Cite it as: Wafa'a Kassab and Khalid A. Darabkh, "A-Z Survey of Internet of Things: Architectures, Protocols, Applications, Recent Advances, Future Directions and Recommendations," Journal of Network and Computer Applications, Elsevier, vol. 163, p.102663, August 2020. DOI: https://doi.org/10.1016/j.jnca.2020.102663

Published via this link: https://www.sciencedirect.com/science/article/pii/S1084804520301375

A-Z Survey of Internet of Things: Architectures, Protocols, Applications, Recent Advances, Future Directions and Recommendations

Wafa'a Kassab[1] and *Khalid A. Darabkh[2]
[1]Department of Infrastructure and Information Security, Ministry of Finance
Amman, 11118, Jordan
[2*]Department of Computer Engineering, The University of Jordan,
Amman, 11942, Jordan
Emails: wafaa.kassab@mof.gov.jo and k.darabkeh@ju.edu.jo

***Abstract* - Ubiquitous sensing, provided via wireless sensor networks technologies, disseminates across many domains of contemporary day living. This provides the ability to sense, process, analyze and infer environmental parameters from natural resources and delicate ecologies to urban environments. The explosion in the number of devices that are connected to the internet has led to the emergence of the Internet of Things (IoT) technological revolution. In these technologies, actuators and sensors incorporate smoothly with the IoT environment. Furthermore, the sensed data is shared through platforms to innovate a common operating picture. This cutting-edge technology is fueled by a diversity of IoT devices that enables technologies such as near field communication, embedded actuator, sensor nodes, radio frequency identification tags, and readers. IoT has emerged from its infancy and has established a fully integrated future internet. Different visions of IoT technologies have been reviewed, however, what emerges currently in this field should be faced and displayed via the research community. In this paper, we are keen to discuss the recent worldwide implementation of IoT, where the prime enabling technologies, recent and future communication protocols and application areas that drive IoT research in the near future are explored. Furthermore, the original, recent, future enhancements of all IoT stack's protocols are extensively discussed. Middleware's definition, usages, types and open research challenges are further illustrated. Not only to this extent but rather, this survey details the simulation tools of IoT networks, IoT sensors along with their recent application areas, broad IoT research challenges, as well as in-depth analysis of IoT research history and recommendations that attract current IoT researchers' attention.**

***Keywords* – IoT architectures; protocols; applications, IoT middleware; IoT simulators; IoT challenges; future directions; recommendations**

1. Introduction

The next revolution in the era of computing will be out of the realm of the classical desktop. In IoT environment, the numerous things that surround us will be connected to the internet in one way or another [1]. Various sensor network technologies and Radio-Frequency Identification (RFID) will emerge to face this novel challenge, where communication and information systems are embedded in the area that surrounds us [2] [3] [4] [5] [6]. This will lead to the creation of tremendous amounts of data that has to be processed, stored and presented in an efficient, easy and seamless manner [7] [8] [9] [10]. The cloud-computing model offers a virtual infrastructure to perform such computing through integrating surveillance and storage devices, analytics tools, client delivery, and visualization platforms [11]. The cost-based paradigm that cloud computing provides will authorize service provisioning for users and businesses to access their applications on-demand from anywhere and at any time [12].

The indispensable part of IoT is its smart connectivity with the present network and context-aware computation utilizing network resources. The evaluation of widespread communication and information networks comes from the growing existence of 4G-LTE and Wi-Fi wireless communication protocols [13]. However, to let IoT vision emerges successfully, computing standards need to go beyond conventional mobile computing technologies and develop into connecting every existing thing around us and embedding intelligence in the surrounding environment. There are essential demands that have to exist in order to achieve context-aware computation and smart connectivity in an IoT environment. These demands are 1) Understanding of IoT users and their appliances, 2) Pervasive communication networks and software architectures to transfer and process the sensed data to where it is relevant, 3) Analytics tools for autonomous and intelligent behavior in IoT systems.

An essential evolution of the present internet into a network of connected-things not only interacts with the physical environment through actuation, monitoring, and control, nor simply harvests data from the surrounding environments, but also utilizes existing internet criteria to facilitate data transmission, analytics, and communication [14] [15] [16] [17] [18]. IoT area is fueled by the propagation of intelligent devices that are enabled by various wireless technologies such as RFID, Bluetooth, telephonic data services and Wi-Fi, in addition to embedded actuators and sensor nodes. IoT has emerged from its infancy and is transforming from the present traditional internet into a completely integrated future internet [1] [19]. The revolution of IoT has led to an increasing interconnection among things at an unprecedented scale and speed to create an intelligent environment. In 2011, the number of interconnected devices overtook the number of people on the face of the earth. Currently, there are 9 billion devices that are connected to the internet and it is anticipated to reach 24-50 billion IoT devices in 2020 [6] [12]. As stated by the global system for mobile communications, this will yield a profit of $1.3 trillion for mobile network operators that cover main sectors such

as automotive, consumer electronics, utilities, and health. Numerous researches, industries, and companies are presently involved in the development of different IoT aspects to satisfy the increasing technological requirements that come with such rapid growth.

1.1 Related Works

Many works have surveyed and covered the different aspects of IoT technology. However, the contributions of these works and the research community on IoT-related topics are still highly fragmented and inadequate, and to a large extent concentrated on only a few aspects of this domain. Also, the involvement of communications and networking societies is still limited, despite the importance of their contributions to the evolution of this field. This subsection presents a literature review for some of these works organized chronologically, with a brief discussion regarding the topics they handled in their surveys.

Atzori et al tried in their survey paper to describe different visions of IoTs model based on diverse scientific communities' points of views [20]. This survey addressed the main communication technologies and identified wireless and wired actuator and sensor networks without providing any details regarding the protocols and their enhancements related to each IoT layer architecture. Furthermore, it illustrated and reviewed the main technologies of the IoT paradigm along with the benefits behind spreading this technology in different domains of everyday life. Finally, their work discussed different proposed issues and open research challenges that faced the IoT domain until 2009. Miorandi et al presented the vision of the IoT model and defined the main related concepts wherein they indicated the significant additions provided by related researches and technology contexts in this field [21]. Additionally, multiple research and security challenges were investigated followed by a brief discussion of possible IoT applications and their impact on different fields. Finally, the authors reviewed related IoT initiatives until 2012. However, their paper not only did not cover all the aspects of the IoT field but is also outdated. The concept and history of IoT were demonstrated with a brief introduction of different IoT architectures by Said et al [22]. Furthermore, they introduced a few applications that can be implemented based on IoT technology, besides demonstrating open problems and research challenges in this field. The definitions, taxonomy, and trends of IoT technology with a brief discussion regarding some technologies and applications of this field were presented by Gubbi et al [12]. Moreover, an example of cloud computing implementation using Aneka/Azure cloud platform was presented, without introducing further details about the specifications of cloud computing technology. Many challenges and open research issues were examined by Whitmore et al, with a brief introduction about IoT technologies, applications, and business models [23]. Al-Fuqaha et al gave some technical details about IoT technologies, architectures, applications, and protocols [24]. In addition, they provided a concise presentation regarding the interaction among IoT solutions, big data, fog, and cloud computing. Finally, some Quality of Service (QoS), security issues, and challenges were examined. A limited discussion was provided by Kraijak and Tuwanut about architecture, protocols, applications, privacy and security problems in the IoT domain. Finally, they presented the applications and future trends of IoT [25]. Masek et al described Machine-to-Machine (M2M) communication protocols in cellular networks, with a summarized presentation of some bidirectional communication protocols [26]. They also offered a brief investigation regarding the convenience of both protocol buffers format and JavaScript Object Notation (JSON) for M2M communication. They also proposed a live smart home project for Telekom Austria group using JSON and protocol buffer techniques to implement M2M communication. Many IoT aspects were covered by Ray [27]. Firstly, he tried to give multiple definitions of IoT paradigm from different researchers' perspectives. Secondly, different architectures of this technology were discussed. Thirdly, the main domains and applications that can be implemented by IoT technology were presented, followed by sections about previous wireless and wired technologies and protocols that were implemented in this field. Finally, possible research and security challenges were investigated. The Survey paper of Sethi and Sarangi covered different taxonomies of IoT stacks with a brief description of each layer's technologies and protocols [28]. Also, they profiled some types of IoT sensors with their related applications. Some challenges behind proposing the term of middleware were discussed along with identifying their types. Burhanuddin et al provided a theoretical background of the IoT paradigm, with an analysis review of many surveys on this field [29]. Further, they discussed the requirements needed to implement IoT applications, followed by a discussion about future directions and challenges that face this domain. An inadequate interpretation was given to cover relevant sides of IoT middlewares such as the needs behind middleware, its capabilities, enabling technologies, and challenges by Ngu et al [30].Moreover, the authors classified various types of IoT middlewares and gave many examples for each architecture, without taking into consideration other aspects of IoT model. Silva et al presented one type of IoT architectures with its relevant technologies and then went to simply summarize some of the prevalent communication protocols and standards that are adopted by the IoT field [31]. Although some IoT applications were identified, followed by a brief discussion of some issues and security challenges that face the domain, it was overall, an insufficient study that left the reader with many questions about the intricacies of the subject at hand. Different points of view were presented regarding various types of IoT stack architectures with possible attacks relevant to each layer and suggestions to overcome and solve these issues by Burhan et al [32]. In addition, a number of communication technologies were highlighted along with their drawbacks and characteristics. An overview of different procedures that were proposed to secure IoT environment with their restrictions until 2018 was discussed, where a novel IoT architecture model was proposed by the authors to fill security gaps in the previous architectures. To this end, a section for some issues and challenges that face the security of IoT environment was provided. Atlam et al identified the general notion characteristics of IoT, followed by a presentation of simple IoT stack architecture [33]. On the other hand, they discussed some communication technologies and applications of the IoT field. However, limited discussion regarding IoT challenges and future directions was introduced.

