A Priority-based Fair Queuing (PFQ) Model for Wireless Healthcare System

Bana, S.

A PhD thesis awarded by the University of Westminster. #

A PRIORITY-BASED FAIR QUEUING (PFQ) MODEL FOR WIRELESS HEALTHCARE SYSTEM

SUHAIR ABDULMALIK BANA

A Thesis Submitted in Partial Fulfillment of the Requirements of the University of Westminster

For the Degree of Doctor of Philosophy

January 2020

Abstract

Healthcare is a very active research area, primarily due to the increase in the elderly population that leads to increasing number of emergency situations that require urgent actions. In recent years some of wireless networked medical devices were equipped with different sensors to measure and report on vital signs of patient remotely. The most important sensors are Heart Beat Rate (ECG), Pressure and Glucose sensors. However, the strict requirements and real-time nature of medical applications dictate the extreme importance and need for appropriate Quality of Service (QoS), fast and accurate delivery of a patient's measurements in reliable e-Health ecosystem.

As the elderly age and older adult population is increasing (65 years and above) due to the advancement in medicine and medical care in the last two decades; high QoS and reliable e-health ecosystem has become a major challenge in Healthcare especially for patients who require continuous monitoring and attention. Nevertheless, predictions have indicated that elderly population will be approximately 2 billion in developing countries by 2050 where availability of medical staff shall be unable to cope with this growth and emergency cases that need immediate intervention. On the other side, limitations in communication networks capacity, congestions and the humongous increase of devices, applications and IOT using the available communication networks add extra layer of challenges on E-health ecosystem such as time constraints, quality of measurements and signals reaching healthcare centres.

Hence this research has tackled the delay and jitter parameters in E-health M2M wireless communication and succeeded in reducing them in comparison to current available models. The novelty of this research has succeeded in developing a new Priority Queuing model ''Priority Based-Fair Queuing'' (PFQ) where a new priority level and concept of ''Patient's Health Record'' (PHR) has been developed and

integrated with the Priority Parameters (PP) values of each sensor to add a second level of priority. The results and data analysis performed on the PFQ model under different scenarios simulating real M2M E-health environment have revealed that the PFQ has outperformed the results obtained from simulating the widely used current models such as First in First Out (FIFO) and Weight Fair Queuing (WFQ).

PFQ model has improved transmission of ECG sensor data by decreasing delay and jitter in emergency cases by 83.32% and 75.88% respectively in comparison to FIFO and 46.65% and 60.13% with respect to WFQ model. Similarly, in pressure sensor the improvements were 82.41% and 71.5% and 68.43% and 73.36% in comparison to FIFO and WFQ respectively. Data transmission were also improved in the Glucose sensor by 80.85% and 64.7% and 92.1% and 83.17% in comparison to FIFO and WFQ respectively. However, non-emergency cases data transmission using PFQ model was negatively impacted and scored higher rates than FIFO and WFQ since PFQ tends to give higher priority to emergency cases.

Thus, a derivative from the PFQ model has been developed to create a new version namely "Priority Based-Fair Queuing-Tolerated Delay" (PFQ-TD) to balance the data transmission between emergency and non-emergency cases where tolerated delay in emergency cases has been considered. PFQ-TD has succeeded in balancing fairly this issue and reducing the total average delay and jitter of emergency and non-emergency cases in all sensors and keep them within the acceptable allowable standards. PFQ-TD has improved the overall average delay and jitter in emergency and non-emergency cases among all sensors by 41% and 84% respectively in comparison to PFQ model.

List of Figures

List of Tables

List of Abbreviations

3GPP – Third Generation Partnership Project

AP – Access Point

API – Application Programme Interface

RPM–Remote Patient Monitoring

AL– Assisted Living

ATM – Asynchronous Transfer Mode

CCTV – Close Circuit Television

CD – Control Device

CP – Control Point

DLNA – Digital Living Network Alliance

DMP – Digital Media Player

DSCL – Device Service Capabilities Layer

GSCL – Gateway Service Capability Layer

H2H – Human to Human communication

IOT – Internet of Things

IP – Internet Protocol

M2M – Machine To Machine communication

MAC – Media Access Control

NSCL –Network Service Capability Layer

PAN – Personal Area Network

PDN – Public Data Network

QoS – Quality of Service

QPH – Quality of service Policy holder

RAN – Radio Access Network

RFID – Radio Frequency Identity

TR – Technical Report

VoIP – Voice over Internet Protocol

WHO – World Health Organisation

WLAN – Wireless local Area Network

WMM – Wifi Multimedia

Table of Contents

Acknowledgement

To my Mentor Professor G. Terstvanszky whose support and guidance derived me to ultimate success. To my parents, husband and family who's their continues support and words of encouragement have pushed me for my limits and beyond. To my children, for being with me throughout the entire PhD program. Above all to the all mighty GOD and his kindness and mercy. I thank you all and value your support.

Declaration

I declare that the contents of this thesis are results of my personal research. Appropriate references have been given for works of other researchers cited.

Chapter 1: Introduction

As the Information and Communication Technology (ICT) is rapidly advancing, the healthcare sector has gained and benefited from the important role which the wireless networking plays in delivering sensitive and non-sensitive information to the healthcare players. It sets the base for using communication technologies in the healthcare environment and strive to provide assurance of the Quality of Service (QoS) for the healthcare community and various healthcare applications.

Healthcare organizations use various e-Health applications to run their business and to have the medical staff and the patient continuously connected using wireless technologies. Hence the patient's health status and records can be continuously monitored by sending the medical records to the medical staff via wireless sensors. Previously wired devices were connected to the patients to monitor and measure various health conditions such as heart rate, blood pressure etc. which limited the mobility of the patients and restrained the patient's movements in and out of the health care centers.

E-Health is a very active research area, primarily due to the increase in the elderly population. According to the report of American Heart Association, the survival chance of the patient who is experiencing ventricular fibrillation is 48% to 75% within the first 12 minutes [1]. The research areas in e-health is very wide. One of these is the improvement of the QoS and critical data handling. Gains in this research area will increase the reliability and dependability on the e-Health Remote Patient Monitoring (RPM) and Assist Living (AL) ecosystem to promote wide range usage of e-Health applications among patients.

1.1. Background

External services provided by hospitals or out of hospital services such as ambulances and medical relief evacuators can benefit from the wireless technology supporting the e-Health business and applications. However, big attention must be given to some factors that affect the Quality of Service (QoS), such as latency, confidentiality, jitter, privacy, availability, reliability, mobility, security, maintenance etc. using wireless technologies. The out of hospital applications works within the Body Area Networks (BAN) which supports the Machine-2-Machine (M2M) environment providing secured, high quality and efficient data transfer between medical devices and the data collectors.

Security measures are of a great importance when dealing with sensitive medical data. Therefore, some attributes were established to ensure safe data transmission. These attributes include: authenticity which ensures user's identity; authority ensuring user's level of authority to perform the requested operation; integrity also ensures that the data received is the same as that transmitted; and confidentiality ensures the data encryption so that the communication between users whenever seen by an outsider does not uncover the genuine substance of the correspondence [8].

As M2M network of e-Health involves communication among different heterogeneous devices within the wireless network and BAN therefore it is mandatory to standardize the communication protocols and frequencies to have reliable and safe data transmission. This research analyses the challenges facing the M2M in e-Health and focuses on two important attributes of QoS which are delay and jitter and how solutions can be adopted to achieve the minimum delay and jitter in emergency data transmission which is the research ultimate objective.

1.2. Motivation of this Research

My main motivation to start this research comes from my personal experience. My father could have been saved if proper M2M communication exists. He was 64 when he died living at home alone. He had heart stroke and no one knew about his emergency situation. Doctors said later that there was high possibility to save my father's live if he would have got proper intervention within the first 20 minutes after the heart attack. Therefore, I decided to tackle this issue and minimize the loss of humans' lives by advocating for the usage of the M2M and improving the QoS especially for elderly patients and emergency cases.

Increase in the elderly age and population is one of the significant characteristics of the 20^{th} and 21^{st} centuries due to the advancement in Medicine and Medical Care. In the recent two decades, the rapid increase in the elderly adult population (65 years and over) has proved to be a major challenge in Healthcare. With this increase in population, the number of patients requiring continuous monitoring has risen proportionally. By 2025, this number will be approximately 1.2 billion and will be 2 billion in developing countries by 2050, with 80% in this age group of 65 plus [77]. While in developed countries, elderly adults will constitute nearly 20% of the overall population according to the population reference bureau [78]. The growth in healthcare centers and availability of medical staff does not cope with growth of the elderly population or emergency cases that need immediate attention. Therefore, prompting the M2M technology and improving the QoS deemed essential in mitigating this critical situation.

Improving the healthcare centers operation system by promoting and using the wireless e-Heath M2M technology shall reduce the burden on the medical staff, healthcare overhead and operational costs. Thus, better healthcare service could be achieved.

One of the aims of this research is improving the data transmission QoS by looking at the situation from different angle. Currently all the data being transmitted is prioritized according to the sensor type, for example the heart beat sensor has the highest priority since it is the most critical and vital data that indicate the health condition of the patient However, in my research I will extend the priority parameters and criticality not only to the sensor type but also to the patient's health history and profile such as: age, gender, history and pregnancy etc. and will focus on prioritizing the data transmission based on the sensor criticality and patient profile.

As the increase of using smartphones, laptops, tablets or any ubiquitous device is raising exponentially in addition to the increase in the numbers of the medical and health applications, the need of researches and development in mobile health technology motivated me to work in this field [83].

1.3. Problem Statement and Scope of this Research

Wireless network technologies have advanced to a stage where they can enable a large variety of heterogeneous devices to be deployed and support medical applications. They have a capacity to form various sized networks such as Wireless Body Area Network (WBAN) and Wireless Personal Area Network (WPAN) to support healthcare applications addressing challenges of Internet of Things (IOT) and advanced communication technologies in Healthcare.

The stringent requirements and real-time nature of medical applications introduces the need for appropriate QoS provisioning in wireless medical networks. The fast and accurate delivery of a patient's measurements is an extremely important factor in reliable eHealth ecosystem specially in emergency or life threating situations. Accordingly, the problem statement could be itemized as follows:

1. e-Health systems should be totally reliable and efficient; therefore, a strict real-time and delay-intolerant data transmission are required due to the sensitive nature of e-Health systems and patient's critical and vital data specially in emergency and life-threatening situation.

2. e-health systems should also be reliable in terms of jitter. In order to ensure steady stream at emergency points, diminishing the variation in the delay of received packets (data) due to improper queuing, configuration errors or network congestion is deemed required.

3. Handling a large number of M2M devices connection in the same time were 50 billion devices is expected to be connected by the year 2020. Thus, the following problems shall arise:

    a. Generating data transmission congestion, which will inevitably increase delay, jitter, packet loss or service unavailability to M2M users.

    b. Overloading of Radio Access Network (RAN) and the Core Network (CN), will impact both M2M and non M2M connection.

The Scope of this Thesis is to improve the current e-Health QoS in terms of delay and jitter by defining the related attributes and using simulation techniques to create a model that shall improve the e-Health QoS under different scenarios.

1.4. Aims and Objectives of this Research

The novel aspect of this research is to develop and propose a Dynamic packet handling Priority Queue protocol ensuring a minimum delay and jitter in data transmission from critical sensors and emergency situations.

Delay is a performance characteristic and important design parameter of both computer and telecommunications networks. The delay indicates to what extent it takes for a bit of data to travel across the network from one node to another node or from one endpoint to the other endpoint.

The variation in the delay of received packets is called jitter. Packets are transferred in a continuous stream from the sending side with an evenly apart spaced packet. Due to the congestion in the network, improper queuing, or configuration errors, this steady stream can become lumpy, or the delay between each packet can vary instead of remaining constant.

This research aims to reduce delay and improve jitter during emergency data transmission in M2M e-health applications. A strict real time and delay intolerant data transmission of patients is highly required in emergency and life-threatening situations. Similarly, jitter should be minimized in order to avoid stream disruption and ensure smooth transmission.

In order to achieve the above presented aim this research has the following objectives:

1. To study the literature about the existing priority queuing models in M2M e-health applications.
2. To develop a new priority queuing model to decrease delay and improve jitter.
3. To design and simulate the new priority queuing model.

4. To evaluate the new priority queuing model performance and compare the results with the current priority queuing algorithms.

Although the main focus of the Thesis is to address emergency cases in M2M environment, but shall also consider the non-emergency cases delay and jitter attributes since the non-emergency cases constitute the majority of the signals or data being transmitted daily worldwide.

1.5. Methodology of this Research

This section gives and overview on the methodology used to achieve the objectives of this research. As illustrated in Figure 1.1, the Methodology framework is divided into six main perspectives:

1. Review related literatures on current priority queuing in M2M e-health applications.
2. Identify the problem statement of this research.
3. Develop the research objectives and aims.
4. Propose and develop new priority queueing model.
5. Build the new model and simulation program.
6. Data analysis and model performance evaluation.

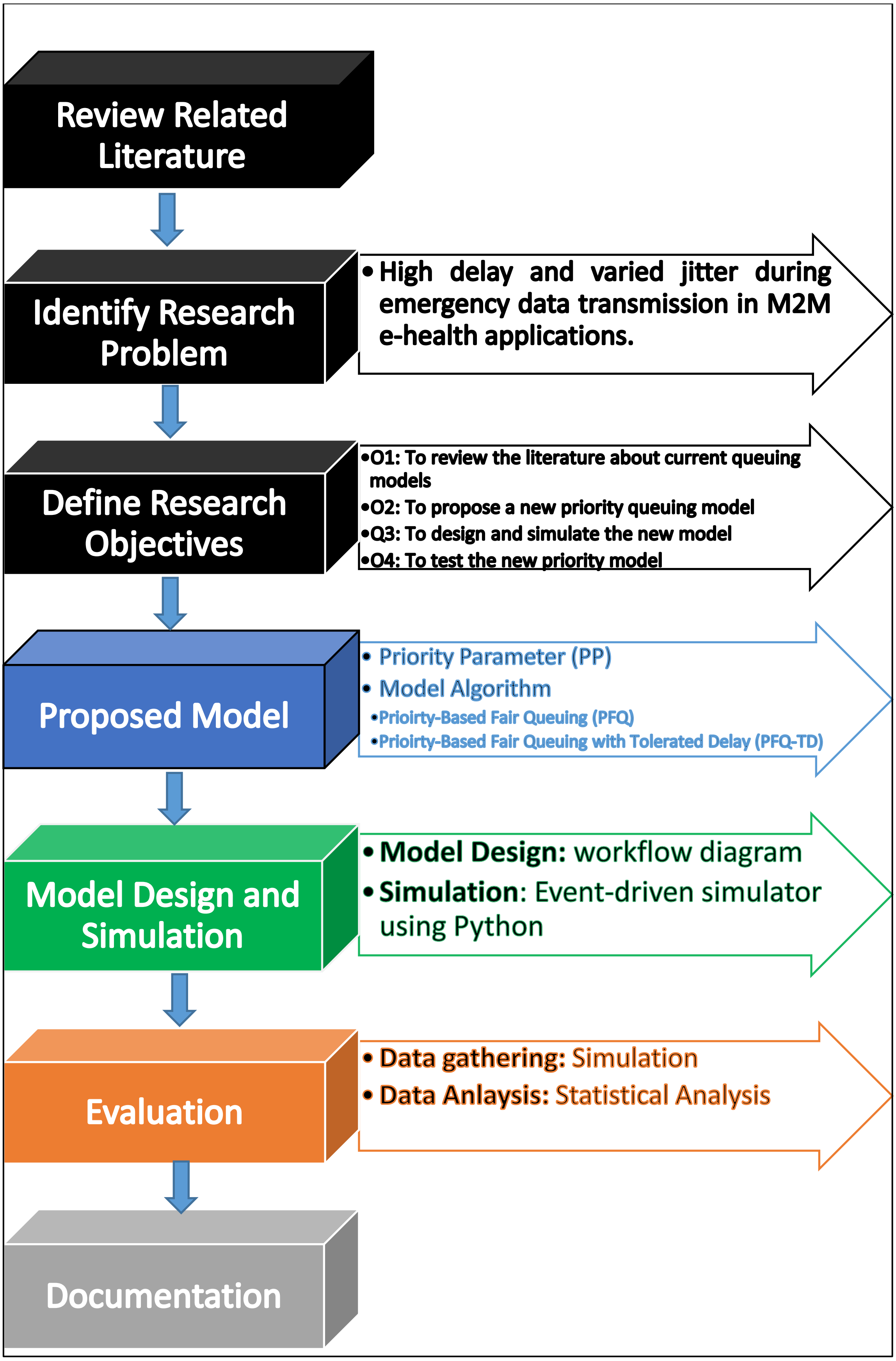


Figure 1.1. Methodology Framework

1.5.1. Review Related Literature

To achieve the first objective, a literature review was carried out on the current priority queuing models used in M2M e-health applications. The results from the investigation gave a clear understanding of the current priority queuing models used in M2M e-health applications, how they work in general, how they deal with emergency and non-emergency cases and their advantages, disadvantages, capabilities and limitation. This information gives better understanding to the current issues and helped in formulating the problem statement and objectives of the research.

1.5.2. Identify the Problem Statement

The problem statement is developed and identified based on the investigation done on the current priority queueing models used in M2M eHealth wireless application, the Wireless communication that support healthcare applications and the stringent requirements and real-time nature of medical applications relative to the QoS provisions of the wireless medical networks. As a result, it was apparent that the current priority queuing models and their techniques do not differentiate between real emergency and life-threatening situations that should be addressed immediately with fast and high-quality data transmission and normal or non-emergency situations.

1.5.3. Identify the Research Objectives and Aims

Hence, the aim of the research is set to develop new priority queueing models that differentiate between real emergency and non-emergency situations and deliver the patient’s vital measurements in fast and accurate fashion in real emergency and life threating situations by improving and decreasing the delay and jitter attributes.

1.5.4. Propose new Priority Queuing Model

To achieve the main objective of this research, it deemed necessary to propose and develop new queuing model to address the issue of high delay and varied jitter of emergency transmission in M2M eHealth applications.

The proposed model is composed of two parts, Priority Parameter (PP) and algorithm parts. PP is a value that is calculated at sensor node and is based on the sensor criticality and the Patient Health Record (PHR) where the algorithm part is implemented at the Local Data Processing Unit (LDPU). Accordingly, two models are proposed namely Priority-based Fair Queueing ‘’PFQ’’ and Priority-based Fair Queueing -Tolerated Delay (PFQ-TD) which addresses data delay and jitter attributes not only in real emergency situations but also in non-emergency situations.

1.5.5. Model Design and Simulation

The new model was developed to dynamically control the queuing priorities according to the QoS needed and to reduce delay and improve jitters under different scenarios and traffic conditions. The new model was represented using a workflow diagram before being simulated considering the models’ parameters and process in handling emergency and non-emergency packets based on data criticality i.e. sensor type and PHR profile.

The data criticality refers to the importance of the data to the patient’s life that helps prioritizing the transmission of the data. PHR is a unique identifier that has the patient current and previous health status and adds another level of prioritization after the sensor type criticality. PHR contains the patient’s personal information such as name, age, gender, address and the medical history such as previous heart stroke, high-risk pregnancy, diabetes, chronic disease etc. PHR can easily be viewed and updated from any device which a health care clinic, hospital or a doctor uses to store the patient's health history and treatment information.

The algorithm is implemented in the LDPU and simulated using Python language under various scenarios and traffic conditions to measure the effectiveness of the algorithm in reducing and improving the delay and jitter for emergency and non-emergency cases.

1.5.6. Data Analysis and Evaluation

The new model was tested against First In First Out (FIFO) and Weighted Fair queueing (WFQ) models to evaluate its effectiveness in terms of average delay and jitter. Further the new model was tested under various scenarios and traffic conditions to evaluate its performance in emergency and non-emergency situations.

1.6. Research Contribution

This research introduces two new queuing models. These models will improve the data transmission of emergency and non-emergency data. They reduce the delay and improve jitter associated with data transmission therefore ensuring high quality and reliable communication between the patient and the medical staff.

research contribution 1: Priority-based Fair Queuing (PFQ) model.

- It schedules the packets in Wireless Body Area Network (WBAN) for eHealth systems. It assures that all emergency data transmitted with minimal delay checking whether the packet is emergency or non-emergency one. Emergency packets are queued in the high-priority queue, whereas non-emergency packets are queued in the low-priority queue. Packets in the high-priority queue are always served first. Further, it manages transmission of the non-emergency data alternating between

PFQ and PFQ-TD algorithm to give fair priority distribution between emergency and non-emergency data transmission.

Research contribution 2: Priority-based Fair Queuing with Tolerated Delay (PFQ-TD) model.

- This algorithm adds the Tolerated Delay (TD) mechanism to Priority-based Fair Queuing algorithm. TD refers to the maximum delay that is acceptable for a packet to reach its destination. As a matter of fact, PFQ is expected to increase the delay and jitter of the non-emergency packets due to the high-priority assignment for emergency packets all the time. According to this algorithm emergency packets are not given priority over non-emergency packets to have fair distribution with balanced delay. However, TD considers the priority of the emergency cases over the non-emergency cases.

This research will ultimately benefit the elderly and emergency cases by providing solutions for better and reliable communication thus faster medical response can be achieved and the possibility of saving lives is increased as well.

Healthcare centers will benefit from this research since improving the M2M communication reliability and QoS will ultimately increase the number of patients using this affordable technology. Therefore, hospitals and healthcare centers can provide their services remotely and reduce the overhead and operational costs. Having more patients using M2M technology less patients shall visit the hospitals hence, will lead to increase the hospitals' efficiency and reduce the waiting time. Researchers may capitalize on the outcome of this research and develop further the eHealth QoS and improve the eHealth M2M communication.

1.7. Thesis Organization and Structure

The structure of this thesis is based on the University of Westminster guidelines and accepted format. This thesis contains 6 chapters plus supplements such as table of contents, list of tables and figures, acronyms, references and appendix.

In Chapter 1, The subject and the focus area of this thesis were introduced which is mainly improving the M2M QoS to ensure and provide better M2M service. A brief background on the development of the eHealth M2M ecosystem, followed by the motivation, problem statement, objectives, methodology, and research contribution were given. In Chapter 1, the background of the M2M ecosystem highlighted the significance of the wireless technology in improving the healthcare provided to the patients especially elderly patients. The Motivation behind this effort and research is spelled out where personnel, technical and social reasons are stated to highlight the significance of this topic from various perspectives. The problem statement and objective sections clearly define and specify the Delay and Jitter attributes as the targeted QoS parameters to be improved under different conditions and scenarios. The Methodology section described the techniques and procedures which were followed to achieve the objectives and resolve the problem statement. Priority Queuing algorithm development and Simulation under different conditions and scenarios are the two fundamental techniques used in this research to reach the objectives.

The contributions of this research are listed in Chapter 1. The outcome of this research and effort shall benefit the eHealth M2M users and medical centers. It shall also improve the medical services provided in the healthcare centers on the long run. Finally, the structure sequence and content of this thesis is described in this chapter to facilitate the navigation through its various chapters and sections.

Chapter 2 covers the literature review exploring all the works and topics related to the subject Thesis in order to have full understanding of the previous and current development in eHealth M2M and specifically in QoS. In this chapter, various topics that contribute to the eHealth M2M concept and data transfer such as M2M communication paradigm, wireless sensors types and usage, Wireless Sensors Networks (WSN) and Wireless Communication Technologies types, applications, technologies, Quality of Service (QoS), advantages and disadvantages etc were described. Also, related works and researches and the various priority queuing models being used in data transfer such as First in first out (FIFO), Round Robin (RR), and Fair Queuing (FQ) etc. were explained.

Chapter 3 includes the methodology and approach being followed to achieve the objectives of this research. A flow chart describes the development of new priority queuing model, simulation and testing under different conditions and scenarios is illustrated in this chapter. The Workflow chart presents the network architecture model and how the priority models or algorithms can be implemented. Also, the three selected Wireless Body Area Sensors namely; the ECG senor which monitors the heart rates and activities, Blood Pressure sensor and Glucose sensor that monitor that heart rate and the sugar level of the blood respectively are tabulated showing their different data rates and prioritization. Each sensor monitors one specific medical information and transmits its signal using ZigBee, to the Local Data Processing Unit (LDPU). As a hub, the LDPU collects all medical information form sensors, and store them temporarily in its buffer.

LDPU acts as a network regulator which is responsible to determine the allocation of transmission path, capacity and bandwidth among sensors during each time frame. It implements the proposed workflow. The LDPU decides its strategy based on its utility function, which is determined by the priority of the medical data and the transmission cost. The setup of the priority levels and emergent and non-emergent cases for each packet and sensor which this research is adopting are well defined and tabulated in this chapter.

The simulation evaluation outcome is the main topic of Chapter 4. A brief description on the Python language used to develop the algorithm is indicated. The two new developed priority queuing models namely 'Priority-based Fair Queuing' (PFQ) and 'Priority-based Fair Queuing and Tolerated Delay' (PFQ-TD) are also well described.

Based on the system setup, prioritization, emergent and non-emergent classification defined in earlier chapter, Chapter 4 spells out the results of the first and second simulation runs which prove the dynamic approach of the newly developed priority queuing models PFQ and PFQ-TD and their achievements in reducing the delay in jitter attributes compared to the levels recorded by implementing the current priority queuing models such as FIFO and WFQ for emergent and non-emergent cases. It proves that the flexibility given in alternating between the usage of the PFQ and PFQ-TD models to treat each packet adds value to the novelty of this research.

Chapter 5 outlines the outcomes of the simulation and results that were achieved to reduce the delay and jitter under different scenarios and conditions. The numbers indicated that in the first scenario PFQ superseded WFQ and FIFO and achieved lower delays and Jitter for Emergent cases since PFQ gave high priority for emergency cases all the time. However, the PFQ did not address any improvement in the non-Emergent delay or jitter data due to the high priority given to the emergency cases all the time.

In the second run of the first scenario the PFQ-TD algorithm has considered this drawback and did not give the Emergency cases high priority all the time by introducing the tolerated delay parameter. The PFQ-TD has considered the Finishing time of non-Emergent cases and gave it fair consideration over Emergent cases thus balancing the outcome and gave fair and balance delay and jitter distribution for emergent and non-emergent cases.

Finally, the conclusion and future work is outlined in Chapter 6. In conclusion the research work has succeeded in meeting the objectives of this research. It improved the eHealth M2M QoS Delay and Jitter attributes by developing a new priority queuing algorithms and classifying the data packets in a new way that mainly reduces the overall delay and jitter especially for true emergency cases and outperformed current priority queuing models. Capitalizing on the outcomes of this research, future work to develop and improve the eHealth M2M QoS attributes is highlighted under the Future Work section.

Chapter 2: Literature Review

Based on the aims of this research to reduce and improve the delay and jitter in Wireless Remote Patients Monitoring eHealth System. The main objective of this research is defined as to develop and propose a dynamic packet handling Priority Queue protocol ensuring a minimum delay and jitter in emergency data transmission from critical sensors and emergency situations. In this research, we used 3 different sensors; Heart Rate (ECG), Blood Pressure and Glucose sensors. Those sensors are attached to a number of patients with different health profiles and criticality living at their homes or elderly care centers. Each sensor sends readings continuously and direct from the patients to the medical staff and healthcare centres wirelessly. The data are prioritized according to its emergency and non-emergency situation, sensor criticality and patient health profile then sent accordingly. Critical and emergency classified data are sent first before non-critical and non-emergency readings to ensure minimum delay with respect to data transmission.

In general, the main focus of this research is to improve the data transmission QoS in eHealth ecosystem mainly the delay and jitter parameters in emergency and life-threatening cases by looking at the situation from different perspective in order to ensure steady stream at emergency cases, diminishing the variation in the delay of received packets (data) due to improper queuing, or configuration errors. Currently all the data being transmitted is prioritized according to the sensor type and criticality. However, in this research the priority parameters and criticality shall not be only limited to the sensor type but also to the patient's health history and profile such as: age, gender, history and pregnancy etc. Therefore, the focus on prioritizing the data transmission in this research shall be based on the sensor criticality and patient's profile.

Accordingly, new priority queuing model namely “Priority-based Fair queuing” (PFQ) is developed taking into consideration the priority parameters represented in sensor type and patient’s profile. Several scenarios and sub-scenarios have been designed to test the new PFQ model, under which the performance of the new model is compared to the current models being used in eHealth data transmission such as FIFO and WFQ models.

Three scenarios were carefully designed to stimulate the real environment condition which the data transmission is subjected to. The simulation is based on several scenarios of group of elderly patients living in a care center. Each patient has one or more sensor attached or embedded in his or her body namely Heart Rate (ECG), Blood Pressure and Glucose sensor which send vital data regularly to the health care center. Each sensor, monitors one specific medical information and transmits its signal using ZigBee to the Local Data Processing Unit (LDPU) where all medical information from sensors are collected and stored temporarily in its buffer.

Scenario one “Number of Patients” has four sub-scenarios where we start with 1 patient in sub-scenario 1, 4 patients in sub-scenario 2, 8 patients in sub-scenario 3 and finally 12 patients in sub-scenario 4. In scenario two “High Traffic Volume’’ the number of patients has been gradually increased from 1 to 4 to 8 to 12 to 20 to 30 to 40 to 50 patients across 8 sub-scenarios. Scenario three “Variant Emergency Rate” the data emergency rate has been increased gradually while the number of patients is set to 12. The data emergency rate is increased from 20% to 40% to 60% to 80% across 4 sub-scenarios rather than 30% emergency data and 70% non-emergency data rate used in scenarios 1 and 2.

Accordingly, in this chapter, the literature review and the search in previous studies is focused on the contest of this research. That’s to say all various topics that contribute to eHealth ecosystem, Machine to Machine (M2M) data transmission concept and data transfer in wireless communication sensors. Also, the wireless sensors types which are commonly used in M2M, wireless sensors networks (WSN) types, their applications and technologies. It also explained the Quality of Service (QoS) requirements and parameters in order to explore all options and common practices in the industry. Also, we highlighted the related works and researches and the various priority algorithms being used in data transfer such as First in first out (FIFO), Round Robin (RR), and Fair Queuing (FQ) to understand their functions, priority techniques, advantages and limitations. All of the above search work clarifies the overall attributes that affect the data transmission in wireless M2M eHealth ecosystem which in turn helps the development of the research and meet the set aims and objectives.

2.1 Overview of eHealth Systems

eHealth is a cost-effective way allowing communication technologies to support healthcare services. As shown in Figure 2.1 eHealth encompasses several eHealth applications such as Remote Patient Monitoring (RPM), clinical decision and support systems, electronics and personal health records.

Using the M2M communication platform improves the delivery of eHealth applications and services in various aspects by integrating the capabilities of the communication technologies in data sensing, data analysis and area networking using wired and / or wireless communication [7].

eHealth applications and services are of a great benefit to the patients and to the healthcare centers as it enhances the healthcare workers' productivity and involvement by monitoring the patient information and medical records from remote or inaccessible locations without having the patient to visit the healthcare centers. It also empowers the patients and healthcare workers by improving their participation in self-monitoring, chronic disease management and medication compliance monitoring [7]. This research program will focus on two important branches of eHealth Monitoring System: Remote Patient Monitoring (RPM) and Assist Living (AL) systems.

Both RPM and AL utilize the M2M as a communication platform. They include services such as post operation monitoring, chronic diseases management, preventive medicine, medication compliance and wellness and fitness programs to monitor patients based on vital signs and environmental data [7]. The key functional components of the RPM-AL are shown in Figure 2.2 below.

Within the RPM-AL ecosystem, data is transferred from the sensors linked to the patient to a gateway which serves as a link to the internet or external networks. Data transfer within a Local Area Network (LAN) or Body Area Network (BAN) could be continuous or time dependent then stored in a local gateway or uploaded to a medical information system located in a Wide Area Network (WAN). In the gateway, gathered data is put into the patient's medical file which allows the health professionals, such as medical doctors, nurses and emergency services to access the patient's medical file in accordance with privacy and security requirements [7].

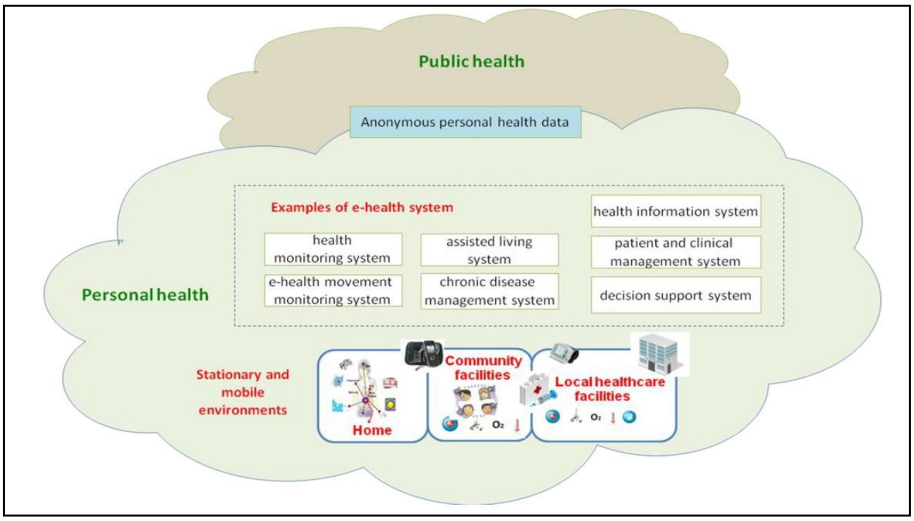


Figure 2.1. Overview of eHealth System

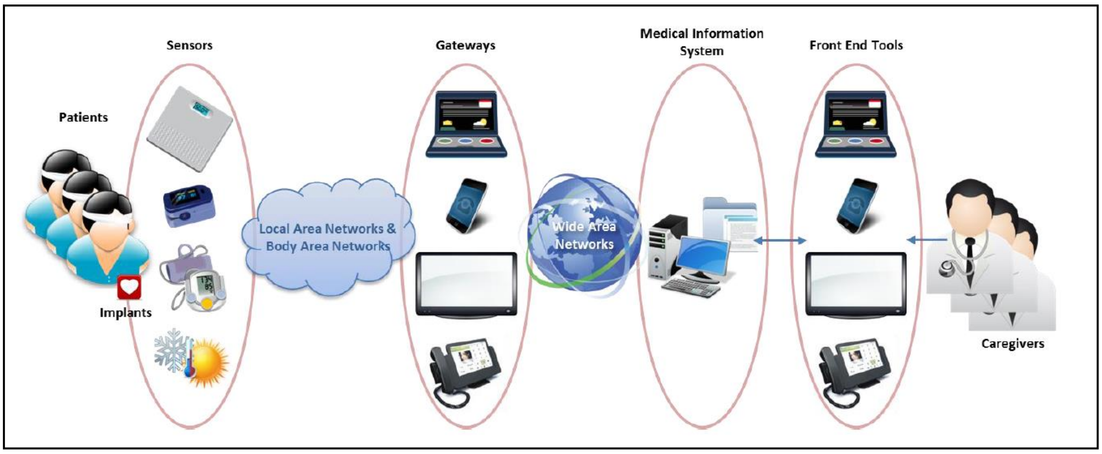


Figure 2.2. Key Functional Components in eHealth Ecosystem for RPM/AL

2.2 Sensors

A Sensor is a type of device that is able to respond to and detect a form of input which can be from a physical or an environmental condition. These types of conditions may include sensing medical conditions such as heart rate and pressure, heat, light, wind etc. which is then transmitted as an electrical signal to a controller or receiver for further processing. Transmission media for sensors can either be through wired cable or wireless and when the sensor is part of a network is referred to as a Node. There are numerous applications of wireless sensors today such as smart home, security systems, eHealth, environment monitoring etc [1].

Sensors play a crucial role in healthcare and the M2M industry even as research in these areas continue to rapidly grow. The wireless sensors provide various functionalities in a global environment that allows sensors to connect to internet anytime and anywhere [1].

eHealth means the use of digital information or electronic technology to access and monitor human health conditions. Sensors are a crucial part of eHealth as they produce electrical signal responsible for transmitting health data. Such data transmitted include temperature, blood pressure, heart beats etc. [1]. Figure 2.3 shows the basic working of sensor, the parameter being measured serves as the physical quantity (heat, movement etc.) while the electrical is the data being transmitted.

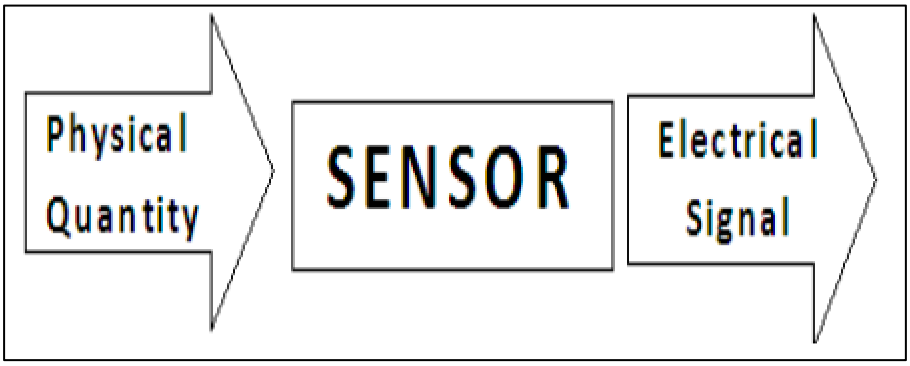


Figure 2.3. Basic Working of Sensors

2.2.1 Types of Sensors

The types of sensors can mainly be classified based on their wide range of their applications. Figure 2.4 shows the classification of sensors that are based on different criteria [2].

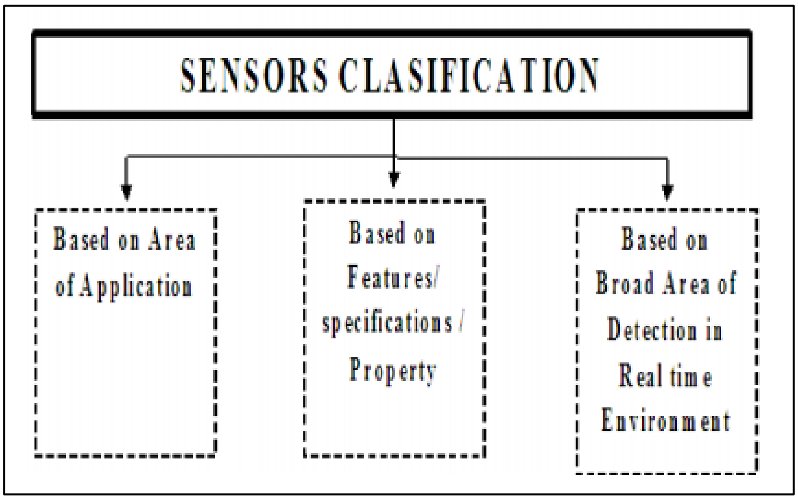


Figure 2.4. Sensors Classifications

According to [2], sensors are classified into three main categories:

1. Classification based on Area of Application.
2. Classification based on Features / Specifications / Property.

3. Classification based on broad area of detection in real-time environment.

In classification based on area of application, there are various real-time applications in which sensors play a very important role. These various applications are healthcare applications, military, manufacturing, space, aircraft, automation, customer electronics, etc. Also, in classification based on features / specifications / property, there are various factors that come across like accuracy, sensitivity, stability, environmental condition, range, calibration, resolution, cost, size, weight, repeatability, response time, linearity, etc [2].

Finally, with classification based on broad area of detection in real-time environment it can be divided into huge and vast areas as follows as found in [2]:

1. Acoustic, sound, vibration.
2. Automotive, transportation.
3. Chemical.
4. Electric current, electric potential, magnetic, radio.
5. Environment, weather, moisture, humidity.
6. Flow, fluid velocity.
7. Ionizing radiation, subatomic particles.
8. Navigation instruments.
9. Position, angle, displacement, distance, speed, acceleration.
10. Optical, light, imaging, photon.
11. Pressure.
12. Force, density, level.
13. Thermal, heat, temperature.
14. Proximity, presence.

2.3 Sensors Used in Healthcare Applications

Use of sensors in healthcare applications is to help minimizing the risk for patients. It is important to minimize the drain of disease treatment by focusing on prevention and early detection, with the help of sensors to sense various physical parameters. Various sensors used in healthcare applications as stated in [2] includes: Biosensors, Chemical sensors, Flow sensors. Fingerprint sensors, Force sensors, Heart rate sensor/ pulse rate sensors, Humidity sensors, Hour monitor sensor, IR sensors, Image sensors, Level sensors, Muscle sensors, Position sensors, Pressure sensors, Thermistor sensors and Temperature sensors, etc. Figure 2.5 shows examples of different medical sensors available in different devices today.

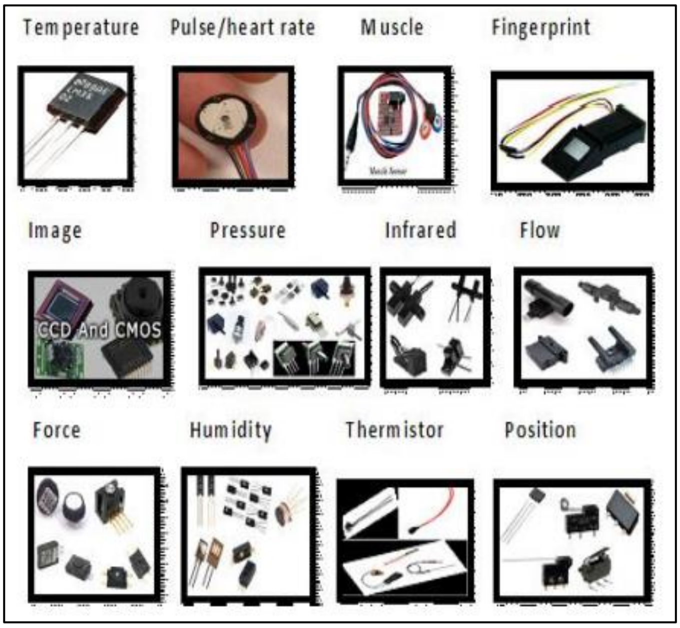


Figure 2.5. Sensors used in healthcare applications

On the other hand, Table 2.1 lists examples of medical sensors used in healthcare applications as found in [2].

Table 2.1: Sensors used in healthcare applications

Sensor Type	Data Type	Data Rate	Bandwidth
ECG	Blood pressure, Heart rate	71-288 KBPS	100-1000 Hz
Pulse oxi-meter	Blood oxygen saturation	16 bps	0-1 Hz
Gyroscope insulin actuator	Blood Glucose	1600 bps	0-50 Hz
Temperature sensor	Body temperature	120 npd	0-1 Hz
Accelerometer	Post-Operative monitoring Fall detection for elderly patents Parkinson's disease	35 kbps	0-500 HZ

These sensors enable health workers and doctors take proactive measures to provide better care to patients including preventative measures. From all of these sensors the temperature and pulse or heart rate sensors are the most common. Temperature and heart rate are regularly used to measure vital signs in the human body. Vital signs are the basic elements which can detect the health problems at very early stage [2]. Some of the sensors used to carry out these functions are described below:

2.3.1 Temperature Sensors

Factors like gender, levels of activity, food and the levels of fluid consumption or the time of day can affect the normal body temperature [2]. In women, for example, the stage at which they are in their menstrual cycle also affects the temperature of the body. Normal body temperature typically has a range between 97.8° F (Fahrenheit, equivalent to 36.5° C, or Celsius) to 99° F (37.2° C) for most healthy adults [2].

