed denial-of-service (DDoS) attacks.(see the figure 3.4).

e- DNS amplification attack : all amplification attacks exploit a disparity in bandwidth consumption between an attacker and the targeted web resource. When the disparity in cost is magnified across many requests, the resulting volume of traffic can disrupt network infrastructure. By sending small queries that result in large responses, the malicious user is able to get more from less. By multiplying this magnification by having each bot in a botnet make similar requests, the attacker is both obfuscated from detection and reaping the benefits of greatly increased attack traffic [45].

f- Black Hole Attack: become part of many routes, drop all packets.

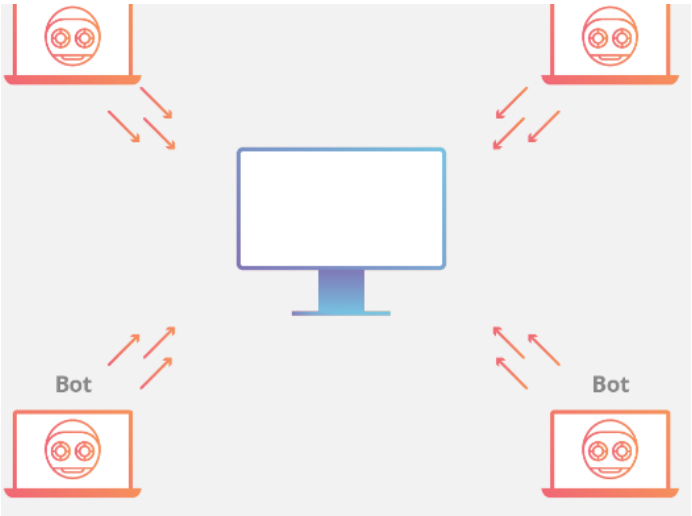

Figure 3.4: Bootnet attack

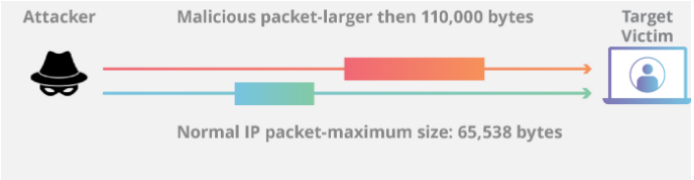

Figure 3.5: DNS amplification

3.4 Common security techniques

they have many technique of security like encryption , cryptography ..

- Encryption: that mechanism provides security against passive attacks like eavesdropping. Sensor network mostly run in public or wild area over inherently unconfident wireless channels. It is therefore insignificant for a device to eavesdrop or even add messages into the network. The traditional key to this problem has

been two espouse techniques such as method authentication codes, symmetric key encryption schemes and public key cryptography [47].

- Cryptography: cryptography constitutes the main theoretical concept, along with key infrastructures, underlying approaches that ensure integrity and confidentiality. Elliptic curve cryptography (ECC) has been recognized as a viable approach for WSN.ECC is a promising alternative to RSA-based algorithms, as the typical size of ECC keys is much shorter for the same level for security [48].
- Artificial Intelligence: used in this technique machine learning.
- Security protocols: used the protocol security (leap , spins , snips)
- IDS(Intrusion Detection System) : this solution for simulated for next chapter , and explained in next section .
- And we have more other solution .

3.4.1 Security protocols

- LEAP protocol: LEAP is also a very popular security solution in WSN and it was proposed by Zhu et al in 2004. Localized encryption and authentication protocol (LEAP) is a key management protocol used to provide security and support to sensor networks. It uses micro TESLA (timed, efficient, streaming, loss-tolerant, authentication) to provide base station broadcast authentication and one-way-hash-key to authenticate source packet [49].
- SPINS: It stands for Security protocols for sensor networks. It is an economical security scheme with low overhead. SPINS consist of two components called micro TESLA and SNEP. Micro TESLA's purpose is to provide authenticated broadcast, since this communication mode is standard in WSN. The problem solved by it is that the authenticated broadcast requires a costly asymmetric mechanism that sensor nodes cannot afford usually. The protocol emulates asymmetry by sending encrypted messages and key information independently [50] .

3.5 Securing CoAP protocol

3.5.1 DTLS(Datagram Transport Layer Security)

Security for CoAP protocol requires the existence of an additional encrypted protocol as a cover, similar to traditional XML-data representations and protocols (e.g., HTTP and Transport Layer Security (TLS)) [51].

The CoAP uses a UDP protocol and encryption is most commonly accomplished using Datagram Transport Layer Security (DTLS) and sometimes with IPSec. A binding of DTLS to CoAP is required to secure the messages. DTLS is applied in the transport layer and the fundamental AES/CCM provides confidentiality, integrity, authentication, and non-repudiation. DTLS is based on the Transport Layer Security (TLS) protocol and not the application layer. DTLS records are 8 bytes longer than in TLS. Once a handshake on DTLS is completed an additional [52].
13 bytes will be over headed overhead per datagram. The process of handshake comes in outgoing and incoming messages scenarios. In the incoming scenarios the protocol will verify through decryption and decompressing. For outgoing, the protocol applies encryption algorithm, add authentication code (MAC) and compress the message as shown in Fig 7. As the header of 6LoWPAN compression employs 10 bytes and CoAP header 4 bytes, therefore the focus is to compress the DTLS [51,52].

