PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

Ministry of Higher Education and Scientific Research

Mohamed Khider University - BISKRA

Faculty of Exact Sciences, Natural Sciences and Life

Computer Science department

Order N°: RTIC/M2/2020

Memory Thesis

presented to obtain the academic master's degree in

Computer Science

Option: Networks and Technologies of Information and Telecommunications

Privacy preservation for publish/subscribe communications in the context of IoT-connected e-healthcare

By: **ZIRARA Imane**

Defended on September 27, 2020, in front of the jury composed of:

	MCA	President/ chairwoman
Sahraoui Somia	MCB	Reporter
	MCB	Examiner

2019/2020

Acknowledgements

My thanks go first to the almighty ALLAH, for providing me with courage and patience especially, in corona virus quarantine, and for guiding my steps towards the path of knowledge.

I would like to extend my sincere thanks to my supervisor Dr. SAHRAOUI Somia, for supporting me throughout the development of my work, for her seriousness in accomplishing her mission, effectiveness of her directives, as well as her wise advices, which got me out of several dead ends. So, I want to thank her from the bottom of my heart. Additionally, I would apologize for any disappointment I've caused.

I am grateful to the committee members of my jury for their valuable inputs and time spent reading this thesis.

I want to express my special gratitude to my sweet small family mentioning my dear father NACEREDDINE, my kind mother BASLI HAYAT and my only fun-loving sister and my twin in same time AMINA for their unconditional support, love and trust throughout my studies and did not hesitate to deprive themselves to provide the necessary means to achieve my education.

Finally, last but not least, warm thanks to my close friends and colleagues for all the unforgettable enjoyable moments we spent together and their helps that allowed me to hang on to the end and never let go.

Without forgetting to thank anyone who has participated directly or indirectly in the achievement of this work.

Abstract

We are living in a high-rate development era where Internet of Things plays the role of technical tool by empowering physical resources into smart entities through mixture of various communications and embedded technologies in its architecture. Message Queuing Telemetry Transport (MQTT) is a lightweight publish/subscribe-based messaging protocol that is considered from the widely used for data delivery in IoT applications. MQTT-SN is designed to be as close as possible to MQTT with peculiarities of a wireless communication. However, it lacks from security and privacy features that remains one of the most important issues that baffle the development of e-healthcare application because of its sensitive data that it is tightly linked to user's privacy. Our work is to guarantee good levels of privacy protection for the publisher (patient privacy) via MQTT-SN by hiding his ID and registered topics. The proposed solution is simple, lightweight, confidential and was tested in Cooja simulator.

Key words: IoT, e-healthcare, MQTT, MQTT-SN, privacy, patient' privacy, hidden ID, anonymity, topic encryption, Lightweight, confidentiality, Cooja, simulation

ملخص

نحن نعيش في عصر تطور عالي المستوى حيث إنترنت الأشياء تلعب دور الأداة التقنية من خلال تمكين الموارد المادية في الكيانات الذكية عبر مزيج من الاتصالات المختلفة والتقنيات المضمنة في هندستها المعمارية. يعد النقل عن بُعد لقائمة انتظار الرسائل (MQTT) بروتوكول رسائل خفيف الوزن يعتمد على النشر/ الاشتراك ويُعتبر من البروتوكولات المستخدمة على نطاق واسع لتسليم البيانات في تطبيقات إنترنت الأشياء. تم تصميم **MQTT-SN** ليكون أقرب ما يمكن إلى **MQTT** بخصائص الاتصال اللاسلكي. ومع ذلك، فإنه يفتقر إلى ميزات الأمان والخصوصية التي تظل واحدة من أهم المشكلات التي تعرقل تطوير تطبيقات الرعاية الصحية الإلكترونية بسبب بياناتها الحساسة التي ترتبط ارتباطًا وثيقًا بخصوصية المستخدم. عملنا هو ضمان مستويات جيدة من حماية الخصوصية للناشر (خصوصية المريض) عبر **MQTT-SN** عن طريق إخفاء هويته والموضوعات المسجلة. الحل المقترح بسيط، خفيف الوزن، سري وتم اختباره في جهاز محاكاة **Cooja**.

كلمات البحث: إنترنت الأشياء، الرعاية الصحية الإلكترونية، **MQTT**، **MQTT-SN**، الخصوصية، خصوصية المريض، المعرف المخفي، إخفاء الهوية، تشفير الموضوع، خفيف الوزن، السرية، **Cooja**، المحاكاة.

Acronyms

6LoWPAN	IPv6 over Low -Power Wireless Personal Area Networks
AES-CTR	Advanced Encryption Standard -The Counter Mode
AMQP	Advanced Message Queuing Protocol
BG	Blood Glucose
BLE	Bluetooth Low Energy
BP	Blood Pressure
CoAP	Constrained Application Protocol
Cooja	Contiki OS java simulator
CP-ABE	Ciphertext Policy Attribute Based Encryption
CSRNG	Cryptographically Secure Random Number Generator
ECC	Elliptic Curve Cryptography
ECG	Electrocardiogram
E-Healthcare	Electronic Healthcare
EHR/EMR	Electronic Health Record/ Electronic Medical Record
EMG	Electromyography
EPR	Electronic Patient Records
GPRS	General Packet Radio Service
GPS	Global Positioning System
GW	Gateway
HF	High Frequency
HTTP	Hyper Text Transfer Protocol
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
ISM	Industrial, Scientific and Medical (radio spectrum)
ISO	International Organization for Standardization
IT	Information Technology
KP-ABE	Key Policy Attribute Based Encryption
LAN	Local Area Network
LoRa	Long Range
LPWANs	Low Power Wide Area Networks
LSB	Least Significant Byte
LTE	Long-Term Evolution
LWEA	Light-Weight Encryption Algorithm
M2M	Mobile to Mobile
MAC	Media Access Control
MQTT	Message Queuing Telemetry Transport
MQTT-SN	Message Queuing Telemetry Transport for Sensor Network
MSB	Most Significant Byte (multiple byte sequences)
NFC	Near Field Communication

OASIS	Organization for the Advancement of Structured Information Standards
OpenHAB	Open Home Automation Bus
PAN	personal area network
PCG	Permuted Congruential Generator
PHI	Personal Health Information
PKG	Private Key Generator
PPCP	Privacy-Preserving Comparison Protocol
PRC	Privacy Rights Clearinghouse
PSS	Pre-Shared Secret
Pub/Sub	Publish / Subscribe
QoS	Quality of Service
RFID	Radio Frequency Identification
RHM	Remote Health Monitoring
RPi	Raspberry Pi
RSMB	Really Small Message Broker
RSN	RFID sensor network
SBC	Single-Board Computer
SMS	Short Messaging Service
SoA	Service-oriented Architectures
TCP	Transmission Control Protocol
TS	Trusted System
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UNB	Ultra-Narrow Band
WBAN	Wireless Body Area Network
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WSN	wireless sensor networks
WWW	World Wide Web
XML	Extensible Markup Language
XOR	Exclusive Or

Table of contents

List of figures

List of tables

General Introduction

"It's about networks, it's about devices, and it's about data," Caroline Gorski, the head of IoT at Digital Catapult explains. IoT allows devices on closed private internet connections to communicate with others and "the Internet of Things brings those networks together. It gives the opportunity for devices to communicate not only within close silos but across different networking types and creates a much more connected world."

Among various messaging protocols for IoT, MQTT is widely used protocol based of publish/subscribe pattern that is decoupled communication paradigm that allows data broadcasting without any link between the sender (publisher) and the receiver (subscriber) through a server/broker. Despite the benefits of the system, it also raises security and privacy issues.

With the rapid growth of Internet technologies as well as the explosion of connected objects, Internet of Things (IoT) is considered an Internet revolution that positively affects several life aspects. Its effectiveness has been proven in many fields like the medical field as often used in telemedicine or the Wireless Area Network since e-healthcare is the most requiring IoT application in term of security and privacy preservation. This is mainly due to the fact that healthcare data are often sensitive.

This work, which aims to provide a patient privacy through MQTT-SN communication, is organized into five chapters:

1. The 1st chapter introduces generalities on internet of things, including their characteristics and their context.
2. The 2nd chapter studies firstly, the publish/subscribe communication and more specifically MQTT protocol. Then, we devote to the topic e-healthcare. Besides, some scenarios of MQTT in that field.
3. In the 3rd chapter, we target the issue of privacy by describing a state of the art of the solutions developed for e-healthcare and on MQTT protocol before comparing them.
4. The 4th chapter starts with the explanation of our problematic. Then, we describe our contribution where we define the principle functions of the proposed feature via MQTT-SN, as well as the main steps of the implementation.
5. The 5th chapter summarizes the results obtained after simulation.

Chapter one

Generalities on Internet of Things

1.1 Introduction

The Internet of Things (IoT) is arguably one of the most exciting topics in the research community. While the traditional internet is one of the most important and powerful creations in human history by facilitating communication between a number of limited devices and humans, the IoT connects all kinds of “things” without human intervention.

In another words, represents the next evolution of the Internet, taking a huge leap in its ability to gather, analyze, and distribute data that we can turn into information, knowledge, and, ultimately, wisdom.[1]

But also faced with equally strong barriers in terms of security risks that threaten to slow it down. However, as standards bodies and universities work together to solve these challenges, the IoT will continue to progress.

In this chapter, we will explore the important concepts of IoT and other details such as its characteristics, application domains and most challenges.

1.2 IoT History

The emergence of the Internet of Things is only a result of the convergence of multiple technologies, namely the Internet, wireless communication, embedded systems, microelectronics and nanotechnology. In this section, we list the most significant events that made the IoT a reality.

The concept of a smart device network was first brought up in 1982, with the first internet-connected device at Carnegie Melon University (a Coca-Cola dispenser connected to Internet) capable of signaling to its inventory whether newly loaded drinks are indeed cold. This experience has inspired many inventors around the world to create their own connected devices. Thereby, in 1991, Mark Weiser introduced ubiquitous computing through his paper titled "The Computer for the 21st Century" and he then presented the contemporary vision of the Internet of Things. Shortly after, in 1994, Steve Mann had created the WearCam which was among the first cameras to appear on the web. WearCam consists of the following parts: (1) a group of cameras (or only one) which are attached to the body in some way with two hands free (2) means for recording, processing and transmission of images captured by the cameras (3) a display means which has the capability of presenting an image or a stream of images from the camera. The captured images will be communicated through an entity (a base station) available to the user. Then, in 1998, ubiquitous computing began to gain attention by allowing the flexible incorporation of computing into daily life. Mark Weiser once said: "where virtual reality puts humans inside the world of computers, ubiquitous computing instead forces the computer to take root in the real world."

In 1999, the designation Internet of Things was first uttered by Kevin Ashton. Then, in 2000, the LG company announced its first smart refrigerator connected to the Internet. In addition, RFID (Radio Frequency Identification) technology, which is one of the constitutional technologies of the IoT, began to be massively deployed around 2003 and 2004. On the other hand, a very interesting initiative was taken in 2008; a research group called the IPSo Alliance has dedicated itself to promote the use of Internet Protocol (IP) in networks of "smart objects"

Much research work has followed and has all focused on achieving, under the best conditions, the vision of the Internet of Things and bringing it to fruition despite all the challenges raised. This with the consideration of the continuous technological advancements in the market of intelligent devices and in the field of telecommunications technologies.[2]

1.3 IoT Definition

The Internet of Things (IoT) is the network of physical objects accessed through the internet, as defined by technology analysts and visionaries. These objects contain embedded technologies such as wireless sensor networks (WSN) and Radio frequency identification (RFID) to interact with internal state or external environment. [3]

In general, we can say IoT allows people and things to be connected Anytime, Anyplace, with anything and anyone using any network and any service as shown in Figure 1.1 [4]

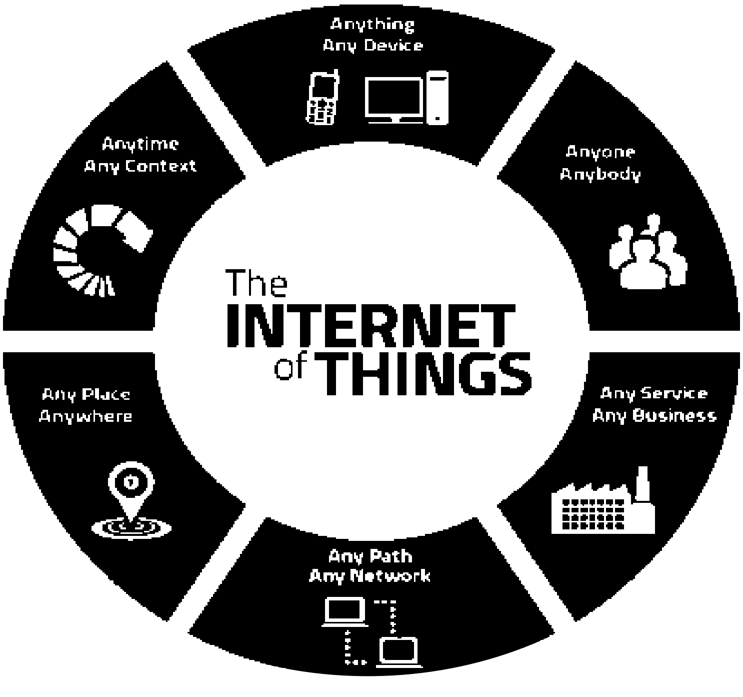


Figure 1.1 Internet of Things

1.4 IoT Characteristics

The IoT is a complex system with a number of characteristics, it varies from one domain to another. Some of the characteristics are as follows:[5]

1.4.1 Intelligence

IoT comes with the combination of algorithms and computation, software & hardware that makes it smart. Ambient intelligence in IoT enhances its capabilities which facilitate the things to respond in an intelligent way to a particular situation and supports them in carrying out specific tasks. In spite of all the popularity of smart technologies, intelligence in IoT is only concerned as means of interaction between devices, while user and device interaction are achieved by standard input methods and graphical user interface.

1.4.2 Connectivity

Connectivity empowers Internet of Things by bringing together everyday objects. Connectivity of these objects is pivotal because simple object level interactions contribute towards collective intelligence in IoT network. It enables network accessibility and compatibility in the things. With this connectivity, new market opportunities for Internet of things can be created by the networking of smart things and applications.

1.4.3 Dynamic Nature

The primary activity of Internet of Things is to collect data from its environment, this is achieved with the dynamic changes that take place around the devices. The state of these devices changes dynamically, example sleeping and waking up, connected and/or disconnected as well as the context of devices including temperature, location and speed. In addition to the state of the device, the number of devices also changes dynamically with a person, place and time.

1.4.4 Enormous scale

The number of devices that need to be managed and that communicate with each other will be much larger than the devices connected to the current Internet. The management of data generated from these devices for application purposes becomes more critical. The enormous scale of IoT in the estimated report where it stated that 5.5 million new things will get connected every day and 6.4 billion connected things will be in use worldwide in 2016 The report also forecasts that the number of connected devices will reach 20.8 billion by 2020.

1.4.5 Sensing

IoT wouldn't be possible without sensors which will detect or measure any changes in the environment to generate data that can report on their status or even interact with the environment. Sensing technologies provide the means to create capabilities that reflect a true awareness of the physical world and the people in it.

1.4.6 Heterogeneity

Heterogeneity in Internet of Things as one of the key characteristics. Devices in IoT are based on different hardware platforms and networks and can interact with other devices or service platforms through different networks. IoT architecture should support direct network connectivity between heterogeneous networks. The key design requirements for heterogeneous things and their environments in IoT are scalabilities, modularity, extensibility and interoperability.

1.4.7 Security

IoT devices are naturally vulnerable to security threats. As we gain efficiencies, novel experiences, and other benefits from the IoT, it would be a mistake to forget about security concerns associated with it. There is a high level of transparency and privacy issues with IoT. It is important to secure the endpoints, the networks, and the data that is transferred across all of it means creating a security paradigm.

1.5 IoT architecture

Typically, this architecture consists of three layers: Perception Layer, Network Layer, and Application layer (Figure 1.2). A brief description of each layer is given: [6]

1.5.1 Perception layer

also known as the sensor layer, is implemented as the bottom layer in IoT architecture. The perception layer interacts with physical devices and components through smart devices (RFID, sensors, actuators, etc.).Its main objectives are to connect things into IoT network, and to measure, collect, and process the state information associated with these things via deployed smart devices, transmitting the processed information into upper layer via layer interfaces.

1.5.2 Network layer

also known as the transmission layer, is implemented as the middle layer in IoT architecture. Network layer is used to receive the processed information provided by perception layer and determine the routes to transmit the data and information to the IoT hub, devices, and applications via integrated networks. The network layer is the most important layer in IoT architecture, because various devices (hub, switching, gateway, cloud computing perform, etc.), and various communication technologies (Bluetooth, Wi-Fi, Long-Term Evolution (LTE), etc.) are integrated in this layer. The network layer should transmit data to or from different things or applications, through interfaces or gateways among heterogeneous networks, and using various communication technologies and protocols.

1.5.3 Application layer

also known as the business layer, is implemented as the top layer in IoT architecture. The application layer receives the data transmitted from network layer and uses the data to provide required services or operations. For instance, the application layer can provide the storage service to backup received data into a database, or provide the analysis service to evaluate the received data for predicting the future state of physical devices. A number of applications exist in this layer, each having different requirements. Examples include smart grid, smart transportation, smart cities.

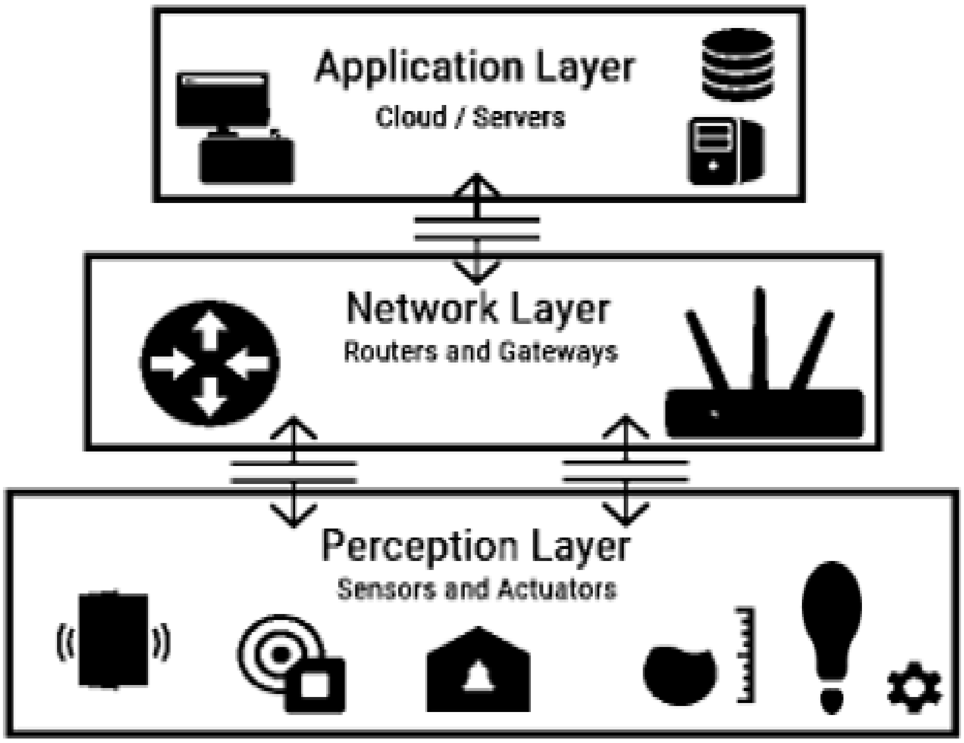


Figure 1.2 Three-Layer IoT architecture [7]

The three-layer architecture is basic for IoT and has been designed and realized in a number of systems. Yet, despite the simplicity of the multi-layer architecture of IoT, functions and operations in the network and application layers are diverse and complex. For example, the network layer not only needs to determine routes and transmit data, but also provide data services (data aggregation, computing, etc.). The application layer not only needs to provide services to customers and devices, it must also provide data services (data mining, data analytics, etc.). Thus, to establish a generic and flexible multi-layer architecture for IoT, a service layer should be developed between network layer and application layer to provide the data services in IoT. Based on this concept, Service-oriented Architectures (SoA) have recently been developed to support IoT.

1.6 Enabling technologies

IoT can be realized with several enabling technologies:[6]

1.6.1 RFID

Generally speaking, RFID, as a non-contact communication technology, is used to identify and track objects without contact. It supports data exchange via radio signals over a short distance. The RFID-based system consists of RFID tag, RFID reader, and antenna. RFID tag can be a microchip attached to an antenna. Each RFID tag is attached in an object and has its unique identification number. A RFID reader can identify an object and obtain the corresponding information by querying to the attached RFID tag through appropriate signals. An antenna is used to transmit signals between RFID tag and RFID reader. In comparison with other technologies, RFID has the following benefits (fast scanning, durability, reusability, large storage, noncontact reading, security, small size, low cost, etc.). Because of these benefits, RFID can be useful in the perception layer of IoT to identify and track objects and exchange information.

1.6.2 Wireless Sensor Networks (WSN)

WSN can play a very important role in IoT. WSN can monitor and track the status of devices, and transmit the status data to the control center or sink nodes via multiple hops. Thus, WSN can be considered as the further bridge between the real world and the cyber world. In comparison with other technologies, WSN has a number of benefits, including scalability, dynamic reconfiguration, reliability, small size, low cost, and low energy consumption. All these benefits help WSN to be integrated in various areas with diverse requirements.

Notice that both RFID and WSN can be used for data acquisition in IoT, and the difference is that RFID is mainly used for object identification, while WSN is mainly used for the perception of real-world physical parameters associated with the surrounding environment.

In addition, RFID sensor network (RSN) is an integration of RFID system and sensor network. In an RSN, sensor network can cooperate with RFID system to identify and track the status of objects. In an RSN, small RFID-based sensing devices and RFID reader are implemented, where the RFID reader works as a sink node to generate data and provides power for network operations.

1.6.3 Barcode

Barcode, also denoted one-dimensional code, stores the information in several black lines and white spacings. These lines and spacings have different widths, organized in a linear or one-dimensional direction, and are arranged with special encoding rules. The information included in the barcode can be read by a machine that scans the barcode with an infrared beam. A two-dimensional code records the information by using black and white pixels laid out on the plane, in which black pixel represents a binary of '1' and white pixel represents a binary of '0'. With special encoding rules, the black and white pixels can store a significant amount of information. In comparison with barcode, two-dimensional code has the benefit of high information content, high reliability, high robustness, etc.

1.7 Transmission technologies

1.7.1 Short-range technologies

1.7.1.1 NFC is a wireless communication technology designed to build on existing High-Frequency (HF) (13.56 MHz) contactless and RFID technology. Using 13.56 MHz on the ISM band and with a typical operating distance of up to 4 cm, today NFC enables an exchange rate of between 106 Kbps and 848 Kbps. NFC creates a short-range wireless connection able to operate in three different modes of operation: card emulation, read/write, and peer-to-peer. NFC technology enables a wide range of use cases from keyless access to e-wallet in smartphone and smart tags for medical applications. This is due to ease of implementation and the ability to embed tags into credit cards, smartphones, and other wearable devices.[8]

1.7.1.2 Bluetooth is a PAN technology primarily used today as a cable replacement for short-range communication operates in the unlicensed ISM band at 2.4 GHz using a spread spectrum, frequency hopping, and full-duplex signal at a nominal rate of 1600 hops/sec. Its range varies from 1 m to 100 m depending on which class of radio is used. Class 2 is the most commonly used radio. It has a range of around 10 m and uses 2.5 mW of power. It supports data throughputs up to 2 Mbps, with up to eight connected devices. [8]

1.7.1.3 Zigbee is based on the IEEE 802.15.4 link layer and designed, promoted, and maintained by the Zigbee Alliance. The Zigbee protocol suite includes standard commissioning, security, network, and device management procedures. It takes full advantage of IEEE 802.15.4 physical radio standard and operation in unlicensed bands worldwide at 2.4 GHz (global). Raw data throughput rates of 250 Kbps can be achieved at 2.4 GHz (16 channels), 10 Kbps at 915–921 MHz (27 channels), and 100 Kbps at 868 MHz (63 channels). Transmission distances range from 10 to 100 meters, depending on power output and environmental characteristics. [8]

1.7.2 Medium-range technologies

1.7.2.1 Z-wave is a low power MAC protocol developed by Zensys that uses wireless home automation to connect 30-50 nodes and has been used for IoT communication, especially for smart home and small commercial domains. This technology is designed for small data packets at relatively low speeds up to 100 kbps and 30 meters point to point communication. Therefore, it is suitable for small messages in IoT applications, like light control, energy control, healthcare control.