The human body acts as an external source that provides readings through the use of temperature sensors, these sensors convert it to forms that are readable by another device or a person for measurement [2]. There are two popular types of temperature sensors; contact sensors and non-contact sensors [2]. Contact sensors are sensors that need direct contact with the object or media that needs to be measured. They can be used to sense temperature of materials such as solid, liquid and gases over a wide range of temperatures. Non-contact sensors are the direct opposite, they do not require any form of physical contact with the material or object being measured [2]. They can be used to sense different non-reflective solid and liquids but are not used in sensing gases because they are naturally transparent [2]. The different types of temperature sensors include some of the following; thermocouple, resistance temperature detectors Thermistor, etc [2].

2.3.2 Pulse / Heart Rate Sensors

The pulse rate involves measuring the heart rate, or refers to the number of times the heart beats in a minute. The arteries contract and expand based on blood being pumped into it by the heart. Apart from measuring the heart rate, a pulse reading can also reveal the heart rhythm and strength of the pulse [2].

The pulse rate in a healthy regular adult has a range of 60 to 100 bpm (beats per minute). Differences in pulse rate may occur due to changes in certain conditions such as suffering from an injury, engaging in a form of exercise, poor health and emotions etc. Females from the ages of 12 and older in general, usually have faster heart rates when compared to males. Also, athletes such as runners (footballers, rugby players etc.) who do a lot of cardiovascular conditioning, may have heart rates as low as 40 bpm and encounter no issues [2].

When measuring the pulse rate of the heart, pulse sensors can be used to monitor the pulse rate of patients or users. An example is the Arduino sensor that is a plug and play device with open source monitoring showing the results of patients in real time [2]. As shown in Figure 2.6 the circuit consists of two sensors, very bright red LED and light detector. The LED has to be very bright because the light must pass through the finger to detect when blood flow from the heart. For this to happen, the finger becomes less transparent so that light gets to the light detector. For every pulse, the detector signal varies. This difference is changed to electrical pulse. This signal is amplified using an amplifier with outputs of +5 v logical level signal [2].

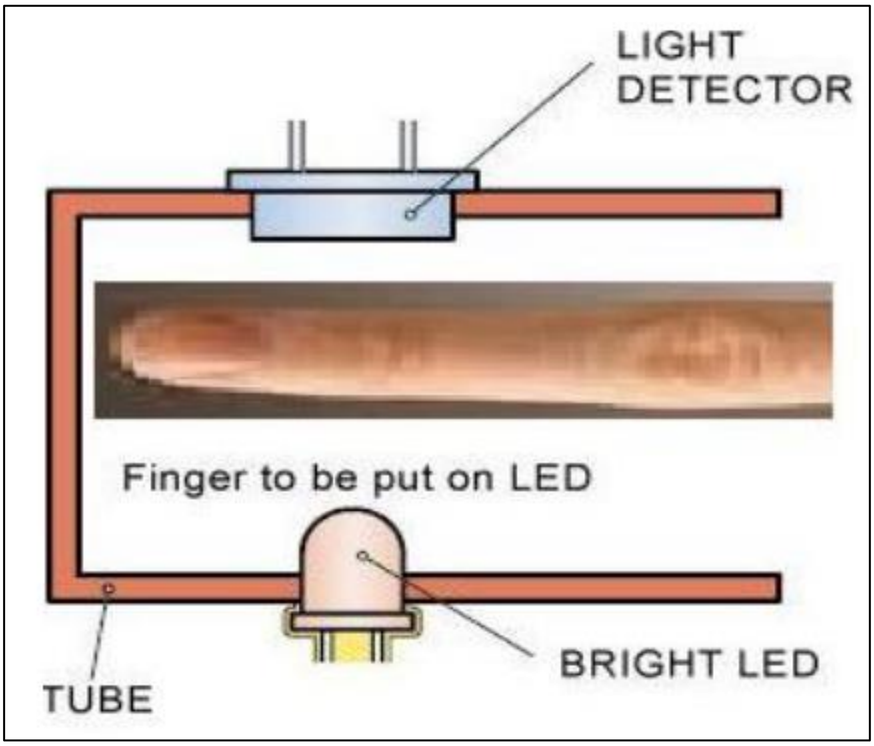


Figure 2.6. Heart Rate Sensor Construction

2.4 Wireless Sensor Network (WSN)

This is a collection of network devices which can be used to transmit the information collected from a field or environment being monitored with the use of a wireless medium. The data is sent across via multiple nodes, and with the presence of a gateway, other networks can share the same data. Figure 2.7 illustrates the Wireless Sensor Network varieties [1].

WSN is made up of base stations and numbers of end devices known as wireless sensors. WSN help monitoring environmental or physical conditions like temperature, sound, changes in pressure and send data via the network to a chosen main location.

The devices within the WSN help to monitor different environmental conditions, collecting and organizing data centrally. Examples of things it measures and detects include pressure, humidity, temperature, sound, direction and speed, levels of pollution, vibrations and host of similar conditions [1].

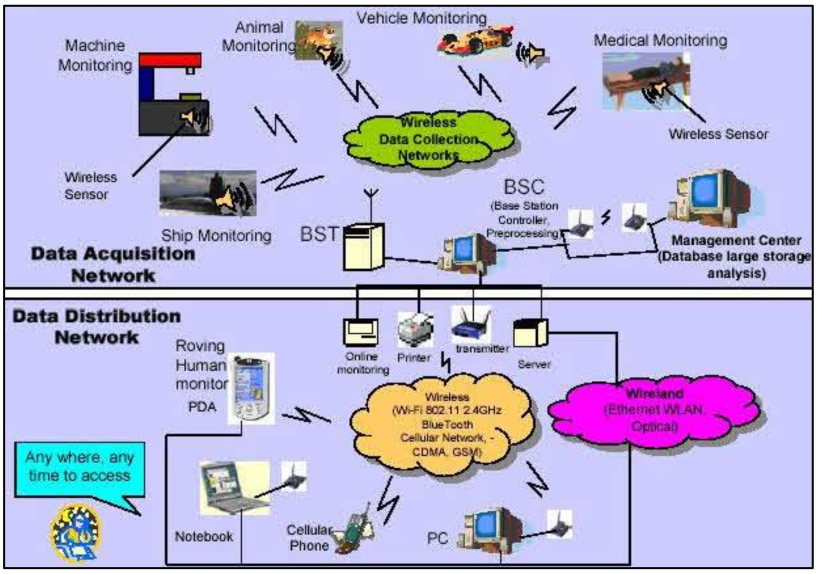


Figure 2.7. Wireless Sensor Networks Ecosystem

Each sensor network can contain numerous nodes which due to their size act as detection stations. A sensor node is made up of a sensor / transducer, a transceiver, a microcontroller and a form of power source. The transducer is responsible for sensing the physical condition and if a change in condition(s) is present, it generates electrical signals and sends it to the microcomputer to be processed. A central computer is used to forward commands to the transceiver which is then further sent back to that computer as shown in Figure 2.8 [1].

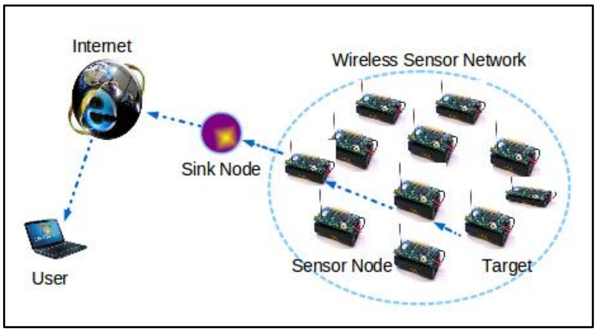


Figure 2.8. Wireless Sensor Networks Components

2.4.1 Wireless Sensor Networks (WSN) Topologies

There are four common sensor network topologies mainly Point to Point Network, Star Network, Tree Network and Mesh Network. Below paragraphs briefly describe the main aspects of these network topologies where more details can be found in reference [1] and [3].

Point to Point Network topology is one of the most common topologies and it has a one data communication medium that ensures communication paths are well secure. A central hub is not required in this design whereas participating nodes communicates directly with other participating nodes. Each node has the distinct ability to both act as a client and a server.

Unlike the Point to Point network, Star Network has a central node / hub performing the roles of a server (central hub) where it is not possible to bypass it for any form of communication amongst the nodes which in turn makes the nodes as clients.

In the Tree Network topology, the server (central hub) is referred to as the root node or the parent node. Other nodes are called leaf nodes (non-parent nodes) and data is sent from these nodes to the root node. One of the advantages of this design is that it does not need to consume as much power when compared to other topologies.

In the Mesh Network topology, all the nodes are interconnected so each node has the ability to send data without the need for a server or central node. The advantage of this design is that there is no single point of failure and if one node fails other nodes can still communicate easily, thus reliable. However, it consumes a lot of power and the structure can become complex.

2.4.2 Types of Wireless Sensor Networks (WSNs)

Wireless Sensors Networks (WSNs) have five common types which includes Terrestrial Network, Underground Network, Under Water Network, Multimedia Network and Mobile Network. Below paragraphs summarise the configuration of these networks and how they work where more details can be found in reference [1] and [3].

The Terrestrial WSNs are made up of hundreds and up to thousands of sensor nodes organized either in an unstructured (ad hoc i.e. random distribution) or in a structured manner i.e. intentional layout, optimal placement etc. In these networks, energy or power can be conserved by using low duty cycle operations, minimizing delays and implementing optimal routing.

The Underground WSNs types are made up of sensor nodes that are hidden in the ground to monitor underground conditions; whereas additional sink nodes are located above the ground to transmit information from the underground sensors to the base station. The issue with underground WSNs is that they are expensive, difficult to maintain, recharge and face high level of attenuation and signal loss.

The Under Water WSNs are made up of a number of sensor nodes and vehicles that can be positioned under water. Independent underwater vehicles can be used to collect information from such nodes. Some of the issues with having sensors underground include long delays in broadcasts and failures in sensors and bandwidth. Similar to underground sensors, the issue with underground WSNs is that they are expensive, difficult to maintain and recharge.

The Multimedia WSNs are made up of cheap sensor nodes fitted out with cameras and microphones. A wireless link is used to connect nodes in this network to aid compression, recovery and correlation of data. It can be used to monitor and track media related events such as audio, images and video. Problems the multimedia WSN includes high consumption of energy, requirement of high bandwidth, and the need of lot of data processing power and compressing techniques and high bandwidth for the contents to be delivered properly and easily.

The Mobile WSNs are made up sensor nodes capable of moving independently and can easily be networked within a given physical location to process, sense and transmit data. Mobile nodes are able to receive much improved network or bandwidth coverage, more efficient and have a superior channel capacity etc. On the other side, it possesses very little storage capacity and modest processing power, works in short communication range and its batteries have a finite life time.

2.5 Wireless Sensor Networks Applications

Wireless Sensor Networks are used in various applications; Figure 2.9 shows the wide range of applications where WSN are used. WSN can be used in eHealth applications to track and monitor the health of patients, environmental conditions, animals, floods and can also be deployed in forecasting changes to the weather. Commercially, they can be applied to predict when seismic activities may occur. In military purposes they are used to monitor and track intruders or unauthorised activities and events from a local or remote location. They can be used in home applications and transportation industry to help in monitor and control traffic, car parks etc [1].

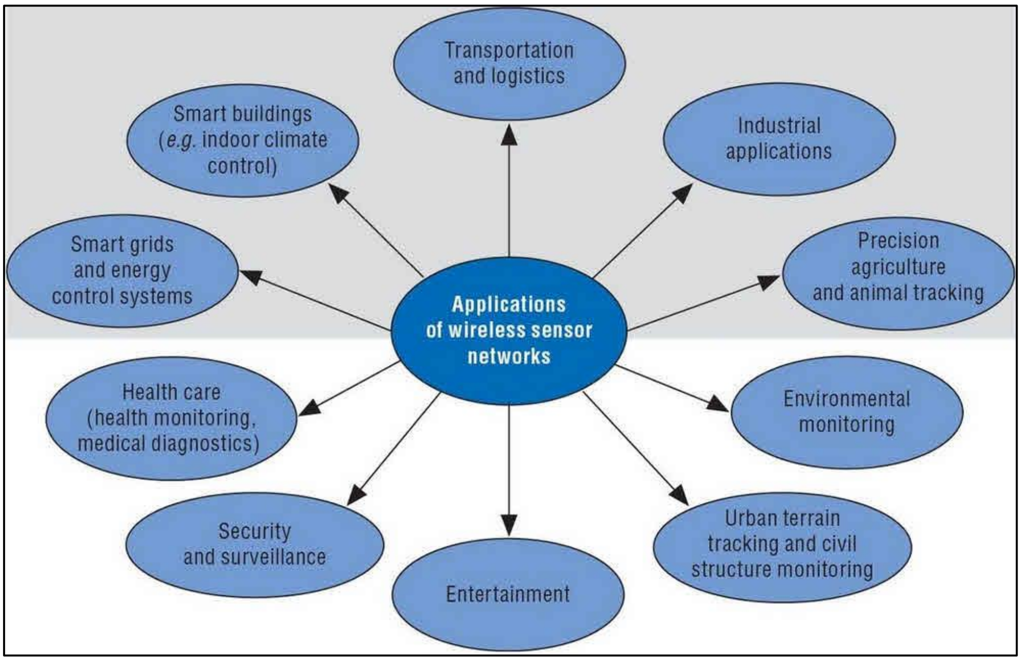


Figure 2.9. Wireless Sensor Networks Applications

In eHealth-related applications, integrated monitoring of patients can be achieved using WSNs. Movements and processes of the patients can be checked and used to provide diagnosis. Also, they help to monitor the dispensation of medicines in hospitals and help doctors to monitor their patients. For example, the 'artificial retina, which is designed to aid patients to detect light and object movement. They can also identify objects and count items [1] [3]. Sensors enable health workers and doctors take proactive measures to provide better care to patients including preventative measures. From all of these sensors the temperature and pulse or heart rate sensors are the most common. Temperature and heart rate are regularly used to measure vital signs in the human body. Vital signs are the basic elements which can detect the health problems at very early stage.

WSN are very useful in fire disaster conditions where sensor nodes are dropped from a distance like from an aircraft. The nodes then monitor the conditions sending data to a control centre where it is monitored and analysed to find solutions [3].

The self-organising and easy deployment feature of the WSNs make them very useful in military operations. They can be used for different sensory functions such as sense and monitor enemies or hostile movements, track authorised movements and can also help control stock (ammunitions, food and other equipment needed on the battlefield). It can also help to detect more dangerous attacks such as chemical, biological and nuclear attacks [3].

WSNs can be implemented in various environmental applications to help improve and enhance it. For example, they can track and record the development of certain creatures and record progress or changes noticed; check the make-up of the soils in various locations such as water content. They can also be used to enhance farming or different areas in agriculture. They can likewise be used for the detection of fire, flood, earthquakes, and chemical or biological outbreak etc [1] [3].

Technological advancement has also led to the introduction of such sensors and applications into the homes. For example, sensors can now be found in home appliances such as fridges, microwave ovens, security systems etc. These sensors help optimise performance and efficiency in these devices. With the help of WSNs the user can control devices locally and remotely [3].

2.6 Wireless Communication Technologies

The growth in Wireless Communication Technologies have made it more possible to easily establish communication from and to remote locations with the help of wireless enabled devices such as mobile devices and cordless telephones. Wireless Communication Technologies such as Satellite, Infra-red, Broadcast Radio, Wireless Microwave, Wi-Fi, Wi-Fi Direct, WiMAX, Fibre Optics, 4G Mobile and Bluetooth facilitate and create the media of which data can be transmitted wirelessly from point A to point B. Each technology has its own characteristics, advantages and drawbacks which eventually determine its best usage [4]. The following literature describes the most common wireless communication technologies and their characteristics. Additional details can be found in listed references.

Satellite communication is a self-contained wireless communication technology, widely distributed worldwide to enable users to stay connected almost anywhere on earth. When the signal is sent close to the satellite, the satellite amplifies the signal and returns it to the antenna receiver on the earth's surface. Satellite communication has two main components, the ground segment comprises of mobile or fixed transmission, reception and auxiliary equipment and the satellite segment. Satellite communication systems use satellites to communicate between two remote terrestrial locations; a terrestrial location and a mobile station (aircraft, ship, land vehicles etc.) or two mobile stations. Reference [4] elaborate on the communication ways that Satellite communication system operates.

Satellite communication makes it easier for mobile communications to be established by providing high bandwidth and wide coverage range. Additional receiving sites or network nodes can be relocated or added easily in few hours by introducing new ground equipment. Furthermore, Satellite is cheaper to use in remote areas where there is little or no communication infrastructure. On the contrary, Satellite communication has Delay in propagation, echo effects, susceptible to noise and interference, high setup cost and difficult to fix major faults in the space segment remotely [4].

Infrared wireless communication communicates information through Infra-Red (IR) radiation in a device or system. It is an electromagnetic energy that has a wavelength longer than red light. IR is usually used for short distance communications for example, security controls, remote controls etc. A photo LED transmitter and a photo diode receiver are required for successful infrared communication. The IR has a photoreceptor that captures and saves non-visible light (signals sent from LED transmitter). This transfers information between the source and the target. Mobile phones, TVs, security systems, laptops, etc. may be the source and destination [4].

Broadcast Radio is the first wireless communication technology which is still used widely today. Handy multi-channel radios allow a user to speak over short distances. Broadcast Radio enthusiasts share data and function emergency communication aids with their powerful broadcasting equipment throughout disasters, and can even communicate digital information across the radio frequency range. Broadcast Radio uses a transmitter that transmits data to a receiving antenna in the form of radio waves. Stations are associated with radio N/W's to broadcast common programming. Broadcasting can be carried out through cable, FM, network and satellites. The speed at which a broadcast is sent in AM/FM Radio over a long distance is up to two megabits/Sec (AM/FM Radio) [4].

Wireless Microwave communication is an efficient type of communication, mainly using radio waves and measuring the wavelengths of radio waves in centimeters. In this type of network, satellite and terrestrial methods are two ways in which data can be transmitted. Satellites are placed above earth and they orbit the earth from over 22,300 miles which allows them to be used to transmit data easily. However, they need to be able to communicate with other equipment (earth stations) in order to be able to transmit. They send data at a speed of 1 Mbps to 10 Mbps with signals of 11 GHz-14 GHz. With the terrestrial method, a direct line of sight is needed between communicating microwave towers. The disadvantage of this method is that obstacles between devices can interfere with signals and transmission. Also, they are easily affected by rain and bad weather conditions [4].

Wi-Fi is a low-power wireless communication system used by different electronic devices such as smartphones, laptops, etc. In this setup, a router works wirelessly as a communication hub. These networks enable users to connect to the router only in close proximity. Wi-Fi is very common in networking applications that offer wireless portability. These networks must be protected with passwords for security purposes, otherwise they will be accessed by others. Wi-Fi's main advantage is that it is compatible with almost all operating systems, game devices and advanced printers [4].

Wireless connectivity, often referred to as Wi-Fi, is the technology that enables a PC, laptop, mobile phone or tablet to connect to the Internet at high speed without a physical wired connection. Communicating devices need to be Wi-Fi enabled as radio waves are needed to transmit information between them or between a Wi-Fi device and the internet. The networking standards of 802.11 varies depending on the needs of the user; the 802.11a transmits data in 5 GHz frequency where data can be sent up to 54 megabits per second and 2.4 GHz frequency at 11 megabits per second in 802.11b. The 802.11g transmits data at 2.4 GHz but transmits up to 54 megabits of data per second because it also uses an OFDM code and the more advanced 802.11n can transmit up to 140 megabits of data per second using 5 GHz [4].

Wi-Fi wireless communication users are constantly connected to the network easily overcoming physical obstacles such as buildings and doesn't have the complexities and high cost of wired connections and maintenance. However, like any transmission of radio frequencies, wireless networking signals are subject to a wide range of interference and complex propagation effects beyond the control of the network administrator. Also, the most commonly used encryption methods have weaknesses that can be compromised by a dedicated adversary. Lastly but not last, the typical range of an 802.11g common network with standard equipment is tens of meters in length. Although it is enough for a typical home, it is not enough in a larger structure. To obtain an additional range, you must purchase repeaters or additional access points. Costs for these items can quickly add up [4].

Wi-Fi Direct is another network type that allows devices to connect directly and makes it easy and convenient to print, share, sync and display things. The devices enabled by Wi-Fi Direct can connect to each other without a traditional Wi-Fi network. Mobile phones, cameras, printers, PCs and gaming devices connect directly to each other in order to quickly and easily transfer content and share applications. Devices can connect to one device or a group of multiple devices can connect at the same time [5] [6]. Wi-Fi Direct device connections can be made anywhere, anytime, even if Wi-Fi network access is not available. Wi- Fi Direct devices send a signal to other devices in the area, informing them that they can connect. Users can view and request a connection to available devices or can be invited to connect to another device. When two or more Wi-Fi direct certified devices are directly connected, they form a Wi- Fi direct group using Wi- Fi protected configuration [5].

Wi-Fi Direct can establish a connection or transmission with other devices even if only one of the communicating devices is Wi-Fi Direct enabled. Routers are not needed to manage connections or traffic. Also, it has Wi-Fi Protected Setup (WPS) and is security compliant as it uses WPA2 security. Participating devices have equal network rights (i.e. does not assign a root device) and provides a speed of up to 250Mbps. Finally, Wi-Fi Direct is a portable technology that has been certified and participating devices do not require internet communication to establish a connection or exchange information [5] [6]. On the other side Wi-Fi Direct is still a new technology and so lots of bugs exist on enabled devices at the moment and not currently supported on all platforms or device types. It is also bounded by a range, when a user leaves the range the connection drops.

WiMAX (Worldwide Interoperability for Microwave Access) is a 4^{th} generation technology that has similar features with other 4G wireless systems such as LTE. It offers very significant improvements in throughput over wireless access technologies that are already in place [7]. The Electrical and Electronics Engineers Institute (IEEE) is in charge of defining the protocols used with the 802.16 extension set for WiMAX technology [7] [8]. Commercially, there are two versions of WiMAX; fixed and mobile. Fixed WiMAX is based on the IEEE 802.16d standard and as its name implies does not support mobility; while the mobile WiMAX (IEEE 802.163) can both serve as a fixed or mobile WiMAX.

WiMAX is popular due to its low cost and flexible nature [7]. It can be installed faster than other Internet technologies because it can use shorter towers and less cabling to support even non-linear visual coverage (NLoS) throughout the city or country. Wi-Max, like at home, is not just for fixed connections either; users can also subscribe to a WiMAX service for mobile devices because USB dongles, laptops and phones are equipped with the technology. Yet, Since Wi-Max is naturally wireless, the further away from the client's source, the slower the connection becomes and Similar to the bandwidth of several devices connected to a single router, multiple users in one Wi-Max radio sector reduce performance for the rest [8].

Fibre optics (optical fibres) are long, thin strands of very pure glass about the size of a human hair. They are arranged in bundles called optical cables and are used for long distance transmission of signals [9]. Fibre optic data transmission systems send fibre information by lighting electronic signals. Light refers to more than the electromagnetic spectrum that is close to what can be seen in the human eye. The electromagnetic spectrum consists of visible and near-infrared light such as fibre and all other wavelengths used for the transmission of signals such as AM and FM radio and TV. Spectrum of electromagnetic. Only a very small portion of it is seen as light by the human eye [9].

As stated in [9] [10] [11] Fibre Optics possess high bandwidth that allow high amount of information to pass through them at the same time. It also, have low power loss and so are able to successfully transmit data over longer distances when compared to copper (Example the distance for copper is 100 m while for fibre is 2 km). Fibre Optics are not affected by high levels of noise or interference and its raw materials is cheaper when compared to copper. However, when bent too much can easily break and requires Specialists and special equipment to test, install and maintain the fibre, thus making it more expensive compared to copper. Also, laying fibre has to be done in a straight line as much as possible, or will have to be repeated at regular intervals because they can lose signals when installed around curves. Fibre Optics become opaque when exposed to radiation and when deployed underwater, can be easily affected by chemicals such as hydrogen gas.

4G is the fourth generation of cellular wireless standards in telecommunications. It is a successor to the 3G and 2G standard families. 4G system provides laptop computer wireless modems, smartphones and other mobile devices with a comprehensive and secure all-IP mobile broadband solution. Users may be provided with services such as very high-speed broadband, streaming and gaming services, IP telephony. Pre-4G technologies such as mobile WiMAX and first-release 3G Long-term evolution (LTE) have been on the market since 2006 and 2009 and are often referred to as 4G technologies. The current versions of these technologies have not met the original ITU-R data rate requirements for 4G systems of 1 Gbit/s. Marketing materials still use 4G in their current stage to describe Mobile-WiMAX and LTE [12] [13].

4G is a term widely used to define different types of wireless/broadband access systems, and not limited to cell phones only. 4G systems, i.e. cellular broadband wireless access systems, have attracted a great deal of interest in the mobile communication arena as a promise for the future. Not only will the 4G systems support the next generation of mobile services, but the fixed wireless networks [12].

4G is versatile and enables providers to provide services to different types of devices. It is also flexible in terms of mobility, cost and usage. It provides stable and reliable connection to multimedia services and broadband services and when compared to 3G and Wi-Fi, it provides massively improved connection speed, bandwidth, throughput and wider coverage. Not withstand It is susceptible to interference and can be attacked (jam frequencies) and privacy invasion is increased. It Consumes high amounts of battery power and requires expensive complex hardware [13].

Finally, Bluetooth technology is a low-power wireless connectivity technology used for audio streaming and data transmission between devices. It provides the means to replace cables and infrared connections that connect one device to another with a universal short distance radio connection. Although initially developed to replace cables, this technology has now evolved into a way to create small radio LANs. Bluetooth technology allows users to establish a form of wireless connections over different devices in order to communicate and exchange/transfer information. The information from one device to another device is used with the Bluetooth device. This technology has different functions and is commonly used on the wireless market [14].

Bluetooth technology can easily create cheap wireless ad-hoc connection networks very fast transferring voice and data without consuming a lot of power. It uses FHSS technology and hence data communication less interference than other wireless technologies. Signals can transmit through walls and covers more distance when compared to Infrared. Bluetooth technology is used in many products such as head phones, in car system, printer, web cam, GPS system, keyboard and mouse etc. Security is one of the big disadvantages of Bluetooth because it works on the radio frequency. [14].

In conclusion, Wireless Communication technologies transmits data and information at a high speed, cheaper to maintain and install compared to other networks, provides wireless connection to the internet from any location and can be easily deployed in remote areas where there is little or no infrastructure. However, high security measures need to be taken into consideration to protect the network from intruders who can intercept or easily monitor wireless signals [16].

2.7 Machine to Machine (M2M)

Machine to Machine (M2M) communication is a paradigm in which end-to-end communication is executed without human intervention. M2M communication is also defined as the exchange of information between a subscriber station and a server in the core network (via a base station) or between subscriber stations that can be executed without any human interaction [17].

It is envisaged that the M2M paradigm will replace machine-to-human communications with the rapid development of networking technology and the continuous penetration of embedded devices in our environment [18].

Networks are growing rapidly to include a wide range of devices, including laptops, TVs, mobile phones, personal computers and home electronics. These robust networks and the embedded devices create an ideal environment for M2M communications to be dominated by low power consumption, low cost and low human intervention. M2M communication technology is currently widely used in intelligent homes, automotive applications, smart cities and eHealth [18].

Home entertainment is, of course, more adaptable to M2M communication technology than to the healthcare environment or what is known as eHealth due to strict requirements such as security, privacy and reliability across the eHealth system and the real-time nature of medical applications. However, intensive research and application to improve the Quality of Service (QoS) in the eHealth sector and promote the M2M eHealth environment is under way [19].

The demo shown in the next section illustrate how M2M could be configured to enable the audience to use different client devices (e.g. smart phone, tablet PC, notebook) to access the M2M applications and control sensors, actuators, and devices (e.g. lamp, fan). Also, the user can feed the M2M system with policies to trigger automated sequences of actions and thereby steer and control the M2M communication that is performed without human intervention [20].

However, it is worth mentioning that the architecture of this demo aims to be generic and focused on the communication mechanisms to support different M2M in terms of a common platform. Figure 2.10 and 2.11 show the scope of the demo [20].

MACHINE	-- TO --	MACHINE
Communication terminal independent of human interaction ▪ Acting automatically or on remote request ▪ Managed remotely ▪ Mobile and fixed terminals ▪ Monitoring device (sensor) ▪ Actuator device (e.g. switch) ▪ Associated order of magnitude: trillion = 10^{12}	**Network facilitating the M2M communication** ▪ Access & core network, backhaul, app. server ▪ Enabling connectivity (AAA & security, session management, QoS, charging, mobility) ▪ Supporting data traffic of terminals ▪ Supporting the signaling of terminals	**Terminal automating the services** ▪ Sensor data aggregation, processing and presentation ▪ Data caching and interpretation ▪ Real-time communication ▪ Automatic decisions and control ▪ Policies, instructions and notifications

Figure 2.10. M2M Communication

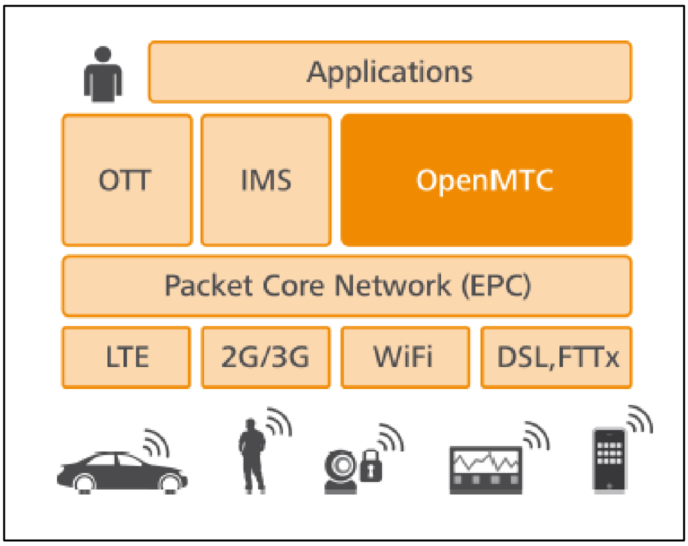


Figure 2.11. The Open Machine Type Communication (MTC) framework

Health, medical and industry organizations are also a major user of the M2M communication field called eHealth. eHealth provides partial but significant solutions to healthcare problems faced by most countries, such as increased numbers of elderly people, chronic diseases, limited financial resources, limited access to healthcare outside hospitals and clinics and, last but not least, recruitment and retention of qualified staff in healthcare services, especially in home and elderly care. Potentially, eHealth will be able to provide customized and highly effective care for patients irrespective of certain technical issues such as compatibility, privacy, reliability and security [20].

As a result of the benefits, implementation of M2M technologies are already being used in a wide variety of applications. Based on this M2M trend, much more services will be adopted and smart devices will eventually become ubiquitous. Whether it is smart cars or 3D smart pens, M2M communication has far-reaching applications in the technological world. This increased use demonstrates an enormous market growth potential for companies producing M2M communication products [21].

M2M applications are becoming increasingly popular because they are constantly evolving in terms of size reduction, efficiency and have sensors that are tiny in nature. Also, applications are becoming wearable which it even makes it more convenient and provides users with flexibility of use [18]. It also provides collaborative opportunities for services providers, for example car manufacturers partnering with mobile and 4G providers to make their cars 4G smart. More advanced projects will include the plan by Mercedes Benz and Google to create driverless cars [20].

In almost every industry we see the internationalization of connected solutions, one industry in which surprising growth has occurred is insurance telematics. Companies such as Octo Telematics have started paying for insurance solutions that work by providing the user with a tracking box that logs information such as driving frequency, etc. The insurance company can then issue insurance quotes on the basis of that information [22].

Case studies highlight the use of intelligent meters by energy providers in Finland was highly beneficial to companies and consumers. The government supported this initiative by making it a requirement for providers to give users the option of installing smart meters in their homes [22].

Current M2M setups work on a more technical level with a centralized hub that is responsible for setting up and receiving data from wired and wireless clients. An operator processes this information and sends out a reactive command. This central hub would be removed in the future and the devices would communicate and perform the tasks themselves. This would create a very fast response time and also help to reduce costs in companies.

Currently, a cellular network is the most common wireless communication media for M2M networks. Knowing that these cellular companies have invested a great deal in improving their M2M communication network. There are a number of other M2M communication media, such as Bluetooth and ZigBee are currently used which are in continuous improvements for example, improvements have been made in Bluetooth from version 4.1 to version 4.2 which promises to be faster and even more suitable for M2M communication due to features such as GATT that makes it easier to connect to the internet.

M2M is projected to be very successful in the near future with forecasts such as an annual increase of 30 percent in M2M products for the coming year and an increase of 23 percent in 2020. With the current global M2M revenue of 121 billion dollars, analysts say that M2M is on track to reach more than 900 billion dollars by 2020. Ericson also predicted that the total number of connected devices will reach 50 billion by 2020. However, four main challenges lie in achieving all these forecasts [23].

1. Technology standardization: To thrive M2M Standard protocols such as communication between devices that are unique to M2M must be established.
2. Platforms for innovation: More open source tools should be available to enable users to develop advanced solutions.

3. Consolidation: Vendors or manufacturers can collaborate more to create singular products, thus reducing independent solutions and increasing universal solutions.
4. Sales: Efforts to ensure privacy should be heightened to reduce uncertainties and fear of privacy issues. Data policies by companies should be made very clear and acceptable to the users.

In general, M2M applications will play greater roles in everyday life in the very near future.

2.8 How Machine to Machine (M2M) Works

M2M communications usually consist of networked devices and a gateway. The gateway has two main functions: the connection between the networked devices and the connection between the M2M communications area and other networks, such as the Internet. The M2M network can also use a suitable standardized radio technology based on the requirements and flexibility of a particular application [24].

From the data management perspective, M2M communications consists of three phases [24]:

1. Data collection
2. Data transmission
3. Data processing

The data collection phase refers to the physical data collection procedure. While the data transmission phase includes the mechanisms to transmit collected data to an external server from the communication area. Finally, the data processing phase is the process of data processing and analysis and also provides feedback on how the application can be controlled [24].

With all the benefits of M2M, it is understandable that it will be used in a wider range of fields. Due to its increasing popularity and success, M2M needs to have a standardized structure to provide guide and help maintain the technology. This is provided by institutions such as ETSI, ANSI and C12 [19].

Figure 2.12 shows an example of an M2M System designed by ETSI. It also shows the 3 domains of the architecture as shown below [19]:

1. Application Domain.
2. Network Domain.
3. Device Domain.

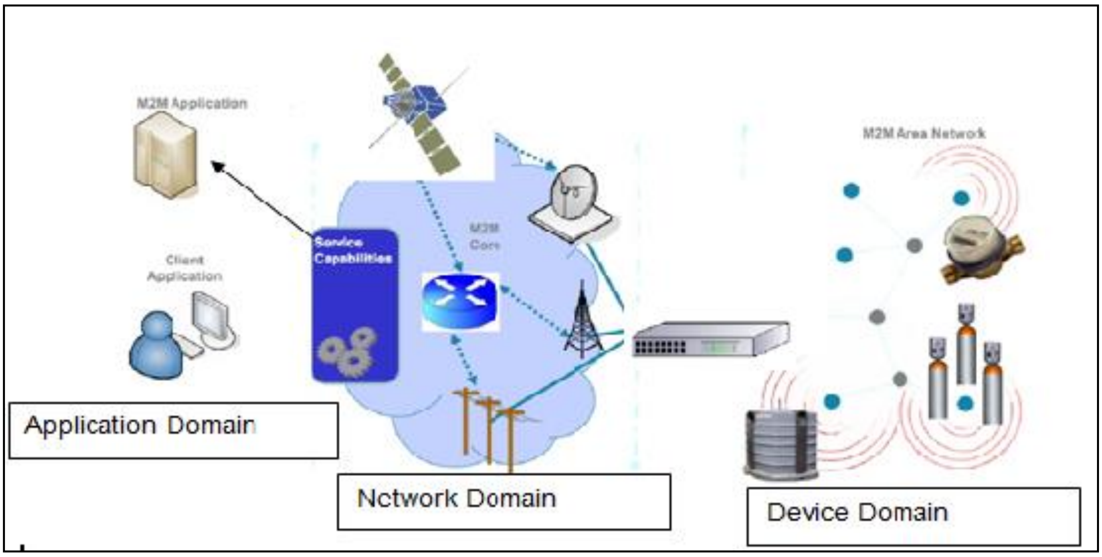


Figure 2.12. M2M System Architecture

2.8.1. M2M Usage Models

M2M concept allows the Internet of Things (IoT) which is a set of technologies to communicate between devices, users and applications. Devices such as smartphones, sensors, smart meters etc. which has the ability to communicate over the internet and interact with services being provided and establish an interface with application domains such as healthcare and self-tracking, home automation and monitoring, smart cities and mobility.

Smartphone is one of the most advanced technologies that establish interface with application domains and drive information collection and interaction with the users into new experience. Smartphones utilize integrated sensors such as location sensors which gives the applications' developers a great opportunity to develop location-based services and application such as weathers apps, maps apps and recommendation apps for retail and promotions apps.

M2M application is widely used in various sectors and industries. The relevant M2M usage models could be shortlisted in the following applications [25].

1. Healthcare
2. Secured Access & Surveillance
3. Tracking, Tracing, & Recovery
4. Public Safety
5. Remote Maintenance and Control
6. Smart Metering
7. Consumer Device market
8. Retail

Each model has its own parameters and requirements with regards to data security, reliability, privacy and Quality of Service (QoS).

M2M systems work in many real-world applications, learning about these areas of application, a better understanding of how M2M works is achieved. Grouping the application areas into categories as shown in Table 2.2 is a realistic approach to have better understanding of the M2M systems [25].

Table 2.2. 3GPP M2M Usage Grouping

Service Area	M2M Application
Health Sector	• Support for old aged or handicapped people • Remote diagnosis • Monitoring vital signs of comatose patients • Web based tele-medical services
Consumer electronic Devices	• Smartwatches • eBooks • Smart glasses • Digital Photo frames • Automatic back up in consumer devices
Metering	• Gas • Grid • Electricity • Temperature • Broadband • Water
Payment and billing	• ATM's • Point of Purchase (PoP) • Vending Machines
Remote control and Maintenance	• Sensors • Vehicular telemetry • Vehicular diagnostics and troubleshooting • Pumps • Switches • Pneumatic actuators • Lifts • Relays
Security Systems	• Controlling physical access • Vehicular security • Backup systems • Surveillance systems
Track and trace	• Tolls on roads • Fleet management • Navigation systems • Asset tracking • Logistical order tracking • Traffic management systems

2.8.2. eHealth

A huge benefit of the M2M technology is its use in life critical applications such as eHealth. Doctors and other health workers are able to better monitor patients with different conditions and report changes or carry out required actions based on results analyzed. Wearable devices also help monitor temperature of the body, weight, heart rate and blood pressure etc.

Figure 2.13 illustrates M2M application of eHealth over a WAN network. Here, the devices use the IEEE 802.16 standard to establish communication link the devices and the healthcare management system. Readings from patients at different locations is sent from the devices the healthcare management system and vice versa which doctors are able to monitor [31] [32].

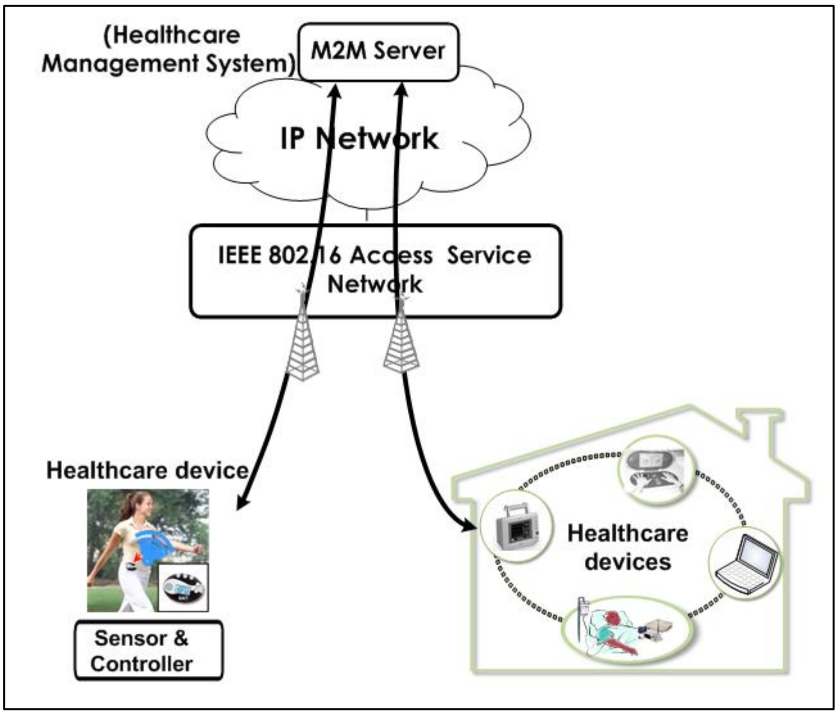


Figure 2.13. Example of M2M Scenario for Healthcare

Brief details of areas in eHealth where they are being implemented include:

Disease Management: This allows doctors remotely monitor patients who have certain diseases and analyze their condition accurately. For example, pulse levels, glucose levels etc. can all be monitored. Glucose levels are crucial and sensors can inform doctors and patients when those levels are about to reach a dangerous point and appropriate action can be taken [31] [32].

Remote monitoring of patients: This is the most common benefit where patients can gain remote access to medical help and doctors are able to monitor and treat patients based on changes in their conditions [31].

Elderly citizen applications: Saves elderly people trips to monitor vital signs as those readings are sent to doctors via sensors. For example, blood pressure, heart rate, glucose levels can be monitored in their homes and sent to the doctors [31] [32].

Personal fitness application: This is very popular with the implementation of wearable sensors that are able to measure the number of calories being burnt, oxygen blood levels and heart rate. Readings are sent to smartphones instantly and can be further analyzed by health service providers to provide custom solution for the users [31] [32].

eHealth deals with the use of technology to carry out health and medical functions with the aim of enhancing the delivery and access of these health services. eHealth system involves all medical personnel, government authorities involved in the delivery of health services to patients [33]. Examples of these applications have already been discussed in the sections above.

According to the above definition, eHealth has a wide range of applications in hospitals, clinics, homes and government. In eHealth applications, M2M and many other ICT-based tools are used to collect, transmit and process data. The strict requirements needed in terms of privacy security and reliability when using real-time health applications that could include the transmission of multimedia data, introduce the need for adequate quality of service (QoS) in wireless medical networks dominated by M2M communication [33].

The various advances in wireless communication technology have contributed to the improvement of health services in remote and accessible areas throughout the world. Wireless network technologies have advanced to a stage where they can be used to support medical applications with a wide variety of devices. Several research projects have proposed the replacement of wired links between medical devices with wireless links [33].

Wireless medical devices with different sensors for measuring and reporting vital signals have the ability to form different size networks such as the Wireless Body Area Network (WBAN) and the Wireless Personal Area Network (WPAN) to support healthcare applications in different geographical regions. This brings new vibrations from the Internet of Things (IOT) and Machine to Machine Communication (M2M) in healthcare research [33].