˘Colakovi´c and Hadžiali´c identified features and visions of IoT and provided insights of some enabling technologies and communication protocols based on their functionalities, while they slightly reviewed the middleware and network domains [34]. The authors focused further on addressing and discussing the challenges and open issues that face the IoT model. A detailed presentation was provided for the communication protocols of the application layer, such as Hypertext Transfer Protocol (HTTP), Constrained Application Protocol (CoAP), Message Queuing Telemetry Transport (MQTT), Data Distribution Service (DDS), Advanced Message Queuing Protocol (AMQP), and Extensible Messaging and Presence Protocol (XMPP) along with their implementations in different segments of the IoT environment (IoT, cloud, fog) by Dizdarević [35]. Thereafter, the author conducted a comparison between these protocols considering distinctive aspects such as latency, bandwidth consumption and throughput,

energy consumption, security, and developer's choice. Finally, a concise description of open issues and challenges were provided. Balaji et al [36] presented a few technologies and protocols that are utilized in IoT domain, followed by a summary of some security issues that face this field. In addition, they mentioned the popular IoT- based lifesaver tools and discussed a number of real-time applications. Finally, few of the issues prevalent in the IoT field were explained and the future scope and applications were left out. A comprehensive focus on IoT forensic was presented by Yaqoob et al [37]. This work demonstrated novel factors that affect and enable forensics in the IoT domain. It further provided an investigation of several IoT forensics literatures and categorized them depending on sources of evidence, forensics phases, networks, enablers, forensics data processing, forensics tools, forensics layers, etc. to analyze their strengths and weaknesses. Several research challenges and issues were identified as future research directions. Sharma et al presented many definitions for IoT notions based on different researchers' perspectives [38]. The authors chronologically addressed the evolution of this technology. In the end, a slight discussion was provided to handle different IoT aspects such as its technology trends, communication standards, architecture and an overview of its future.

1.2 Findings and Impacts

There are many surveys that have been done to investigate different fields and issues of IoT domain till now. To the best of our knowledge, there are no prior surveys similar to ours. Interestingly, Table 1 displays how this work is distinctive from other highly cited papers mentioned in the previous section considering many perspectives out of which IoT paradigm, architectures, spreading spectrum techniques, layers protocols (original, recent, future enhancements), IoT middleware (recent challenges), IoT simulation tools, IoT applications, research security and challenges, and research history analysis and recommendations. In light of the aforementioned deficiencies of the related works, the major findings of this work can be summarized as follows:

1. Having higher value for researchers, as this survey is considered to be a starting point for their future researches because it gives the reader the opportunity to comprehend what has been done in IoT field, what still needs to be developed as well as what the risks and weakness factors are that need to be addressed. In addition, it exhibits the current trends in IoT research that are encouraged by the need for the convergence in multiple interdisciplinary technologies and IoT applications.
2. Highlighting diverse visions, definitions, and a thorough overview of IoT components and features for the reasoning of expediting a better comprehension of different IoT specifications by researchers and technicians.
3. Providing a detailed demonstration of different spreading spectrum techniques (i.e., Direct-Sequence Spread Spectrum (DSSS), Frequency-Hopping Spread Spectrum (FHSS), Chirp Spread Spectrum (CSS), Time-Hopping Spread Spectrum (THSS)). Based on such important information, the network designers can use the proper or suitable spreading spectrum techniques in their IoT communication systems, which will reduce crosstalk interference, obtain less static noise and data integrity, reduce signals susceptibility to multipath fading, avoid signals interference, and guarantee security implementation by making IoT data signals hard to detect, intercept or demodulate.
4. Providing insights and deep synopsis of the most recent standards and protocols, which are classified based on different IoT stack architecture (i.e., application, transport, network, and data link layers), thereby making sure that the reader will be aware of the full picture of the original, current, and future enhancements of each protocol. Matter of fact, conducting comparisons between all protocols in each layer from different perspectives will help the researchers and technicians in deciding which one suits them more quickly in professional and organized manners without digging through precise details provided in standard specifications, sources, and Request for Comments (RFC).
5. Presenting a comprehensive overview of the emerging challenges and issues in the IoT domain in order to be tackled through future researches. In fact, after studying numerous IoT research papers we have come to the conclusion that most of the challenges and security issues emerge from the remarkable increase in data traffic, the huge variety of traffic types, diversity of IoT devices, great variances in data forms, heterogeneous networks, etc. All of these concerns have a dramatic effect on the performance and QoS of the IoT systems.
6. Detailing the most and recent trends and specifications of IoT middleware aspects. In other words, we make sure that the readers get a full understanding of the recent challenges and issues that face the middleware field, the diverse classification of middleware architectures, and differences of emerging middleware platforms for each type of architecture.
7. Introducing a comprehensive overview of IoT simulators that are currently available through classifying them into categories according to their functions and then conducting comparisons while considering prevailing needs and aspects. Besides, the major challenges raised through moving from simulating the Wireless Sensor Network (WSN) environment into IoT are highlighted, thereby allowing the developers to upgrade the current versions of these simulators to suit IoT environment's requirements.
8. Analyzing the IoT research history, utilizing Scopus database through 2011 to 2020, in a very professional manner which primarily includes both IoT stack and middleware architectures. In particular, as far as the former is concerned, we analyze the growth and diminishment of publications in the whole IoT stack which includes data link and communication protocols, network, transport, as well as application layers. In regards to the latter one, we analyze its publications' growth and diminishment considering actor-based, event-based, cloud-based, and service-based architectures bearing in mind that addressing the challenges and limitations of middleware architectures has to take the functional components as service composition, registration, and discovery and non-functional needs, such as ease of deployment, privacy, security, availability, reliability, timeliness, and scalability all into consideration. As a result, we provide recommendations that will certainly attract most IoT researchers.