Z-Wave depends on two types of devices (controlling and slave). Slave nodes properties are low cost devices unable to initiate messages. It can only reply and execute commands sent by controlling devices that initiate messages within the network. Z-Wave support mesh network topology. [9]

1.7.2.2 Wi-Fi is a wireless connectivity technology based on the IEEE 802.11 standards. Initially created for Wireless Local Area Network (WLAN) applications, Wi-Fi is also increasingly used for peer-to-peer and Wireless Personal Area Network connections (WPAN). It provides secure, reliable, and fast wireless connectivity. A Wi-Fi network can be used to connect electronic devices to each other, to the Internet, and to wired networks that use Ethernet technology. It operates in the 2.4 GHz and 5 GHz radio bands, with some products that contain both bands (dual band). It offers low power consumption and low-cost relative to cellular. Unlike cellular, Wi-Fi operates in unlicensed spectrum, resulting also in lower data transmission costs. Range is limited by proximity to a wireless router or relays, and the quality of connection can be diminished by network congestion. [8]

1.7.2.3 BLE Bluetooth provides a short distance wireless connection with low power consumption, even compared to Wi-Fi. Bluetooth Low Energy (also known as Bluetooth Smart or BLE) further reduces the power consumption profile of traditional Bluetooth. Data transfer rates are somewhat limited at about 1 Mbps (though theoretical throughput is up to 24 Mbps), though the range extends up to about 100 meters (300+ feet). uses the 2.4 GHz ISM band, it is not compatible with Bluetooth Classic. Bluetooth Low Energy uses 402 MHz-wide channels, whereas Bluetooth Classic uses 791 MHz-wide channels. Compared to Bluetooth Classic, Bluetooth Low Energy greatly reduces the power consumption of Bluetooth devices by supporting lower data throughput and enables lengthy lives for battery-operated devices. Bluetooth Low Energy also offers a beaconing capability and location-based services. Bluetooth Low Energy has proven to be very popular, triggering an explosion of new applications in spaces as diverse as fitness, toys, and automotive applications. It is now the main driving force behind many new Bluetooth standards. [8]

1.7.3 Long-range technologies

1.7.3.1 Mobile cellular networks cellular technologies provide "always-on" connectivity. Similar to mobile phones for consumer applications, cellular data for IoT can be connected over 2G, 3G, or 4G networks. Benefits include broad coverage leveraging existing base station infrastructure as well as mobility (e.g., cars). Potential drawbacks include power consumption, fees associated with data transfer over licensed spectrum owned by carriers, and potential gaps in coverage. The first-generation mobile network (1G) was all about voice and used analogy technology. 2G enabled voice and texting (Short Messaging Service – SMS) using digital technology. 3G was about voice, texting, and data. 4G was everything in 3G but faster, and 5G will be even faster. 5G will be fast enough to download a full-length HD movie in seconds. 5G is much more than just faster networks. It supports the unique combination of high-speed connectivity, very low latency, and ubiquitous coverage, making it natively suitable for supporting IoT use cases. 5G will enable us to control more devices remotely in applications where real-time network performance is critical, enabling new user experiences in many different verticals.[8]

1.7.3.2 Low Power Wide Area Networks (LPWANs) are the new phenomenon in IoT. By providing long-range communication on small, inexpensive batteries that last for years, this family of technologies is purpose-built to support large-scale IoT networks sprawling over vast industrial and commercial campuses. LPWANs can literally connect all types of IoT sensors facilitating numerous applications. Nevertheless, LPWANs can only send small blocks of data at a low rate, and therefore are better suited for use cases that don't require high bandwidth and are not time-sensitive. There exist technologies operating in both the licensed (NB-IoT, LTE-M) and unlicensed (e.g. MYTHINGS, LoRa, Sigfox) [10]

Sigfox the first LPWAN technology proposed in the IoT market, was founded in 2009 and has been growing very fast since then. Sigfox low powered connectivity solutions not only improve existing business cases but also enable a new range of opportunities for businesses across all industries. Its physical layer based on an Ultra-Narrow Band (UNB) wireless modulation, it has its proprietary system with low throughput (~100 bps) and low power Extended range (up to 50 km) , 140 messages/day/device ,also it is Subscription-based model , it has its own Cloud platform with and defined API for server access, moreover it offer roaming capability.[11]

LoRa LoRaWAN is recently developed long range communication protocol designed by the LoRa^{TM} Alliance which is an open and non-profit association. It defines Low Power Wide Area Networks (LPWAN) standard to enable IoT. Mainly its aim is to guarantee interoperability between various operators in one open global standard. LoRaWAN data rates range from 0.3 kb/s to 50 kb/s. LoRa operates in 868 and 900 MHz ISM bands. LoRa communicates between the connected nodes within 20 miles range, in unobstructed environments. Battery life for the attached node is normally very long, up to 10 years.[12]

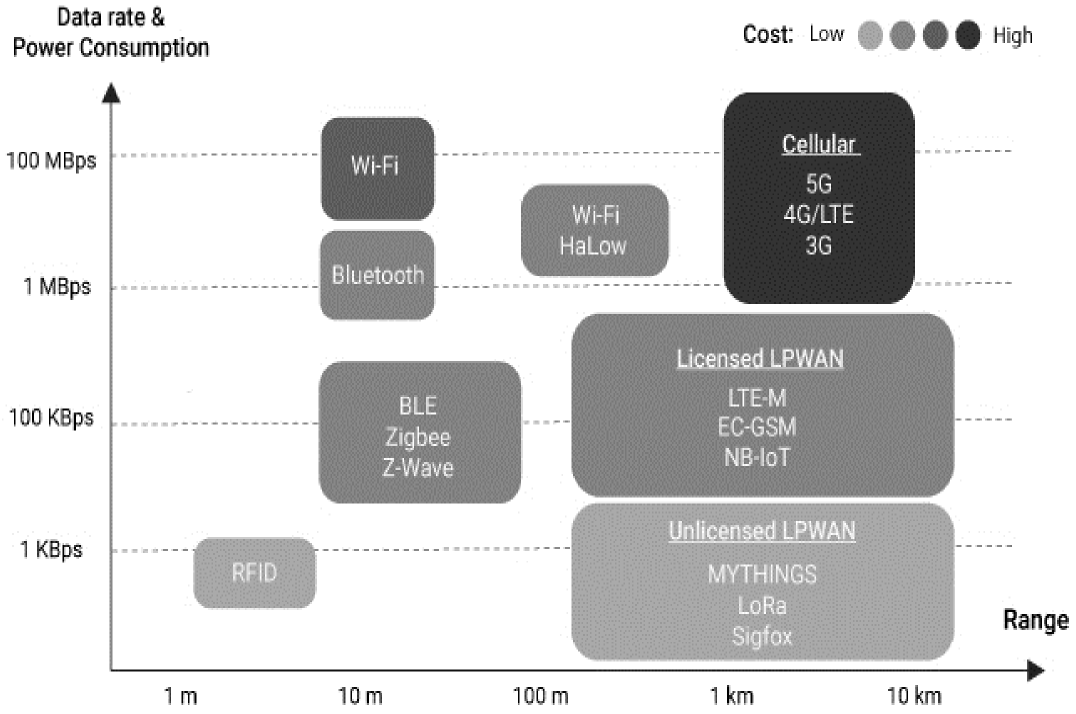


Figure 1.3 Communication technologies for connected objects [10]

Companies choose the telecommunications technology that will connect their fleet of communicating objects according to a number of criteria, in particular technical ones, such as range, speed, etc.

1.8 Data Transmission Protocols

There are protocols we aren't diving into here (table 1.1), but will occasionally show up in IoT applications: [13]

HTTP the foundational protocol of the WWW but with limited use for IoT because of it is unable to support bi-directional communication, and its high overhead.

TCP & UDP Old kids on the block. Most of the IoT protocols based on these two and add IoT-specific features on top of the basic protocols.

Table 1.1 comparison between data transmission protocols

Protocol	Advantages	Disadvantages
MQTT	• Easy to implement • Useful for connections with remote location • Small code footprint • Lightweight • Asymmetric client-server relationship	• No error-handling • Hard to add extensions • Basic message queuing implementations • Doesn't address connection security
CoAP	• Multicast support • Low overhead • Minimizing the complexity of mapping with HTTP • Communication models flexibility • Low latency	• Doesn't enable communication level security • Few existing libraries and solution support
AMQP	• Complex message queuing implementations • ISO standard • High routing reliability and security • Easily extensible • Symmetric client-server relationship	• Larger package size than other protocols • Doesn't support Last Value Queue (LVQ)

WebSocket	- Simplifies the web communication and co-network compatibility - Connection management	- Specific hardware requirements - No useful open source implementations targeted at embedded systems
XMPP-IoT	- Real time - Low latency - Easily understandable - Easily extensible - Any XMPP server may be isolated	- XML-based protocol, heavy data overhead - Not suitable for embedded IoT applications

1.9 IoT Application Domains

Presently, IoT might not have widespread visible effects on the society, but its impact is already noticeable in many industrial sectors. Considering a few among many of the IoT application domains that have emerged in recent years. It focuses mainly on some of the domains that have great potential for exponential growth in the fourth industrial revolution, namely home automation, energy, developed urban areas, transportation, healthcare, manufacturing, supply chain, wearables, and agriculture, as depicted in Figure1.4 [14]

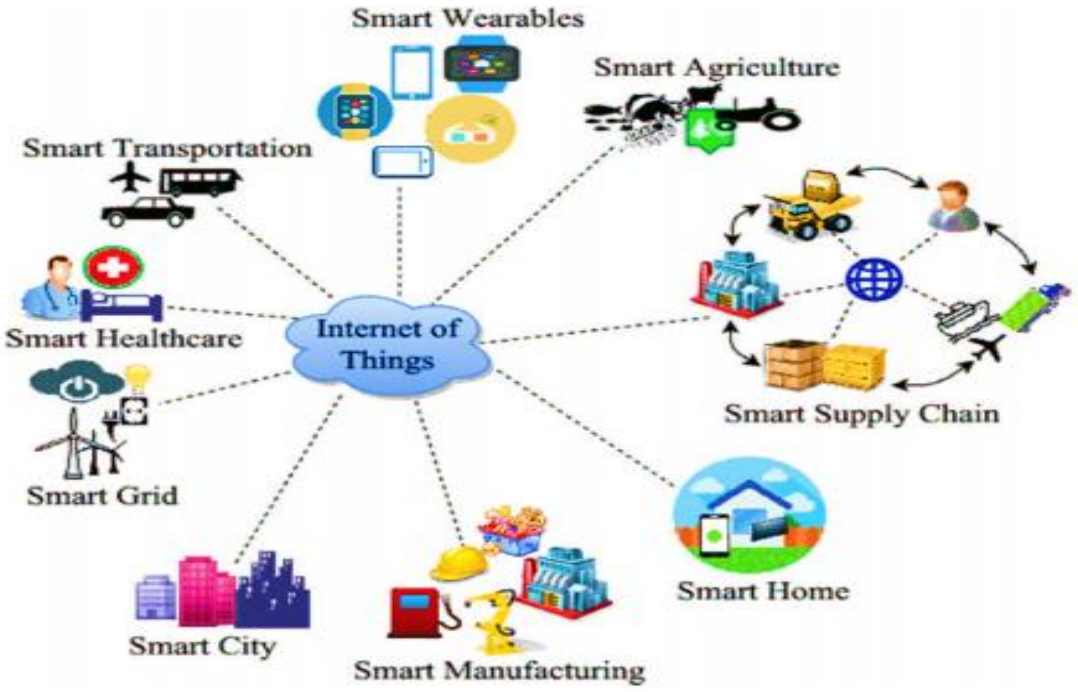


Figure 1.4 IoT application domains

1.9.1 Smart home system model

Smart home is a house or living environment where household appliances like toasters, washing machines and other everyday devices can be remotely monitored and controlled using smartphones, tablets, or laptop computers from anywhere in the World via the Internet or private network. This might enhance home care and monitoring, access control, energy efficiency, convenience and quality of everyday life. Smart TVs, security cameras, refrigerators, door locks, garage door openers, and smart hubs are typical cyber assets in the smart home domain.

The system model in Figure 1.5 comprised of a controller, which can be any IoT home automation hub, a wireless router, and 8 Wi-Fi enabled smart home appliances that can interface with a home automation software platform such as OpenHAB and Home Assistant. The software platform allows a user to wirelessly control devices from a smartphone or any computer on the home network. The controller is connected to the home network via the router Ethernet interface. A Raspberry Pi Single-Board Computer (SBC) can also be configured as the controller. The messaging protocol often employed for communication between the home automation software platform server and the smart devices is the MQTT. A typical lightweight server that implements the MQTT protocol is the Eclipse mosquito. [14]

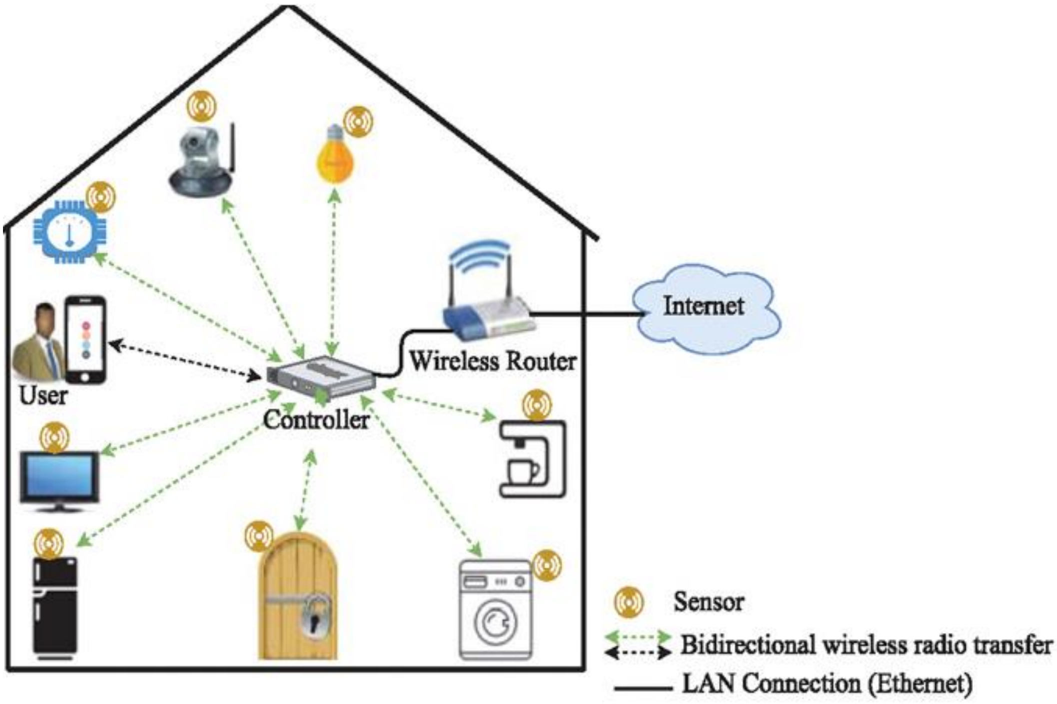

Internet
Wireless Router
User
Controller
Sensor
Bidirectional wireless radio transfer
LAN Connection (Ethernet)

Figure 1.5 Smart home system model

1.9.2 Smart transportation system model

Smart transportation provides a safer, cleaner and more efficient transport system by using real-time traffic information and interconnecting vehicles and roadside infrastructures for more efficient data acquisition. Additionally, it will improve quality of life by decreasing congestion, and hence shortening travel time, as well as reducing fuel/electricity usage.

The system model (Figure 1.6) shows a robust transportation system that exploits seamless information coordination and exchange across the different components of the network to provide an efficient and safe transportation. The system components include smart traffic lights, cameras, radar, GPS, smart vehicle, smart trains, and smart road infrastructure. The system utilizes real-time GPS location data, traffic flow prediction mechanism, location-based services, smart traffic light scheduling management, and enormous amount of sensor data collected from different areas in the network for making final decisions.[14]

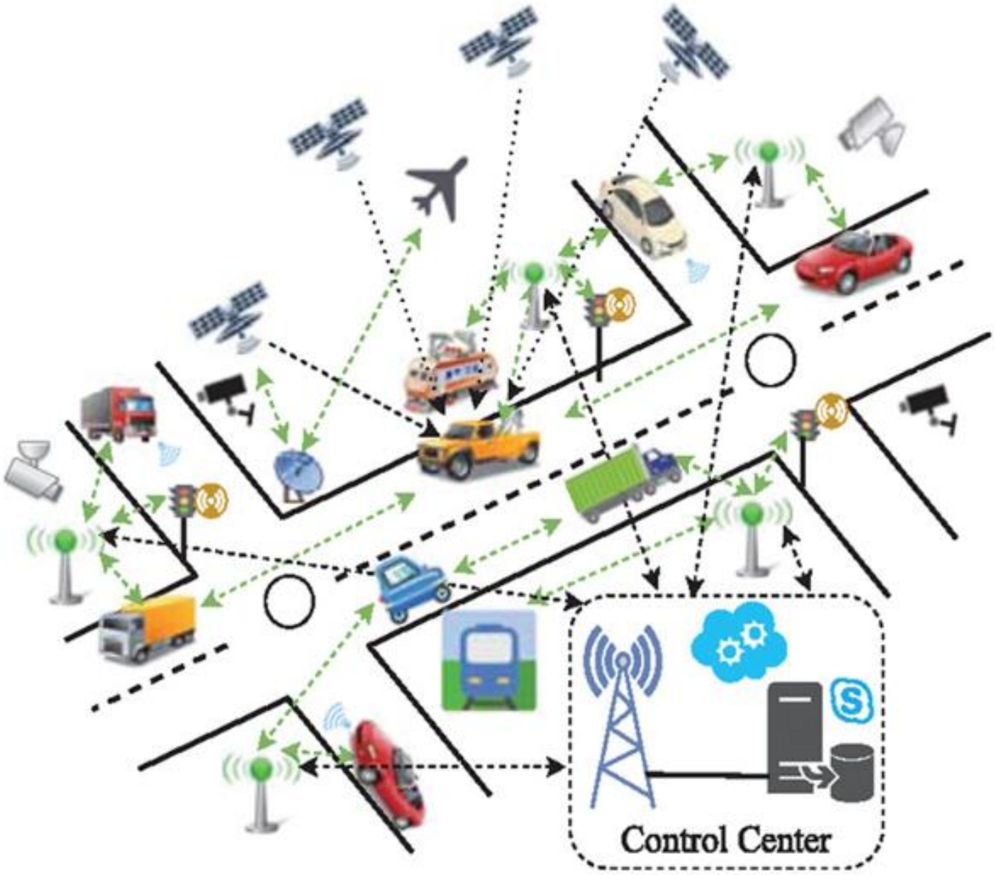


Figure 1.6 Smart transportation system model

1.9.3 Smart supply chain system model

Smart supply chain Refers to a proactive and customer-centric monitoring, tracking, and remote asset management system that integrates different technologies (IoT, WSNs, RFID, big data, cloud computing…) to provide information on location, status, environment, and functionality of products and services. to reduce operational costs, and allows for timely responses to unexpected events.

Figure 1.7 represents a system model for smart supply chain based on Blockchain comprising production and distribution systems of geographically dispersed enterprises that collaborate in a secure manner to jointly and efficiently produce and deliver end products to customers. In the context of supply chain, IoT security issues will typically revolve around authentication, connection and transaction. However, using the distributed ledger in Blockchain, the enterprises, namely, supplier of raw materials, factory, and distributor can conduct business transactions in a trusted and secure environment. The immutable record of Blockchain enables reliable creation of networks histories, allowing for tracking the actions of network devices. [14]

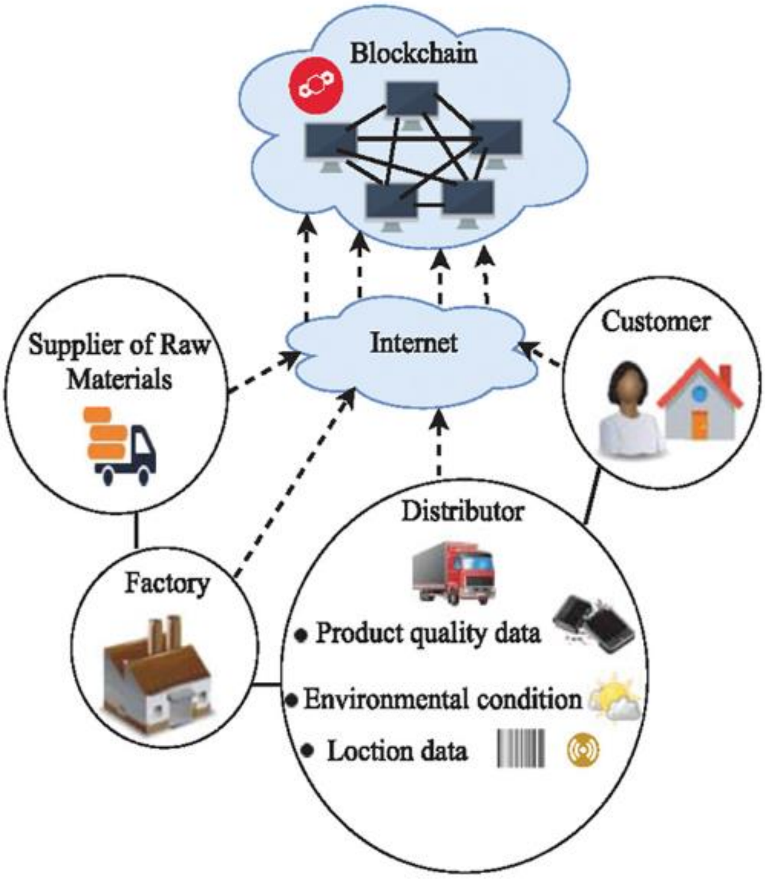


Figure 1.7 Smart supply chain system model

1.9.4 Smart agriculture system model

Smart agriculture is a sustainable farming practice that employs IT and other relevant technologies to increase the per unit yield of farming land by optimizing water use and preserving other natural resources in order to increase crop yields and financial returns.

The Figure 1.8 shows a diagram representing the proposed smart agriculture system model. Plants normally require specific environmental conditions for optimal growth, health and overall crop yield. Thus, the system model consists of different sensor networks for measuring soil moisture, temperature, sun light, humidity, as well as sensor networks for monitoring the location of farm animals, and for prompt disease detection. Onboard the UAV are a SBC, a flight control system (autopilot) like Pixhawk, sensor and camera. The sensor is used to collect real-time data from the flight control system every second and sent to a server via 4G LTE dongle attached to the SBC.

The autopilot can also receive commands through the server, hence by connecting to the server via the Internet, the farmer can control the drone by viewing the real-time data and issuing control commands.[14]

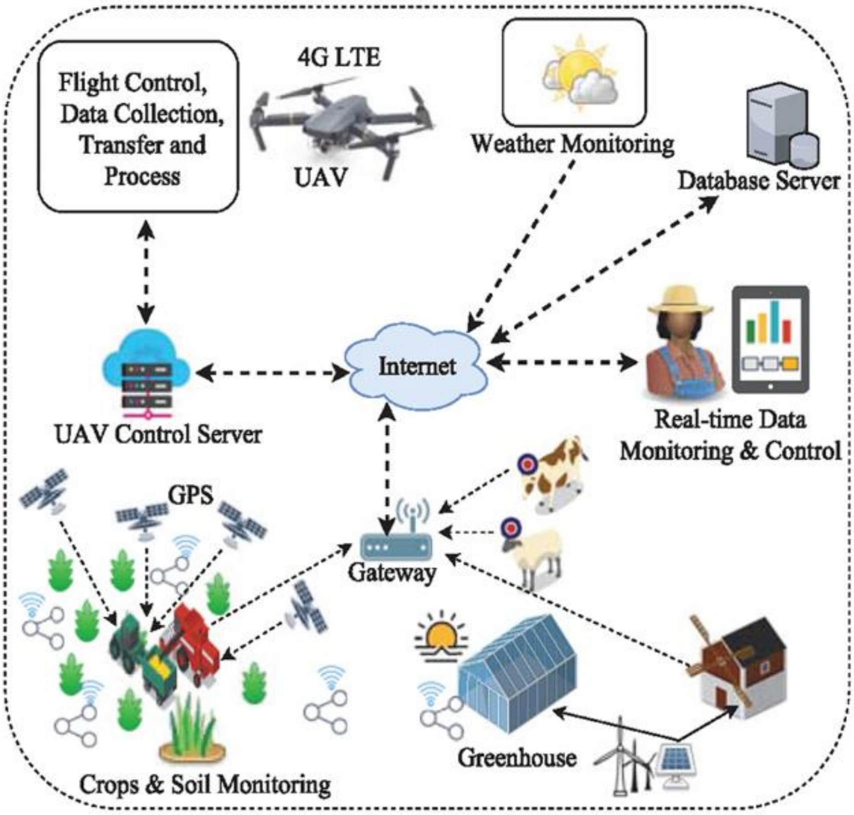


Figure 1.8 Smart agriculture system model

1.9.5 Smart wearables system model

Smart wearables are end-to-end integrated gadgets embedded with smart sensors and actuators that connect wirelessly to the smartphone or tablet of the user, often using Bluetooth Low Energy (BLE) technology. They are usually worn on the wrist, clipped to the body, or hung around the neck for the purpose of staying fit, being more organized, losing weight, staying active, or for tele-medicine purposes.