To achieve an eHealth M2M network architecture, shown in Figure 2.13 structures, and illustrate the network needs. The eHealth M2M network is primarily a heterogeneous network with a backbone and multiple subnetworks. There is a central gateway in the backbone network that manages the entire network and connects the network to the external world (e.g. the Internet). In addition, each subnetwork operates in a self-organized manner and the network-related functionalities in the central gate, including access control, security management, quality of service (QoS) management and multimedia conversion, can be designed for a specific application [33].

This paper focuses on the technical characteristics of a health system structure. In reaching this structure, however, it is also important to consider how the users are able to interact with the system. An example is shown in Figure 2.14.

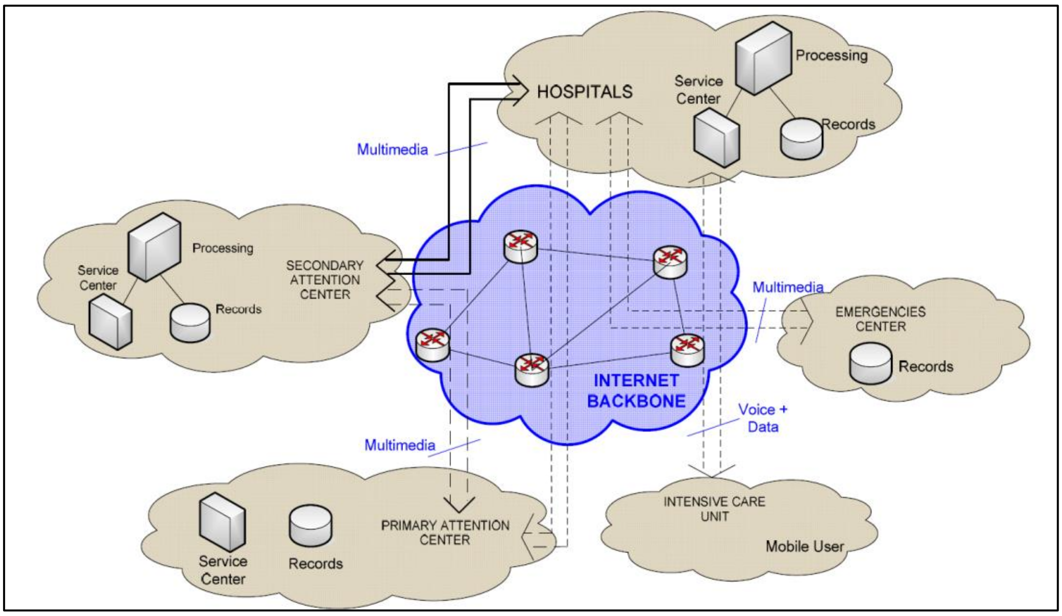


Figure 2.14. The eHealth System illustrating links and users

2.8.3. Other M2M Usage Models and Applications

Consumer electronic Devices: As consumer electronics technology grows, specialist devices such as cameras, PDNs and electronic readers are replaced by all devices such as smartphones and tablets that are able to carry out multiple functions. The wearable technology market is another genre of technology that is growing. Mobility, personalization and connectivity are the selling point of such devices. Notifications from different news and social media platforms is fast becoming a requirement in the design of these devices. These multiple functions can be carried out by these devices with the help of M2M networks. Examples of such devices include Smartwatches, Smart glasses, Consumer health devices, Personal tracking devices and Digital photo frames [22].

Secured Access & Surveillance: M2M applications in this category are integrated with security appliances such as cameras, intrusion systems and car alarms/burglar alarms to help transmit real-time data to monitoring devices. The M2M enabled devices transmit signals to an M2M server at a specific location. This can be done over local or wide area networks depending on range and size of the network or coverage required [26].

Smart Metering: In terms of energy services, M2M is implemented to create smart meters that will help monitor energy consumption thus saving both customers and providers money while also improving the energy efficiency of the users [26].

Measurement and metering services are one of the areas in which M2M communications are widely used. We call this intelligent metering. This is because consumers can receive information on the use of utilities a lot faster and more accurately. Thus, giving the user a better idea of how their consumption is distributed [27]. The challenge with smart meters a certain quality of service (QoS) has to be adhered to. This can be an issue as most smart meters are commonly installed in remote buildings or locations making the meters easily affected by interference and loss of signal. The issue gets worse if higher signals are used for transmission [26] [27].

Payment and Billing: M2M enables the use for applications for different payment scenarios thus making it easier to pay on the go and more accessible for both vendors/companies and their customers. As it's a wireless payment system it can be used in restaurants, parks, taxis etc. with the aid of an internet connection [28].

However, security measures and legislations exist that ensure connections during such transactions are encrypted and sensitive information of users are kept private.

This helps reduce security incidents from attackers and malicious users thus improving customer satisfaction and experience [28]. Some control measures that can be taken include; regular software updates to fix bugs and vulnerabilities in the system, installing current anti-malware systems ad strong firewall, encrypted connections and usage monitoring and intrusion prevention systems etc. [28].

Remote Control and Maintenance: Various applications are designed to monitor data and networks especially in large locations like a data centre. Certain parameters are defined on these applications and notifications are sent either in the form of emails, alarms sounding or both when certain limits are reached or exceeded. In other forms, they can trigger another system for example shutting down a server room in the case of fire or opening doors when flooding is sensed. [30]. In Formula 1 cars for example, vehicles are equipped with many sensors that measure and monitor different aspects of the cars which is then transmitted to the backbone network for analysis. These remote systems rely heavily on a full functioning network connection for effectiveness of these applications [30].

Security Systems: similar to remote monitoring applications, security applications will require reliable connectivity to function properly. M2M technologies have now been embedded in most security products such as CCTV, alarm detection systems, fire alarms etc. with instant access provided to the users over the internet or some of form of WAN or private connection. High cost of installation is the major issue affecting such products as they are expensive to install and require specialists to install the system. They are also affected by jamming signals or will be unable to function effectively when offline. Feeds from such security devices are transmitted to a remote server and can be analyzed. The quality of the devices and the network connection determines the quality of the video resolution or images generated [31].

2.8.4. eHealth Challenges

Based on the literature review performed part of this research, it became obvious the lack of adequate priority queueing algorithm that satisfies the stringent eHealth QoS requirements especially in emergency cases and life-threatening situations. Thus, it deemed necessary to improve real emergency data transmission in terms of delay and jitter through adopting new priority queuing algorithm [34].

1. Despite the fact that today's communication networks, such as mobile operator networks, have been designed and developed to meet the needs of human communication, M2M communication solutions that have emerged and are available on the market today are predominantly monolithic infrastructures that do not interoperate have been identified as major show stoppers for open, customizable and universal M2M system users.
2. Machines are normally small and cheap, which places several limitations on M2M communications, including energy, computing, storage and bandwidth. These challenges pose a number of unique challenges in the design of home M2M networks in order to achieve a connected, efficient and reliable home.
3. There is increasing interference with more radio systems in the home, including unlicensed frequency band, electronic equipment and domestic appliances in the industrial, scientific and medical (ISM) systems. Due to such interference in self-existence/co-existence, the performance of M2M communication can be seriously degraded.

4. Wireless channels in M2M communications are notoriously unreliable because of channel fluctuations and noise, which may get worse because of the complex construction in an indoor environment.
5. Machines can be extremely limited in terms of computing, storage, bandwidth and power supply. The balance between energy, reliability and flexibility due to resource constraints is always essential.

2.9 Quality of Service (QoS)

QoS refers to the quality of a network based on various performance criteria. It is a qualitative measurement to ensure the reliability and usability of the network in relation to the network application which requires different QoS requirements. The qualitative measurement criteria include some of the following [35].

1. Throughput
2. Delay
3. Packet loss
4. Reliability
5. Jitter

The QoS therefore aims to guarantees a certain level of performance quality usually stipulated by regulatory bodies [35] [36]. If QoS levels are poor or low, applications will not function effectively and will put users off. The traditional QoS approach aims to prioritize particular users. This can be done in several ways firstly, different types of traffic can be given high, medium or low priority. The second method is to assign priority levels to particular services or ports. Finally, prioritization can occur by assigning it a network or hardware address (IP or MAC address) [35] [36].

Concerning to eHealth enhancements, QoS parameters of telecommunication networks, like bandwidth, availability, delay, security and ubiquity are important to ensure successful implementation of eHealth applications. eHealth is divided into six separate fields according to set of activities: Consumer health, clinical care, administrative and financial transactions, public health, professional education and biomedical research [59].

Consumer health refers to a set of activities giving consumers an obvious role in their own health and health care such as self-assessment of health risks and management of chronic diseases, to home-based monitoring of health status and delivery of healthcare. In this perspective, reliable consumer-oriented health web sites, secure transfer of data between patients and providers, effective online health records and efficient home care monitoring devices should be available. Retrieval of information from health-related Web sites requires low bandwidth on the consumer end, but the potentially large volume of requests made of any particular site could drive up the aggregate bandwidth requirement on the information provider's side. Also, consumer health applications demand high levels of security because electronic health records contain personal information thus ubiquity is very important.

Clinical care and remote video consultation, for instance, give the patients great access to skilled health professionals regardless of geographic proximity. Remote consultations require adequate bandwidth for real-time video. The transmission of large medical images also requires high bandwidth to support the transfer of large numbers of images. Remote consultation requires lower latencies to facilitate more natural interactions between participants. For safety and timeliness considerations in patient care, availability of the network should be vital. Clinicians awaiting a lab result, radiological examination, or connection to a patient's home can't tolerate any unavailability of the network or the clinical applications running on it. Likewise, remote consultations will not be viable if network availability cannot be assured and connections are broken frequently.

Financial and administrative transactions of healthcare organizations require the network and system outage to be low to ensure successful transaction in addition to the sensitive information being transmitted require robust security measures.

Public health workers who promote health and the quality of life by preventing and controlling the spread of disease, injury, and disability require the availability of the network and Security since the public health system depends on the public's trust that sensitive health data are being used for the benefit of the public.

The following sections look at different QoS parameters. Each of these parameters is described below.

2.9.1 Throughput

This is defined as the number of successful packets that can be sent via a single communication channel (logical or physical) at a given time measured in bits per second or packets per second [37].

In communication instances, throughput refers to the performance of each component of the system or aggregate throughput. For example, when a laptop is connected to the internet, the throughput will be the capacity of the wireless card of the router over which it is connected, cables and processing power add up to give the throughput [37].

Some major factors limiting the throughput essentially are restrictions due to the use of analogue signals. This is so because analogue signals' bandwidth and SNR affects the output (SNR). This type of limitation is inevitable because signals that all use analogue signals via wireless media and wireless media. The relationship that exists between throughput, bandwidth and SNR is demonstrated by the theorem of Shannon Hartley [37].

$$C = Blog_2\left(1 + \frac{S}{N}\right) \qquad \text{Formula 2.1}$$

Where: "C" is the channel capacity in bit/s

"B" is the bandwidth of the channel

"S" is the received signal power

"N" is the average noise

The processing power of devices is another factor that can lead to bottleneck; this is when there is very high competition for bandwidth or not enough bandwidth. In addition, when multiple devices connect to a network, the throughput of the individual devices can be reduced because all the users connected share the same bandwidth. In addition, multiple network users can cause packet loss due to congestion, bit errors and priority queuing. Priority queuing is when some network devices prioritize packet size devices connected to the network. This results in poor performance for other equipment [37]. The processing power of devices is another factor that can lead to bottleneck; this is when there is very high competition for bandwidth or not enough bandwidth.

2.9.2 Delay

Delay and latency in general engineering terms refer to the amount of time required for complete communication. Latency is referred to as the amount of time a packet is held by a network or device or system. This can be the time taken for a packet to traverse a router. While delay is the time it can take one bit to travel the entire network from device to device and can be caused by conditions such as errors in bits, correcting errors, distance and congestions etc. delay can be divided into 4 parts and are briefly described below [38]:

1. Processing delay: This refers to the time it takes to process the header of a packet and check for errors on the packet. These types of delays are usually very minimal and are normally ignored [38].
2. Queuing delay: This refers to the time where the packet is waiting to be processed or transmitted. Other packets waiting to be processed (by the router for example) will have to remain in what is called a queue buffer while the system processes one data at a time [38].
3. Transmission Delay: This refers to the time it takes for all the bits to be sent to the communication medium. Such delays are usually affected by the quality of the connection or the speed of the link [38]. Transmission Delay is measured using the following formula:

$$D_t = \frac{N}{R} \qquad \text{Formula 2.2}$$

Where: D_t = the transmission delay

N = the number of bits

R= the transmission speed

4. Propagation Delay: This type of delay is determined or caused by the size or the distance the signals have to cover. It is the time it takes the signal to travel the length of the entire system to the end [38].

In a networking, latency is described as the time it takes to reach the source destination of a packet [39] In VoIP latency, the delay for delivery of packets. The effect of latency is very visible in satellite communication. This is because geosynchronous satellites are based far away from receptors and transmitters. For example, because of high latency when a reporter from another country speaks, there is a noticeable delay in speaking in a live news release [38] [39].

2.9.3 Packet Loss

This refers to the number of packets that are not able to reach their destinations during after they have been sent over the network. The percentage of the number of lost packets to the total number of sent packets is how packet loss is measured. The TCP protocol tries to recover lost packets in failed transmissions by creating a retransmission request. Sometimes depending on the number of packets lost, data corruption may occur as a result [40].

One of the main causes of packet loss is bottleneck due to congestions where packets are dropped as a result. Another way could be due to reduced signal strength as a result of increased distance in wireless or when wireless networks come across obstacles that affects the broadcast of the radio waves. Packets can be deliberately dropped and hardware failures can also lead to packets being dropped [38] [40]. Packet loss has a direct effect on latency, because when the system is trying to retransmit lost packets, this will increase the time it will take to travel the entire network and get to its destination [39].

2.9.4 Reliability

This is defined as the degree of availability of a network and its ability to stay connected or disconnected. Therefore, it means that the network or system should be able to perform and meet the stipulated standards consistently [41]. Four main categories include [41]:

1. Connectivity: This means the reliability of the initial connection and the success rate, that is, the percentage of total successful relationships. For instance, it is 99.98 percent in a 4G network [41].
2. Capacity: This refers to the amount of data which can be sent from source to destination [41].
3. Multi commodity: This is the ability of the network to carry out successful communications between source and destination for different devices at the same time [41].
4. Perform ability: This is a network's ability to continue to perform against adversities such as component failure and network overload. For instance, if a connection between nodes drops, the network should be able to find alternate paths via other nodes to ensure the communication is complete [41].

2.9.5 Jitter

The International Telecommunication Union (ITU-I) defines Jitter as: The short-term variation of the significant instants of a digital signal from their ideal positions in time [60]. `Jitter is a term used in the telecommunication domain. As information is transmitted as bits in a data stream, it requires to be available strictly at a certain time and be present for an exactly predetermined period. However, the real-world scenario is far from ideal. Designers have to overcome a variety of factors that influence the quality of the transmitted data that happen to be a source of jitter. Jitter is commonly recognized as a high frequency quantity. Jitter-like behavior at frequencies below 10 Hz is called “wander” and “drift” at even lower frequencies [60].

Timing variations relative to the ideal transition time are called phase jitter whereas, signal level variations that occur in digital systems; are called amplitude jitter. Jitter can be bounded or unbounded. Bounded jitter is related to frequency and magnitude to system events; therefore, bounded jitter is deterministic. This means disabling the source will stop related bounded jitter too. Bounded jitter always has a limited magnitude. An example would be Inter Symbol Interference (ISI): signal transitions cause interference to the neighboring channels.

Unbounded jitter does not depend on events however system components or external influences can cause it. Most prominent is Random Jitter (RJ), which is caused by white noise prevalent in all active and passive components.

System Jitter is caused by a variety of sources, and is either random or deterministic. The latter means that its causes are clearly linked to system events. Such causes could be interference with neighboring channels during level transitions or insufficiently filtered switching pulses. Table 2.3 shows provides a full description of jitter types and causes.

2.10 Queueing Algorithms Techniques

Queuing algorithms are very important factor in data transmission from the source to the destination as it is the responsible element in sorting, scheduling, classifying and prioritizing data being transmitted over the network. Proper and adequate queueing algorithm is required for each industry using data transfer to meet its requirement and demands. eHealth is one of those industries that depends on M2M and data transfer under strict QoS requirements. Therefore, the need for a robust queueing algorithm that meets those requirements and improve the current algorithms available in the market deemed extremely important to meet the huge forecasted demand in eHealth applications and the M2M growth in the few coming decades.

In recent years, several Queueing algorithms have been developed by researchers to describe the data and packet transfer protocol. Different Queuing algorithms serve different applications and purposes [42] [44]. First of all, one of the basic and primitive techniques (FIFO). Round Robin (RR) and Fair Queueing (FQ) or Weight Fair Queuing (WFQ) are advanced Queueing algorithms that meet the demanding needs of the industry and the satisfaction of end users. They also reduce delay and jitter to certain extend [42] [44].

Table 2.3. Jitter Types and Causes

Jitter Type	Causes
TJ Total Jitter	The summation (or convolution) of deterministic and random jitter. Total jitter is the peak to peak value obtained. TJ = DJ + n × RJ where n = number of standard deviations corresponding to the required BER.
RJ Random Jitter	The principal source is Gaussian (white) noise within system components. It interacts with the slew rate of signals and produces timing errors at the switching points.
DJ Deterministic Jitter	Jitter with non-Gaussian probability density function. It is always bounded in amplitude and with specific causes. Sources are imperfections of devices, crosstalk, EMI, grounding problems.
PJ Periodic Jitter	Also called Sinusoidal Jitter due to its sinusoidal form. The source is usually interference form signals related to the data pattern, ground bounce or power supply variations.
DDJ Data dependent Jitter	Consists of Inter Symbol interference (ISI), Duty Cycle distortion (DCD), and Echo Jitter (ECJ). Timing errors vary with data pattern. Primary source are component and system bandwidth limitations. Higher frequency signals have less time to settle than lower frequency ones. This leads to changes in the start conditions for transitions at different frequencies and produces timing errors dependent on the data pattern being applied.
ISI Inter Symbol Interference	Inter symbol interference is the most common form of DDJ. It is usually caused by bandwidth limitations of transmission lines. It affects single bits surrounded by the bit of the opposite state.
DCD Duty Cycle Distortion	Duty Cycle Distortion Jitter is caused when certain bit states have different durations. "1" is always longer than "0" or vice versa. Caused by bias setting, and insufficient VCC supply of a component.
ECJ Echo Jitter	Echo Jitter is caused by component/line mismatch, it depends on the data pattern. Line length influence the magnitude of ECJ as well.

2.10.1 First-In First-Out (FIFO)

One of the most basic and fundamental simultaneous data structures is First-In-First-Out (FIFO). FIFO queues are an important block of simultaneous libraries for data structure. FIFO planning is easy to implement. It's intuitively fair too (the first in line runs first). The biggest drawback of the first schedule, however, is that it is not pre-emptive. It is therefore not suitable for interactive occupations. Another disadvantage is that a long-term process delays all the work behind it [42] [44].

The First-In-First-Out (FIFO) queues are the easiest approach to planning processes. New processes reach the end of the queue and when the scheduler needs to run a process, the process at the head of the queue is selected. This planner is categorized as non-pre-emptive. If the process has to block input / output (I / O), the waiting state is entered and the scheduler selects the process from the queue head. When I / O is finished and the (blocked) process is ready to run again, it is placed at the end of the queue [42] [44].

The queue is an abstract data type which complies with the First In First Out rule (FIFO). It is used when elements in the order in which they arrive are processed. There are therefore many uses in computer operating systems, e.g. the process queue and the print queue [44].

Packets are usually transmitted in the order of arrival in the outbound interface queue of a router; technically, this is FIFO (first-in, first-out). In the absence of priority levels, FIFO tail-drop occurs when a packet arrives at a router whose queue is full for the desired outbound interface and the packet at the tail is dropped [44].

Alternatively, in a completely different context, RFC 4681 states that "When a queue is overflowing, the new arrival should replace the oldest entry" [42].

FIFO random drop is an alternative drop policy mechanism for the queue router. Under FIFO's random drop policy, when a packet arrives but the destination queue is full, while N packets are waiting, one of the N+1 packet in all the N waiting plus the new arrival–is randomly selected for drop. The latest arrival therefore has a very good chance to get an initial place in the queue, but also a reasonable chance to be dropped later [42] [44]. A typical example of the First-in-First-Out (FIFO) algorithm for Queue Abstract Data Type is illustrated below [43]:

```
type queue = empty_queue | addqElement_Type × queue

Operations:
front :queue ->Element_Type
addq  :Element_Type × queue -> queue
popq  :queue -> queue
empty :queue ->boolean

Rules:
front(addq(e, empty_queue)) = e
front(addq(e, q)) = front(q),    if q not empty
   front(empty_queue) = error

popq(addq(e, empty_queue)) = empty_queue
popq(addq(e, q)) = addq(e, popq(q)),  if q not empty
popq(empty_queue) = error

   empty(empty_queue) = true
empty(addq(e, q)) = false
```

For most interfaces, FIFO is the default queuing discipline. The hardware queue (TxQ) also processes FIFO-based packets. Every queue is a FIFO queue in a multi-queue discipline. FIFO is a simple algorithm that does not require any effort to setup. Parameters such as the type of the FIFO queue, priority of queue and class of the packets do not count when considering a single FIFO queue. And in such single queues, bandwidth can easily be used up by applications. Multiple queues and scheduling are therefore needed to reduce packets dropped and lost. In faster less congested interfaces, it can be ideal to use FIFO [44]

The main difference between a stack and a queue is first-in-first-out FIFO and last-in-first-out stacks (LIFO). The LIFO mechanism is well represented at the bank, supermarket or check-in counter in the ordinary line (queue) example. While people can be added to the queue and the opportunity to serve the next customer, the person to be served is not only the first person in the queue, but also the others shuffle up to the front of the queue and the process continues [45].

The aim of a stack data structure is to save items so that the latest item is first found. It gives access only to the top element of the stack (the latest element). Items are therefore processed in the order Last-In-First-Out (LIFO). In a printer for example, the first FIFO mechanism is well illustrated. Since the printer can only deal with one thing at a time, it has an integrated queue, which retains several jobs for printing and takes those jobs in the order in which they are submitted [45].

The objective of a queue data structure is to store items in a way that first finds the least recent (older) item. It only gives access to the front of the queue and always adds the oldest element to the back of the queue. Items are therefore processed in the order of first-in-first-out (FIFO) [44].

The first circular queue in the first place (FIFO) is also useful for data flow problems. It is an extremely common data structure used to interface input / output (I / O). The data structure preserving order temporarily saves data generated by the source (producer) before the sink (consumer) processes it. The advantage of using a FIFO structure for a data flow problem is that the source (producers) and sink (consumers) processes can be separated. Without the first-in-first-out (FIFO), we would have to produce one piece of data, process it, produce another piece of data, process it and continue the process. With the FIFO decoupling, the source process can continue to generate data without waiting for the sink to complete the previous data processing. This decoupling can improve system performance significantly [44] [45].

A typically Application Programming Interface (API) functions for a Stack data structure are defined below [46].

```
push(e): insert element e, to the top of the stack
pop(): remove from the stack and return the top element on the stack
size(): return the number of elements on the stack
isEmpty(): return a boolean indicating if the stack is empty
top(): return the top element on the stack, without removing it
```

A typically Application Programming Interface (API) functions for a Queue data structure are defined below [46]:

```
enqueue(e): insert element e at the rear of the queue
dequeue(): remove and return from the queue the element at the front
size(): return the number of elements in the queue
isEmpty(): return a boolean indicating if the queue is empty
front(): return the front element in the queue, without removing it
```

2.10.2 Round Robin (RR)

The scheduling algorithm Round-Robin (RR) is specifically designed for time-sharing systems. We have a few queues in the simple Round Robin queuing and we can assign them traffic. This method does not use any major form of prioritization a packet from a queue is processed before moving to another queue and then repeats the process after it gets to the last queue. No queue gets priority but the bandwidth is divided according to the size of the queues that way all queues get a part of the bandwidth to use for processing [42] [47].

Round Robin scheduling is a pre-emptive first-coming schedule. In a first sequence, processes are dispatched. Each process can only run for a limited amount of time. This time interval is referred to as the time slice. If a process does not complete or is blocked within the time frame due to an I/O operation, the time frame expires and the process is pre-empted. The pre-empted process is located at the back of the running queue, where it has to wait for all the processes already in the queue to cycle through the Central Processing Unit (CPU) [42].

If a process gets blocked due to an I/O operation before its time slice expires, it enters a blocked because of that I/O operation. Once I/O operation completes, it is placed on the end of the run queue and waits its turn [42] [48]. Figure 2.15 illustrates RR Scheduling.

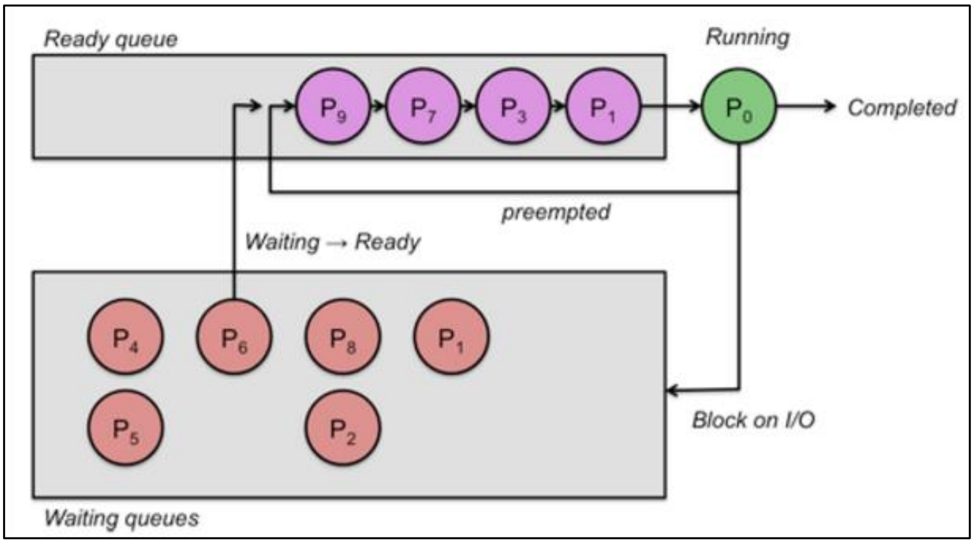


Figure 2.15. Round Robin Scheduling

A big benefit of Round Robin planning over non-pre-emptive planners is that it significantly improves the average time taken to respond significantly. The Round Robin model gives each task a certain amount of time. By limiting each task to a restricted time, the OS can make sure that it can carry out all ready tasks and that each task has the opportunity to perform. The obvious advantage of the Round Robin model is its easy implementation and fairness by giving the central processing unit (CPU) the same share (quantity) in each process. A short quantum is good because it enables many processes to circulate quickly through the processor, each process has a short chance of running [48].

On the other hand, due to its fairness, Round Robin scheduling drawbacks can also be observed by giving each process the same share (quantity) of the Central Processing Unit (CPU), highly interactive processes are not scheduled more often than CPU bound processes. Highly interactive jobs, which usually don't use their quantity, won't have to wait until they get the CPU again, thus improving interactive power. A short quantum, on the other hand, is bad because the operating system must switch the context whenever a process is pre-empted. This setting is spotted as an overhead. Overhead is defined as anything other than the user code execution of the CPU.

A short quantum means that many such context switches per unit time do not perform useful work in the CPU [48] [49].

2.10.3 Weighted Round Robin (WRR)

Weighted Round Robin (WRR) is a modified version of Round Robin (RR) with a slight modification of the same methodology. A weight is assigned to each queue and each queue receives an effective portion of the interface bandwidth on the basis of that weight, which might not be equal to the others [42]. Custom Queuing (CQ) is a Weighted Round Robin (WRR) example in which queues are processed based on the number of bytes configured before processing the next queue or before the next queue is allowed to be processed. Basic weighted round robin and custom queuing have a shared weakness if the weight assigned to a queue is similar in weight to the interface of the maximum transmission unit (MTU), the bandwidth division between the queues may not prove to be exactly what was planned [48] [49].

2.10.4 Fair Queue (FQ)

Fair Queue is a key alternative to FIFO and priorities. Where FIFO and its alternatives have a single input class and all incoming traffic is placed in a single physical queue, Fair Queuing keeps a separate logical FIFO sub queue for each input class. Fair Queuing gives equal shares to flows. Consider there are several competing flows at a router [34]:

```
at 12 packets per second:   flow1-------\
                                         \
at 4 packets per second:    flow2----------[R]----output
                                         /
at 2 packets per second:    flow3--------/
```

In a normal routing device, the output and input bandwidth allocation are proportional to each other. So, from the flow representation above, if the inputs 12, 4 and 2 are packets being sent for a total of 18 packets per second. The flow will only transmit 6, 2 and 1 packets per second if the output flow capacity is 9 packets per second. Thus, bandwidth is proportional to demand. This can be considered to be fair approach or a greedy one as the flow with the bigger number of packets are responsible for using up the bandwidth at the expense of others [34].

Fair Queuing tries to consider ensuring that all flows as show in the example above receive equal bandwidth based on the current demand. Flow 3 can easily be limited to 3 packets per second, but since flow 3 does not actually send 3 packets per second, the router processes 3 + 3 + 2= 8 packets per second, and there is idle capacity. It is essential that a queuing strategy is work-conservative and does not plan idle output times unless all inputs are idle [34] [42].

The router would allow flow 3 to send 2 packets / sec at this stage and divide the remaining 7 packets / sec of output flow equally between flow1 and flow2. This is shown as fair queueing [42].

The simplest Fair Queuing algorithm is the Round Robin Queue service with all packages of equal size. This means that a separate input queue is maintained for each flow and the non-empty queues in Round Robin cyclical fashion are serviced while empty queues do not bind resources. Each non-empty queue sends an equal share of packets over time. Assuming that all packets have the same size, every queue has the same bandwidth and chance. When certain flows are underused, Simple Fair Queuing allows other flows to exceed their equal share. Shares are equally divided between the active flows and as soon as the flow is active (i.e. its queue becomes empty) it begins to share in the allocation of bandwidth. The flow doesn't have to wait until other flows complete the backlog [42].

2.10.5 Weight Fair Queuing (WFQ)

Weighted Fair Queuing (WFQ) is a flow-based algorithm that schedules delay-sensitive traffic at the front of a queue to reduce response time and also fairly divides the remaining bandwidth between high-bandwidth flows. WFQ ensures that low volume traffic is transferred in a timely manner by breaking up packet trains. Weighted fair queuing gives priority over high volume traffic such as Telnet sessions, such as File Transfer Protocol (FTP sessions). Weighted fair queuing provides a balanced use of the connection capacity for simultaneous file transfers. Weighted fair queuing adapts automatically to changing conditions of network traffic [25].

Weight Fair Queuing (WFQ) is a simple but important queuing mechanism for two important reasons. First, weight Fair Queuing is the default queuing at 2.048 Mbps (E1) or lower speeds on serial interfaces. Second, WFQ is used by Class Based Weighted Fair Queueing (CBWFQ) and Low Latency Queuing (LLQ) two popular modern and advanced queuing methods. CBWFQ extends the WFQ standard to provide user-defined traffic class support. Traffic classes can be defined in CBWFQ on the basis of matching criteria, including protocols, ACLs and input [25] [42].

WFQ can prevent high bandwidth traffic from encumbering a network's resources. Overwhelming phenomena can lead to partial or complete failure of low-bandwidth communications in poorly managed networks during high traffic periods. WFQ has little or no effect on the speed at which narrowband signals are transmitted, but tends to slow down broadband transmission, especially during peak traffic times. Broadband signals share the resources which remain after the full transmission of low bandwidth signals. The sharing of resources is done by pre-assigned weights.

In flow-based WFQ, also known as standard WFQ, packets are classified into flows according to one of four criteria: The Transmission Control Protocol (TCP) or User Datagram Protocol (UDP), the IP address of the destination, the Internet Protocol address (IP address) or the TCP or UDP port of the destination. Each flow receives an equal [48] allocation of network bandwidth; hence the term is fair [25] [42].

Weighted Fair Queuing is enabled by default for physical interfaces whose bandwidth is less than or equal to T1/E1. Furthermore, there are four forms of WFQ:

1. Flow-Based WFQ
2. Distributed WFQ
3. Class-Based WFQ
4. and VIP-Distributed WFQ

In which the traffic is categorized into user-defined classes. Both of these forms of the VIP-Distributed WFQ and Class-Based WFQ operate according to principles similar to that of standard (flow-based) WFQ [42] [48]. Figure 2.16 shows Weight Fair Queueing schematic example.

Goals and objectives of WFQ includes [25] [42]:

1. To break up traffic into flows.
2. To ensure active flows get fair bandwidth allocations.
3. To ensure low-volume interactive flows get faster scheduling.
4. To ensure the flows defined as high priority get extra bandwidth. WFQ solves the issues of both FIFO and PQ.

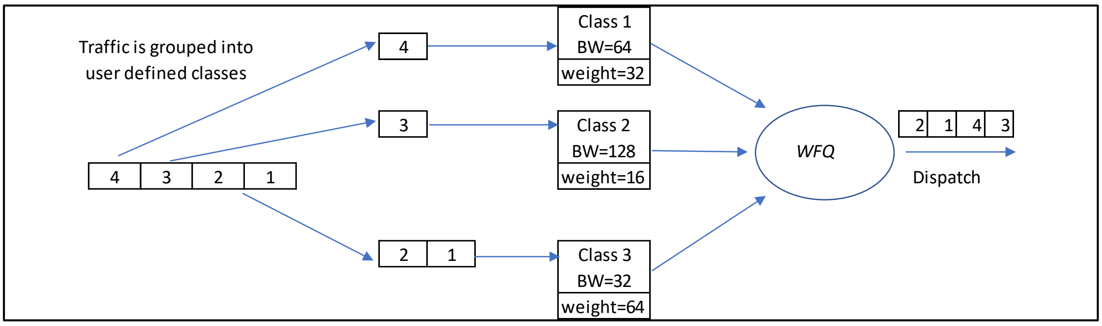


Figure 2.16. Weight Fair Queueing Schematic Example

2.10.6 Priority Queuing (PQ)

Priority queueing is defined as the collection of elements in which the next element to be removed from the queues is the element that has the highest priority of the whole element and was the longest element in the queue with the same priorities. Table 2.4 shows the comparison between different types of priority queue algorithms and Table 2.5 shows the priority queue levels for different traffic designation [42]. Priority queue can be structured in a number of configurations, including linear structure, binary heap, Pipelined van Emde Boas Tree, etc.

2.10.6.1 Linear Structures

Four simples but very powerful concepts are considered by linear structures. Stacks, lists, queues. Once an item has been added, it remains in that position compared to the other elements that came before and after. Such collections are often referred to as linear data structures. Linear structures can be considered to have two ends left and right or front and back or top and bottom [50]. A queue is an ordered collection of items where new items are added at one end, called the “back” and existing items are removed at the other end, commonly referred to as the “front” As an element enters the queue, it begins at the back and leads to the front, waiting until the next item is removed.

Table 2.4. Comparison between Different Types of Priority Queue Algorithms

Priority Queuing Models	Advantages	Disadvantages
Merging heaps	• Adding new values to it efficiently after initially constructing it. • Using clever algorithms, we can build a heap in time O(n) • The Heap sort algorithm exhibits consistent performance equally in the best and worst cases. • The Heap sort algorithm is simpler to understand than other equally efficient sorting algorithms since it does not use advanced computer science thus easier for programmers to implement correctly	• FindMin, DeleteMin and Insert only • does not support fast merges of two heaps • For some applications, the items arrive in prioritized clumps, rather than individually
Binomial Queue Algorithms	• Reduce the storage overhead of the structure • Increase the efficiency of operations • Allows any element of an unknown priority queue to be deleted in log time • Binomial Queues are designed to be merged quickly with one another • Using pointer-based design we can merge large numbers of nodes at once by simply pruning and grafting tree structures	• More overhead than Binary Heap, but the flexibility is needed for improved merging speed
Leftist Heaps	• Leftist heap priority queue uses the min heap data structure which supports operations such as insert, minimum, extract-min, decrease-key • leftist trees are maintained so the right descendant of each node has the lower s-value.	• O(log N) time for insert, deletemin, merge
Skew Heaps	• Skew heaps are advantageous because of their ability to merge more quickly than binary heaps • No structural constraints, so there is no guarantee that the height of the tree is logarithmic.	• O(log N) amortized time for insert, deletemin, merge
Calendar Queues	• O(1) average time for insert and deletemin	
List based PQ Unsorted list	• insertItem takes O(1) time • using an unsorted list implementation and performing a PQ sort with it results in a selection sort	• RremoveMIn, minKey and minElement takes O(n) time
List based PQ sorted list	• RremoveMIn, minKey and minElement takes O(1) time • using an unsorted list implementation and performing a PQ sort with it results in a insertion sort	• insertItem takes O(n) time

Table 2.5. Priority Queue Levels for Different Traffic Designation

Priority level	Traffic Designation
1	Best Effort
2	Excellent effort
3	Controlled Load
4	Video
5	Voice
6	Media data
7	Emergency

At the end of the collection, the most recent item added at the queue must wait. The item that was the longest in the collection is at the front. Sometimes this ordering principle is called first-in-first and also, known as first served [42] [50].

For example, a computer lab has 30 computers that are connected to a single printer. If students want to print, their printing tasks match all the other printing tasks that are waiting for them. The first task to complete is the next one. If you're the last in line, you have to wait and print all the other tasks ahead of you [42].

2.10.6.2 Binary Heaps

The classic way to implement a priority queue is to use a binary heap data structure. A binary heap allows us to drag and drop items in O(logn). The binary heap is interesting because it looks like a tree data structure when we diagram the heap. The trees data structure is used in many computer sciences, including operating systems, graphics, database systems and computer networking. A structure of tree data has a root, branches and leaves. The difference between a tree in nature and a tree in informatics is that it has its root at the top and its leaves at the bottom.

The binary heap has two common variations: the min heap, which always has the smallest key at the front, and the max heap, which always has the largest key value at the front [50].

2.10.6.3 Pipelined van Emde Boas Tree Algorithms

Priority queues are essential for various network processing applications, including quality-of-service (QoS) per-flow queueing, management of large fast packet buffers and statistical counter management. In [66], we propose a new data structure for the implementation of high-performance priority queues based on the van Emde Boas tree pipeline version [51].

We show that we can achieve O (1) amortized time operations using our architecture, but we can achieve this algorithmic efficiency using only O(log u) number of pipelined stages, where u is the size of the universe used to represent the priority keys [51].

Applications include advanced scheduling of QoS-based per-flow queues, management of large fast packet buffers and management of accurate statistical counters. The need to support extremely fast line rates is a key challenge in the implementation of priority queues. The speed of networks is growing rapidly. For example, a new entry can be inserted into the priority queue to support advanced scheduling on a 10 Gb / s (OC-192) link with a packet size of 40 bytes and an existing entry can be deleted once every 32ns [51].

In advanced QoS scheduling literature, a binary heap data structure is often assumed for priority queue implementation, which is known to have O(log n) time complexity for heap operations, where n is the number of heap elements. This algorithmic complexity, however, does not scale well with increasing queue sizes and is not fast enough for connection rates of 10 Gb/s or more [51].

2.10.7 Conclusion

From the above description of the current priority queuing algorithms being used in the communication industry, it became obvious that each queuing algorithm has its own capabilities and limitations in meeting different user and industry needs.

Hence, considering the eHealth industry stringent requirements and the importance of differentiating between real emergency and non-emergency cases with respect to allocating priority levels, none of the current queueing algorithms meet these requirements fully. Nevertheless, WFQ appears to be the closest algorithm to meet eHealth requirements yet it does not address real emergency situation.

Starting with FIFO, it is considered as the most primitive and data structure as it serves the first data in the queue without prioritizing them regardless of the nature of the data whether it is emergency or non-emergency. Another drawback of FIFO is the long-term process delay of the data behind the first data in the queue if it happens to be of a big size which could lead to starvation. Therefore, it is not suitable for interactive occupations such as eHealth taking into consideration emergency situations.

In the Round Robin algorithm, the time-sharing method being adopted under this algorithm does not use any major form of prioritization a packet from a queue being processed before moving to another queue and then repeats the process after it gets to the last queue. Thus, no queue or data is given a priority but the bandwidth is divided according to the size of the queues which allows that all queues get a part of the bandwidth to use for processing. RR scheduling is viewed as an advanced form of the primitive first-coming schedule where each process can only run for a limited amount of time regardless of the nature of the data within the process whether it is emergency or non-emergency. Therefore, RR algorithm is not the best solution for addressing emergency situations requirements in eHealth industry considering the tolerated delay limitation.

In Weighted Round Robin algorithm, the time-sharing method in RR is modified by adding weight to queues which allows each queue to have its portion of the bandwidth on the basis of the given weight. However, when the weight assigned to a queue is similar in weight to the interface of the maximum transmission unit (MTU), the bandwidth division between the queues may not prove to be exactly what was planned. Furthermore, the given weight in this process lacks the identification of real emergency situation in eHealth which depends not only on the senor type and criticality but also the patients' health profile.

In Fair queueing algorithm, FQ tends to give fair shares of bandwidth to all sub-queues and flows. Having the output and input bandwidth allocation to flows proportional to each other is considered greedy approach in some cases when flows with the bigger number of packets will be using up the bandwidth at the expense of others flows. Thus, in eHealth emergency cases FQ will not be so effective in addressing the priority requirement.

Finally Weight Fair algorithm is an advanced version of FQ where it gives weight to flows to reduce response time of sensitive traffic and also fairly divides the remaining bandwidth between high-bandwidth flows it also allow flows and traffic classification. Nevertheless, WFQ tends to prevent high bandwidth traffic from encumbering the network's resources, yet at overwhelming traffic, partial or complete failure of low-bandwidth communications can occur since WFQ has no control over the speed at which narrowband signals are transmitted.

WFQ appears to be the closest algorithm to meet eHealth scenarios yet it does not address emergency situation tolerated delay and lacks the identification of real emergency situation in eHealth which depends not only on the senor type and criticality but also the patients' health profile. Thus, our new Priority Fair-based Queuing (PFQ) algorithm simulation and data analysis is compared to WFQ results under same scenarios to evaluate the effectiveness of the PFQ and the improvement it achieves in reducing delay and jitter.

2.11 Related Works and Researches

1. In a Queuing Theory and Birth-Death process is used to develop a mathematical model that can reduce the energy consumption and transmit important data packets in less time [52].
    a. MatLab (version 7.11.0.584) has been used to experiment. Five parameters: data input rate, service rate, collision rate, threshold and queue size were defined. Four system states were also used: sleep condition, idle state, busy state and state of transmission.
    b. In this experiment, in each system state, the parameters value is changed to obtain the result of the energy consumption.
    c. To reduce the delay, a dual priority queuing system, two priority levels, high and low, was used to ensure the timely transmission of critical data.
    d. The shortcoming of this paper is the absence of the mechanism that been used to assign the packets to the priority levels.
2. In a DTD-MAC protocol, CSMA / CA (Carrier Sensed Multiple Access / Collision Avoidance) based environment (Carrier Sensed Multiple Access / Collision Avoidance) was proposed in medical signal monitoring to reduce time delays and packet losses in the TDMA multiple access time division [53].
    a. The proposed MAC Protocol assigns each node with its priority level

a guaranteed time slot. It also has a PSAP (pre-slot assignment period) which delays the application of a requirement.

b. When any node, regardless of its original priority, reaches the maximum delay of 250ms, it is considered a high priority in the proposed protocol and allocated a channel.

c. The shortcoming of this paper is, in one point it will treat the low priority nods as high and give it priority to possess its data over the real high priority emergency data.