1.3 Paper Outline

The remaining of this paper is organized as follows:

- Overview of the IoT paradigm (definition, functional blocks, basic components of smart devices) (section 2).

Table 1: Comparison of this survey with other related IoT works considering several IoT aspects

Articles	Survey subject	Year	Overview of IoT paradigm	Architecture of IoT	Spreading spectrum techniques	IoT layers protocols			IoT middleware		IoT simulation tools	IoT applications	Research security and challenges	Research recommendations
						Original	Recent	Future enhancements	Recent	Challenges				
Atzori et al [20]	The internet of things: A survey	2010	✕[1]	✕	✕	★	✕	✕	☆	✕	✕	★	☆	✕
Miorandi et al [21]	Internet of things: Vision, applications and research challenges	2012	✕	✕	✕	☆	✕	✕	✕	✕	✕	☆	★	✕
Said et al [22]	Towards internet of things: survey and future vision	2013	✕	☆	✕	✕	✕	✕	✕	✕	✕	☆	☆	✕
Gubbi et al [12]	Internet of things (IoT): A vision, architectural elements, and future directions	2013	☆	☆	✕	✕	✕	✕	✕	✕	✕	☆	☆	☆
Whitmore et al [23]	The internet of things A survey of topics and trends	2014	✕	✕	✕	☆	✕	✕	☆	✕	✕	✕	☆	☆
Al-Fuqaha et al [24]	IoT survey on enabling, technologies, protocols and applications	2015	☆	✓[2]	✕	✓	✕	✕	✕	✕	✕	☆[3]	✓	✓
Kraijak et al [25]	Survey on IoT architecture, protocols, applications, security, privacy, implementation and future trends	2015	✕	☆	✕	☆	✕	✕	✕	✕	✕	☆	☆	✕
Masek et al [26]	Implementation of true IoT vision: Survey on enabling protocols and hands-on experience	2016	✕	✕	✕	★	✕	✕	✕	✕	✕	☆	✕	✕
Ray [27]	A survey on Internet of things architectures	2016	☆	★	✕	★	✕	✕	☆	✕	✕	✓	✓	★
Sethi et al [28]	Internet of Things: Architectures, protocols, and applications	2017	★	★	✕	☆	✕	✕	☆	✕	✕	☆	☆	✕
Burhanuddin et al [29]	Internet of things architecture: current challenges and future direction of research	2017	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓	☆
Ngu et al [30]	IoT middleware: A survey on issues and enabling technologies	2017	✕	✕	✕	✕	✕	✕	★	☆	✕	✕	☆	✕
Silva et al [31]	Internet of things: A comprehensive review of	2017	☆	★[4]	✕	☆	✕	✕	✕	✕	✕	☆	☆	✕

[1] ✕: Aspect is not existed in the survey

[2] ✓: Aspect is covered to the core in the survey

[3] ☆: Aspect is shallow covered in the survey

[4] ★: Aspect is sufficiently covered in the survey

	enabling technologies, architecture, and challenges													
Burhan et al [32]	IoT elements, layered architectures and security issues: a comprehensive survey	2018	☆	✓	✕	☆	✕	✕	✕	✕	✕	☆	✓	☆
Atlam et al [33]	Internet of things: State-of-the-art, challenges, applications, and open issues	2018	✕	☆	✕	☆	✕	✕	✕	✕	✕	☆	☆	☆
Colakovi ´c et al [34]	Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues	2018	☆	☆	✕	☆	✕	✕	✕	✕	✕	✕	✓	☆
Dizdarević et al [35]	A Survey of communication protocols for internet of things and related challenges of fog and cloud computing integration	2019	✕	✕	✕	★	☆	✕	✕	✕	✕	✕	☆	✕
Balaji et al [36]	IoT technology, applications and challenges: A contemporary survey	2019	✕	✕	✕	☆	✕	✕	✕	✕	✕	☆	☆	✕
Yaqoob et al [37]	Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges	2019	☆	✕	✕	✕	✕	✕	✕	✕	✕	☆	☆	✕
Sharma et al [38]	The History, present and future with IoT	2019	☆	★	✕	☆	✕	✕	✕	✕	✕	☆	☆	✕
Kassab and Darabkh	This paper	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

- The taxonomy of IoT architecture (IoT layers, Fog layers, cloud computing) (section 3).
- Distinct spread-spectrum telecommunications techniques such as (DSSS, FHSS, CSS, THSS) (section 4).
- IoT layers' protocols (origin, recent, future enhancements) (section 5).
- Middleware's definition, uses, types and open research challenges (section 6).
- Different simulation tools of IoT networks (section 7).
- Various types of IoT sensors with recent application areas in IoT (Section 8).
- Broad and open IoT research challenges (section 9)
- Conclusions and recommendations (Section 10).

2. Overview of the IoT Paradigm

IoT paradigm has opened the doors to new inventions, discoveries and interactions among things and people, which will, in turn, improve the exploitation of scarce resources and human quality of life. To comprehend the full picture of the IoT model, the following sections will address different IoT definitions, functional blocks, and smart devices' basic components.

2.1 Internet of Things Definition

During the previous decade, the IoT field has acquired a considerable interest in the industry as well as in the academic domains, the primary reason for this interest comes from the abilities that IoT provides [1]. It also guarantees to establish a world wherein all smart objects and devices are connected to the internet and can communicate with each other with minimum human interference [19]. The supreme purpose of IoT technology is to enhance people's life, wherein all smart objects around us realize what we require, what we want, what we like and behalf accordingly without explicit orders [19] [39]. IoT includes an enormous amount of technologies that form its vision but researches this field is still in its early stages. Thus, there is not a unified definition for IoT term, the subsequent definitions have been provided from distinctive researchers.

- Definition 1: Objects have virtual personalities and identities, where they are embedded with smart interfaces that allow them to communicate and connect with user contexts and social environments [40].
- Definition 2: Interconnected things that have active roles in what could be called the internet of the future [41].
- Definition 3: This expression consists of two words: Internet which is defined as the worldwide network of an enormous number of networks depending on communication protocols standards, whereas the word Things refers to all objects that are connected to that network based on the same standards [41].
- Definition 4: The environment of IoT network composes of physical and virtual entities, where these entities turn into virtual things inside a cyber-world. These things are embedded with different abilities as sensing, analyzing and processing and self-management based on interoperable communication protocols and specific criteria, these smart things should have unique identities and virtual personalities [6].
- Definition 5: IoT notion is anything that can be accessed from anywhere at any time by anybody for any service through any network. Thus, IoT can be called as 6Anys [27].

2.2 IoT Functional Blocks

An IoT paradigm is composed of a number of functional blocks, which ease different functionalities of smart objects like sensing, actuation, identification, management, and communication. Figure 1 shows these blocks with brief explanations in the following bullet points [27]:

- **Device:** Smart devices are the main units of the IoT system, where they are able to perform many operations such as sensing, monitoring, control, and actuation activities. They are also capable of exchanging data with applications and other smart servers. Each IoT device must be prepared with many interfaces to enable it to connect with other smart devices, where it consists of interfaces for Internet connectivity, I/O for sensors, audio/video, memory, and storage. IoT devices are varied according to the application they are utilized for. These applications could be smartwatches, wearable sensors, automobiles، industrial machines, LED lights, etc. [27].
- **Services:** There is an enormous number of applications that utilize IoT techniques from office automation and home appliances to manufacturing lines and goods tracking etc. Thus, it is required to apply specific IoT services in order to enhance IoT application development and to speed up its implementation. These services can be classified into identity-related services, services for device modeling, information aggregation services, devices discovery, devices control, collaborative aware services, ubiquitous services, data analytics and data publishing [27] [42].
- **Management:** The main feature of the IoT device, which distinguishes it from traditional devices that can be managed and controlled using mechanical buttons or switches, is remote management with or without human intervention. Furthermore, these devices can exchange data between each other to take a suitable decision later on [27] [40].
- **Security:** The data of networks, specifically data of wireless networks, is vulnerable to a massive number of attacks as a denial of service, spoofing, and eavesdropping, etc. Thus, IoT system tries to mitigate these attacks via the implementation of many security functions like privacy, authorization, authentication, data security, content integrity and message integrity [27] [39].
- **Application:** The application layer provides IoT users with interfaces that enable them to monitor and control diverse aspects of IoT applications. Furthermore, they permit users to analyze and visualize the status of IoT system at any time and from anywhere to take suitable action [27].

