**Arab American University – Jenin**

**Faculty of Graduate Studies**

# Performance Improvement in Lightweight Security Protocol for Internet of Things Networks

By

**Haytham Abdlkareem Qushtom**

Supervisor

**Dr. Khalid Rabaya'h**

**This thesis was submitted in partial fulfillment of the requirements for the Master`s degree in**

**Computer Sciences**

**May / 2017**

# Performance Improvement in Lightweight Security Protocol for Internet of Things Networks

By

**Haytham Abdlkareem Qushtom**

This thesis was defended successfully on 15/05/2017 and approved by:

| Committee members | Signature |
|---|---|
| 1. Dr. Khalid Rabaya'h | ……………………. |
| 2. Dr. Adwan Yasin | …………………….. |
| 3. Dr. Wasel Ghanem | …………………….. |

## Dedication

Praise and thanks to God for the all the blessing and for the product of knowledge that, for this thesis would not be possible.

And to Prophet Muhammad, peace be upon Him, the eternal light and blessing for mankind.

To my Dad, Allah bless his soul, who paved the path before me upon where I stand now, this is for you, for every second that you fight and struggle for us. This thesis is the result, of our promise to you, to follow your steps to be a good man.

To my mom, the lady of dignity, honesty and generosity, your embrace had been my shelter, I hold on to every word that you taught us and we learn from you. For the sleepless nights that you had been through when caring for me when I was little, this is my payment for everything that you had given to me.

Dedicated to my siblings and to all the people who had been with me through this thesis. An immeasurable appreciation and deepest gratitude for the help and support.

To my little sister who brings happiness and color to our life, to whom I get my strength and power. To whom where I get my patience because of her kind heart and soul.

To all my friends who never stop supporting me, wishing God will always bless them and give them the desires of their heart.

## Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Khalid Rabaya, who has always been providing support, encouragement, and guidance throughout the completion of this thesis.

Also, I extend my sincere thanks and appreciation to the staff of the faculty of engineering and information technology at AAUJ- master program.

Finally, I would like to thank my friends for supporting and helping me throughout this work.

# Abstract

Through various internet technology developments, the world is moving towards Internet of Things (IoT) as a prevailing application of the future of Internet. IoT is a descriptive term of a vision that everything should be connected to the internet.

IoT is considered as one of the hottest research topics in computer networks in particular, and in Information Technology (IT) in general.  The concept in its very basic meaning opens up wide range of opportunities for new services and new innovations. It also covers vast range of application from personal, to organizational, to industrial, agricultural, to national or even international domains.

IoT has grown very rapidly during the last few years, since it is launched in 2009. This rapid growth of the field, and the mounting interest of people in its applications, is leaving large number of issues unresolved, especially at research level. One of the most critical issue has to do with delivery of secured and classified data.

Many scholars are reporting vulnerabilities in IoT networks security as most protocols were inherited from the traditional low secured Internet protocols. This thesis is set to tackle the issue of securing data delivery in our case at the data link layer level. One of the basic principles on which IoT was built on is constrained resources together with complex structure of hardware, sensors, applications, and communication and networking protocols.

In this research work, the consideration is given to enhancing the quality of service (QoS) provisioning at the MAC sub-layer level. The choice of the MAC sub-layer stems from the

fact that it is responsible for the management of the access to the wireless channel. And the channel is counted as the main element in the whole network which controls the system performance. Upon analyzing the quality of services provisioning of the traditional and the most dominant MAC protocol used by IoT systems that is the CSMA/CA. We proposed several enhancement ideas to boost the delivery of the secured data over IoT. The proposed solution is divided between adaptation layer and data link layer, where the data link layer plays the main role for the network performance.

The performance of proposed solution or protocol is analyzed using the best known simulation environment used by IoT, the Contiki OS, together with the Cooja simulator, which was specifically designed for IoT systems. Our proposed solution specifically targets the improvement of the quality of service (QoS) that supports the requirements of any application and uses requirements of the MAC layer. As part of the simulation environment, IPsec protocol is used to provide secure traffic.

Our proposed solution for providing secured traffic is denoted as Secure Traffic Priority Differentiation (STPD), which can be readily used by IoT networks that are facing challenge in the provision of quality of service in secure traffic. Our proposed STPD algorithm is a modified version of the MAC protocol with QoS that supports a heterogeneous IoT network. The STPD is an advanced scheme to access the channel and uses a contention-base mechanism that favors high priority traffic.

STPD outperforms CSMA/CA in all simulated scenarios, mainly when the number of intermediate nodes is high. Which STPD achieved improvements in transmission channel utilization with average around 25%. In regards for packets latency, STPD exhibited its

superiority with average improvements of 50% than CSMA/CA, particularly when the system deals with secured data. The third parameter, which was tested is the packet delivery ratio (PDR). Yet again, STPD showed improved PDR with an average percentage of 20% in contrast with CSMA/CA.

Theses significant improvements, will definitely enhance the overall performance of the entire IoT systems, starting from source down to destination.

**Keywords**: Wireless sensor networks, Internet of Things, quality of service, medium access control, secure traffic, IPsec, MAC Layer.

# Table of content

**Abstract** ........ III

**List of table** ........ IX

**List of Figures** ........ X

**Abbreviations** ........ XI

**Chapter 1** ........ 1

**Introduction** ........ 1

**1.1. Motivation** ........ 1

**1.2. Contributions** ........ 2

**1.3. Thesis Outline** ........ 4

**Chapter 2** ........ 5

**Basics of Wireless Sensor Networks and Internet of Things (IoT)** ........ 5

**2.1. Wireless Sensor Network (WSN)** ........ 5

**2.2. Application Examples** ........ 6

**2.2.1. Environmental applications** ........ 6

**2.2.2. Industrial applications** ........ 7

**2.2.3. Military applications** ........ 8

**2.2.4. Health care** ........ 8

**2.3. Application requirements and constraints** ........ 9

**2.4. Sensor networks design, architecture, and protocols** ........ 11

**2.4.1. WSN topologies** ........ 11

**2.5. Communication protocols architecture** ........ 13

**2.6. Wireless Sensor Networks challenges** ........ 15

**2.6.1. General design considerations** ........ 15

**2.6.2. Communication architecture challenges** ........ 15

**2.7. Internet of Things (IoT)** ........ 16

**2.8. Protocol stack in Internet of Things** ........ 17

**2.8.1. IEEE 802.15.4 protocol (Physical and Data-link layer)** ........ 18

**2.8.2. 6LoWPAN protocol (Adaption Layer)** ........ 18

**2.8.3. IPv6 (network layer)** ........ 19

**2.9. Security in Internet of Things** ........ 20

2.9.1. IPsec ........ 21

2.10. Overview of Quality of Service provisioning ........ 23

2.10.1. Factors affecting the quality of Service ........ 23

2.10.2. QoS techniques ........ 24

2.10.3. QoS provisioning at the MAC layer ........ 26

Chapter 3 ........ 27

Related Work on QoS Metrics and Protocols ........ 27

3.1. State of the art of QoS-aware MAC protocols for IoT ........ 31

Chapter 4 ........ 35

Efficient QoS Provisioning at the MAC Layer in IoT Network ........ 35

4.1. Motivation ........ 35

4.2. Design considerations and assumptions ........ 36

4.3. Basic Principles of Secure Traffic Priority Differentiation (STPD) ........ 39

4.3.1. packet prioritization ........ 40

4.4. STPD Operation ........ 43

Chapter 5 ........ 46

Performance Evaluation of STPD-MAC protocol ........ 46

5.1. Simulation environment ........ 46

5.1.1. Contiki Operating System ........ 46

5.1.2. Cooja simulator ........ 48

5.2. Simulation scenario and parameters ........ 49

5.2.1. MAC layer parameters (STPD-MAC protocol) ........ 50

5.3. Simulation results ........ 51

5.3.1. Transmission Channel Utilization ........ 52

5.3.2. Delivered Packets Latency ........ 54

5.3.3. Packet delivery ratio (PDR) ........ 57

5.4. Discussion ........ 60

5.4.1. Review of main results, ........ 60

5.4.2. Explanation of results ........ 61

5.4.3. Comparison with results of other researchers, ........ 65

5.4.4. Limitation to research ........ 67

5.4.5. Practical implications ........ 67

**5.4.6. Future research** ........ 68

**Chapter 6**........ 69

**Conclusion**........ 69

**Bibliography** ........ 71

**Arabic abstract**........ 75

# List of table

Table 5.1: Simulation settings ........................................................................................... 50
Table 5.2: Summary of Collisions and Transmission Attempts for HP and LP flow ....................... 64
Table 5.3: Summary of End-to-End delay attempts for STPD and CSMA/CA flow........................ 65
Table 5.4: Summary of End-to-End delay attempts for HP and LP flow......................................... 65
Table 5.5: Summary of average throughput attempts for STPD and CSMA/CA flow ..................... 66

# List of Figures

Figure 2.1: Architecture of a simple WSN ........ 6
Figure 2.2: Illustration of fire detection applications ........ 7
Figure 2.3:WSN rehabilitation monitoring application ........ 9
Figure 2.4: Star network ........ 12
Figure 2.5:Tree network ........ 12
Figure 2.6 : Mesh network ........ 13
Figure 2.7: The sensor networks protocol stack ........ 13
Figure 2.8: Architecture of a simple IoT ........ 17
Figure 2.9: 6LoWPAN protocol stack ........ 18
Figure 2.10: 6LoWPAN Architecture in IoT ........ 19
Figure 2.11: IPv6 Header ........ 20
Figure 2.12: IPv6 Address Format ........ 20
Figure 2.13: Transport Mode ........ 22
Figure 2.14: Tunnel Mode ........ 22
Figure 2.15: QoS interdependence ........ 23
Figure 2.16: interdependence of design factors ........ 26
Figure 3.1: classification of MAC protocols ........ 27
Figure 4.1: Basic idea for STPD ........ 36
Figure 4.2: network architecture used STPD ........ 37
Figure 4.3: STPD arbitration scheme ........ 39
Figure 4.4: state machine of STPD ........ 41
Figure 4.5: STPD flow diagram ........ 43
Figure 5.1: Cooja simulation graphical user interface ........ 49
Figure 5.2: scenario network topology for 36 intermediate nodes ........ 53
Figure 5.3: Comparative channel utilization ........ 54
Figure 5.4: Comparative channel utilization ........ 55
Figure 5.5: Comparative average latency ........ 57
Figure 5.6: Comparative average latency ........ 57
Figure 5.7: Comparative successful packet delivery ratio of STPD ........ 59
Figure 5.8: Comparative successful packet delivery ratio of STPD ........ 60
Figure 5.9: Ratio of collisions to total transmission ........ 61
Figure 5.10: Average number of attempts to create secure connection for IPsec protocol ........ 64
Figure 5.11: percentage enhancement of studied protocol compared with MAC protocol ........ 66

# Abbreviations

| | |
|---|---|
| **ACK** | **Ack**nowledgment |
| **CCA** | **C**lear **C**hannel **A**ssessment |
| **CSMA/CA** | **C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **A**voidance |
| **CTS** | **C**lear **To S**end |
| **CW** | **C**ontention **W**indow |
| **ECN** | **E**xplicit **C**ongestion **N**otification |
| **EDF** | **E**arliest **D**eadline **F**irst |
| **FCFS** | **F**irst **C**ome **F**irst **S**erved |
| **FDMA** | **F**requency **D**ivision **M**ultiple **A**ccess |
| **FIFO** | **F**irst **I**n **F**irst **O**ut |
| **HP** | **H**igh **P**riority |
| **IoT** | **I**nternet **o**f **T**hings |
| **IP** | **I**nternet **P**rotocol |

| | |
|---|---|
| **IPsec** | **I**nternet **P**rotocol **S**ecurity |
| **IPv6** | **I**nternet **P**rotocol **V**ersion **6** |
| **IPv4** | **I**nternet **P**rotocol **V**ersion **4** |
| **IEEE** | **I**nstitute **o**f **E**lectrical and **E**lectronics **E**ngineers |
| **LLC** | **L**ogical **L**ink **C**ontrol |
| **LP** | **L**ow **P**riority |
| **MAC** | **M**edia **A**ccess **C**ontrol |
| **OS** | **O**perating **s**ystem |
| **PDR** | **P**acket **D**elivery **R**atio |
| **QoS** | **Q**uality of **S**ervice |
| **RED** | **R**andom **E**arly **D**rop |
| **RFID** | **R**adio **F**requency **I**dentification |
| **RTS** | **R**equest **T**o **S**end |
| **RAM** | **R**andom **A**ccess **M**emory |
| **SP** | **S**trict **P**riority |
| **SSH** | **S**ecure **Sh**ell |
| **STPD** | **S**ecure **T**raffic **D**ifferentiation **P**riority |
| **TCP** | **T**ransport **C**ontrol **P**rotocol |

| | |
|---|---|
| **TLS** | **T**ransport **L**ayer **S**ecurity |
| **TDMA** | **T**ime **D**ivision **M**ultiple **A**ccess |
| **WFQ** | **W**eighted **F**air **Q**ueuing |
| **WLAN** | **W**ireless **L**ocal **A**rea **N**etworks |
| **WPAN** | **W**ireless **P**ersonal **A**rea **N**etworks |
| **WSN** | **W**ireless **S**ensor **N**etworks |
| **6LoWPAN** | IPv**6** over **L**ow Power **W**ireless **P**ersonal **A**rea **N**etworks |

# Chapter 1

## Introduction

### 1.1.Motivation

Next to the World Wide Web and mobile Internet technologies, a new technological trend is prevailing, it is the Internet of Things (IoT). These technologies continue to be smaller, faster, and more intelligent. IoT can be used to monitor and/or control valuable things for humans, society and industry. They allow people to perform any action at any time anywhere on the surface of the earth. The advancement in computing capabilities let smart objects interact with each other's, in a heterogeneous network using different hardware and software platforms. IoT supports wide range of applications. Some of these applications require the network to support different quality of service (QoS).  These QoS might be requested for variable rates, or variable traffics type (e.g., secure traffics).

IoT comprises mainly of distributed smart devices that have very constrained resources. These devices are in general made very small in size, and have very limited storage and memory size. They are in many cases installed in remote locations where they have to rely on batteries.  This makes them work in a very constrained environment. These requirements make these devices and the network that connects them unable to work effectively with standard TCP/IP protocol suits. Additionally, classical protocols are not made to meet the requirements of diverse types of applications, each with very specific requirements and network design.

Applications of IoT such as healthcare, military, and home automation requires different level of quality of services. This represents a challenge to classical TCP/IP protocol stack. The challenge even doubles when these networks operate in heterogeneous environments with constrained resources. To provide an array of applications the level of service they require, we need to design new protocol that is able to achieve this level of service requirements on the Data link layer level. As will be details in the literature review, substantial research efforts were paid to end to end issues such as end-to-end delay or throughputs. However, the performance of these solutions are not efficient in case of applications that have different QoS requirements. In this work, the focus will be to improve the performance of the IoT systems with classified data such as healthcare, military, air traffic and home automation applications. The work will focus on designing an efficient protocol that supports different level of QoS needed to support the service requirements of these applications.

## 1.2.Contributions

In this thesis, we are reporting on a proposed protocol that will improve the mechanism that controls quality of service provisioning for classified data applications. The solution requirements work at the MAC layer of the IoT network, where IPsec, the protocol that works at the IP layer level is used to provide the secured traffic.

Moreover, the research will analyze the specific QoS requirements for secured traffic in IoT applications. All ideas and algorithms thought of to enhance the management of the provision of different levels of quality of service, are collected and integrated in a modified version of the MAC protocol, called Secure Traffic Priority Differentiation (STPD). STPD adopts the strengths of the contention-based medium access technique which is used by the CSMA/CA protocol. Contention-based scheme is used by our proposed protocol to achieve higher channel utilization.

STPD divides traffic into two types secure and non-secure. It gives priority to secure traffic when contends with low priority traffic. The secured traffic is further divided into two distinct classes; high priority and low priority. Priority is distinguished by the device that is the source of the data. Other than that STPD gives high priority to connection setup traffic that is used to create an end-to-end secure connection.

One major goal of STPD is to achieve higher channel utilization. This goal is achieved through an efficient contention mechanism which relies on assigning longer random back-off times for low priority traffic than for high priority traffics. These mechanisms can easily be adapted for a larger number of traffic classes.

The performance of proposed solution or protocol is analyzed using the best known simulation environment used by IoT, the Contiki OS, together with the Cooja simulator, which was specifically designed for IoT systems. Our proposed solution specifically targets the improvement of the quality of service (QoS) that supports the requirements of any application and uses requirements of the MAC layer. As part of the simulation environment, IPsec protocol is used to provide secure traffic.

## 1.3.Thesis Outline

The present document is a detailed description of the work done in the course of testing a proposed solution that targets IoT application, and meant to enhance the quality of service provision offered for these applications.

After this chapter, which describes the research problem, and details that proposed solution, and in chapter 2, we provide some basic background theories on wireless sensor networks (WSN) and Internet of Things (IoT). In chapter 2 we presented a brief description of the WSN design, the underlying stack protocols based on which it works, and concludes the chapter with a list of some challenges that are still to be overcome. The second part of the chapter briefly introduces IoT, and how it differs from WSN. Part of the chapter is devoted to the 6LoWPAN protocol, and the security service that are needed by Internet of Things (IoT).

Chapter 3 details the concept of quality of service QoS, and the mechanisms used to realize it, especially in wireless networks. The chapter briefly came across some related work on QoS provisioning at the MAC layer for both WSN and IoT networks.

Chapter 4, reports on the details of proposed design to enhance the QoS provisioning for IoT networks, the STPD protocol. The chapter provides details on how STPD is operating.

Chapter 5 presented the performance evaluation results of STPD, as were reported by Cooja simulator. Simulation scenarios and parameters, and environment were all described in the chapter. Finally, chapter 6 concludes the research work by presenting a brief description of the main conclusion and gives a glue on some ideas for future work.