3. The deficiency of this paper is that it treats the low priority nodes as high in one point and gives it priority to have its data above the real high priority emergency data [54].

   a. Eight WBAN (Ups) user priorities are defined and grouped into four categories of access to WLAN (ACs). Evaluation of the performance of the healthcare network using two AC differentiation mechanisms in the WLAN; CW size and AIFS.

   b. Priority differentiation is achieved through tow parameters: (AIFS arbitrary inter-frame space) the amount of time per station since the channel is idle before transmission and (CW) the length of the window of contention.

   c. The results of this work indicate that an aggressive differentiation leads to a lower overall performance of the network. Small CW sizes increase the probability of collision for the competing nodes and trigger early saturation in the WLAN. This would lead to greater contention in the second hop and lead to large delays in the end- to- end frame. Moreover, the saturation in the WLAN causes buffer overflow in the bridges (WBAN hubs), leading to an undesirable loss of medical data.

This work confirms that AIFS is better suited to the differentiation of WLAN ACs in order to achieve moderate differentiation and lower frame collision probability.

d. The short outcome of this paper, the simulation was done under low to medium traffic only.

4. In the WBAN eHealth data transmission delay is calculated [55].
   a. In ZigBee-based WBANs with the coexistence of the Wi-Fi network, an adaptive load control algorithm is proposed to overlap between the ZigBee and the Wi-Fi channels. This overlap increases the ZigBee packets ' delay.
   b. The key signals are categorized into two types: information collected by regular (saved and transmitted after a period of time) and emergency messages (saved immediately).
   c. By controlling only, the Wi-Fi traffic with the aim of guaranteeing that delay by ZigBee sensors do not exceed 1 second.
   d. Data from ZigBee sensors was then sent to the IP cloud via the Wireless Local Area Network (WLAN), Worldwide Microwave Access Interoperability (WiMAX), Universal Mobile Telecommunication System (UMTS).
   e. Three different communication paths are used to send data to healthcare centers.
   f. Finally, the data reaches the healthcare center from the IP cloud.
5. In an Urgency-based Priority Queuing (UPQ) is proposed. It depends on two tables, one is the scheduling priority index consisting of the scheduling

sequence and the other is the emergency length table, which the packet number served in a scheduling time for each queue [56].

   a. The idea is to control and keep the flow delay below the deadline by analyzing both flows and packet numbers in the delay organization.
   b. The downside in this proposed method includes additional bandwidth costs for collecting delay information. In this work, the balance between cost and performance is not applied.

6. In this scheme, the traffic in WBAN is classified to real-time and non-real-time traffic, to achieve low latency of the real-time traffic in WBAN [57].
   a. It is focused only in reducing the delay in the WBAN side between the sensors the PDA.
7. In analyzing the similarities and differences in the multi- task scheduling and multi- data transmission of the real- time system, priorities are dynamically assigned to different data according to their criticality. Historical data transmission situation, speed transmission and time gap [58].
   a. It is applied to data transmission in the environment of non-stable communication link.

Chapter 3: Model Design and Simulation

3.1 Introduction

This chapter aimed to explain the proposed scheduling model named Priority- based Fair queuing (PFQ) in minimizing the delay and jitter when transmitting data from eHealth applications in emergencies. It also aimed to describe the simulation testbed that has been used to test the model and collect the data.

Reducing the delay and Jitter which in turn improve the M2M QoS required redefining the priority queuing and the data classification and look into the situation from different angle. Our Methodology was divided into two main perspectives. First is to develop a new Priority Queuing model and second is to build a simulation program to run and test the model and compare the results with the current models.

The limitation found in other studies and current Priority base solutions are:

1. The high priority level given to a packet in the current priority queuing models is based on the sensor's priority for example if the sensor is collecting heart rate, then all the packets sent from this sensor is considered high priority. In reality, not all heart rate readings are emergency cases. Should the heart rate read within the normal average or range, the packet should not be treated as an emergency packet.
2. Similarly, current algorithms and priority queuing models treat all packets that fall in the range of the high reading as emergency packets and gave them same high priority. For example, if the temperature reading is between 39 and 42 degree cellulous, it will be given high priority according to the current models. However, in reality, those readings could be different in their emergency situation if we consider the patient profile such

as: age, gender, history and pregnancy etc. Pregnancy and elderly cases seem more emergent than gender for instance.

3. Most of the studies had proposed solutions for the delay within the WBAN only i.e. from the sensors to the gateway, but they did not deal with the full transmission from the sensors to gateway and then from the gateway to the hospital or the healthcare center.

In our proposed model, we have considered all the above observations and limitations and found a dynamic solution that dramatically reduces the delay and jitter in emergency packet from sensor to sink.

Different scenarios of group of elderly patients living in a care center shall be considered in building the simulation. Each patient has three different sensors attached or imbedded in his body such as heart rate, Blood Pressure and Glucose. Those sensors send vital data regularly to the health care center. The first scenario is that one patient will send three readings one from each sensor simultaneously. In the second scenario four patients will send twelve readings one from each sensor simultaneously. In the third scenario eight patients will send twenty-four readings one from each sensor simultaneously and in the fourth and last scenario twelve patients will send thirty-six readings one from each sensor simultaneously.

Those scenarios were tested and simulated based on the newly proposed priority model and was run also on current priority queuing models such as FIFO and WFQ to compare the results of the Delay and Jitter parameters for emergent and non-emergent cases.

Therefore, this chapter is divided into two main sections that describe the newly elaborated PFQ model and its simulation testbed. In the first section the PFQ model is mainly based on scheduling priority models like Priority Queuing (PQ), Fair

Queuing (FQ) and Weighted Fair Queuing (WFQ) models. So, we first outline how these scheduling models work, and what are the deficiencies that we consider in PFQ model. A workflow diagram has been presented to show the procedure of PFQ model. This workflow also depicts the main parts of the PFQ model: the PFQ algorithm and the Priority Parameter (PP). This section also provides the details of Priority-based Fair Queuing with Tolerated Delay (PFQ-TD) model.

Simulation testbed section provides an overview of the simulation setup. It also presents the model that has been used to build the simulation. This section further describes simulation parameters and scenarios.

3.2 Queuing Scheduling Models

As IP protocol-based internet should support various types of services such as email, file transfer, real time and video streaming services, the traffic characteristics of these applications require high quality of services with regard to delay and bandwidth parameters especially with the exponential growth of the Internet of Things (IoT) in the last decade. The limited network resources such as bandwidth has to be efficiently managed and shared to address the different requirements of the users and to maximize the performance. This can be achieved by altering the behavior of the routers with regard to packet handling including classification, shaping, queuing, scheduling, prioritization, admission and discard.

Packet classification differentiates between different types of traffic, i.e. routers firstly classify the packets into flows where all packets that belong to the same flow are processed in a predefined manner by the router. Once classification is done admission control process checks the availability of the network resources and packets are handled according to their service classification.

Afterwards traffic shaping process controls the traffic volume of packets entering the network and the rate at which packets are sent. Packets are then scheduled into queues and managed in a way to ensure each queue gets the level of services required for its class. A number of queuing techniques and models were developed and widely used such as First-In-First-Out (FIFO), Priority Queuing (PQ), Fair Queuing (FQ) and Weighted Fair Queuing (WFQ).

The mentioned queuing models vary in the way they handle the packets transfer whereas; the FIFO is considered the simplest and most straight forward technique. Advanced studies and researches led to the development of more sophisticated models such as PQ from which FQ had evolved followed by WFQ and so on to meet the demanding environment. Our proposed model, PFQ, is developed based on the concept of the priority and fairness queuing models. Thus, in this section we elaborated on these models and highlighted the gap that has been taking into consideration in PFQ model.

3.2.1. First in First Out (FIFO) Model

First In First Out (FIFO) queues are among the most basic and fundamental highly studied simultaneous data structures. FIFO queues are an important building block of concurrent data structure libraries. FIFO scheduling is simple to implement. It is also intuitively fair (the first one in line gets to run first). However, the greatest drawback of first come first served scheduling is that it is not proactive. Therefore, it is not suitable for interactive jobs. Another drawback is that a long-running process will delay all jobs behind it.

First In First Out (FIFO) queues is the most straightforward approach to scheduling processes. New processes go to the end of the queue and when the scheduler needs to run a process, it picks the process that is at the head of the queue. This scheduler is classified as non- proactive. If the process has to block on Input / Output (I/O), it enters the waiting state and the scheduler picks the process from the head of the queue. When I/O is complete and that waiting (blocked) process is ready to run again, it gets put at the end of the queue.

FIFO is a simple algorithm that requires no configuration effort. Packets line up in a single FIFO queue; packet class, priority, and type play no role in a FIFO queue. Without multiple queues and without a scheduling and dropping algorithm, high-volume and ill-behaved applications can fill up the FIFO queue and consume all the interface bandwidth. As a result, other application packets for example, low volume and less aggressive traffic such as voice might be dropped or experience long delays. On fast interfaces that are unlikely to be congested, FIFO is often considered an appropriate queuing discipline.

First In First Out FIFO mechanism is well visualized in a printer example. As the printer can only deal with one thing at a time it has a built-in queue, queuing up multiple jobs for printing and taking those jobs in the order that they were submitted. Chapter 2 of this research gives more details on FIFO algorithm. Figure 3.1 describes the FIFO mechanism process [61]

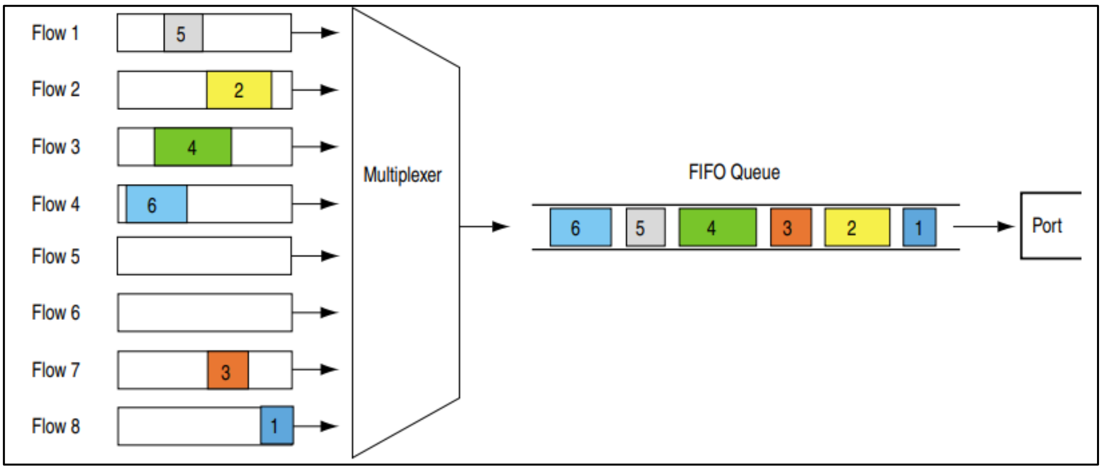


Figure 3.1. First In First Out (FIFO) Workflow

3.2.2. Priority Queuing (PQ) Model

Priority Queuing (PQ) is a queuing scheduling mechanism where each packet is marked with a priority or class. That is, it uses multiple queues in which packets are placed in one of the queues according to their classification or priority. Each queue has its own priority level and queues might have different priority levels. Queues with higher priority are served first. Packets are scheduled in FIFO order within each priority queue only if the higher priority queues are empty. In case of congestion packets are dropped from lower priority queues Figure 3.2 shows the priority queuing mechanism with two priority levels [62]

PQ algorithm has many types and can be structured differently to serve different purposes such as Merging Heaps, Binomial Queue Algorithms, Leftist Heaps, Skew Heaps, Calendar Queues, List based PQ Unsorted list and List based PQ sorted list. Each structure or type has its advantage and disadvantage. In general, the advantage of the Priority Queuing algorithms in some application and as needed is that higher priority queues yield lowest delay and jitter, and highest bandwidth. In other words, the nodes can be weighted, allowing those with greater precedence to be moved towards the head of the queue, in front of those with lesser priority, rather than always being added to the tail of the queue as would happen in a normal queue such as FIFO.

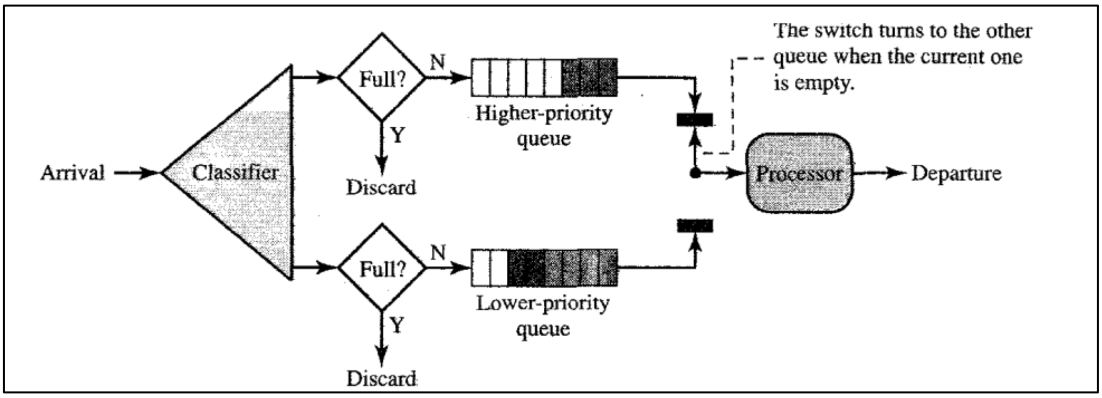


Figure 3.2. Priority Queuing (PQ) Workflow

The disadvantage of the PQ is that the lower priority queues suffer from starvation if the volume of the higher priority queue is excessive. Having the high priority queue served first all the time, the queues allocated to lower priority traffic may overflow and experience a large delay and in the worst scenario, called complete resource starvation, until all the high priority queue is served. Hence adopting the PQ algorithm in developing the new proposed PFQ algorithm is not feasible since it will not full fill the objective of having a balanced and fair priority queuing methodology. Chapter 2 of this research gives more details on PQ algorithm and its different types and forms.

3.2.3. Fair Queuing (FQ) and Weighted Fair Queuing (WFQ) Algorithms

Fair Queueing (FQ) and Weight Fair Queueing (WFQ) algorithm have been introduced to overcome the shortcomings of the PQ where one flow with high priority level can dominate and use all the available bandwidth of the network and hence, lead the lower priority flows to bandwidth starvation.

According to the Fair Queuing algorithm packets are classified into different flows and stored in queues dedicated to that particular classified flow. One of the best-known Fair Queuing algorithm is the Round Robin (RR) algorithm that is used to service all the queues, so that queues can be served in a fair way and one flow cannot use more than its share of network bandwidth. FQ is good to share the same portion of bandwidth among many flows, but it cannot be used for handling different flows bandwidth requirements or provide real-time services.

A big advantage of Round Robin scheduling over other schedulers is that it significantly improves average response times by allocating certain amount of time to each task thus the operating system can ensure that it can run through all ready tasks, giving each task a chance to run. Another obvious advantage of Round Robin model is its easy implementation and its fairness by giving an equal share (quantum) of the Central Processing Unit (CPU) to each process. On the other side Round Robin scheduling drawback could be observed through its fairness also, by giving each process an equal share (quantum) of the Central Processing Unit (CPU), highly interactive processes will get scheduled no more frequently than CPU bound processes. Highly interactive jobs that usually do not use up their quantum will not have to wait as long before they get the CPU again, hence improving interactive performance. Figure 3.3 shows RR workflow mechanism where RR basic formula is implemented. Chapter 2 of this research gives more details on RR algorithm and its different types and forms.

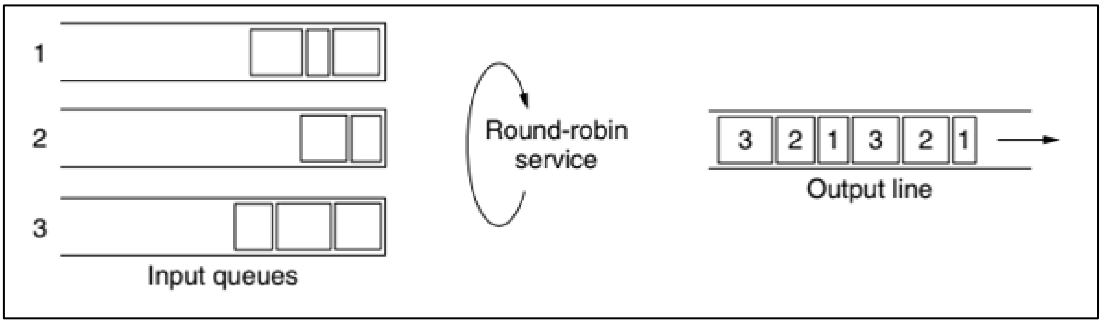


Figure 3.3. Round Robin (RR) Workflow

Round Robin (RR) model formula:

$$Fi = max\ (A_i, F_{i-1}) + L_i \qquad \text{Formula 3.1}$$

where

F_i is the *Finishing Time* of the packet *i* in the queue

A_i is the *Arrival Time* of the packet *i* in the queue

F_{i-1} is the *Finishing Time* of the packet *i-1* in the queue

L_i is the *Length of* the packet *i* in the queue

The Weighted Fair Queuing (WFQ) algorithm adopts a combination of PQ and FQ algorithms. Similar to the FQ algorithm, all queues are served so that there is no bandwidth starvation, but in WFQ some queues are assigned more weight so that they can receive more service according to their packet classification. In other words, a weight is given to each queue to assign different priorities to the queues and packets are stored into the appropriate queue according to their classification. Weights in FQ is given based on one parameter such as the application or sensor type when dealing with eHealth M2M packets i.e. the weight are singular and kind of pre-set values. For example, higher weights are given to heart rate application followed by blood pressure, diabetes etc. regardless of the patient's age, gender, medical history or situation.

Priority term has been clarified in Priority Queuing section. Weight as defined above represents the number of bytes being processed per round. For example, in WFQ the packets are assigned to different classes and sorted in different queues. The Queues are weighted based on the priorities of the queues where higher priority means a higher weight. Letting the number of bytes pe round be the weight of a flow 'W' the formula for computing the finish time in WFQ is defined as follows:

Weighted Fair Queuing (WFQ) Formula:

$$Fi = max\ (A_i, F_{i-1}) + L_i/W \qquad \text{Formula 3.2}$$

where

F_i is the *Finishing Time* of the packet *i* in the queue

A_i is the *Arrival Time* of the packet *i* in the queue

F_{i-1} is the *Finishing Time* of the packet *i-1* in the queue

L_i is the *Length of* the packet *i* in the queue

W is the *weight of* the packet

The system processes packets in each queue in a RR style with the number of packets selected from each queue is based on the corresponding weight. For example, if the weights are 3, 2 and 1, three packets are processed from the first queue, two from the second queue, and one from the third queue. If the system does not enforce or impose priority on the classes, all the weights can be equal. Therefore, in this way, fir queueing is achieved. Figure 3.4 shows this technique with three classes.

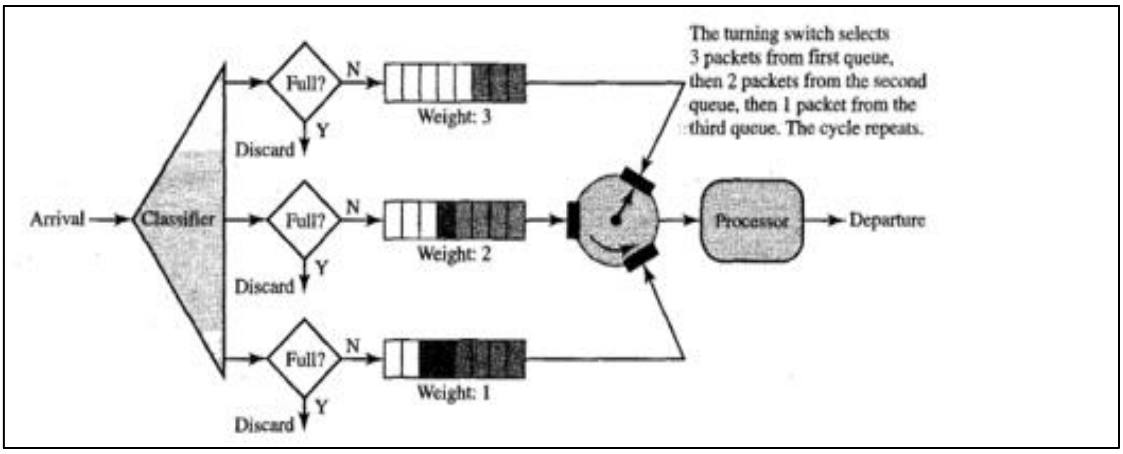


Figure 3.4. Weight Fair Queueing (WFQ) Workflow

WFQ algorithm can have different forms such as Flow-Based WFQ, Distributed WFQ, Class-Based WFQ and VIP-Distributed WFQ. In general, the advantage of the WFQ is that it assures that low-volume traffic is transferred in a timely fashion and gives low-volume traffic, such as Telnet sessions, priority over high-volume traffic, such as File Transfer Protocol (FTP) sessions. It also gives concurrent file transfers balanced use of link capacity and automatically adapts to changing network traffic conditions.

The Main drawback of the WFQ algorithm is it has slight or no effect on the speed at which narrowband signals are transmitted and tends to slow down the transmission of broadband signals, especially during the peak time of network traffic. Chapter 2 of this research explains FQ and WFQ in more details.

The aforementioned queuing models cannot guarantee the strict real time intolerant delay and minimal jitter of data transmission in emergency and life-threatening situation. This is because these mechanisms lack the adoption of the real time change of emergency cases in M2M eHealth ecosystem. Consequently, an intolerant delay and stream disruption of jitter are expected.

Reaching this advance stage of queueing, it adopts the FQ and the advance version of the WFQ techniques and developed the Priority based Fair Queuing (PFQ) and Priority-based Fair Queuing with Tolerated Delay (PFQ-TD) algorithm to address the emergency packets priority transmission and provide fair and balanced traffic without causing network starvation for non-emergency packets.

3.3 Priority-based Fair Queuing (PFQ) Model

Unlike current priority Queuing models such as WFQ and FIFO, we have set the criteria for two new priority queuing algorithms namely ‘Priority-based Fair Queuing’ (PFQ) ‘Priority-based Fair Queuing and Tolerated Delay’ (PFQ-TD) and prioritized the emergent and non-emergent data based on the patient profile and the sensor type, rather than the sensor type only.

Including the patient’s health profile in the prioritization criteria had added great value to the emergent cases since a heartbeat sensor for instance can be transmitting data from a 30 years or 60 Years old patient, as such prioritization should not be taken form the sensor types only but also from the patient’s profile and conditions.

PFQ is basically a scheduling method that dynamically controls the queuing priorities according to the QoS needed. That is, PFQ schedules packets according to their Priority Parameter (PP) value. Figure 3.5 shows the workflow of the PFQ scheduling model. The model has two parts, PP value and PFQ algorithm. PFQ algorithm is implemented in the Local Data Processing Unit (LDPU), whereas PP value is calculated at the sensor node.

One of the main advantages of the PFQ and PFQ-TD algorithm that is not found in any of the existing algorithms such as PQ, FQ and WFQ that the new proposed algorithms can be integrated and the router or switches can use them alternatively depending on the incoming traffic volume. Hence flexibility feature is added to the system and on demand model becomes available that best fit the network situation. For example, if the network is having low incoming traffic volume while emergency packets are to be transmitted, PFQ queuing model will be used to ensure extra quick transmission of the emergent packets. In case of high incoming traffic volume and while emergency and non-emergency packets to be transmitted, PFQ-TD queueing model will be used to ensure relatively and acceptable quick transmission of emergency packets and provide fair traffic distribution and avoid starvation for non-emergency packets.

3.3.1. Priority Parameter (PP)

Many researches such as those in [1] [2] prioritize each application or sensor based on its data type. This is because sensor data are different in terms of time- tolerance, importance, and traffic intensity. For example, the packet of ECG application is different than the one of blood pressure and temperature with respect to time-tolerance and traffic intensity.

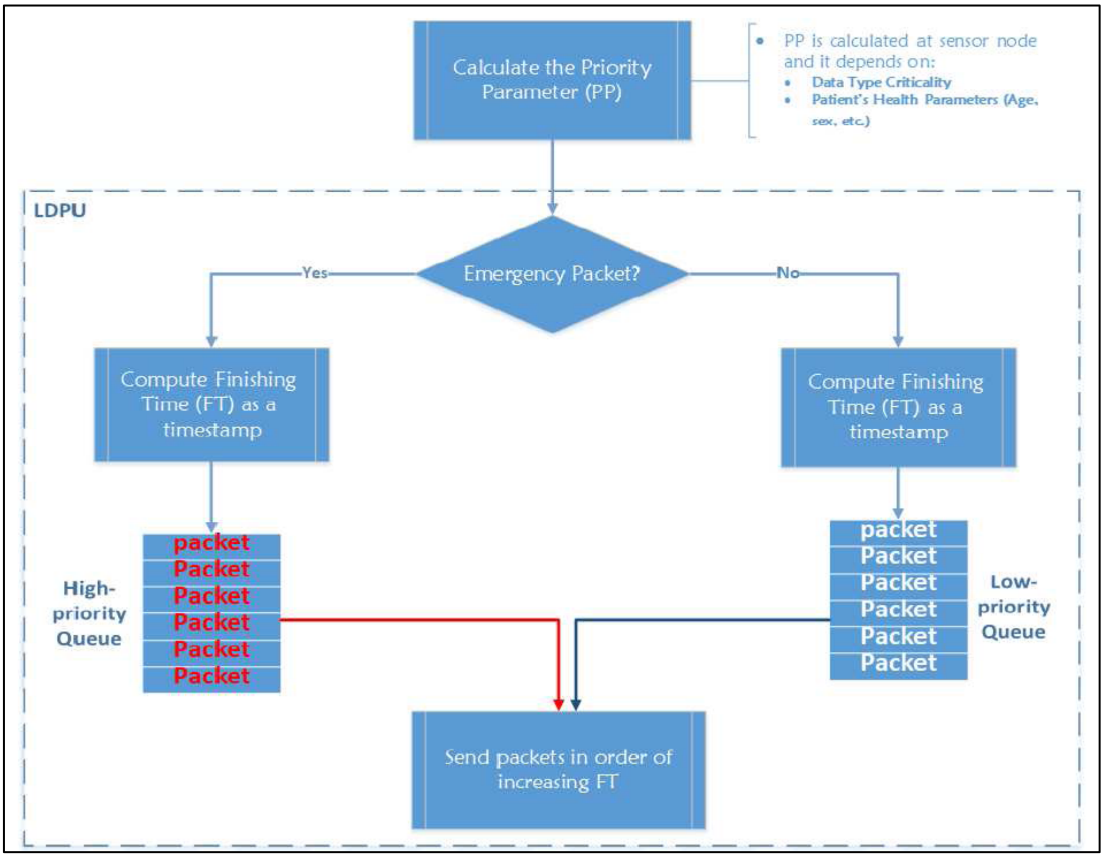


Figure 3.5. Priority Fair Queuing (PFQ) Workflow

From the above point, we came up with the idea of the PP. Definition: Priority Parameter (PP) is value that relies on two factors, data type criticality and Personal Health Record (PHR). Point I and II below define the components of the PP used in developing the priority algorithm.

I. Data Type Criticality

In eHealth, different sensors generate different vital data with different criticality. The criticality here simply refers to the importance of the data to the patient life. In our proposed algorithm, data criticality is one of the priority parameter (PP) factors that helps prioritizing the transmission of the data. For example, the packet of ECG application is different than the one of blood pressure and temperature with respect to time-tolerance and traffic intensity. Table 3.1 below shows an example of user priorities or criticality for some data types of eHealth systems presented in [1].

Table 3.1. Data Types and Their Priority

Data Type	User Priority
ECG	7
Blood Pressure	5
Glucose	4
Temperature	4
EMG	3

II. Personal Health Record (PHR)

Definition: Personal Health Record (HR) is a unique identifier that has the patient current and previous health status background and information.

Nowadays almost each one of us has his/her own PHR. PHR contains patient's personal information such as name, age, gender, address, etc and the medical history, (e.g. the previous heart stroke or high-risk pregnancy). PHR can easily be viewed and updated from any device that a health care clinic, hospital or a doctor uses to store the patient's health history and treatment information. PHR offers a massive amount of benefits to the patients and the eHealth system. It helps the health care staff to know the complete patient's health history and provide the right care anywhere any time. It is authorized the patients to access and monitor their own medical status.

In our proposed algorithm, we take advantage of the existing PHR and use the patient's personal and health history information such as age, gender, pregnancy, previous heart shock or attack, etc. In the PFQ model, PHR is used as one of the priority parameters (PP) that indicates the criticality of the transmitted vital data, and gives different priority according to their criticality. Table 3.2 shows different parameters and its critical value.

Table 3.2. PHR Parameters and Their Values

Parameter	Details	Priority value
Age	50+	+2
	50–	+1
Gender	Male	+1
	Female	+1
	Female (pregnant)	+2
Previous heart shock	Yes	+2
	No	+1

3.3.2. Priority-based Fair Queuing (PFQ) Algorithm

PFQ algorithm is the core of the PFQ model. It is responsible for scheduling the packets sent from sensors nodes at the LDPU. This algorithm is similar to FQ and WFQ algorithms in the sense that it employs the fairness principle. However, it considers the fact that each sensor node or application has a different priority.

At LDPU, the packet is checked if it is emergency one or not. Emergency packets are queued in the high-priority queue, whereas non-emergency packets are queued in the low-priority queue. Packets in the high-priority queue are always served first. Hence, emergency packets are served first. PFQ algorithm prioritizes packets in both queues according to the following formula:

$$FT_i = \max(FT_{i-1}, AT_i) + \frac{PacketSize}{PP_i} \qquad \text{Formula 3.3}$$

where

FT_i is the *Finishing Time* of the packet *i* in the queue

AT_i is the *Arrival Time*

PP_i *is the priority parameter that is set in the sensor node*

FT is calculated independently for each queue (i.e., high-priority and low priority). Schedulers use FT as a priority value for packets to choose which packet should be served next. PFQ algorithm then sends packets in order of increasing FT. That is, packets with low FT will be served first.

3.3.3. Priority-based Fair Queuing with Tolerated Delay (PFQ-TD) Algorithm

PFQ-TD is an algorithm that incorporates the Tolerated Delay (TD) mechanism to Priority-based Fair Queuing algorithm. TD refers to the maximum delay that is acceptable for a packet to reach its destination. As a matter of fact, PFQ is expected to increase the delay and jitter of the non-emergency packets due to the high priority assignment for emergency packets. Consequently, these non-emergency packets become useless since they exceed the TD.

So, in PFQ-TD emergency packets are not given priority over non-emergency packets to have fair and balanced delay distribution. However, TD is included to consider the priority of the emergency cases over the non-emergency cases. To illustrate, the authors in [2] assign 50ms of the TD for emergency packets and 150ms for non-emergency packets. Hence, if two packets, one is emergency and the other is non-emergency, has same value of PP and Finishing Time (FT) or Arrival Time (AT), the emergency one will be served first because of its TD value.

Figure 3.6 illustrates the workflow of the PFQ-TD. This algorithm is similar to PFQ algorithm in calculating the FT of the packet. The only difference is the inclusion of the TD parameter. It prioritizes packets according to the following formula:

$$FT_i = \max(FT_{i-1}, AT_i) + \frac{PacketSize}{PP_i} + TD_i \qquad \text{Formula 3.4}$$

So, *TD* has a direct effect on the *FT* of the packet. That is mean, packets with less *TD* will have a chance to be served before those with high TD, if they have same packet size and PP value.

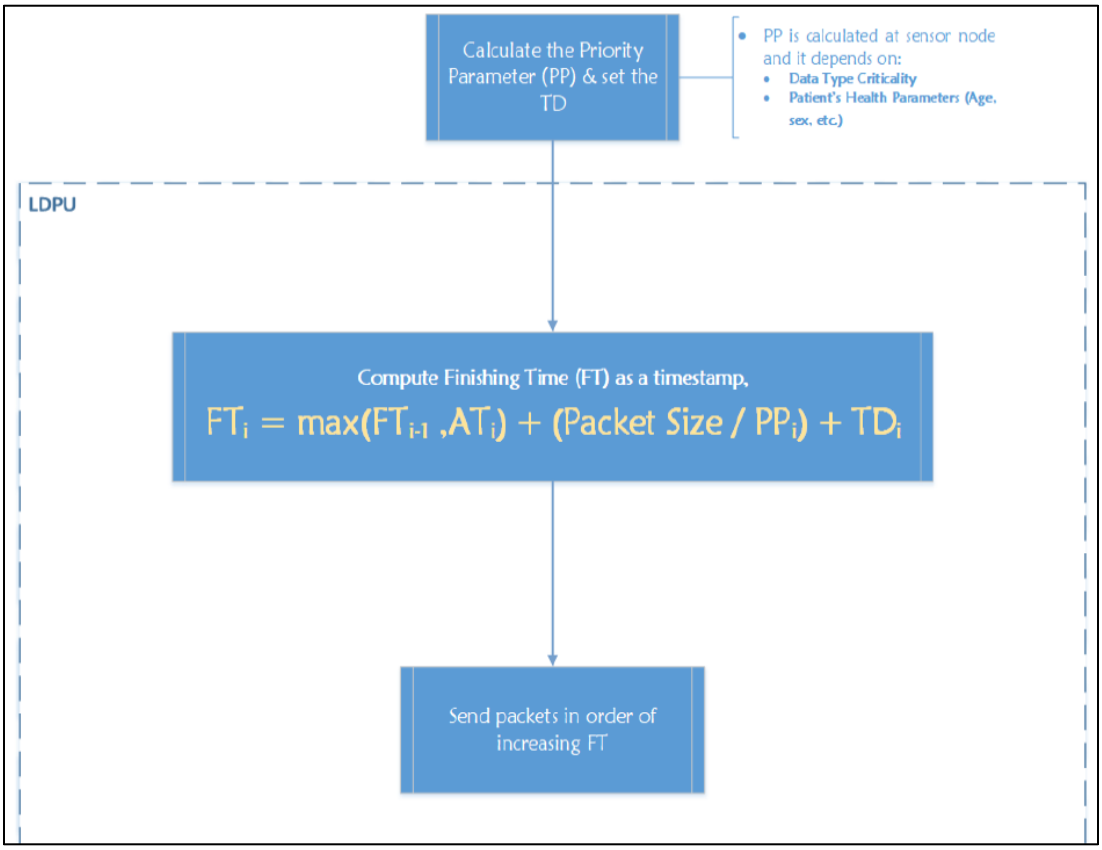


Figure 3.6. Priority Fair Queuing with Tolerated Delay (PFQ-TD) Workflow

3.3.4. Net Neutrality Principle and Priority-Queuing Models

Net Neutrality principle and law set by the USA Federal Communication Commission (FCC) in 2015 dictate that the Internet Service Providers (ISP) and broadband companies to treat all data without discrimination.

The rules prohibited the following practices; Blocking where ISPs could not discriminate against any lawful content by blocking websites or applications, Throttling where ISPs could not slow the transmission of data because of the nature of the content, as long as it was legal and lawful and Paid Prioritization where ISPs could not create an internet fast lane for companies and consumers who pay premiums, and a slow lane for those who does not buy this service.

That’s to say ISPs are not allowed to blocking or censorship; throttle, slow down or prioritize lawful data transmission. This law has ensured that giant search engine, entertainment, e-commerce companies etc. shall not be favored over smaller companies in terms of speeding up the traffic going to their web sites [63].

However, in June 2018 FCC officially ended network neutrality rules which opens the door for ISPs to create and sell fast internet lanes on the web for giant companies and websites who can pay premium to the ISPs. Ceasing the Net Neutrality law shall change the landscape of the internet and data transmission protocol and shall fall into legitimate debates and business ethics dilemma.

The end of the Net Neutrality era might have a negative effect on the M2M ecosystem data transmission nevertheless eHealth as the ‘fast or paid lanes’ and the prioritization that could happen by the ISPs and the broadband companies might slow down the eHealth M2M data transmission emergency and non-emergency data unless health organizations and medical centers who implement M2M concept buy such lanes.

This will jeopardize monitoring the health and saving the lives of the elderly people and patients who use M2M and require immediate and continuous attention as the QoS in terms of delay, jitter and speed shall be affected due to the priority that shall be given to fast lanes especially video traffic which will burden the network and makes it increasingly congested.

To continue providing adequate QoS, network operators and service providers can build more infrastructure; yet this will require huge investments to deal with the enormous growth expected in traffic. However, traffic management can be a parallel solution to make the systems more efficient, while also setting restrictions on the amount of data that can be sent, and who gets priority as a sender or receiver.

While Net Neutrality law was in effect, Priority Queuing models were attending the data transmission prioritization and transmission from the source to the sink in a fair and in-discriminated world wide web. Yet with the end of the Net Neutrality law and the possibility of censorship on data transmission by the ISPs and the so-called fast lanes in the near future, Priority Queuing models shall be deemed more significant and important in ensuring that data is prioritized to be sent at least from the source to its destination.

3.4 Simulation Testbed

In this section, we describe the details of the simulation testbed that has been used to test the proposed model.

3.4.1. Simulation Setup and Performance Evaluation Overview

We developed a simulation model to evaluate the effectiveness of PFQ and PFQ-TD algorithm, in terms of packet delay and jitter. We further study their effectiveness against First Input First Output (FIFO) and Weighted Fair Queuing (WFQ) Models. The simulation was built using Python programming Language [3]. We used Python to build our simulation because it is easy and quick to learn. More importantly, Python has SimPy library [4], which is a discrete-event simulation environment. Thus, we made use of this library to build our simulation.

We took the benefit of Grotto's model named SimComponents.py [2], which has a set of components to create a network simulation. SimComponents.py is basically developed based on SimPy library.

Table 3.3 shows the parameters that have been used in the simulation and Figure 3.7 shows the basic components of our simulation. They are Packet Generator, Local Data Processing Unit (LDPU), and Packet Sink. Packet Generator simulates sending packets with a specified inter-arrival time and packet size. At LDPU component, we model our algorithm such FIFO, WFQ, PFQ, etc. Packet Sink records the arrival time information of the packet. We created 3 packet generators, ECG, Glucose, and Pressure. These packet generators (i.e., sensors) are imbedded in each patient.

Table 3.3. Simulation Parameters

Parameter	**value**
Arrival time	1 packet per second
Packet size	133 bytes
LDPU data rate	40 kbps
Emergency packets	30%
Non-Emergency packets	70%
Max no. of patients	12
Simulation time	10000 s

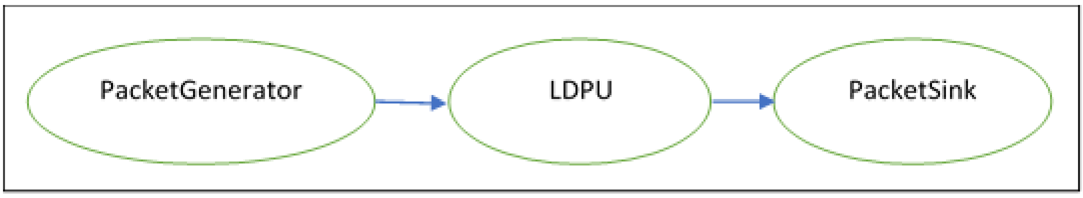


Figure 3.7. Simulation Components

In our Simulation, we have used a constant arrival rate (i.e., 1) with exponential interarrival time. The packet size was 133 bytes, which is ZigBee packet size. The LDPU data rate was 40 kbps (ZigBee bit rate) [3]. Normally, emergency packets are less than non-emergency packets, so that we put more percentage for the latter. It can be varied. We relied on the number of patients to reach the peak time. We started with 1, then 4, then 8, until 12 patients were each patient is sending 3 signals one from each the Heart rate, Blood pressure and Glucose sensor attached to his or her body. This makes the signals being transmitted reach 3, 12, 24 and 36 signals respectively.

Based on our simulation setup and results analysis, we found that all algorithms suffer from Congestion at 12 patients.

As proven in the first simulation (simulation number 1) i.e. in first scenario and sub scenarios, PFQ superseded WFQ and FIFO and achieved lower delays and Jitter for Emergent cases since PFQ gave high priority for emergency cases all the time by using the second formula in the PFQ algorithm. However, the PFQ did not address any improvement in the non-Emergent delay or jitter due to the high priority given to the emergency cases all the time.

By running another simulation (simulation number 2) under scenario 1, PFQ-TD algorithm has considered this draw back and did not give the emergency cases high priority all the time. The PFQ-TD has considered the Finishing time or non-emergent cases and gave it fair consideration over emergent cases thus balancing the outcome and gave fair and balance delay and jitter distribution for emergent and non-emergent cases.

In the new proposed algorithms, each packet in the eHealth system has a priority and it is based on both, the application and sensor type criticality such as heart rate, blood pressure and glucose which monitors the sugar level of the blood and the Patient Health Record (PHR) such as age, health history, gender, pregnancy etc.

PFQ model has been simulated for emergency cases only (simulation number 3) under scenario 2 and sub scenarios i.e. different levels of traffic volume from low to high. Results have shown the effectiveness of the PFQ model under different traffic volume scenarios in emergency cases. We started with 1, 4, 8, 12, 20, 30, 40 then 50 patients. Each patient sends 3 signals one from each the Heart rate, Blood pressure and Glucose sensor attached to his or her body. This makes the signals being transmitted reach 3, 12, 24 and 36, 60, 90, 120 and 150 signals respectively. This scenario is mainly focusing on the performance of PFQ in emergency cases under high traffic volume (i.e., 50 patients).

PFQ algorithms have been simulated (simulation number 4) further under scenario 3 and sub scenarios i.e. variant emergency rates to test the effectiveness of the FQ-TD and PFQ algorithm under various emergency rate sub-scenarios. These scenarios have been simulated to study how PFQ performs for different rate of emergency cases, with respect to average delay and jitter. The simulation was performed from low to high rate of emergency cases volume

The sub scenarios were performed from low to high rate of emergency cases volume. In this scenario twelve (12) patients send three (36) readings or signals one from each of the 3 (03) sensors; heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU). Each sensor, monitors one specific medical information, and transmits its signal using ZigBee to the LDPU which acts as a hub that collects all medical information form sensors, and store them temporarily in its buffer before forwarding them to the healthcare centre.

The variable in this scenario is in the emergency and non-emergency signals percentage or volume being transmitted. In the first sub-scenario we started with seven (7) emergency signals and (29) non-emergency signals, in the second sub-scenario we increased it to fourteen (14) emergency signals and twenty two (22) non-emergency signals, in the third sub-scenario we increased it to twenty two (22) emergency signals and fourteen (14) non-emergency signals, and in the fourth sub-scenario we increased it to twenty nine (29) emergency signals and seven (7) non-emergency signals. In other words, we increased the volume of the emergency signals in each sub-scenario i.e. in the first sub-scenario the emergency signals forms 20% of the total number of the thirty-six (36) signals being transmitted. In the second sub-scenario the emergency signals forms 40% of the total number of the thirty-six (36) signals being transmitted. In the third sub-scenario the emergency signals forms 60% of the total number of the thirty-six (36) signals being transmitted and in the last and fourth sub-scenario the emergency signals forms 80% of the total number of the thirty-six (36) signals being.