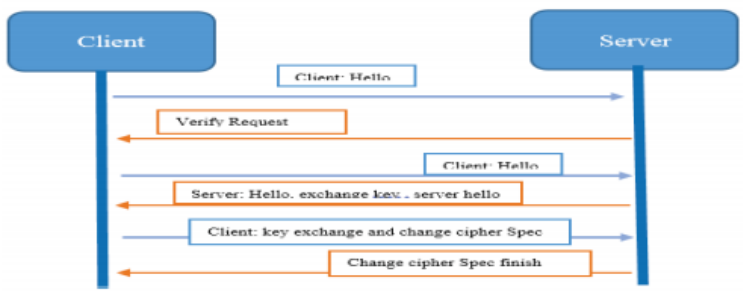

Figure 3.6: HandShake process [51

3.5.1.1 Security modes defined for CoAP

- NoSec: this alternative assumes that security is not provided in this mode or in the CoAP transmitted message.
- PreshardKey: this mode is enabled by sensing devices preprogramed with symmetric cryptographic keys. This mode is suitable for applications that support devices that are unable to employ the public key cryptography. Also, applications can use one key per device or one key for a group of devices.
- RawPublicKey: the mandatory mode for devices that require authentication based on public key. The devices are programmed with pre-provisioned list of keys so that devices can initiate a DTLS session without certificate.
- Certificates: supports authentication based on public key and application that participate in certification chain. The assumption of this mode is that security infrastructure is available. Devices that include asymmetric key and have unknown X.509 certificates can be validated using the certificate mode and provisioning

trusted root keys [53].

3.5.2 IDS(Intrusion Detection System)

3.5.2.1 Definition

An Intrusion Detection System (IDS) is a tool or mechanism to detect attacks against a system or a network by analyzing the activity in the network or in the system it-self. Once an attack is detected an IDS may log information about it and/or report an alarm. Broadly speaking, the detection mechanisms in an IDS are either signature based or anomaly based [54].

3.5.2.2 Types of IDS

- Signature : based detections match the current behavior of the network against predefined attack patterns. Signatures are pre-configured and stored on the device and each signature matches a certain attack. In general signature based techniques are simpler to use. They need, however, a signature of each attack and must also store it. This requires specific knowledge of each attack and storage costs grow with the number of attacks. This approach is more static and cannot detect new attacks unless their signature is manually added into the IDS [54].
- Anomaly : based detection tries to detect anomalies in the system by determining the ordinary behavior and using it as baseline. Any deviations from that baseline is considered an anomaly. On one hand, anomaly based systems have the ability to detect almost any attack and adapt to new environments, but on the other hand these techniques have rather high false positive rates (to raise an alarm when there is no attack) as deviations from the baseline might be ordinary. Also, they have comparatively high false negative rates (no alarm when there is an attack) as attacks might only show a small deviation that is considered within the norm [54].

3.5.2.3 IDS in IoT

Intrusion Detection Systems (IDS) are devices or software applications that monitor network or system activities for malicious activities or policy violations and send reports to a management station.
On constrained nodes in 6LoWPAN networks. Based on the novel requirements of the IoT, we propose SVELTE , a lightweight yet effective intrusion detection system for the IoT. We also compliment SVELTE with a distributed mini-firewall in order to filter malicious traffic before it reaches the resource constrained nodes.
We design SVELTE for a 6LoWPAN network that uses message security technologies,

such as IPsec and DTLS to provide end-to-end message security [55].
The placement of an IDS is an important decision that reflects the design of an IDS and the detection approaches. Keeping in view the resource constrained nature of the devices and the IoT setup , we use a hybrid, centralized and distributed, approach and place IDS modules both in the 6BR and in constrained nodes.
SVELTE intrusion detection system: This a lightweight and effective IDS First designed specially for IOT . It have an integrated firewall, it consists of6LoWPAN Mapper that Get the information about network and construct it using RPL (IPv6 routing Protocol).In detects intrusion by analyzing the mapped data [56].

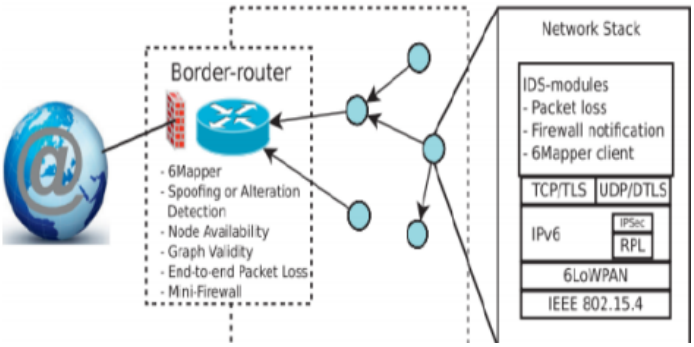

Figure 3.7: IDS in IoT [56]

- Network-based Intrusion Detection System (NIDS)
  This is a category of intrusion detection system where every transmitted packet through the network isproperly analysed. The NIDS can detect harmful packets that are calculated to be ignoredby a firewall's filtering guidelines.A network based IDS sensor has two interfaces and one of them is manageable. The IDS management console communicates with the sensor through the management interface. The other interface of the IDS is in promiscuous (listening) mode and is not accessible over the network hence not manageable. The observing interface is linked to the network section that is being observed. The sensor observes each packet that crosses the network section. Network based sensors apply known attack signatures to every frame. If it detects a match against any signature, it alerts the management console [50].

- Hybrid Intrusion Detection method
  This method of intrusion detection in internet of things was proposed by Sedjelmaci et al in [57] based on the use of Game Theory. This method mixed the usage of signature and anomaly ways for IoT intrusion detection. It achieve this by creating the game model of intruder and normal user.