# Chapter 2

## Basics of Wireless Sensor Networks and Internet of Things (IoT)

In this chapter, we present the main concepts and theories based on which wireless sensor networks (WSN) and Internet of Things (IoT) do work. Firstly, the major fields of application will be described. Secondly, an overview of sensor networks design, architecture and protocols will be detailed. This part concludes with a brief description of the major challenges facing these technologies. The remaining part of the chapter describes the protocol stack used by IoT, and gives a brief description of the main security protocol used by these technologies.

## 2.1.Wireless Sensor Network (WSN)

Figure 2.1 shows a simple schematic diagram of Wireless Sensor Network. as is shown by the diagram it consists of nodes and sensors that collaborate with each other's to collect data. The collected data is transmitted to a sink node which in turn transmits the data to a gateway that forwards the data to the Internet. The WSN user can obtain the data from anywhere at any time using his / her smartphone or computer via the Internet. Wireless node can be used for sensing, processing, storing, and communicating collected data. These nodes in most of the time are equipped with batteries, or power harvesting facility. Wireless nodes are marketed at very low prices, since they are used on very large scale and used in very large numbers.

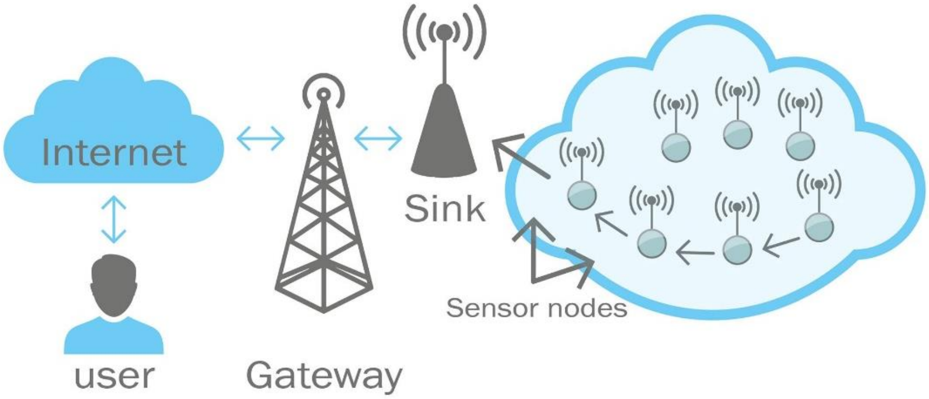


*Figure 2.1: Architecture of a simple WSN*

## 2.2.Application Examples

Wireless sensor networks have made it possible for so many application ranges from personal to environmental, to industrial application to name some. The availability of so many different kind of sensors with different sensing capability like temperature, humidity, light sensor, cameras, acoustic, infrared light sensors, accelerometers, gyroscopes, and medical sensors (heart rate and blood pressure sensors), makes it possible to have wide range of application for WSN. The following section, describes some of these common applications

### 2.2.1. Environmental applications

- Crisis management applications such as Fires, Earthquake, Storm, Tsunami, and Disease Alert are common applications of WSN. Sensor nodes can be simply used in complicated environment by using several things like flying drones over the crisis area or any other techniques used to collect data from the sensor. The sensor nodes identify their locations, collect environmental readings such as moisture and temperature, and transmit them to a base station which conducts the measurement and processing in a safe area. Figure.2.2 illustrate an example of some sensors trying to allocate the location of the fire and transmit the data into the base station.

- Agriculture applications can include harvesting, irrigation, overheating, and detection of plant diseases. Sensor nodes will collect barometric pressure, light, humidity, temperature, carbon dioxide, gasses, soil moisture data, which can be used to set the triggers of real-time alarm systems in the field.

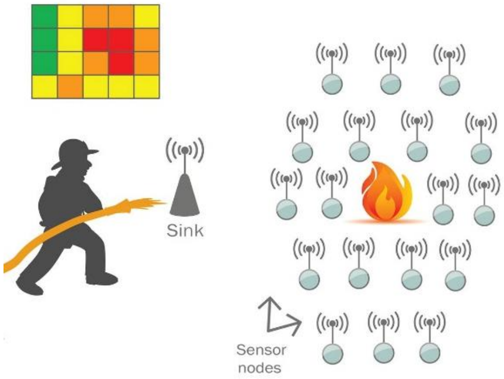


*Figure 2.2: Illustration of fire detection applications*

- Intelligent buildings such as optimizing energy consumption, building automation, and emergency. By monitoring temperatures, air quality and light levels in the building, steel distortion, and earthquake, the system collects the data and performs in real-time to make the right decision.

### 2.2.2. Industrial applications

- Applications for logistics such as monitoring freight shipping, tracking of goods**,** detection of unexpected container openings, monitoring of transport conditions like humidity and temperature**,** and identification of storage incompatibilities. Where the main goals of these applications are monitoring logistics and minimizing economic losses**.**

- Surveillance and preventive maintenance such as axles of train, manufacturing, and tire pressure monitoring. Which use sensor node into difficult-to-reach places and the humans difficult to control. The sensors node can detect up normal patterns and transmit the collected data into based station to process and determine the need for maintenance.

### 2.2.3. Military applications

- Battlefield surveillance applications: can detect the presence of nuclear, biological, and chemical agents. WSN applications for battlefield surveillance can thus save the lives of many soldiers.
- Border control applications: by using different types of sensors such as camera, motion sensor, etc. it supplies secure perimeter and monitor the border to prevent any unauthorized access to the country by alert patrols.

### 2.2.4. Health care

- Mobile patient monitoring applications: such as monitoring blood pressure, diabetes, and heart rate. Where wireless sensor in real-time can trigger alarm or make an action to reduce the issue when the system detects the deterioration of a patient condition. Figure.2.3 illustrate an athletic person wearing some sensors to collect important data about heart rate, numbers of steps, etc. and transmit these data into the internet.

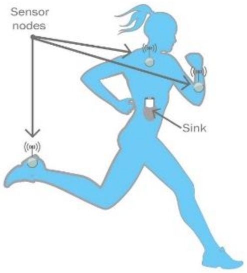


*Figure 2.3: WSN rehabilitation monitoring application*

## 2.3. Application requirements and constraints

Every WSN application has its own requirements that distinguish it from other applications. These requirements affect the network requirements and design. The constraints on the networks themselves are also parameter affecting the network design. Below is a description of some application requirements and network constraints that must be taken into account when designing an efficient WSN.

### Application requirements

Application Requirements are defined by the end user needs and demands to run the application without any degradation in performance. Some of these requirements are listed below;

- **Data precision:** sensing and transmitting data by the nodes must be accurate and do not produce faulty data.
- **Availability**: The application service in wireless network should not suffer from any failure in any node and should be always accessible when needed.

- **Long Lifetime:** which means the application must be available and under demand for user to monitor any phenomena as long as possible.
- **Quality of Service (QoS) support:** when the user requires a certain level of Service from an application and the network, they must satisfy these requirements regardless of what they are, and how they will be implemented.

**Constraints**

Depending on the environment where WSN application is implemented, many constraints must be considered to achieve high performance. Some of the most common constraints are listed below;

- **Size of the monitored area:** translated into number of nodes that is needed by the network.
- **Coverage**: translated into range of the wireless radiation can cover.
- **Type of target data:** determines the sensors type e.g., humidity sensors, light sensor, etc.
- **Power resources**: many of WSN applications require nodes connection through limited power sources and the batteries may be irreplaceable.
- **Mobility**: some applications require nodes with moving ability.
- **Deployment**: the distributed nodes into the area can be fixed (predicted) or randomly positioned. However, in both cases nodes must have the self-organization ability, such that they can organize the way they communicate among each other by themselves.

- **Cost**: the application requirement determines the type and number of nodes that effect the cost of the budget, which is high application constraint that must be taken into consideration.

WSN is designed for specific applications that have different interdependent requirements with many constraints, which require designing efficient WSN mechanism to overcomes these challenges and improve the performance of the network. For example, when the hardware of the sensor has some problems that may produce faulty data or may cause node failures that affect network availability.

## 2.4.Sensor networks design, architecture, and protocols

### 2.4.1. WSN topologies

There are many forms for WSN topology that can be used in sensor network. The simplest and the most common is star topology. Topology can be more complicated in a case when multi hop wireless mesh topology is used. Below is a brief description of the most common topologies.

**Star Topology:** is the simplest WSN topology. It can contain multiple nodes that are in the same radio range to the base node. The base node works to collect data from sensor nodes and transmit data to a sink node without using the neighbor nodes, an example of a star network is shown in in Figure.2.4.

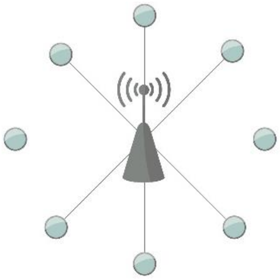


*Figure 2.4: Star network*

**Tree Topology**: consists of two or more Star networks, each base node is connected with hierarchical sink nodes. This topology is used for connecting areas that cannot be monitored with one Star network. An example of a tree network is shown in Figure.2.5.

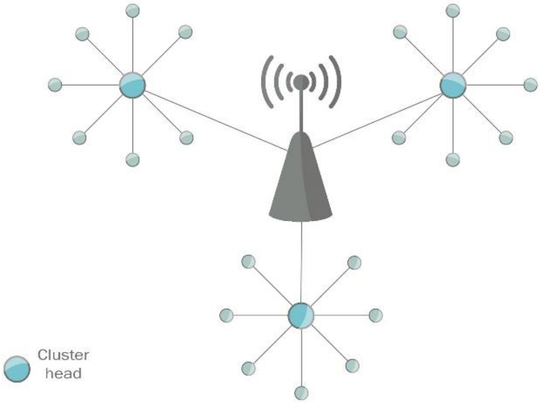


*Figure 2.5: Tree network*

**Mesh Topology**: is the most reliable topology which is designed for large WSNs with many nodes that are not in the same radio range from the sink node. When the node transmits a packet to the sink, intermediate nodes (between the sender and the receiver) pass the packets to each other until the sink node receives the data. An example of a mesh network is shown in Figure.2.6.

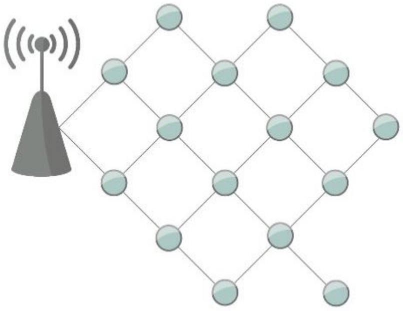


*Figure 2.6: Mesh network*

## 2.5. Communication protocols architecture

Figure.2.7 illustrates the WSN protocol stack which has almost the same design as the standard TCP/IP. However, the WSN protocol has three distinctive planes; power management plane for energy consumption, mobility management plane, to manage movement, and task management plane to manage the execution of tasks.

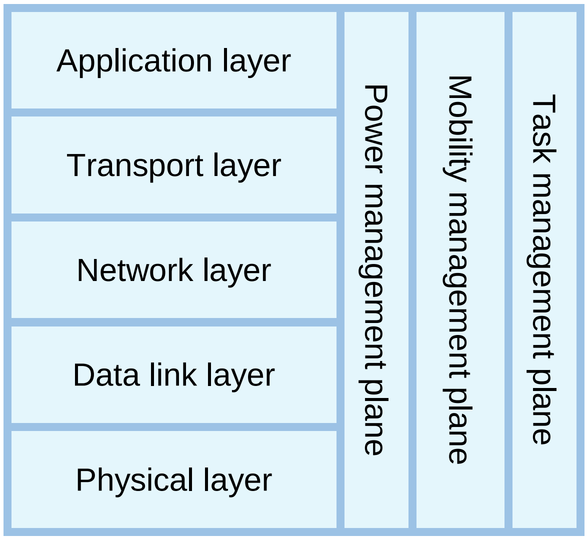


*Figure 2.7: The sensor networks protocol stack*

**Physical layer:** The physical layer is responsible for actual wireless network sending and receiving process, through individual nodes. The layer manages transceivers, frequency selection and clear channel assessment (CCA). In WSN the IEEE 802.15.4 is a protocol for low-rate wireless personal area networks (WPAN). This standard includes both physical (PHY) and MAC layer specifications.

**Data link layer:** The data link layer is divided into two sub-layers: the logical link control (LLC) sub-layer and the media access control (MAC) sub-layer. The LLC layer handles the error that occurs in the physical layer. The MAC layer is responsible for the management of the access to the radio channel. The MAC sub-layer works to avoid collisions between nodes when more than one node wants to use the channel at the same time. The collision avoidance is done via the MAC layer by deciding which node can access the channel. The management of the channel access can be classified into two approaches: contention-based and contention-free approach. Contention-based protocols are implemented by assigning random number of time slots to contending nodes, hoping to avoid collision, while Contention-free protocol is done through dividing resources (frequency, time, space, or code) by nodes in order to reduce the risk of collision.

**Network layer:** responsible for organizing end-to-end packet delivery by managing routing of the packets through intermediate nodes down to the right destination. This routing process is implemented using routing tables stored in intermediate nodes. Routing tables are used by routing algorithms to calculate the cost of each path and to select the path with the lowest cost towards the destination, i.e. the shortest bath.

**Transport layer:** this layer provides mechanisms to manage connection between source and destination through opening up and closing down the connection. It is also ensuring the reliable arrival of messages, provides error checking mechanisms, data flow, and congestion flow controls.

**Application layer**: This is the layer with which the user interacts and works to implement that application requested by the user.

## 2.6. Wireless Sensor Networks challenges

In this section we shall discuss the main research issues and challenges involved in designing and implementing WSN. The section is intended to give some details on these challenges and to show which of these we shall be tackling.

### 2.6.1. General design considerations

**Energy and network lifetime:** one of the major challenges before WSN is the ability to reduce power consumption and increase network lifetime. For this goal to be met, all protocols must have to be power consumption conscious. This implies that these protocols have to implement every possible means to be economic in power consumption. Furthermore, sensing nodes can be made to produce energy by implementing technologies that use sunlight, thermoelectric, or vibration in order to increase the battery, and ultimately network lifetime.

**Scalability:** This implies the ability to adapt to the changeable traffic load, and has the ability to deal with low and high data traffic without any degradation in performance such as increased latency and / or packet loss.

### 2.6.2. Communication architecture challenges

**Quality of service (QoS):** These are parameters that must be taken into consideration to satisfy the application requirements like minimum delay, reliability, and throughput.

**Mobility:** Improving the performance of WSN can be done by increasing the network capacities through for instance enlarging the coverage area. One way to do this is through moving a data collection node e.g. a flying drone, or a flying balloon, to collect data from sensing nodes rather than forwarding these data hop by hop.

**Internetworking:** this feature triggers the need for a gateway node in order to link WSN with the Internet.

**Security:** Application in the domain of health care and military are examples of critical application that requires security and confidentiality. There are many security concerns which must be avoided such as attacks that turns down the network or steal some critical data.

**Heterogeneity:** WSN should operate efficiently despite diversity in devices used, traffics types, and network architecture employed.

## 2.7. Internet of Things (IoT)

Figure 2.8 illustrates the Internet of Things (IoT) which is a system of smart interrelated physical and technological objects with unique identifiers connected to the Internet. IoT includes sensor, computing devices, or actuator. One major issue in these items is the constrained resources such as memory size, battery lifetime, and CPU processing power. These objects have the ability to transfer data over a network using the Internet Protocol (IP) without direct human interaction. Smart objects use wireless low power lossy networks to communicate with the internet, and with each other in order to make the environment more intelligent [1]. IoT networks have complex structures that use IPv6 to communicate. The first emergence of IoT was in 2009. However, at that time the capabilities of the used nodes were minimum and were not fully integrated with the Internet. The IEEE release of the 802.15.4, designed to operate in a low-power Wireless Personal Area Network (WPAN) environment, made a big boost for IoT deployment.

The other major boost came when the in 2011 the 6LoWPAN was launched. This protocols allow the IEEE 802.15.4 to utilize IPv6 over Low Power Wireless Personal Area Networks. 6LoWPAN protocol makes it possible to connection constrained devices with full power IP-based devices i.e. the Internet [2].

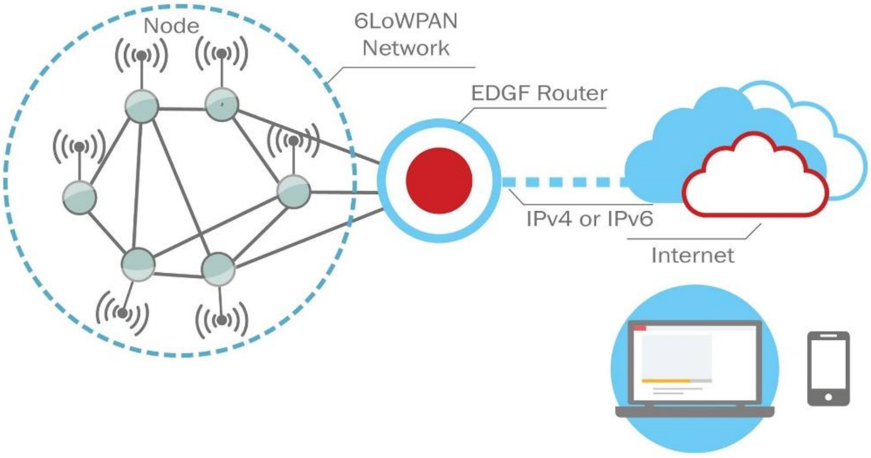


*Figure 2.8: Architecture of a simple IoT*

## 2.8.Protocol stack in Internet of Things

In order to improve Quality of Service (QoS) in IoT, the protocol stack must be fully understood to see how we can introduce modifications that will improve QoS. 6LoWPAN has similar structure like TCP/IP with an additional thin layer between the Data link and Network layer called adaptation layer, as shown in Figure.2.9.