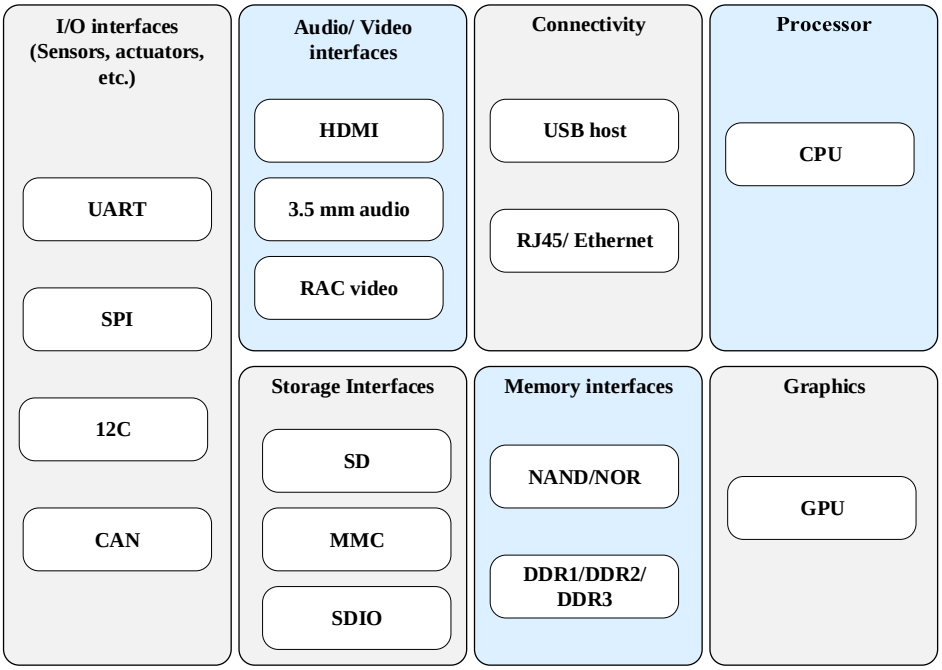


Figure 1: IoT device components

2.3 Basic Components of IoT Devices

IoT systems as mentioned before consisting of devices and applications, in order to allow them to communicate with each other they must have basic components, as will be illustrated below:

1. **Identification (ID):** Each object in IoT system must have a unique identification; an ID is assigned to an entity based on conventional parameters like universal product code, Media Access Control (MAC) ID, IPv6 ID or another custom method [27].
2. **Meta information:** Metadata consists of information about each device in IoT system such as device model, ID, revision, hardware, serial number and manufactured date.
3. **Security controls:** It resembles the "friend list" of Facebook, as the device owner can place restrictions on the devices types that can connect to his device [28].
4. **Service discovery:** This feature enables each IoT device to store details of all other smart devices that belong to the network in a specific directory. It is very important to keep these directories updated in order to get information about new devices that recently have joined the IoT network [28].
5. **Relationship management:** It allows each IoT device to start, update and terminate the relation between itself and other devices. Furthermore, it stores a list of the devices types that it should be connected with, according to the service type they are provided and based on human settings [27] [43]. For instance, a light sensor can create a relationship with a light controller device.
6. **Service composition:** This component enables interaction between smart objects and aims to provide users with the best-integrated service. To achieve such goals, the discovery service tries to find the required service that is provided by the smart object, to get benefit from it later on. It is also in charge of processing the data obtained from different objects to provide the user with the best solution [43].

3. Architecture of IoT

IoT connects millions of smart objects, which leads to more data traffic and the need for large data processors and storages [19]. Based on the above, IoT will face challenges regarding QoS, privacy, and security [44]. Thus, IoT architecture must take into consideration many issues such as interoperability, scalability, QoS, reliability, etc. [45]. In the literature, various IoT architectures have been suggested [46] [47] [48]. Nevertheless, each proposed architecture brings many shared drawbacks and fails to cover all of the IoT characteristics, which are summarized as follows [49]:

a. **Distributive:** IoT model is probably developed in an enormously distributed environment, where data can be collected from various sources and consequently can be processed via distinctive smart entities in a distributed procedure.
b. **Interoperability**: IoT devices that belong to distinct vendors have to communicate with each other to obtain mutual goals. Protocols and systems must be also designed in a manner that permits smart devices from numerous manufacturers to exchange their sensed data in an interoperable manner.
c. **Scalability**: Billions of objects are expected to join the network of any IoT environment. Thus, applications and systems that run on these environments must be able to manage and process a tremendous amount of data.
d. **Resources scarcity**: Both of computation units and energy are considered to be highly scarce resources.
e. **Security**: Users' feelings of being helpless and exposed under the control and dominant of an unknown external device could sorely handicap IoT deployment.

To overcome these issues, many researchers follow a specific-layered architecture for IoT infrastructure. In every proposed IoT architecture, similar techniques, functionalities, and services will be grouped into the same layer, which will facilitate the

development and enhancement of the architecture of each layer in the future [50]. There is no global consensus on the architecture of IoT, so different IoT architectures have been suggested by many researchers [49]. To the best of our knowledge and after an extensive search on IoT architecture models, we found that the superior model with respect to the elements that compose this environment is the "Three Based Architecture" model that is described in [51]. This architecture composes of the following three layers:

a. **IoT layer**: This layer contains all smart devices, entities, and end-users that are located in the IoT system.
b. **Fog layer:** All fog nodes are placed in this layer.
c. **Cloud layer*:*** All distributed cloud servers exist in this layer, where these servers consist of multiple processing units like a rack of high capabilities servers or it could be a huge server with multiple processing cores.

In every layer a set of nodes are grouped into domains, wherein a single IoT domain, that is composed of Nodes-Fog-Cloud agents, an application can be performed as depicted in Figure 2. The basic method that permits any IoT node, fog computing node and cloud server of communicating and interacting with each other is demonstrated as follows; firstly, an IoT node transmits its sensed data directly to a fog node that belongs to its domain application. As a result, the fog node processes the received data directly or sends it to another fog node or cloud server belongs to the same domain in order to send the reply back to the related IoT node. This step will reduce the service delay[5] of IoT node in receiving a response for any request, this comes from the location of the fog layer which allows its nodes to handle most of the requests come from the IoT layer [51]. The following sections demonstrate the architecture of each layer in the three-based architecture model.