3.6 Related works for Securing CoAP

Many researchers are currently focusing on the IoT security issue; therefore, a lot of security solutions have been provided

a- As proposed by Raza et al, they proposed mechanisms to exploit the compression capabilities of 6LoWPAN to compress the DTLS (Datagram Transport Layer Security) headers and messages. Since DTLS was designed for the internet not for constrained IoT devices, it is a heavyweight protocol with too long headers to fit in a single IEEE 802.15.4 MTU (Maximum Transmission Unit).
Therefore, 6LoWPAN is used in IoT to compress the long IP layer headers. The authors also defined the Record header, Handshake header, Client Hello message, and the Server Hello message and their compression techniques.
One of the results showed that for the DTLS record header the number of additional bits can be reduced by 62 However, their study faces a weakness that it did not go further to ensure that compression would not affect the security.

b- As proposed by Kothmayr [53] and proposed by Kothmayr et al. , a security solution based on RSA, the most widely used public cryptography algorithm. Their goal was to achieve high interoperability and low overhead. However, because of the overhead of DTLS handshake process, RSA consumes a large amount of energy which considers a weakness feature for IoT devices.

c- Therefore, as proposed by Raza and al. , another solution by using DTLS compression as well. The authors proposed Lithe – an integration of DTLS and CoAP for the IoT. Lithe consists of four components: DTLS, CoAP, DTLS header compression (using 6LowPAN), and a CoAP-DTLS integration module which was developed to allow the application to access CoAP automatically The evaluation results show significant gains in the processing time, network response time and energy consumption by reducing the packet size. With this work, they were able to avoid fragmentation or decrease the number of fragments by using compression when the payload was slightly above the fragmentation threshold.

d- As proposed by Ukil et al, the authors proposed a lightweight security approach using AES (Advanced Encryption Standard) 128-bit symmetric key algorithm. They came up with an Auth-Lite approach which enables the authentication mechanism, and by modifying the CoAP header, they came up with CoAPLite which enables lightweight security for CoAP. But this approach faces a weakness that it can only be used in vehicle tracking systems, and it depends on the application, so for other applications it may or may not be efficient.

3.7 Conclusion

In terms of security, CoAP can be seen as not yet mature. This is because there are still many security challenges with that protocol. The biggest challenge is to keep the high performance while maintaining the security standards and providing protection.

Chapter 4

Conception And Implementation

4.1 Introduction

The Internet things has many problems but the most sever one is how to save energy while providing QoS-effective communications with acceptable security services. DoS attacks target to subvert all of the aforementioned aspects, and that is why they are considered among the most dangerous attacks in the IoT.
This chapter consists of two parts: the first one presents the assumed network model, as well as the threat model. The second part of the chapter provides detailed presentation of the realization and evaluation contexts of our solution.

4.2 Conception

In this parts we start by the design of the network, threat, solution

4.2.1 Model Dos attaque (C/S)

Denial of service or distributed denial service of service attack is an attempt to make a machine or network resource unavailable to its intended users (the attack has been addressed with more details in chapter 3) .
The figure bellow contains 4 client and 1 server only 1 client is hacker, now it is very important to detect this attacker used the solution (Detection Intrution System). The countermeasure has been also detailed in the previous chapter.

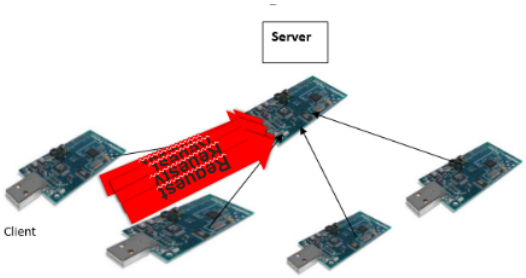

Figure 4.1: Dos attack Client and Server

4.2.2 Model Of Solution

The role of solution is to detect and blaklist malicious CoAP clients that attempt to exercise DoS attack. So, when any client send more request in 1 clockseconds() greater than a given threshold, that CoAP client will be henceforth considered as attacker. When the number of time the same malicious CoAP client had been detected as DoS attacker reaches a well-determined isolation threshold, we should blacklist that client node. The figure bellow models all of this discussion.