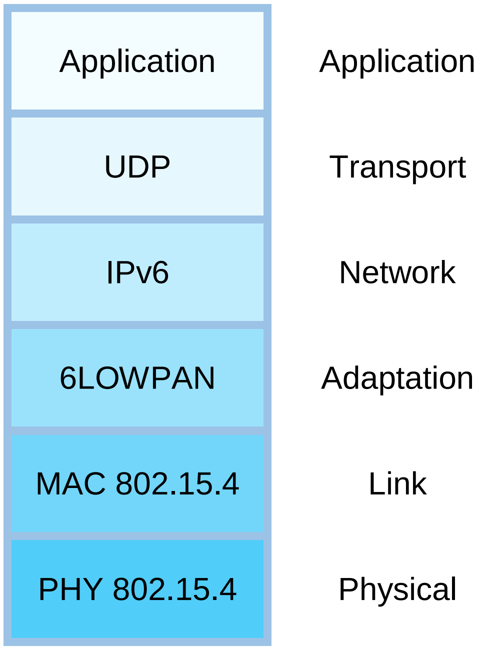


*Figure 2.9: 6LoWPAN protocol stack*

### 2.8.1. IEEE 802.15.4 protocol (Physical and Data-link layer)

IEEE 802.15.4 is a protocol used in wireless networks by constrained devices with limited resource such as low power, and low memory. This protocol is designed to support physical and MAC layer. At the physical layer level, IEEE 802.15.4 provides variable data rates such as 250 kbps (2.4GHz), 40 kbps (915MHz), 20 kbps (868MHz). At the level of MAC layer IEEE 802.15.4 supports the access to the radio channel by using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) algorithm. On top of these layer come adaptation layer which implements the 6LoWPAN protocol.

### 2.8.2. 6LoWPAN protocol (Adaption Layer)

Many problems emerged from trying to make sensor networks IP-based. 6LoWPAN protocol works very efficiently to solve these interfacing problems. It enhances the ability to connect constrained devices to the real world Internet [3]. Also the 6LoWPAN consider as the technology behind the wide spread deployment of IoT. 6LoWPAN allows IEEE 802.15.4 to utilize IPv6 over Low Power Wireless Personal Area Networks [3].

The 6LoWPAN protocol provides three functionalities: packet fragmentation and reassembly, header compression, and data link layer routing for multi-hop connections. 6LoWPAN manages packet fragmentation and reassembly, this is required since the maximum transmission unit (MTU) size for IPv6 packets over IEEE 802.15.4 is 1280 octets, and a full IPv6 packet does not normally fit in an IEEE 802.15.4 frame. If the IPv6 packet size is less than 127 no fragmentation is needed from 6LoWPAN side. In addition, 6LoWPAN works to compress the IPv6 header to increase data payload. Figure.2.10 shows the architecture of 6LoWPAN network.

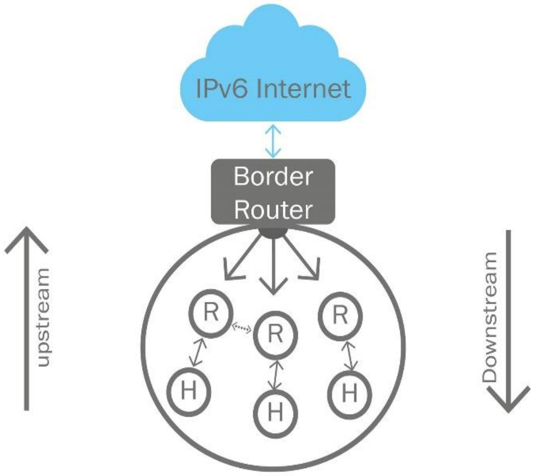


*Figure 2.10: 6LoWPAN Architecture in IoT*

### 2.8.3. IPv6 (network layer)

IPv6 is the new invention after ipv4 protocol that fills the issue of out of range addresses which is encountered by IPv4. Ipv6 uses 128-bits of addressing provide approximately 340 trillion IP addresses, while IPv4 uses only 32-bits. Figure.2.11 presents the structure of an

IPv6 header. This header includes the IP version, traffic class, flow label, payload length, next header, hop limit, and the source and destination addresses.

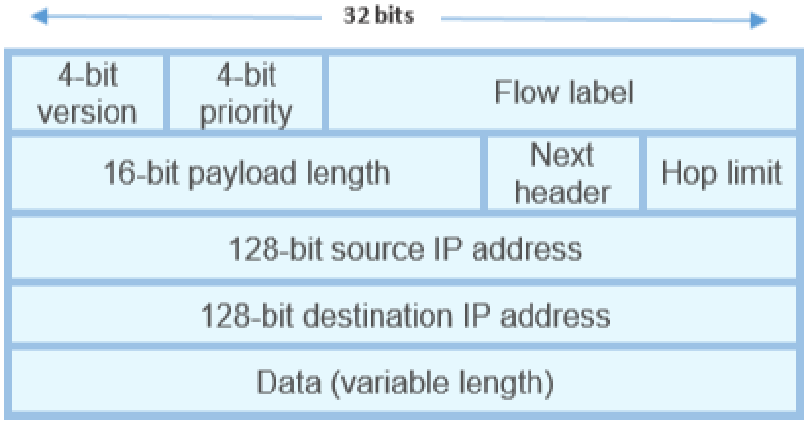


*Figure 2.11: IPv6 Header*

Typically, Ipv6 subnets 128 bits into two halves, first half denotes the network portion and second half denotes the host portion. Figure.2.12 presents the ipv6 address format.

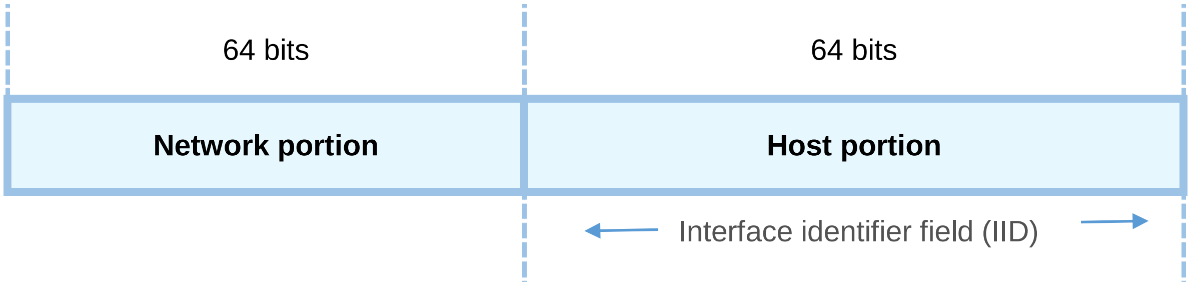


*Figure 2.12: IPv6 Address Format*

## 2.9.Security in Internet of Things

Security is a basic service that has to be provided in IoT networks. To achieve security different requirements are to be met. The most important of them are listed below:

- **Confidentiality**: this implies that the transmitted packets between the sender and the receiver are encrypted, where the third-party cannot access the transmitted packets.

• **Authentication**: is used to identify each side; sender and receiver to each other's. Authentication prevents any devices or attackers that claim different identity to reach important data or inject invalid information.

• **Integrity**: Ensures that the transmitted packets do not suffer from any error or change, and the destination receives the packets exactly as they were sent. Cyclic redundancy checksum (CRC) is used to detect random errors during packet transmission.

• **Availability**: IoT devices should be allowed to reach any other devices anytime, anywhere. This requires an appropriate mechanism to prevent any possible attacks such as Denial of Service (DoS) which works to turn down the service. Availability in IoT is hard to achieve since Denial of Service (DoS) attacks can be launched at any layer. Other possible means to impact availability is jamming of the radio channel, exhausting of the power supplies of the nodes.

• **Authorization**: is a security method that ensures just the authorized IoT devise is capable of utilizing the network rescues.

### 2.9.1. IPsec

Internet Protocol Security (IPsec) [4] is different from other security protocols like Transport Layer Security, TLS or Secure Shell, SSH which are transportation layer protocols. IPsec protocol operates at the network layer level, which is considered more appropriate for securing IP packets on an end-to-end basis. IPsec works to encrypt and authenticate each IP packet of a communication session. This protocol enhances the packets protection without the need for the intervention of the application or any other layer.

IPsec uses two mechanisms one is optional and the other is mandatory. The first one is denoted as the Authentication Header, (AH) [5] protocol. It is implemented to give integrity for packets and provides authentication for communicating sides. The other one is the Encapsulating Security Payload (ESP) [6], which is a mandatory protocol. It provides the basic security services and confidentiality. The IPsec can be implemented both in transport or tunnel modes. In transport mode as shown in Figure.2.13, the IPsec header (ESP or AH header) is inserted after the IP header and only the payload of the IP packet is encrypted. The IP header (the IP addresses) is not affected because the packet header is not encrypted or altered.

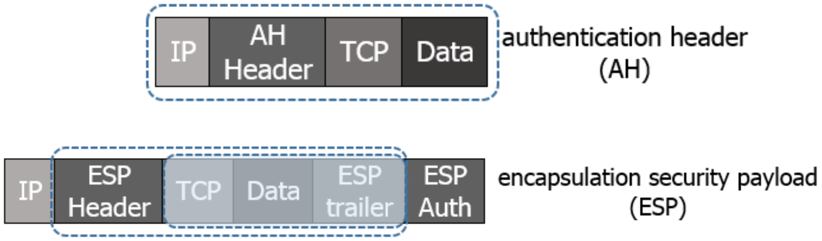

IP
AH Header
TCP
Data
authentication header (AH)
IP
ESP Header
TCP
Data
ESP trailer
ESP Auth
encapsulation security payload (ESP)

*Figure 2.13: Transport Mode*

When tunnel mode as shown in Figure.2.14 is used, a Virtual Private Network (VPN) is created. In this case the complete IP packet is encrypted and/or authenticated and then encapsulated into a new IP packet with a new IP header. This type of security is more robust, since a new IP header is used which is unrelated to the nodes, (using the tunnel).

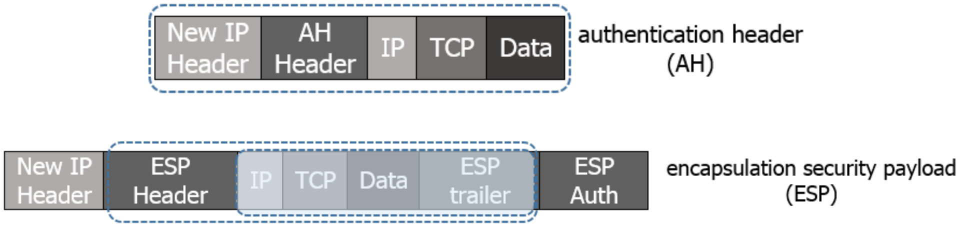


*Figure 2.14: Tunnel Mode*

Another vital feature of IPsec is the Internet Key Exchange protocol (IKE or IKEv2). IKEv2 [7] is often used as a key management scheme to store all security associations and policies for each device. It is also used to set-up an automatic connection between devices by determining and distributing secret keys without the need for pre-shard key.

## 2.10. Overview of Quality of Service provisioning

Quality of Service (QoS) have different meanings depending on point of view. QoS is the guarantee a certain level of performance by using existing resource efficiently. QoS can be defined by the user or application as requirements, which are to be satisfied by the system, or the network. Figure 2.15 illustrates two interdependent viewpoints, the application and user viewpoints. This work considers providing QoS support from the network perspective.

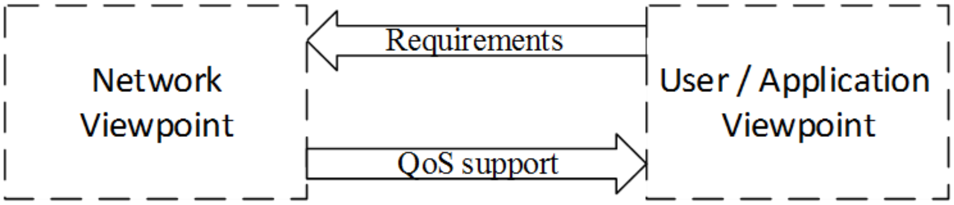


*Figure 2.15: QoS interdependence*

### 2.10.1. Factors affecting the quality of Service

Some IoT applications have certain requirements such as high availability, stability, and/or low delay. These requirements are interpreted by network in terms of packet latency, throughput, and/or reliability. Some of the most common factors affecting QoS are detailed in the next section.

**Low throughput**: this factor is created due to reasons like contending data flows, which results from sharing the same network resources. Moreover, when node receives traffic more than it can handle, the number of received packets inside queue buffer reaches the maximum and after the queue buffer becomes full all received packets will be discarded.

**Packet loss**: Some applications can tolerant packets loss than others such as multimedia. On other hand losing packets causes degradation in the quality and reliability.

Delays Propagation, queuing, and processing delays adds up to latency time. Moreover, high load may increase delay time too.

**Jitter**: this factor is a measure of the variation of time needed for a packet to arrive from source to destination. This might create problems, especially for real-time applications like multimedia. To fix the delay, variation one can use a buffer to compensate for the variation in delay time.

**Out-of-order delivery**: delay variation changes order of packets arriving at destination side. In this situation reordering mechanisms are needed.

### 2.10.2. QoS techniques

Difference QoS techniques are needed in order to overcome common issues that reduce the network performance. In the following section, we present several QoS techniques at various layers of the protocol stack to give minimum delay, reliability, and throughput.

### Scheduling

This technique aims to organize the transmission for simultaneous traffics flow. There are many scheduling techniques such as First in First Out (FIFO), Weighted Fair Queuing (WFQ), and Earliest Deadline First (EDF) technique. All of these techniques try to make through of some issue like starvation packets arrives.

### Rate limiting

Rate limiting is using common techniques to control the rate of transmitting packets from the node. Traffic shaping is one of rate limiting techniques. Another technique for rate limiting is flow control by adjusting the transmission to prevent the flooding of the receiver node or the overflow of the network. The transmission process can be managed by receiving a sent back acknowledgment or after timeouts of the adjustment sending rate. This mechanism is typically run on the transport layer.

### Congestion avoidance

This technique observes congestion indications such as packet delay, dropping. The most common congestion avoidance mechanisms are fair queuing, scheduling algorithms, explicit congestion notification (ECN), and Random early detection or random early drop (RED). This mechanism is typically run on the transport layer.

When the network has resource constrained conditions or the network load increases and reaches to the peak capacity, it will generate an issue that may affect the network performance, in order to improve the performance, QoS control technique is required which aims to optimize network resource utilization and to prevent network overload.

### 2.10.3. QoS provisioning at the MAC layer

The goal of MAC protocol is organizing the channel access to improve the network performance and satisfy the application requirements. In IoT network with limited resources, QoS-aware MAC protocols must be able to support applications with high requirements. The performance of MAC protocols can be expressed in terms of energy efficiency, throughput, latency, reliability, jitter, and fairness. Figure 2.16 presents the correlation between factors that affect the design of efficient MAC layer protocol which tying to compromises between different factors to achieve the requirement application. For example, when provide high reliability may increase the energy consumption and reduce the throughput because of the overhead by retransmissions, acknowledgments, and control messages.

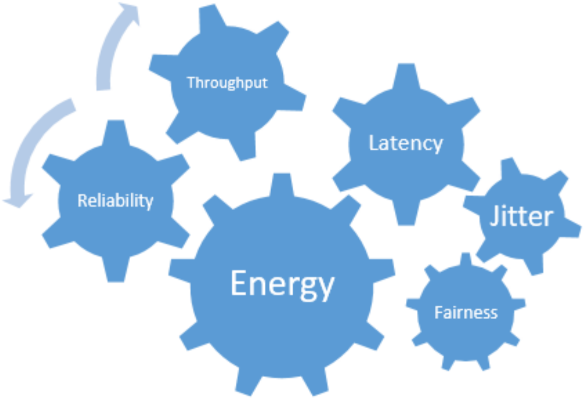


*Figure 2.16: interdependence of design factors*

# Chapter 3

## Related Work on QoS Metrics and Protocols

In this chapter we review the research efforts in the domain of QoS in general and their implementation at the MAC layer level in particular. The review is classified according to the techniques used to the channel as depicted in Figure. 3.5.

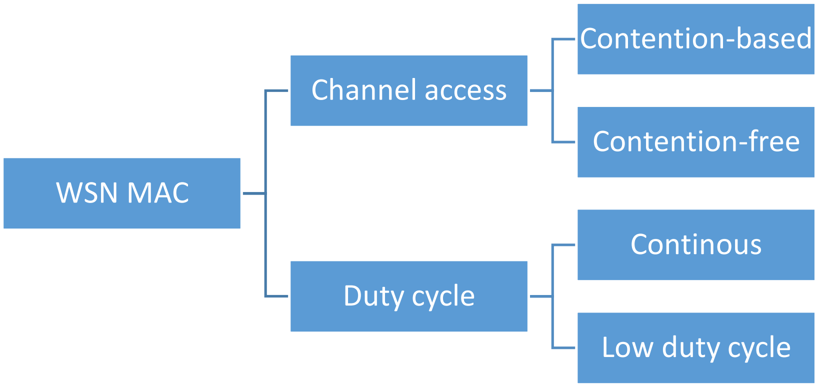


*Figure 3.1: classification of MAC protocols*

In the contention mode the channel is shared by all nodes and the bandwidth is divided among nodes on-demand basis. The main feature of contention-based protocols is the low latencies introduced, and the good bandwidth utilization [8]. In this protocol a collision occurs when multiple nodes in the same collision domain try to transmit simultaneously. Therefore, collision avoidance is needed to reduce the degradation of the network performance. In WSNs, contention protocols typically utilize CSMA/CA [9] which check the channel for any transmission and avoid the collisions by giving the transmitting node random back-off time exponentially [8], [10], [11], [12]. The back-off time which is referred to as Contention Window (CW) makes influences latency, collision probability, and network utilization, substantially.