The model shown in Figure 1.9 is made up of different components, including an athlete, an elderly outpatient, a coach, an emergency team, a doctor, and a weather forecast server. The athlete is wearing ECG sensor and motion sensors, and the elderly patient is wearing ECG sensor, respiratory rate sensor, and motion sensors. These sensors collect and send data to the smartphones via Bluetooth on regular basis, and the smartphones upload the data to a cloud server via cell phone network. The coach can only access information pertaining to the progress of the athlete; similarly, the doctor can only access information pertaining to the health of his patient. However, in case of emergency condition, the emergency team and the doctor will receive an urgent message from the ECG sensor on the athlete, or from the ECG sensor and/or respiratory rate sensor on the patient. [14]

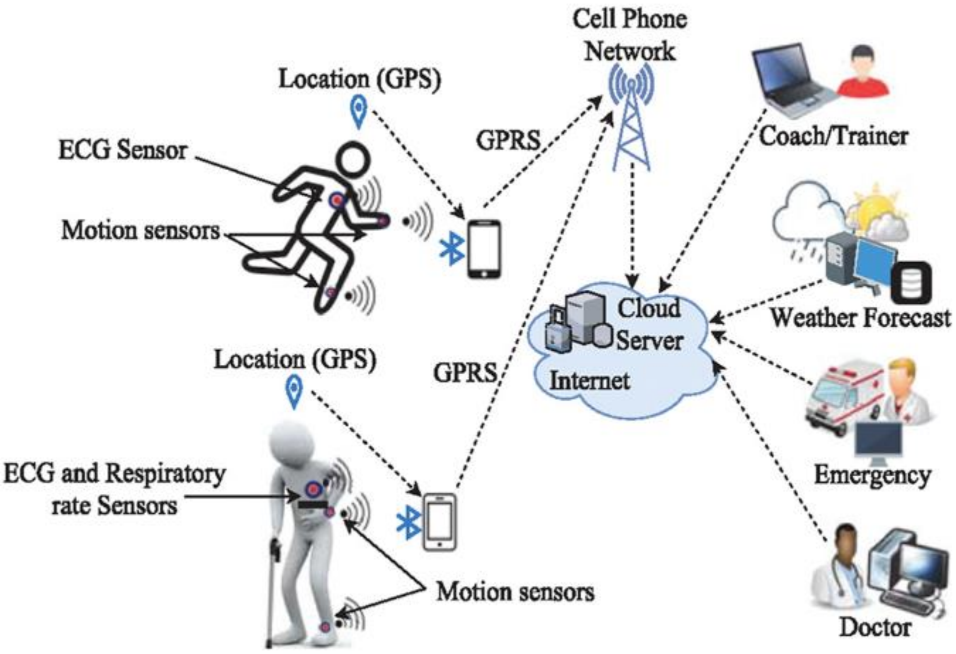


Figure 1.9 Smart wearables system model

1.9.6 Smart healthcare monitoring system model

Smart healthcare refers to a health care paradigm that allows for remote healthcare monitoring and Telehealth, where doctors and other medical practitioners can examine, diagnose and treat patients remotely. Smart healthcare services are becoming commonplace, especially in countries like India. Typical cyber assets in this domain include: glucose monitoring systems, infusion pumps, implantable, pacemakers, insulin pumps, electrocardiograms, medical databases and mobile devices like smartphones.

Figure 1.10 shows a typical smart healthcare monitoring system which consists of 2 outpatients, a smart hospital, and an emergency team. The 2 outpatients are: a typical patient wearing Electroencephalography (EEG) sensor, Blood Pressure (BP) sensor and Blood Glucose (BG) sensor, and an elderly patient that needs constant monitoring, wearing Electrocardiogram (ECG) sensor and BP sensor. The sensors on the patient's bodies continuously collect and send data to their smartphones via Bluetooth, and the smartphones in turn upload the data to the medical server via the Internet. In case patients are in critical condition, these sensors can immediately report the physical condition of the patient to the emergency team and to their doctors for appropriate actions to be taken. [14]

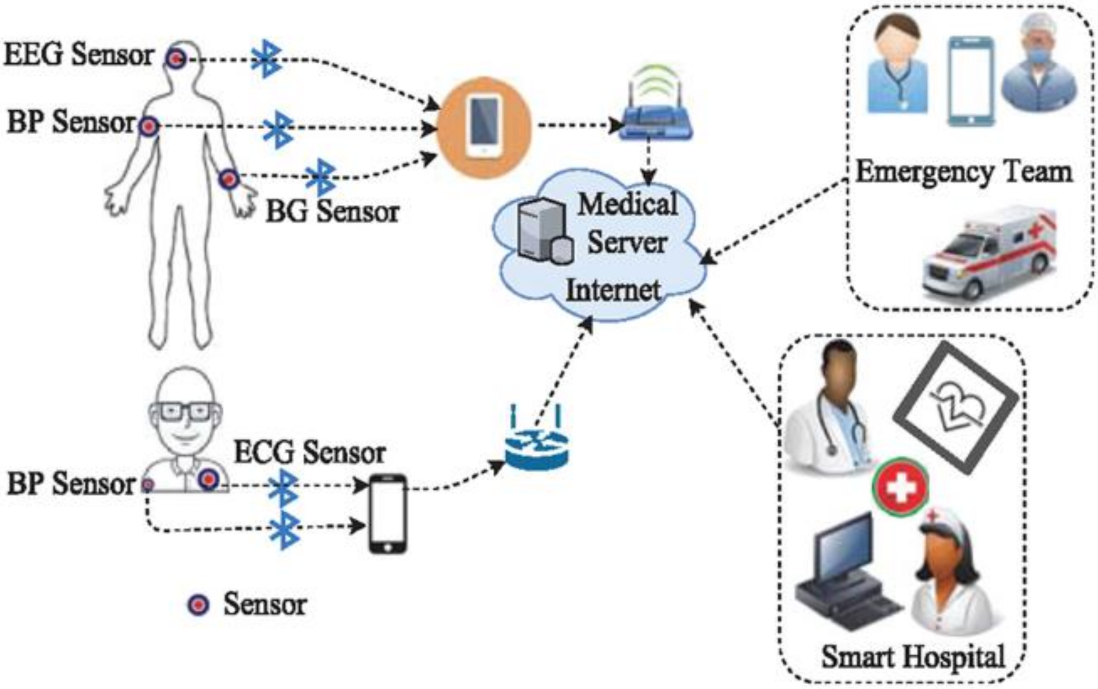


Figure 1.10 Smart healthcare monitoring system model

1.10 IoT Challenges

Challenges Beyond costs and the ubiquity of devices, other security issues plague IoT:[15]

1.10.1 Privacy and Security

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

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

1.10.2 Cost versus Usability

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

1.10.3 Interoperability

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

1.10.4 Data Management

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

1.10.5 Device Level Energy Issues

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

1.11 Conclusion

Obviously, the Internet of Things as an evolution of the current Internet has enormous improvement in our lives and the way in which smart devices around us interact with each other or with their users in such a way that our activities in various sectors (industry, health, agriculture, etc.) can be effectively controlled. In this chapter, we mainly discussed the important technologies as well as the featured applications of IoT. We have also mentioned some drawbacks that should be carefully achieved.

Chapter two

publish subscribe communications in the IoT: the case of e-healthcare applications

2.1 Introduction

Nowadays, typical systems tend to work at Internet-scale around geographically distributed data centers. Their servers host ephemeral application modules for handling user traffic. Seasonal traffic spikes, latency, and network data corruption add more layers of complexity to information systems dealing with today's volume of data. [16]

The decoupled nature of the Pub/Sub pattern makes it a good candidate for governing the architecture of dynamically scalable systems. Pub/Sub makes it possible to manage scale without overloading the program logic of system components.

In the other hand, the recent advancements in wireless communication side and integrated circuits led to the development of low-power, small sized, intelligent, micro/nano-technology sensor nodes that can be placed inside/outside the human body to be utilized for various applications like health monitoring.

Thus, the primary focus in this chapter will be firstly in the pub/sub communication model generally and MQTT protocol specifically. Then, we will talk about the healthcare monitoring and its benefits services in terms of treating illness from distance. And sum up with some scenarios implemented in that domain.

2.2 Publish subscribe

2.2.1 Definition

Publish/subscribe (pub/sub) is a many-to-many communication model that directs the flow of messages from senders to receivers based on receivers' data interests. In this model, publishers (i.e., senders) generate messages without knowing their receivers; subscribers (who are potential receivers) express their data interests, and are subsequently notified of the messages from a variety of publishers that match their interests. [17]

2.2.2 The Basic Interaction scheme

The publish/subscribe interaction paradigm provides subscribers with the ability to express their interest in an event or a pattern of events, in order to be notified subsequently of any event, generated by a publisher, that matches their registered interest. In other terms, producers publish information on a software bus (an event manager) and consumers subscribe to the information they want to receive from that bus. This information is typically denoted by the term event and the act of delivering it by the term notification.

The basic system model for publish/subscribe interaction (Figure 2.1) relies on an event notification service providing storage and management for subscriptions and efficient delivery of events. Such an event service represents a neutral mediator between publishers, acting as producers of events, and subscribers, acting as consumers of events. Subscribers register their interest in events by typically calling a subscribe () function on the event service, without knowing the effective sources of these events. This subscription information remains stored in the event service and is not forwarded to publishers. The symmetric function unsubscribe () terminates a subscription.

To generate an event, a publisher typically calls a publish () function. The event service propagates the event to all relevant subscribers; it can thus be viewed as a proxy for the subscribers. Note that every subscriber will be notified of every event conforming to its interest (obviously, failures might prevent subscribers from receiving some events). [19]

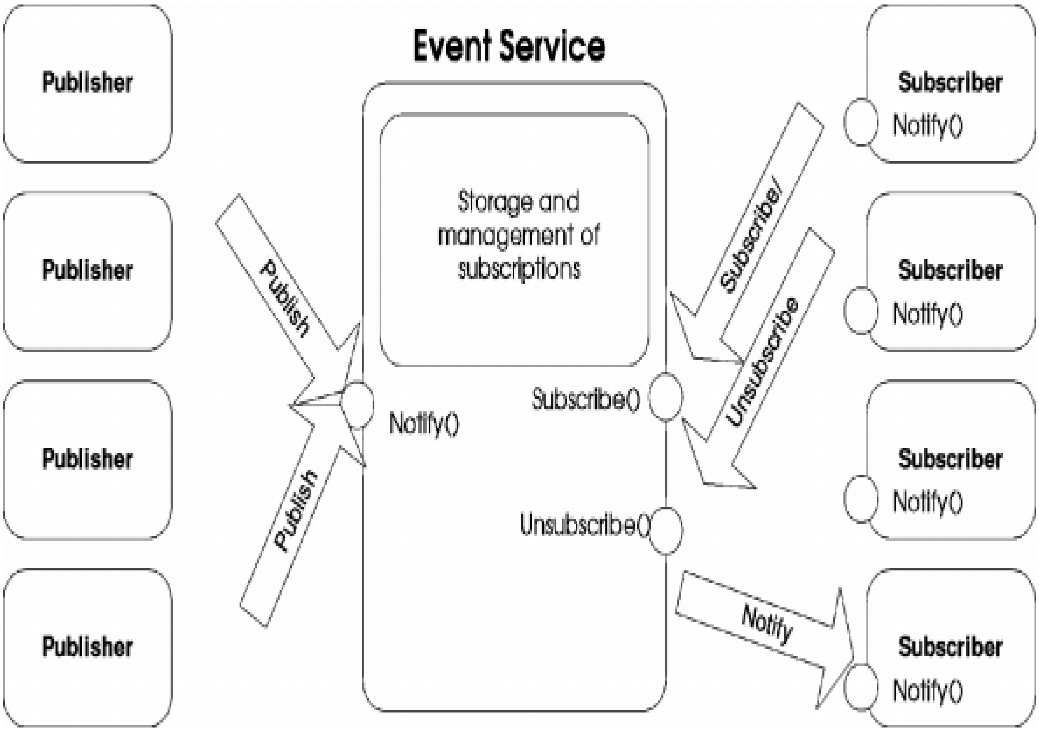


Figure 2.1 A simple object based publish/ subscribe system

The decoupling that the event service provides between publishers and subscribers can be decomposed along the following three dimensions:

2.2.2.1 Space decoupling the interacting parties do not need to know each other. The publishers publish events through an event service and the subscribers get these events indirectly through the event service. The publishers do not usually hold references to the subscribers, neither do they know how many of these subscribers are participating in the interaction. Similarly, subscribers do not usually hold references to the publishers, neither do they know how many of these publishers are participating in the interaction.

2.2.2.2 Time decoupling the interacting parties do not need to be actively participating in the interaction at the same time. In particular, the publisher might publish some events while the subscriber is disconnected, and conversely, the subscriber might get notified about the occurrence of some event while the original publisher of the event is disconnected.

2.2.2.3 Synchronization decoupling publishers are not blocked while producing events, and subscribers can get asynchronously notified (through a callback) of the occurrence of an event while performing some concurrent activity. The production and consumption of events do not happen in the main flow of control of the publishers and subscribers, and do not therefore happen in a synchronous manner.

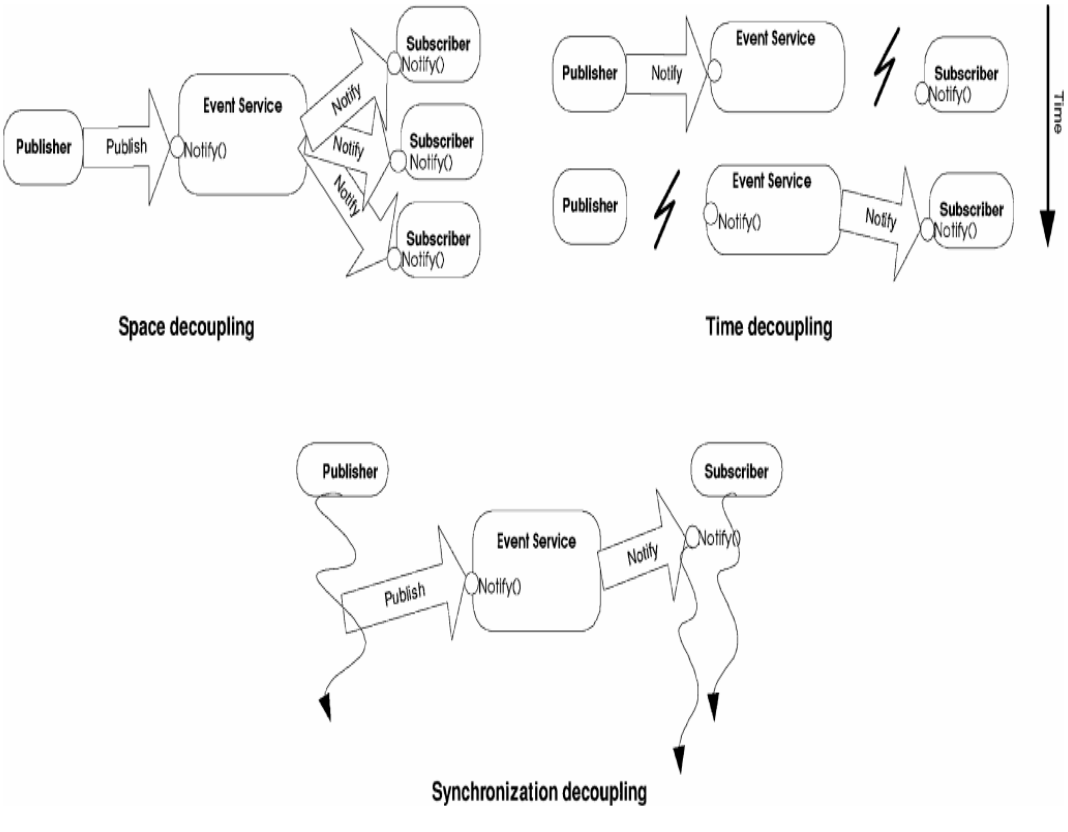


Figure 2.2 Space, time, and synchronization decoupling with the pub/sub paradigm

2.2.3 Key functions implemented by the broker

There are three main components to the Publish/subscribe Model: publishers, subscribers and broker. It's clear that the broker plays a pivotal role in the pub/sub process. We mention:[19]

2.2.3.1 Event filtering (event selection) the process which selects the set of subscribers that have shown interest in a given event. Subscriptions are stored in memory and searched when a publisher publishes a new event. But how does the broker manage to filter all the messages so that each subscriber receives only messages of interest? the broker has several filtering options:[20]

- **Option 1: subject-based filtering**

Is based on the subject or topic that is part of each message. The receiving client subscribes to the broker for topics of interest. From that point on, the broker ensures that the receiving client gets all message published to the subscribed topics. In general, topics are strings with a hierarchical structure that allow filtering based on a limited number of expressions.

- **Option 2: content-based filtering**

The broker filters the message based on a specific content filter-language. The receiving clients subscribe to filter queries of messages for which they are interested. A significant downside to this method is that the content of the message must be known beforehand and cannot be encrypted or changed.

- **Option 3: type-based filtering**

When object-oriented languages are used for the implementation of publish/subscribe protocols, filtering based on the type/class of a message (event) is a common practice. For example, a subscriber can listen to all messages, which are of type Exception or any sub-type.

2.2.3.2 Event routing (event delivery) the process of routing the published events from the publisher to all interested subscribers after filtering the events, and can be done in unicast, multicast.

N.B P/S middleware service, event service, event bus, server are different terms, but indicate the same meaning as broker, they are just in different resources.

2.2.4 Classification of Pub/Sub Architectures

There are many architectures mentioned as follow: [21]

2.2.4.1 Centralized Broker

Consists of multiple publishers and subscribers with centralized broker.

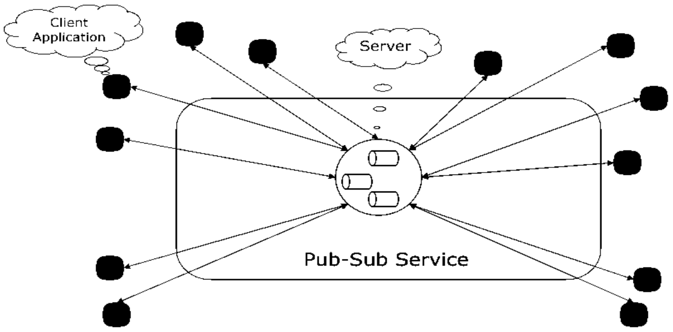


Figure 2.3 Centralized Broker Architecture

2.2.4.2 Centralized Multi-Broker

Many brokers/servers existed so that each topic can be placed on a different one.

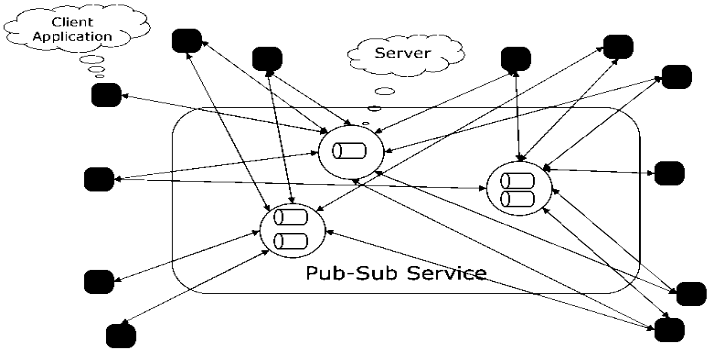


Figure 2.4 Centralized Multi-Broker Architecture

2.2.4.3 De-centralized Brokered

Application use messaging to interact with Service Access points. Pub/Sub Service distributes messages internally between servers and can be peer-to-peer, multicast, etc.

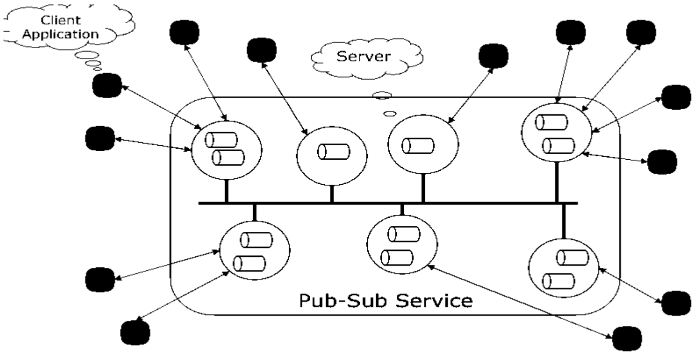


Figure 2.5 De-centralized Brokered Architecture

2.2.4.4 De-Centralized Unbrokered (Peer-to-peer)

Each node can be publisher, subscriber or broker. Subscribers subscribe to publishers directly and publishers notify subscribers directly. Therefore, they must maintain knowledge of each other.

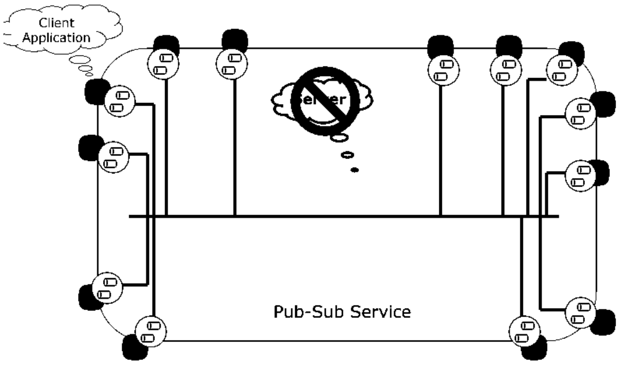


Figure 2.6 Peer to Peer Architecture

2.2.5 **Advantages of Pub/Sub**

- Highly suited for mobile applications, ubiquitous computing and distributed embedded systems.
- **Robust** failure of publishers or subscribers does not bring down the entire system.
- **Scalability** suited to build distributed applications consisting a large number of entities.
- **Adaptability** can be varied to suit different environments (mobile, internet game, embedded systems etc....).[22]

2.2.6 **Disadvantages of Pub/Sub**

- **Reliability** no strong guarantee on broker to deliver content to subscriber. After a publisher publishes the event, it assumes that all corresponding subscribers would receive it.
- **Potential bottleneck** in brokers when subscribers and publishers overload them. (Solve by load balancing techniques)
- **Security issues**

Encryption hard to implement when the brokers has to filter out the events according to context.

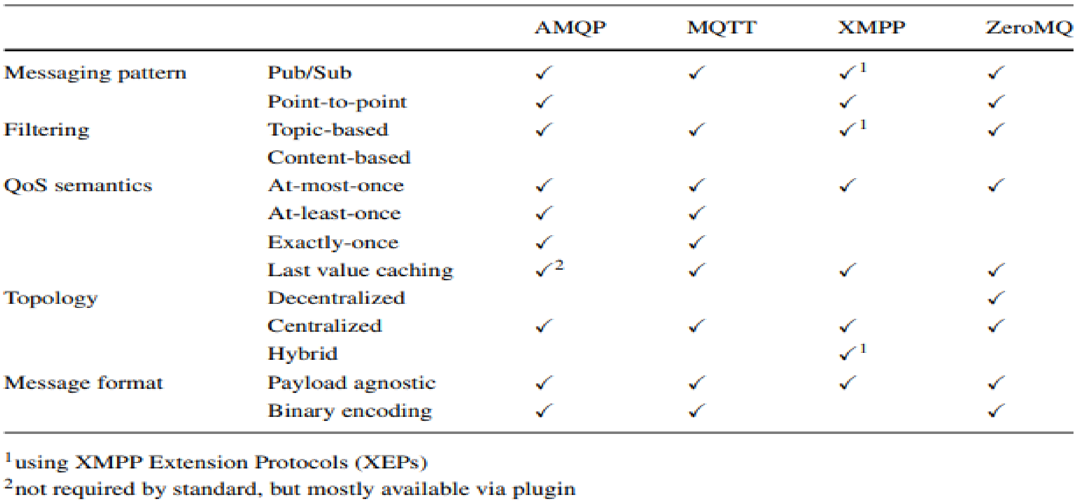
Brokers might be fooled into sending notifications to the wrong client, amplifying denial of service requests against the client.[22]

2.2.7 Pub/Sub Protocols

Many standardized messaging protocols that implement Publish/Subscribe pattern exist. In the area of application level protocols, the most interesting ones are: [23]

- AMQP, Advanced Message Queueing Protocol
- MQTT, MQ Telemetry Transport
- XMPP, Extensible Messaging and Presence Protocol
- ZeroMQ, Also known as ØMQ, 0MQ, or zmq

Table 2.1 Classification of Open pub/sub middleware protocols[23]



		AMQP	MQTT	XMPP	ZeroMQ
Messaging pattern	Pub/Sub	✓	✓	✓[1]	✓
	Point-to-point	✓		✓	✓
Filtering	Topic-based	✓	✓	✓[1]	✓
	Content-based				
QoS semantics	At-most-once	✓	✓	✓	✓
	At-least-once	✓	✓		
	Exactly-once	✓	✓		
	Last value caching	✓[2]	✓	✓	✓
Topology	Decentralized				✓
	Centralized	✓	✓	✓	✓
	Hybrid			✓[1]	
Message format	Payload agnostic	✓	✓	✓	✓
	Binary encoding	✓	✓		✓

[1] using XMPP Extension Protocols (XEPs)
[2] not required by standard, but mostly available via plugin

Since our solution is based on the implementation of the MQTT will be explain in the next chapters, the next section will focus on that protocol in details.