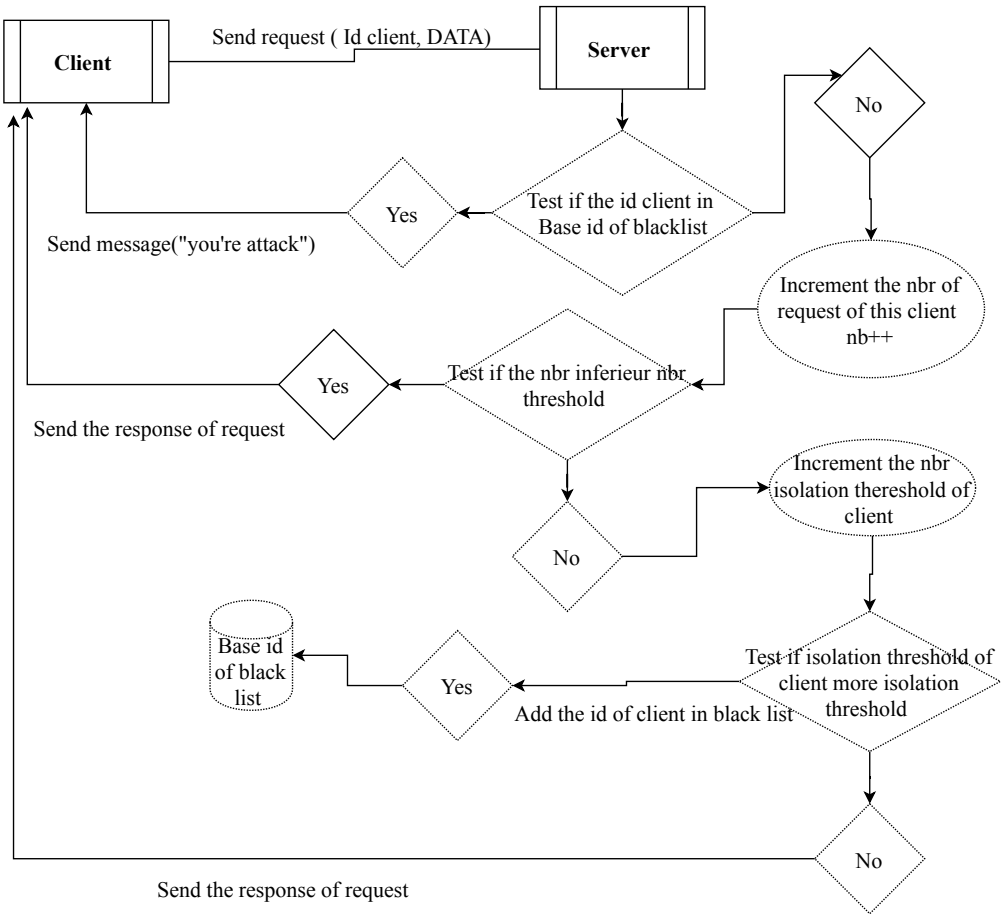

Figure 4.2: Archetecture of solution Detection attack

4.2.3 Pseudo code of the solution

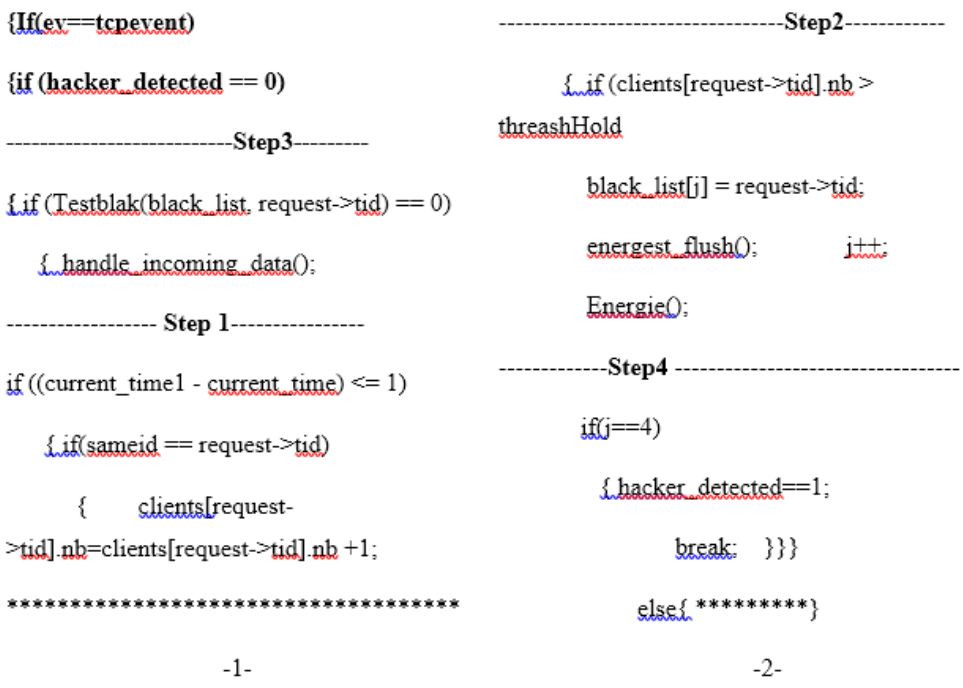

Figure 4.3: Pseudo code of the solution

4.2.4 Brief Explanation of the solution

This algorithm contains the following main steps:

- The first step is when a request is received, here, we firstly compute the number of request arriving per second for each CoAP server
- The second step is when the number of request exceeds the threshold (the max number of request that a CoAP client sends). Here we should add the concerned Client to the black list.
- The third step is when the black list contains the detected malicious CoAP clients.
- The final step is to test if the detected malicious CoAP client ha been blacklisted for more than once. Thus, the CoAP server should isolate it and never reply to its future requests.

4.3 Implementation

4.3.1 Introduction

This part contains the platform used for this simulation, and discuss the evaluation results for this simulation.

4.3.2 Contiki OS

Contiki[58] is an open source OS for WSN sensor nodes. It is a lightweight and portable OS written in C language and it is build around an event-driven kernel. This OS provides preemptive multitasking that can be used at the individual process level. A typical Contiki configuration consumes 2 kilobytes of RAM and 40 kilobytes of ROM. A full Contiki installation includes features like: multitasking kernel, preemptive multithreading, proto-threads, TCP/IP networking, IPv6, a Graphical User Interface, a web browser, a personal web server, a simple telnet client, a screensaver, and virtual network computing.
Contiki contains two communications : uIP and Rime

- uIP is a small stack of TCP / IP RFC-COMFORME that allows Contiki to communicate on the internet .
- Rime is a lightweight communication stack designed for low power radios. It provides a wide range of primitive communications

4.3.2.1 Contiki OS Architecture

The Contiki OS is based on a modular architecture. At the kernel level it follows the event driven model, but it provides optional threading facilities to individual processes. This kernel comprises of a lightweight event scheduler that dispatches events to running processes. Process execution is triggered by events dispatched by the kernel to the processes or by a polling mechanism. This polling mechanism is used to avoid race conditions. Any scheduled event will run to completion, however, event handlers can use internal mechanisms for preemption. Asynchronous events and synchronous events are supported by Contiki OS. Synchronous events are dispatched immediately to the target process that causes it to be scheduled. On the other hand asynchronous events are more like deferred procedure calls that are en-queued and dispatched later to the target process. The polling mechanism can be seen as high-priority events that are scheduled in between each asynchronous event. When a poll is scheduled, all processes that implement a poll handler are called in order of their priority. All OS facilities: senor data handling,

communication, device drivers, etc. are provided in the form of services. Each service has its interface and implementation. Applications using a particular service need to know the service interface and an application is not concerned about the implementation of a service [59].