To improve QoS for WSN, Klepec and Kos technique in [13] described the behaviour of packet transit times for a delay sensitive applications with minimum bandwidth constraint. They present two First Input First Output (FIFO) queues that operate under a complete sharing scheme. Although that gives low delay for higher priority packets, the solution makes the queue size for high priority packets smaller at the cost of high data loss for low priority packets.

Taj Rahman [14] propose a method which differentiates between high and low priority when routing sensory data to the sink node(s). The priority of sensory data is determined through a capacity assignment mechanism to mitigate congestion and packets dropping. The main idea come from, differentiate the data packets and then schedule the data packets according to the priorities among three levels of queues first come first served scheduling policy. After that, the total path capacity will be calculated. This calculation requires to send a burst of control packets to its parent nodes, and transfer data according to the path capacity. It minimizes end-to-end data packets lost for high and low priority data, also reduce delay for both high and low priority data at different levels. But the Taj Rahman, do not test other QoS performance metric like throughput and packet delivery ratio.

Saxena et al. [15] proposed a QoS mechanism at the MAC layer based on a CSMA/CA using a contention window (CW) and a dynamic duty cycle techniques. This mechanism is designed for multimedia applications that targets reduction of energy consumptions in constrained WSN network. This approach implements priority mechanism using multiple queues and CW according to traffic priority class. This approach has a major disadvantage, which is a significant increase in the overhead complexity.

Yigitel et al. [16] proposed Diff-MAC priority protocol, which uses different techniques for packet prioritization and contention. Diff-MAC uses weighted fair queuing (WFQ) techniques to control the throughput traffic for each class. Saxena et al, solution and Diff-MAC solution both improved the performance of the network that use contention-based protocols as channel access techniques by enhancing throughput and latency. Diff-MAC protocol improves fairness between competing traffic flows that have different types and fast adaptive to changing network conditions. However, Diff-MAC solution has a major disadvantage since it entails a complex adaptive technique, which downgrades the performance of the entire network.

The main issue in contention-based protocols is the issue of hidden node and idle listening. Hidden node is solved by using additional transmit Request to Send (RTS)/Clear-To-Send (CTS) messages or a combination of carrier sensing and control packets. However it consumes extra bandwidth and increase in communication overhead [17]. This overhead prevents the system from reaching an optimal channel utilization. The idle listening issue stems from unknown transmission times from other nodes, which makes the receiver continuously sensing the channel for incoming packets [18], [19].

The transmission in Contention-free protocols is typically predetermined by using Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), and Code

Division Multiple Access (CDMA) [17]. These techniques are collision free channel access. They perform well when the traffic flows are heavy as the predetermined transmissions allow to reduce idle listening, enables accurate QoS control, and prevent collisions. However, the disadvantages of Contention-free protocols under low traffic flows are low channel utilization and high latency since transmission process from the node must wait the reserved time slot. In addition, the scheduled transmissions are assigned by a central manager which reduces scalability than contention-based protocols. Therefore, many WSN proposals use distributed methods where nodes exchange known reservation information within two-hop neighborhood [20], [21], [22], [23]. Another issue is to know the right number of reservations allocated whether it is over or under reserved. In the case of high reserved capacity the energy and capacity are consumed unnecessarily, while low reserved capacity increases transfer delays and may cause packet losses [24]. As a result contention-free protocol can only be used on centrally controlled networks [17]. And not appropriate for WSN, in large scale network with various traffic loads.  However, there are some methods to get rid of some issues by reserving only a part of the slots, while using other slots dynamically on-demand. Such as Y-MAC [25] node used additional channels in case of high traffic load beside the original channel when the traffic load is normal. Still, the reservation issue for the base channel slots remain in Y-MAC.

## 3.1.State of the art of QoS-aware MAC protocols for IoT

In the following, we will review latest research efforts in the field of QoS techniques design at MAC layer protocol for IoT networks. The discussion will detail both advantages and disadvantages of each technique. Awan in [26] proposes queuing system with pre-emptive resume (PR) service priority with complete buffer sharing scheme by all classes of traffic under a push out mechanism. These approaches do not look into the QoS requirements like throughput, and packets delivery ratio in IoT networks. Pushing out low priority traffic technique used to avoid data loss of high priority traffic reduces the throughput, and packet delivery ratio of low priority packets. In our model we overcame this issue by giving the low priority an exponential back off time more than high priority.  By doing so we reduce the packet loss at the same time increase the throughput of the network and give the high priority packets the advantage of passing through the network.

Min Y.U. et al. [27] proposes packet scheduling techniques that are used in Tiny OS [28] [29]. There are two types of scheduling techniques in IoT. These are cooperative or pre-emptive. Cooperative scheduling technique depends on two queues with each one has different priorities. This technique switches dynamically between the two queues according to the deadline of newly arrived packets. If the new arrived packet has low deadline it will store into high priority queue, and if the packet has longer deadline it will store into low priority queue.

Adil A Sheikh [30] propose a new routing framework for VSN (visual sensor network) to deliver critical imagery information with system's time constraint. He implemented his proposed framework using Contiki and simulated it on Cooja simulator. The proposed priority-based routing framework makes sure that intermediate nodes forward high priority packets (first pass image layer) faster than low priority packets (second pass image layer). The VSN nodes send advertisements to their neighbors declaring identities and their number of hops from sink. These advertisements are sent periodically to allow the intermediate nodes decide the priority of the incoming packets based on the number of hops from sink, also the intermediate nodes using priority queue mechanism to organize the incoming packets.

Tanmay Chaturvedi [31] investigate a scalable multimode-based MAC protocol, they propose the IoT-MAC to reduce contention of channel access due to coexist of many IoT devices, which consists of a channel contention period and a data transmission period. The two periods interchange periodically and are synchronized by the Base Station. The proposed data transmission scheduling algorithm used to maximize data collection under the constraints of radio link quality and remaining energy of the IoT node, while ensuring a fair access to the radio channel. So the nodes find their transmission slot within the super frame and only transmit during their scheduled time to prevent interference.

Thien D. Nguyen [32] introduce an adaptive energy efficiency algorithm, known as ABSD (Adaptive Beacon Order, Superframe Order and Duty cycle) that changes the MAC parameters of the IEEE 802.15.4 sensor nodes in response to the queue occupancy level of sensor nodes and the offered traffic load conditions. The ABSD algorithm minimizes the network contention which could in turn improve the energy efficiency as well as the throughput.

Irfan Al-Anbagi [33] introduce medium-access approach, namely, delay-responsive cross-layer (DRX) data transmission. DRX is based on delay-estimation and data-prioritization steps that are performed by the application layer. The delay-estimation is done by prediction the E2E (End to End) delay and creating cross-layer measures. DRX uses application-layer to control the medium access by performs delay estimation, if the estimation delay is higher than the delay requirements from the application layer, they give higher priority to the node to access the channel by reduces the CCA duration. These schemes achieve delay responsiveness by modifying the parameters of the physical layer of the IEEE 802.15.4 protocol.

Muhammad Akbar [34] propose a tele-medicine protocol (TMP) under IEEE 802.15.4 slotted CSMA/CA with beacon enabled mode. They combine two optimizations methods, MAC layer parameter tuning optimization and duty cycle optimization. Duty cycle optimization adjusted by offered network traffic load, which delay reliability factor archives minimum latency by estimation channel access and collision probability. And super-frame duration used in beacon that exchange all network information to estimate the total required time for transmitting data 100 Kbps.

Sabin Bhandari [35] propose a priority-based adaptive MAC (PA-MAC) protocol for WBANs, which use the beacon channel for transmitting and reception of beacon frames to exchanges control information with coordinator and use data channel for rest of the communication. PA-MAC classify the Data traffic into four priority level and allocates time slot dynamically according to the number of nodes in each traffic priority category. Prioritize the data traffic is done by using priority-guaranteed CSMA/CA in CAP (contention access period). The downgrade of PA-MAC, when the node wants to reserve the resources for periodic traffic should send a request to the network coordinator.

Saima’s paper [36]  proposed message scheduling with service provisioning technique. This technique classify messages as either high priority or best effort messages, and uses the best QoS algorithm. Saima uses clustering based approach which categorized IoT nodes into subgroups. And assign each subgroup with a broker node, which collects the data from other nodes in the subgroup and redirects the messages to the base station by handling separate queues for best effort and high priority messages.

In conclusion, most of the efforts in this research area aim to improve one aspect of QoS requirements (e.g. latency), and limited support to different traffic types like secure traffics. As a result, the proposed protocols are optimal only for specific use cases. This Thesis presents QoS designs at MAC layer for traffic differentiation with multiple QoS metrics. Thus, the protocol designs enable heterogeneous IoT applications with varying QoS requirements to operate in the same network.

# Chapter 4

## Efficient QoS Provisioning at the MAC Layer in IoT Network

This chapter deals with QoS provisioning at the MAC layer in IoT networks designed for secured traffic. Chapter 3 clearly indicated a lack of QoS mechanisms in IoT networks. This work is meant to fill in this gap. The proposed mechanism presented in this work aims at enhancing QoS at the MAC layer level, in what is termed as Secure Traffic Priority Differentiation (STPD). STPD uses a prioritization mechanism employed in conjunction with Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) protocol, which is the main protocol used in wireless data network. The proposed approach intends to improve channel utilization, in addition to providing better QoS for the data delivery. This chapter is set to describe the design and the operation of our proposed solution.

### 4.1.Motivation

Internet of Things (IoT) provides communication and networking among physical objects, devices, systems, in addition to computers and smart devices. It enables the collection of diverse types of data that help making smart and educated decisions. This in turn helps in making human life smooth, safe and efficient. However, this comes at the expense of privacy and security. Many IoT networks contain classified data which requires high level of security and priority. Classified data need to arrive at the destination side faster than other competing messages sent over the same network, at the same time. To our knowledge, and as has been stated in the literature review, see chapter 3 for details, existing protocols are not designed to provide such badly need secured and efficient QoS in IoT networks.

MAC layer plays a central role in improving the performance of IoT systems and networks. To improve network performance and efficiency, one needs to be fully aware of the mechanism used to transmitting data packets in IoT networks. In this research, we implement the well-known IPsec security protocol which is presented by Raza et al. in [37] with our proposed STPD mechanism that works at the Adaptation and Data Link layer level. Changes were introduced to the IEEE 802.15.4 MAC protocol in addition with 6LoWPAN protocol The proposed changes introduced to the aforementioned protocols, are expected to produce better performance for the networks, especially in terms of throughput, correct packets delivery ratio, and end-to-end delay.

## 4.2. Design considerations and assumptions

The proposed solution aims at generating a more efficient MAC protocol for IoT secured applications. We call these applications "secured" as the traffic generated by nodes with various sensing capabilities requires high level of security to be delivered.

Indeed, different kind of sensors generate data traffic with special characteristics, such as predefined data rate and packet size, as well as different level of QoS requirements. These QoS requirements, are defined in terms of latency, reliability and bandwidth. Additionally, traffic load generated by these networks fluctuates, from very low to very high data rates. Data traffic too is not distributed evenly among all nodes in IoT networks.

Our proposed solution targets both transmission modes; the connection setup and the application data transfer modes. Both modes are to be secured. Initial connection setup for secured connection uses the IPsec protocol, which consists of the following exchanged massages characteristics;

- Initiation of secured connection is done via the exchange two pairs of messages between sender and receiver.
- The single message are sized between 3 to 5 packets, which are considered large.
- When any of these packets get lost, the two communicating sides have to start over again.
- If the time is over, due to long connection will fail and has to start over again.
- The connection will start all over again after the communicated security key gets expired.

Our proposed solution is designed to overcome the issues that introduce delay in initiating the secured connection. One issue that our solution resolved has to do with the possibility of breaking the connection, which requires it to start over again. When this happens, it leads to increase in energy consumption, increase in latency, and decrease the reliability. This will also reduce the number of correctly received data packets, as the first phase in which the exchange of authentication and security association was not done correctly. Figure 4.1 illustrate the basics idea of our proposed modification on the MAC protocol. STPD meant to overcome this issue, as is explained below.

- Our proposed solution will be able to classify traffic into a high or a low priority traffic. It will too be able to provide low latency as well as high throughput to high priority traffic.
- Our proposed solution will be able to distinguish between data and connection setup traffic. And this will allow the system to offer the highest possible priority for the secured connection setup.
- Our proposed solution is ultimately adaptive, and that makes it operating efficiently under different circumstances, such as variable traffic loads. What makes our proposed solution adaptive is the implementation of our priority mechanism and the use of dynamic memory management that is assigned to the queue which allows to adapt to different traffic loads.
- Our proposed solution will function to enhance the network resources utilization of the entire network. This goal will be achieved through utilizing the unused memory of the node.

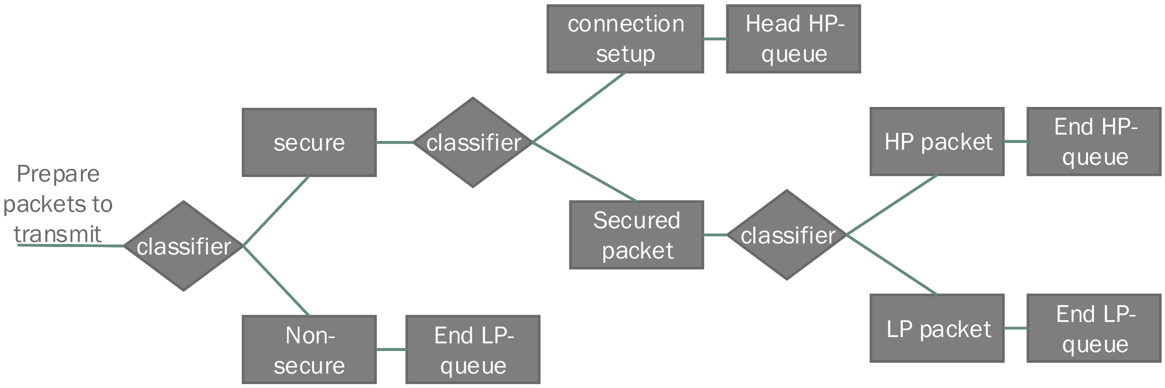


*Figure 4.1: Basic idea for STPD*

## 4.3.Basic Principles of Secure Traffic Priority Differentiation (STPD)

This section presents the design details of STPD, our novel adaptive MAC protocol for IoT networks. The novelty of our approach stems from the idea of utilizing the strengths of contention-based protocols in maximizing channel utilization. This will enhancing network utilization and providing reliability in establishing a secure connection. Additionally, STPD provides a novel prioritization mechanism designed mainly to fulfill QoS requirements.

The STPD is divided between adaptation layer 6LoWPAN and medium access control layer as shown in Figure 4.2. Functional details of these layers are provided in the sub-sections below.

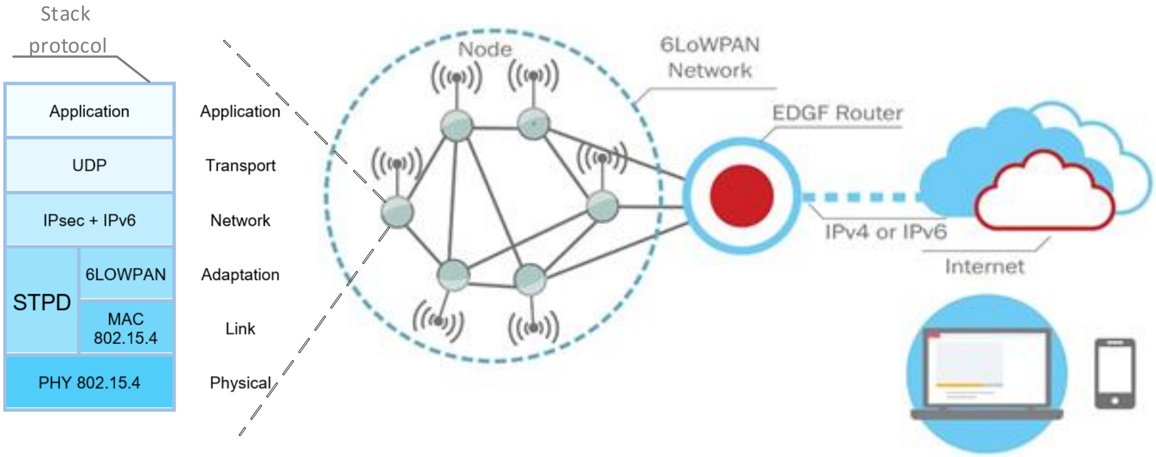


*Figure 4.2: network architecture used STPD*

### Adaptation Layer (6LoWPAN protocol):

When the packet crossing the adaptation layer to be send, STPD parts in 6LoWPAN protocol gets the source IPv6 address from the prepared packet. And compare the IPv6 with the priority table to find the priority class for this packet. After that our STPD protocol passing the value into lower layer (MAC layer). We presume that the priority table is distributed among all nodes in the network. This table contains the priority level and the IPv6 address for all nodes in the network. In addition, STPD protocol can identify the IPsec initial secure connection and gives it the highest priority. Afterwards the priority value is passed to the lower layer (MAC layer).

**Medium access control layer (MAC protocol):**

Once the MAC protocol gets the priority class for the packet, it implements the STPD packet prioritization mechanism to improve the network performance and achieve the QoS requirements. In next sub-sections we describe this mechanism in more detail.