2.2.7.1 What is MQTT ?

MQTT is a lightweight message queueing and transport protocol, as its name implies, is suited for the transport of telemetry data (sensor and actor data) ,and it is very lightweight and thus suited for M2M (Mobile to Mobile), WSN (Wireless Sensor Networks) and ultimately IoT (Internet of Things) scenarios where sensor and actor nodes communicate with applications through the MQTT message broker.[24]

2.2.7.2 MQTT characteristics

- Lightweight message queueing and transport protocol.
- Asynchronous communication model with messages (events).
- Low overhead (2 bytes header) for low network bandwidth applications.
- Publish / Subscribe (Pub/Sub) model.

- Decoupling of data producer (publisher) and data consumer (subscriber) through topics (message queues).
- Simple protocol, aimed at low complexity, low power and low footprint implementations (e.g. WSN - Wireless Sensor Networks).
- Runs on connection-oriented transport (TCP). To be used in conjunction with 6LoWPAN (TCP header compression).[24]

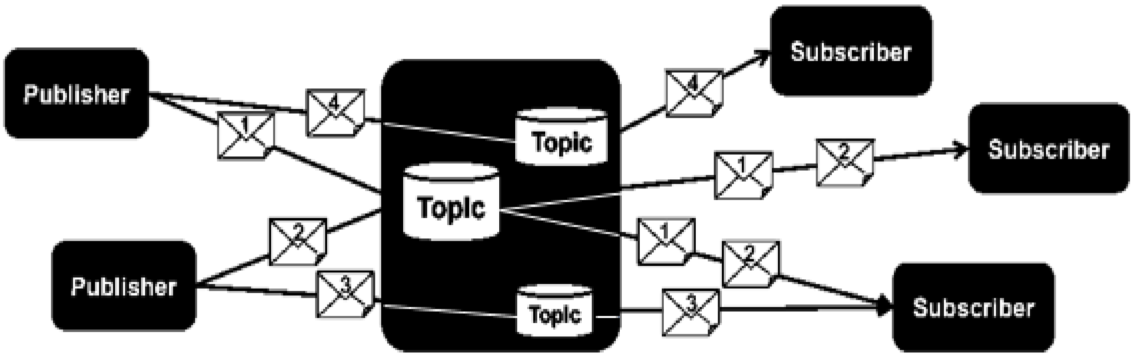


Figure 2.7 MQTT Protocol

2.2.7.3 Origins and future of MQTT standard

MQTT was initially developed by IBM and Eurotech. The previous protocol version 3.1 was made available under http://mqtt.org/.

In 2014, MQTT was adopted and published as an official standard by OASIS (published V3.1.1). As such, OASIS has become the new home for the development of MQTT. The OASIS TC (Technical Committee) is tasked with the further development of MQTT.

Version 3.1.1 of MQTT is backward compatible with 3.1 and brought only minor changes:

- Changes restricted to the CONNECT message.
- Clarification of version 3.1 (mostly editorial changes).[24]

2.2.7.4 MQTT model

The core elements of MQTT are clients, servers (brokers), sessions, subscriptions and topics: [24]

- **MQTT client (publisher, subscriber)** clients subscribe to topics to publish and receive messages. Thus, subscriber and publisher are special roles of a client.
- **MQTT server (broker)** servers run topics, i.e. receive subscriptions from clients on topics, receive messages from clients and forward these, based on client's subscriptions, to interested clients.

- **Topic** technically, topics are message queues. Topics support the publish/subscribe pattern for clients. Logically, topics allow clients to exchange information with defined semantics.
- **Session** identifies a (possibly temporary) attachment of a client to a server. All communication between client and server takes place as part of a session.
- **Subscription** unlike sessions, a subscription logically attaches a client to a topic. When subscribed to a topic, a client can exchange messages with a topic.
- **Message** messages are the units of data exchange between topic clients.

2.2.7.5 MQTT message format

MQTT messages contain a mandatory fixed-length header (2 bytes) and an optional message specific variable length header and message payload. Optional fields usually complicate protocol processing. However, MQTT is optimized for bandwidth constrained and unreliable networks (typically wireless networks), so optional fields are used to reduce data transmissions as much as possible. [24]

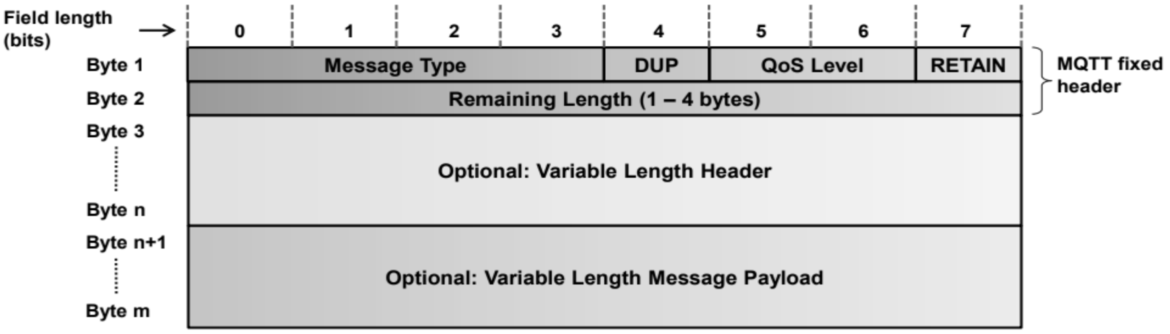
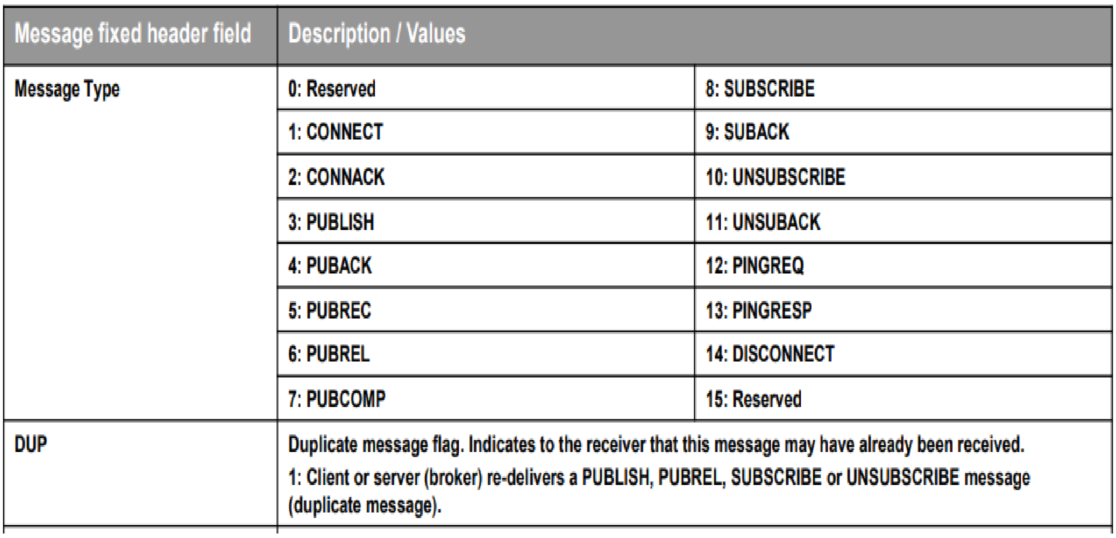
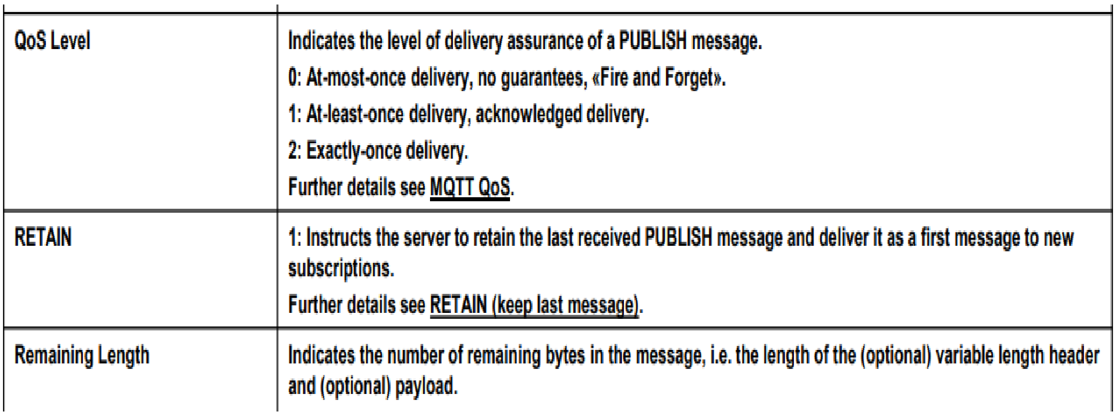


Figure 2.8 MQTT message format

Table 2.2 Overview of fixed header fields



Message fixed header field	Description / Values	
Message Type	0: Reserved	8: SUBSCRIBE
	1: CONNECT	9: SUBACK
	2: CONNACK	10: UNSUBSCRIBE
	3: PUBLISH	11: UNSUBACK
	4: PUBACK	12: PINGREQ
	5: PUBREC	13: PINGRESP
	6: PUBREL	14: DISCONNECT
	7: PUBCOMP	15: Reserved
DUP	Duplicate message flag. Indicates to the receiver that this message may have already been received. 1: Client or server (broker) re-delivers a PUBLISH, PUBREL, SUBSCRIBE or UNSUBSCRIBE message (duplicate message).	



QoS Level	Indicates the level of delivery assurance of a PUBLISH message. 0: At-most-once delivery, no guarantees, «Fire and Forget». 1: At-least-once delivery, acknowledged delivery. 2: Exactly-once delivery. Further details see MQTT QoS.
RETAIN	1: Instructs the server to retain the last received PUBLISH message and deliver it as a first message to new subscriptions. Further details see RETAIN (keep last message).
Remaining Length	Indicates the number of remaining bytes in the message, i.e. the length of the (optional) variable length header and (optional) payload.

- **CONNECT message format** The CONNECT message contains many session-related information as optional header fields. Brokers require from clients that when sending the CONNECT message, they should define the username/password combination before validating the connection, or refusing it in case the authentication was unsuccessful.

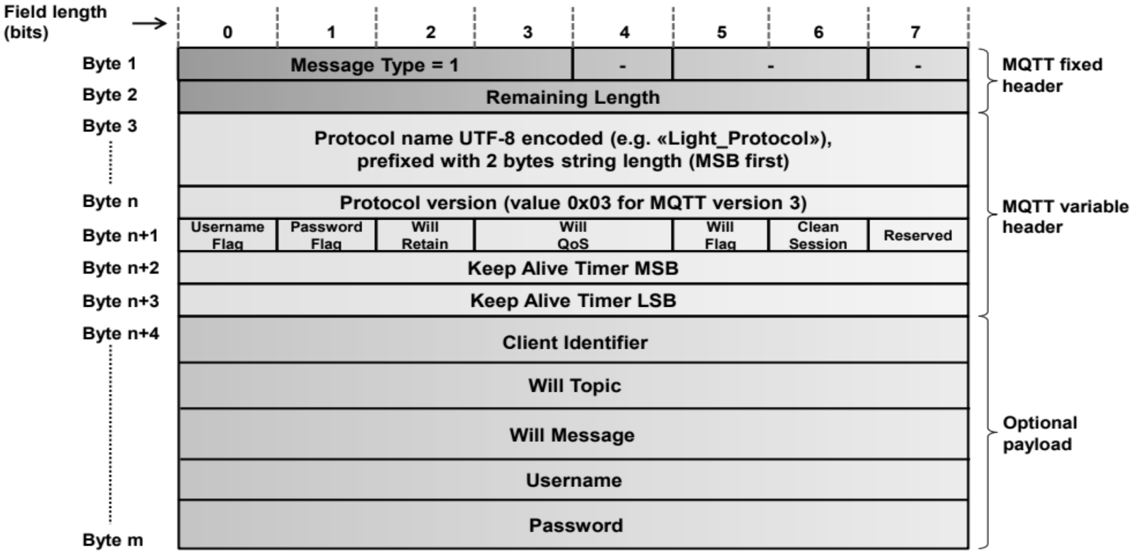
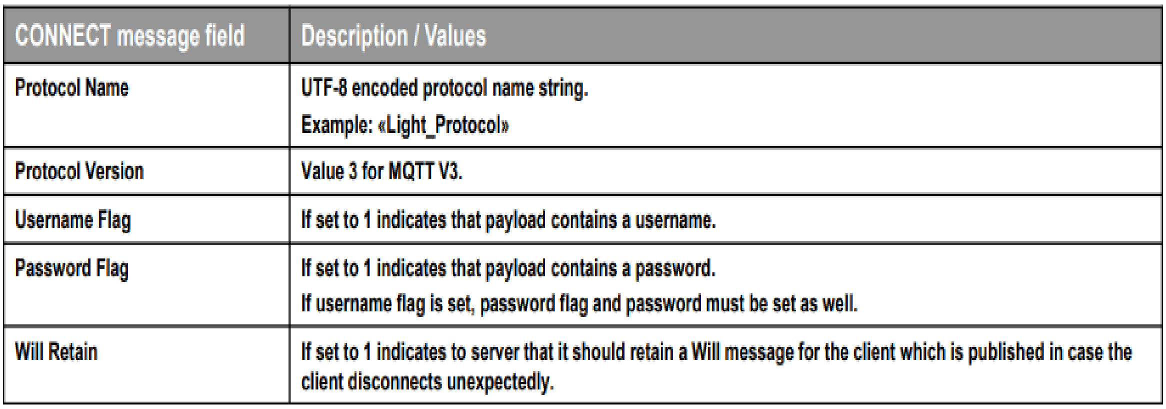
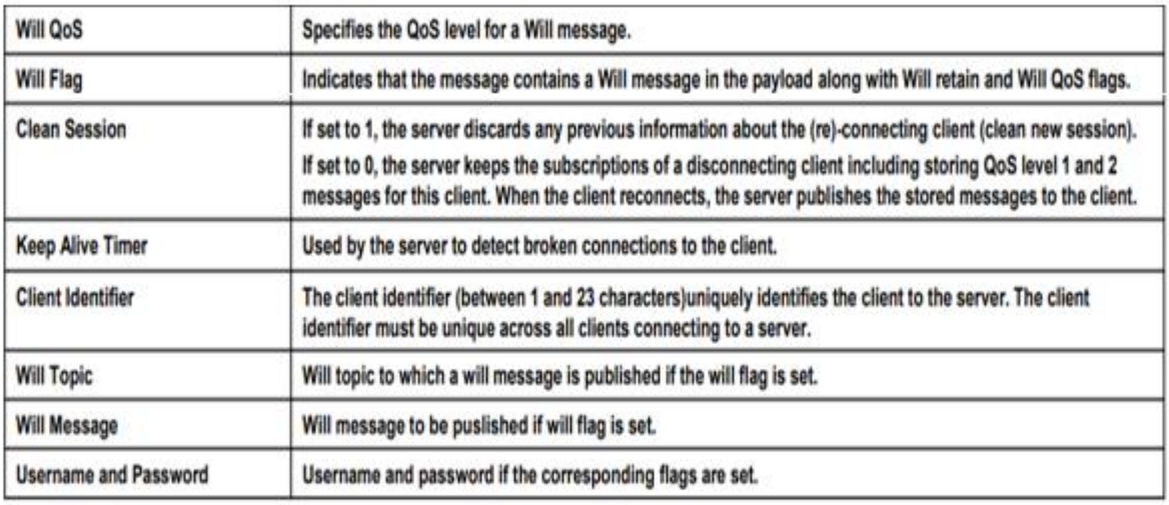


Figure 2.9 CONNECT message format

Table 2.3 Overview of CONNECT message fields



CONNECT message field	Description / Values
Protocol Name	UTF-8 encoded protocol name string. Example: «Light_Protocol»
Protocol Version	Value 3 for MQTT V3.
Username Flag	If set to 1 indicates that payload contains a username.
Password Flag	If set to 1 indicates that payload contains a password. If username flag is set, password flag and password must be set as well.
Will Retain	If set to 1 indicates to server that it should retain a Will message for the client which is published in case the client disconnects unexpectedly.



Will QoS	Specifies the QoS level for a Will message.
Will Flag	Indicates that the message contains a Will message in the payload along with Will retain and Will QoS flags.
Clean Session	If set to 1, the server discards any previous information about the (re)-connecting client (clean new session). If set to 0, the server keeps the subscriptions of a disconnecting client including storing QoS level 1 and 2 messages for this client. When the client reconnects, the server publishes the stored messages to the client.
Keep Alive Timer	Used by the server to detect broken connections to the client.
Client Identifier	The client identifier (between 1 and 23 characters)uniquely identifies the client to the server. The client identifier must be unique across all clients connecting to a server.
Will Topic	Will topic to which a will message is published if the will flag is set.
Will Message	Will message to be puslished if will flag is set.
Username and Password	Username and password if the corresponding flags are set.

- **CONNACK message format**

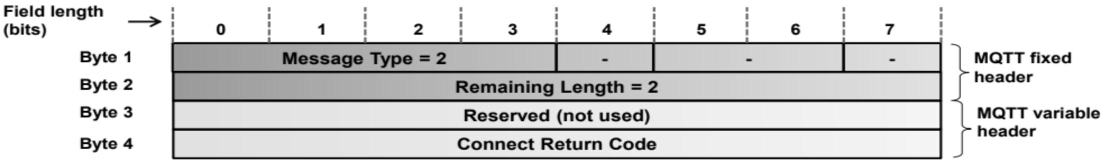
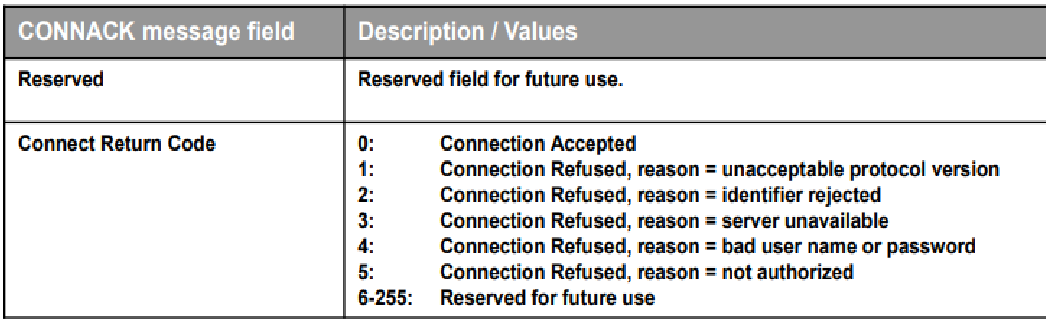


Figure 2.10 CONNACK message format

Table 2.4 Overview of CONNACK message fields



CONNACK message field	Description / Values
Reserved	Reserved field for future use.
Connect Return Code	0: Connection Accepted 1: Connection Refused, reason = unacceptable protocol version 2: Connection Refused, reason = identifier rejected 3: Connection Refused, reason = server unavailable 4: Connection Refused, reason = bad user name or password 5: Connection Refused, reason = not authorized 6-255: Reserved for future use

- **PUBLISH message format**

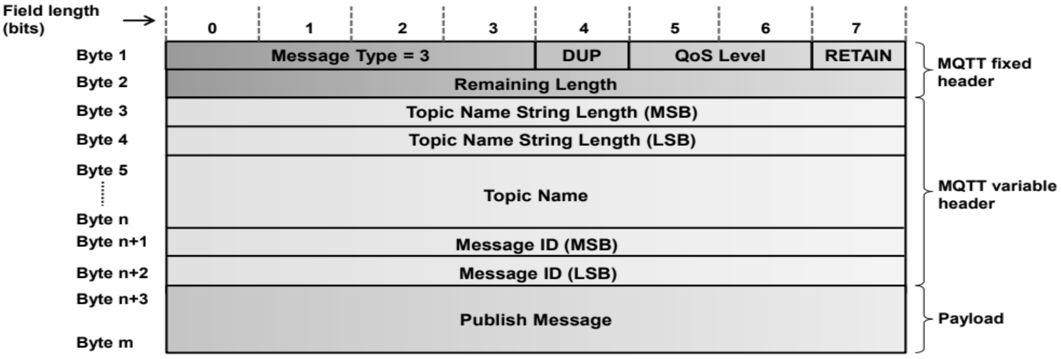
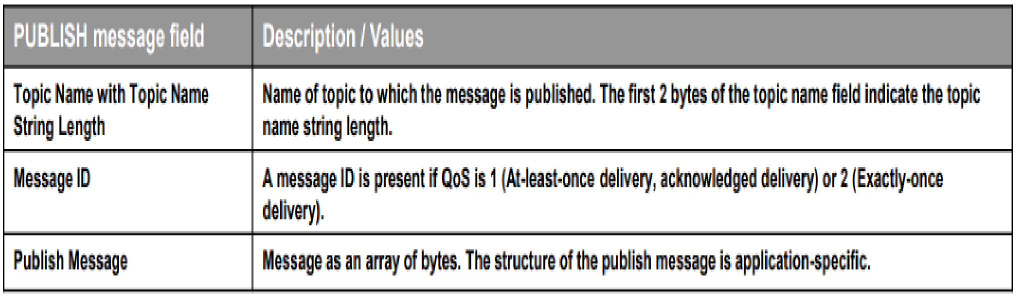


Figure 2.11 PUBLISH message format

Table 2.5 Overview of PUBLISH message fields



PUBLISH message field	Description / Values
Topic Name with Topic Name String Length	Name of topic to which the message is published. The first 2 bytes of the topic name field indicate the topic name string length.
Message ID	A message ID is present if QoS is 1 (At-least-once delivery, acknowledged delivery) or 2 (Exactly-once delivery).
Publish Message	Message as an array of bytes. The structure of the publish message is application-specific.

- **PUBACK message format**

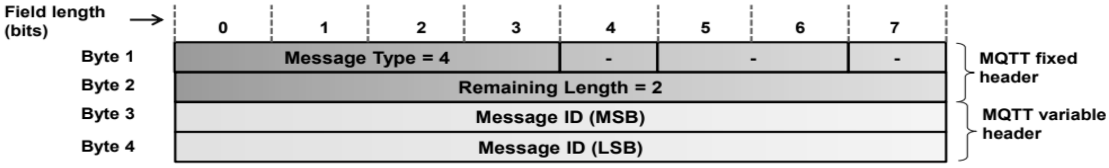
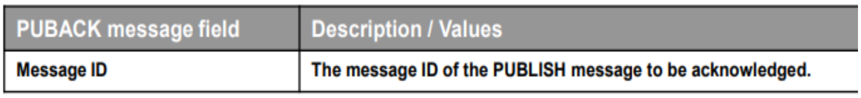


Figure 2.12 PUBACK message format

Table 2.6 Overview of PUBACK message field



PUBACK message field	Description / Values
Message ID	The message ID of the PUBLISH message to be acknowledged.

- **PUBREC message format**

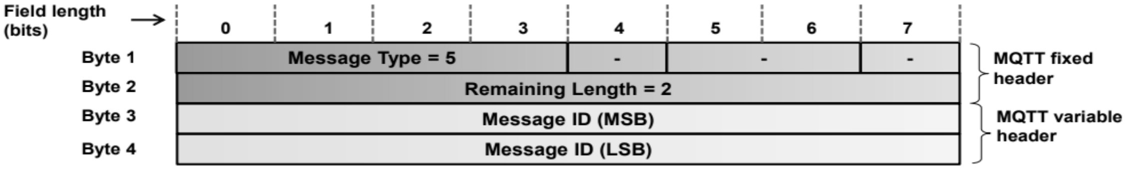
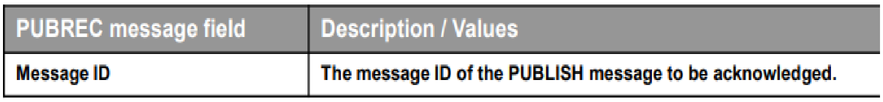


Figure 2.13 PUBREC message format

Table 2.7 Overview of PUBREC message field



PUBREC message field	Description / Values
Message ID	The message ID of the PUBLISH message to be acknowledged.