Notice that we have started with low percentage or volume of emergency signals and high percentage or volume of non-emergency signals and reverse the same percentage or volume of the emergency and non-emergency signals in the last and fourth sub scenario. Also, a total of thirty-six (36) signals being transmitted from the twelve (12) patients are maintained across the four sub scenarios.

Simulation number 4 setup was maintained during the run of scenario 1 and 3 (i.e. number of patients and variant emergency rates respectively) were twelve (12) patients send three (36) readings or signals one from each of the 3 (03) sensors; heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU). Each sensor, monitors one specific medical information, and transmits its signal using ZigBee to the LDPU which acts as a hub that collects all medical information form sensors, and store them temporarily in its buffer before forwarding them to the healthcare centre.

In Summary, several simulations have addressed the number of patients, traffic volume and variant emergency signals under different scenarios and sub scenarios to visualize the algorithm effectiveness in tackling the delay and jitter attributes under emergency and non-emergency situations.

3.4.2. Simulation Setup

This part explains how we prepare the simulation testbed. a Dell laptop Inspiron- 13 equipped with windows 10 home (64-bit), Core i5 Processor, 8GB RAM, has been used to install the Python language. Python version was 3.6.2. Visual Studio Code [5] has been used for coding.

To install Python 3.6.2, we downloaded the file from their official website [3] and install it easily using the wizard. Similar procedures were done to install Visual Studio Code. The standard way to run a Python file is to open the command window and navigate to the folder that contains the file python.exe. Then, you can run your file from this file by typing its name (e.g., hello.py).

To run or execute Python files directly from Visual Studio Code, we set a path to Python interpreter, python.exe, in Windows's PATH variable. To do so, click the right button on the computer Icon and click properties. From the opening panel, click on the advanced system settings. Then, click on the button Environment Variables. In the new window, scroll until you see the system variable, Path; click on it and Press Edit. Finally type or copy the path of the python interpreter (i.e., pyhton.exe) to this field. Thus, we can run the simulation through the Visual Studio Code directly. Figure 3.8 shows a screenshot of executing a simple python code using Visual Studio Code.

3.4.3. Simulation Model

To build our simulation, we took the benefit of Grotto's model named SimComponents.py [6], which has a set of components to create a network simulation. SimComponents.py is basically developed based on SimPy library.

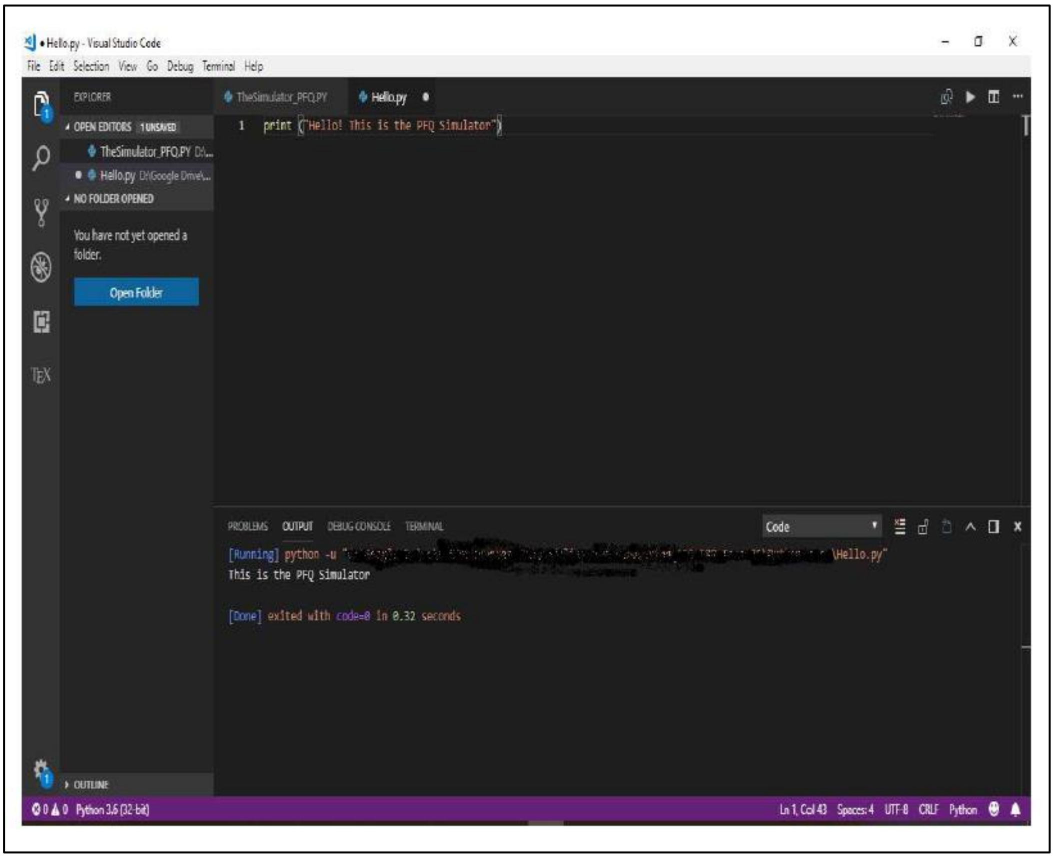


Figure 3.8. A Successful Execution of a Simple Python Code using VB Code

Figure 3.9 shows the basic components of our simulation. The basic components of the simulation are Packet Generator, Local Data Processing Unit (LDPU), and Packet Sink.

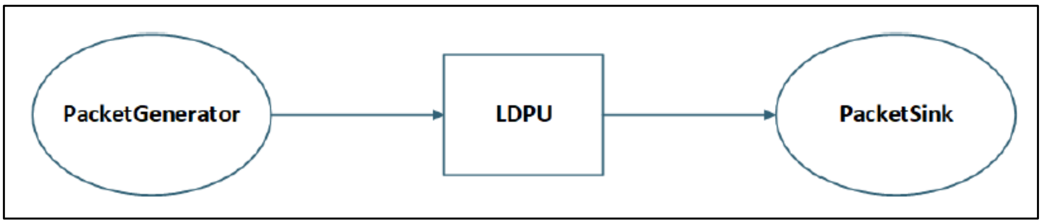


Figure 3.9. Simulation Components

A. Packet Generator

This component simulates the generation of packets with a specified an inter-arrival time and specified packet size. Generally, this component is responsible for creating packets. The packet object includes fields like generation time, size, flow id, packet id, emergency, PP, source, and destination. For instance, the emergency field has a value of 0 or 1. 0 denotes for non-emergency packet whereas the 1 denotes the emergency packet.

*PacketGenerator*class has parameters named env, sdist, initiaLdelay, finish. env represents the simulation environment based on SimPy library. adist and sdist are functions that return the successive inter arrival times of the packets and the successive packet sizes of the packet respectively. initiaLdelay is a number that starts generating packets after an initial delay while finish is a number that stops generating packets at the finish time. The default value for the initial delay is 0 and infinity for the finish time.

Figure 3.10 shows how packets are generated using packet generators. We created 3 packet generators named pg-ECG, pg-Gluco.se and pg-Pressure, as clarified on lines 2, 3, and 4 respectively. These generators were created based on PacketGenerator class. On line 6, 7, and 8, we wire the three packet generators to the LDPU.

```
# Create  packet  generators
pg_ECG = PacketGenerator(env, "ECG", constECGArrival, constECGSize,
    finish = 10000, flow_id=0,PPfact=ECG_PPfact, MaxPkts=1000))
pg_Glucose = PacketGenerator(env, "Glucose", constGlucoseArrival,
  constGlucoseSize, finish = 10000, flow_id=1, PPfact=
  Glucose_PPfact,MaxPkts=1000))
pg_Pressure = PacketGenerator(env, "Pressure", constPressureArrival
  , constPressureSize, finish = 10000, flow_id=2, PPfact=
  Pressure_PPfact,MaxPkts=1000))
# Wire packet generators and LDPU together
pg_ECG.out =       LDPU
pg_Glucose.out =  LDPU
pg_Pressure.out = LDPU
```

Figure 3.10. Connecting the Packet Generator to the LDPU

From the PacketGenerator class, packet Generator component is connected to the LDPU using out variable and put() function, as presented in Figure 3.11. The last line (i.e., line 11) in the code shows how the run() function of the Packet Generator sends a packet P to the LDPU throughout variable and put() function.

B. Local Data Processing Unit (LDPU)

LDPU is the core component of our model. We model our algorithms such FIFO, WFQ, and PFQ at this component. At LDPU, we can determine the rate of the output (i.e., how many packets per second) and the queue size. Further, LDPU component records the information of the received and dropped packets.

```
def run(self):
"""The generator function used in simulations.
"""
yield self.env.timeout(self.initial_delay)
while self.env.now < self.finish:
# wait for next transmission
yield self.env.timeout(self.adist())
self.packets_sent += 1
self.NumOfGeneratedPkts +=1
p = Packet(self.env.now, self.sdist(), self.packets_sent, src=self.id
    , flow_id=self.flow_id, PPfact=self.PPFactor)
self.out.put(p)
```

Figure 3.11. Packet Generator Sending a Packet

In code Figure 3.12, we model the three algorithms, FIFO, WFQ, and PFQ at *LDPU* component as classes in our simulation. Each class has parameters named *env, rate. env* represents the simulation environment based on SimPy library. *rate* represents the line rate of the LDPU. *env* represents the simulation environment based on SimPy library. For WFQ and PFQ, we add a parameter named *phis* which is a list that assign each packet to the flow id. In PFQ, all flows have the same weight, whereas different weights are assigned for each flow (i.e., application or sensor) in WFQ. We assigned these different weights based on Table 3.1, as represented by parameters *PhLECG, PhLGlucose, and PhLPressure.* The queue model system that has been used in our simulation is M/M/1 queue, where there is no limit on queue sizes.

```
# Create Local Data Processing Unit (LDPU)
LDPU = PFQ(env, source_rate, [0.33*phi_base,0.33*phi_base,0.33*
  phi_base])
#LDPU = WFQ(env, source_rate, [Phi_ECG*phi_base,Phi_Glucose*
  phi_base, Phi_Pressure*phi_base])
#LDPU = FIFO(env, rate=source_rate)
```

Figure 3.12. Modelling FIFO, WFQ & PFQ Algorithms at LDPU Component

In code Figure 3.13, we wire the LDPU and a multiplexer together. Demultiplexing process is used to split packets based on their flow id parameter. In our simulation we have three application/sensors and they send packets through the same LDPU (multiplexing). So, this step eases the process of study the packet delay and jitter for each sensor or application. On line 2, we created an object named demux from the class FlowDemux. On line 3, we connect the LDPU to the demultiplexer demux using out parameter. The default value of this parameter is none, as shown in Figure 3.14. The put() method is used to split packets based on their flow id. Jnit_ is a function that is executed automatically when creating an object of a class.

```
1 # wire LDPU and demux together
2 demux = FlowDemux()
3 LDPU.out = demux
```

Figure 3.13. Connecting LDPU and Demultiplexr Together

```
class FlowDemux(object):
  def __init__(self, outs=None, default=None):
    self.outs = outs
    self.default = default
    self.packets_rec = 0

def put(self, pkt):
  self.packets_rec += 1
  flow_id = pkt.flow_id
  if flow_id < len(self.outs):
    self.outs[flow_id].put(pkt)
  else:
    if self.default:
      self.default.put(pkt)
```

Figure 3.14. Demultiplexr Class

C. PacketSink

Packet Sink is the component that records the arrival time information of the packets. This information is collected in a list so as to look at the delay statistics. The PacketSink class has parameters named env, debug, rec-arrivals, absolute-arrivals, rec-waits, selector, as shown in Figure 3.15. These parameters are defined under Jnit_ function that is executed automatically when creating an object of a class. env represents the simulation environment based on SimPy library. debug is a boolean which prints the content of each packet upon its receipt at the sink. It should be true to enable this feature. rec-arrivals has a boolean value and it records the arrivals if it is true. Similarly, absolute-arrivals is a boolean and record the absolute arrival times if it true. Otherwise, it records the consecutive arrivals. rec-arrivals has a boolean value and it records the waiting times experienced by each packet, if it is true. selector is a function that takes a packet and returns a boolean used for selective statistics. The default value is none.

```
class PacketSink(object):
  def __init__(self, env, rec_arrivals=False, absolute_arrivals=False
    , rec_waits=True, debug=False, selector=None,emg_waits=True):
    self.store = simpy.Store(env)
    self.env = env
    self.rec_waits = rec_waits
    self.emg_waits=emg_waits
    self.rec_arrivals = rec_arrivals
    self.absolute_arrivals = absolute_arrivals
    self.waits = []
    self.emgWaits=[]
    self.NonemgWaits=[]
    self.arrivals = []
    self.debug = debug
    self.packets_rec = 0
    self.bytes_rec = 0
    self.selector = selector
    self.last_arrival = 0.0
    self.packets_emg=0.0
    self.jitter = 0.0
    self.emg_jitter=0.0
    self.Nonemg_jitter=0.0
    self.emg_last_arrival = 0.0
    self.PPwaits=[0.0 for i in range(6)]
    self.PPCount=[0.0 for i in range(6)]
    self.packet_drop_sink = 0
    self.countt = 0
```

Figure 3.15. PacketSink Class

PacketSink class has many functions that do a certain task. For instance, in code Figure 3.16 we define functions under to calculate the Jitter. On line code 1, we define a function named emgJitter to calculate the jitter of the emergent packets of a certain sensor. On Code line 9, NonemgJitter function has been defined. This function calculates the jitter of the non-emergency packets of a certain sensor.

```
1  def emgJitter(self):     # a function to calculate jitter at sink
2    self.emg_jitter=0.0
3    if  len(self.emgWaits) > 1:
4    for i in range(len(self.emgWaits)-1):
5    self.emg_jitter += abs(self.emgWaits[i] - self.emgWaits[i+1])
6    self.emg_jitter/=(len(self.emgWaits)-1)
7    return self.emg_jitter
8
9  def NonemgJitter(self):     # a function to calculate jitter at sink
10   self.Nonemg_jitter=0.0
11   if  len(self.NonemgWaits) > 1:
12   for i in range(len(self.NonemgWaits)-1):
13   self.Nonemg_jitter += abs(self.NonemgWaits[i] - self.NonemgWaits[i
      +1])
14   self.Nonemg_jitter/=(len(self.NonemgWaits)-1)
15   return self.Nonemg_jitter
16
17
```

Figure 3.16. Jitter function at PacketSink Class

In Figure 3.17 we create 3 packet sinks named *ps-ECG*, *ps-Gluco.se*, and *ps-Pressure*. In Figure 3.18 we wire the 3 packet sinks with the demultiplexer *demux* that is connected with the LDPU.

```
# Create  packet sinks
  ps_ECG=PacketSink(env, rec_arrivals=True, absolute_arrivals=False)
  ps_Glucose=PacketSink(env, rec_arrivals=True, absolute_arrivals=
    False)
  ps_Pressure=PacketSink(env, rec_arrivals=True, absolute_arrivals=
    False)

```

Figure 3.17. Jitter function at PacketSink Class (EGC, Glucose & Pressure)

```
# wire demux and sinks together
  demux.outs = [ps_ECG, ps_Glucose, ps_Pressure]

```

Figure 3.18. Jitter function at PacketSink Class (demux)

3.4.4. Simulation Parameters

Simulation parameters that have been used in our simulation are summarized previously in Table 3.3. We have used a constant arrival rate (i.e., 1) with exponential interarrival time. The packet size was 133 bytes, which is ZigBee packet size. The LDPU data rate was 40 kbps (ZigBee bit rate) [7]. Normally, emergency packets are less in size than non-emergency packets, so that we put more percentage for the latter. It can be varied. We relied on the number of patients to reach the peak time. We started with 1, then 4, then 8, until 12. Based on our simulation setup, we found that all algorithms suffer from Congestion at 12 patients, except PFQ.

3.4.5. Simulation Scenarios

As it was clearly stated in previous chapters, the main objective of this research is to reduce the delay and improve jitter during emergency data transmission in M2M eHealth ecosystem. Accordingly, two new priority queuing models namely the ‘‘Priority based fair Queuing” (PFQ) and ‘‘Priority-based Fair Queuing with Tolerated Delay” (PFQ-TD) have been developed as part of this research to address the emergency and non-emergency packets transmission.

While the PFQ model addresses the emergency cases, the PFQ-TD model considers the non-emergency cases by not giving the emergency cases high priority all the time. PFQ-TD model has considered the finishing time for non-emergency cases and gave it fair consideration over emergency cases. Thus, balancing the outcome and gave fair and balance delay and jitter distribution for emergency and non-emergency cases. In the new proposed models, each packet in the eHealth system has a priority and it is based on both, the application and sensor type criticality such as heart rate, blood pressure and glucose which monitors the sugar level of the blood and the patient’s health status (age, health history, gender, pregnancy etc.).

Python Language is used to simulate the PFQ and PFQ-TD queuing algorithm in real time focusing on Remote Patient Monitoring (RPM) and the Assisted Living (AL) systems. This research aimed to design a test platform to collect data and attributes in various scenarios based on the above technologies to measure delay and jitter.

The simulation is based on several scenarios of group of elderly patients living in a care center. Each patient has three sensors attached or embedded in his or her body such as Heart Rate (ECG), Blood Pressure and Glucose sensors which send vital data regularly to the health care center. Each sensor, monitors one specific medical information, and transmits its signal using ZigBee, to the Local Data Processing Unit

(LDPU). As a hub, the LDPU collects all medical information form sensors, and store them temporarily in its buffer.

The data is transferred from the sensor (source) to the Local Data Processing Unit (LDPU) where the proposed new PFQ and PFQ-TD algorithms are executed. The LDPU send the data according to its criticality and classification to the health care center (Sink). LDPU acts as a network regulator which is responsible for determining the allocation of transmission path, capacity and bandwidth among sensors during each time frame. It implements the proposed workflow. The LDPU decides its strategy based on its utility function, which is determined by the priority of the medical data and the transmission cost.

The simulation is conducted in several scenarios and different transmission environment to improve our contribution of decreasing the jitter and delay to the minimal we can achieve. We increased the number of patients gradually to test the effectiveness of the models for the three ECG, Blood Pressure and Glucose sensors simulating high peak time. The scenarios were tested and simulated based on the newly proposed priority algorithms and was run also on current priority queueing models such as FIFO and WFQ to compare the results of the Delay and Jitter parameters for emergency and non-emergency cases.

Moreover, we separated the traffic for emergency and non-emergency cases to see how each algorithm shall process them in peak and off-peak time. The results of jitter and delay will be presented according to the Priority Parameter (PP) values.

In the simulation work, we have conducted two main simulation-based scenarios, to evaluate the effectiveness of models in ZigBee for healthcare applications. We repeated each scenario under each simulation setup for 10 times.

The first main simulation setup as described in Table 3.3 was run under scenario 1 and 2 only i.e. number or patients and high traffic volume respectively, bearing in mind that the emergency signals constitute 30% of the total number of signals being sent while the remaining 70% is non-emergency signals.

In the second main simulation setup which was run under scenario 3 only i.e. variant emergency rate, all parameters are kept unchanged as per Table 3.3 except for the emergency rate which varies and increases according to scenario 3 sub scenarios i.e. emergency signals increase form 20%, to 40% to 60% to 80% of the total number of signals being sent.

3.4.6. Scenario 1: Number of Patient Scenarios

The purpose of this scenario is to count for the number of patients using eHealth ecosystem. Normally, different eHealth centers have varied number of patients. Therefore, a varied number of patients have been used in this scenario to simulate the real eHealth centers. In our simulation setup, emergency rate was set to 30% while non-emergency rate was set to 70%. This scenario is used to evaluate the performance of PFQ with FIFO and WFQ in terms of average delay and jitter. It also compares PFQ with PFQ-TD. Four (04) sub-scenarios were established to simulate the real time transmission from the sensors to healthcare center. Each sensor, monitors one specific medical information, and transmits its signal using ZigBee, to the LDPU which acts as a hub that collects all medical information form sensors, and store them temporarily in its buffer before forwarding them to the healthcare center.

1 SC1-Sub1: As shown in Table 3.4 the first sub-scenario consists of one (01) patient shall send three (03) readings one from each of the 3 (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).

2 SC1-Sub2: As shown in Table 3.4 the second sub-scenario consists of four (04) patients shall send twelve (12) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).

3 SC1-Sub3: As shown in Table 3.4 the third sub-scenario eight (08) patients shall send twenty-four (24) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).

4 SC1-Sub4: As shown in Table 3.4 the fourth and last sub-scenario twelve (12) patients shall send thirty-six (36) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).

Table 3.4 Scenario 1 Simulation Parameters

Sensor Type	**Scenario #**	**Sub-scenario #**	**Number of Patients**	**Number of Readings**
Heart Rate	1	1	1	1
Blood Pressure	1	1	1	1
Glucose	1	1	1	1
Total Number of Readings				**3**
Heart Rate	1	2	4	4
Blood Pressure	1	2	4	4
Glucose	1	2	4	4
Total Number of Readings				**12**
Heart Rate	1	3	8	8
Blood Pressure	1	3	8	8
Glucose	1	3	8	8
Total Number of Readings				**24**
Heart Rate	1	4	12	12
Blood Pressure	1	4	12	12
Glucose	1	4	12	12
Total Number of Readings				**36**

3.4.7. Scenario 2: High Traffic Volume Scenarios

This high traffic volume scenario has been developed based on the first scenario, simulation setup and results analysis. It has been found that all models suffered from Congestion at 12 patients. So, this scenario has been used to test PFQ model performance in emergency cases under high-volume traffic gradually increased from one (1) patient with three (3) readings to fifty (50) patients and hundred and fifty (150) readings. In our simulation setup, emergency rate was set to 30% while non-emergency rate was set to 70%.

This scenario has been performed to study the performance of PFQ model under a high traffic volume. Eight (08) simulation sub-scenarios were performed from low to high traffic volume. Similar to previous runs each sensor, monitors one specific medical information, and transmits its signal using ZigBee, to the LDPU which acts as a hub that collects all medical information form sensors, and store them temporarily in its buffer before forwarding them to the healthcare center.

1. SC2-Sub1: As shown in Table 3. 5 the first sub-scenario consists of one (01) patient shall send three (03) readings one from each of the 3 (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).
2. SC2-Sub2: As shown in Table 3.5 the second sub-scenario consists of four (04) patients shall send twelve (12) readings one from each of the three (03) sensors; Heart rate, Blood Pressure 30 and Glucose simultaneously to the Local Data Processing Unit (LDPU).

3. SC2-Sub3: As shown in Table 3.5 the third sub-scenario consists of eight (08) patients shall send twenty-four (24) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).
4. SC2-Sub4: As shown in Table 3.5 the fourth sub-scenario consists of twelve (12) patients shall send thirty-six (36) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).
5. SC2-Sub5: As shown in Table 3.5 the fifth sub-scenario consists of twenty (20) patients shall send sixty (60) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).
6. SC2-Sub6: As shown in Table 3.5 the sixth sub-scenario consists of thirty (30) patients shall send ninety (90) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).
7. SC2-Sub7: As shown in Table 3.5 the seventh sub-scenario consists of forty (40) patients shall send one hundred twenty (120) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).

8. SC2-Sub8: As shown in Table 3.5 the eighth sub-scenario consists of fifty (50) patients shall send one hundred fifty (150) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU).

Table 3.5. Scenario 2 Simulation Parameters

Sensor Type	Scenario #	Sub-scenario #	Number of Patients	Number of Readings
Heart Rate	2	1	1	1
Blood Pressure	2	1	1	1
Glucose	2	1	1	1
Total Number of Readings				**3**
Heart Rate	2	2	4	4
Blood Pressure	2	2	4	4
Glucose	2	2	4	4
Total Number of Readings				**12**
Heart Rate	2	3	8	8
Blood Pressure	2	3	8	8
Glucose	2	3	8	8
Total Number of Readings				**24**
Heart Rate	2	4	12	12
Blood Pressure	2	4	12	12
Glucose	2	4	12	12
Total Number of Readings				**36**
Heart Rate	2	5	20	20
Blood Pressure	2	5	20	20
Glucose	2	5	20	20
Total Number of Readings				**60**
Heart Rate	2	6	30	30
Blood Pressure	2	6	30	30
Glucose	2	6	30	30
Total Number of Readings				**90**
Heart Rate	2	7	40	40
Blood Pressure	2	7	40	40
Glucose	2	7	40	40
Total Number of Readings				**120**
Heart Rate	2	8	50	50
Blood Pressure	2	8	50	50
Glucose	2	8	50	50
Total Number of Readings				**150**

3.4.8. Scenario 3: Variant Emergency Rate Scenarios

This scenario has been developed to simulate the emergent and non-emergent traffic rate under the second main simulation-based scenario setup where variant emergency rate is increased gradually. In our first main simulation setup, emergency rate was set to 30% while non-emergency rate was set to 70%. This is because emergency packets usually are less than non-emergency packets. However, there are situations where emergency packets are higher than the non-emergency packets. Thus, this scenario has been used to test the performance of PFQ with variant emergency rates.

This scenario has been simulated to study how PFQ performs for different rate of emergency cases, with respect to average delay and jitter. One simulation was performed from low to high rate of emergency cases volume representing the four (04) sub scenarios. Each sensor, monitors one specific medical information, and transmits its signal using ZigBee, to the LDPU which acts as a hub that collects all medical information form sensors, and store them temporarily in its buffer before forwarding them to the healthcare center.

1. SC3-Sub1: As shown in Table 3.6 the sub-scenario consists of twelve (12) patients shall send three (36) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU). These readings varied from (7) emergency (29) non-emergency i.e. 20% emergency signals and 80% non-emergency signals.
2. SC3-Sub2: As shown in Table 3.6 the sub-scenario consists of twelve (12) patients shall send three (36) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU). These readings varied from (14) emergency (22) non-emergency i.e. 40% emergency signals and 60% non-emergency signals.

3. SC3-Sub3: As shown in Table 3.6 the sub-scenario consists of twelve (12) patients shall send three (36) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU). These readings varied from (22) emergency (14) on-emergency i.e. 60% emergency signals and 40% non-emergency signals.
4. SC3-Sub4: As shown in Table 3.6 the sub-scenario consists of twelve (12) patients shall send three (36) readings one from each of the three (03) sensors; Heart rate, Blood Pressure and Glucose simultaneously to the Local Data Processing Unit (LDPU). These readings varied from (29) emergency (7) non-emergency i.e. 80% emergency signals and 20% non-emergency signals.

Table 3.6. Scenario 3 Simulation Parameters

Scenario #	Sub-scenario #	Number of Patients	Number of Readings	Emergency Readings	Non-Emergency Readings
3	1	12	36	7	29
3	2	12	36	14	22
3	3	12	36	22	14
3	4	12	36	29	7

Chapter 4: Results and Discussion

4.1 Introduction

This chapter presents the delay and jitter results obtained from simulating the proposed models PFQ and PFQ-TD under different scenarios and setups to visualize their effectiveness in improving the delay and jitter parameters in emergency and non-emergency cases. It also compares those results with the one obtained from the WFQ, and FIFO models to stand over the improvements and drawbacks. The results of the three scenarios (number of patient's scenario, high traffic volume scenario and variant emergency rate scenario) and their sub sub-scenarios to test the new model performance under different conditions are presented. This chapter also includes a comparison between PFQ and PFQ-TD in terms of delay and jitter metrics, for both emergency and non-emergency cases.

4.2 Test Results Standard Deviation

As it will be shown in the coming sections, test results have shown significant improvement in Delay and Jitter time for Emergency cases using the PFQ model in scenario 1 satisfying the main objective of the thesis. The mean (average) time was reported as the final result; whereas the calculated Standard Deviation has also shown minimum variations from the mean value for the PFQ model which implies that the obtained results are robust and the performance of the PFQ model is steady and within the acceptable range of change. Since the Standard Deviation value is small and constant, thus it became evident that it shall not add any extra information to the analysis of the graphs or clutters the image of the test results. Full details of the calculated Standard Deviations are shown in section 4.9.

4.3 Analyzing Delay and Jitter Results of FIFO, WFQ, and PFQ Algorithms under Scenario 1 (Number of Patients)

In scenario number 1 'number of patients', the simulation is performed from low to high number of patients where at sub-scenario 1 (SC1-Sub1) one (01) patient sends three (03) readings one from each of the three (03) sensors; heart rate (ECG), Blood Pressure and Glucose simultaneously. Following this, at sub-scenario 2 (SC1-Sub2) four (04) patients send twelve (12) readings, sub-scenario 3 (SC1-Sub3) eight (08) patients send twenty-four (24) readings, and in sub-scenario 4 (SC1-Sub4) twelve (12) patients send thirty-six (36) readings. Table 4.1 summarizes the high-level structure of scenario 1 and sub scenarios.

Table 4.1. Scenario 1 High Level Structure

Scenario 1 ''Number of Patients"	Number of Patient	Number of Readings (Emg and Non-Emg cases)
SC1-Sub1	1	3
SC1-Sub2	4	12
SC1-Sub3	8	24
SC1-Sub4	12	36

4.3.1 Emergency Case

Figure 4.1 compares the performance of FIFO, WFQ, and PFQ models with respect to the average delay of emergency cases for ECG, Glucose, and Pressure sensors. The average delay of the three sensors with 1, 4, and 8 patients is almost the same for, FIFO, WFQ, and PFQ. This is because at this number of patients there is no starvation at LDPU. A significant increase is noticed with 12 patients due to the resource starvation.

Table 4.2 refers to scenario 1, sub scenarios, models and emergency delay and jitter results referenced figures.

Table 4.2. Scenario 1 Emg Delay and Jitter Results Referenced Figures

Scenario	Patient	Readings	Model	Emg-Delay Results	Emg-Jitter Results
SC1-Sub1	1	3	FIFO WFQ PFQ	Fig 4.1	Fig 4.2
SC1-Sub2	4	12		Fig 4.1	Fig 4.2
SC1-Sub3	8	24		Fig 4.1	Fig 4.2
SC1-Sub4	12	36		Fig 4.1	Fig 4.2

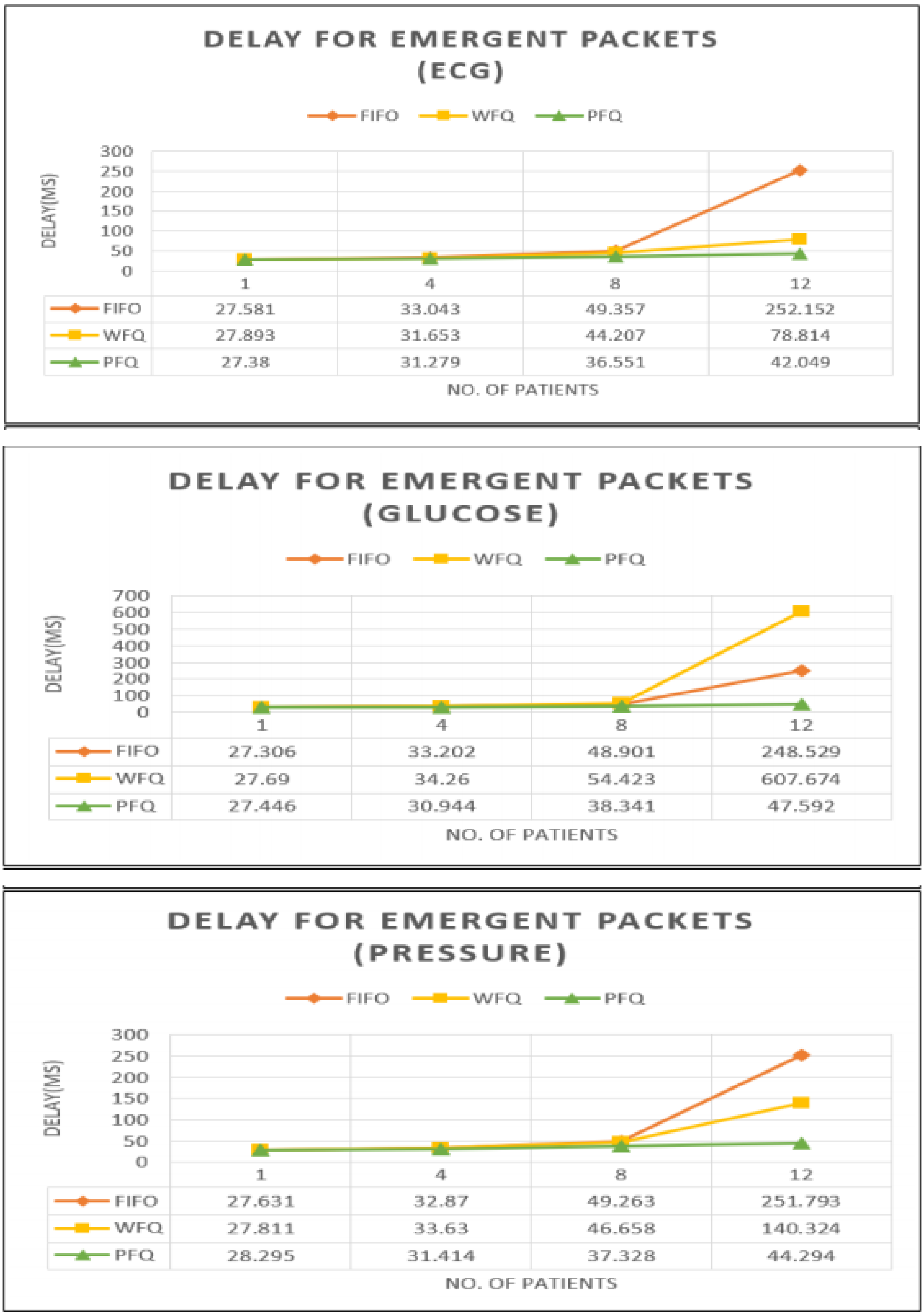
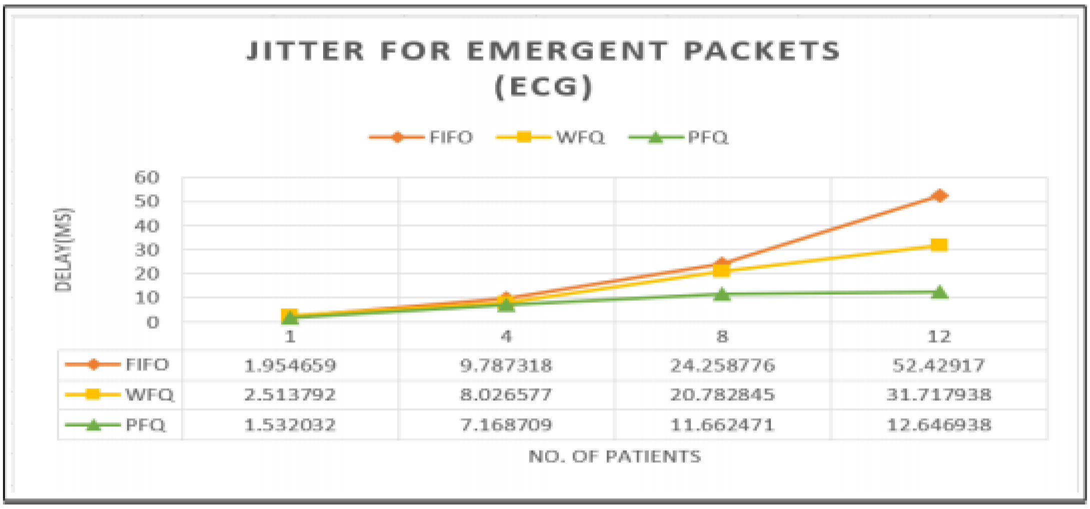
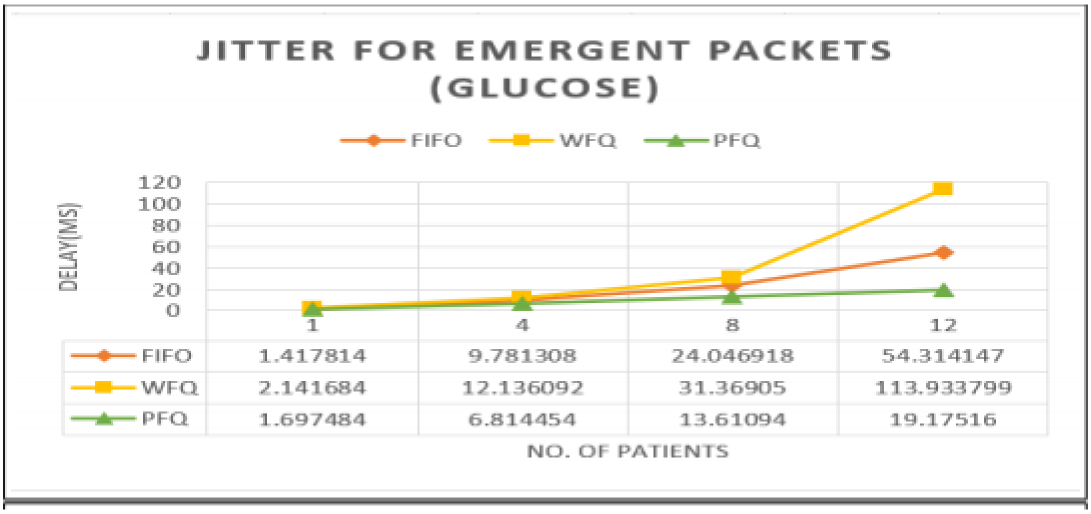


Figure 4.1. Av. Delay of Emergency Cases (ECG, Glucose & Pressure Sensors)





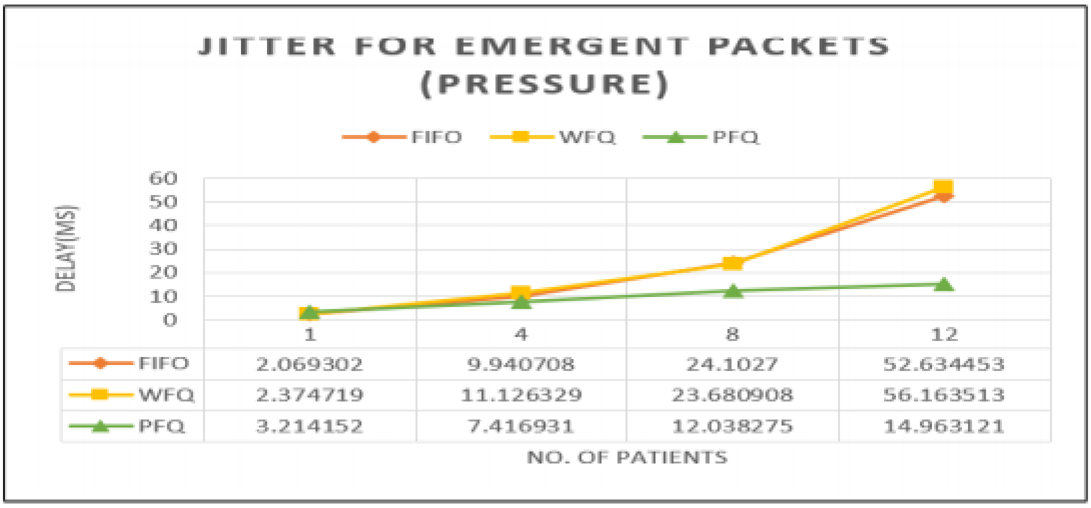


Figure 4.2. Av. Jitter of Emergency Cases (ECG, Glucose & Pressure Sensors)

From Figure 4.1 and 4.2, It is clear that PFQ outperformed FIFO and WFQ models in the three sensors for Delay and Jitter under emergency cases. PFQ succeeded to keep the delay of emergency packets quite low because of giving priority to emergency cases over non-emergency cases all the time. It is worth mentioning that in our simulation setup, each sensor type sends both emergency and nonemergency packets. The emergency rate was set to 30% while non-emergency rate was set to 70%.

On the other side WFQ has assigned a weight for each flow (i.e., sensors or application) to prioritize packet transmission based in Table 3.1. Therefore, emergency cases of ECG sensor have a less average delay because its weight is greater than the weight of Glucose and Pressure. FIFO always maintains the average delay is nearly the same for all sensors due to its scheduling procedure.

Similarly, PFQ succeeded to keep the average jitter quite low for emergency cases, as shown in Figure 4.2. For WFQ, the average jitter of Glucose packets is higher than pressure and ECG due to assigning a lower weight to this sensor. Similar to average delay, FIFO always maintains the average jitter is nearly the same for all sensors due to its scheduling procedure.

4.3.2 Non-Emergency Case

Figure 4.3 compares the performance of FIFO, WFQ, and PFQ models with respect to the average delay of non-emergency cases for ECG, Glucose, and Pressure sensors. PFQ model recorded similar values as of WFQ and FIFO models up to medium traffic volume at 8 patients, however it increased dramatically at 12 patients (i.e., high traffic volume). This significant increase of the average delay for the three sensors in case of PFQ is attributed to giving priority to emergency cases over non-emergency cases.

Hence, PFQ recorded the highest average delay of 394 msec for ECG, 444 msec for Pressure, and 618 msec for Glucose measure readings. It is known that PFQ uses data type criticality of the sensor or application to priorities packet transmission n. Based on Table 3.1, Glucose readings have a lower value than ECG and Pressure readings. Therefore, the average delay of non-emergency cases of Glucose sensor readings is higher than the average delay of non-emergency cases in of other sensors. For FIFO and WFQ, the average delay of non-emergency cases is similar to the average delay of emergency cases. This is because these models do not differentiate between emergency and non-emergency cases.

Figure 4.4 compares the performance of FIFO, WFQ, and PFQ algorithms with respect to the average jitter of non-emergency cases for ECG, Glucose, and Pressure sensors. The performance of all models in terms of average jitter is comparable to their performance in terms of average delay. For instance, a dramatic increase is noticed in the average delay of ECG in case of PFQ when the number of patients is increasing after a certain threshold, for example if there are more than eight patients. Similarly, the average jitter is dramatically increased of ECG. For WFQ and FIFO, similarly, there is no difference in average jitter between emergency cases and non-emergency cases. Table 4.3 refers to scenario 1, sub scenarios, models and non-emergency delay and jitter results referenced figures.