### 4.3.1. packet prioritization

The proposed mechanism will work to perform prioritization to ensure the following;

- Nodes with high priority traffic will be given better chance to compete to access the channel over low priority nodes. This is achieved through allocating low priority packets longer back off time than is given to high priority packets.
- Packets with high priority will be processed before any other packets soon as they are created or received at any node. This is achieved through forwarding these packets towards destination soon as they arrive at the queue. Subsequently, low priority packet will be treated.
- Highest priority is given to connection setup exchanged messages, as is required by the IPsec protocol. The exchange of these messages is given precedence over all other kind of packets, what so ever, being high or low priority.

To keep the operation of STPD simple, we dived traffic into two classes: high priority (HP) and low priority (LP) traffic. High priority traffic takes precedence over delay-tolerant low priority traffic. This classification minimizes latency of HP traffic. We believe that there is no need to consider any intermediate priority in our system, as most applications requires either LP or HP. Adding a third priority level will add to the solution complexity substantially. STPD consists mainly of two First Input First Output –FIFO- queues. Each of these queues will have different mechanism for assigning priority, as is depicted in Figure.4.3. Next we explain the STPD scheme in detail.

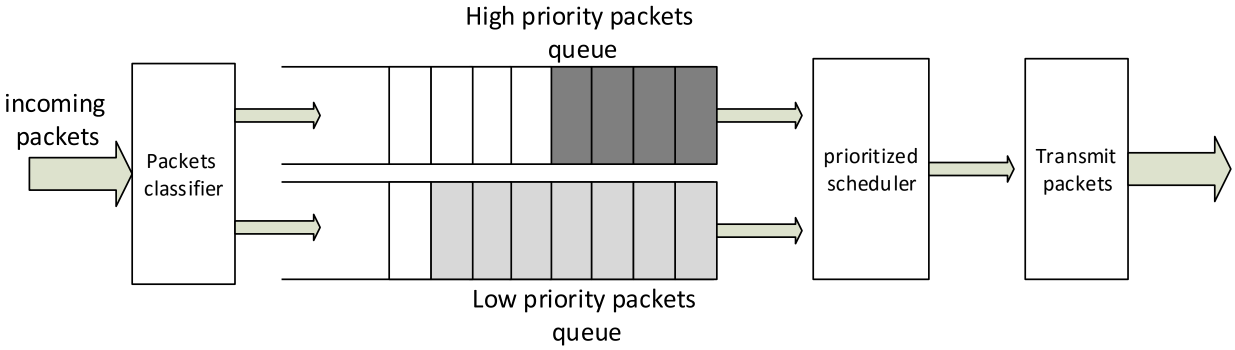


*Figure 4.3: STPD arbitration scheme*

**Packets classifier:**

- We presume that the traffic class is statically set based on pre-defined priority table. This table is distributed to all nodes in the network to be implemented.
- When a packet is submitted to the data link layer from the upper layers, a classifier checks whether the packet belongs to the HP or LP traffic class and puts it into the appropriate queue. But if the packet belongs to the highest priority class it is inserted at the head of the HP queue, which guarantee that it will be immediately processed.

- In addition to the HP packets, HP queue contain packets that are used to generate secure connection. On the other hand, low priority queue contains low priority data packet in addition to routing and control packets.

- STPD has a dynamic memory allocation mechanism in the priority queue. Every queue can handle fixed number of packets. However, with our proposed STPD model, this has changed into a dynamic by allocate half of the size of the original queue to be static and the other half to be dynamic. The dynamic memory can be used when the number packets that are allocated to the queue reaches the default size. Thereafter, the dynamic memory borrows one packet size from the other queue and after that when the node finish transmission successfully, memory free will be executed to release the allocated memory.

**Prioritized scheduler:**

- STPD uses a strict priority scheduler to decide which packet is to be sent, so that HP traffic has always priority over LP traffic. The scheduler systematically selects HP packets as long as the queue is occupied or is not in a state of back-off from collision. If this is not the case, the scheduler continues with the transmission of LP packets.

**Transmit packets:**

- The proposed system provides low access delay for HP packets compared with LP packets when collision occurs. This takes place via giving the transmitting node low random back off time.

## 4.4.STPD Operation

STPD operation is decided based on the following criteria; whether the node has data to transmit, or the node receives packets from neighboring nodes. In principles, nodes can perform on of the following operations; transmits while the channel is idle, transmits while the channel is not idle, receives data, and does nothing, see Figure 4.4.

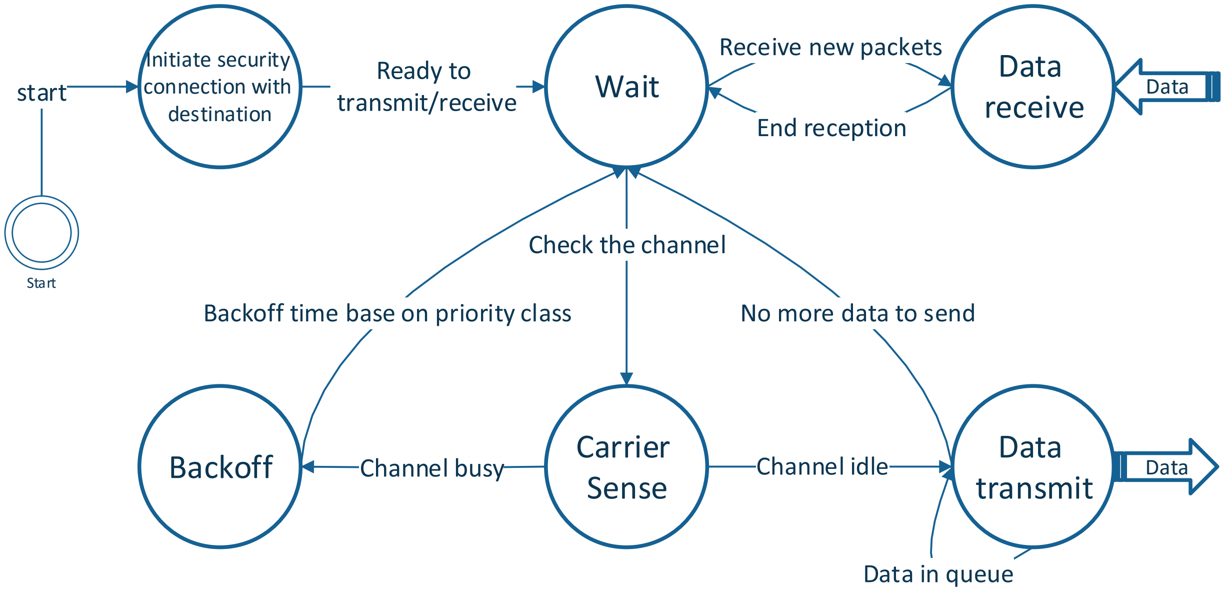


*Figure 4.4: state machine of STPD*

**States of nodes depicted in the diagram are described in what follows;**

**Initiation of Secure Connection:** In this phase, the node creates a secure connection with the destination. Once the data is sent, the node switches to the Wait state.

**Wait**: The node goes into Wait state when it finishes sending the data. To reduce the energy consumption in this state, the node switches off the radio signal transmitter.

**Back-off**: if the node experiences a collision which indicates that the channel is busy, the node goes into this state. The node then checks the priority queues to see whether there is any packets waiting to be transmitted. Subsequently, it calculates the back off time for HP and gives it lower random value than that of LP packets. While waiting the back-off time to finish, the node goes into the Wait state and continues listening to the radio channel for any new arrived packet. If new packet is detected it goes immediately into Data receive state.

**Carrier Sense:** When the back-off time expires or the node wants to transmit for the first time, the node switches to the Carrier Sense state. In this state the node listens to the channel to determine whether another node is transmitting. When the channel detected idle, the node goes into the Data transmission state and starts transmitting, otherwise it goes into the back-off state.

**Data transmission:** This is the state when the node is allowed transmitting. Once transmission is completed or stopped the node goes back into the Wait state.

**Data receive:** During the Wait state the node will be also in the listen state. Soon as a packet is detected, the node goes into Data receive state. When the reception process is completed, the node goes back into the Wait state. Remark that reception has priority over transmission, and when the reception starts it cannot be interrupted by any action.

Finally, we illustrate the flow diagram of our STPD protocol in Figure 4.5.

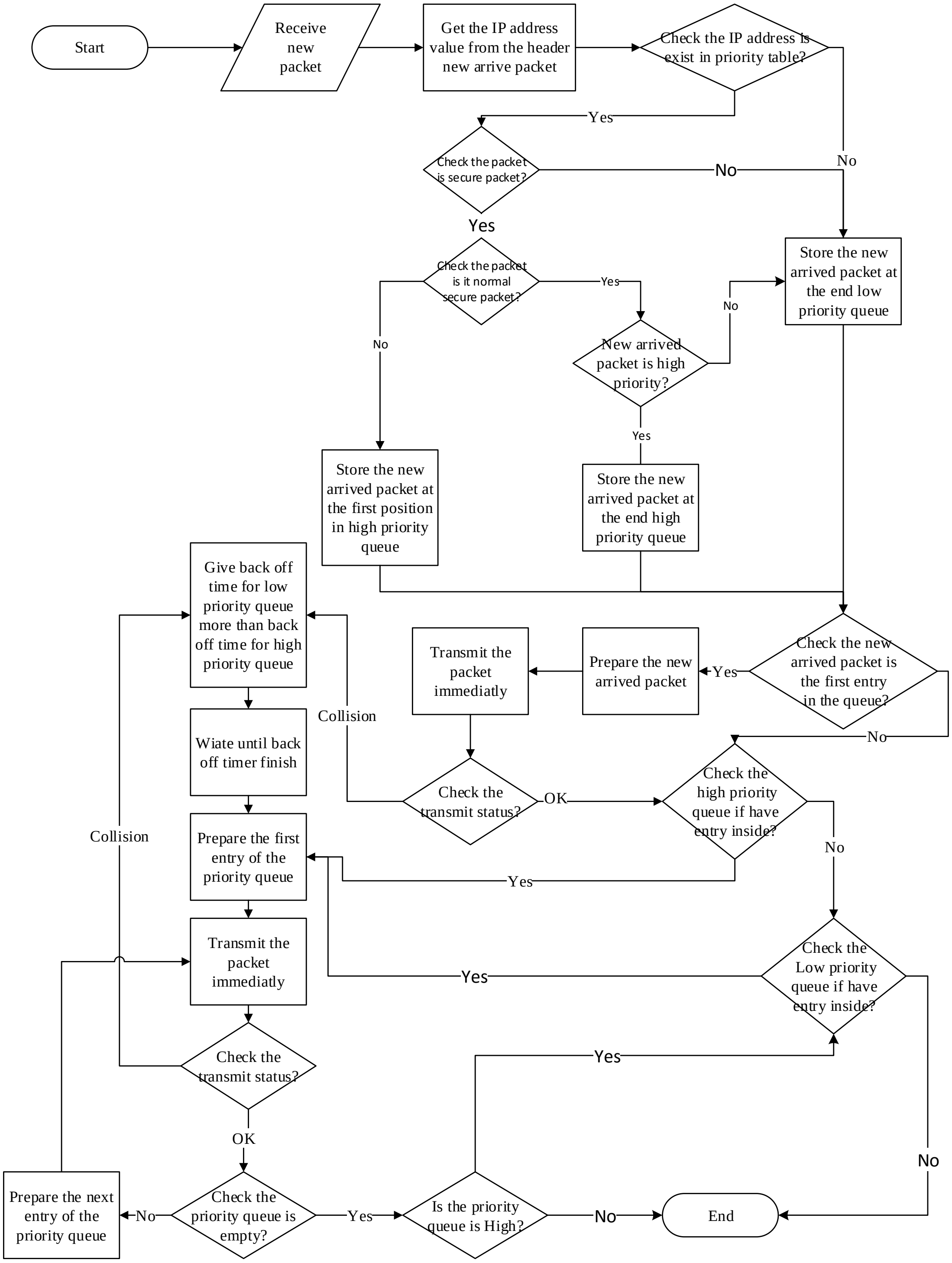

Start
Receive new packet
Get the IP address value from the header new arrive packet
Check the IP address is exist in priority table?
Yes
No
Check the packet is secure packet?
No
Yes
Check the packet is it normal secure packet?
Yes
No
Store the new arrived packet at the end low priority queue
No
New arrived packet is high priority?
Yes
Store the new arrived packet at the first position in high priority queue
Store the new arrived packet at the end high priority queue
Give back off time for low priority queue more than back off time for high priority queue
Transmit the packet immediatly
Prepare the new arrived packet
Yes
Check the new arrived packet is the first entry in the queue?
Collision
No
Wiate until back off timer finish
Check the transmit status?
OK
Check the high priority queue if have entry inside?
Collision
Prepare the first entry of the priority queue
Yes
No
Transmit the packet immediatly
Yes
Check the Low priority queue if have entry inside?
Check the transmit status?
Yes
OK
No
Prepare the next entry of the priority queue
No
Check the priority queue is empty?
Yes
Is the priority queue is High?
No
End

*Figure 4.5: STPD flow diagram*

# Chapter 5

## Performance Evaluation of STPD-MAC protocol

In this chapter, we report on the efficiency and effectiveness of our proposed STPD protocol throughout series of experiments carried out using the Cooja simulator as it is implemented in Contiki environment. Our simulation results are contrasted with that of the traditional CSMA/CA protocol. The simulation results demonstrated the superiority of our solution in improving the channel utilization, enhancing reliability and decreasing latency high priority traffic, and low priority traffic as well. The proposed STPD system is presented in chapter 4, where the design of our proposed protocol is detailed. It is to be stated that adaptive and high QoS for IoT networks are the main targets of our protocol. This is achieved via prioritization mechanism for high priority and secured traffic. CSMA/CA is the base with which we contrasted our results, as it is the main protocol in use by most wireless networks, which is contention-based. Remark that our proposed protocol is contention-based too, which makes the comparison quite fair. The comparison is done via assessing throughput, latency, and reliability metrics for both protocols.

## 5.1. Simulation environment

### 5.1.1. Contiki Operating System

Contiki Operating System [38] is a small, open source, highly portable multi-tasking operating system, based on Ubuntu Linux. Which was developed by the Swedish Institute of

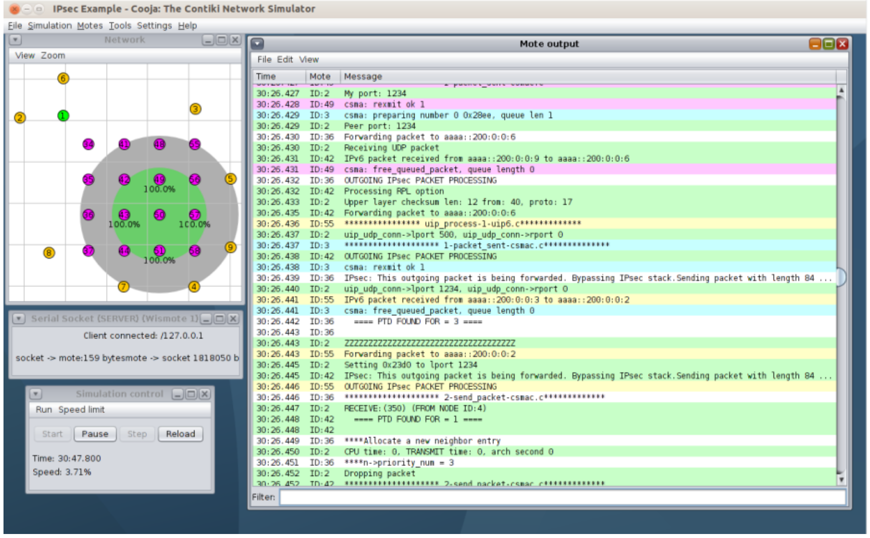
Computer Science, to be used in memory-constrained networked embedded systems and wireless sensor networks. Contiki connects tiny low-cost, low-power microcontrollers to the Internet. It is lightweight OS written in C, and has a built-in TCP/IP stack. Contiki only needs about 10 kilobytes of RAM and 30 kilobytes of ROM. The full functioning system, equipped with a graphical user interface, needs about 30 kilobytes of RAM. The OS is freely available under a BSD license. Contiki have three network mechanisms:

- uIP TCP/IP stack: support IPv4 networking
- uIPv6 stack: support IPv6 networking and the RPL routing protocol for low-power lossy IPv6 networks, and the 6LoWPAN header compression and adaptation layer for IEEE 802.15.4 links. We used the uIPv6 stack in our simulation.
- Rime stack: support lightweight protocols designed for low-power wireless networks.

Therefore, the implementation of 6LoWPAN in Contiki is based on RFC 4944 to transmit IPv6 Packets over IEEE 802.15.4 Networks. The Contiki OS consists of an event-driven kernel on top of which application programs are dynamically loaded and unloaded at runtime. Each program must start at least one process and only one process can be running and using the CPU at a time. It is the responsibility of each process to give up the execution and to prevent the entire system from deadlocking. Inter-process communication is done via posting events. Once one process needs to notify another process about something, it places a respective event to the event queue. The kernel goes over the queue and dispatches the event to the requested process (or to all running processes, if the event was broadcasted). It is also possible to pass data between the processes by posting an event together with the pointer to the data.