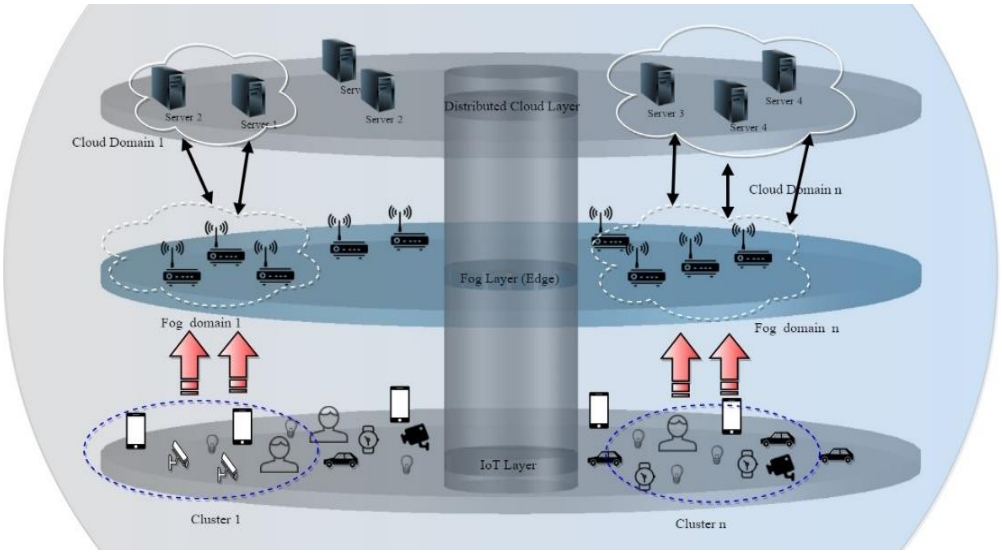


Figure 2: IoT three-based architecture layered

3.1 IoT Stack Architecture

Based on our thorough readings of a massive number of prior relevant surveys and books, we propose that the division of IoT stack consists of five layers which include perception, data link, network, transport, as well as application layers as shown in Figure 3, where all are discussed as follows:

i. **Perception Layer:** The primary mission of this layer is to sense the physical attributes of the entities that surround us and within the dominant of IoT network, where it depends on many sensing technologies such as RFID, WSN, Global Positioning System (GPS), etc. [49] [52]. Moreover, it is responsible for converting the sensed data to digital signals to be appropriate for network transmission. As a matter of fact, embedding intelligence and nanotechnology play an important role in this layer, as it enhances the processing capabilities of any object through inserting small chips (microcontroller) into smart devices that are used in everyday life [49].
ii. **Data Link Layer:** The IoT data link layer includes various communication protocols, which primarily provide services to the network layer. In fact, there are different standard technologies and protocols indicated by organizations for data link protocols out of which, Bluetooth, ZigBee, RFID, low power wide-area-networks, Z-wave, cellular [28].
iii. **Network Layer:** It is in charge of providing data with routing paths to be transmitted in packets form over the network area. The network layer establishes logical connections, delivers error reporting, manages and selects the routing path for data transmission. Moreover, this layer contains all network devices such as switches, firewalls, bridges, and routers, which are required to work along with suitable communication and routing protocols, such as 3G, 4G, 5G, Wi-Fi, infrared technology, ZigBee, fiber-to-the-x [49].
iv. **Transport Layer:** It works transitionally with the application layer to transmit and receive data without errors. The transmitting side of this layer is responsible for breaking messages that are received from the application layer into segments, and then send them to the network layer. In turn, the received segments will be reassembled into messages to be directly passed into the application layer by the receiving side. The transport layer provides features, such as packet delivery order, congestion avoidance, multiplexing, byte orientation, data integrity and reliability over the transmitted data.

[5] Service delay: Is the time period between the moment that IoT node transmits a service request and the time it receives the reply for its request [51].

v. **Application Layer:** This layer represents the front end of IoT architecture, where most of IoT potential will be exploited, because it provides IoT developers with interfaces, platforms, and tools that are required to implement IoT applications such as smart homes, intelligent transportation, smart health, and smart cities [49]. Moreover, it is responsible for receiving the processed data from the network layer.

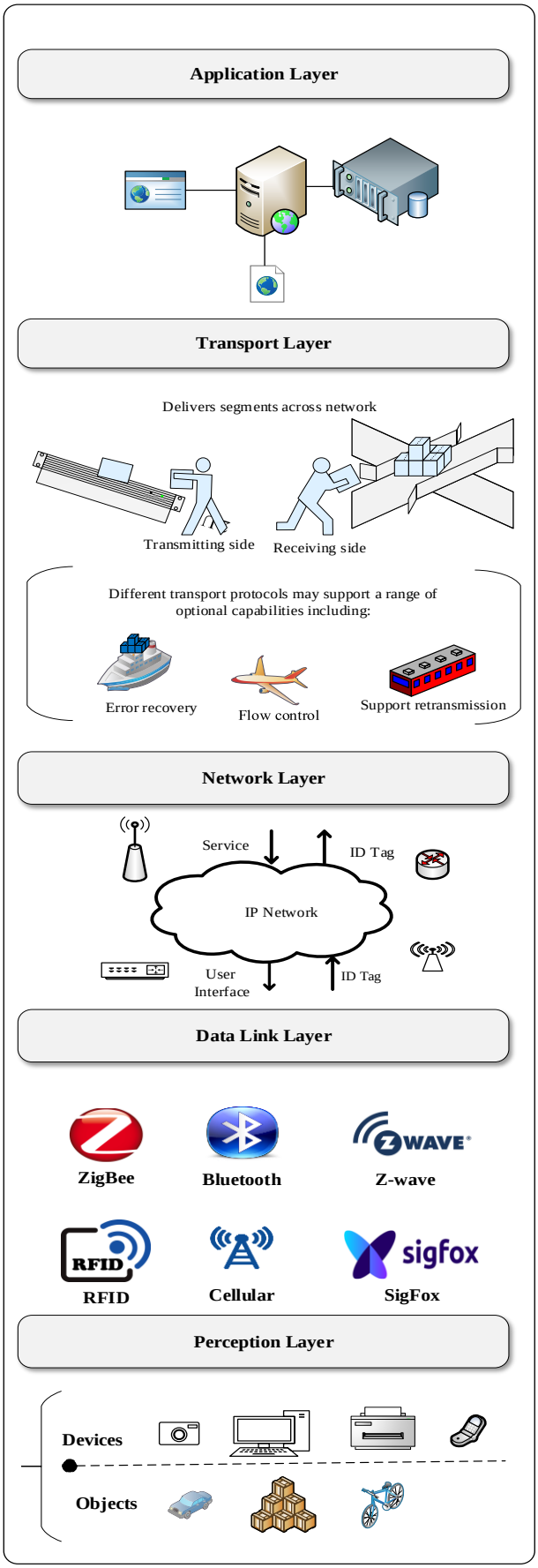


Figure 3: Five layers IoT architecture

3.2 Fog and Cloud Computing Layers

Big data that are generated by different IoT applications presents a new characteristic called Geo-distribution [53]. This new dimension requires that the sensed information has to be processed at the edge of the network area close to the smart devices instead of processing it by remote servers of cloud computing [54]. It is worth mentioning that it is indispensable to offer low latency response in order to allow smart objects to take the right action at the suitable time and to protect the integrity of sensitive infrastructure components. As a result, fog computing paradigm was suggested to extend cloud-computing services to the edge of IoT networks, to provide a highly virtualized platform that supplies many networking, storage and computational services between smart devices and cloud computing services [55]. Fog architecture comprises of four layers as depicted in Figure 4, which are monitoring, pre-processing, storage, and security layers [56].

3.2.1 Fog Layers Architecture

i. **Monitoring layer:** This layer is responsible for observing the activities of smart devices and networks. For example, it detects which sensor node performs some task, what task the node performs and at what time it is executed. Besides, this layer is in charge of monitoring the energy level of different network devices [28] [56].
ii. **Pre-Processing layer:** Performs data management, analyzing, filtering and trimming processes to generate useful and meaningful data.
iii. **Temporary storage layer:** After the pre-processing layer processes sensed data, it will be stored temporarily in the resources of this layer. The temporary storage layer offers many storage functionalities such as data storing, distribution, and replication [28].
iv. **Security layer:** It implements encryption and decryption techniques to protect the privacy and integrity of data.