- **PUBREL message format**

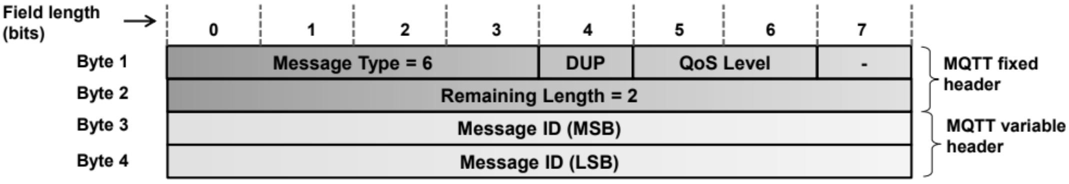
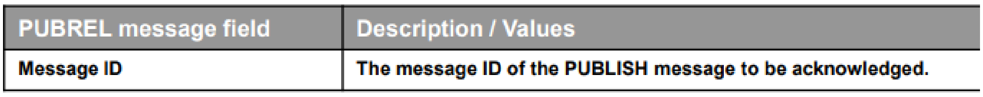


Figure 2.14 PUBREL message format

Table 2.8 Overview of PUBREL message field



PUBREL message field	Description / Values
Message ID	The message ID of the PUBLISH message to be acknowledged.

• PUBCOMP message format

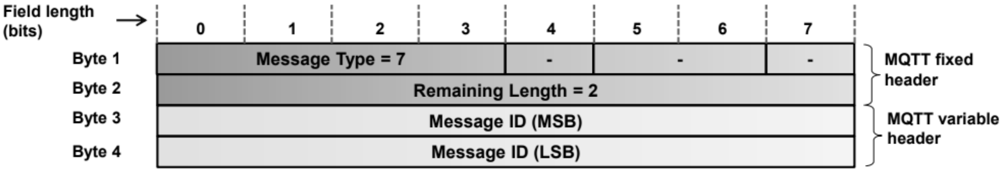


Figure 2.15 PUBCOMP message format

Table 2.9 Overview of PUBCOMP message field

PUBCOMP message field	Description / Values
Message ID	The message ID of the PUBLISH message to be acknowledged.

• SUBSCRIBE message format

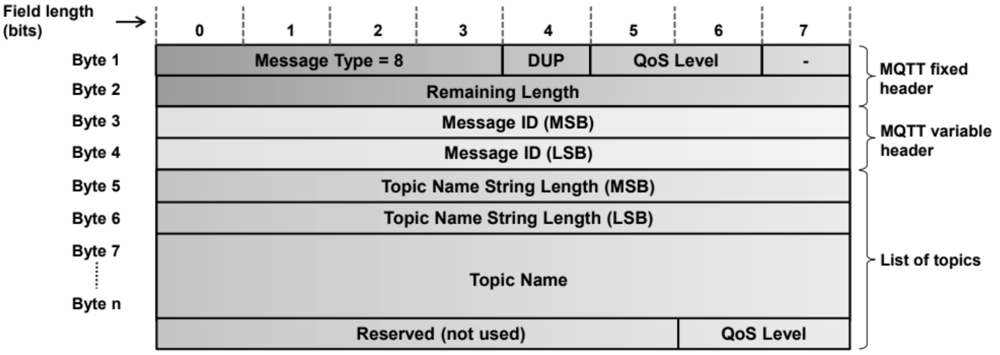


Figure 2.16 SUBSCRIBE message format

Table 2.10 Overview of SUBSCRIBE message fields

SUBSCRIBE message field	Description / Values
Message ID	The message ID field is used for acknowledgment of the SUBSCRIBE message since these have a QoS level of 1.
Topic Name with Topic Name String Length	Name of topic to which the client subscribes. The first 2 bytes of the topic name field indicate the topic name string length. Topic name strings can contain wildcard characters Multiple topic names along with their requested QoS level may appear in a SUBSCRIBE message.
QoS Level	QoS level at which the clients wants to receive messages from the given topic.

• SUBACK message format

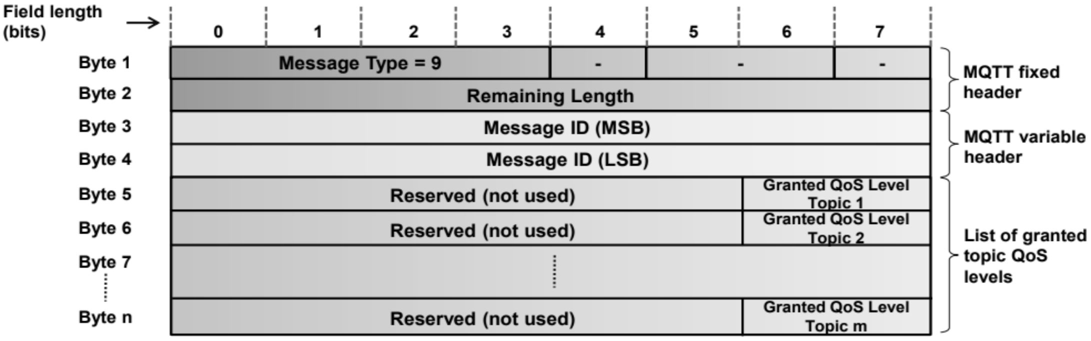
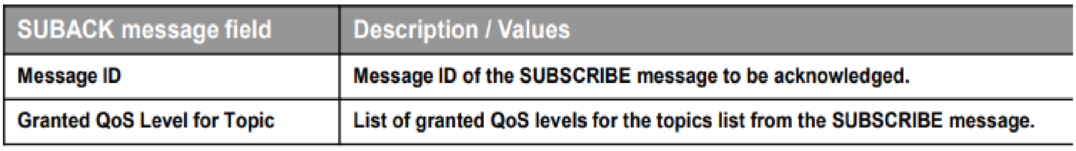


Figure 2.17 SUBACK message format

Table 2.11 Overview of SUBACK message fields



SUBACK message field	Description / Values
Message ID	Message ID of the SUBSCRIBE message to be acknowledged.
Granted QoS Level for Topic	List of granted QoS levels for the topics list from the SUBSCRIBE message.

• UNSUBSCRIBE message format

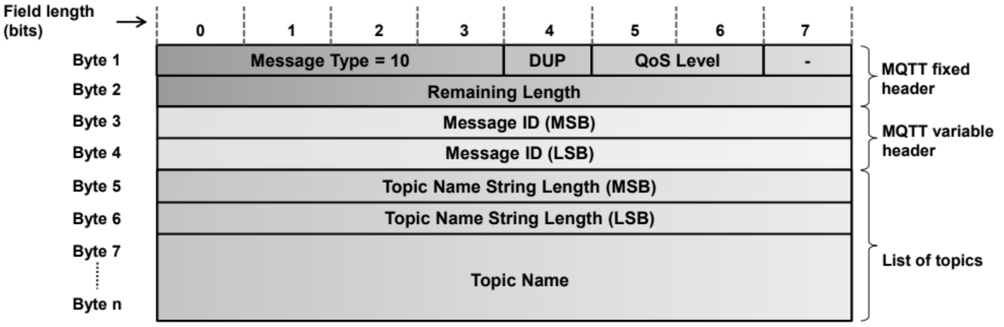
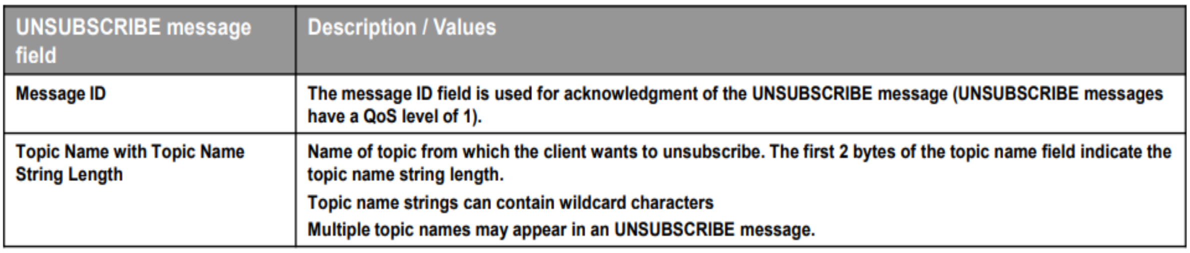


Figure 2.18 UNSUBSCRIBE message format

Table 2.12 Overview of UNSUBSCRIBE message fields



UNSUBSCRIBE message field	Description / Values
Message ID	The message ID field is used for acknowledgment of the UNSUBSCRIBE message (UNSUBSCRIBE messages have a QoS level of 1).
Topic Name with Topic Name String Length	Name of topic from which the client wants to unsubscribe. The first 2 bytes of the topic name field indicate the topic name string length. Topic name strings can contain wildcard characters Multiple topic names may appear in an UNSUBSCRIBE message.

• UNSUBACK message format

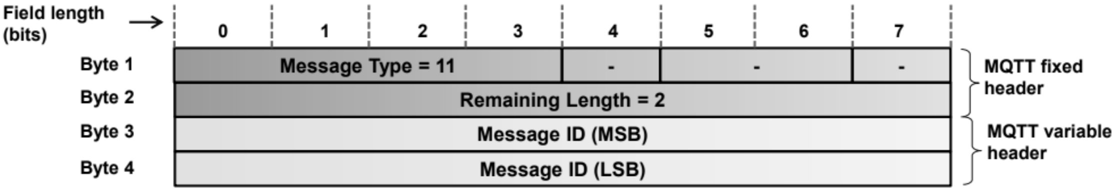
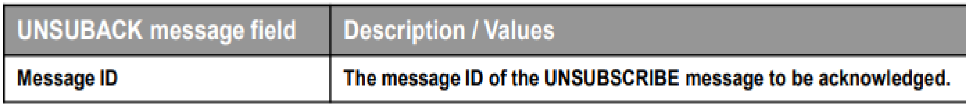


Figure 2.19 UNSUBACK message format

Table 2.13 Overview of UNSUBACK message field



UNSUBACK message field	Description / Values
Message ID	The message ID of the UNSUBSCRIBE message to be acknowledged.

• DISCONNECT, PINGREQ, PINGRESP message formats

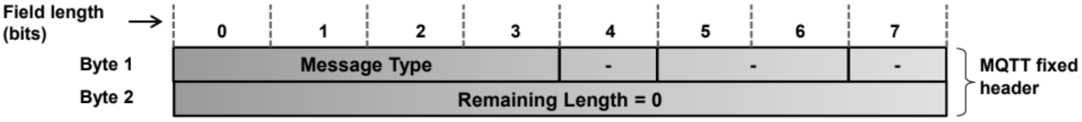


Figure 2.20 DISCONNECT, PINGREQ, PINGRESP message formats

2.2.7.6 MQTT Quality of Service

MQTT provides the typical delivery quality of service (QoS) levels of message-oriented middleware. Even though TCP/IP provides guaranteed data delivery, data loss can still occur if a TCP connection breaks down and messages in transit are lost. Therefore, MQTT adds 3 quality of service levels on top of TCP: [24]

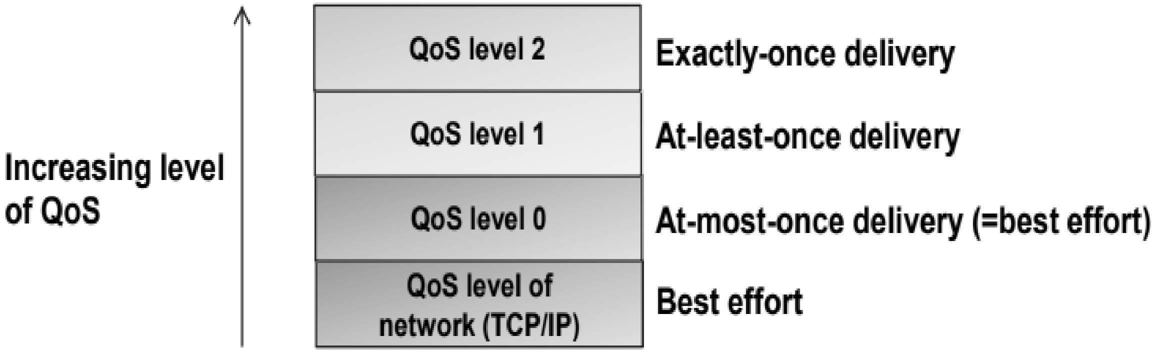


Figure 2.21 MQTT QoS

- **QoS level 0** At-most-once delivery (best effort)
  Messages are delivered according to the delivery guarantees of the underlying network (TCP/IP).

  Example application: Temperature sensor data which is regularly published. Loss of an individual value is not critical since applications (consumers of the data) will anyway integrate the values over time and loss of individual samples is not relevant.

- **QoS level 1** At-least-once delivery
  Messages are guaranteed to arrive, but there may be duplicates. Example application: A door sensor senses the door state. It is important that door state changes (closed->open, open->closed) are published losslessly to subscribers (e.g. alarming function). Applications simply discard duplicate messages by evaluating the message ID field.
- **QoS level 2** Exactly-once delivery.
  This is the highest level that also incurs most overhead in terms of control messages and the need for locally storing the messages. Exactly-once is a combination of at-least-once and at-most-once delivery guarantee.

  Example application: Applications where duplicate events could lead to incorrect actions, e.g. sounding an alarm as a reaction to an event received by a message.

2.2.7.7 CONNECT and SUBSCRIBE message sequence

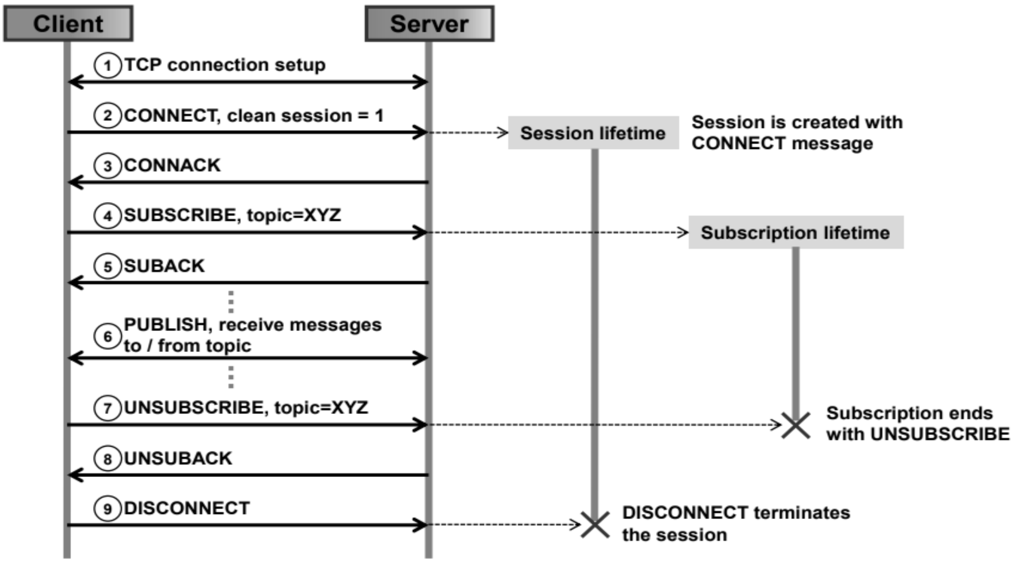

Client
Server
1 TCP connection setup
2 CONNECT, clean session = 1
Session lifetime
Session is created with CONNECT message
3 CONNACK
4 SUBSCRIBE, topic=XYZ
Subscription lifetime
5 SUBACK
6 PUBLISH, receive messages to / from topic
7 UNSUBSCRIBE, topic=XYZ
Subscription ends with UNSUBSCRIBE
8 UNSUBACK
9 DISCONNECT
DISCONNECT terminates the session

Figure 2.22 CONNECT and SUBSCRIBE message sequence[24]

2.2.7.8 Keep alive timer, breath of live with PINGREQ

The keep alive timer defines the maximum allowable time interval between client messages. The timer is used by the server to check client's connection status. After 1.5 * keepalive-time is elapsed, the server disconnects the client (client is granted a grace period of an additional 0.5 keepalive-time). In the absence of data to be sent, the client sends a PINGREQ message instead. Typical value for keepalive timer is couple of minutes. MQTT broker will acknowledge with the PINGRESP response. [24]

2.2.7.9 MQTT Will message

In case of an unexpected client disconnect, depending applications (subscribers) do not receive any notification of the client's demise. MQTT solution: Client can specify a will message along with a will QoS and will retain flag in the CONNECT message payload. If the client unexpectedly disconnects, the server sends the will message on behalf of the client to all subscribers (last will). [24]

2.2.7.10 Topic Wildcards

Subscribers are often interested in a great number of topics. Individually subscribing to each named topic is time- and resource-consuming. MQTT solution: Topics can be hierarchically organized through wildcards with path-type topic strings and the wildcard characters '+' and '#'. Subscribers can subscribe for an entire sub-tree of topics thus receiving messages published to any of the sub-tree's nodes. [24]

2.2.7.11 MQTT-SN

The Message Queuing Telemetry Transport for Sensor Network (MQTT-SN) protocol was developed specially for Wireless Sensor Networks (WSNs), normally made up of low cost and easy developed environments. Typically, a WSN has a large number of sensors and actuators, from different device types, that present a limited amount of storage and processing capabilities.

The propose of the MQTT-SN is to be a publish/subscribe protocol for wireless sensor network, considering a version of MQTT that best applies to the particularities of wireless communication environment. Furthermore, it is optimized for implementation on low-cost, battery-operated devices with limited processing and storage resource. The MQTT-SN was developed to not rely on network services. In addition, the MQTT-SN can be supported for any network which provides a bi-directional data exchange service between any node and a particular broker or gateway. [25]

Some features of the MQTT-SN are:

- To work with short message length and limited transmission bandwidth in a wireless network, the "topic name" in PUBLISH message can be replaced for a two-byte “topic-id”. The clients register their topic name in the server/gateway and obtain the corresponding topic id;
- “Pre-defined” topic ids and “short” topic names are introduced. The short topics presents a length of two bytes, so they are shorter enough for being carried simultaneously with data in PUBLISH messages;
- The discovery procedure is used to assist clients that do not know the server/gateway’s address to discover the network address;
- A new offline keep-alive procedure is defined to support sleeping clients. The battery-operated devices can go to a sleeping state, all messages designed to them are buffered at the server/gateway and delivered later when they wake up.

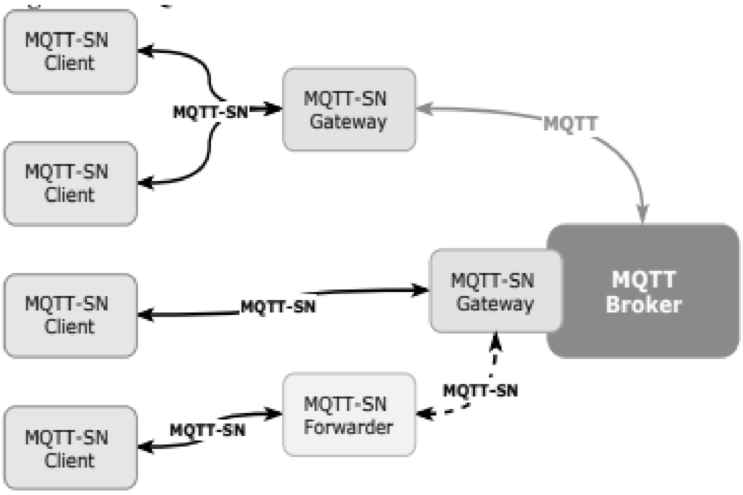


Figure 2.23 MQTT-SN architecture

As it is illustrated in Figure 2.23, the classic architecture is composed of the MQTT-SN client (publisher), the MQTT-SN Gateway (GW), and the MQTTSN forwarder. The MQTT-SN protocol transfer messages between an MQTT-SN client and a broker using an MQTT-SN GW as the middleware. The function of an MQTT-SN GW is to translate messages from MQTT to MQTT-SN or vice-versa, if it is a stand-alone topology. The MQTT-SN Forwarder is configured when the MQTT-SN GW is not present in the same network. To access a MQTT-SN GW that is not attached in the same network topology is used a MQTT-SN forwarder. The forwarder encapsulates the MQTT-SN frames when receives them from the gateway and sends to the clients. There are two types of GWs to translate frames (make a syntax translation) between MQTT and MQTT-SN, and vice versa. They are:

- A transparent gateway where each MQTT-SN connection has a corresponding MQTT connection. This is the easiest type to implement.
- An aggregating gateway where multiple MQTT-SN connections share a single MQTT connection.[26]

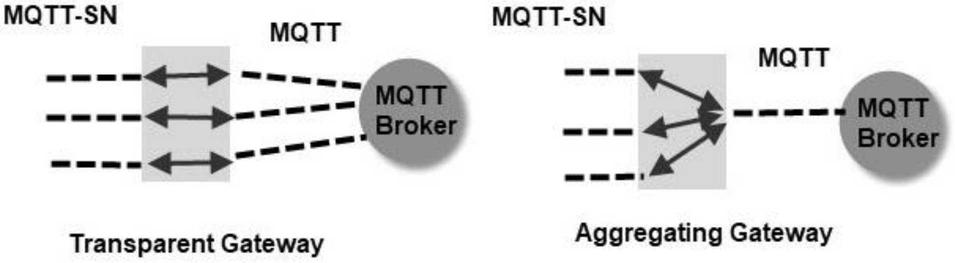


Figure 2.24 MQTT-SN Gateways

2.3 E-Healthcare

2.3.1 Background

Healthcare is the efforts made to maintain or restore physical, mental, or emotional well-being especially by trained and licensed professionals.[27]

Healthcare systems are able to deliver prompt and efficient services to patients not only in medical intensive care units in hospitals but also in their homes and even workplaces, which in turn is cost-effective and improvises the patient's life quality. The impending health crisis attracts researchers, organizations, and scientists to search for optimal and best health solutions. A term called "eHealth" evolved where healthcare was supported via electronic processes, and now healthcare is extended to becoming mobile known as mHealth. In order to fully utilize and optimize wireless technologies, a new type of network has evolved termed as Wireless Body Area Network (WBAN).[28]

2.3.2 Telemedicine

The rise of healthcare gave birth to an innovative concept of practicing medicine and clinical services from a distance, using telecommunication, named as telemedicine. Telemedicine enables doctors to serve patients remotely by accessing health information through a telecommunication link. It has been a great relief for people in the rural and remote areas where there is a limited medical facility including doctors and infrastructure.