### 5.1.2. Cooja simulator

Cooja [39] is the Contiki network simulator. Cooja is a Java based simulator that allows large and small networks of Contiki nodes (sometimes called motes) to be simulated. Motes can be emulated at the hardware level. In that case the simulation is slower but it allows precise inspection of the system behavior. Otherwise the simulation is run at a less detailed level, which makes it faster. Running the simulation at less detailed level allows to simulate larger networks. Cooja is a significantly useful tool for Contiki developers, as it allows them to test their code and their systems long before implementing them in hardware. Developers regularly set up new simulations scenarios both to debug their software and to verify the behavior of their systems. The Cooja simulation environment is performed in an environment as the one depicted in Figure.5.1.



```
Client connected: /127.0.0.1

socket -> mote:159 bytesmote -> socket 1818050 b
```

```
Time: 30:47.800
Speed: 3.71%
```

```
30:26.427  ID:2   My port: 1234
30:26.428  ID:49  csma: rexmit ok 1
30:26.429  ID:3   csma: preparing number 0 0x28ee, queue len 1
30:26.429  ID:2   Peer port: 1234
30:26.430  ID:36  Forwarding packet to aaaa::200:0:0:6
30:26.430  ID:2   Receiving UDP packet
30:26.431  ID:42  IPv6 packet received from aaaa::200:0:0:9 to aaaa::200:0:0:6
30:26.431  ID:49  csma: free_queued_packet, queue length 0
30:26.432  ID:36  OUTGOING IPsec PACKET PROCESSING
30:26.432  ID:42  Processing RPL option
30:26.433  ID:2   Upper layer checksum len: 12 from: 40, proto: 17
30:26.435  ID:42  Forwarding packet to aaaa::200:0:0:6
30:26.436  ID:55  **************** uip_process-1-uip6.c*************
30:26.437  ID:2   uip_udp_conn->lport 500, uip_udp_conn->rport 0
30:26.437  ID:3   ******************** 1-packet_sent-csmac.c**************
30:26.438  ID:42  OUTGOING IPsec PACKET PROCESSING
30:26.438  ID:3   csma: rexmit ok 1
30:26.439  ID:36  IPsec: This outgoing packet is being forwarded. Bypassing IPsec stack.Sending packet with length 84 ...
30:26.440  ID:2   uip_udp_conn->lport 1234, uip_udp_conn->rport 0
30:26.441  ID:55  IPv6 packet received from aaaa::200:0:0:3 to aaaa::200:0:0:2
30:26.441  ID:3   csma: free_queued_packet, queue length 0
30:26.442  ID:36    ==== PTD FOUND FOR = 3 ====
30:26.443  ID:36
30:26.443  ID:2   ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
30:26.443  ID:55  Forwarding packet to aaaa::200:0:0:2
30:26.445  ID:2   Setting 0x23d0 to lport 1234
30:26.445  ID:42  IPsec: This outgoing packet is being forwarded. Bypassing IPsec stack.Sending packet with length 84 ...
30:26.446  ID:55  OUTGOING IPsec PACKET PROCESSING
30:26.446  ID:36  ******************** 2-send_packet-csmac.c*************
30:26.447  ID:2   RECEIVE:(350) (FROM NODE ID:4)
30:26.448  ID:42    ==== PTD FOUND FOR = 1 ====
30:26.448  ID:42
30:26.449  ID:36  ****Allocate a new neighbor entry
30:26.450  ID:2   CPU time: 0, TRANSMIT time: 0, arch second 0
30:26.451  ID:36  ****n->priority_num = 3
30:26.452  ID:2   Dropping packet
30:26.452  ID:42  ******************** 2-send_packet-csmac.c*************
```

*Figure 5.1: Cooja simulation graphical user interface*

## 5.2. Simulation scenario and parameters

In this section, we provide a detailed overview of the implementation parameters of STPD and of the simulation scenarios as well. Since our protocol is designed for IoT networks, we set up a scenario which simulates a secure end to end communication system, of course in a wireless environment. The IoT nodes are equipped with IPsec protocol. Consequently, integrity, confidentiality, and authentication of services are all guaranteed.

The STPD is distributed into adaptation layer and medium access control layer. Which our modification in the 6LoWPAN protocol in the Adaptation layer gives secured data traffic more importance and priority than other data traffic**.** Also treats some IoT devices with higher preference than others by setting up the packet priority class based on the priority table that distribute to all nodes in the network. Which allows the MAC layer to differentiate between high and low priority packets. This classification of the packets allows the MAC layer to provide the appropriate services to all data flows.

In our simulation scenario, STPD in the adaptation layer tags transmitted secure data packets as high or low based on the priority class that assigned to the sending devices in the priority table which compare the priority table entry with the data packet source address. The priority tags will be passing into MAC layer to run the packet prioritization mechanism. We assume each sending device has a fixed priority class. This tagging mechanism will allow packets generated by some devices to be delivered faster and more secured than other devices. For instance, healthcare IoT devices, will be tagged with high priority than devices that sensing the moves in a lobby or in a front building gate. Table 5.1 detailed the parameters used in both simulation cases.

*Table 5.1: Simulation settings*

| Parameter | Value |
|---|---|
| **Transmitting Range** | 60m |
| **Interference Range** | 100m |
| **Number Of Intermediate Nodes** | 9,16,25 and 36 nodes |
| **Number Of Sender Nodes** | 6 nodes |
| **Number Of Receiver Nodes** | 2 nodes |
| **Packets Rate** | 2.5 packets/sec |
| **Channel Type** | Wireless Channel |
| **Routing Protocol** | RPL |
| **Security Protocol** | IPsec |
| **Adaptive layer Protocol** | 6LoWPAN |
| **MAC layer Protocol** | CSMA/CA, STPD-MAC |
| **Radio duty cycling layer Protocol** | ContikiMAC |
| **Physical layer Protocol** | IEEE 802.15.4 |
| **Packet Size** | 127 bytes |
| **MAC Layer Header Size** | 25 bytes |
| **Payload Size** | 102 bytes |
| **MAC Layer Queue Size** | 8 packets |
| **Ratio Of High Priority To Low Priority Nodes** | 50-50% |
| **Queue Mechanisms** | Default queue, PQ (Priority Queuing) |
| **Service Types** | Default service (one queues to one transmission line), Priority service (several queues to one transmission line) |

### 5.2.1. MAC layer parameters (STPD-MAC protocol)

To distinguish between high, the low priority traffic, the back-off Time for low priority packets was set twice that of the high priority packets. In addition, Contiki by default, equal and static memory was allocated to the packets queue. The Memory Block Allocator (MEMB) used as the default library which is a block allocator that use a statically declared memory area to store objects of a fixed size. In our STPD-MAC protocol, Dynamic memory allocation was set to packets in the queues. This is needed in all nodes with limited memory and scarcity of resources. STPD implement Managed Memory Allocator (MMEM) library which enable dynamic allocations with automatic defragmentation by using pointer indirection.

In our simulation, 16 packets memory size can be allocated inside single node. The minimum number of queued packets is set to 4 packets/queue. And the maximum number of queued packets can be reach 12 packets. By default, we give each queue the minimum number of queued packets. This mechanism is useful to get over the issue of dropping packets when maximum number of transmission attempts is reached or the buffer is overflowed. Also it improves the resource utilization

## 5.3. Simulation results

As has been explained before, our proposed STPD, and the traditional CSMA/CA were simulated using the Cooja simulation engine. Exactly the same simulation environment was used in both cases. In Figure 5.2 the simulated network consisted of six IPsec sending nodes. To that network one border router was allocated to connect the entire IoT network with the Internet. As receivers, two IPsec nodes were set. Number of intermediate nodes was set as a variable to see how it impacts system overall performance. Nodes in the network use mesh topology as the mode for communication and packet exchange. As for transmission range, each node is made within the transmission range of four neighboring nodes. And we examined the comparative behavior of STPD and CSMA/CA under various number of intermediate devises (length between sender and receiver). Each simulation experiment was repeated 10 times for each scenario where different seeds were used. The results of the 10 trials were averaged and used as the final outcome of the experiment. MATLAB simulation environment, was used to further analyze the outcome of the simulation. In what follows we shall focus on the results of the simulation experiments in terms of channel utilization, average latency, and successful packet delivery ratio.

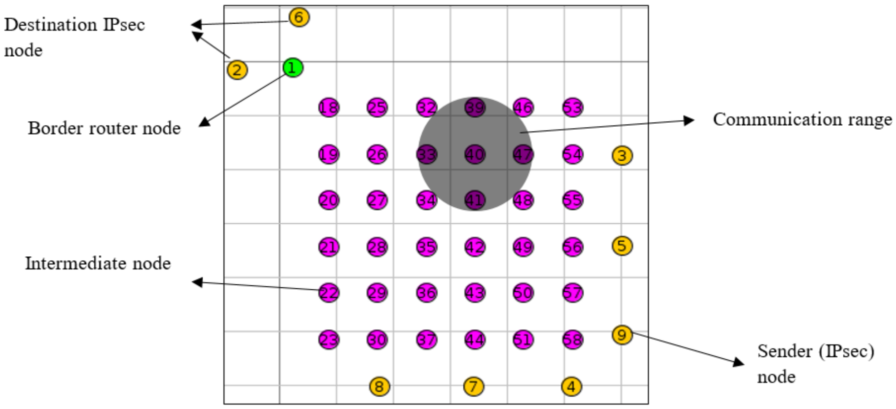


*Figure 5.2: scenario network topology for 36 intermediate nodes*

### 5.3.1. Transmission Channel Utilization

Channel utilization or throughput which is measured in packets/sec, is defined as the ratio of the packets successfully delivered to the destinations to the total number of packets sent in the direction of destination in a certain period of time. This metric indicates the effectiveness of a protocol in use by the network. Since secure communication applications requires high level of throughput, achieving high channel utilization is one of the primary goals of STPD.

The simulation results depicting the channel utilization parameters is shown in Figure. 5.3 And 5.4. The Figures compare STPD results with that of CSMA/CA. As is shown by the

Figures, STPD outperforms CSMA/CA in all simulated scenarios, mainly when the number of intermediate nodes is high. This clearly proves that STPD allows for high channel utilization than CSMA/CA regardless of the numbers of intermediate nodes that separate the sending from the receiving side. The enhancement can be referred to the use of back-off time which is shorter for HP than LP packets, which in turn enhances the contention decision. The priority classifier likewise plays a role in maximizing the channel utilization. Additionally, storing the control messages such as RTS / CTS and ACK inside the low priority queue contributed to that enhancement**.** Mathematically, it can be expressed as:

$$\text{Throughput} = \frac{\sum_{i=1}^{N} r_i \times ps}{T} \qquad (1)$$

Where r: number received data packet, T: time interval, ps: packet size, and N: number of the received messages

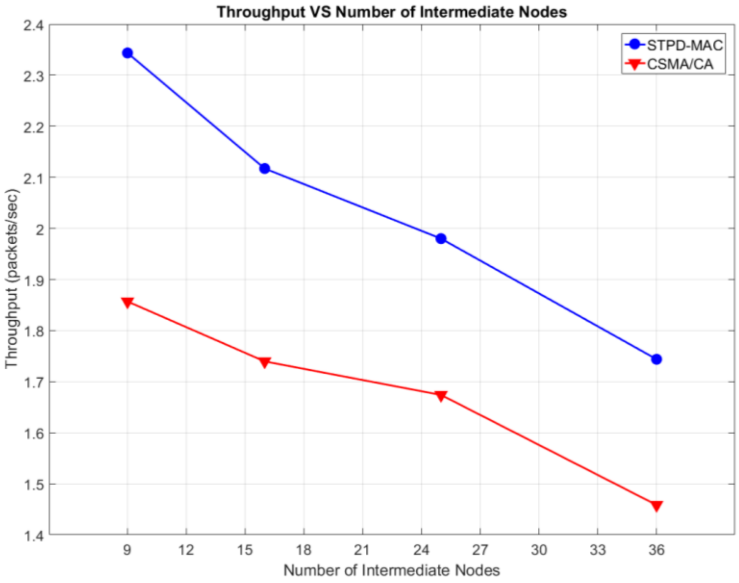


*Figure 5.3: Comparative channel utilization*

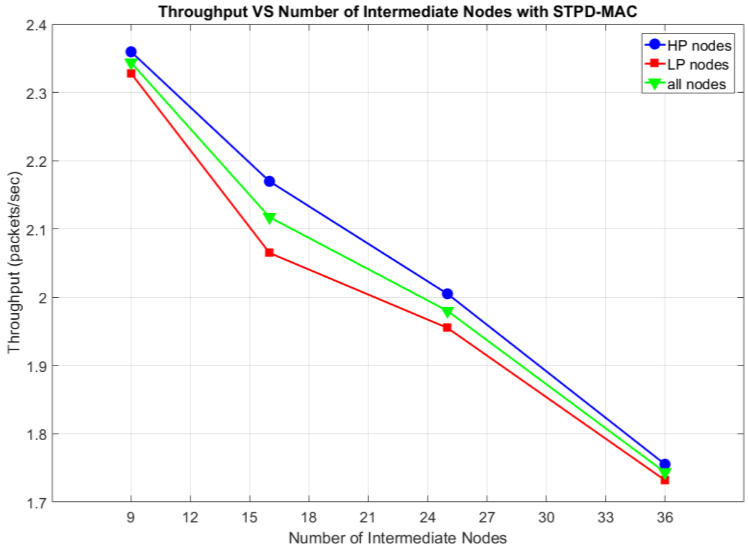


*Figure 5.4: Comparative channel utilization*

### 5.3.2. Delivered Packets Latency

Latency or average end-to-end delay which is measured in millisecond, comprises of processing delay, queuing delays, retransmission delay at the MAC, in addition to propagation and transmission delays. This metric measures the total delay time from source to a destination. As has been described in chapter 4, STPD aims at providing swift packets delivery especially for secure data application. The latency experienced by packets in our simulated systems is once more measured for STPD, and CSMA/CA. Simulation results were depicted in Figure 5.5.

Figure 5.5 shows the average end-to-end delay for secure traffic as measured for STPD and CSMA/CA. As was for channel utilization, STPD exhibited its superiority. Using STPD, the average latency of secured packets stays very low (≤ 1.3 s), thus demonstrates the efficiency of our arbitration and QoS mechanisms. At the same time, the average latency of CSMA/CA reaches 2.82 s. In almost all simulated scenarios, when the number of intermediate nodes is set to 9, the latency of CSMA/CA is estimated to be 1.7 s. however, when the number of intermediate nodes is set 36, the latency values rises to about 3 s. When the same experiment is repeated for STPD the latency value is increased by 1 s.  So the response to the increase in number of intermediate nodes is recorded as a higher latency time for CSMA/CA, than is for our proposed STPD.

Figure 5.6, shows the average end-to-end delay of high and low priority traffics for STPD protocol. HP packets reach the destination faster than LP packets. This is referred to that fact that the priority mechanism improves the channel utilization by way of letting the HP packets pass through the network shorter delay as compared with LP packets. This metric measures the total delay time from a sender to a destination. Mathematically, it can be expressed as:

$$\mathrm{E2ED} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{tr_i} - \mathrm{ts_i})}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{r_i}} \qquad (2)$$

Where tr: received time, ts: sent time, r: number received data packet, and N: number of the received messages

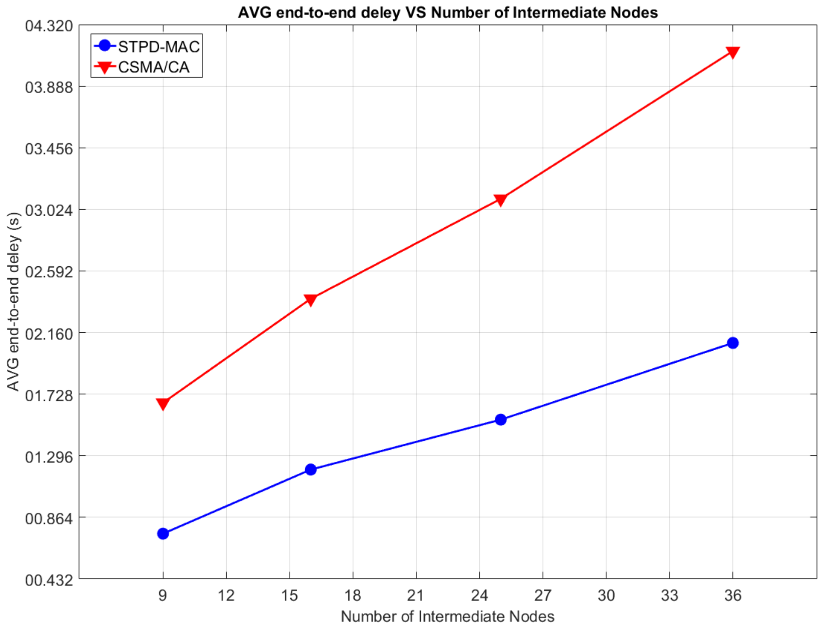


*Figure 5.5: Comparative average latency*

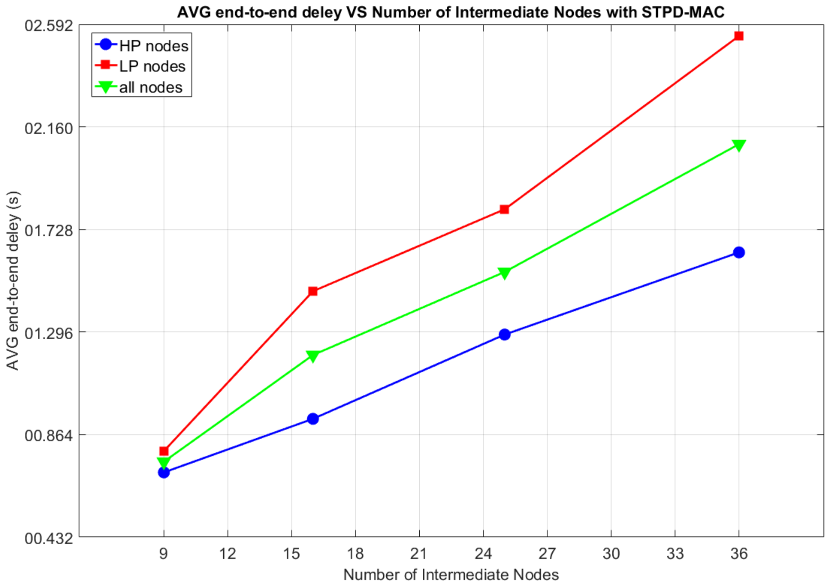


*Figure 5.6: Comparative average latency*

### 5.3.3. Packet delivery ratio (PDR)

Packet Delivery Ratio (PDR) is defined as the ratio of packets that are successfully delivered to a destination compared to the total number of packets sent out by the sender. This performance metric signifies the network reliability. Reliable data delivery is very critical for classified and secure communication applications.