Table 4.3. Scenario 1 Non-Emg Delay & Jitter Results Referenced Figures

Scenario	Patient	Readings	Model	Non Emg-Delay results	Non Emg-Jitter results
SC1-Sub1	1	3	FIFO WFQ PFQ	Fig 4.3	Fig 4.4
SC1-Sub2	4	12		Fig 4.3	Fig 4.4
SC1-Sub3	8	24		Fig 4.3	Fig 4.4
SC1-Sub4	12	36		Fig 4.3	Fig 4.4

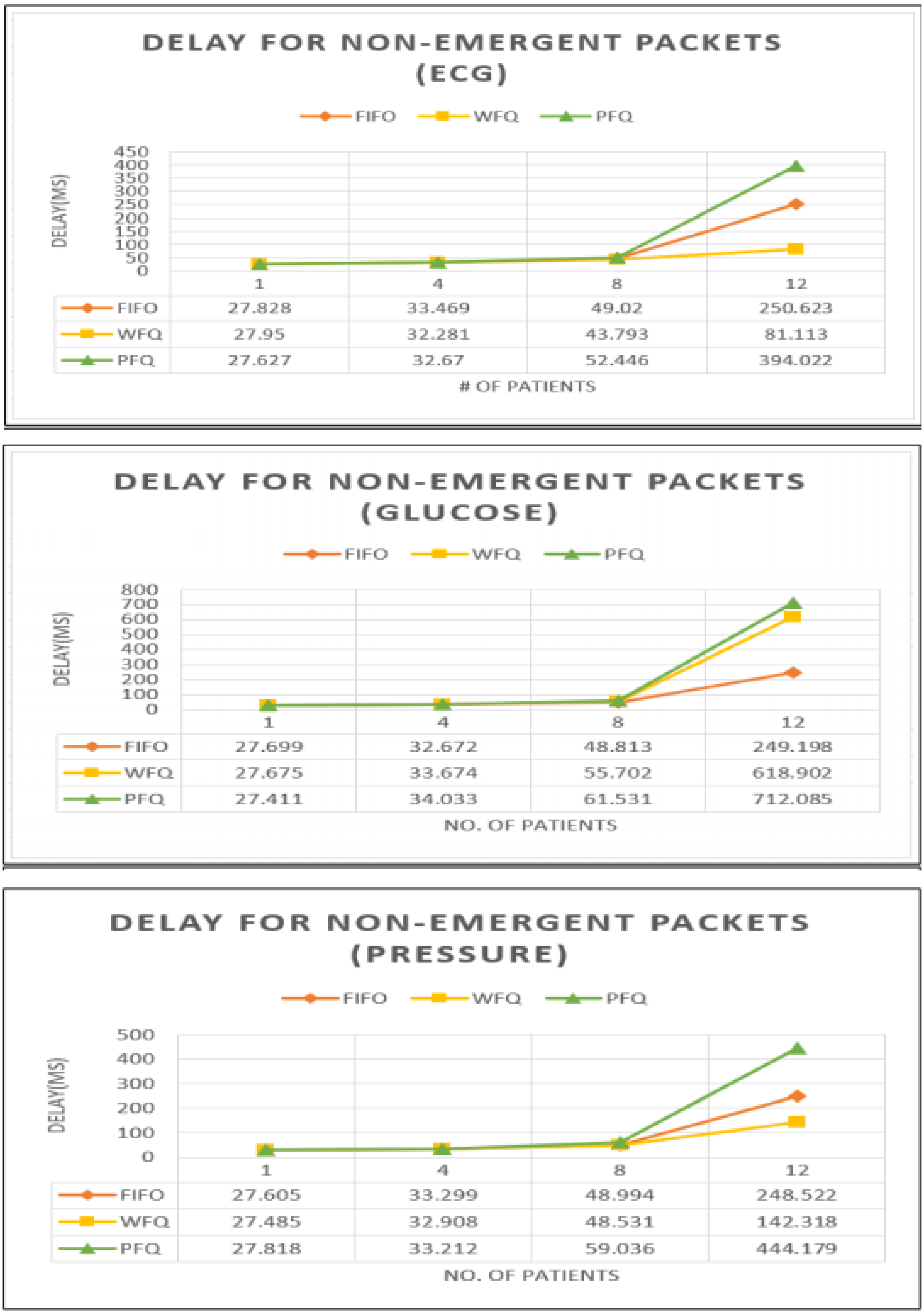
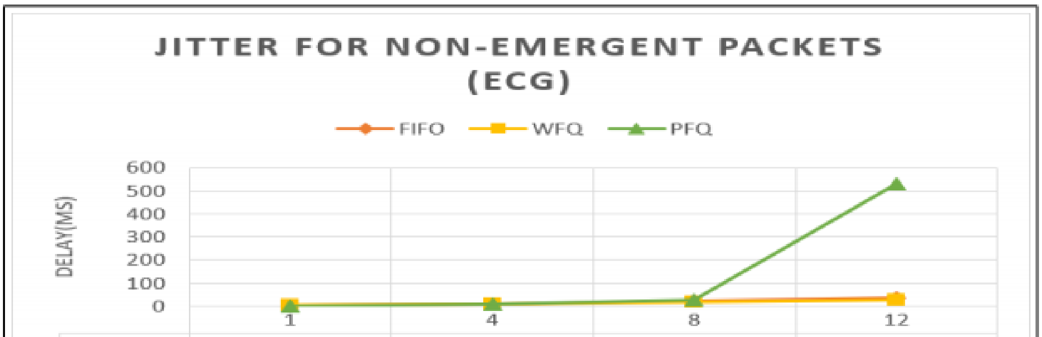
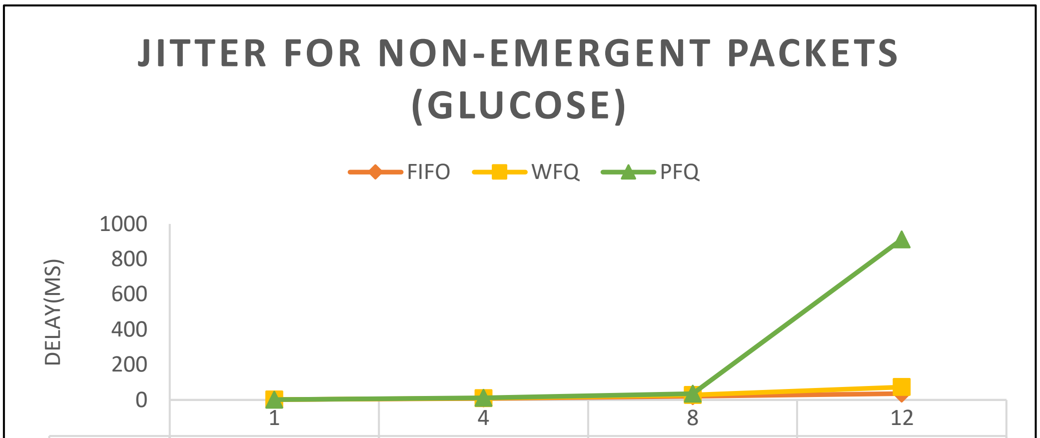


Figure 4.3. Av. Delay of Non-Emerg Cases (ECG, Glucose & Pressure Sensors)



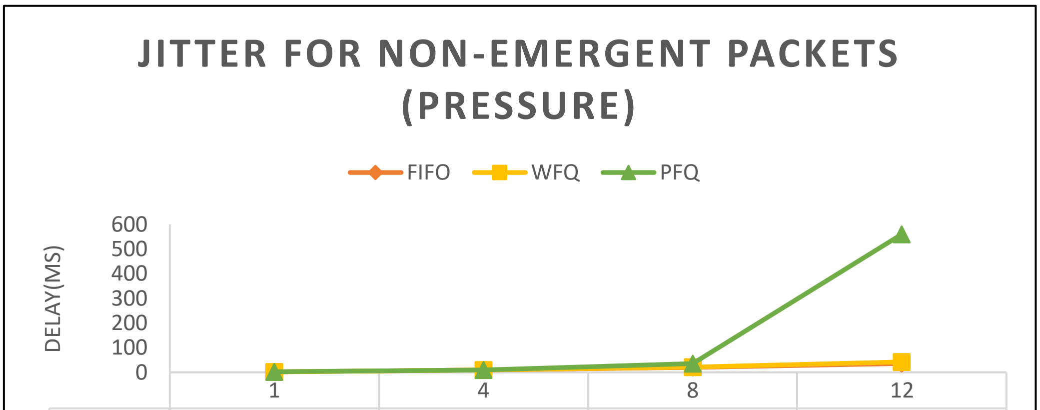
	1	4	8	12
FIFO	2.310413	9.863689	20.671448	36.590692
WFQ	2.445802	8.381492	18.152422	25.517299
PFQ	1.854844	9.386815	27.995709	529.07363

OF PATIENTS



	1	4	8	12
FIFO	2.085163	8.795875	20.576839	35.947554
WFQ	2.022108	10.531762	27.678363	72.843729
PFQ	1.62106	11.522596	35.104327	912.614872

NUMBER OF PATIENTS



	1	4	8	12
FIFO	1.950282	9.736172	20.076547	35.745676
WFQ	1.717354	9.646029	21.759938	42.102552
PFQ	2.351818	10.02881	35.0055	560.058431

NUMBER OF PATIENTS

Figure 4.4. Av. Jitter of Non-Emerg Cases (ECG, Glucose & Pressure Sensors)

4.4 Comparing Delay and Jitter Results of PFQ and PFQ-TD Models under Scenario 1 (Number of Patients)

Since PFQ model out performed FIFO and WFQ models under emergency cases as proven under scenario 1 'Number or Patients' yet it didn't improve the non-emergency cases, it deemed necessary to compare the PFQ to PFQ-TD under the same conditions of scenario 1 and its sub-scenarios. As a matter of fact, PFQ is expected to increase the delay and jitter of the non-emergency packets due to the high priority assignment for emergency packets all the time, as shown in Figures 4.3 and 4.4. Consequently, the non-emergency packets become useless since they exceed the tolerated delay (TD). PFQ-TD has been proposed to give a balance between emergency and non-emergency cases while considering the priority of emergency cases using TD factor. This section presents the results of comparing PFQ and PFQ-TD in terms of average delay and jitter.

4.4.1 Emergency Case

Figures 4.5 and 4.6 compare the performance of PFQ and PFQ-TD with respect to the average delay and average jitter of emergency cases for ECG, Glucose, and Pressure sensors. It is clear that PFQ-TD performed nearly the same as PFQ up to 8 patients for all sensors. Then, PFQ outperformed PFQ-TD at high traffic volume of 12 patients.

It can be noticed that PFQ-TD maintains the average delay of emergency cases not too high as compared to WFQ and FIFO. In fact, PFQ-TD doesn't give priority to emergency cases all the time but rather it implements the tolerated delay concept and sets priority accordingly to balance the emergency and nonemergency cases.

Table 4.4 refers to scenario 1, sub scenarios, models and emergency delay and jitter results referenced figures.

Table 4.4. Scenario 1 Emg Delay & Jitter Results Referenced Figures

Scenario	Patient	Readings	Model	Emg-Delay Results	Emg-Jitter Results
SC1-Sub1	1	3	PFQ PFQ-TD	Fig 4.5	Fig 4.6
SC1-Sub2	4	12		Fig 4.5	Fig 4.6
SC1-Sub3	8	24		Fig 4.5	Fig 4.6
SC1-Sub4	12	36		Fig 4.5	Fig 4.6

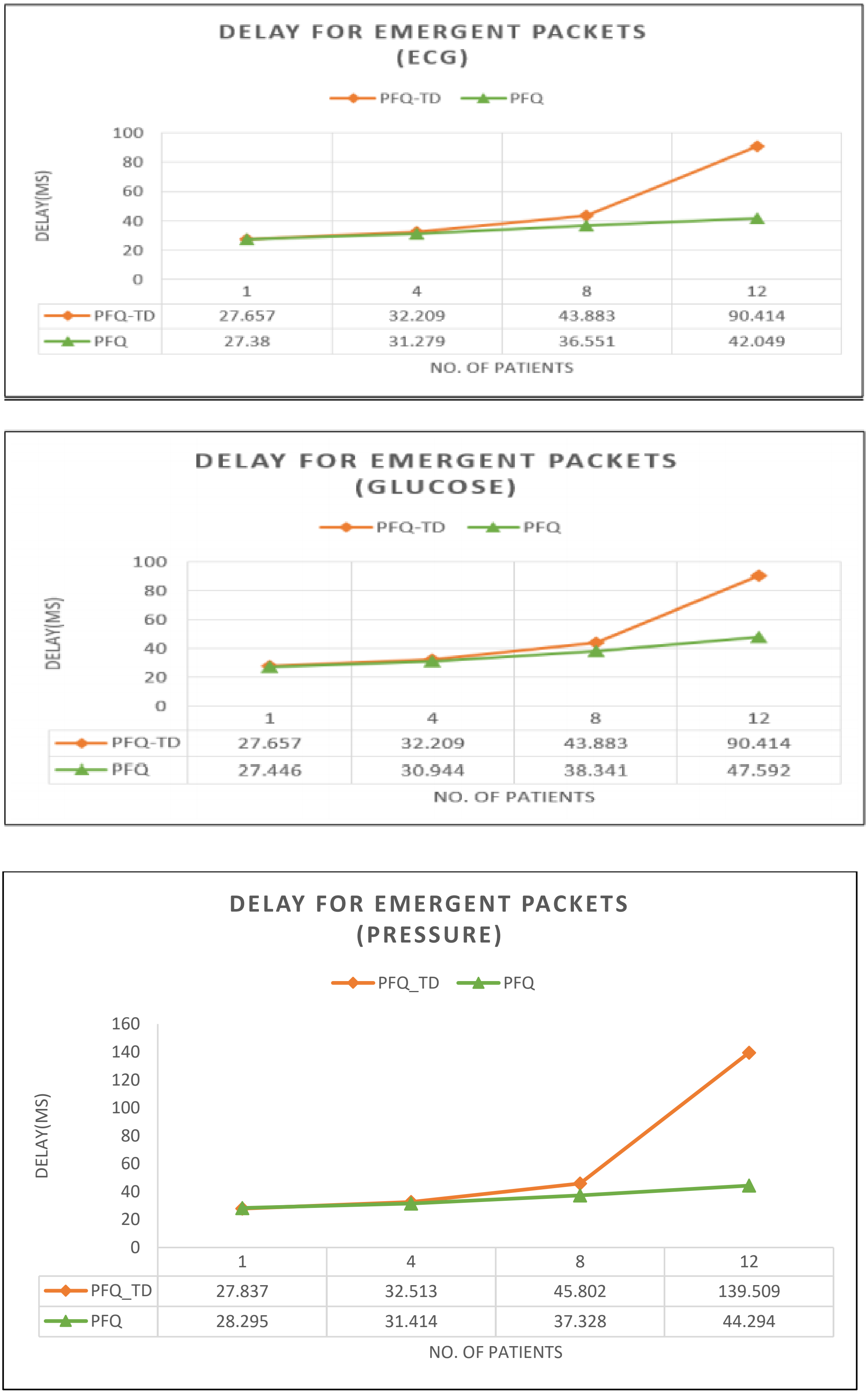
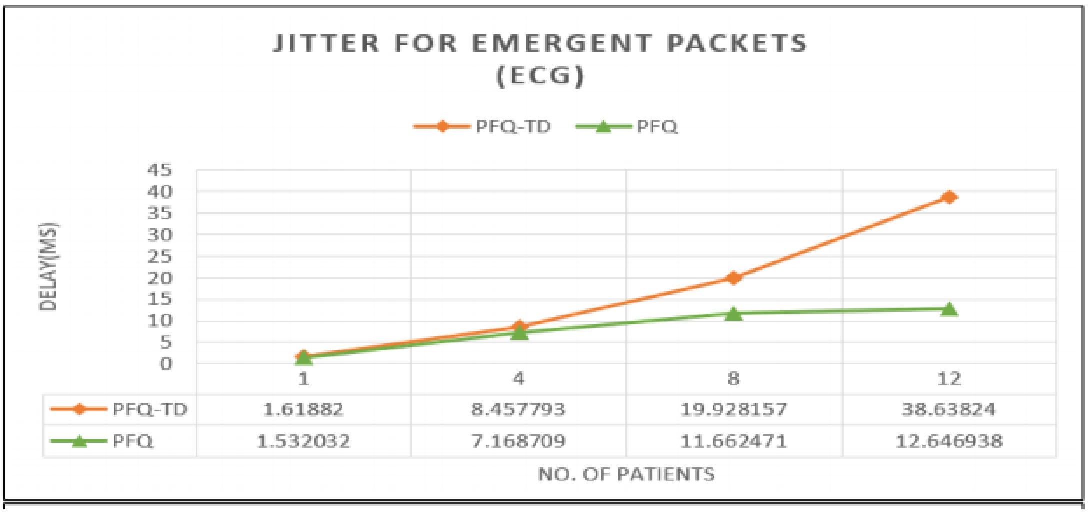
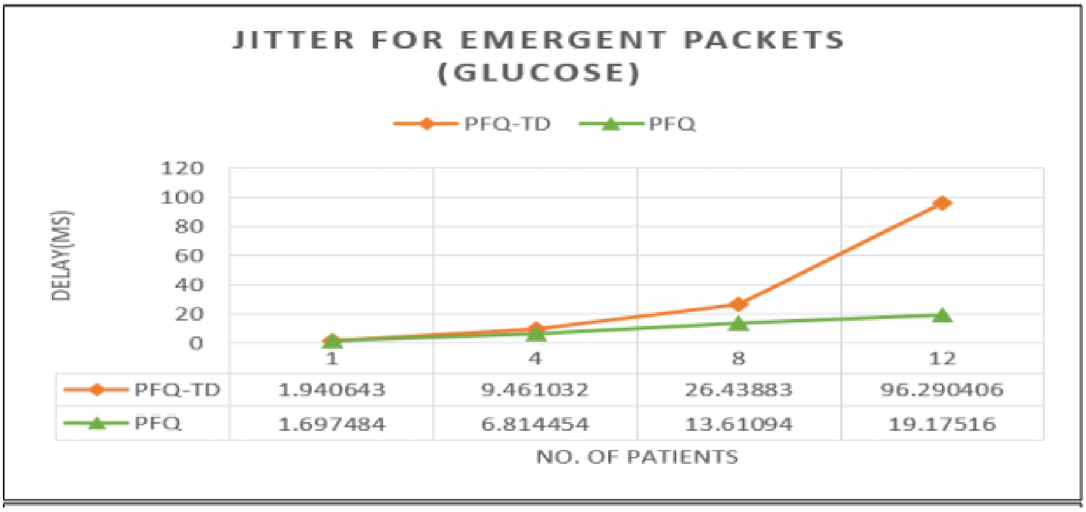


Figure 4.5. Av. Delay of Emergency Cases (ECG, Glucose & Pressure Sensors)





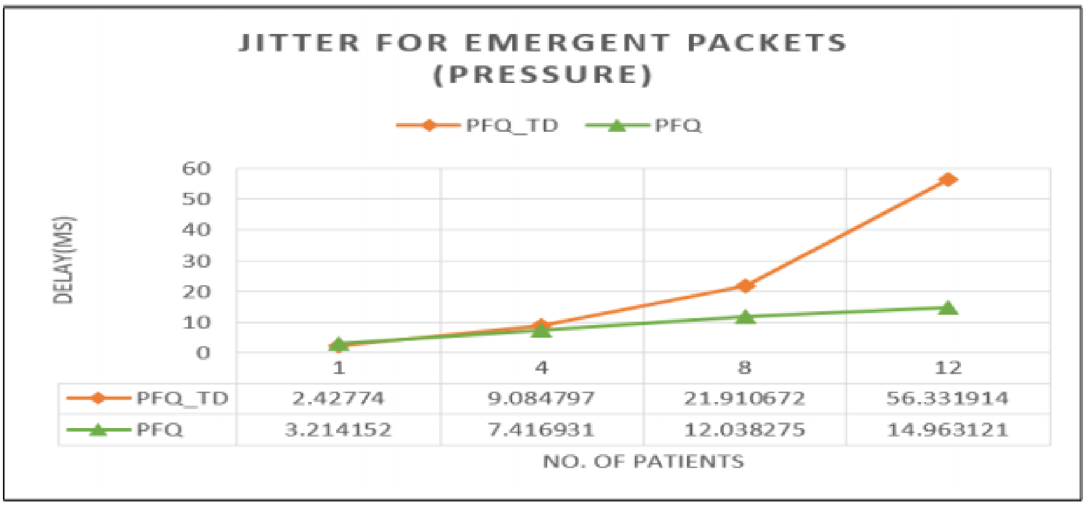


Figure 4.6. Av. Jitter of Emergency Cases (ECG, Glucose & Pressure Sensors)

4.4.2 Non-Emergency Case

The average delay and average jitter of non-emergency cases for ECG, Glucose, and Pressure sensors are presented in Figure 4.7 and Figure 4.8 respectively. The figures show that both models scored almost the same results up to patient number 8 yet at patient 12 (high traffic volume) PFQ-TD outperformed PFQ model in all sensors. This is because PFQ-TD model doesn't give priority to emergency cases all the time. Hence, PFQ-TD succeeded to keep the average delay and jitter quite lower than PFQ while balancing the emergency signals transmission within the TD as shown in previous section.

Table 4.5 refers to scenario 1, sub scenarios, models and emergency delay and jitter results referenced figures.

Table 4.5. Scenario 1 Non-Emg Delay & Jitter Results Referenced Figures

Scenario	**Patient**	**Readings**	**Model**	**Non Emg-Delay Results**	**Non Emg-Jitter Results**
SC1-Sub1	1	3	PFQ PFQ-TD	Fig 4.7	Fig 4.8
SC1-Sub2	4	12		Fig 4.7	Fig 4.8
SC1-Sub3	8	24		Fig 4.7	Fig 4.8
SC1-Sub4	12	36		Fig 4.7	Fig 4.8

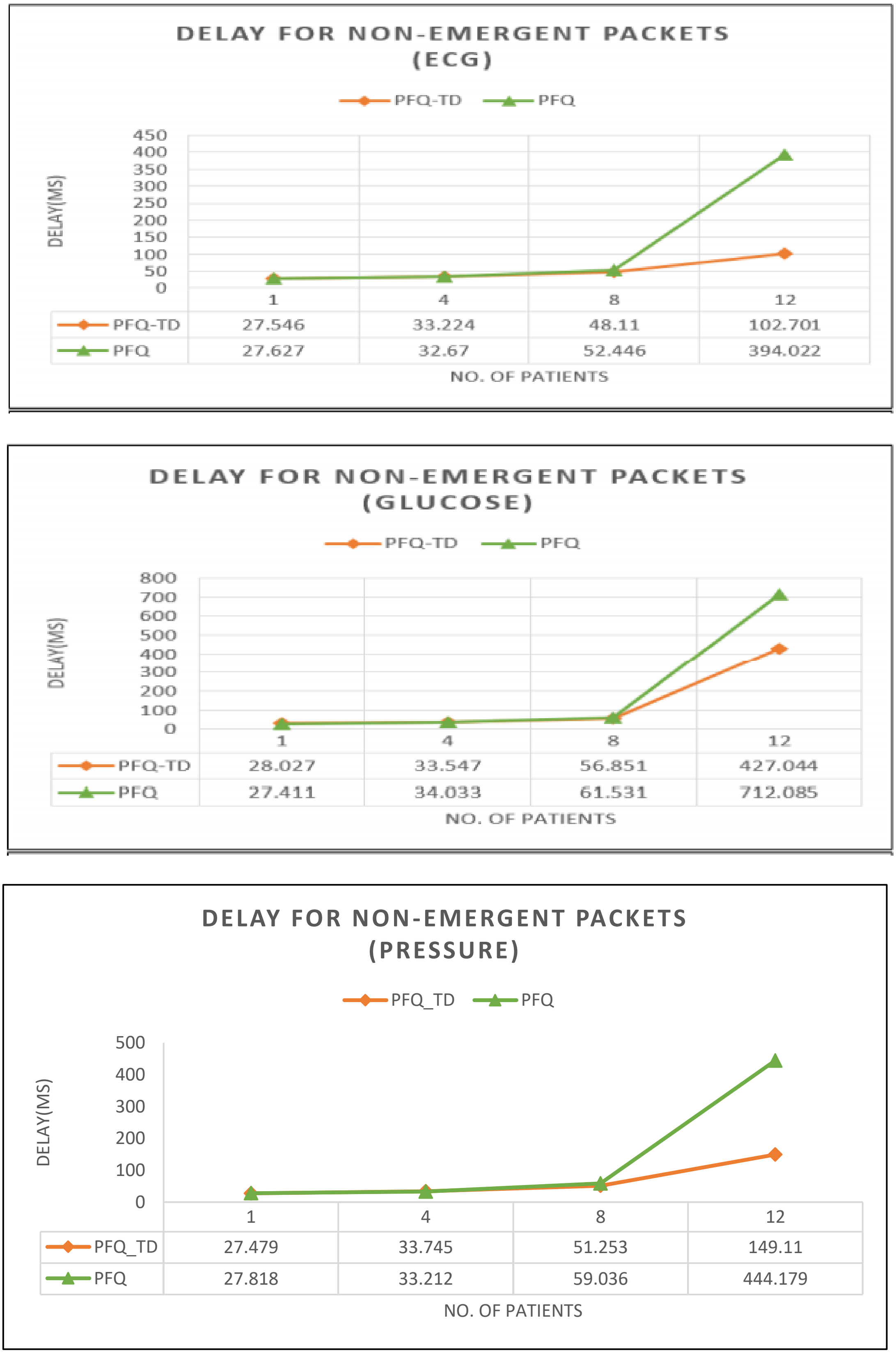


Figure 4.7. Av. Delay of Non-Emerg Cases (ECG, Glucose & Pressure Sensors)

Figure 4.8. Av. Jitter of Non-Emerg Cases (ECG, Glucose & Pressure Sensors)

4.5 Average Delay and Jitter Results of the Models

In summary, according to the data presented above, PFQ-TD model has balanced the delay and jitter results in emergency and non-emergency cases over the PFQ model and eventually FIFO and WFQ. Table 4.6 shows that PFQ-TD has reduced the average delay in emergency and non-emergency cases in all sensors at the highest patient number 12 compared to the other models in particular PFQ model by 40%.

Table 4.6. Average Delay of Emg & Non-Emg of all models

		PFQ-TD Delay Results at patient 12	PFQ Delay Results at patient 12	WFQ Delay Results at patient 12	FIFO Delay Results at patient 12
Pressure	Non-Emergent	149.11	444.18	142.32	248.52
	Emergent	139.509	44.29	140.32	251.79
Glucose	Non-Emergent	427.04	712.09	618.90	249.20
	Emergent	90.41	47.59	607.67	248.53
ECG	Non-Emergent	102.70	394.02	81.11	250.62
	Emergent	90.41	42.05	78.81	252.15
Average		**166.53**	**280.70**	**278.19**	**250.14**

Similarly, PFQ-TD has reduced the average jitter in emergency and non-emergency cases in all sensors at the highest patient number 12 compared to the other models in particular PFQ model by 84% as shown in Table 4.7.

Table 4.7. Average Jitter of Emg & Non-Emg of all models

		PFQ-TD Jitter Results at patient 12	PFQ Jitter Results at patient 12	WFQ Jitter Results at patient 12	FIFO Jitter Results at patient 12
Pressure	Non-Emergent	44.75	560.06	42.10	35.75
	Emergent	56.33	14.96	56.16	52.63
Glucose	Non-Emergent	54.31	912.61	72.84	35.95
	Emergent	96.29	19.18	113.93	54.31
ECG	Non-Emergent	32.46	529.07	25.52	36.59
	Emergent	38.64	12.65	31.72	52.43
Average		**53.79**	**341.42**	**57.05**	**44.61**

4.6 Analyzing PFQ Model under Scenario 2 'High Traffic Volume'

This scenario has been performed to study the performance of PFQ model in emergency cases under a high traffic volume. The simulation was performed from low to high traffic volume where at sub-scenario 1 (SC2-Sub1) one (01) patient sends three (03) readings one from each of the three (03) sensors; heart rate (ECG), Blood Pressure and Glucose simultaneously. Following this, in sub-scenario 2 (SC2-Sub2) four (04) patients send twelve (12) readings, in sub-scenario 3 (SC2-Sub3) eight (08) patients send twenty-four (24) readings, in sub-scenario 4 (SC2-Sub4) twelve (12) patients send thirty-six (36) readings, in sub-scenario 5 (SC2-Sub5) twenty (20) patients send sixty (60) readings, in sub-scenario 6 (SC2-Sub6) thirty (30) patients send ninety (90) readings, in sub-scenario 7 (SC2-Sub7) forty (40) patients send one hundred twenty (120) readings, and in sub-scenario 8 (SC2-Sub8) one fifty (50) patients send one hundred fifty (150) readings.

Table 4.8 summarize the high-level structure of scenario 2, sub scenarios, model and results referenced figures.

Table 4.8. High Level Structure of Scenario 2 and Sub Scenarios

Scenario 2 ''High Traffic Volume"	Number of Patient	Number of Readings	Model	Results
SC2-Sub1	1	3	PFQ	Fig 4.9
SC2-Sub2	4	12	PFQ	Fig 4.9
SC2-Sub3	8	24	PFQ	Fig 4.9
SC3-Sub4	12	36	PFQ	Fig 4.9
SC3-Sub5	20	60	PFQ	Fig 4.9
SC3-Sub6	30	90	PFQ	Fig 4.9
SC3-Sub7	40	120	PFQ	Fig 4.9
SC3-Sub8	50	150	PFQ	Fig 4.9

Figure 4.9 shows the performance of PFQ at high traffic volume for emergency cases of ECG, Glucose, and pressure sensors, with respect to average delay and jitter. It is inferred from the previous results that PFQ still performs good for emergency cases at patient number 40 of in terms of average delay and jitter.

Hence, this scenario was focused only the performance of PFQ in emergency cases under high traffic volume (i.e., 50 patients). It is clear that PFQ succeeded to keep the average delay and jitter below 65 msec up to 40 patients, for ECG sensor. Following this, a significant increase is noticed in terms of average delay and jitter at 50 patients, due to resource starvation at LDPU. Similarly, the average delay and jitter for emergency cases of Glucose and Pressure sensors increased significantly at 50 patients. The average delay and jitter of emergency cases of Glucose and Pressure sensor are higher than the average delay for emergency cases of ECG, because the latter has assigned a higher data criticality value in the PP.

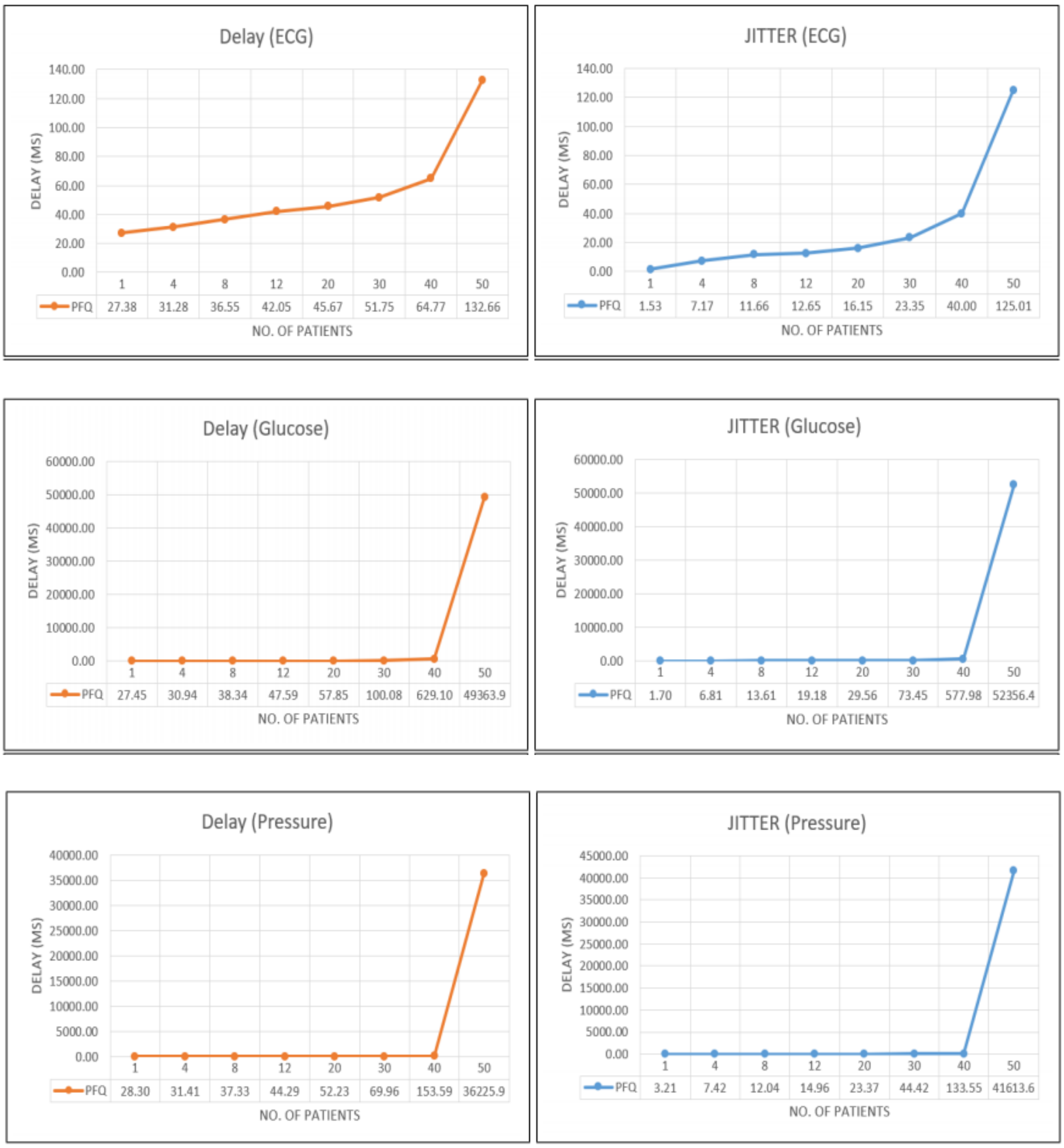


Figure 4.9. PFQ Performance at High Traffic Volume in Emergency Cases

4.7 Analyzing PFQ Model under Scenario 3 'Variant Emergency Rate'

This scenario has been simulated to study how PFQ performs for different rate of emergency cases, with respect to average delay and jitter. The simulation was performed from low to high rate of emergency cases volume where twelve (12) patients send thirty-six (36) readings, one from each of the three (03) sensors; heart rate (ECG), Blood Pressure and Glucose simultaneously. In the first sub-scenario (SC3-Sub1), the readings varied from (7) emergency (29) non-emergency i.e. 20% emergency signals and 80% non-emergency signals. In the second sub-scenario (SC3-Sub2), these readings varied from (14) emergency (22) non-emergency i.e. 40% emergency signals and 60% non-emergency signals. In the third sub-scenario (SC3-Sub3), the readings varied from (22) emergency (14) non-emergency i.e. 60% emergency signals and 40% non-emergency signals and in the fourth sub-scenario (SC3-Sub4), the readings varied from (29) emergency (7) non-emergency cases i.e. 80% emergency signals and 20% non-emergency signals.

Table 4.9 summarize the high-level structure of scenario 3, sub scenarios, model and results referenced figures.

Table 4.9. High Level Structure of Scenario 3 and Sub Scenarios

Scenario 3 'Variant Emergency Rate"	Number of Patient	Number of Readings	Emergency Readings	Non-Emg Readings	Model	Results
SC2-Sub1	12	36	7 (20%)	29 (80%)	PFQ	Fig 4.10
SC2-Sub2	12	36	14 (40%)	22 (60%)	PFQ	Fig 4.10
SC2-Sub3	12	36	22 (60%)	14 (40%)	PFQ	Fig 4.10
SC3-Sub4	12	36	29 (80%)	7 (20%)	PFQ	Fig 4.10

Figure 4.10 shows the performance of PFQ at variant emergency rates in emergency and non-emergency cases in ECG, Glucose, and pressure sensors, with respect to average delay and jitter. It is noticed that the average delay and jitter for emergency cases for all sensors are always quite low because PFQ gives priority to emergency cases all the time. The figure shows that increasing the emergency rate has caused an increase in the average delay and jitter for non-emergency cases in all sensors. This because PFQ model always transmits emergency cases before non-emergency cases.

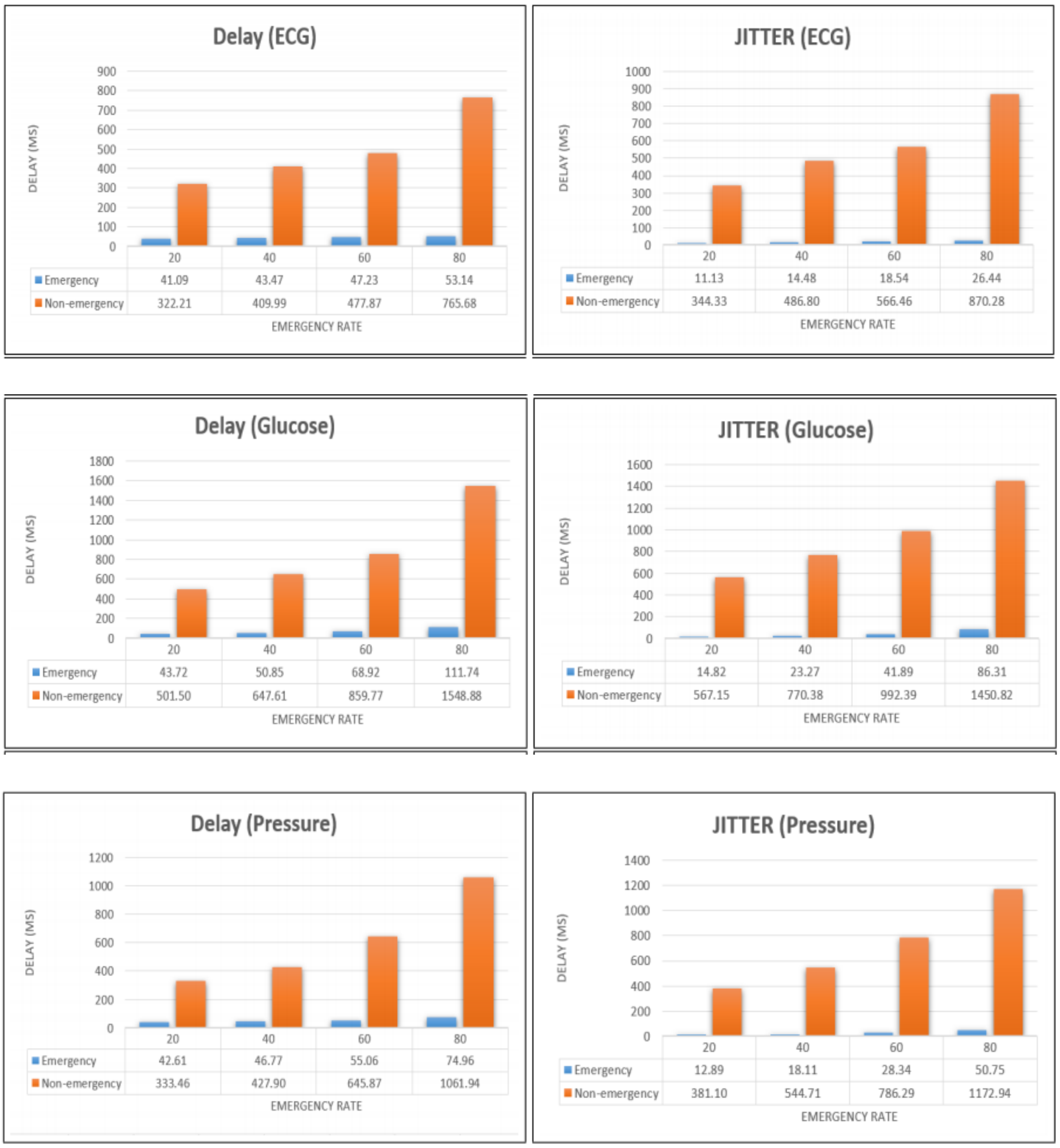


Figure 4.10. PFQ-TD Performance at Variant Emergency Rate

4.8 Numerical and Graphical Representations of PFQ and PFQ-TD improvements over FIFO and WFQ Models

This section illustrates the percentile improvement achieved in reducing and balancing the delay and jitter in emergency and non-emergency cases using PFQ and PFQ-TD models compared to FIFO and WFQ models.

4.8.1 PFQ vs FIFO and WFQ Delay and Jitter Performance in Emergency and Non-Emergency cases

Based on the data results of scenario 1 and its sub scenarios i.e. number of patients, it is evident that PFQ outperformed FIFO with respect to delay and jitter for emergency cases. By implementing PFQ model, delay has been reduced significantly in the ECG sensor by 83.32% when compared to FIFO model and 46.66% when compared to WFQ model at patient 12 in emergency cases. Similarly, Jitter has been reduced by implementing the PFQ model in emergency cases by 75.88% when compared to FIFO model and 60.13% when compared to WFQ model.

These significant improvements achieved by using the PFQ model have satisfied the aim of this research in the most critical senor i.e. ECG in emergency cases but on the account of the non-emergency cases where it suffered from starvation and negatively impacted the delay and jitter. In non-emergency cases PFQ model did not improve the delay and jitter but rather increased them due to the fact the PFQ model gave priority for emergency cases all the time.

Table 4.10 shows the improvements and drawbacks in percentile of the PFQ model vs FIFO and WFQ models with respect to Delay and Jitter in ECG in emergency and non-emergency cases.

Table 4.10. PFQ Delay and Jitter Performance vs FIFO and WFQ in ECG Sensor in Emg and Non-Emg cases

	EMG DELAY				NON-EMG DELAY				EMG JITTER				NON-EMG JITTER			
	1	4	8	12	1	4	8	12	1	4	8	12	1	4	8	12
FIFO	27.581	33.043	49.357	252.152	27.828	33.469	49.02	250.623	1.955	9.787	24.259	52.429	2.310	9.864	20.671	36.591
WFQ	27.893	31.653	44.207	78.814	27.95	32.281	43.793	81.113	2.514	8.027	20.783	31.718	2.446	8.381	18.152	25.517
PFQ	27.38	31.279	36.551	42.049	27.627	32.67	52.446	394.022	1.532	7.169	11.662	12.647	1.855	9.387	27.996	529.074
PFQ vs FIFO	0.73%	5.34%	25.95%	83.32%	0.72%	2.39%	-6.99%	-57.22%	21.62%	26.76%	51.92%	75.88%	19.72%	4.83%	-35.43%	-1345.92%
PFQ vs WFQ	1.84%	1.18%	17.32%	46.65%	1.16%	-1.21%	-19.76%	-385.77%	39.05%	10.69%	43.88%	60.13%	24.16%	-11.99%	-54.23%	-1973.39%

Figure 4.11 to Figure 4.14 shows the improvement and drawback trends in delay and jitter using PFQ model over FIFO and WFQ models in ECG in emergency and non-emergency cases.