Figure 2.25 presents a general architecture of a telemedicine system. A patient requiring medical attention approaches the nearby local health center where a local health professional (may not be a certified doctor) attends the patient and does the primary health check-up. This unit consists of basic diagnostic equipment and tele-consultation devices linked via PC and Internet to the city hospital. The primary responsibility of the local healthcare unit is to acquire all the vital statistics of the patient in terms of physiological data (e.g., blood, urine, etc.) and images (e.g., ultrasound) and transmits the data to the remote city hospital. After receiving the records, the remote medical practitioner goes through every detail, before proceedings with live Interaction with patients. After carefully examining the basic vital signs, the meeting is booked online between doctor and patient at remote healthcare unit. The doctor makes use of an audio or video conferencing system as well as automation live feeds to have live interaction with the patient. These remote hospitals are connected to a centralized database where all the data of the patient as well as other details and even the recorded audio/video interaction between doctor and patient are also stored. The stored information can be accessed using mobile apps or web-based interface. The main hospitals are also linked to specialist hospitals to provide specialized support to the patients in case of an emergency and these specialized hospitals have same teleconferencing units enabled to support remote patients.[29]

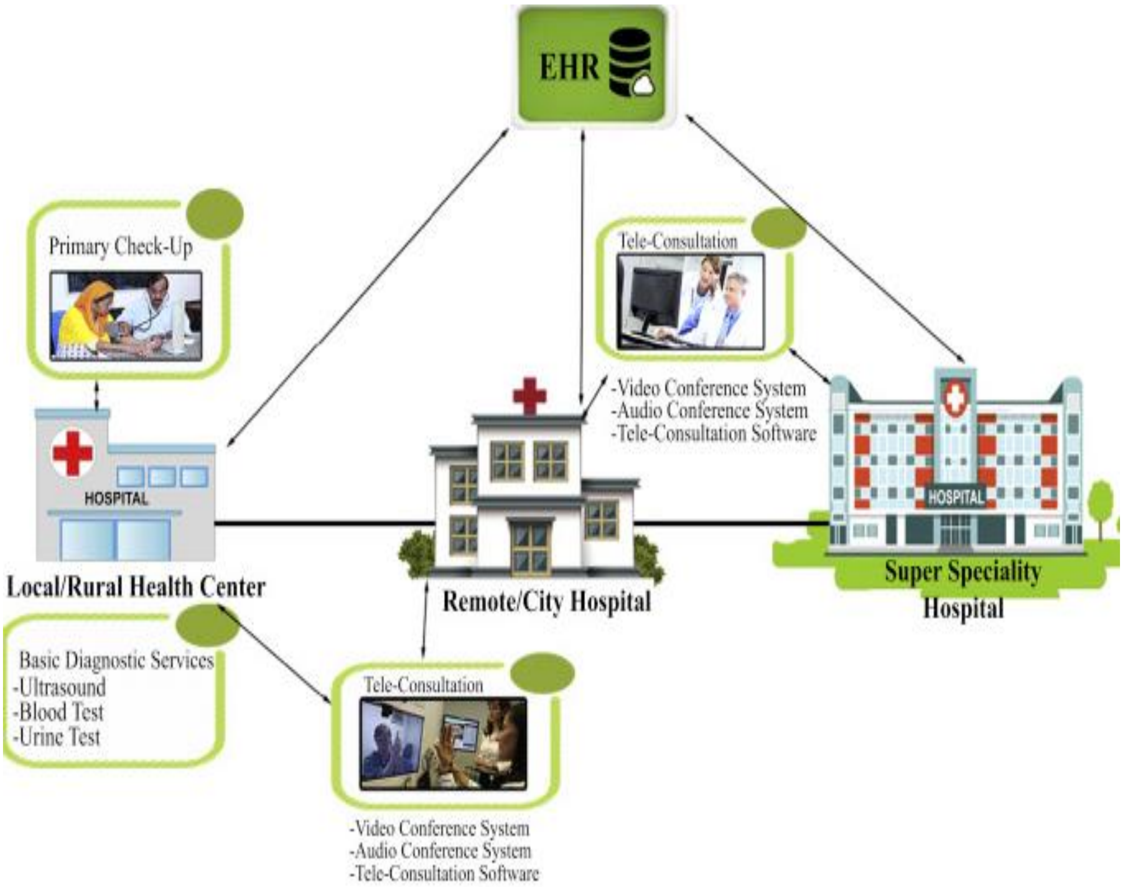


Figure 2.25 A general telemedicine system

2.3.3 Remote Health Monitoring (RHM)

Is a part of remote healthcare or e-healthcare. It is an approach for automated health monitoring from anywhere. This has been possible thanks to the advancement in sensors and WSN and other newer technologies such as WBAN and IoT. Different health sensors planted within and on the body sense different physiological data like body temperature, heart and pulse readings, blood pressure, blood sugar, brainwave, the oxygen level in blood, etc. These data are sent to the concerned health professionals who interpret them for assessing a patient's health status, diagnosis, and medication or treatment recommendation. On analyzing the data, it might be decided whether a preventative therapeutic intervention is required or the patient's prescription is needed to be changed. If the monitoring devices are connected to the internet directly, doctors can monitor patients in real-time. Furthermore, integrating the remote monitoring system with sophisticated analytical tools provides physicians with greater visibility and insights into a patient's health status. [28]

2.3.4 Telemedicine Vs Remote Health Monitoring

There are no universal definitions for the terms "telemedicine" and "remote health monitoring". Consequently, there are no agreed and specific differentiating factors between the two. Both refer to the exchange of medical information electronically between two sites. Both aim to help patients remotely. That is why people often use the terms interchangeably. Probably, the involvement of remote operation in both cases is the reason for the misperception.

But, as a matter of fact, they are quite different approaches to e-healthcare. In fact, remote health monitoring is the key differentiating factor between traditional telemedicine and today's e-healthcare which comprises both telemedicine and remote health monitoring. [28]

2.3.5 Benefits of remote health monitoring

Remote health monitoring exhibits several benefits as follows: [28]

- Remote health monitoring ensures delivering constant and quality care to patients in remote locations.
- Healthcare becomes more available. Remote monitoring allows doctors to reach out to potential patients especially to those people who can't afford to visit a doctor or not been able to go to the hospitals, for certain reasons.
- Offers better quality of life, improved mobility and decrease the mortality rate for the unprivileged populating in terms of healthcare services.
- Faster access to relevant patient data enables quick treatment initiation and also shortens treatment duration.
- Monitored data are automatedly fed into the expert systems and advanced data analysis tools for better insight into the patient's health.
- Automated monitoring reduces the chance of erroneous diagnosis and incorrect treatment.

2.3.6 Architecture of Remote Health Monitoring Using WBAN

A Wireless Body Area Network WBAN is a special type of WSN that is associated with the human body, where fewer sensor nodes are deployed on the patient’s body as compared to traditional WSNs. The primary requirement of any WBAN is tiny, low-power, wearable or implantable sensor node. A WBAN comprises of a set of heterogeneous sensors and medical devices performing individual roles to perform monitoring of patient’s health.

WBAN applications range from medical to non-medical applications.

Non-medical applications comprise motion and various gestures for detecting one’s fitness, social interactions and even medical assistance in varied situations like floods, earthquakes, fires, etc.

Medical applications include healthcare-based applications. Depending on situations, varied sensors are available either wearable or implantable to transmit physiological signals. Generally, the following body sensors are used for health monitoring (typically termed as biosensors):

- **As wearable sensors** there are: Electrocardiographic (ECG) sensor for monitoring the electrical energy produced during a heartbeat, heart rate sensor, Electromyography (EMG) sensor which used to measure all patient’s body muscle’s electrical activity, Body temperature sensor, Photoplethysmography often implemented using a pulse sensor or pulse oximeter and is used to measure non-invasive heart rate and also the blood oxygen levels.
- **As Implantable sensors** which have very small size and compatible with human tissue and can withstand all physical forces inside the human body. Examples of implantable sensors are: Pacemaker which placed at the chest or abdomen to control all sorts of abnormal heart rhythms, deep brain stimulator for treating all types of movement and neuropsychiatric disorders.[28]

Figure 2.26 [30] represents the general architecture of a remote health monitoring system. The primary component of a remote healthcare unit is the WBAN which collects patient's health-related data which are sent through the network gateway to a dedicated medical server from where the doctor gets access of the patient's health record and recommend treatment.[29]

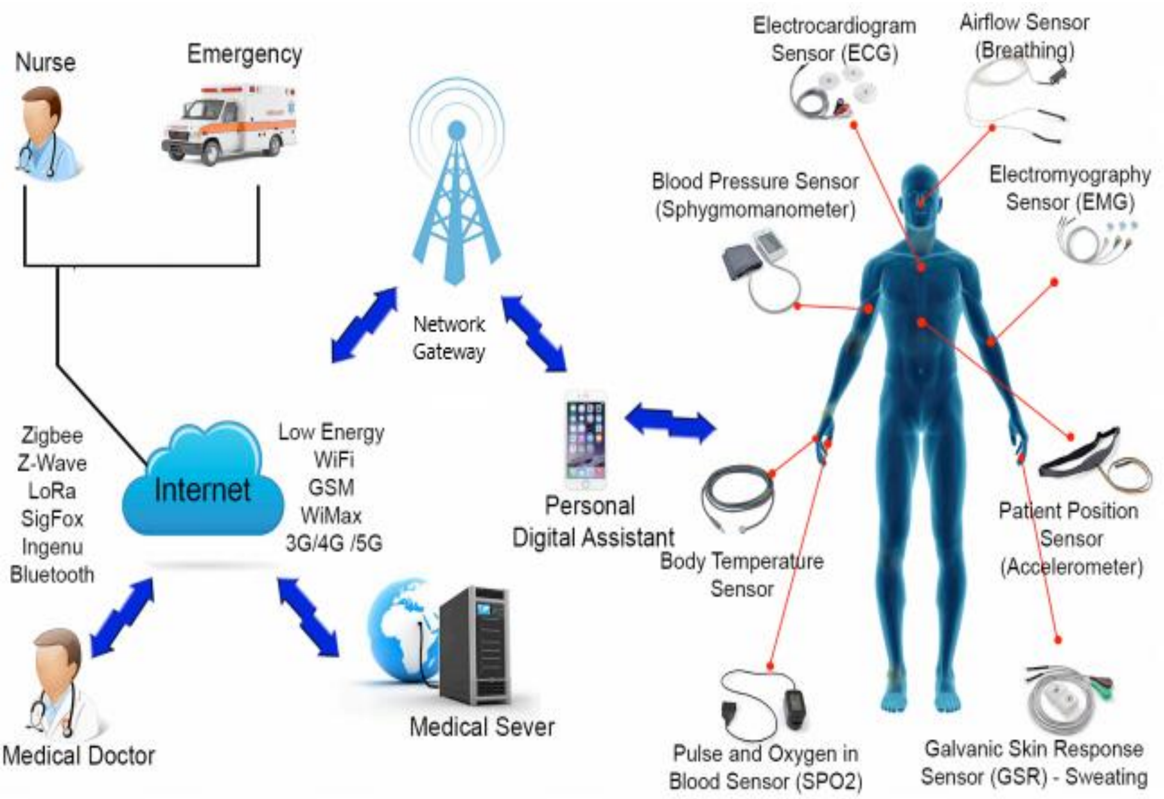

Nurse
Emergency
Network Gateway
Electrocardiogram Sensor (ECG)
Airflow Sensor (Breathing)
Blood Pressure Sensor (Sphygmomanometer)
Electromyography Sensor (EMG)
Zigbee
Z-Wave
LoRa
SigFox
Ingenu
Bluetooth
Internet
Low Energy
WiFi
GSM
WiMax
3G/4G /5G
Personal Digital Assistant
Body Temperature Sensor
Patient Position Sensor (Accelerometer)
Medical Doctor
Medical Sever
Pulse and Oxygen in Blood Sensor (SPO2)
Galvanic Skin Response Sensor (GSR) - Sweating

Figure 2.26 Illustration of an architecture for remote healthcare monitoring system

2.3.7 **Major Drawbacks of E-Healthcare Systems**

Although, there are many benefits that are drawn with the implementation of e-healthcare framework, still there are certain drawbacks too. The primary challenges or limitations of using an e-healthcare system are elaborated in the section below:[31]

- The procurement cost is too high which is why most often organizations do not prefer to implement this system.
- It requires uninterrupted internet connection on both the sides for effective exchange of information. Therefore, any lack of internet connection may result in inconvenience for its usability.
- It is already known that a patient's record will be stored onto an electronic database, therefore it imposes a huge threat to the privacy of a patient's medical information because the online systems are highly vulnerable to network breaches such as cyber assaults.
- The information contained in the electronic database system needs to be regularly updated. This is because of the fact that decisions are evaluated on the basis of the stored medical record. Therefore, it might raise certain health issues in case some wrong information is updated or go missing.

2.3.8 Most common challenges of E-Health

E-Health promises to be a cost-effective and efficient way of providing healthcare at an affordable cost to patients who would otherwise be excluded or underserviced. However, e-health also comes with a series of ethical and legal challenges which, if not met before its implementation, could undermine its success.[32]

2.3.8.1 Technical Reliability and Appropriateness

There are challenges that are rooted in the technology itself. Device safety and standardization are obviously important issues; as are the technology's ability to ensure data integrity and reliability and its power to gather and communicate data accurately with appropriate back-up measures to guard against malfunction or interruption.

2.3.8.2 Privacy

Privacy issues also acquire a new dimension. By and large, national laws and international conventions stipulate that healthcare professionals and institutions have a duty to protect the confidentiality of patient data to the best of their ability, and that breaches in this regard should be communicated to the subjects of the data in due time and in an appropriate manner.

2.4 Implementation of MQTT in e-healthcare application

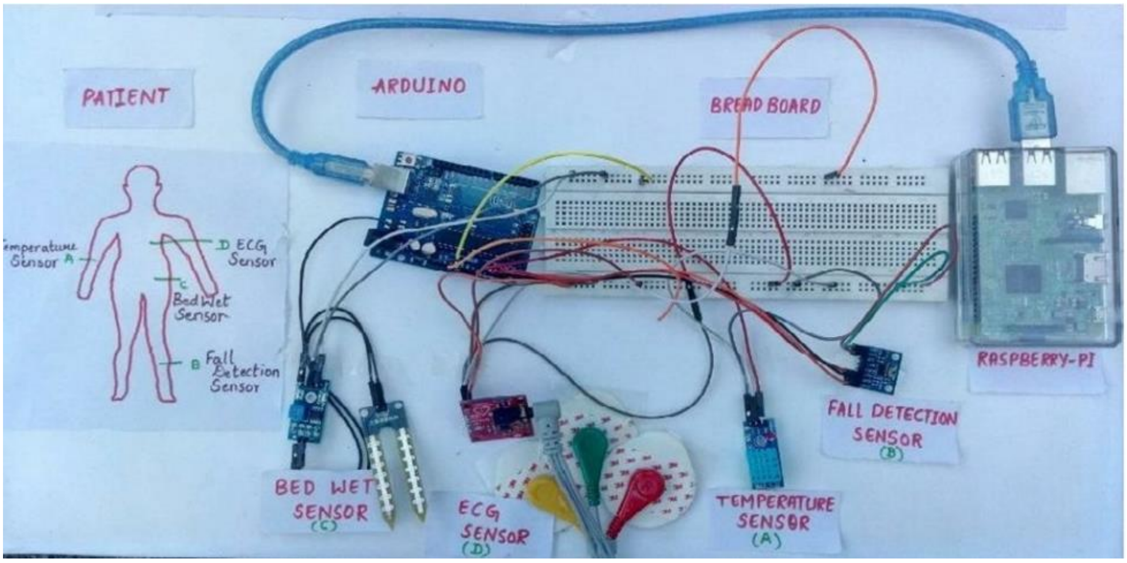


Figure 2.27 MQTT Based Patient Health Monitoring

The MQTT broker in this system (Figure 2.27) is the Mosquito software which is an open source implementation of an MQTT broker. In this system [33] RPi 3 acts like MQTT broker. The RPi 3 sends an alert message to the multi-user (Doctor or nurses) mobile (Using MQTT

Dashboard Application) whenever patients vital data crosses the threshold values.

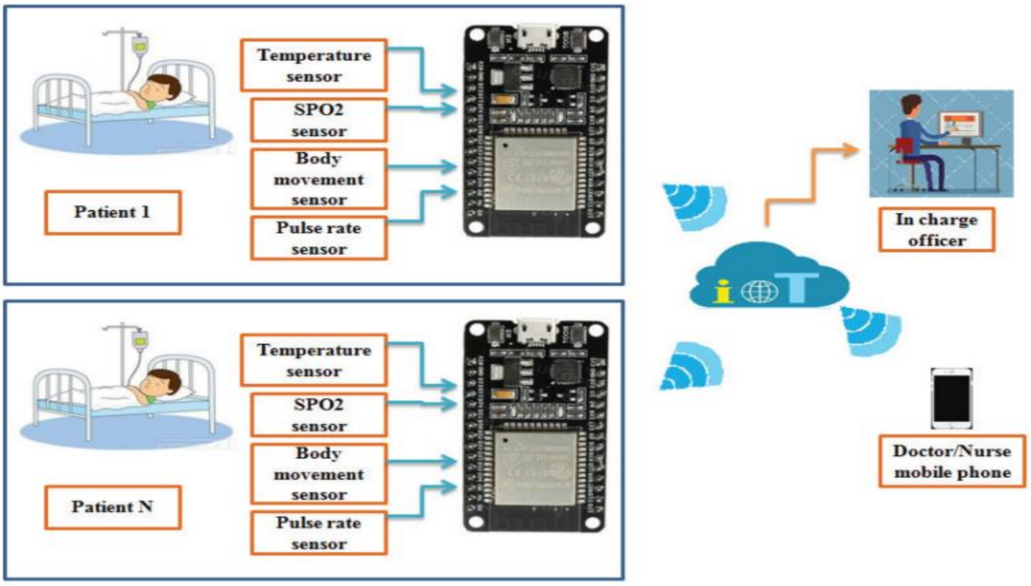


Figure 2.28 Smart Healthcare Monitoring System Using MQTT Protocol

The Figure 2.28 in [34] shows the system architecture of the smart health care system using ESP32 and MQTT protocol. The smart health care system is designed using the sensors like pulse rate sensor,spo2 sensor(hemoglobin containing oxygen), temperature sensor, body movement sensor or Accelerometer. All these sensors are connected to the Esp32 Microcontroller that supports Wi-Fi/Bluetooth and connected to the MQTT server, this server is used for displaying sensed data in a webpage or mobile app.

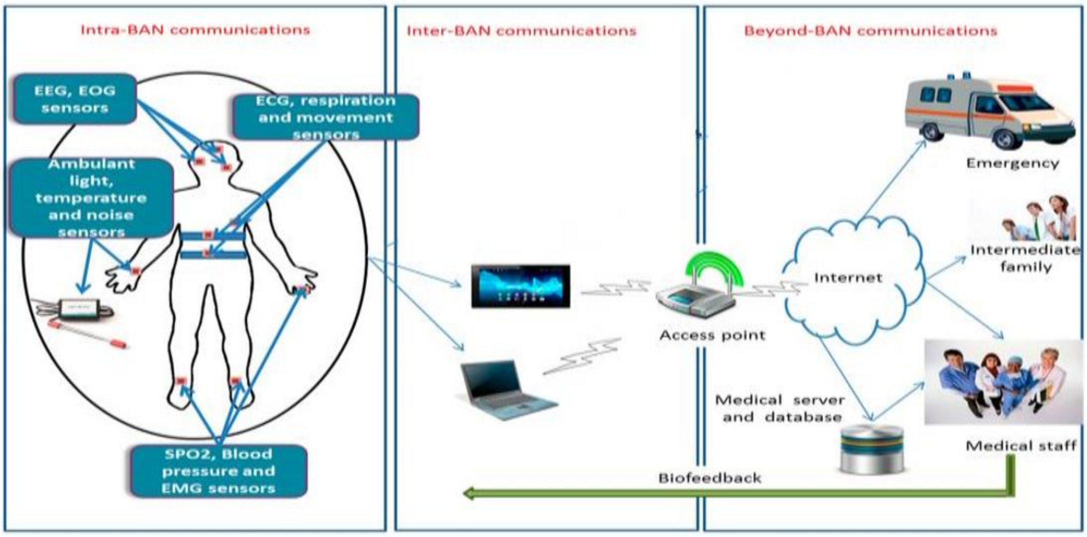

Intra-BAN communications
Inter-BAN communications
Beyond-BAN communications
EEG, EOG sensors
ECG, respiration and movement sensors
Ambulant light, temperature and noise sensors
SPO2, Blood pressure and EMG sensors
Access point
Internet
Emergency
Intermediate family
Medical server and database
Medical staff
Biofeedback

Figure 2.29 IoT Healthcare network

The proposed design for health monitoring in [35] consists of telemedicine system with different line as shown in the figure 2.29. The Medical Broker is developed to serve hundreds or more end users. The sensor nodes which are either implanted or worn on the body are designed to privately sample vital signs and transfer the sampled signs (data) through a wireless personal network implemented using ZigBee (IEEE802.15.4). The sensor nodes on the patient's body publish the information to a specific topic in the Broker, then the subscribes subscribe to these topics and keep track of patient's status.

2.5 Conclusion

In this chapter, we have thoroughly presented publish/subscribe communications with a projection on IoT area, especially the IoT-connected e-Health application. Among all publish/subscribe protocols, we have focused on MQTT protocol because of its various advantages that we already presented. We mention its lightweight and low power consumed which is necessary in healthcare applications.

Overall, privacy and security are an ongoing effort for MQTT, and probably the most important one since MQTT is one of the most widely adopted and mature communication protocol solutions. Solving the privacy issue would create an important and big advantage for MQTT, in comparison with other available solutions.

Chapter three

Privacy issues and solutions for publish/subscribe-based communications

3.1 Introduction

From the users' perspective, it is evident that lot of benefits exist for the users for being part of e-health systems. It is practical to consider that any person will probably seek the help of different care services and care providers during his or her lifetime. Given the fact that a patient's health record is available electronically and can be shared between different care providers, it would definitely help the patients to convey information regarding previous consultations and diagnosis while also helping the healthcare professionals to make better judgments with the help of comprehensive nature of supportive electronic documents.[36]

Among various messaging protocols for IoT, MQTT is widely used protocol being simple, easy to implement and having small code footprint, but it lacks default security features.

In this chapter, we will target privacy issues and focus on presenting existing solutions based on e-healthcare and on MQTT protocol. Finally, we will compare the studied research work.

3.2 Privacy issues and solutions

3.2.1 E-healthcare

Considering the fact that one's health data probably be distributed among several care providers; it is necessary to incorporate secure and efficient mechanisms to achieve the intended goals. In reality, lot of incidents can be found where e-health systems become found wanting due to the miss-management and privacy violations of sensitive health data. Privacy Rights Clearinghouse (PRC) provides significant evidence for the above fact, where they have reported more than 22 million healthcare related privacy violations even with the existence of regulations for security and privacy of health information. Moreover, further statistics reveal the privacy violations by providing examples where Personal Health Information (PHI) is stolen or acquired without the authorization of legally obliged parties. Thus, it is quite clear that sophisticated security and privacy measures must be adopted in order to experience the benefits of e-health systems while assuring users regarding the safety of their private data.[37]

Privacy in eHealth systems has attracted much research effort. For this, a variety of different privacy-enabling methods have been proposed. This section provides a brief overview of previous work on privacy in eHealth, focusing on patient privacy.

The importance of patient privacy in eHealth is traditionally seen as vital to establishing a good doctor-patient relationship. This is even more pertinent with the emergence of the Electronic Patient Record (EPR) [38]. Cryptography is a necessary tool for privacy in eHealth systems [39].

In this work [40] the authors proposed a patient-centric cloud based EHR system that is encrypted by the patient using a symmetric key and stored in a cloud platform. The metadata file (attribute-based access policy and the location information) is encrypted using broadcast CP-ABE. When a doctor wants to access user EHR, he needs to send a request to the owner while encrypting it with the public key of the data owner and wait for an acknowledgment to access these data. This scheme has a higher computational cost on patient end due to the patient centric nature of the solution, where all encryption of data and updating of access policies are handled by the patient side.

In this work [41] they proposed a privacy preserving chaos-based encryption cryptosystem for patients' privacy protection. The proposed cryptosystem can protect patient's images from a compromised broker. In particular, they propose a fast-probabilistic cryptosystem to secure medical keyframes. The encrypted images produced by their cryptosystem exhibits randomness behavior, which guarantee computational efficiency as well as a highest level of security for the keyframes against various attacks including differential, statistical, and exhaustive attacks for finding secret keys. Furthermore, it processes the medical data without leaking any information, thus preserving patient's privacy by allowing only authorized users for decryption. The experimental results verify the excellent performance of their encryption cryptosystem compared to other systems.

In this work [42] the authors proposed a new efficient and privacy-preserving pre-clinical guidance scheme (hereafter referred to as PGuide) for mobile e-Healthcare (Figure 3.1), designed to offer both self-diagnosis and hospital recommendation services to users in a privacy-preserving way. To provide users the capability to present a detailed health profile for accurate disease risk prediction, they introduce a Privacy-Preserving Comparison Protocol (PPCP) in PGuide, which will improve the accuracy of disease risk prediction. They also employ a single-attribute encryption technique to devise a privacy-preserving hospital recommendation service in PGuide, which can further guide users to choose a hospital appropriate for their visit after conducting a self-diagnosis. They demonstrate the efficiency of PGuide, in terms of computational cost and communication overhead.

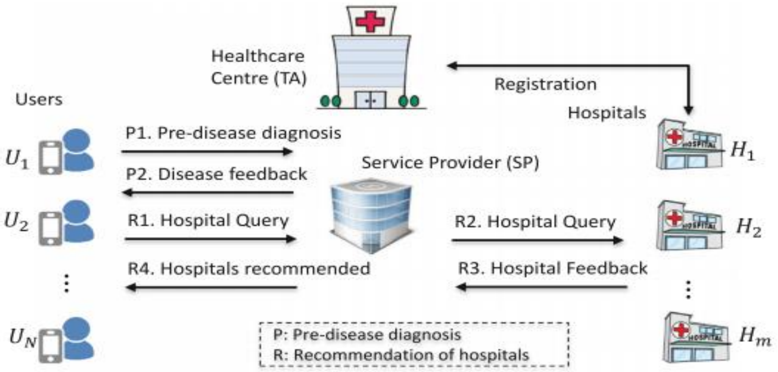


Figure 3.1 Proposed PGuide scheme

3.2.2 MQTT

Even though MQTT is used everywhere, it was not designed with security in mind. According to a survey conducted by the IoT developer community, MQTT is the 2nd most popular IoT messaging protocol [43]. The stupendous growth of MQTT successively requires a private secure version and the alleviation process should consider the resource constraint nature of the device.