4.3.2.2 Cooja Simulator

COOJA is a network simulator which permits the emulation of real hardware platforms. COOJA is the application of Contiki OS concentrating on network behavior. COOJA is capable of simulating wireless sensor network without any particular mote. Cooja supported following set of standards; TR 1100, TI CC2420, Contiki-RPL, IEEE 802.15.4, uIPv6 stack and uIPv4 stack. There are four propagation models in the COOJA simulator which must be selected before starting a new simulation [60]. The first model is constant loss Unit Disk Graph Medium (UDGM) and it take the ideal transmission range disk in which motes inside the transmission disk receive data packets and motes outside the transmission disk do not get any packet. The second model is distance loss UDGM is the extension of constant loss UDGM and it also consider the radio interferences. Packets are transmitted with "success ratio TX" probability and packets are received with probability of "success ratio RX". The third model is Directed Graph Radio Medium (DGRM) and it states the propagation delays for the radio links. Last path loss model is multipath Ray-tracer Medium (MRM) and it uses the ray tracing methods such as Friis formula to calculate the receiver power. MRM is also capable of computing the diffractions, reflections and refractions along the radio links [61].
In a simulation we have several windows (you can see the figure 4.4)

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

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

3. The Mote Output window: on the right side of the screen, shows us all serial port impressions of all nodes.

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

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

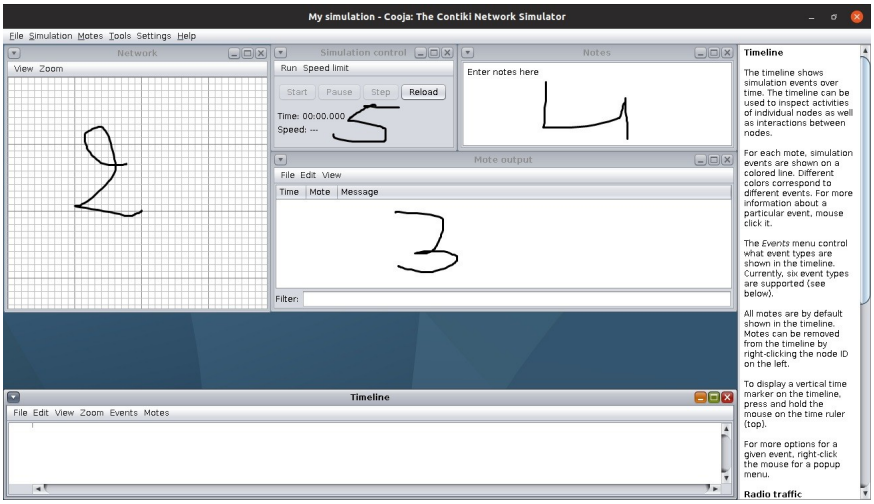

Figure 4.4: Cooja Simulator

4.3.3 Simulation context

They have two but for this simulation , firstly etude the consomation energy and amliorer the performance . secondly detecter the haker between many client for securite the network WSN from dos or ddos attack .
I can get this Result of simulation because ability for the tools,In figure 4.5 bellow we show the tools of simulation

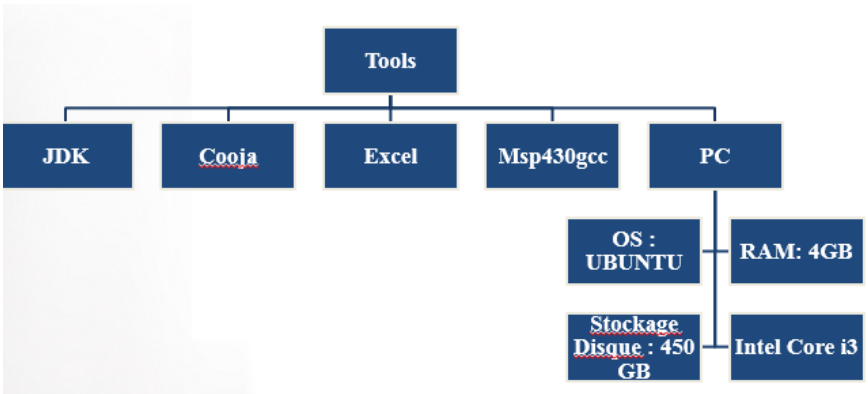

Figure 4.5: Tools of Simulation

In table bellow we show the parameters of simulation

Parameter	Values
Simulator	Cooja
Size of the simulation area	350m X 350m
Number of motes	10
Time of simulation	4 min(240 second)
Platform	Sky
Type of mote	3 motes coap servers 7 motes coap clients

Table 4.1: Parameters of Simulation

4.3.3.1 Senario of Network

I used in sumulation 3 Server and 7 client melange (Clients and attacks) , In figure 4.6 bellow we show Senario of network