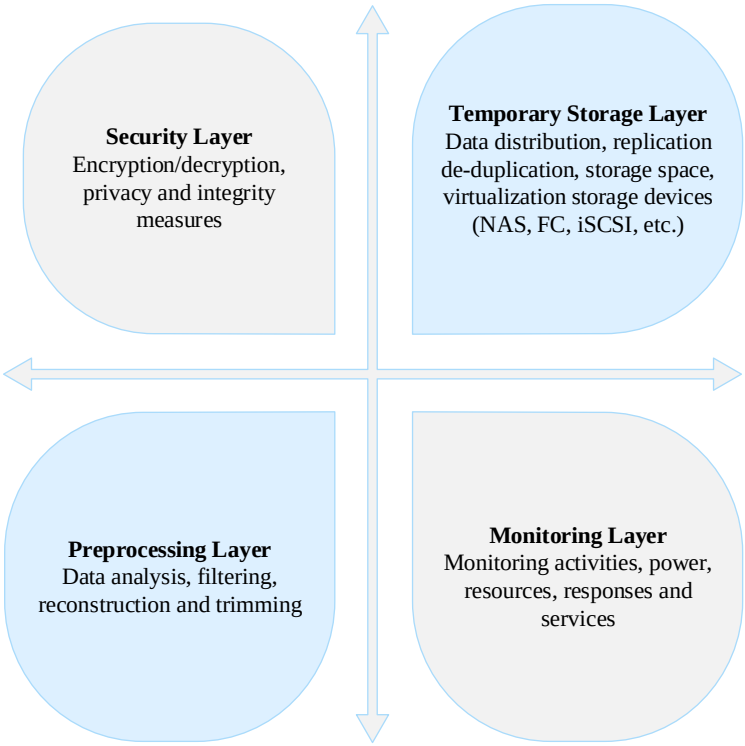


Figure 4: Layered architecture of fog computing

3.2.2 Why to Use Fog Computing Nodes

Fog computing nodes act as a bridge between smart objects, storage services, and large-scale cloud computing servers. This model extends network resources and services to the underlying network [56], so it has the capability of providing end-users with better delay performance services. Despite that, there is an important difference between the cloud and fog computing paradigms, where the cloud has enormous computational, communication and storage capabilities compared with fog computing [57], Figure 5 shows the roles of cloud computing and fog computing in the delivery of IoT services [58].

Connecting a massive number of smart objects to the internet such as smartphones, PCs, animals, and humans tracking, creates what is called the “Big Data” term that needs high capabilities to be stored, processed and analyzed. Fog computing nodes provide end-users with such abilities and are the best choice for many applications rather than farthest cloud computing for the following reasons:

1. Edge location, low latency, location awareness: According to that, fog computing provides its clients with rich applications and services with low latency requirements [57].
2. Geographical distribution: Applications and services that are hosted and processed by the fog nodes require widely distributed deployment of these nodes closer to the end-user. Fog, for instance, plays an essential role in delivering quality streaming to vehicles via access points and proxies that are positioned along tracks and highways.

3. Mobility supporting: It is common that fog applications communicate directly with mobile smart entities. Thus, fog computing is able to support mobility standards such as locator identifier separation protocol [28] [59] [60].
4. Real-time interactions: It has the ability to implement real-time interaction services since it can give an instantaneous response.
5. Dominance of wireless access.
6. Supporting online analytic and interaction with the cloud, as it plays a significant role in the ingestion and processing of a massive amount of data that are received from close smart devices.
7. Scalability: Fog permits IoT environments to grow, so as the number of smart devices increased, as a result, the number of fog nodes will be increased too to handle the new load. Such resource expansion cannot be achieved from the cloud side since the deployment of new servers is highly cost.
8. On the fly analysis: Fog resources aggregate data to transmit it partially processed to the cloud servers for additional processing.
9. Power constraints: Since most of the smart devices are battery-powered, long-distance communication toward the cloud will deplete their energy faster.

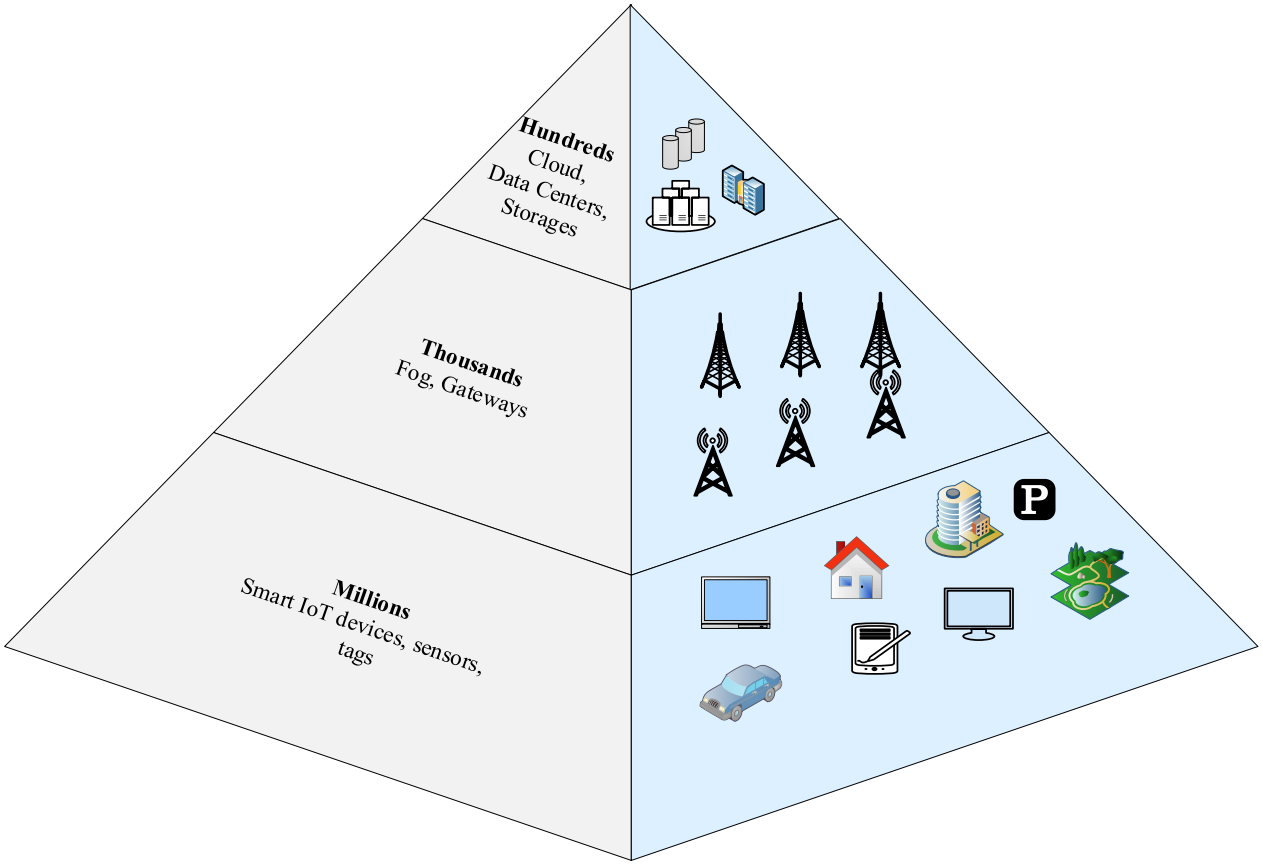


Figure 5: The role of the cloud and fog computing in the delivery of IoT services

3.2.3 Cloud Computing Architecture

In the IoT model, communication and information systems are embedded in a smart environment that surrounds us. This will lead to the generation of a massive amount of data that needs to be presented, processed and stored in an efficient, seamless and easy interpreting manner. According to [12], cloud-computing technology is the latest paradigm that proves its efficiency, scalability, autonomy, and reliability, as it provides high capabilities in dynamic resources discovery, ubiquitous access and composability[6], which are important for the prosperity of the future of IoT applications [49]. This platform plays several roles such as a data receiver from smart devices, a computer that analyzes and interprets distinct types of data, and as a supplier of web-based visualizations [61]. Many researchers try to construct a compatible architecture that can describe the function of the cloud computing paradigm as shown in Figure 6. This model consists of three layers, which are; the base layer that includes a database to keep details of all smart devices in the IoT network. The next layer is the component layer, which includes the codes that are required to interact with all IoT entities and employ a subset of these entities to execute a service or to query their status, where the last layer in this model is the application layer, which is in charge of providing users with the needed services [28].