Confidentiality is indispensable for people to safeguard their sensitive, private information from unauthorized access . It is nearly tantamount to privacy which means that only the authorized individuals/system should be able to view sensitive or classified information which means that illegitimate access shouldn't be allowed. [44]

In this work [45] the authors proposed a Secure MQTT (SMQTT) which augments security feature for the existing MQTT. In this protocol (Figure 3.2), a new publish service 'Spublish' is proposed which uses message type '0000', wherein the message is encrypted using Attribute Based Encryption (ABE) based on lightweight Elliptic Curve Cryptography (ECC) that is generated by the broker Private Key Generator (PKG). As a result, Subscribers who satisfy the access policy are capable of decrypting the message. They evaluated and compared performance in terms of the time taken to perform encryption, decryption, key generation and validation against number of attributes with different key sizes (256, 512 bits), Encryption and decryption time for CP-ABE is more when compared to KP-ABE scheme, since additional information needs to be computed and, provided in the access policy. The proposal is efficient, robust and scalable.

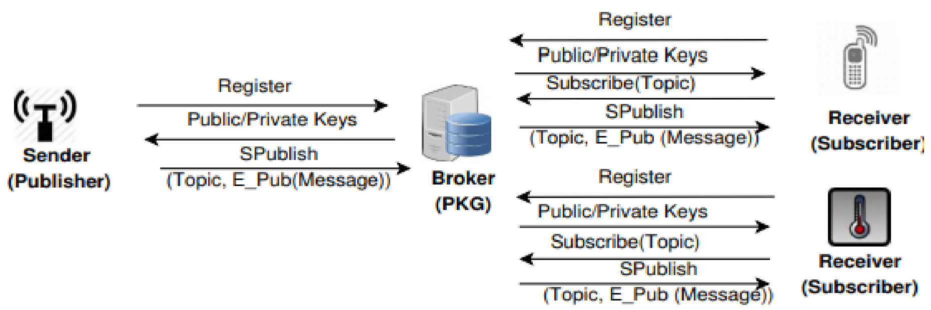


Figure 3.2 The proposed SMQTT

In this work [46] the authors proposed a security mechanism that safeguards MQTT payload of sensed data in IoT domain assumes that there is a Trusted System (TS) that generate IDs and cryptographic keys for publishers and subscribers. TS is trusted by all nodes and responsible for managing trust associations among IoT nodes which is done by sharing a Pre-Shared Secret (PSS) during an initial registration phase. The proposed solution (Figure 3.3) uses a Cryptographically Secure Random Number Generator (CSRNG) algorithm to generate cryptographic keys and asymmetric cryptosystem for encrypting MQTT payload. Their solution uses PCG-Rand as CSRNG because of its difficult predictability rate, excellent statistical quality, and efficiency in terms of time and code space. The system is resilient to likely attacks such as brute force attack and spoofing attack and requires less memory space. Comparative analysis for energy consumption shows an increase of 2.71% node energy consumption in their proposed system than in unencrypted mode which is affordable for IoT nodes.

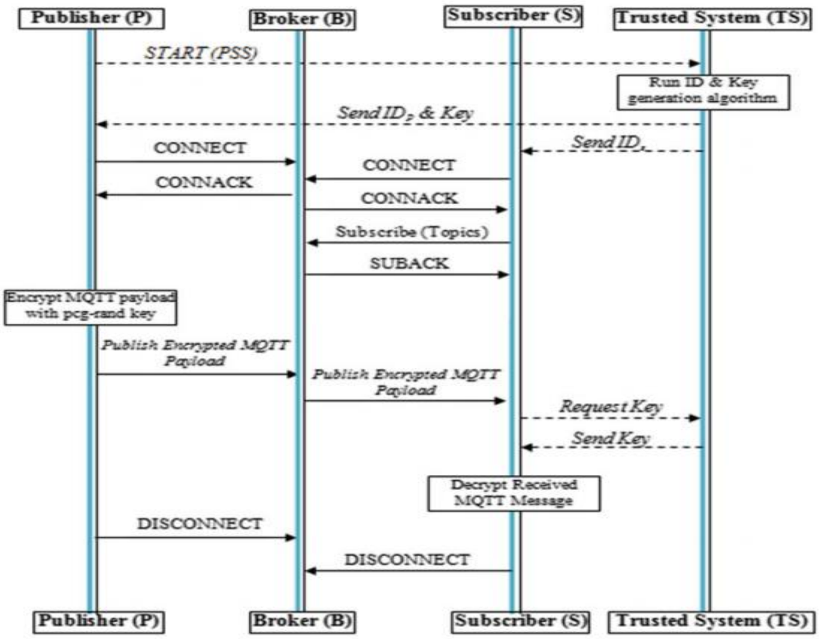
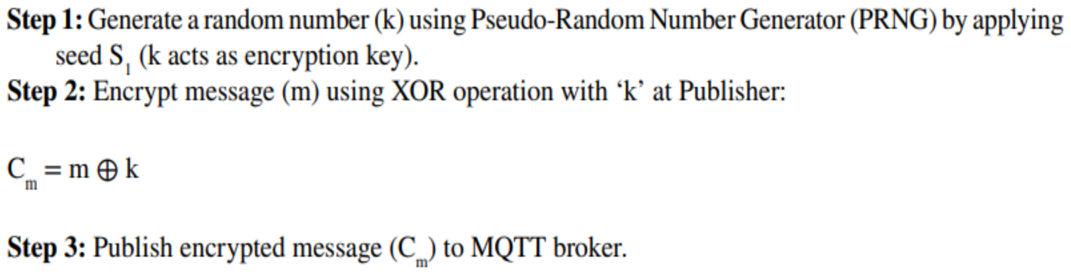


Figure 3.3 Message flow diagram of the proposal system

In this work [47] the authors proposed a framework where published Electronic Patient Records (EPR) messages are encrypted by LWEA algorithm (Figure 3.4) and are decrypted by subscribers only, the broker being unable to decipher messages as it does not have keys used for decryption. In this framework, classification of IoT application scenarios based on publisher's location is performed. For example, when the patient (publisher) and medical consultant (subscriber) are both inside hospital, then EPR as MQTT payload need to traverse over the local network, and when patient is at home and sensed data needs to travel over the internet to reach hospital server. A LWEA is designed and implemented to secure message confidentiality in the local network. A highly secure cryptographic protocol (AES/CTR-128) is implemented to secure messages travelling over the internet. Their result shows an increase of 1.14% node energy consumption than in unencrypted mode which is affordable by smart biosensor devices. Besides its resistance against brute force attack, spoofing attack and also provides end-to-end encryption for communicating messages.



Step 1: Generate a random number (k) using Pseudo-Random Number Generator (PRNG) by applying seed S_1 (k acts as encryption key).

Step 2: Encrypt message (m) using XOR operation with 'k' at Publisher:

$$C_m = m \oplus k$$

Step 3: Publish encrypted message (C_m) to MQTT broker.

Figure 3.4 Encryption Algorithm-Light-Weight Encryption Algorithm (LWEA)

3.3 Comparison between proposed solutions

Table 3.1 Comparison between the proposed works

Solution	[40]	[41]	[42]
Contribution	Patient-centric cloud based EHR system that incorporates symmetric key cryptography, public key cryptography and CP-ABE	fast-probabilistic cryptosystem to secure medical keyframes that are extracted from compromised broker.	Pre-clinical guidance (PGuide) scheme mobile e-Healthcare, designed to offer self-diagnosis and hospital recommendation services
Limitation	- User can only use all possible combinations of attributes in a single set issued in their keys to satisfy policies. -Storage and time requirements	-Privacy preservation based on patient's images -Time requirements	- Restricted amount of storage as well as the size of the screen - Decrease the accuracy of clicks. This is also known as the "fat finger problem". -Slow Hardware
Efficiency	Average	High	Average
Complexity	High	Average	Average
Confidentiality	Yes	Yes	Yes
Scalability	No	Yes	Yes
Attack resistant	Yes	Yes	No
Energy consumption	High	High	Medium

<table>
<tr><th>Solution</th><th colspan="2">[45]</th><th>[46]</th><th>[47]</th></tr>
<tr><td><b>Contribution</b></td><td colspan="2">SMQTT uses Spublish command to publish an encrypted message using CP/KP-ABE</td><td>Add a 4th party (TS) in order to generate IDs and cryptographic keys by PCG-Rand and send it to the publisher crypted by the shared key PSS
XOR is simple logical invertible operation used as an encryption operation</td><td>Framework where published Electronic Patient Records (EPR) messages are encrypted by LWEA algorithm and are decrypted by subscribers only</td></tr>
<tr><td rowspan="2"><b>Limitation</b></td><td colspan="2">-KPABE cannot decide who can encrypt data. -CPABE decrypt key only.
Time requirement</td><td rowspan="2">- The need to use another component instead of the existing one</td><td rowspan="2">-end-to-end encryption</td></tr>
<tr><td>KPABE</td><td>CPABE</td></tr>
<tr><td><b>Efficiency</b></td><td colspan="2">Average</td><td>High</td><td>High</td></tr>
<tr><td><b>Complexity</b></td><td>High</td><td>Average</td><td>Average</td><td>Low</td></tr>
<tr><td><b>Confidentiality</b></td><td>No</td><td>Yes</td><td>Yes</td><td>Yes</td></tr>
<tr><td><b>Scalability</b></td><td colspan="2">No</td><td>Yes</td><td>Yes</td></tr>
<tr><td><b>Attack resistant</b></td><td colspan="2">Yes</td><td>Yes</td><td>Yes</td></tr>
<tr><td><b>Energy consumption</b></td><td>Low</td><td>High</td><td>Medium</td><td>High</td></tr>
</table>

3.4 Conclusion

The privacy issue becomes much more critical in data publication since it is important in eHealth systems, keep it in mind when designing the architecture of such systems like MQTT protocol is a promising path towards ensuring privacy.

Despite MQTT's meteoric rise to fame, it did not prevent to be flawless. Thus, many innovations have been done to improve its security and privacy levels as long as its properties remain. mentioning its lightweight in order to be an effective solution.

Chapter four

Overview on our solution

4.1 Introduction

We have witnessed the improvement of the quality of life of older people especially through intelligent environments by technological advancements in e-healthcare.

However, such applications require a certain level of privacy to be efficient and secure.

Patient privacy is a must, and the last thing that could happened is a data breach, a compromise of patient information, or a penalty for failing to meet compliance.

In this chapter, we will explore the needs of privacy in e-healthcare, otherwise, explain our problematic. Then, we will take MQTT protocol as a base of communication and propose a betterment in term of privacy.

4.2 Problematic

If the configuration of MQTT is insecure, then the entire environment is compromised. However, with e-healthcare applications, the problem is exacerbated.

Indeed, publishers may send sensitive publications which include their identity along with the content of their medical files; location, topics and more personal data. Thus, malicious entities could reveal these data. The leakage of such information can lead to serious consequences.

Our concern is the communication between publisher of data and the broker. Therefore, we would just assume that WBAN sensors are already collected patient's vital signs to send it to his smartphone, laptop, smartwatch or any device (via Bluetooth, BLE, ZigBee, Wi-Fi…) that would publish these gathered data via MQTT protocol. To avoid confusion of concepts, we would facilitate the scenario by considering the publisher device as the patient.

As a network model, publisher is represented as a patient who publishes medical data values (e.g. 90/60mmHg, 1g/L, 37.2°C..) in the appropriate topic (e.g. Blood Pressure, Blood Glucose, temperature etc.) via Publish message where these data would be stored and routed through the broker to the endpoints ; dedicated subscribers (doctors, caregivers, hospitals..) in one condition: these publications satisfy subscribers 'interests for consultation or other purposes.

In another words, these published data would be forwarded as notifications to the doctors who subscribed to the same topic that data published in.

Assuming that the broker is a trust worthy, our focus is to preserve data publication privacy exactly in Publish phase which is illustrate in (Figure 4.1)

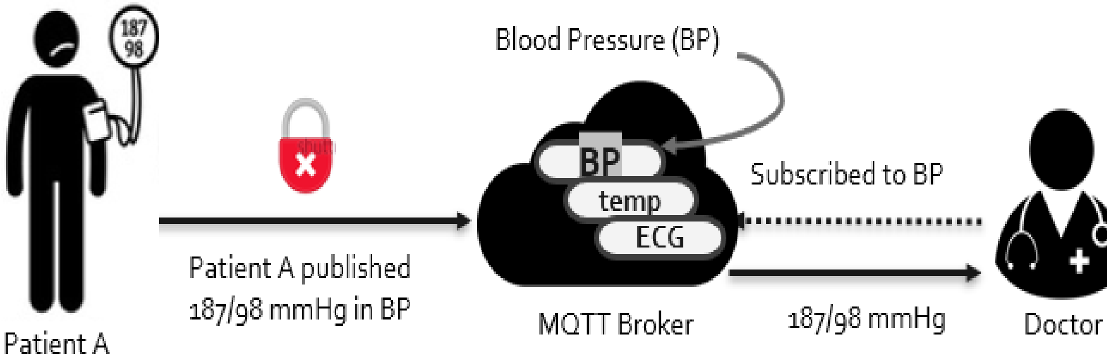


Figure 4.1 Privacy issue in MQTT

N.B MQTT topics are very lightweight. The client does not need to create the desired topic before they publish or subscribe to it. The broker accepts each valid topic without any prior initialization.

4.3 Privacy context of our solution

4.3.1 Description

Since patient's information can be compromised, hacked, or sniffed out by adversaries during the communication between publisher and broker, our solution prevents curious harmful entities to harvest the published data that would only be accessed by the authorized entity which is the broker.

To do so, we planned to firstly, hide publisher ID by merge it with a random number generator (RNG). Besides, topic name by a corresponding anonymous name in matching table exists in both publisher and broker. Then, implement Lightweight encryption algorithm for encrypting these anonymous topics before their publication to be decrypted through broker. That encryption algorithm consists of random number generator as a symmetric shared key between both entities (publisher/broker) for the encryption of topics using XOR operation with that key. We briefly illustrate our solution for tackling this issue in figure 4.2

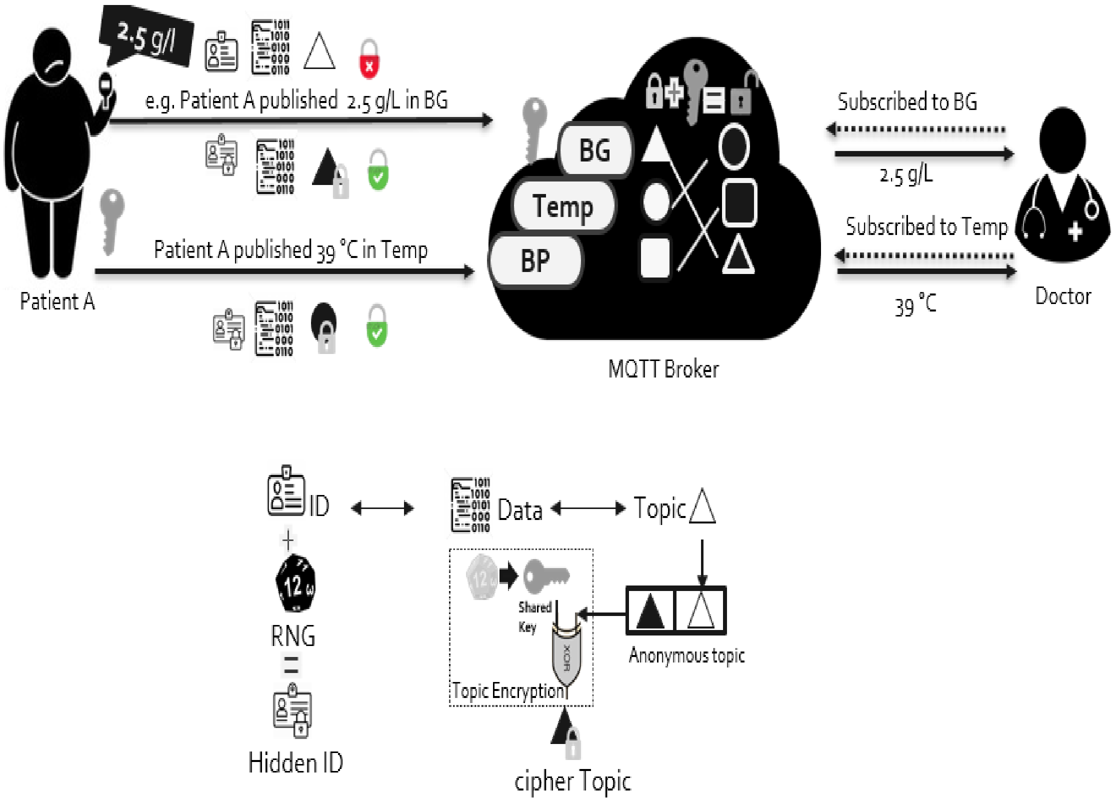


Figure 4.2 Description of our solution

4.3.2 **Justification of used algorithms**

Light weigh mechanisms are more suitable to maintain the most important feature of MQTT "the lightness". Furthermore, they are low computational cost, obviously simple and effective.

4.3.3 **Modelling of our solution**

Figure 4.3 shows the general interaction model of our approach.

Firstly, the patient (publisher) connect to broker via his fake ID. After the broker Ack, he registers to topics. In the other side, doctor(subscriber) open a session too to the broker with connect, than subscribe to all topics.

Our work in the publisher, where he uses a matching table of anonymous topics to encrypt them before publishing each topic with its msg (data). The broker decrypts the received topics to send them to the doctor.

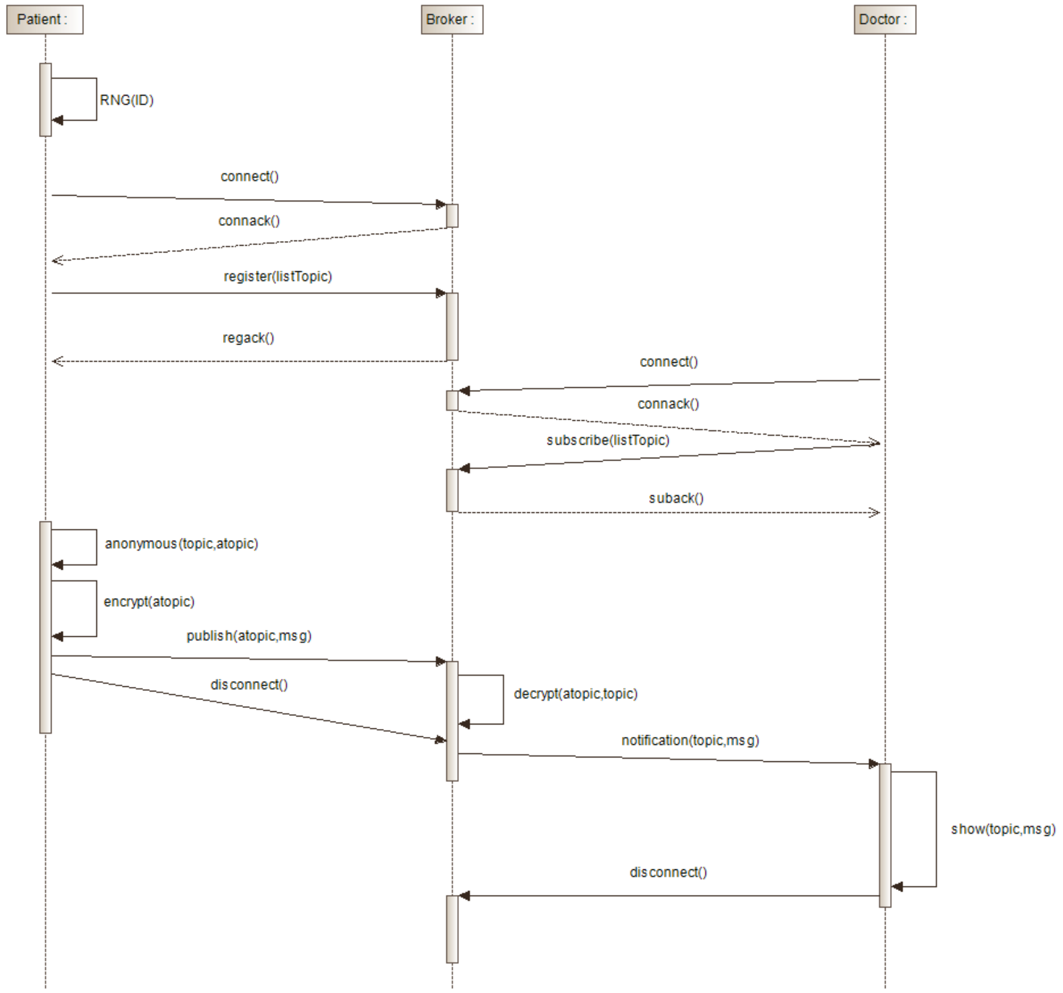


Figure 4.3 Sequence diagram of our approach

4.4 Implementation of the proposed system

In this section we describe the context of the implementation of the MQTT modified version protocol as well as the main steps followed.

4.4.1 The context of the implementation

The language with which we have implemented the proposal protocol is the C language which is considered as the mother language of all the modern programming languages. It is characterized by its powerful general-purpose and extremely popular, simple and flexible. It is machine-independent, structured programming language which is used extensively in various applications; to develop software like operating systems, compilers, and so on. This choice allowed us to implement the protocol within the Cooja network simulator, which we will present it in the next chapter. The platform of our work is the Contiki which is an open source operating system for connecting the tiny low-cost microcontrollers, sensors to the internet. Contiki is preferred because it supports the various internet standards, rapid development, selection of hardware. Contiki can be downloaded as "Instant Contiki", that is an Ubuntu Linux virtual machine that runs in VMWare player.

4.4.2 The main steps of the implementation

The process followed for the implementation of our proposal is summarized in all of the following steps:

4.4.2.1 Installation of Cooja simulator

- Download Instant Contiki, which is a virtual machine created with all necessary toolchains and software for ContikiOS development.
- Unzip the downloaded file.
- Start the virtualization software like VMware or VirtualBox and load the Instant Contiki File. In VMware, click Open a Virtual Machine
- Navigate to the extracted folder .vmx file.
- Power on your machine.
- Select "I copied it" option in VMware Workstation.
- Enter your credentials and start using Instant Contiki (Username and Password is user).
- Start Cooja (Contiki network simulator).
- Open terminal window and go to Cooja directory by `cd Contiki/tools/cooja`
- Start Cooja by `ant run`
- Run Contiki in simulation (Whether run a new simulation or existed one).

4.4.2.2 Implementation of privacy mechanisms in MQTT-SN

In order to apply privacy for publish/subscribe communications in e-healthcare, we have worked at the application layer. we used the MQTT protocol in which we introduced 3 levels of privacy; the 1st level applied on publisher ID and the 2 levels left on the topics as an anonymity and encryption phases in order to preserve confidentiality and privacy of patient publications according to the mentioned mechanisms.

4.5 Conclusion

In this chapter, we have described how patient privacy can be achieved by hiding both of his ID and the topic he published into through RNG and corresponding anonymous topic list as a primary focus then its encryption. We took the MQTT protocol as a test case and application of our proposal. The resulting performances will be evaluated by simulation in the next chapter.

Chapter five

Assessment and results

5.1 Introduction

In the field of networks, each time a new solution emerges and before putting it into practice, it must first go through phases of tests, evaluations and validation, on a conceptual level, according to a simulation model in order to save both cost and time.

In this chapter we use the Cooja simulator to evaluate our proposed approach, according to well-defined parameters. The results obtained will be analyzed and interpreted.

5.2 Overview of the simulator Cooja

5.2.1 Definition

Cooja is a simulator of Contiki operating system developed for simulations of sensor nodes. Each node in the simulated network can be different not only concerning its installed software but also the hardware platform may vary. The applications for Contiki are written in C and the simulations are written in Java.[48]

5.2.2 Cooja interface

Cooja network simulator interface comprises of 5 windows: [49]

5.2.2.1 Network shows the location of each node in the network. Can be used to visualize the status of each node, including LEDs, mote IDs, addresses, etc. Initially this window is empty and we need to populate it with our sensors.

5.2.2.2 Simulation control this panel is used to Start, Pause, Reload or execute Steps of the simulation. It shows the time of execution and the speed of simulation. It means that we can run the events several times faster than it would take in real-time execution.

5.2.2.3 Notes this is a simple notepad for taking notes about the simulation.

5.2.2.4Timeline simulation timeline where messages and events such as channel change, LEDs change, log outputs, etc are shown.

5.2.2.5 Mote output shows all output of serial interface of the nodes. It is possible to enable one window of Mote output for each node in the simulation.

In addition to the default tools, it is possible to exhibit other tools such as Breakpoints, Radio messages, Script editor, Buffer view and Mote duty cycle, which can be enable in the Tools menu.