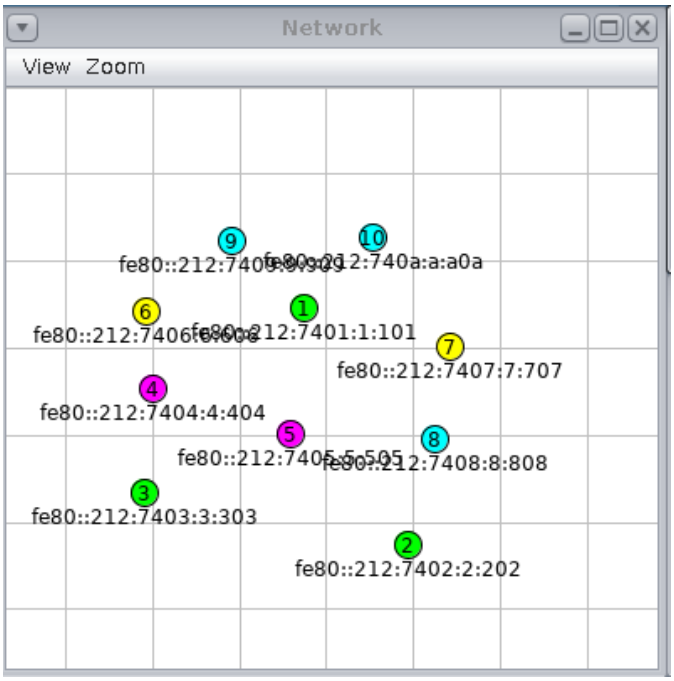

Figure 4.6: Senario of Network

4.3.4 Evaluation the result

a- Energy consumption

Formula to calculate energy consumption in cooja :

Energy (milliJule)= [[(tempsLPM*0.545 milliampère) + (tempsCPU*1.8 mA)+ (tempsTX*17.7 mA) +(tempsLISTEN * 20 mA)] * 3 volt] / 32768.

Times are expressed in ticks. 32768 is the number of ticks generated in a second, in contiki 2.7.

Consumed energy amounts with the solution is less then without solution integration (see the figure 4.7). This is because when we detect attackers we do not need to send them responses. This makes the solution more energy-effective.

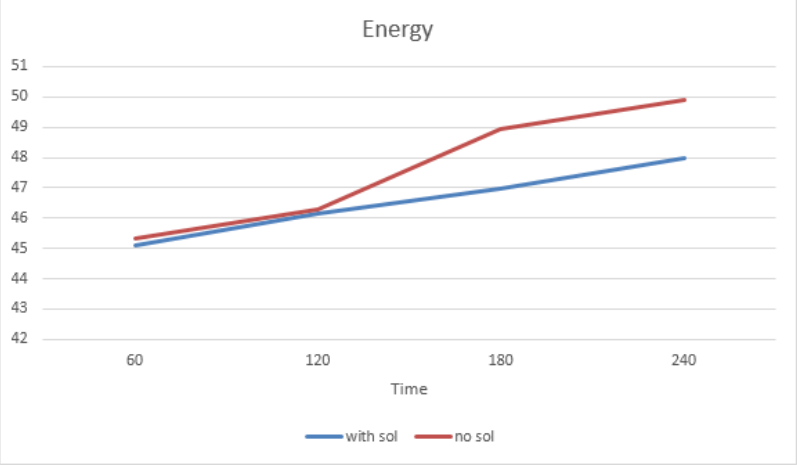

Figure 4.7: Energy with and Without Solution

b- Detection accuracy:

This parameter represents the percentage of full detections. Firstly, with different values for isolation threshold. Figure 4.8 shows that with increased values for the isolation threshold the detection accuracy decreases which is entirely logical.

The second case is with increasing number of attackers in the network. Figure 4.9 shows that when we have more attackers, the detection is reduced; this is due to the possible network collisions.