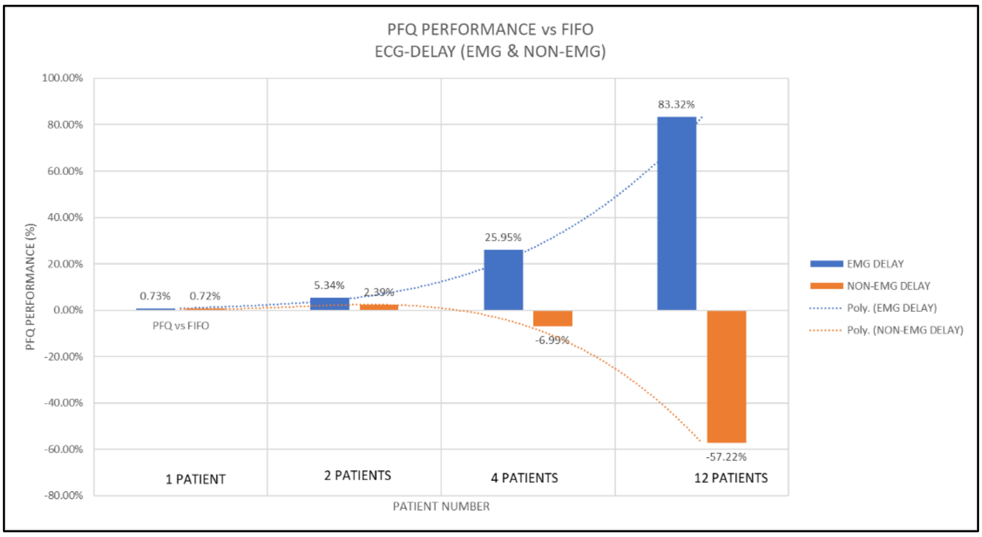


Figure 4.11. PFQ vs FIFO Delay Performance in ECG- (Emg & Non-Emg)

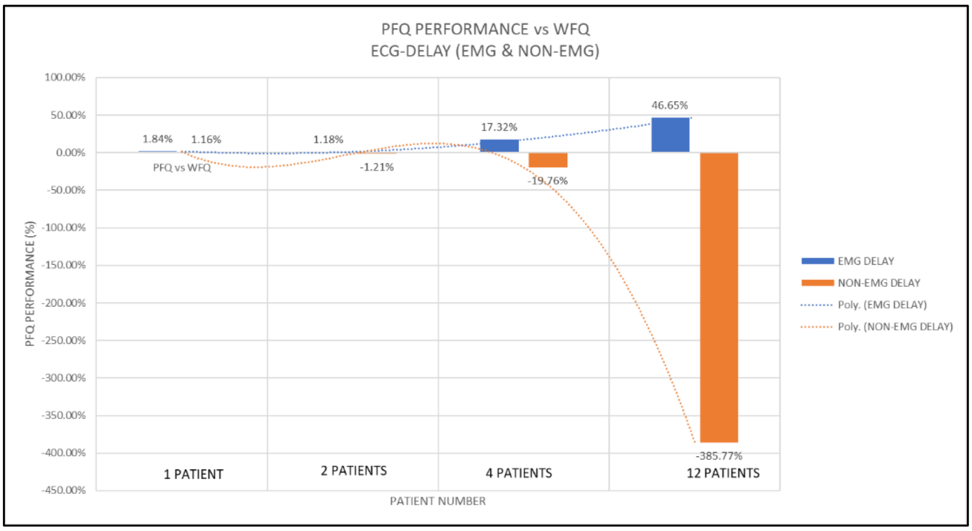


Figure 4.12. PFQ vs WFQ Delay Performance in ECG- (Emg & Non-Emg)

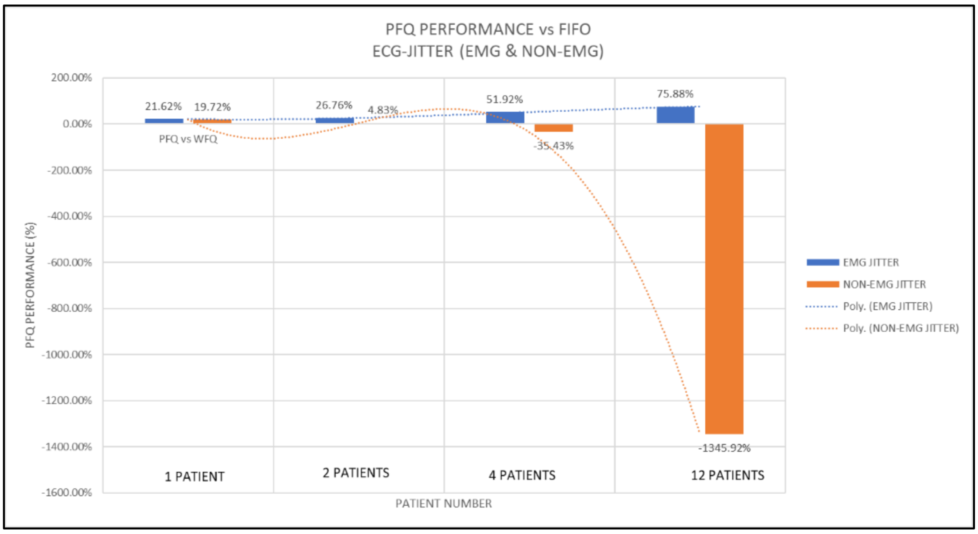


Figure 4.13. PFQ vs FIFO Jitter Performance in ECG- (Emg & Non-Emg)

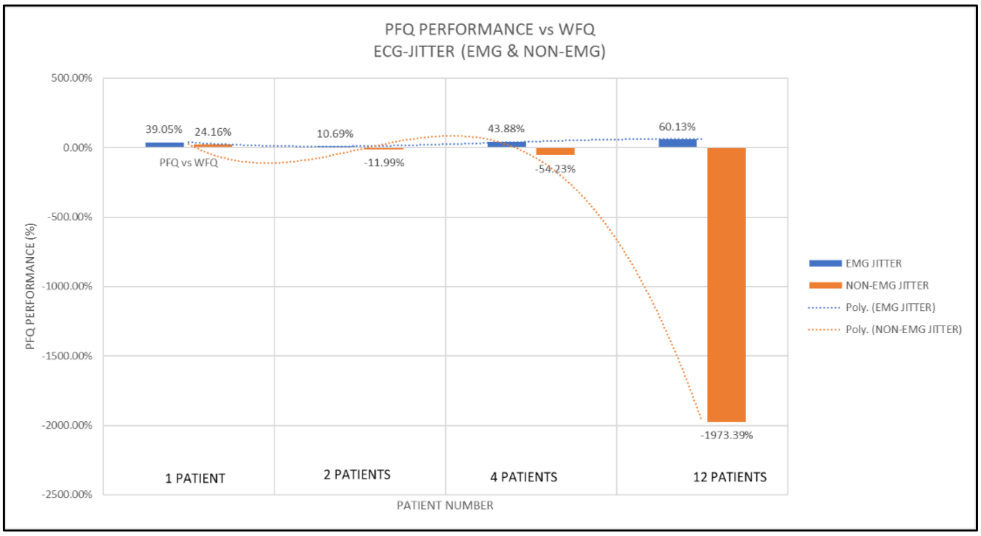


Figure 4.14. PFQ vs WFQ Jitter Performance in ECG- (Emg & Non-Emg)

Similarly, in the Glucose sensor PFQ model out performed FIFO and WFQ models in emergency cases and had similar draw back in non-emergency cases. Based on the scenario 1 and its sub scenarios results, it is obvious that PFQ model has improved the delay and jitter for emergency cases. By implementing PFQ model, delay has been reduced significantly in the glucose sensor by 80.85% when compared to FIFO model and 92.1% when compared to WFQ model at patient 12 in emergency cases.

Similarly, Jitter has been reduced by implementing the PFQ model in emergency cases by 64.7% when compared to FIFO model and 83.17% when compared to WFQ model.

These significant improvements achieved by using the PFQ model have satisfied the aim of this research in the glucose sensor which aligns with the improvements achieved in the ECG sensor in emergency cases. Yet these improvements were on the account of the non-emergency cases where it suffered from starvation and negatively impacted the delay and jitter. In non-emergency cases PFQ model did not improve the delay and jitter but rather increased them due to the fact the PFQ model gave priority for emergency cases all the time.

Table 4.11 indicates the improvements and drawbacks in percentile of the PFQ model vs FIFO and WFQ models with respect to Delay and Jitter in Glucose sensor in emergency and non-emergency cases.

Table 4.11. PFQ Delay and Jitter Performance vs FIFO and WFQ in Glucose Sensor in Emg & Non-Emg cases

	EMG DELAY				NON-EMG DELAY				EMG JITTER				NON-EMG JITTER			
	1	4	8	12	1	4	8	12	1	4	8	12	1	4	8	12
FIFO	27.306	33.202	48.901	248.529	27.699	32.672	48.813	249.198	1.418	9.781	24.047	54.314	2.085	8.796	20.577	35.948
WFQ	27.690	34.260	54.423	607.674	27.675	33.674	55.702	618.902	2.142	12.136	31.369	113.934	2.022	10.532	27.678	72.844
PFQ	27.446	30.944	38.341	47.592	27.411	34.033	61.531	712.085	1.697	6.814	13.611	19.175	1.621	11.523	35.104	912.615
PFQ vs FIFO	-0.51%	6.80%	21.59%	80.85%	1.04%	-4.17%	-26.05%	-185.75%	-19.73%	30.33%	43.40%	64.70%	22.26%	-31.00%	-70.60%	-2438.74%
PFQ vs WFQ	0.88%	9.68%	29.55%	92.17%	0.95%	-1.07%	-10.46%	-15.06%	20.74%	43.85%	56.61%	83.17%	19.83%	-9.41%	-26.83%	-1152.84%

Figure 4.15 to Figure 4.18 shows the improvement and drawback trends in delay and jitter using PFQ model over FIFO and WFQ models in glucose in emergency and non-emergency cases.

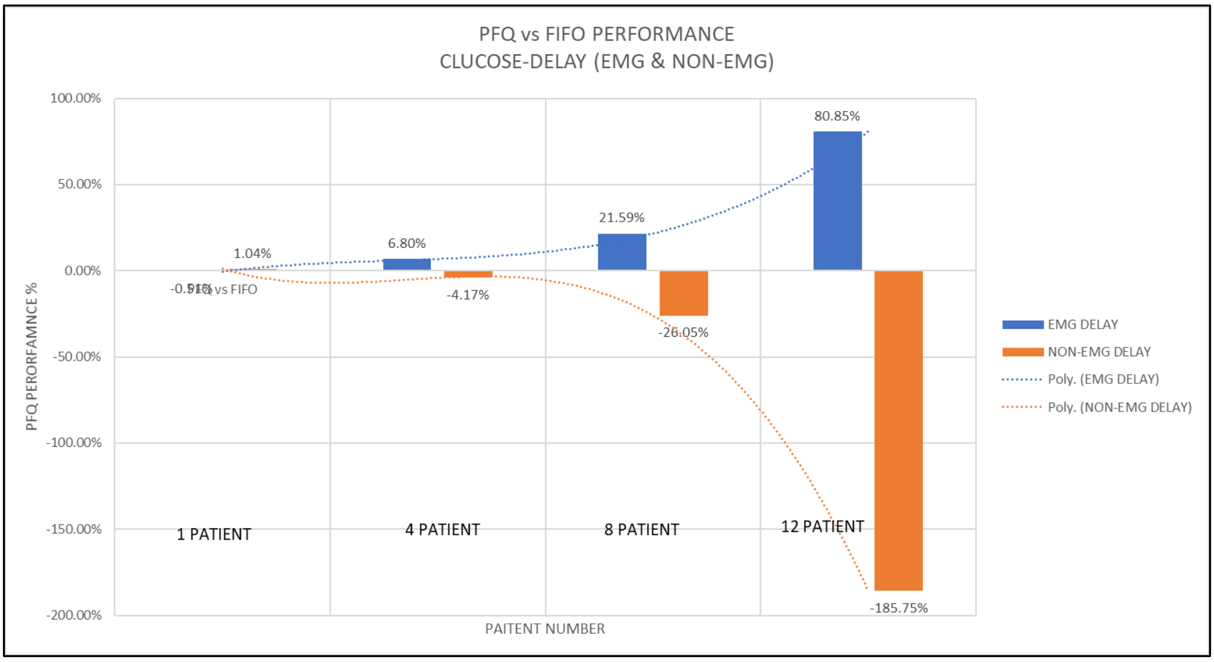


Figure 4.15. PFQ vs FIFO Delay Performance in Glucose- (Emg & Non-Emg)

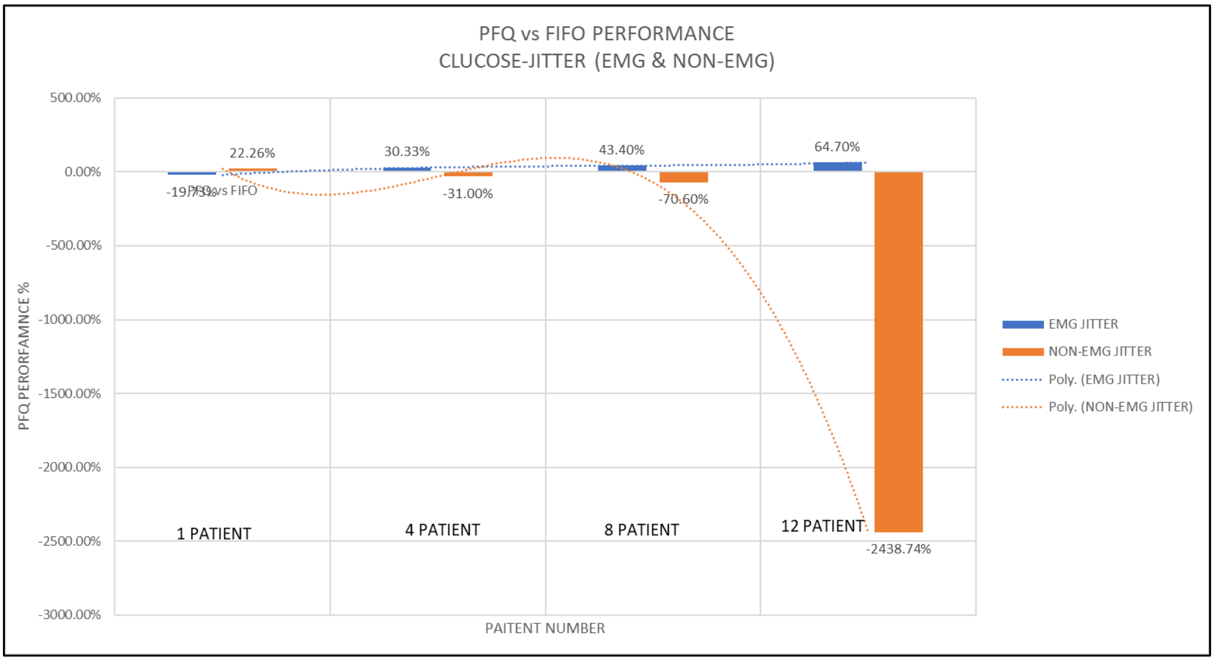


Figure 4.16. PFQ vs FIFO Jitter Performance in Glucose- (Emg & Non-Emg)

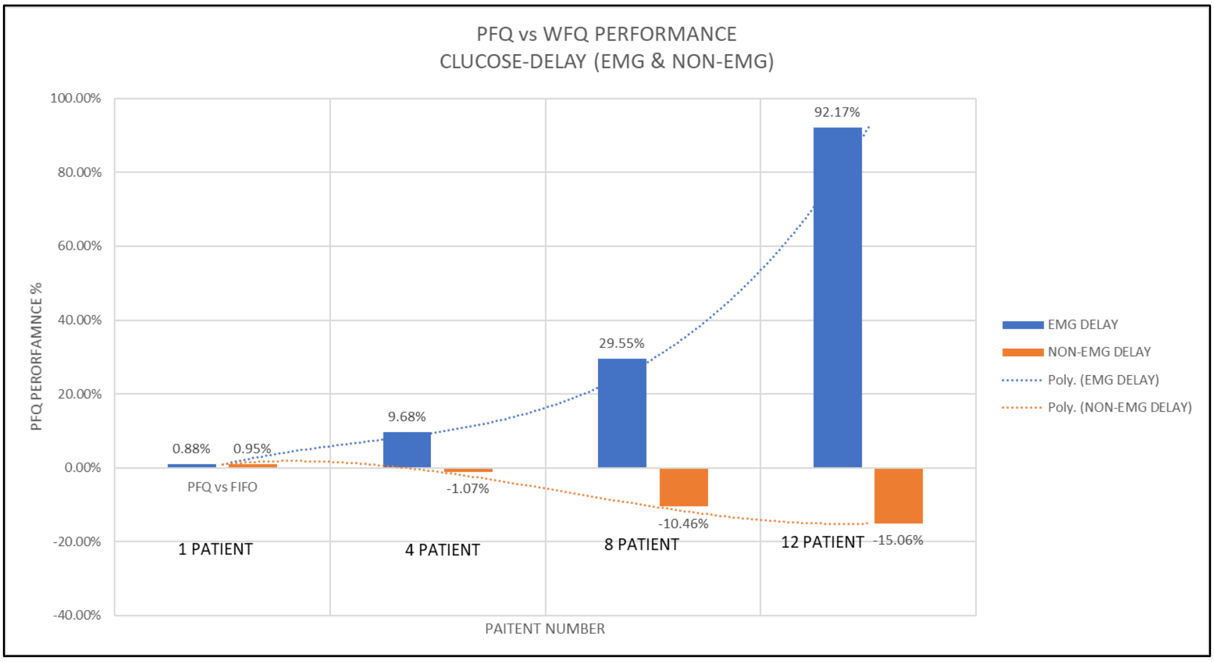


Figure 4.17. PFQ vs WFQ Delay Performance in Glucose- (Emg & Non-Emg)

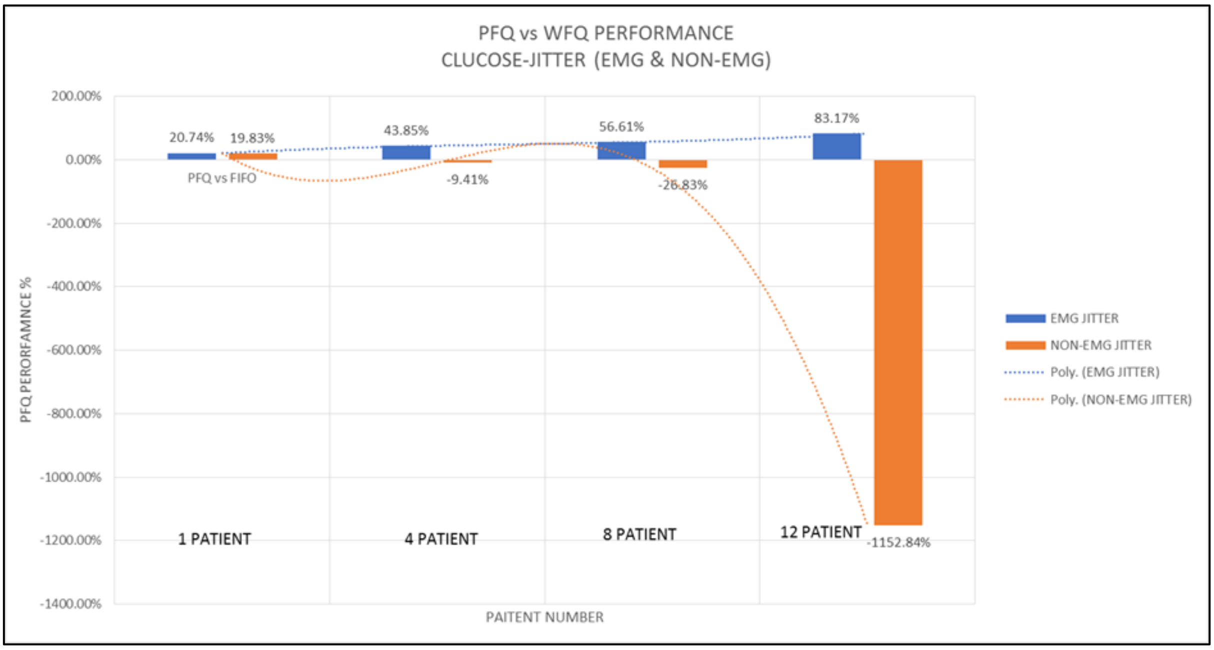


Figure 4.18. PFQ vs WFQ Jitter Performance in Glucose- (Emg & Non-Emg)

The PFQ model performed in the same fashion in Pressure sensor and has improved the delay and jitter over the FIFO and WFQ models for emergency cases while it did not improve the non-emergency cases.

Based on the scenario 1 and its sub scenarios results, it is apparent that PFQ model has improved the delay and jitter for emergency cases. Implementing the PFQ model, delay has been reduced significantly in the pressure sensor by 82.41% when compared to FIFO model and 68.43% when compared to WFQ model at patient 12 in emergency cases. Similarly, Jitter has been reduced by implementing the PFQ model in emergency cases by 71.57% when compared to FIFO model and 73.36% when compared to WFQ model.

These significant improvements are aligned with the results and improvement achieved in the ECG and glucose sensors which proofs that by using the PFQ model the aim of this research has been achieved. Also, the delay and jitter non-emergency cases in the pressure sensor was not improved but was negatively impacted and suffered from starvation. In non-emergency cases PFQ model did not improve the delay and jitter but rather increased them due to the fact the PFQ model gave priority for emergency cases all the time.

Table 4.12 indicates the improvements and drawbacks in percentile of the PFQ model vs FIFO and WFQ models with respect to Delay and Jitter in pressure sensor in emergency and non-emergency cases.

Table 4.12. PFQ Delay and Jitter Performance vs FIFO and WFQ in Pressure Sensor in Emg & Non-Emg cases

	EMG DELAY				NON-EMG DELAY				EMG JITTER				NON-EMG JITTER			
	1	4	8	12	1	4	8	12	1	4	8	12	1	4	8	12
FIFO	27.631	32.87	49.263	251.793	27.605	33.299	48.994	248.522	2.069302	9.940708	24.1027	52.63445	1.950282	9.736172	20.07655	35.745676
WFQ	27.811	33.63	46.658	140.324	27.485	32.908	48.531	142.318	2.374719	11.12633	23.68091	56.16351	1.717354	9.646029	21.75994	42.102552
PFQ	28.295	31.414	37.328	44.294	27.818	33.212	59.036	444.179	3.214152	7.416931	12.03828	14.96312	2.351818	10.02881	35.0055	560.058431
PFQ vs FIFO	-2.40%	4.43%	24.23%	82.41%	-0.77%	0.26%	-20.50%	-78.73%	-55.33%	25.39%	50.05%	71.57%	-20.59%	-3.01%	-74.36%	-1466.79%
PFQ vs WFQ	-1.74%	6.59%	20.00%	68.43%	-1.21%	-0.92%	-21.65%	-212.10%	-35.35%	33.34%	49.16%	73.36%	-36.94%	-3.97%	-60.87%	-1230.22%

Figure 4.19 to Figure 4.22 shows the improvement and drawback trends in delay and jitter using PFQ model over FIFO and WFQ models in pressure sensor in emergency and non-emergency cases.

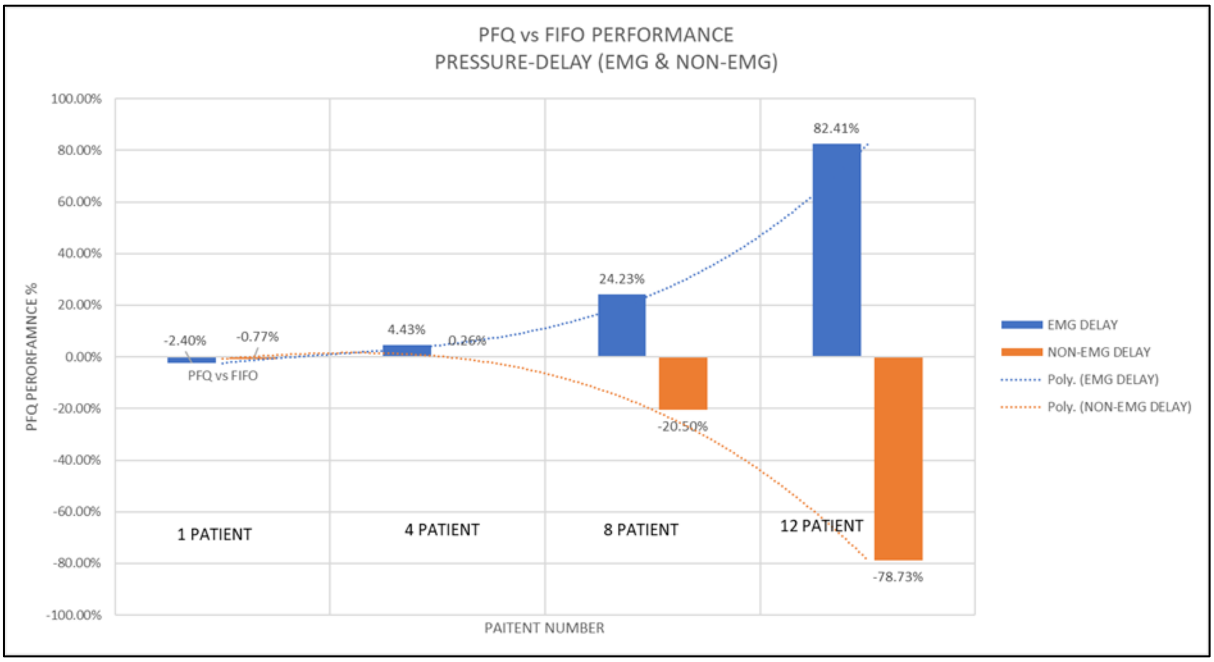


Figure 4.19. PFQ vs FIFO Delay Performance in Pressure- (Emg & Non-Emg)

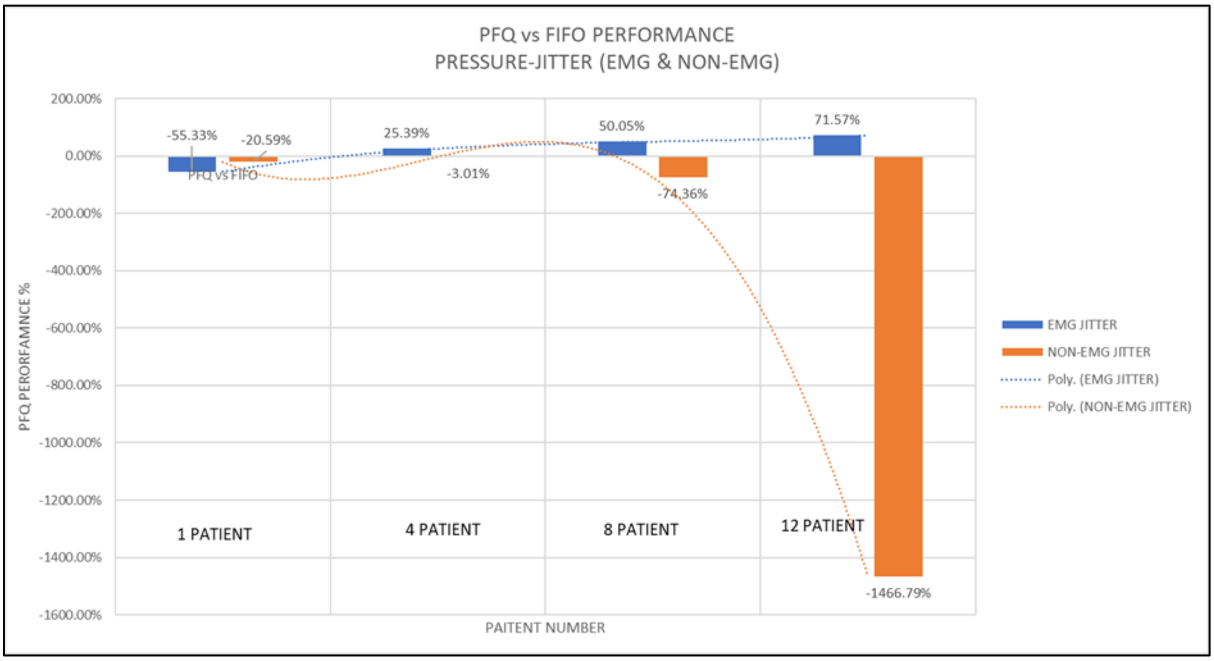


Figure 4.20. PFQ vs FIFO Jitter Performance in Glucose- (Emg & Non-Emg)

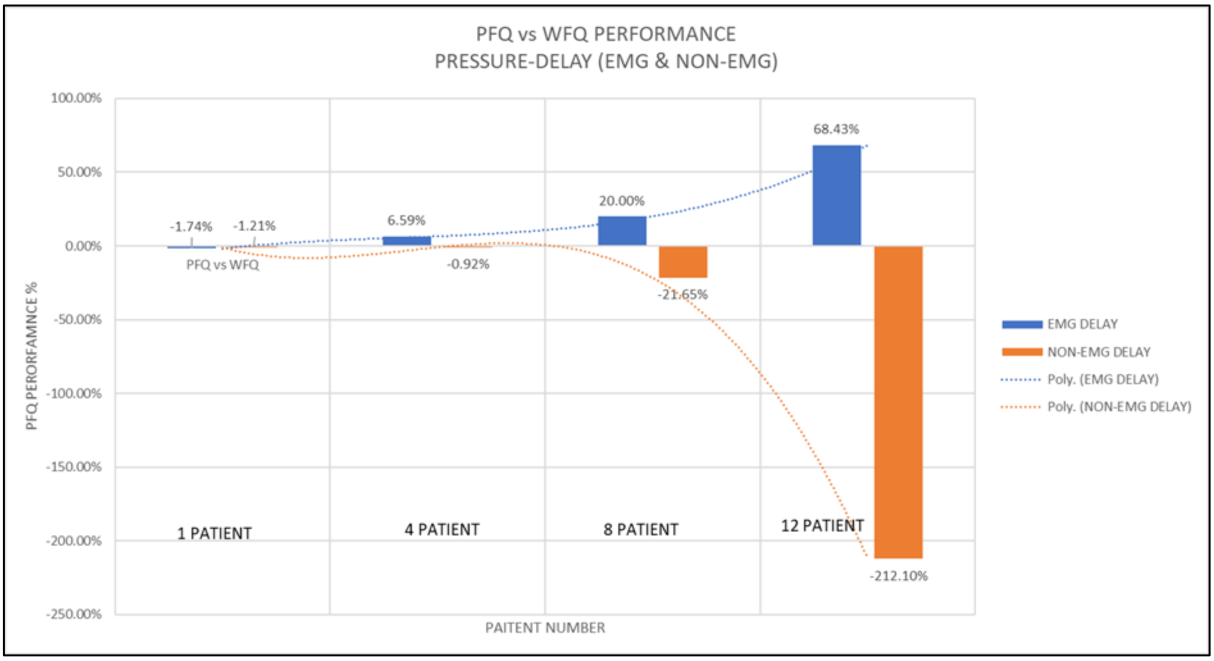


Figure 4.21. PFQ vs WFQ Delay Performance in Pressure- (Emg & Non-Emg)

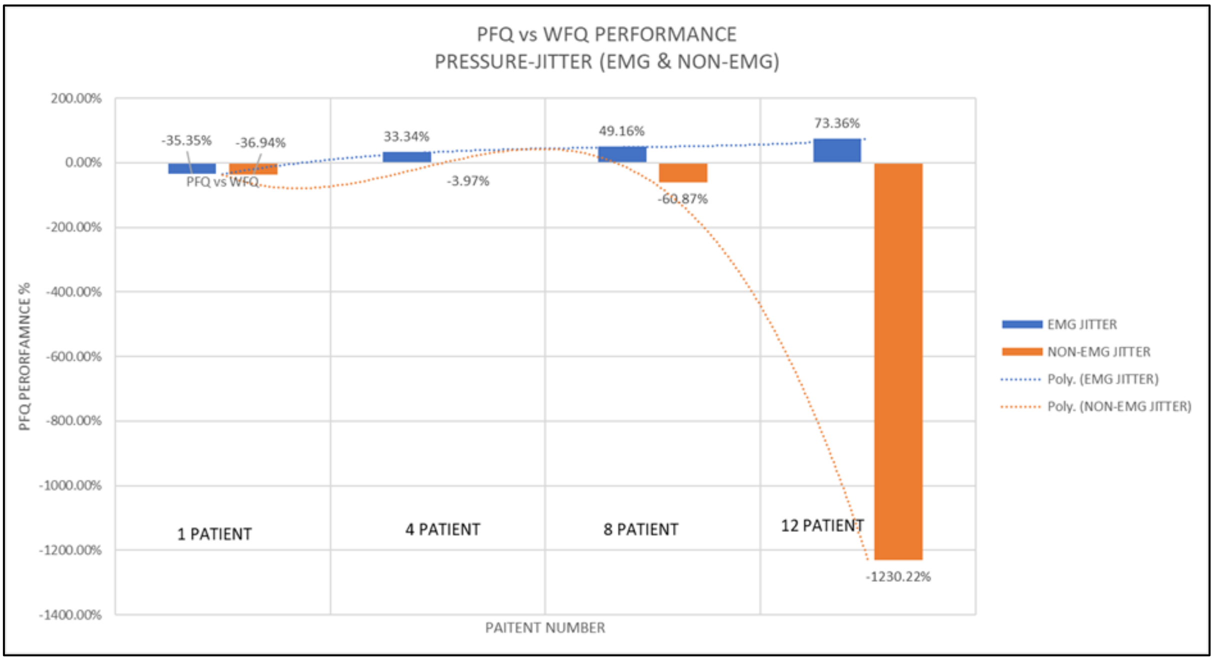


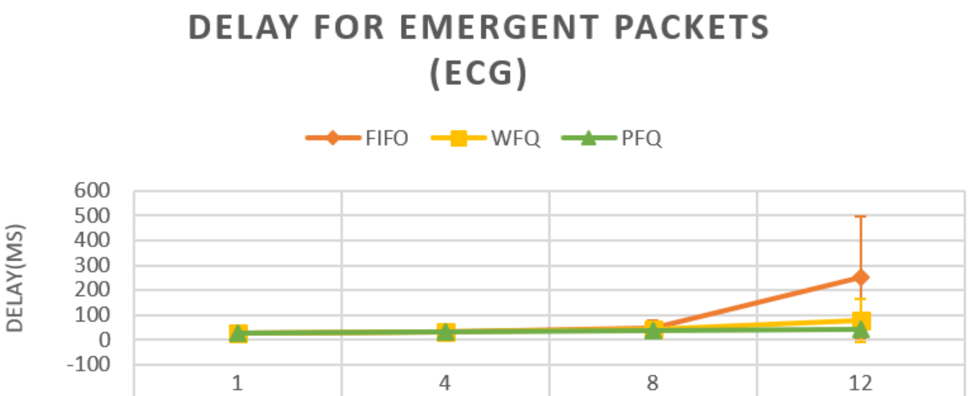
Figure 4.22. PFQ vs WFQ Jitter Performance in Pressure- (Emg & Non-Emg)

4.9 Calculated Standard Deviation

As described in the above sections, the test results have shown significant improvement in Delay and Jitter time for Emergency cases using the PFQ model in scenario 1 satisfying the main objective of the thesis. Tests simulation have been repeated 10 times where many signals or packets are transmitted each time as per scenario 1 setup. The mean (average) time was reported as the final result; whereas the calculated Standard Deviation has also shown minimum variations from the mean value for the PFQ model which implies that the obtained results are robust and the performance of the PFQ model is steady and within the acceptable range of change. Table 4.13 and Figure 4.23 below show the standard deviations of Emergency case for PFQ, WFQ and FIFO models.

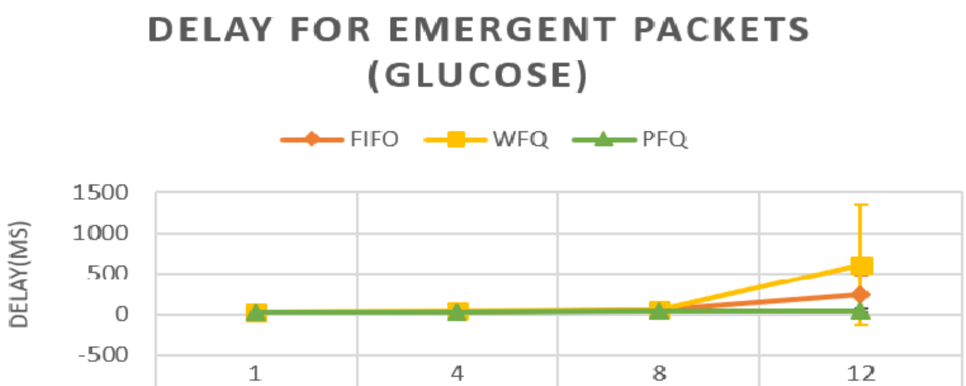
Table 4.13 Scenario 1: Emg Delay Av. & Standard Deviations

	ECG Average Delay			ECG Standard Deviation		
Patient #	FIFO	WFQ	PFQ	FIFO	WFQ	PFQ
1	27.581	27.893	27.38	4.7924	4.4072	4.4613
4	33.043	31.653	31.279	12.2788	10.9289	8.2221
8	49.357	44.207	36.551	30.8855	29.7589	11.2462
12	252.152	78.814	42.049	243.3205	87.6838	12.6419
	Glucose Average Delay			**Glucose Standard Deviation**		
Patient #	FIFO	WFQ	PFQ	FIFO	WFQ	PFQ
1	27.306	27.69	27.446	4.2323	5.1353	4.5282
4	33.202	34.26	30.944	12.1898	14.9165	9.4001
8	48.901	54.423	38.341	30.6687	43.6214	14.625
12	248.529	607.674	47.592	243.5154	736.4461	20.4569
	Pressure Average Delay			**Pressure Standard Deviation**		
Patient #	FIFO	WFQ	PFQ	FIFO	WFQ	PFQ
1	27.581	27.893	27.38	4.7924	4.4072	4.4613
4	33.043	31.653	31.279	12.2788	10.9289	8.2221
8	49.357	44.207	36.551	30.8855	29.7589	11.2462
12	252.152	78.814	42.049	243.3205	87.6838	12.6419



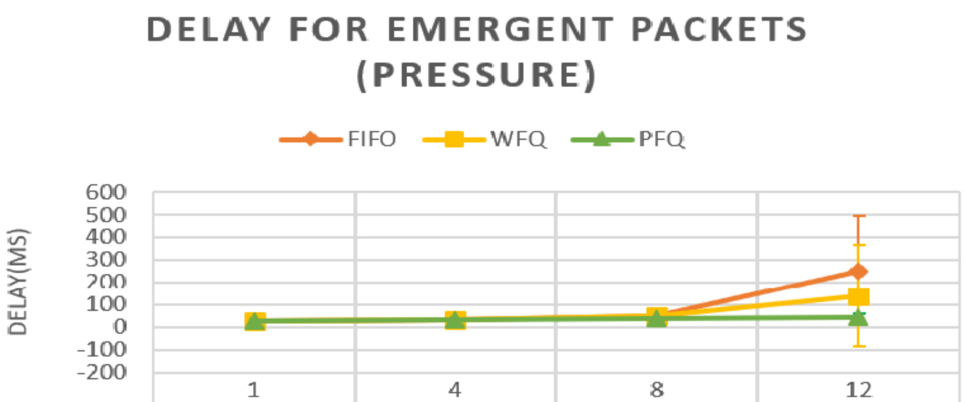
	1	4	8	12
FIFO	27.581	33.043	49.357	252.152
WFQ	27.893	31.653	44.207	78.814
PFQ	27.38	31.279	36.551	42.049

NO. OF PATIENTS



	1	4	8	12
FIFO	27.306	33.202	48.901	248.529
WFQ	27.69	34.26	54.423	607.674
PFQ	27.446	30.944	38.341	47.592

NO. OF PATIENTS



	1	4	8	12
FIFO	27.631	32.87	49.263	251.793
WFQ	27.811	33.63	46.658	140.324
PFQ	28.295	31.414	37.328	44.294

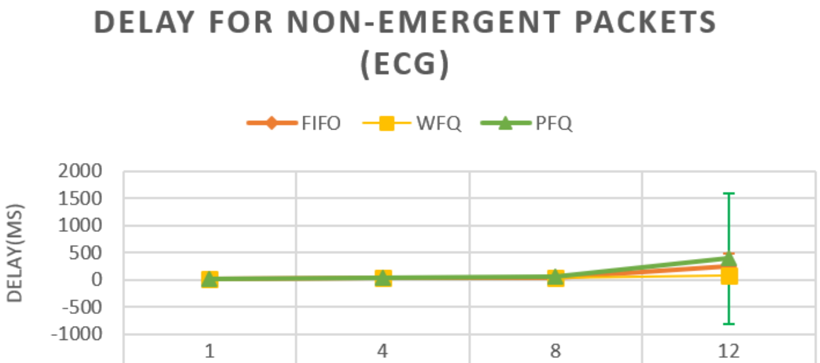
NO. OF PATIENTS

Figure 4.23 Scenario 1: Emergency Delay average and Standard Deviations

Similarly, for non-Emergency cases standard deviation was calculated for the Delay results in scenario 1 although and as mentioned in the previous sections that PFQ model did not perform as good as other models specially at patient 12 due to the fact that PFQ model is designed to give priority to Emergency cases all the time. Table 4.14 and Figure 4.24 below show the standard deviations of non-Emergency case for PFQ, WFQ and FIFO models. Standard Deviation has also been computed and represented in Figure 4.25 and Figure 4.26 for Non-Emergency and Emergency Delay results of the PFQ-TD model vs the PFQ model. It can be seen that PFQ-TD standard deviation range supports the improvement which was designed for and achieved in the non-Emergency Delay over the PFQ model.