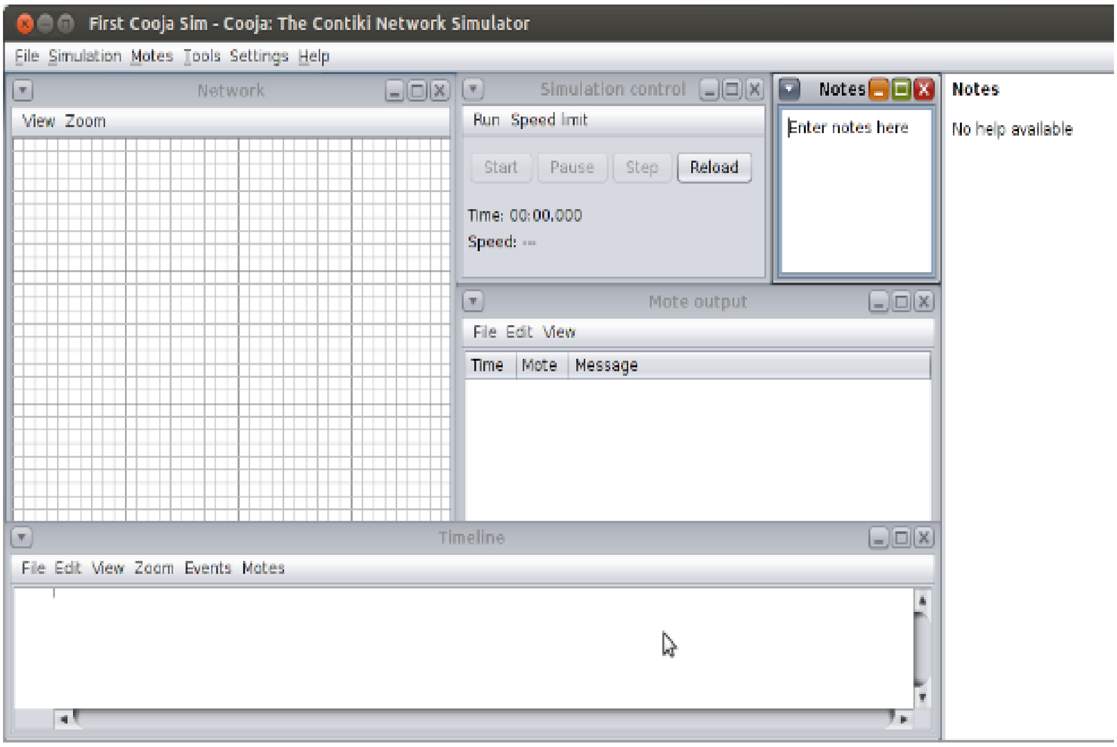


Figure 5.1 Different windows of Cooja network simulator

5.3 Justification for the choice of the simulator

The following points summarizing the arguments justifying the choice of the Cooja simulator for the evaluation of the proposed approach:

- It is a flexible simulator and many parts may be replaced or extended.
- Works in different levels.
- Free tool.
- Allows the emulation (imitates behavior closely) of the hardware of a set of sensor nodes.
- Networks can be tested before being printed into hardware.

5.4 Evaluation of performances

In this part we list the details of the configuration and parameters used for the simulations performed to evaluate and compare the performance of the MQTT-SN and proposed system.

5.4.1 Simulation setup

Implementation of the proposed system is performed in the Contiki (Figure 5.1). 5 sensors were deployed in this field and an RPL Border Router node was placed in the center of the field as a gateway between publisher node and RSMB broker. The 2 publisher nodes (patients) were placed one in the right and another on the bottom. The 2 subscribers (Doctors) one to the left and the last on the top.

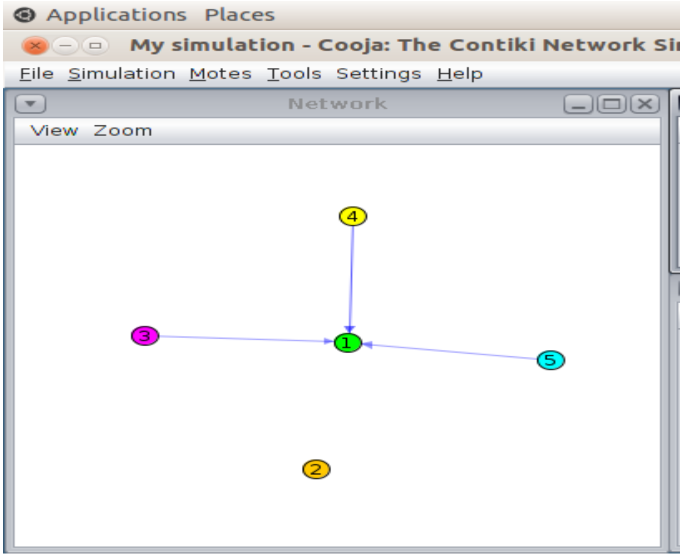

Applications Places
My simulation - Cooja: The Contiki Network Si
File Simulation Motes Tools Settings Help
Network
View Zoom
4
3
1
5
2

Figure 5.2 Simulation setup

To link the border router to the broker, we used 2 terminals:

1st: The directory: home/user/contiki/examples/ipv6/rpl-border-router

The command: make connect-router-cooja

2nd: The directory: contiki/mqtt-sn/tools/mosquito.rsmb/rsmb/src

The command: sudo ./broker_mqtts config.mqtt

The parameters that were used in the simulation are given in Table 5.1.

Table 5.1 Simulation parameters

Parameter	Value
Mote Type	Zolertia
Number of motes	5 motes
Application Layer	MQTT-SN
MAC layer	802.15.4
Simulation time	Variant
Node Transmission range	20m
Node interference range	25m
Tx/Rx radio	100 %

5.4.2 Performance Analysis

Using the evaluation environment cited above, the performance metrics measured are:

5.4.2.1 Energy consumption

Energy efficiency is the key requirement to maximize sensor node lifetime. Sensor nodes are typically powered by a battery source that has finite lifetime. Most Internet of Thing (IoT) applications require sensor nodes to operate reliably for an extended period of time. Here, we are interested to measure the energy of patient side by the following formula:

Energy (mJ) =[[(LPM $_{time}$ * 0.545 mA) + (CPU $_{time}$ * 1.8 mA) + (TX $_{time}$ * 17.7 mA) + (LISTEN $_{time}$ * 20mA)] * 3 v] / 32768 Eq (1)

5.4.2.2 Delay

The delay is the lag of time which can be calculated by the difference between publication time and the reception time in subscriber side.

Delay (s) = reception of notification $_{time}$ (s) – publication $_{time}$ (s) Eq (2)

5.4.2.3 Matching accuracy

Written as a percentage, this is a term to describe the level of correctness of received notifications or the error rate. And can be calculated by dividing the number of right notifications by the number of publications and multiply by 100 %.

Matching accuracy (%) = [number of right received notifications / number of publications] *100 Eq (3)

5.4.3 Results and discussion

The results of the comparison between MQTT-SN with and without privacy feature in terms of time and energy consumption led to the following curves:

5.4.3.1 Energy consumption

The energy consumption rate measured over a time interval of 5 minutes gave the following results:

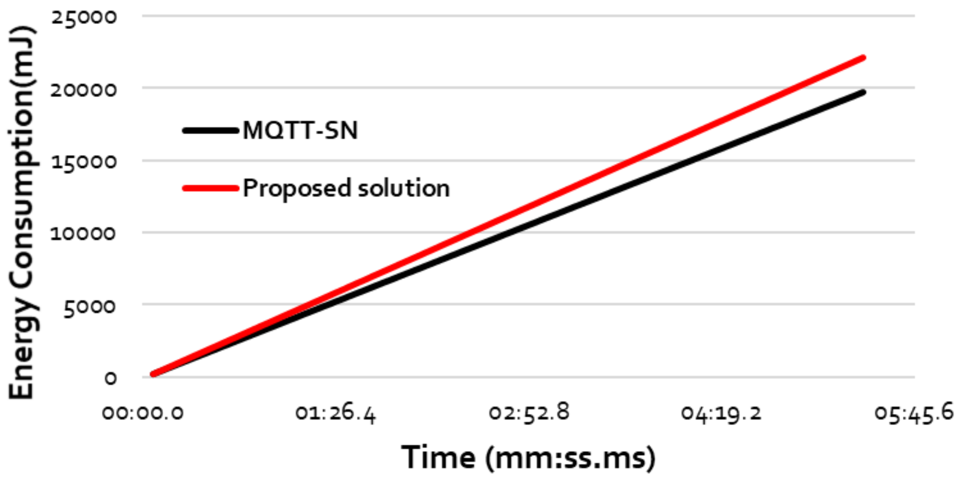


Figure 5.3 Energy consumption comparison

Using Eq (1) and the values obtained from power trace tool, we have calculated the energy consumption of our proposed system and compared that with MQTT-SN original version for the same simulation setup. From the plot in Figure 5.3 it is analyzed that the proposed Privacy system consumes acceptable energy compared to the system without privacy in place.

5.4.3.2 Average delays

The curves presented the delay which is measured over a time interval of 5 minutes:

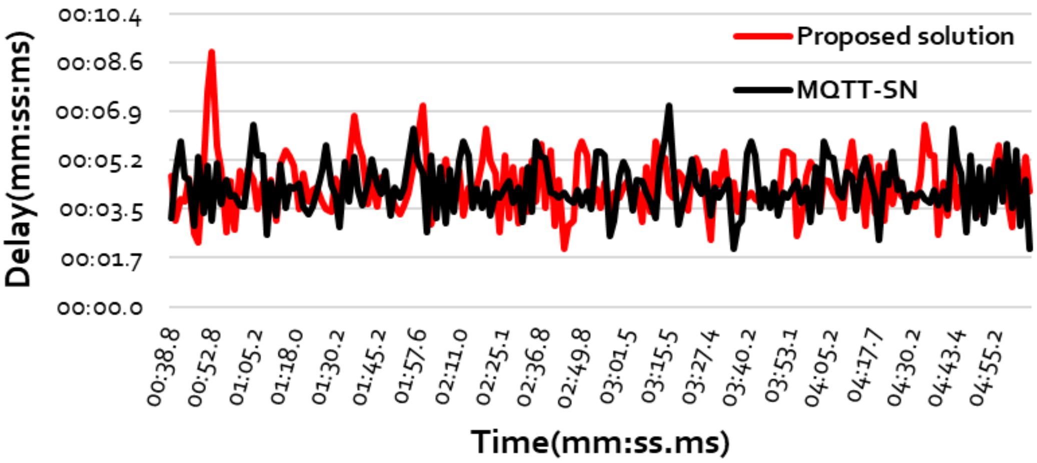


Figure 5.4 Delay comparison

Using Eq (2) and the values obtained from power trace tool. The Figure 5.4 shows that the delay of the proposed system is close enough to the MQTT-SN.

5.4.3.3 Matching accuracy

The presented results are the matching accuracy which is measured over a time interval of 16 minutes:

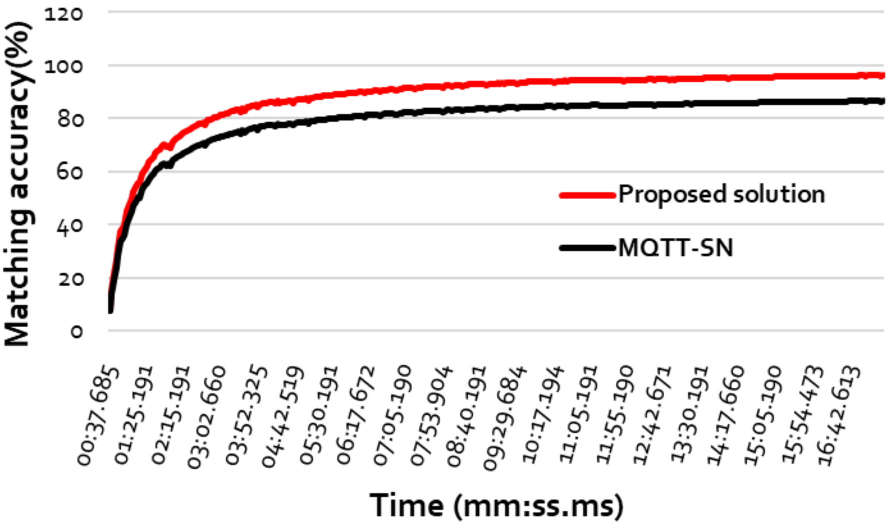


Figure 5.5 Matching accuracy comparison

Using Eq (3) and the values obtained from power trace tool. From the plot in Figure 5.5, we notice that whenever the time increase, the matching accuracy does. That happened in both protocols but ours took less time to approach more.

By observing the simulation results regarding the matching accuracy accomplished by each protocol, we found that our proposal performed better than the MQTT-SN.

Unlike end-to-end delay and the energy of publisher, MQTT-SN was more suitable. Even if the change was not significant and has respected the limits which have been set for it (not to exceed the granted threshold) and therefore remains acceptable for a certain data rate.

5.5 Conclusion

The simulation results demonstrated the performance of our solution in achieving the trade-off between privacy-ensuring and good quality of service.

Such a result remains satisfactory and makes it possible to meet the constraints of the medical applications to which we aimed to adapt that approach (IoT nodes will not affect their lifetime by a substantial amount). The resulting proposed system maintains the privacy of the patients and manages to reduce the accuracy.

General conclusion

During this work, we introduced internet of things by highlighting its essential characteristics, architecture, enabling technologies, protocols, application domains as well as its main challenges.

E-healthcare is a critical domain application with restricted time, energy. because of its biosensors, that lead us to think carefully about its constraints.

MQTT protocol as a Publish/subscribe system proved its worth in communication through its decoupled nature, lightweight. MQTT-SN (MQTT for Sensor networks) was designed specifically to work on wireless networks, and, as far as possible, to work in the same way as MQTT. It uses the same publish/subscribe model and can be considered as a version of MQTT. It is also optimized for the implementation on low-cost, battery-operated devices with limited processing and storage resources which would be suitable for medical applications but it lacks default privacy features.

In developing our work, we faced quite uncounted obstacles, especially in handling the simulator. This comes down to the lack of documentation which can be explained by this field of research. Moreover, MSP430 microcontrollers, which are designed to run with small amounts of memory prevented us from using many libraries that might could help to enhance that work.

Without forgetting the Corona pandemic, which negatively affected our termination, as it targeted many of our family, friends, and from all over the world. No one was spared from it, but Alhamdulillah for everything.

Finally, we were able to carry out an adaptation of a MQTT-SN protocol aiming to guarantee the privacy of published data by hiding publisher ID and integrating an anonymity for the published topics besides applying a lightweight encryption algorithm to these topics to be decrypted via the broker. The resulting protocol has shown its performance in increasing the matching accuracy while keeping energy and delay within agreed limits.

Our future concern is to continue in this direction and find the means to extend more while keeping the delays/energy within the allowed threshold. Additionally, integrate the WBAN to form a complete platform of e-healthcare.

Bibliography

[1] Dave Evans, *The Internet of things: how the next evolution of the internet is changing everything*, Cisco Internet Business Solutions Group, April 2011

[2] S.Sahraoui, *Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things)*, thèse de doctorat, Université de Batna, 2016

[3] Marco Zennaro, *Intro to Internet of Things*, ITU ASP COE TRAINING ON "Developing the ICT ecosystem to harness IoTs, 13-15 December 2016, pp9.

[4] Keyur K Patel, Sunil M Patel PG Scholar Assistant Professor, *Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges*, Volume 6 Issue No. 5, International Journal of Engineering Science and Computing (IJESC)2016, pp2

[5] Kavya Chandrashekhar, *Internet of Things Characteristics*, Business Analyst at Ellucian, published on September 19, 2016, Accessed on August 2020, [Online]. Available: https://www.linkedin.com/pulse/internet-things-iot-characteristics-kavyashree-g-c/

[6] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao, *A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications*, IEEE Internet of Things Journal,2017

[7] Adam Calihman, *Architectures in the IoT Civilization*, January 30, 2019, Accessed on August 2020 , [Online] Available: https://www.netburner.com/learn/architectural-frameworks-in-the-iot-civilization/

[8] Sunil Cheruvu, Anil Kumar, Ned Smith, David M. Wheeler, *Demystifying Internet of Things Security: Successful IoT Device/Edge and Platform Security Deployment.* Publishers :Apress, Berkeley, CA, 2020,pp374-395

[9] Shadi Al-Sarawi, Mohammed Anbar, Kamal Alieyan, Mahmood Alzubaidi, *Internet of Things (IoT) Communication Protocols : Review*, 8th International Conference on Information Technology (ICIT),IEEE,2017,pp687

[10] *6 Leading Types of IoT Wireless Tech and Their Best Use Cases*, BehrTech Blog, Accessed on August 2020 , [Online] Available: https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/

[11] Lorenzo Vangelista, Michele Zorzi Marco Centenaro Andrea Zanella, *Long-Range Communications in Unlicensed Bands: the Rising Stars in the IoT and Smart City*, IEEE wireless communication, 2016, pp3

[12] P.Ray, *A survey on Internet of Things architectures*, Journal of King Saud University -Computer and Information Sciences, volume 30, Issue 3, July 2018, pp194-197

[13] Omri Cohen, *Popular IoT Protocols of 2019*, CTO & VP R&D, Co-Founder, Axonize, March 20, 2019, Accessed on August 2020, [Online].Available: https://www.axonize.com/blog/iot-technology/popular-iot-protocols-2018-an-overview-comparison-updated/

[14] Musa G. Samaila, Miguel Neto, Diogo A. B. Fernandes, Mário M. Freire, Pedro R. M.Inácio, *Challenges of securing Internet of Things devices: A survey*, WILEY, 2018,pp9-15

[15] Keyur K Patel, Sunil M Patel PG Scholar Assistant Professor, *Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges*, Volume 6 Issue No. 5, International Journal of Engineering Science and Computing IJESC 2016, pp2

[16] *Everything You Need to Know About Publish/Subscribe*, Accessed on August 2020, [Online]. Available: https://www.ably.io/concepts/pub-sub

[17] Yanlei Diao, Michael J. Franklin, *Publish/Subscribe over Streams*, University of California-Berkeley, Berkeley,2009, pp201

[18] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, Anne-Marie Kermarrec, *The Many Faces of Publish/Subscribe*, ACM Computing Surveys, Vol 35, No 2, June 2003, pp114–131

[19] Yu-Ling Chang, *Distributed Publish/Subscribe Network*, SlidePlayer, 2019, Accessed on August 2020 , [Online].Available: https://slideplayer.com/slide/15321888/

[20] The HiveMQ Team, *Publish & Subscribe - MQTT Essentials: Part 2*, January 19, 2015, Accessed on August 2020, [Online]. Available: https://www.hivemq.com /blog/mqtt-essentials-part2-pubs sub

[21] Angelo Corsaro, Ph.D, *Analysis of the Advanced Message Queuing Protocol(AMQP) and comparison with the Real-Time Publish Subscribe Protocol (DDS-RTPS Interoperability Protocol)*, RTI,2007

[22] Hendry Gunawan, *DISTRIBUTED PUBLISH/SUBSCRIBE NETWORK*, Sample Page, 2017, Accessed on August 2020, [Online]. Available: http://ueu6911.weblog.esaunggul.ac.id/distributed-publishsubscribe-network/

[23] Daniel Happ, Niels Karowski, Thomas Menzel, Vlado Handziski & Adam Wolisz, *Meeting IoT platform requirements with open pub/sub solutions*, Institut Mines-Telecom and Springer-Verlag France,2017

[24] Peter R. Egli, *MQTT MQ Telemetry Transport an introduction to MQTT, a protocol for M2M and IoT applications*, INDIGOO,2016

[25] Helbert Da Rocha, A*n MQTT-SN-based protocol for QoS adaptation in Wireless Sensor Networks*, Master's thesis, Ponta Grossa, 2018

[26] Steve, *Introduction to MQTT-SN (MQTT for Sensor Networks)*, 2017, Accessed on 2020, [Online]. Available: http://www.steves-internet-guide.com /mqtt-sn/

[27] *Healthcare*, 2018, Accessed on August 2020, [Online]. Available: https:// www.merriamwebster.com/dictionary/health%20care

[28] Pijush Kanti, Dutta Pramanika, Anand Nayyar, Gaurav Pareek, *Telemedicine Technologies*, chapter7 *WBAN: Driving e-healthcare Beyond Telemedicine to RHM: Architecture and Protocols*, Academic Press,2019, pp89-119

[29] Pijush Kanti, Dutta Pramanika, Gaurav Pareekb, *Telemedicine Technologies*, chapter14 *Security and Privacy in Remote Healthcare: Issues, Solutions, and Standards, Academic Press, 2019, pp201-225*

[30] Joel. Rodrigues, Dante. Segundo, Jalal Al-Muhtadi, Victor Hugo, Heres Arantes Junqueira, Rafael Maciel Prince, Murilo Henrique Sabino, *Enabling Technologies for the Internet of Health Things*, IEEE Access, 2018

[31] *An introduction to E- healthcare system and My Health Record System*, Accessed on August 2020, [Online]. Available: https://assignmenthelp4me .com/article-e-health-system-its-advantages-and-disadvantages-367.html

[32] Eike-Henner W. Kluge, *E-Health and its Challenges, University of Victoria*, Canada, Health Management, Volume 5 , Issue 2 ,2010

[33] Raviteja Baleka, Prof. Raghudathesh G P, 3Megha D H, Bindu H V, Madhuri C N, ,*MQTT Based Patient Health Monitoring*, International Journal of Pure and Applied Mathematics Volume 120 No. 6 2018, pp799-807

[34] Borade Samar Sarjerao, Amara Prakasarao, *Smart Healthcare Monitoring System Using MQTT Protocol*, 3rd International Conference for Convergence in Technology (I2CT) The Gateway Hotel, XION Complex, Wakad Road, Pune, India. Apr 06-08, 2018

[35] Hamid M Hasan, *IoT Protocols for Health Care Systems: A Comparative Study*, international Journal of Computer Science and Mobile Computing, Vol.7 Issue.11, November- 2018, pp38-45

[36] O'Kane, Mentis, & Thereska, *Non-static nature of patient consent: Shifting privacy perspectives in health information sharing*, Group and Team Issues in the Health Domain, San Antonio, TX, US, 2013, pp23–27

[37] Harsha S. Gardiyawasam, Pussewalage, Vladimir A. Oleshchuk, *Privacy preserving mechanisms for enforcing security and privacy requirements in E-health solutions*, International Journal of Information Management 36, 2016, pp1161–117

[38] Anderson, R, *A security policy model for clinical information systems*, In: Pro, 17th IEEE Symposium on Security and Privacy, IEEE CS,1996, pp30–43

[39] Biskup, J., Bleumer, G, *Cryptographic protection of health information: cost and benefit,* International Journal of Bio-Medical Computing 43, 1996, pp61–67

[40] Narayan, S Gagné, M Safavi-Naini, R, *Privacy preserving EHR system using attribute-based infrastructure,* In Proceedings of the ACM workshop on cloud computing security workshop, 2010, pp47–52

[41] Rafik Hamza, Zheng Yan, Khan Muhammad, Paolo Bellavista, Faiza Titouna, *A privacy-preserving cryptosystem for IoT E-healthcare*, ELSEVIER Ins,2019

[42] Guoming Wang, Rongxing Lu, Cheng Huang, Yong Liang Guan, *An efficient and privacy-Preserving pre-clinical guide scheme for mobile e-Healthcare*, Journal of Information Security and Applications 46,2019, pp271–280

[43] R. Neisse, G. Steri, Baldini, *Enforcement of security policy rules for the Internet of Things*, IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications, 2014, pp165-172

[44] Joseph Jose Anthraper, Jaidip Kotak, *Security, Privacy and Forensic Concern of MQTT Protocol*, International Conference on Sustainable Computing in Science, Technology & Management (SUSCOM),2019

[45] Meena Singh, Rajan MA, Shivraj VL, and Balamuralidhar P, *Secure MQTT for Internet of Things (IoT)*, Fifth International Conference on Communication Systems and Network Technologies, 2015

[46] Adil Bashir and Ajaz Hussain Mir, *Optical and Wireless Technologies :Lightweight Secure-MQTT for Internet of Things,* Springs, 2020, pp82

[47] Adil Bashir, Ajaz Hussain Mir, *Secure Framework for Internet of Things Based e-Health System*, International Journal of E-Health and Medical Communications Volume 10, Issue 4, 2019

[48] Martin Stehlık, *Comparison of Simulators for Wireless Sensor Networks*, MASTER THESIS, Masaryk University Faculty of Informatics, spring, 2011

[49] Tayyab Mehmood, *COOJA Network Simulator: Exploring the Infinite Possible Ways to Compute the Performance Metrics of IOT Based Smart Devices to Understand the Working of IOT Based Compression & Routing Protocols*, Dept. of Electrical Engineering, SEECS, NUST Islamabad, 2017