[6] Composability: A system design principle that deals with the inter-relations among components, highly composable system supplies components that can be nominated and assembled in innumerable combinations to satisfy particular user requirements [252].

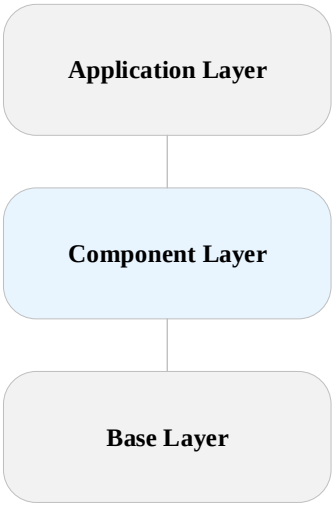


Figure 6: Cloud computing architecture

4. Spread-Spectrum Telecommunications Protocols

In 1941, spread spectrum communications technology was first described by Hollywood actress Hedy Lamarr and pianist George Antheil as the following; to achieve a secure communication in any radio communication system, both transmitter and receiver that are forming this system must be tuned to the same plurality of frequencies. According to their innovation, they were granted U.S. patent #2.292.387.

In 1978, the federal communications commission in the USA assigned specific frequency bands for the systems that utilize SS techniques in their communications, specifically these bands were devoted to Industrial, Scientific, and Medical (ISM) applications. The great success of SS technologies comes from their reliability, immunity against jamming impacts, ability to guarantee privacy and security, low sensitivity to signals interferences and low power exhaustion. SS techniques are implemented in police and military applications to attain a high degree of security and privacy because the signal bandwidth is distributed over enormous frequency ranges, which makes it impossible to track transmissions patterns [62]. It is worth stating that the SS techniques enable numerous users to transmit their data at the same channel simultaneously since they use different spreading frequencies. Figure 7 below describes the main characteristics of any spread spectrum system [63]. Firstly, the digital signal is fed into channel encoder to be converted into analog form with a narrow bandwidth around a specific frequency. Then the digital signal will be modulated with a concatenation of digits known as a spreading sequence or a spreading code that is produced by a pseudorandom number or a Pseudo-Noise (PN) generator in order to increase the bandwidth of the transmitted signal [64]. On the other hand, the received signal will be demodulated on the receiver's side by the same spreading code, to be fed later on into a channel decoder to retrieve the original data. The main pros of employing spreading spectrum techniques in communication systems are summarized below:

1. The signal gains resistance against multipath distortion and different sorts of noise.
2. Spread spectrum techniques can be utilized for encrypting and hiding signals, where the recipient who knows the spreading code can only recover the encrypted signal.
3. The communication channel can simultaneously be shared by multiple signals without any interference, which enables this technique to be utilized in cellular applications.

Pseudorandom numbers are generated by a deterministic algorithm that is fed by an initial value called a seed, so these numbers are not considered to be random. The vital point is that unless you know both the seed and the algorithm, it is unpractical to anticipate SS sequence. Thus, when the recipient gets pseudocode and algorithm from a transmitter, it will be possible to decode the signal effectively. There are four types of SS techniques, which are:

- ✓ Direct Spread Spectrum.
- ✓ Frequency Hopping Spread Spectrum.
- ✓ Chirp Spread Spectrum.
- ✓ Time Hopping Spread Spectrum.

4.1 Direct-Sequence Spread Spectrum

In this technique, every bit in the original signal will be represented by numerous bits that compose the transmitted signal using a spreading code [63]. In other words, every bit of the original signal will be multiplied by a sequence of **n** bits that is called a chip, where its rate is equal to **n** times of the original signal bit rate [64]. The multiplied signal will be then spread across a wider frequency band that is proportional to the chip's PN, size as shown in Figure 8. One procedure of implementing the DSSS technique is to combine the digital signal stream with spreading code bits sequence by utilizing an exclusive-or operation. The spread signal can give security if the intruder does not know the spreading code; also, it can give immunity against signals interferences if each user utilizes a distinctive spreading code.

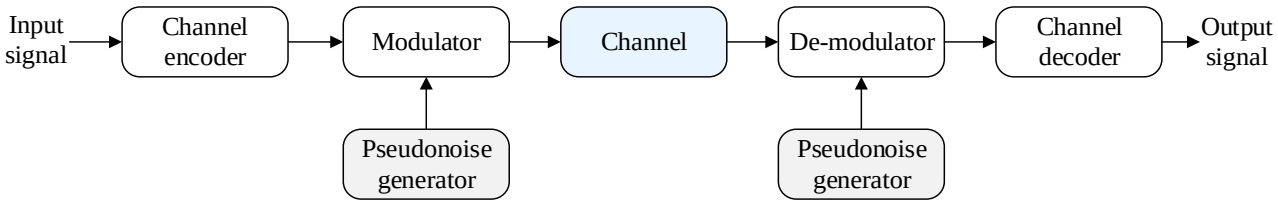
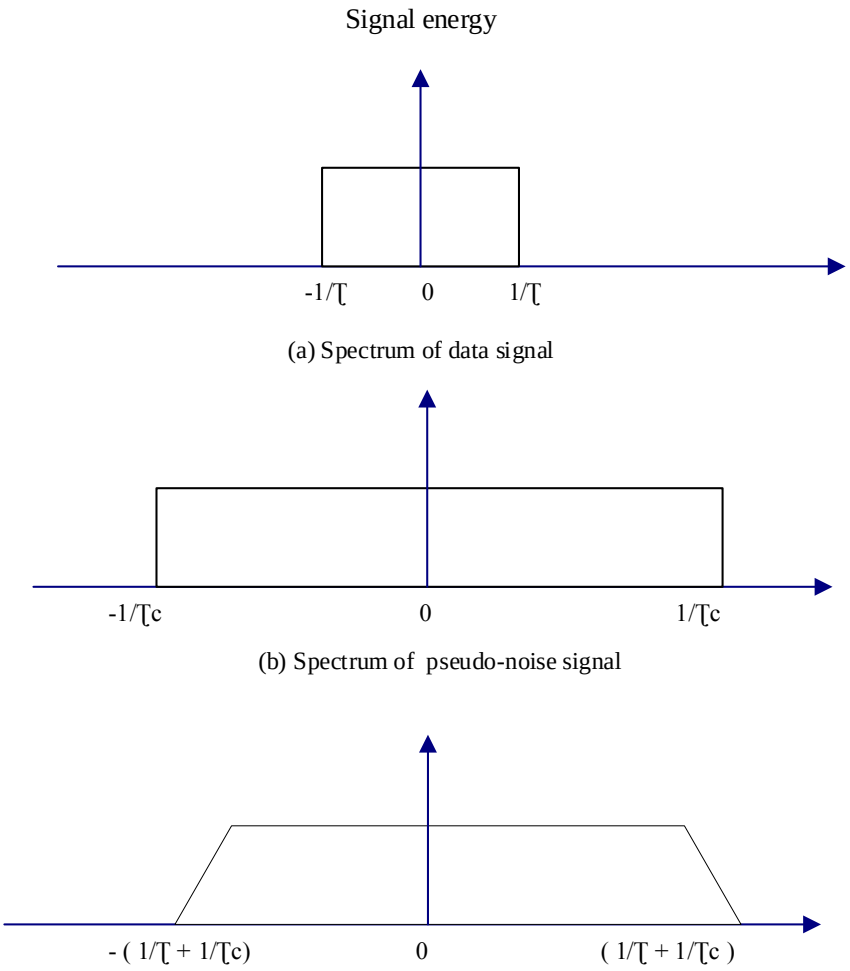