Simulation results depicted in Figure 5.7 shows both STPD and CSMA/CA Packet Delivery Ratio for all types of tested traffic. Results evidently shows that STPD achieves better results. PDR for STPD traffic is recorded at 70% in the worst case scenario, and averaged around 85%. Yet, the PDR for CSMA/CA averaged around 65%, and 57% as the worst case scenario. This is referred to that fact that the STPD improves the channel utilization by scheduling mechanism that organize the channel access from different contention nodes which reduce the collision and packet loss, therefore letting the packets successfully delivered to the destination.

In Figure 5.8 the simulation results indicated that STPD even achieved effective results for the LP traffic, with a PDR of 85% the same level of reliability as it was for HP traffic, which approve that our STPD achieve farness among all traffic class in successful receive packets. PDR Mathematically, can be expressed as:

$$PDR = \frac{\sum_{i=1}^{N} r_i}{\sum_{i=1}^{N} s_i} \qquad (3)$$

Where r: number received data packets, s: number sent data packets, and N: number of the received messages.

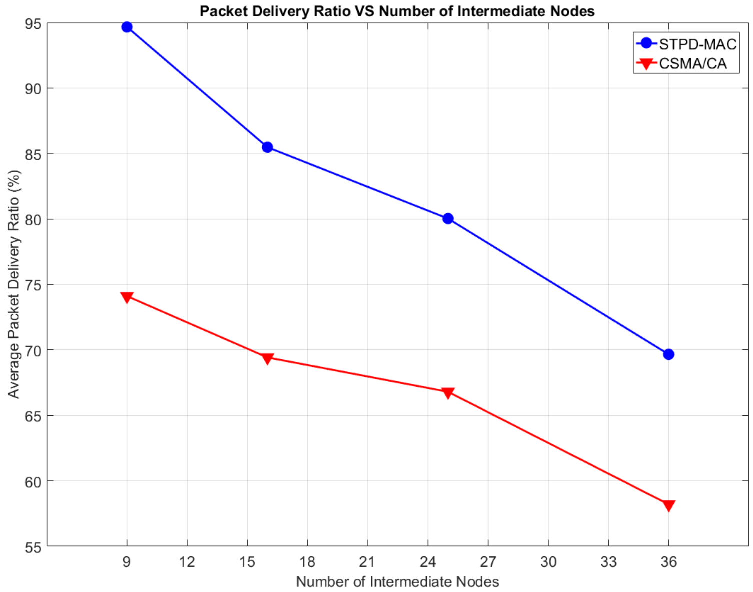


*Figure 5.7: Comparative successful packet delivery ratio of STPD*

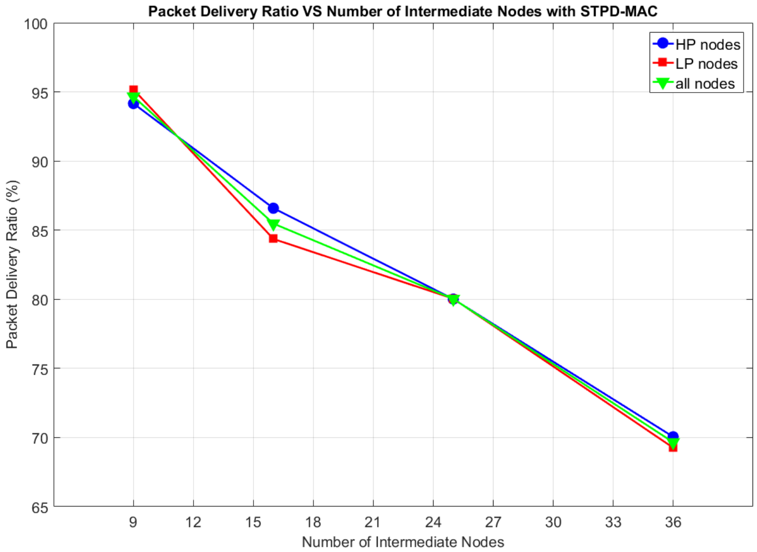


*Figure 5.8: Comparative successful packet delivery ratio of STPD*

Finally, the scheduling and prioritization algorithm of our STPD-MAC protocol is shown in the Algorithm 5.1.

```
new_packet;
packet_received= TRUE;
queue_size=8;
allocate_memory_for(HP_queue, queue_size/2);
allocate_memory_for(LP_queue, queue_size/2);
if (packet_received == TRUE) {
    priority_class  =  get_tag_priority_from_table_by_ipaddress(new_packet);
    // X=1 == HP, X=2 == LP, X='' == LP
    If(priority_class==1) {// This mean the packet is HP
        queue_type= HP_queue;
    } else {// This mean the packet is LP
        queue_type= LP_queue;
    }
    packet_type = get_type_secure_packet(new_packet);
    If(number_packet_queue(queue_type)<=(queue_size+ queue_size/2)) {
        // packet_type (initial_secure_packet, or normal_secure_packet)
        If(packet_type=="initial_secure_packet") {
            insert_into_front_queue (queue_type, new_packet);
        } else {
            insert_into_end_queue (queue_type, new_packet);
        }
        If (number_packet_queue (queue_type)! = 1) {
            // we prefer the HP_queue than LP_queue if LP_queue has more than one
            packet inside
            queue_type= HP_queue;
        }
        transmit_all_packet_in_queue(queue_type);
        if(collision_occur(queue_type) == TRUE) {
            give_backoff_time (queue_type, priority_class);
        } else {
            reset_queue_size(queue_type, queue_size);
        }
    } else {
        drop(new_packet); // the queued packet number reach to the maximum
    }
}
```

*Algorithm 5.1: Algorithm STPD procedure to determine arbitration.*

## 5.4.Discussion

### 5.4.1. Review of main results,

This thesis is based on and supports the idea that security and quality of service concepts for IoT networks must be taken up together. In order to develop a method for satisfying QoS and security requirements of IoT applications, related literature has been reviewed with a focus on studies proposing solutions for WSN and IoT service quality needs. Some studies focus only on the MAC layer and consider providing the required QoS, however, these studies do not take security into account, for instance as provided by the IPsec protocol.

This effort has come of fill in this gap. It tries to make a good tie between supporting secure connection, and providing QoS. The performance evaluation was reported in chapter 5 for three QoS parameters: as throughput, latency and packets delivery ratio. All performance measurements were taken using the IPsec protocol. The results of our simulation test appear to be very promising for our proposed protocol when compared with CSMA/CA protocol.

In order to advocate the argument that collisions have relatively infrequent occurrences compared to successful transmissions, we used Cooja simulation with Contiki OS. Figure 5.9 summarizes the results of the average number of collisions versus successful transmissions during different number of intermediate nodes scenarios. The increased in number of successful transmission when our proposed protocol is used is referred to the priority mechanism in our STPD protocol that achieves lower collision rate than CSMA/CA.

### 5.4.2. Explanation of results

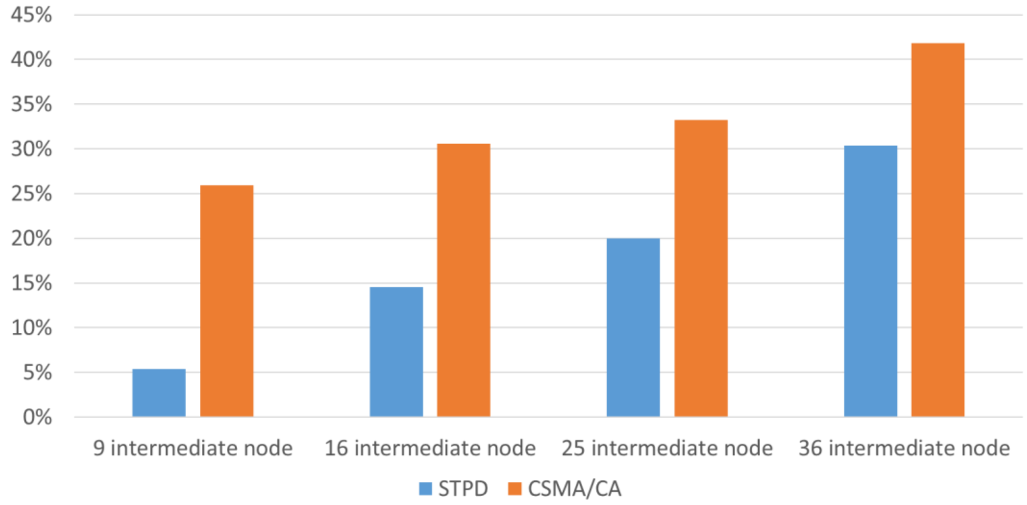


*Figure 5.9: Ratio of collisions to total transmission*

As is depicted by the Figure 5.9, the STPD achieves improvement of about 20% in collision rate, when the number of intermediate nodes is 9. This improvement, decreases to about 11% when the number of intermediate nodes goes up to be 36. This phenomenon can be explained by the fact that the degradation of improvement in STPD packet dropped ratio when the intermediate nodes reach 36, that indicate the number of hop node to reach the destination increase, and it increases the control messages between these nodes which cause more collision and reserve more memory place in the queue node.

In another experiment our proposed protocol was tested, HP against LP data flow. Table 5.2 summarizes the collisions and transmission attempts. The data presented in the table clearly shows a drop in the number of collisions for HP packets, over LP packets. This and the previous experiment can be referred to organizing the transmitting by priority mechanism and dynamic memory managements in the STPD keep the packets dropped ratio less than CSMA/CA.

*Table 5.2: Summary of Collisions and Transmission Attempts for HP and LP flow*

| STPD Protocol | Number of intermediate node | Number of successfully transmission | Number of packets collision | Ratio of collisions to total transmission |
|---|---|---|---|---|
| High priority flow | 9 | 1418 | 82 | 5.8433% |
| Low priority flow | 9 | 1399 | 101 | 4.8299% |
| High priority flow | 16 | 1304 | 196 | 13.413% |
| Low priority flow | 16 | 1241 | 259 | 15.6356% |
| High priority flow | 25 | 1205 | 295 | 19.9867% |
| Low priority flow | 25 | 1175 | 325 | 19.9591% |
| High priority flow | 36 | 1055 | 445 | 29.9469% |
| Low priority flow | 36 | 1041 | 459 | 30.7385% |

Table 5.3 presents the average delay for our STPD protocol as compared to CSMA/CA. When used CSMA/CA, packets experienced higher delay over our proposed protocol. The priority mechanism that organizes the access to the channel and the mechanism used to setup secure connection explain the results. If any message is used to setup secure connection failed, dropped, or delayed it causes significant delay which, the two communicating sides have to start over again before any secure data transmit.

*Table 5.3: Summary of End-to-End delay attempts for STPD and CSMA/CA flow*

| Protocols | Number of intermediate node | Average End-to-End delay |
|---|---|---|
| STPD | 9 | 00.750 s |
| CSMA/CA | 9 | 01.667 s |
| STPD | 16 | 01.199 s |
| CSMA/CA | 16 | 02.397 s |
| STPD | 25 | 01.549 s |
| CSMA/CA | 25 | 03.097 s |

| | | |
|---|---|---|
| STPD | 36 | 02.087 s |
| CSMA/CA | 36 | 04.133 s |

The delay experiment was applied to test the impact of our protocol on the data priority level, being high or low. HP packets reach the destination faster than LP packets. This is referred to that fact that the priority mechanism improves the channel utilization by way of letting the HP packets pass through the network shorter delay as compared with LP packets. See table 5.4.

*Table 5.4: Summary of End-to-End delay attempts for HP and LP flow*

| STPD Protocol | Number of intermediate node | Average End-to-End delay |
|---|---|---|
| High priority flow | 9 | 00.705 s |
| Low priority flow | 9 | 00.795 s |
| High priority flow | 16 | 00.930 s |
| Low priority flow | 16 | 01.468 s |
| High priority flow | 25 | 01.285 s |
| Low priority flow | 25 | 01.813 s |
| High priority flow | 36 | 01.631 s |
| Low priority flow | 36 | 02.543 s |

The overall impact of the protocol on the performance of the IoT network is tested through calculating the average throughput of the network again as a function of the number of intermediate nodes. In terms of bytes/sec, our proposed protocol is out performing CSMA/CA protocol by about 26% when the network has 9 intermediate nodes, and around 20% when intermediate nodes increased to 36. These findings can be explained, as the priority mechanism, dynamic memory management, and preference the IPsec security connection setup over other messages improve throughput in STPD than CSMA/CA. Table 5.5 explain the average throughput for STPD and CSMA/CA.

*Table 5.5: Summary of average throughput attempts for STPD and CSMA/CA flow*

| Protocols | Number of intermediate node | Average throughput (packets/sec) | Average throughput (byte/sec) |
|---|---|---|---|
| STPD | 9 | 2.3436 | 297.6372 |
| CSMA/CA | 9 | 1.8569 | 235.8263 |
| STPD | 16 | 2.1173 | 268.8971 |
| CSMA/CA | 16 | 1.7396 | 220.9292 |
| STPD | 25 | 1.98005 | 251.4663 |
| CSMA/CA | 25 | 1.67385 | 212.5789 |
| STPD | 36 | 1.74375 | 221.4562 |
| CSMA/CA | 36 | 1.4584 | 185.2168 |

Moreover, in Figure 5.10 we present the superiority STPD over the CSMA/CA in the number of attempts to create a secure connection when IPsec is used. If the number of attempts increases, this means that CPU cycles and execution time increased for re-preparing cryptographic algorithm, traffic encryption key, and parameters for the network data to be passed over the connection. As a result of these attempts, energy consumption by the network got increased, and latency too.

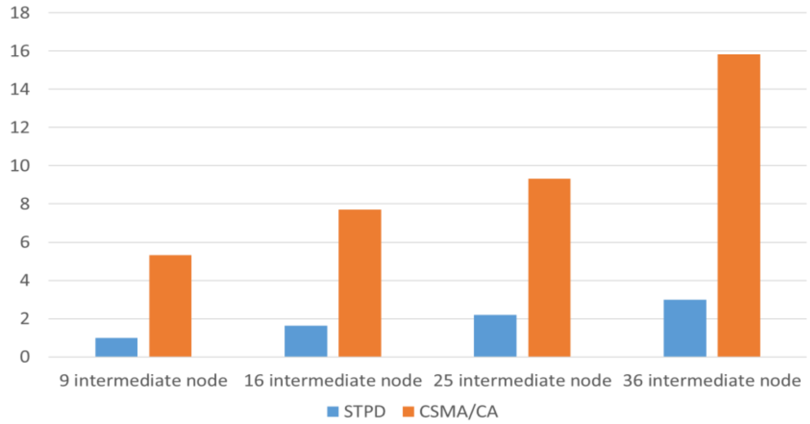


*Figure 5.10: Average number of attempts to create secure connection for IPsec protocol*

### 5.4.3. Comparison with results of other researchers,

Generally speaking, and up to our knowledge, none of the reported results in the literature were generated in the same settings as ours. However, others ideas to enhance QoS can be found in [35], [34], [33].

To see how well is our proposed STPD protocol compared with other suggestions, we contrast our results with other proposed MAC sub-layer priority protocol. Figure 5.11 depicts the percentage enhancement of our protocol with some other proposed mechanisms found in literature. Different QoS metrics such as latency, throughputs, and collision ratio were used.

The first protocol that we compare with is called Priority-Based Adaptive MAC (PA-MAC) [35] protocol proposed by Sabin Bhandari in 2016. PA-MAC is QoS-aware protocol, which allocates time slots dynamically, based on the traffic priority. The beacon channel (BC) in PA-MAC is used for the transmission and reception of beacon frames, while the data channel (DC) is used for the rest of the communication [35]. Prioritize the data traffic by using a priority-guaranteed procedure in the contention access period (CAP) [35]. The second protocol is TMP-MAC protocol [34] proposed by Muhammad Akbar in 2016. He proposed a tele-medicine protocol (TMP) using IEEE 802.15.4 slotted CSMA/CA with beacon enabled mode on the basis of a novel idea which combines two optimizations methods i.e., MAC layer parameter tuning optimization and duty cycle optimization [34]. Irfan Al-Anbagi propose DRX-MAC protocol [33], which aims to address the delay and service requirements of smart grids. DRX is based on delay-estimation and data-prioritization steps that are performed by the application layer, in addition to the MAC layer parameters responding to the delay requirements of the smart-grid application and the network condition [33].