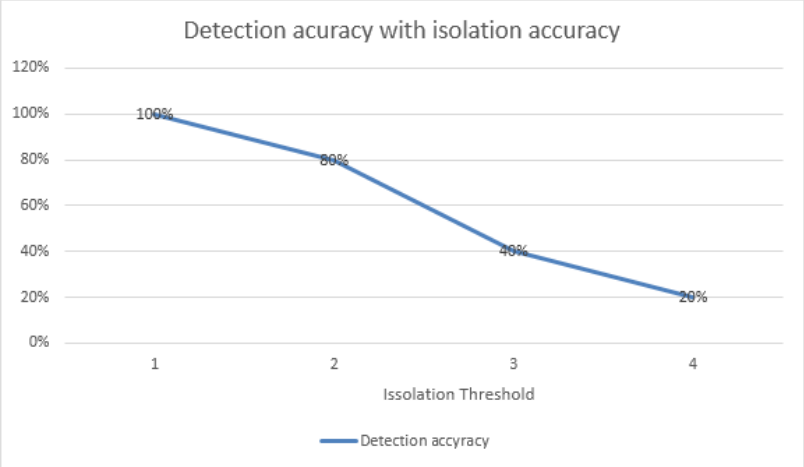

Figure 4.8: Energy with and Without Solution

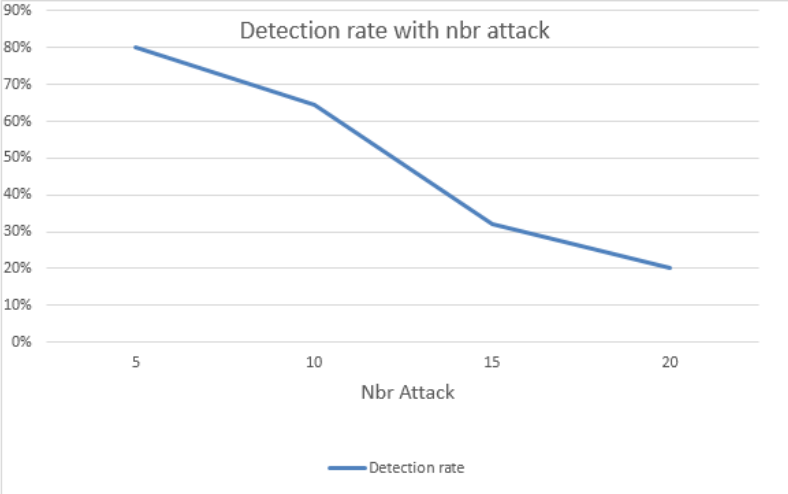

Figure 4.9: Energy with and Without Solution

4.4 Conclusion

In this chapter, we have presented the essence of our work that is about a security mechanism for CoAP communications in the IoT,explained the sumulator cooja . The evaluation results confirm the effectiveness of the solution.

General Conclusion

Throughout this work we have presented the IoT then, we have addressed Application layer security issues in IoT especially those related to CoAP protocol. Thereafter, we have presented our solution that consists in DoS-resilient security solution for CoAP. The obtained simulation results confirm its efficiency.
we see how COOJA network simulator enables the emulation different kinds of motes and how the routing matrices are computed.
As future work, the solution can be enhanced by the consideration of many other forms of DoS attacks, such as packet amplification combined with identity spoofing scenarios.

Bibliography

[1] https://fr.wikipedia.org/wiki/Internet_des_objets last visited 24/06/2019

[2] LABED AYOUB , These Master , Fog computing task scheduling Based on simulated annealing algorithm, University khider Biskra

[3] Zainab H. Ali, Hesham A. Ali, Mahmoud M. Badawy International Journal of Computer Applications (0975 – 8887) Volume 128 – No.1, October 2015

[4] Sarfraz Alam, Mohammad M. R. Chowdhury, Josef Noll, 2010 SenaaS:An Event-driven Sensor Virtualization Approach for Internet of Things Cloud, Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on, 1-6

[5] Marco Zennaro, PhD The Abdus Salam International Centre for Theoretical Physics

[6] https://www.peerbits.com/blog/difference-between-m2m-and-iot.html last visited 24/06/2019

[7] https://pdfs.semanticscholar.org/8c37/0640e788b8bb3e2b9803c2ae752b661ff8df.pdf last visited 24/06/2019

[8] https://www.avl.com/nl/vehicle-system-simulation

[9] http://www.6gsummit.com/

[10]https://www.ticketmaster.com/h/mobile.html

[11] A.M. Vilamovska, E. Hattziandreu, R. Schindler, C.Van Oranje, H. De Vries, J. Krapelse, RFID Application in Healthcare – Scoping and Identifying Areas for RFID Deployment in Healthcare Delivery, RAND Europe, February 2009.

[12] https://www.researchgate.net/figure/IoT-based-Smart-Environments_fig1_309242296

[13] https://www.i-scoop.eu/internet-of-things

[14] Sudip Misra, P. Venkata Krishna, Harshit Agarwal, Anshima Gupta, Mohammed S.Obaidat, 2012 An Adaptive Learning Approach for Fault-Tolerant Routing in Internet of Things. IEEE Wireless Communications and Networking Conference: PHY and Fundamentals, 815 – 819.

[15] D.Giusto, A.lera, G.Morabito, L.Atzori (Eds.), 2010 Objects Communication Behavior on Multihomed Hybrid Ad Hoc Networks, Springer, 3-11

[16] https://www.tutorialspoint.com/internet_of_things/internet_of_things_overview.htmllast visited 06/24/2019

[17] https://www.slideshare.net/akmalamir/internet-of-things-iot-65345214

[18] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," IETF RFC 7252, June 2014. [Online]. Available: http://tools.ietf.org/html/rfc7252

[19] Z. Shelby, K. Hartke, C. Bormann, Constrained Application Protocol(CoAP), University Bremen TZI. [June 2014]

[20] https://tools.ietf.org/id/draft-ietf-core-coap-tcp-tls-08.html

[21] https://www.engineersgarage.com/Articles/CoAP-Protocol-Features-REST--Architecture

[22] Alabbas Alhaj Ali , " Constrained Application Protocol (CoAP) for the IoT " , Frankfurt University of Applied Sciences. [May 2018].

[23] https://www.service-architecture.com/articles/web-services/representational_state_transfer_rest.html

[24] Valérie Issarny, Mauro Caporuscio, Nikolaos Georgantas, Workshop on the Future of Software Engineering : FOSE 2007, 2007, Minneapolis, United States. pp.244-258, 2007

[25] Prof. Raj Jain. Constrained Application Protocol for Internet of Things Xi Chen, chen857 (at) wustl.edu

[26] Z. Shelby, K. Hartke, C. Bormann, Constrained Application Protocol (CoAP), Universitaet Bremen TZI. [June 2014]

[27] Bormann, Carsten, Angelo P. Castellani, and Zach Shelby. 2012. "CoAP: An application protocol for billions of tiny Internet nodes." IEEE Internet Computing, vol. 1, no. 2, pp. 62–67, Mar/Apr. Accessed 2018-06-21.

[28] Bormann, C., S. Lemay, H. Tschofenig, K. Hartke, B. Silverajan, and B. Raymor, eds. 2018. "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets." RFC 8323.