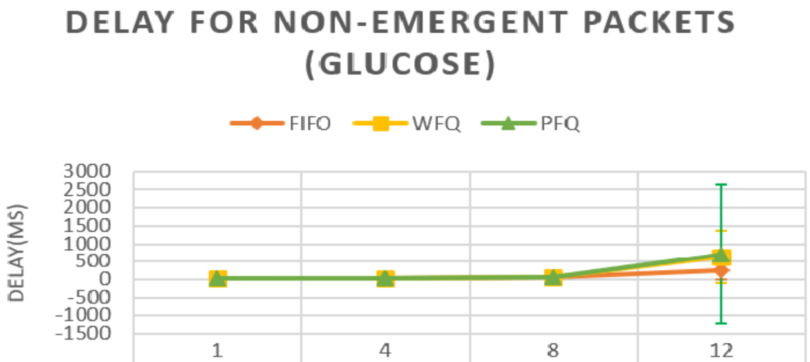
Table 4.14 Scenario 1: Non-Emg Delay Av. & Standard Deviations

	ECG Average Delay			ECG Standard Deviation		
Patient #	FIFO	WFQ	PFQ	FIFO	WFQ	PFQ
1	27.828	27.95	27.627	4.7529	4.3661	4.6018
4	33.469	32.281	32.67	12.2023	10.8423	13.6014
8	49.02	43.793	52.446	30.483	29.6985	45.4022
12	250.623	81.113	394.022	242.7173	89.293	1200.354
	Glucose Average Delay			Glucose Standard Deviation		
Patient #	FIFO	WFQ	PFQ	FIFO	WFQ	PFQ
1	27.699	27.675	27.411	4.5058	5.4998	4.876
4	32.672	33.674	34.033	12.144	14.9514	16.2342
8	48.813	55.702	61.531	30.5085	44.8311	59.5667
12	249.198	618.902	712.085	242.5075	733.253	1922.754
	Pressure Average Delay			Pressure Standard Deviation		
Patient #	FIFO	WFQ	PFQ	FIFO	WFQ	PFQ
1	27.605	27.485	27.818	4.4248	4.9329	4.9362
4	33.299	32.908	33.212	12.1422	13.2426	15.0282
8	48.994	48.531	59.036	30.8321	36.3935	51.8176
12	248.522	142.318	444.179	241.8592	227.0232	1665.75



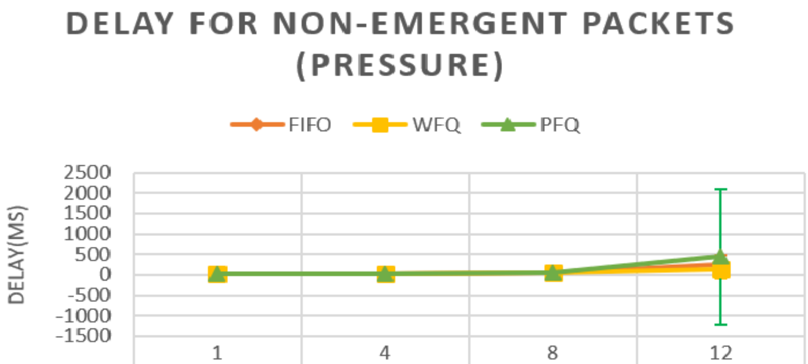
	1	4	8	12
FIFO	27.828	33.469	49.02	250.623
WFQ	27.95	32.281	43.793	81.113
PFQ	27.627	32.67	52.446	394.022

OF PATIENTS



	1	4	8	12
FIFO	27.699	32.672	48.813	249.198
WFQ	27.675	33.674	55.702	618.902
PFQ	27.411	34.033	61.531	712.085

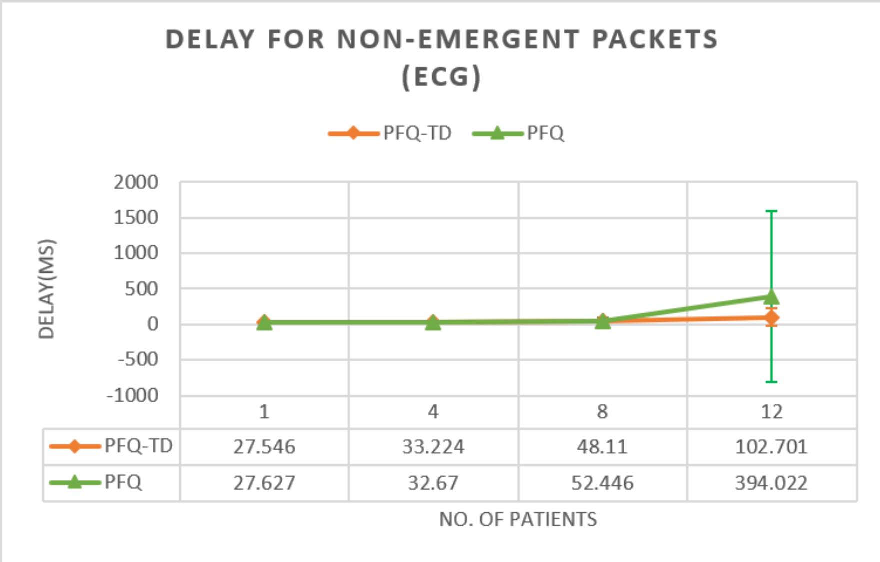
NO. OF PATIENTS

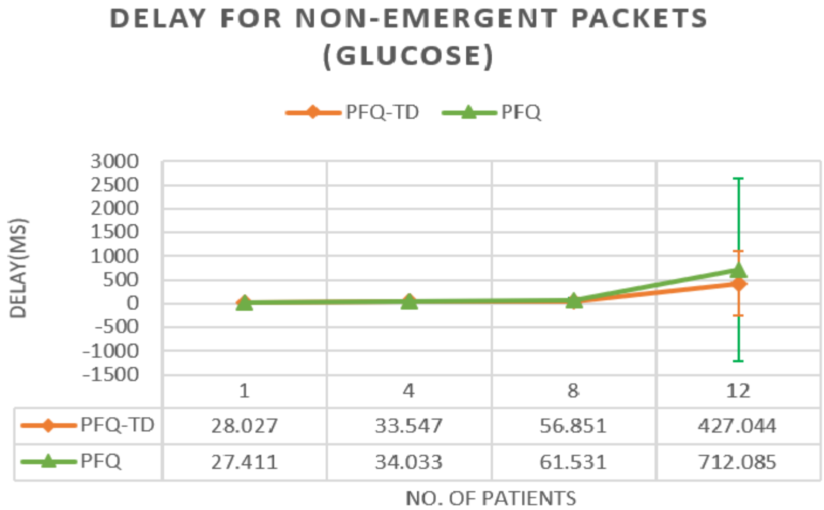


	1	4	8	12
FIFO	27.605	33.299	48.994	248.522
WFQ	27.485	32.908	48.531	142.318
PFQ	27.818	33.212	59.036	444.179

NO. OF PATIENTS

Figure 4.24 Scenario 1: Non-Emg Delay Av. and Standard Deviations





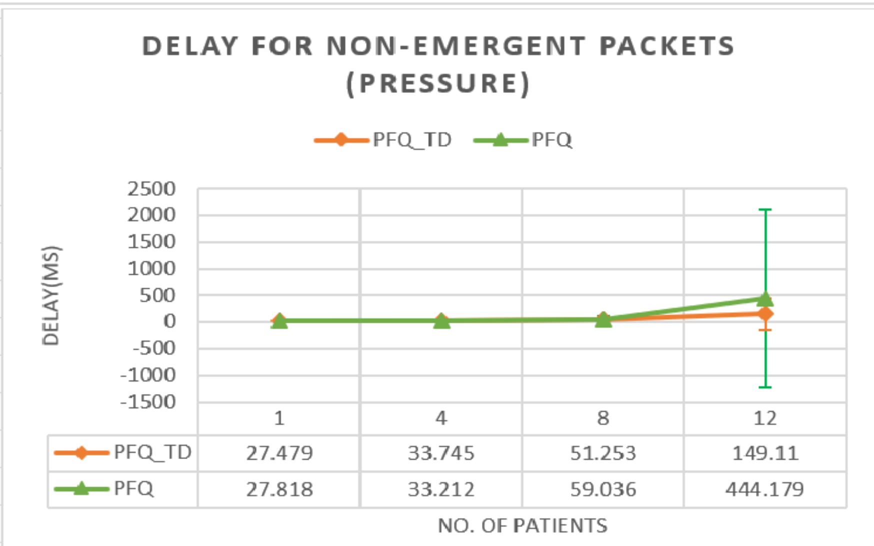


Figure 4.25 Non-Emg Delay Av. and Standard Deviations (PFQ-TD & PFQ)

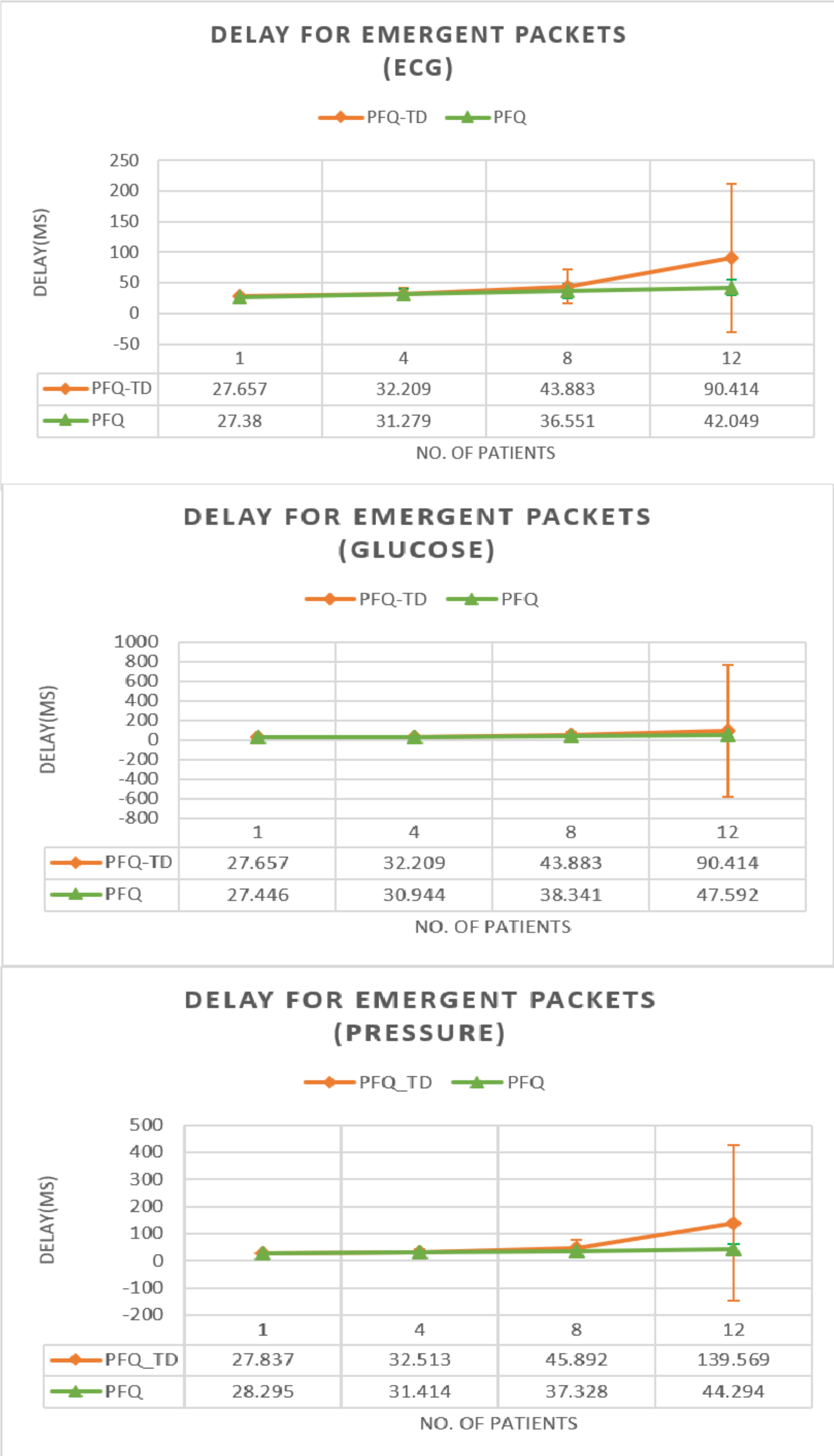


Figure 4.26 **Emg Delay Av. and Standard Deviations of PFQ-TD and PFQ**

4.10 Summary

The results of simulating the new models have been presented in this chapter. Three main scenarios have been used to evaluate the new models' effectiveness to reduce delay and jitter. In the first scenario i.e. number of patients, PFQ has been evaluated against WFQ and FIFO with respect to average delay and average jitter, for both emergency and non-emergency cases. It is proven that PFQ superseded WFQ and FIFO and achieved lower delays and jitter for emergency cases since PFQ gives high priority for emergency cases all the time. However, the PFQ did not address any improvement in the non-emergency packets in delay or jitter at high traffic volume due to the high priority given to the emergency cases all the time.

Therefore, it deemed necessary to address the drawback of the non-emergency cases. Thus, PFQ-TD model was developed and tested under scenario 1 in order to balance the results between emergency and non-emergency cases and was compared to PFQ model results. The results showed that PFQ-TD has improved the non-emergency cases and achieved fair delay distribution between the emergency and non-emergency cases keeping the delay limits within the acceptable industrial limits.

The second scenario i.e. high traffic volume has been performed to study the performance of PFQ model in emergency cases under a high traffic volume. Results have shown that PFQ model has succeeded to keep the average delay and jitter below 65 msec up to 40 patients.

The Third scenario i.e. variant emergency rate has been used to study how PFQ model performs for different rate of emergency cases, with respect to average delay and jitter. Simulation results proved that PFQ succeeded to keep the average delay and jitter for emergency cases quite low at variant emergency rates.

Also; it worth to mention that the “knee” that appears in figure 4.1 to 4.9 graphs which the performance degrade under scenario 1 and 2 (i.e. number or patients and high traffic volume) is due the that Finishing time of PFQ, FIFO and WFQ models are affected by the LDPU capacity at 40 Kbps and the priority parameters given to each packet according to each model.

The capacity of the LDPU was selected to be 40 Kbps since this shall measure the performance of models under relatively low LDPU which is commonly used in the industry and to test the model under stringent low capacity conditions. Using higher LDPU capacity will definitely reduce the finishing time of the models and allow more number or patients or packets to be processed before starvation or the “knee” to happen.

Chapter 5: Conclusion and Future Work

This Chapter concludes the research efforts and the research results in light of the predefined research aims and objectives. It also elaborates on future works that could be performed based on the findings of this research.

A. Achieving the Aims of the Research

The aim of this research is to improve eHealth wireless communication ecosystem in order to deliver high quality eHealth services. The main objective of this research. was studying the at home scenario in eHealth by analyzing literature on communication technologies used in this scenario and how Quality of Service and other attributes enable delivering services considering constraints and requirements of this scenario. Based on this investigation queuing models were considered as a prospective solution to meet the QoS requirements of the at home scenario. These models were studied focusing on existing priority queuing models. This review helped in defining the problem statement and specifying research challenges.

Based on the above, two new priority models namely PFQ and PFQ-TD were developed to reduce and improve the delay and jitter in emergency and non-emergency cases. Simulation test-bed was developed, under which the new models were simulated and tested for performance under pre-set scenarios. Performance evaluation was carried out on the new models in comparison with the existing models currently used.

I. Review Related Literature

Comprehensive review on available literatures, researches and papers was carried out to investigate the current priority queuing technologies in M2M eHealth applications, wireless sensors' types and usage in various applications focusing on those used in eHeath. Also, the review included researching the types, topologies and applications of the wireless sensor networks and the various wireless communication technologies used in transferring data. The literature review also focused on the M2M growth and expansion, how it works, architecture, applications and QoS requirements. Furthermore, the review focused on the most common priority queueing algorithms available in the market to understand their functions, how they work and their advantages and disadvantages. Finally, a review on related work and researches was done to explore related researched and experiments. The results from the overall literature review investigation gave a better understanding of the attributes, developments and current issues of priority queuing models in M2M eHealth applications, limitations, advantages and disadvantages which helped in setting up the directions of this research and path forward.

The limitations found in other studies and current priority base solutions were well defined and concluded that the current priority models and algorithms do not differentiate between real emergency cases and normal cases with respect to readings' value such as heart rate and the patient medical profile such as age, gender, medical history and pregnancy etc. where pregnancy and elderly cases seem more emergent than gender for instance. Similarly, other sensors readings such as glucose and pressure could be prioritized based on the patient's medical history, age and pregnancy. In other words, the priority of each sensor is interdependent on the patient's history and the criticality of the other sensors.

Thus, by adding the Personal Health Record (PHR) i.e. the patient's medical profile as priority criteria for the first time in this research adds a great value to priority queuing algorithms, where priority can be customized in the sensor node as required according to the patients' condition and criticality.

In addition, most of the previous studies had proposed solutions for the delay within the WBAN only i.e. from the sensors to the gateway, but did not deal with the full transmission from the sensors to gateway and then from the gateway to the hospital or the healthcare center. This research has proposed solution for the entire path i.e. starting from the sensor node up to the healthcare center. Thus, a more representative measurement on the priority models was carried out and illustrated.

II. Creating New Knowledge

Based on the above reviews, the goal was set to develop a new priority algorithm that covers and close the gaps in the M2M eHealth ecosystem. Accordingly, the aim of this research was successfully achieved by developing, simulating and evaluating a new innovative priority queuing model under various scenarios simulating the eHealth environment and conditions. The ''Priority-based Fair Queuing'' (PFQ) model ensures a minimum delay and improves jitter while transmitting data from critical sensors in emergency situation from home to the Healthcare centre. PFQ model schedules packets according to their Priority Parameter (PP) value that relies on two factors: data type criticality and Personal Health Record (PHR). PHR concept was introduced for the first time in this research to develop a priority queuing model which integrate not only the critical sensors and data PP values but also the patient's health history and condition. PHR concept has added a great value to the priority queuing model as it categories and prioritizes the data being transmitted differently than other current available models. Thus, giving priority to real emergency cases over normal and non-emergency cases.

III. Testing Contributions

Three scenarios were established to simulate real life situations. Mainly, number of patients, where 12 patients are normally sending signals using M2M to a certain healthcare center 30% of these signals are emergency and 70% are non-emergency. The second scenario focused on the high traffic volume that can occur at any time where 50 patients send signals via M2M. This also can be seen as future projection for the number of patients using M2M with respect to a certain Healthcare center. The last scenario considered variation in emergency signals rates that may increase at any time due to the condition of the patients. These three scenarios cover any situation that might happen in real environment, thus simulating and testing the priority models under these scenarios have covered all real environment possibilities and have disclosed the performance of the models with respect to delay and jitter.

The simulation was built using Python programming Language. We used Python to build our simulation because it is easy and quick to learn. More importantly, Python has SimPy library, which is a discrete-event simulation environment. Thus, we made use of this library to build our simulation. Also, we took the benefit of Grotto's model named SimComponents.py, which has a set of components to create a network simulation. SimComponents.py is basically developed based on SimPy library.

IV. Contribution 1: PFQ Model

The first contribution of this research is developing a "Priority Based-Fair Queuing" PFQ model that reduced the delay and jitter in emergency cases as set in the aims of the research over existing models such as WFQ and FIFO. It actually out performed those two models and gave priority to emergency cases all the time taking into consideration the sensor criticality and the patients' profile while assigning priority to packets. On the other side, delay and jitter were increased at patient number 12 in non-emergency cases in comparison to WFQ and FIFO. That was due to the fact that PFQ model gave high priority to emergency cases all the time.

PFQ delay and jitter results in emergency and non-emergency cases were almost the same as FIFO and WFQ model under "Number of Patients "scenario due to the fact that under low number of patients and with the same simulation parameters setup the three models FIFO, WFQ and PFQ acted almost the same in emergency and non-emergency cases since network capacity is not overloaded. Difference in the models' performance is observed at patient number 12 where network starvation starts occurring. Yet PFQ model outperformed WFQ and FIFO significantly at patient 12 and enhanced the QoS of the eHealth wireless system by first, reducing the delay and improving jitter secondly, outperforming the currently used FIFO and WFQ models in emergency cases and critical data.

PFQ model was progressively simulated under "High Traffic Volume" scenario where the focus was on emergency cases, PFQ succeeded to keep the average delay and jitter values for ECG sensor in emergency cases very low up to 40 patients, however at patient number 50 delay and Jitter increased dramatically due the capacity of the LDPU and start of network starvation. Similarly, the average delay and jitter for emergency cases of Glucose and Pressure sensors increased significantly at 50 patients and scored higher values than ECG due to starvation and the higher data criticality value assigned to the ECG sensor.

Last but not least, Simulating PFQ under "Variant Emergency Rate" Scenario results have indicated that the average delay and jitter for emergency cases for all sensors are quite low because PFQ gives priority to emergency cases. Also, as the emergency signals rate increased, the average delay and jitter in non-emergency case also increased due to the fact that PFQ always forwards non-emergency cases after forwarding the emergency cases. This could be mitigated by increasing the LDPU capacity as required.

V. Contribution 2: PFQ-TD Model

To overcome the increase of the delay and jitter in non-emergency cases observed with the PFQ model, the second contribution of this research come in place where another model has been derived from the PFQ namely “Priority Based-Fair Queuing with Tolerated Delay” PFQ-TD. The PFQ-TD fairly prioritize the emergency and non-emergency packets while considering the acceptable tolerated delays limitations, after which the packets could be useless. The PFQ-TD model has improved the non-emergency data transmission in particular and the overall average delay and jitter in emergency and non-emergency cases in comparison to the PFQ.

PFQ-TD has been developed to give a balanced distribution of bandwidth between emergency cases and non-emergency cases while considering the priority of emergency cases using Tolerated Delay (TD) factor. PFQ-TD had performed in the same fashion giving almost the same results as of PFQ, FIFO and WFQ models under the emergency case “Number of Patients” scenario up to patient 8, then started to exceed PFQ delay readings at patient number 12 in all sensors. Nevertheless, with this increase in delay, PFQ-TD has maintained acceptable tolerated delay of the emergency cases, outperformed FIFO model and close to WFQ model.

PFQ-TD model had overcome the drawback of the PFQ in non-emergency cases and improved the delay and jitter in non-emergency cases by achieving fair delay distribution between the emergency and non-emergency cases keeping the delay limits within the acceptable industrial limits.

VI. Contributions Significance and Value Added

The two new models PFQ and PFQ-TD that were developed in this research have significantly reduced the delay and jitter attributes in emergency and non-emergency situations in comparison with the existing models being used such as WFQ and FIFO thus adding new models in the priority queuing technologies and improving the QoS of the M2M eHealth .

Furthermore, the two new models could be integrated at LDPU to use them alternatively depending on the incoming traffic volume. Hence flexibility is added to the system. The new models can be used as an on-demand model to manage different network workloads. That is, if the network is having low incoming traffic volume while emergency packets are to be transmitted, PFQ queuing model will be used to ensure extra quick transmission of the emergent packets. In case of high incoming traffic volume and while emergency and non-emergency packets to be transmitted, PFQ-TD queueing model will be used to ensure relatively and acceptable quick transmission of emergency packets and provide fair traffic distribution and avoid starvation for non-emergency packets.

Finally, the Patient Health Record that was applied for the first time in this research has added new level of priority criteria in the priority queueing model beside the sensor criticality priority level used in existing models. The Patient Health Record adds a human and patient factor in prioritizing critical and emergency data which could be customized according to the medical needs and history of the patients.

B. Future Work

The newly developed algorithms i.e. PFQ and PFQ-TD has a great potential for further development and enhancement. Further research under different scenarios and environments may lead to improvement in different fields other than eHealth. Future works that could be done beyond the scope of this thesis are listed below:

A. Further research can be done on the PFQ and PFQ-TD models in new scenarios to visualize its effectiveness not only in eHealth ecosystem but also in other wireless application environments.

B. Implementing the new models in healthcare centers that use eHealth ecosystem and measure its effectiveness and improvements in emergency cases and life-threatening situations in real environment.

   As mentioned in this research PFQ model has two parts, PP value and PFQ algorithm. PP value is calculated at the sensor node and PFQ algorithm is implemented and deployed in real network environment at each Local Data Processing Unit (LDPU) component of the Wireless Body Area Networks (WBAN). In WBAN, each sensor monitors one specific medical information, and transmits its signal using ZigBee to the LDPU which acts as a hub/gateway that collects all medical information form sensors and store them temporarily in its buffer (queue) before forwarding them to the healthcare centre. That is, LDPU transmits packets over the Internet to the healthcare centre.

C. Adopting and customizing the PHR concept by adding new parameters to suite special and specific patient's health condition and priority level requirements.

D. Studying other wireless application environment requirements and ecosystems such as disaster relief operation, military applications, environment applications and home applications and modify the PFQ and PFQ-TD in terms of PP value and PHR concept to suite their specific requirements.

E. Expanding the implementation of the new priority models to different fields such as disaster relief operation, military, environment and home applications after customizing the new models and addressing the new requirements of the PP values and PHR concept. Accordingly, sending and receiving of emergency signals within these applications can be improved, eventually the response time to emergency situation shall be improved.

   The results showed that PFQ and PFQ-TD models succeeded to reduce and improve the delay and jitter in emergency and non-emergency cases of eHealth application scenarios. Basically, PFQ and PFQ-TD models schedule packets according to their Priority Parameter (PP) value that relies on two factors: data type criticality and Personal Health Record (PHR). However, these factors cannot be applied directly to other application such as disaster relief operation, military applications, environment applications and home applications. Hence, future activities are suggested to study the applicability of PFQ and PFQ-TD models to different applications, as well as to investigate the requirement of these applications to modify the PFQ and PFQ-TD in terms of PP value and PHR concept to suite their specific requirements.

F. Integrating PFQ and PFQ-TD algorithms at LDPU to use them alternatively based on the incoming traffic volume. Then, simulating different scenarios to study the implications of this integration on emergency and nonemergency cases, in terms of average delay and jitter.

   PFQ algorithm always provides a high priority to emergency cases over non-emergency cases. Therefore, the results showed that PFQ algorithm succeeded to reduce the average delay and average jitter of emergency cases, but it increased the average delay and average jitter of non-emergency cases. Thus, PFQ-TD was introduced to make a fair balance between them while still providing a priority for emergency cases using TD factor.

In this section, I investigate the difficulties and approaches of combining both algorithms (PFQ and PFQ-TD) in a single algorithm and study its effect on the average delay and jitter for emergency and non-emergency cases.

Figure 5.1 shows a prospective workflow of combining PFQ and PFQ-TD models at LDPU.

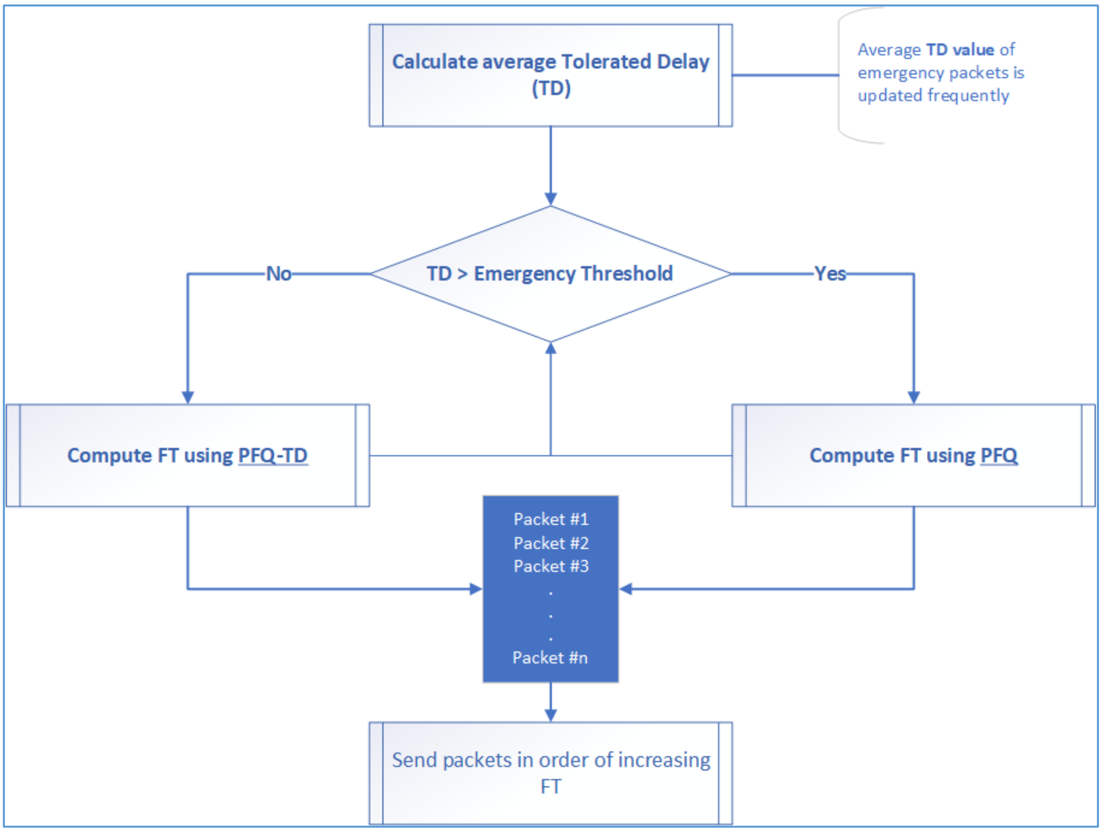


Figure 5.1: A Prospective Workflow of Combining PFQ and PFQ-TD Models

At first, the average TD value of emergency cases is calculated. TD refers to the maximum delay that is acceptable for a packet to reach its destination. Authors in [2] assign 50ms as a threshold of the TD for emergency packets and

150ms for nonemergency packets. So, if the average TD of emergency cases exceeded the thresholds (e.g., 50ms), the LDPU uses PFQ algorithm to maintain the average delay of emergency packets below the threshold. Otherwise, the LDPU uses PFQ-TD to have fair distribution with balanced delay between emergency and non-emergency cases. Thus, this way of combining PFQ and PFQ-TD is most likely to make the average delay of emergency cases below the threshold (i.e., TD) and reduce the average delay of non-emergency cases.

There are some difficulties of Combining PFQ and PFQ-TD. The mechanism of switching between PFQ and PFQ-TD should be automatic and work in a timely manner to achieve the best performance in terms of average delay and jitter of emergency and non-emergency cases. A deep study is needed to identify the parameters and procedures of the switching process. Another point of most importance is the frequent tracking of the average delay for emergency packets in real network environment. As we know that LDPU forwards packets over the Internet and this may add burdens to the tracking process.

G. investigation should be devoted to devise a solution to the problem of information overload whereby during an emergency the LDPU sends frequent request to the information sink to obtain priority parameters thus occupying precious bandwidth

PFQ model recorded a significant increase of the average delay of the non-emergency cases for the three sensors. This is attributed to giving priority to emergency cases over non-emergency cases. As a consequence, LDPU overwhelms the information sink with emergency cases and causing information overload problem due to the queued non-emergency cases. One solution of this problem is to balance between emergency and non-emergency cases to some extent. That is, allow for some non-emergency cases instead of assigning high priority for emergency cases all time. This solution is employed by the PFQ-TD whereby it gives a balance between emergency and non-emergency cases while considering the priority of emergency cases using Tolerated Delay (TD) factor. Hence, TD factor gives priority to emergency cases to some extent.

Another solution to this problem is to increase the capacity rate of the LDPU. It is clear from the results that the average delay of the three sensors with 1, 4, and 8 patients is almost the same for, FIFO, WFQ, and PFQ. This is because at this number of patients there is no starvation at LDPU. A significant increase is noticed with 12 patients due to the resource starvation. Therefore, boosting the data rate of the LDPU is supposed to limit the information overload problem.

References

[1] Elprocus (2017) Wireless Sensor Networks and their Applications. Available at: https://www.elprocus.com/introduction-to-wireless-sensor-networks-types-and-applications/ (Accessed: January 2018).

[2] Shinde, M., Gupta, J. and Gawade, S (2014) Overview of Different Types of Sensors used in EHealth Environment. DOI: V213P03.

[3] Microcontrollers Lab (2018) Wireless Sensor Networks & Applications. Available at: http://microcontrollerslab.com/wireless-sensor-networks-wsn-applications/ (Accessed: January 2018).

[4] Elprocus (2017) Different Types of Wireless Communication with Application. Available at: https://www.elprocus.com/types-of-wireless-communication-applications/ (Accessed: January 2018).

[5] Campus-Mur, D., Garcia-Saavedra, A and Serrano, P (2013) Device-to-Device Communications with Wi-Fi Direct: Overview and Experimentation. IEEE Wireless Communications Vol. 20. DOI: 10.1109/MWC.2013.6549288.

[6] Hyeck Lee, J., Park, M. and Shah, S. (2017) Wi-Fi Direct Based Mobile Ad Hoc Network. International Conference on Computer and Communication Systems (ICCCS). DOI: 10.1109/CCOMS.2017.8075279.

[7] Ahmadi, S (2009). An Overview of Next-Generation Mobile WiMAX Technology. IEEE Communications Magazine Vol.47 Issue 6, Pages 84-98. DOI: 10.1109/MCOM.2009.5116805.

[8] Seyedzadegan, S. and Othman, M. (2013) WiMAX Overview, WiMAX Architecture. International Journal of Computer Theory and Engineering, Vol. 5, No. 5. DOI: 10.7763/IJCTE.2013.V5.796.

[9] Grattan, K. and Sun, T. (2000) Fibre Optic Sensor Technology: An Overview. Sensors and Actuators A: Physical, Volume 82, Issues 1–3, Pages 40-61. DOI: 10.1016/S0924-4247(99)00368-4.

[10] Hisham, H. (2018). Optical Fibre Sensing Technology: Basics, Classifications and Applications. American Journal of Remote Sensing. 6. DOI:10.11648/j.ajrs.20180601.11.

[11] Massa, N. (2000). Fibre Optic Communication. Fundamentals of Photonics, Springfield Technical Community College. Available at: https://spie.org/Documents/Publications/00%20STEP%20Module%2008.pdf. (Accessed: February 2018).

[12] Dahlman, E., Parkvall, S. and Skold, J. (2014). 4G LTE/LTE-Advanced for Mobile Broadband. 2nd Edition, Academic Press USA. ISBN: 978-0-12-419985-9.

[13] Salkintzis, A. (2004). "Interworking techniques and architectures for WLAN/3G integration toward 4G mobile data networks." IEEE Wireless Communications, vol. 11, no. 3, pp. 50-61. DOI: 10.1109/MWC.2004.1308950

[14] ShivamKedia, S. (2013). *Wi-Fi Direct*. [online] Slideshare.net. Available at: http://www.slideshare.net/shivam_kedia/wifi-direct-26864237 [Accessed 7 May 2015].

[15] AL-Taee, M., Jaradat, N. and Ali, D. (2011). Mobile phone-based health data acquisition system using Bluetooth technology. *IEEE*. [online] Available at: http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p [Accessed 7 May 2015].

[16] IEEE WCNC (2009). Newest Advances in Wireless Communications *IEEE Communications Magazine*, 47(6), pp.186-186. [Online]. DOI: 10.1109/MCOM.2009.5116818. (Accessed: December 2017).

[17] Cha, I., Shah, Y., Schmidt, A.U., Leicher, A and Meyerstein, M.V. (2009) "Trust in M2M communication," in *IEEE Vehicular Technology Magazine*, vol. 4, no. 3, pp. 69-75.

DOI: 10.1109/MVT.2009.933478. (Accessed: February 2018).

[18] Schnider, E (2003). “The Future of M2M and the IOT- Vision for 2020” M2M Journal, ISSN 1868-9558.

[19] Antonia C. (2013) M2M Development (System Architecture) Duniaelectronics. Available at: https://duniaelectronic.wordpress.com/m2m/4-m2m-development-system-architecture/. (Accessed: January 2018).

[20] Whitehead, S (2004). "Adopting Wireless Machine-to-Machine Technology," IEE Computing and Control Eng. J, vol. 15, no. 5.

[21] Schnider, E (2013) "The Future of M2M and the IOT- Vision for 2020" M2M Journal, ISSN 1868-9558.

[22] Bersight (2014). Consumer Electronics and Wireless M2M. Bersight. Available at: http://www.berginsight.com/reportpdf/productsheet/bi-ce3-ps.pdf (Accessed: December 2016).

[23] Novotech (2014). Huge M2M Predictions from SAP: Are they Realistic? [Online]. Available at: http://www.novotech.com/blog/huge-m2m-predictions-sap-are-they-realistic. (Accessed: January 2017.

[24] Zhang, Y., Yu, R., Xie, S., Yao, W., Xiao, Y and Guizani, M (2011) Home M2M networks: Architectures, standards, and QoS improvement. IEEE Communications Magazine, vol. 49, no. 4, pp. 44-52. DOI: 10.1109/MCOM.2011.5741145

[25] HanGyu Cho (2010) Machine to Machine (M2M) Communications Technical Report. IEEE Xplore, 802.16p-10/0005, vol. 4, no. 3. Available at: www.ieee.org

[26] Tan, S.K., Sooriyabandara, M and Fan, Z (2011). M2M Communications in the Smart Grid: Applications, Standards, Enabling Technologies, and Research Challenges. *International Journal of Digital Multimedia Broadcasting.* DOI: org/10.1155/2011/289015.

[27] Katusic, D., Weber, M., Bojic, I., Jezic, G. and Kusek, M. (2012). Market, standardization, and regulation development in Machine-to-Machine communications. *IEEE*, [online] pp.1 to 7. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6347649

[28] Riker, A., Cruz, T., Marques, B., Curado, M., Simoes, P and Monteiro, E (2014). Efficient and secure M2M communications for smart metering. *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA).* DOI: 10.1109/ETFA.2014.7005176

[29] Gemalto (2014). Security in M2M Communication - What is secure enough? Available at: http://www.m2malliance.com/fileadmin/user_upload/pdf/2013/Whitepaper/gemalto_whitepaper_secure_M2M_communication_web.pdf. (Accessed: February 2017).

[30] Techweek (2015). “How F1 Technology Is Helping McLaren Make the World A Smarter Place”. Available at: http://www.techweekeurope.co.uk/networks/mclaren-applied-technologies-m2m-health-171082 (Accessed: March, 2017).

[31] Wang, Z. and Xu, X (2013). Smart Home M2M Networks Architecture. *IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks.* pp. 294-299. DOI: 10.1109/MSN.2013.24

[32] Qiuwen, C and Yuting, L (2010). M2M Technology Development Prospects. IEEE. Available at: https://www.2cm.com.tw/technologyshow_view.asp?sn=1011230014 (Accessed: March, 2018).

[33] Yan Zhang et al., “Home M2M Networks: Architectures, Standards, and QoS Improvement” IEEE Communications Magazine April 2011.

[34] Wahle, S., Magedanz, T and Shulze, F (2014). Demonstration of Open MTC – M2M Solutions for Smart Cities and the Internet of Things. Available at: https://www.ieeelcn.org/prior/LCN37/lcn37demos/LCNDemos12_Wahle.pdf. (Accessed: March, 2017).

[35] Abbas, R., Shirvanimoghaddam, M., Li, Y and Vucetic, B (2017). Random Access for M2M Communications with QoS Guarantees. *IEEE Transactions on Communications,* vol. 65, no. 7, pp. 2889-2903. DOI: 10.1109/TCOMM.2017.2690900

[36] Liu, R., Wu, W., Zhu, H and Yang, D (2011). M2M-Oriented QoS Categorization in Cellular Network, *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing. DOI:* 10.1109/wicom.2011.6040143

[37] Cizmar, A and Papaj, J (2009). Theory of Telecommunications Networks, *Department of Electronics and Multimedia and Telecommunications.* Available at: http://kemt.fei.tuke.sk/wp-content/uploads/TTS/TTS_chapter1.pdf . (Accessed: March 2017).

[38] Mesbahi, N and Dahmouni, H (2016). Delay and jitter analysis in LTE networks, International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 122-126. DOI: 10.1109/WINCOM.2016.7777202.

[39] Klepec, B and Kos, A (2011). Performance of VoIP Applications in a Simple Differentiated Services Network Architecture, International Conference on Trends in Communications, pp. 214-217 vol.1. DOI: 10.1109/EURCON.2001.937798.

[40] Cazzaniga, G and Ronchetti, L (2012). Embedded tools for network delay measurement, International Telecommunications Network Strategy and Planning Symposium (NETWORKS), pp 1-6. DOI: 10.1109/NETWKS.2012.6381682.

[41] Jereb, L. (2011). “Network reliability: Models, Measures and analysis” Department of telecommunication University of Budapest

[42] Dordal, P (2018). “An Introduction to Computer Networks” Department of Computer Science, Loyola University Chicago.

[43] Allison, L (2014). “Queues” [Online]. Available at: http://www.allisons.org/ll/AlgDS/Queue/ (Accessed: March 2017).

[44] Mustafa, M. and Talab, S. (2016) The Effect of Queuing Mechanisms First in First out (FIFO), Priority Queuing (PQ) and Weighted Fair Queuing (WFQ) on Network’s Routers and Applications. Wireless Sensor Network, 8, 77-84. DOI: 10.4236/wsn.2016.85008.

[45] Kallmes, M.H., Towsley, D and Cassandras, C.G. (1989). Optimality of the last-in-first-out (LIFO) service discipline in queuing systems with real-time constraints, *IEEE Conference on Decision and Control, pp. 1073-1074 vol.2. DOI:* 10.1109/CDC.1989.70296.

[46] Alqhatani, M (2010). "Stacks and Queues". [Online]. Available at: www.kau.edu.sa/GetFile.aspx?id=152091&fn=Stacks_Queues_imp_smr.pptx (Accessed: March 2017).

[47] Mittal, N., Garg, K and America, A. (2015). A Paper on Modified Round Robin Algorithm, JECRCUDML College of Engineering Rajasthan, Vol. 4. ISSN 2278-2540. Available at: https://www.ijltemas.in/DigitalLibrary/Vol.4Issue11/93-98.pdf .

[48] Verma, M., Kumar, E.K and Monga, H (2017). Comparative Analysis of Job Scheduling Algorithms: A Review, IJEDR Vol. 5. ISSN 2321-9939. Available at: https://www.ijedr.org/papers/IJEDR1703179.pdf (Accessed: April 2018).

[49] Singh, A., Goyal, P and Batra, S (2010). An Optimized Round Robin Scheduling Algorithm for CPU Scheduling, *International Journal on Computer Science and Engineering Vol.2*. Available at: https://pdfs.semanticscholar.org/780d/07fbf09484a2490868d45255a23762a7e21d.pdf (Accessed: April 2018).

[50] Crupi, V.A., Das, S.K and Pinotti, M.C (1996). "Parallel and Distributed Meldable Priority Queues Based on Binomial Heaps," Proceedings of the 1996 ICPP Workshop on Challenges for Parallel Processing, Ithaca, NY, USA, pp. 255-262 vol.1. DOI: 10.1109/ICPP.1996.537167.

[51] Wang, H and Lin, B (2007). Pipelined van Emde Boas Tree: Algorithms, Analysis, and Applications. INFOCOM IEEE. DOI: 10.1109/INFOCOM.2007.303. Available at: https://www.semanticscholar.org/paper/Pipelined-van-Emde-Boas-Tree%3A-Algorithms%2C-Analysis%2C-Wang-Lin/33fc4e8dfc961c6e1723b229db01ade97c952974

[52] Ke, J.F., Chen, W.J and Huang, D.C (2015). Life Extend Approach Based on Priority Queue N Strategy for Wireless Sensor Network, *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE)*, Taipei, 2015, pp. 272-279. Electronic ISBN: 978-1-6319-0063-1.

[53] Kim, R.H.K., Kim, P.S and Kim, J.G (2015). An Effect of Delay Reduced MAC Protocol for WBAN Based Medical Signal Monitoring, *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp. 434-437. DOI: 10.1109/PACRIM.2015.7334876.

[54] Rashwand, S and Misic, J (2014). Impact of Priority Differentiation on the Bridged WBAN/WLAN Healthcare Networks, Wireless Communications & Mobile Computing Volume 14 Issue 12, Pages 1189-1200. DOI: 10.1002/wcm.2264.

[55] Javaid, N., Yaqoob, M., Khan, M.Y., Khan, M.A. and Khan, Z.A (2013).Analysing Delay in Wireless Multi-hop Heterogeneous Body Area Networks, Research Journal of Applied Sciences, Engineering and Technology. Available at: https://arxiv.org/abs/1304.1059.

[56] Lu, Y., Zhang, Z and Zhang, T (2014). UPQ: An effective packet scheduling algorithm for delay-sensitive service in NGEO satellite network, International Conference on Communications and Networking pp. 517-522. DOI: 10.1109/CHINACOM.2014.7054350.

[57] Barua, M., Alam, M.S., Liang, X and Shen, X (2011). Secure and quality of service assurance scheduling scheme for WBAN with application to EHealth , *IEEE Wireless Communications and Networking Conference, Cancun, Quintana pp. 1102-1106*. DOI: 10.1109/WCNC.2011.5779285.

[58] Zhi-Sheng, K., Xiao-Yuan, M., Zheng-Yi, X., Bo, Z and Jiang-Ming W (2014). Real-Time Scheduling Algorithm Based on Dynamic Priority for Multi-data Transmission, *International Journal of Computer Science Issues, Vol. 11, Issue 2, No 2.* ISSN (Online): 1694-0784.

[59] L. Salehi and P. Mirhadi. EHealth Applications Implementation Considerations. Iran University of Science and Technology/Department of electrical Engineering. IFMBE Proceedings book series (IFMBE, volume 21)

[60] Wolfgang Damm, Jitter – An Introduction. Wireless Telecom Group White paper

[61] Chuck Semeria 'Supporting Differentiated Service Class: Queue Scheduling Disciplines' Juniper Networks Inc, 2001

[62] Behrouz A. Forouzan and Sophia Chung Fegen, Data Communication and Networking, 4th ed ISBN-13978-0-07-296775-3 McGraw-Hill Forouzan Networking series, 2007

[63] Jay Pil Choi. Doh-Shin Jeon and Byung-Cheol Kim 'Net Neutrality, Network Capacity, and Innovation at the Edges' July 18, 2017_