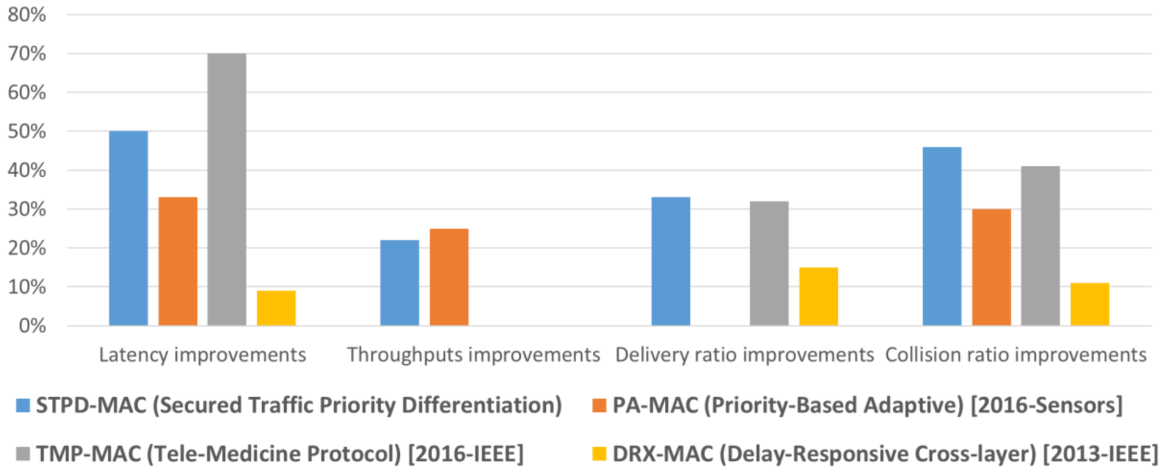


*Figure 5.11: percentage enhancement of studied protocol compared with MAC protocol CSMA/CA*

We can see in Figure 5.11 the superiority of our work than other protocols that proposed by other researchers. In case latency measure, PA-MAC protocol achieve 33% improvement than IEEE 805.15.4 MAC protocol. Latency improvements in PA-MAC protocol less than STPD, which return to the PA-MAC protocol cause end-to-end delay by passing the transmitted packet through coordinator node within the communication range of other nodes to reach the destination. Also the high collision ratio in PA-MAC result of guaranteed timeslots (GTS) number is limited, especially in case of heavy and high data rate traffic. The second protocol that we compare with is TMP-MAC protocol which achieve near result to our protocol, but still we have better performance than TMP-MAC protocol. The last protocol which called DRX-MAC has lower average percentage change than our protocol, which data prioritization depend on the application-layer to control the MAC sub-layer, and the estimated delay mechanism that use are the main reason to hinder the performance. In addition, all these protocols not working with security protocol like we did.

### 5.4.4. Limitation to research

The most important limitation in this study was lack of documentation for Contiki developer. The documentation only consists of source code comments and examples. There is also a serious shortage on tutorials. The unavailability and non-existing support of programs used for compiling Contiki platforms especial on Microsoft Windows operating system. Other limitation, when implementing the IPsec protocol into IoT node to support network security causes network scalability issue. As we know IoT nodes consist of constrained resources, that give ability to store security association (such as cryptographic algorithm, traffic encryption key, and parameters for the network data to be passed over the connection) for few number of node connection setup.

### 5.4.5. Practical implications

Our implementation can be very useful especially when the application is critical and needs some confidential data that are delivered successfully with minimum delay such as health care system, or military application. If we imagine some military institution has different type of IoT devices such as camera surveillance, movement detection etc. and the communication between these devices must be secure, no one can be listen to the communication traffic or manipulate with passing data. In addition, this network requires high QoS support. All of these demand need efficient protocol design. For this our STPD protocol come to fill this gap by supplying good secure connection and QoS support.

### 5.4.6. Future research

In future research we will try to investigate how to improve the scalability of the network which uses IPsec protocol with QoS support. We can start investigating to archive network scalability by studying the ability to use security server, that manage all security parameter for neighbor nodes and studying which best for this server: centralized or distributed.

# Chapter 6

## Conclusion

The main purpose of this thesis was to test the possibility of improving the quality of services provided by the data link layer to the IoT applications. Towards that end, the research efforts were designed around examining the implementation of a proposed protocol denoted as Secure Traffic Priority Differentiation, (STPD). The proposed solution is implemented into Adaptation layer and the MAC layer level, as the MAC layer is counted as the main factor for determining the overall network performance.

The enhancements introduced by the proposed solution were assessed using extensive simulation experiments. In the experiments three major network performance metric were tested; channel utilization, network latency, and packet delivery ratio.

As for the channel utilization, STPD outperforms CSMA/CA, which is used as a reference protocol, in all simulated scenarios. The simulation results were done as a function of intermediate nodes, and the results show that STPD is superior over CSMA/CA regardless of the number of intermediate nodes. This enhancement channel utilization, is referred to the use of back-off time which is shorter for HP than LP packets, which in turn enhances the contention decision, for transmitting packet.

As for the latency parameter, simulation results show great enhancement when STPD is used in contrast to CSMA/CA protocol. The average latency of secured packets stays quite low (≤ 1.3 s), thus demonstrates the efficiency of our arbitration and QoS mechanisms. In almost all simulated scenarios, when the number of intermediate nodes is set to 9, the latency of CSMA/CA is estimated to be 1.7 s. however, when the number of intermediate nodes is set 36, the latency values rises about 3 s. When the same experiment is repeated for STPD the latency value is increased by 1 s. So the response to the increase in number of intermediate nodes is recorded as a higher latency time for CSMA/CA, than is for our proposed STPD.

The third performance parameter that we looked at is the correct packet delivery ratio. Simulation results shows that STPD achieves better results in comparison to CSMA/CA. Packet Delivery Ration (PDR) for HP traffic is recorded at 70% in the worst case scenario, and averaged around 85%. Yet, the PDR for CSMA/CA averaged around 65% and 57% as the worst case scenario.

The results of the extensive simulation experiments, revealed the need for adaptive protocol that is able to provide an appropriate level of service to wide range of applications with different traffic types. Especially when IPsec protocol is used to secure data traffic. Our solution though proposing and implementing a STPD algorithm in the MAC layer, it improves the quality of service of IoT networks as a whole, and is applicable with different type of data traffic.

The work can be further extended with more simulations. One idea is to make the STPD more intelligent by improving the priority assignment to sending nodes. Another idea is to consider parameters in classifying the priority of senders such as power consumption, sender location, and the preference of the user. As a future work too, we may develop a mechanism to distribute the priority table into all nodes in the network to make our STPD algorithm scalable. Finally, testing our model in regards of other parameters like transmit rate and comparing the result with other protocols, are other good ideas for future research.

# Bibliography

[1] Nik Bessis and Ciprian Dobre, "Big Data and Internet of Things: A Roadmap for Smart Environments," *Springer,* 2014.

[2] Zach Shelby and Carsten Bormann, "6LoWPAN: The wireless embedded Internet," *Wiley. com,* vol. volume 43, 2011.

[3] J. Hui and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4 Based Networks," *RFC 6282,* Sept. 2011.

[4] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," *RFC 4301,* December 2005.

[5] S. Kent, "IP Authentication Header," *RFC 4302 ,* December 2005.

[6] S. Kent, "IP Encapsulating Security Payload (ESP)," *RFC 4303,* December 2005.

[7] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen, "Internet key exchange protocol version 2 (IKEv2)," *IETF, RFC5996,* Sep. 2010.

[8] C. E. a. F. A. I. Demirkol, "MAC protocols for wireless sensor networks: A survey," *IEEE Communications Magazine,* vol. vol. 44, no. no. 4, p. pp.115–121, Apr. 2006.

[9] F. A. Tobagi, "Analysis of a two-hop centralized packet radio network–part ii: Carrier sense multiple access," *IEEE Transaction on Communications,* vol. vol. 28, no. no. 2, p. pp. 208–216, Feb. 1980.

[10] S. Du, A. K. Saha, , and D. B. Johnson, "RMAC: A routing-enhanced dutycycle MAC protocol for wireless sensor networks," *Proc. of the 26th Annual IEEE Conf. on Computer Communications (INFOCOM),* p. pp.1478–1486, May 2007.

[11] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: A low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," *Proc. of the Ninth ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc),* pp. p. 53-62, May 2008.

[12] Wei Ye, John Heidemann, Deborah Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking,* vol. vol. 12, no. no. 3, p. pp.493–506, Jun. 2004.

[13] B. Klepec, A. Kos, "Performance of VoIP Applications in a Simple Differentiated Services Network Architecture," *International Conference on Trends in Communications,* pp. 214-217, 2001.

[14] Taj Rahman, Huansheng Ning, Haodi Ping, Zahid Mahmood, "DPCA: Data Prioritization and Capacity Assignment in Wireless Sensor Networks," *IEEE,* 2016.

[15] Navrati Saxena, Abhishek Roy, and Jitae Shin, "Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks," *Computer Networks,,* no. 52, p. 2532–2542, 2008.

[16] M. Aykut Yigitel, Ozlem Durmaz Incel, and Cem Erso, "Design and implementationof a QoS-aware MAC protocol for Wireless Multimedia Sensor Networks," *Computer Communications,* no. 34(16), p. 1991–2001, 2011.

[17] S. Kumar, V. S. Raghavan, and J. Deng, "Medium access control protocols for ad hoc wireless networks: a survey," *Ad Hoc Networks, ,* Vols. vol. 4,, p. pp. 326–358, 2006.

[18] T. van Dam and K. Langendoen, , "An adaptive energy-efficient MAC protocol for wireless sensor networks," *Proc. of the 1st Int'l Conf. on Embedded Networked Sensor Systems (SenSys),* p. pp. 171–180, 2003.

[19] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," *in Proc. IEEE INFOCOM,* vol. vol. 3, p. pp. 1567– 1576, 2002.

[20] S. Chatterjea, L. van Hoesel, and P. Havinga,, "AI-LMAC: an adaptive, information-centric and lightweight MAC protocol for wireless sensor networks," *Proc. of the Intelligent Sensors, Sensor Networks and Information Processing Conference, Dec,* p. pp. 381–388, 2004.

[21] G. P. Halkes and K. G. Langendoen, "Crankshaft: An energy-efficient MACprotocol for dense wireless sensor networks," *Proc. of the 4th European Conf. on Wireless Sensor Networks (EWSN),* p. pp. 228–244, 2007.

[22] W. Li, J.-B. Wei, and S. Wang, "An evolutionary-dynamic TDMA slot assignment protocol for ad hoc networks," *in Wireless Communications and Networking Conference (WCNC),* p. pp. 138–142, Mar., 11–15 2007.

[23] I. Rhee, A. Warrier, J. Min, and L. Zu, "DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks," *Proc. of ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc),* p. pp. 190–201, May, 22– 25 2006.

[24] A. Kanzaki, T. Hara, and S. Nishio, "An adaptive TDMA slot assignment protocol in ad hoc sensor networks," *Proc. ACM Symposium on Applied Computing (SAC),* p. pp. 1160–1165, Mar., 13–17 2005.

[25] Y. Kim, H. Shin, and H. Cha, , "Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks," *Proc. of the 7th Int'l Conf. on Information Processing in Sensor Networks (IPSN),* p. 53–63, 2008.

[26] Awan, I.; Younas, M.; Naveed, W., "Modelling QoS in IoT," *Network-Based Information Systems (NBiS),* no. 17th International Conference, pp. 99-105, 2014.

[27] M. Yu, S. J. Xiahou, and X. Y. Li, "A survey of studying on task scheduling mechanism for TinyOS," *International Conf.Wireless Commun.,,* p. 1–4, 2008.

[28] "TinyOS.," [Online]. Available: http://webs.cs.berkeley.edu/tos. [Accessed 28 11 2016].

[29] Levis, P. A., "TinyOS: an open operating system for wireless sensor," *International Conf. Mobile Data Manag,* p. 63, 2006.

[30] Adil A Sheikh ,Emad Felemban, Saleh Basalamah, "Priority-Based Routing Framework for Multimedia Delivery in Surveillance Networks," *MMEDIA 2014 : The Sixth International Conferences on Advances in Multimedia,* 2014.

[31] Tanmay Chaturvedi, Kai Lia, Chau Yuena, Abhishek Sharmab, Linglong Daic, Meng Zhang, "On the Design of MAC Protocol and Transmission Scheduling for Internet of Things," *SUTD-MIT International Design Center,* 2016.

[32] Thien D. Nguyen, Jamil Y. Khan, and Duy T. Ngo, "An Energy and QoS-Aware Packet Transmission Algorithm for IEEE 802.15.4 Networks," *IEEE 26th Annual International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC): MAC and Cross-Layer Design,* 2015.

[33] Irfan Al-Anbagi, Melike Erol-Kantarci, and Hussein T. Mouftah, "Priority- and Delay-Aware Medium Access for Wireless Sensor Networks in the Smart Grid," *IEEE,* no. 1932-8184, 2013.

[34] Muhammad Sajjad Akbar, Hongnian Yu, ShuangCang, "TMP: Tele-Medicine Protocol for Slotted 802.15.4 with Duty-Cycle Optimization in Wireless Body Area Sensor Networks," *IEEE,* no. 1558-1748, pp. 1-1, 28 December 2016.

[35] Sabin Bhandari and Sangman Moh, "A Priority-Based Adaptive MAC Protocol for Wireless Body Area Networks," *Sensors ,* no. 401, 2016.

[36] Saima Abdullah, Kun Yan, "A QoS Aware Message Scheduling Algorithm in Internet of Things Environmen," *IEEE Online Confer-ence on Green Communications (OnlineGreenComm),* 2013.

[37] S. Raza et al., "Securing communication in 6LoWPAN with compressed IPsec," *Proc. 7th IEEE Int. Conf. Distrib. Comput. Sens. Syst.,* p. pp. 1–8, Jun. 2011.

[38] "Contiki: The Open Source OS for the Internet of Things.," [Online]. Available: http://www.contiki-os.org/. [Accessed 25 1 2017].

[39] Thiemo Voigt, Fredrik Osterlind and Adam Dunkels , "Contiki COOJA Hands-on Crash Course: Session Notes," *Swedish Institute of Computer Science,* July 2009.

# Arabic abstract

## ملخص

من خلال التطورات الكثيرة والمتنوعة في مجال تكنولوجيا الإنترنت ظهر مفهوم إنترنت الأشياء (IoT) كتطبيق منتشر في مستقبل الإنترنت. إنترنت الأشياء: هو مصطلح يصف أن كل شيء يجب أن يكون متصلا بالإنترنت.

يعتبر إنترنت الأشياء واحدا من أهم الموضوعات البحثية في شبكات الحاسوب على وجه خاص، وفي تكنولوجيا المعلومات (IT) على وجه عام. هذا الموضوع في معناه الأساسي يظهر مجموعة واسعة من الفرص لخدمات جديدة وابتكارات جديدة. كما أنه يغطي مجموعة واسعة من التطبيقات الشخصية، والمؤسسية والصناعية والزراعية، إلى مجالات وطنية أو حتى دولية.

تقنية إنترنت الأشياء نمت بسرعة كبيرة خلال السنوات القليلة الماضية، منذ إطلاقه في عام 2006 وهذا النمو السريع في هذا المجال، والاهتمام المتزايد من الناس في تطبيقاته، ترك عددا كبيرا من القضايا دون حل، وخصوصا على مستوى الأبحاث الأكاديمية. وكذلك في واحدة من أكثر القضايا أهمية والتي تتعلق بنقل البيانات بشكل آمن وتصنيفها.

كثير من العلماء يذكر في التقارير عن نقاط الضعف في أمن الشبكات في إنترنت الأشياء كما ورثتها من معظم بروتوكولات الإنترنت التقليدية. تقع في هذه الأطروحة مسألة معالجة ضمان نقل البيانات في هذه الحالة على مستوى طبقة نقل البيانات. أحد المبادئ الأساسية التي بني عليها إنترنت الأشياء وهي الموارد المقيدة جنبا إلى جنب مع بنية معقدة من أجهزة الاستشعار، والتطبيقات، والاتصالات، وبروتوكولات الشبكات.

في هذا البحث، يتم إعطاء الاعتبار لتحسين جودة الخدمة (QoS) على مستوى طبقة MAC. اختيار مستوى طبقةMAC ينبع من حقيقة أنها هي المسؤولة عن إدارة الوصول إلى القناة اللاسلكية. ويتم حساب القناة اللاسلكية باعتبارها العنصر الرئيسي في الشبكة بأكملها والتي تسيطر على أداء النظام. عند تحليل جودة الخدمة (QoS)على بروتوكول MAC المستخدمة من قبل إنترنت الأشياء وهو CSMA / CA اقترحنا عدة أفكار لتعزيز نقل البيانات بطريقة آمنة في إنترنت الأشياء. يبدأ الحل المقترح من طبقة التطبيقات، حيث يتم تحديد مستوى الجودة المطلوب من الخدمات، بغض النظر عن التطبيق الذي في متناول اليد.

يتم تحليل أداء حل أو البروتوكول المقترح باستخدام بيئة المحاكاة الأكثر شهرة التي تستخدم في إنترنت الأشياء، ونظام التشغيل Contiki، جنبا إلى جنب مع جهاز محاكاة Cooja، الذي تم تصميمه خصيصا لأنظمة إنترنت الأشياء. الحل المقترح لدينا يستهدف على وجه التحديد تحسين جودة الخدمة (QoS) التي تدعم متطلبات أي تطبيق ويستخدم متطلبات طبقة MAC. كجزء من بيئة المحاكاة، ويستخدم بروتوكول IPsec لتوفير حركة مرور آمن.

الحل المقترح لدينا لتوفير حركة بيانات آمنه كما المرور الآمن وتمايز الأولوية (STPD)، والتي يمكن استخدامها بسهولة من قبل شبكات إنترنت الأشياء التي تواجه تحديا في توفير جودة الخدمة في حركة المرور بشكل آمن. نظامنا STPD المقترح هو نسخة معدلة من بروتوكول MAC مع جودة الخدمة التي تدعم شبكة إنترنت الأشياء غير المتجانسة. وSTPD هو مخطط متقدم للوصول إلى القناة ويستخدم آلية خلاف القاعدة التي تفضل ارتفاع حركة المرور المهمة.

STPD يتفوق على CSMA / CAفي جميع سيناريوهات المحاكاة، لا سيما عندما يكون عدد من العقد الوسيطة عالية. وحقق STPD تحسينات في استخدام قناة الإرسال مع متوسط حوالي 25٪. وعرضت STPD تفوقها بالتقليل من متوسط الحد الأدنى للتأخير من الحزم المأمنه نحو 85٪ من CSMA / CA. كما أن النتائج أشارت الى ان STPDحققت تحسنا كبيرا في تقليل التأخر في استخدام القناة، وفي فعالية النظام.

**كلمات البحث:** شبكات الاستشعار اللاسلكية، إنترنت الأشياء، وجودة الخدمة، ومراقبة الدخول المتوسطة، وحركة طبقMACالمرور آمن، أمن بروتوكول الإنترنت،