[29] Z. Shelby, K. Hartke, C. Bormann, Constrained Application Protocol (CoAP), Universitaet Bremen TZI. [June 2014]

[30] https://devopedia.org/constrained-application-protocol

[31] https://www.researchgate.net/publication/305522152_Experimental_Evaluation_of_Unicast_and_Multicast_CoAP_Group_Communication

[32] VIPRET Julien , Rapport d'IRL Utilisation du protocole COAP pour la d´ecouverte de ressources dans les r´eseaux de capteurs , Laboratoire LIG - Grenoble INP – Ensimag, [mai 2012]

[33] Klaus Hartke, Hannes Tschofenig, A DTLS Profile for the Internet of Things drafthartke-dice-profile-00, [2013-11- 04].

[34] G.Tolle, Arch Rock Corporation. Embedded Binary HTTP (EBHTTP) draft-tollecore-ebhttp-00.[2010-03-23].

[35] Shetu Rani Guha , Constrained Application Protocol for Internet of Things, Computer Science and Engineering University of Khulna Khulna, Bangladesh

[36] T. A. Alghamdi, A. Lasebae, and M. Aiash, Security analysis of the constrained application protocol in the Internet of Things, in proceedings of 2nd IEEE International Conference on Future Generation Communication Technology (FGCT), UK, Nov 12-14, 2013

[37] Reem Abdul Rahman, Babar Shah, Security analysis of IoT protocols: A focus in

CoAP, College of Technological Innovation Zayed University,[2016]

[38] https://www.slideshare.net/HamdamboyUrunov/the-constrained-application-protocol-coap

[39] Istabraq M. Al-Joboury, Emad H. Al-Hemiary, Internet of Things (IoT): Readme, The 1 stInternational Conference on Information Technology (ICoIT'17) Lebanese French University - Erbil, Kurdistan Region – Iraq, 10thof April, 2017.

[40] E. Rescorla, E. Rescorlai, N. Modadugu, "Datagram Transport Layer Security Version 1.2,"RFC Standard 6347, Jan. 2012.

[41] M. BOUDRAA Hakim Mlle. BOU KERROU Lilia ´epouse GHAZLI , Master Memory , t´echniques d´etection Botnets In IoT, University bejaya

[42] A. Géron, « WIFI PROFESSIONNEL, La norme 802.11, le déploiement, la sécurité », 3ème édition, DUNOD, 2009.

[43] D. Miorandi, S. Sicari, F. Depellegrini, and I. Chlantac. Internet des objets :vision ,applications et defis de la recherche. Reseaux adhoc , 10, (7), 1497 - 1516, (2012).

[44] L. DaXu and S. He, W.and Li. Internet des objets dans les industries : un sondage. IEEE Transactions sur l'informatique indistruelle, 10, (4), 2233 - 2243, (2014).

[45] CM. Medaglia and A. Serbanati. Un apercu des questions de confidentialit´e et de securit´e dans l'internet des objets (pp.389-395). Spriger ,new york ,NY, (2010).

[46] https://www.imperva.com/learn/application-security/ddos-attacks/

[47] C.K.Marigowda,ManjunathShingadi,"Security Vulnerability Issues In Wireless Sensor Networks:AShortSurvey",International Journal Of Advance Research In Computer And communication Engineering .Vol.2,Issue 7,July 2013.

[48] Eric Platon and Yuichi Sei, "Security Software Engineering in Wireless Sensor Networks", Progress in Informatics, NO.5, PP.49- 64,(2008).

[49] DelanAlsoufi , KhaledElleithy , Tariq Abuzaghleh and Ahmad Nassar, "Security in Wireless Sensor Networks- Improving the Leap Protocol", International Journal of Computer Science and Engineering Survey(IJCSES) Vol.3,No.3, June 2012.

[50] Eric Platon and Yuichi Sei, "Security Software Engineering in Wireless Sensor Networks", Progress in Informatics, NO.5, PP.49- 64,(2008).

[51] J. Granjal, E. Monteiro and J. Silva, Security for the Internet of Things: A Survey of Existing Protocols and Open Research issues, IEEE Communications Surveys Tutorials, Vol. 17, no.3, pp. 1294-1312, 2015

[52] T. A. Alghamdi, A. Lasebae, and M. Aiash, Security analysis of the constrained application protocol in the Internet of Things, in proceedings of 2nd IEEE International Conference on Future Generatio n Communication Technology (FGCT), UK, Nov 12-14, 2013

[53] T. Kothmayr, W. Hu, C. Schmitt, M. Bruenig, G. Carle, Securing the internet of things with DTLS, in: Proceedings of the 9th ACMConference on Embedded Networked

Sensor Systems, ACM, 2011, pp. 345–346.

[54] Shahid Raza , Linus Wallgren , Thiemo Voigt , SICS Swedish ICT, Stockholm, SwedenDepartment of Information Technology, Uppsala University, Sweden

[55] S hahid Raza , Linus Wallgren , Thiemo Voigt , SICS Swedish ICT, Stockholm, SwedenDepartment of Information Technology, Uppsala University, Sweden

[56] https://www.sans.org/detection/intrusion-detection-systemsdefinition-challenges-343

[57] https://www.hindawi.com/journals/misy/2017/1750637/ accessed on 13th/05/2017

[58] Dunkels, A.; Gronvall, B.; Voigt, T. Contiki a Lightweight and Flexible Operating System for Tiny Networked Sensors. In Proceedings of the 9th Annual IEEE International Conference on Local Computer Networks, Washington, DC, USA, October 2004; pp. 455-462 .

[59] Contiki Documentation; Available online: http://www.sics.se/~adam/contiki/docs/

[60] M. Stehlik. Comparison of Simulators for Wireless Sensor Networks. PhD thesis, Masaryk University, 2011.

[61] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, And T. Voigt.Cross-level sensor network simulation with cooja. In LCN, 2006,pages 641 – 648, nov 2006 .