Figure 7: General model of SS digital communication system



(a) Spectrum of data signal

(b) Spectrum of pseudo-noise signal

(c) Spectrum of combined signal

Figure 8: Spectrum of DSSS

4.2 Frequency-Hopping Spread Spectrum

With this technique, the signal is broadcast over arbitrary series of radio frequencies jumping from frequency to another at specific interims. The recipient also hops between the same frequencies in synchronization with the transmitter to retrieve the original message as indicated in Figure 9 [63]. In FHSS method, channel bandwidth is partitioned into a large number of non-overlapping frequencies slots, specifically 2^k frequencies will form 2^k channels (frequencies slots). In any signal interval, the transmitted signal will occupy one or more of the accessible frequency slots using a PN generator. Spaces between the frequencies of the signal and the width of each channel commonly correspond to the bandwidth of the input signal. The sender transmits through one channel at a time for a specific period. For example, the IEEE 802.11 protocol utilizes a 300ms interim to transmit a number of bits using a specific encoding technique. A spreading code determines the sequence of the utilized channels, where both the sender and the recipient must utilize the same FHSS code in order to tune the sequence of channels in synchronization.

In the transmitter side, binary data pass into a modulator that converts it from a digital form to an analog form, this modulator could be binary-phase-shift keying or Frequency Shift Keying (FSK). The converted signal will be then centered on a specific frequency based on a pseudorandom number or a PN code that serves as an index for the table of frequencies as shown in Table 2, where k bits of PN forms 2^k frequencies and each k bits of PN refer to a specific frequency [65]. At each consecutive interval, new k PN bits are generated in order to select a new carrier frequency that will be modulated with the signal to create a new one with the same shape but centered on the selected frequency. In turn, the received signal will be demodulated with the same sequence of pseudo-noise codes to derive the frequencies that are required to retrieve the original signal. For a data rate of ρ, the time period needed to transmit one bit equal to T= $1/\rho$, while the required time to transmit any signal consists of (L) bits is $T_s = LT$. Furthermore, the needed time to change the frequency of a signal utilizing the FHSS technique is T_c, if T_c equal or more than

$T_s$then, the FHSS is considered to be a slow-FHSS, else it is known as a fast-FHSS. Typically using a large number of frequencies in the FHSS technique will improve the resistance to signal jamming and interferences.

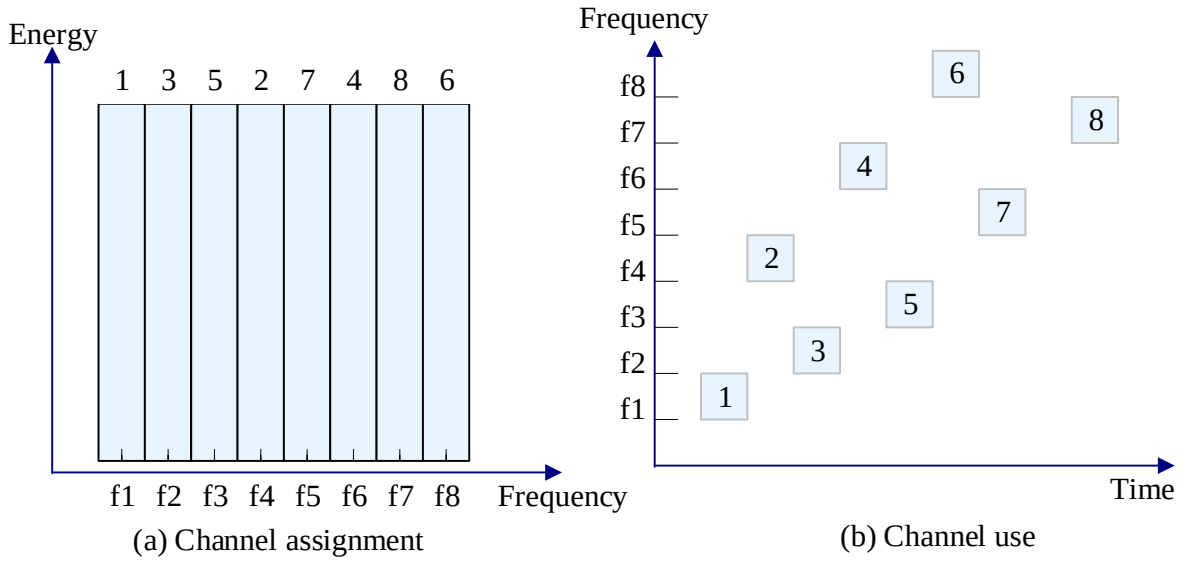
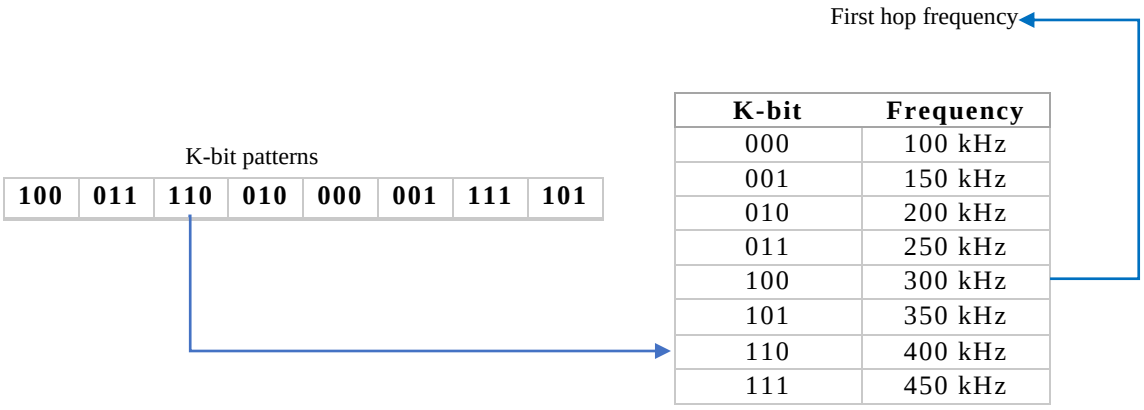


Figure 9: Frequency-hopping example

Table 2: Frequency hopping values based on PN code indices



K-bit	Frequency
000	100 kHz
001	150 kHz
010	200 kHz
011	250 kHz
100	300 kHz
101	350 kHz
110	400 kHz
111	450 kHz

4.3 Chirp Spread Spectrum

This technique is considered to be a good modulation choice for wireless communication systems since it has many capabilities, such as low power data transmission, strong rejection against signals interferences, and its simplicity to be implemented [66]. Unlike DSSS or FHSS, which employ coding techniques in the spread spectrum of a data signal, CSS does not demand any code in order to spread the spectrum. The frequency of a sinusoidal signal that is modulated via CSS is increased and decreased in a specific time duration . It also uses a pulse compression method to decode the information.

CSS technique is classified into two types; which are Direct Modulation (DM) and Binary Orthogonal Keying (BOK) [67]. DM technique relies on using chirps when it performs spreading and dispreading processes in the signal, while the data is modulated via a non-coherent modulation scheme, as it needs a digital modulator to send data. The second type of CSS, which is BOK, uses two distinctive chirps with the same duration and bandwidth, but with opposite polarity, which are, up-chirp and down-chirp, based on 0's and 1's bits. Both up and down chirps are used to represent data symbols, for instance, a bit '1' is used to represent the positive chirps while '0' bit is used to represent the negative instantaneous frequency change. At the receiver side, there is a matched filter to decode received signals [68].

4.4 Time Hopping Spread Spectrum

This technique is based on splitting the signal transmission period into '*N*' short time slots as shown in Figure 10, where $N=2^{n}$[7] [69] [70]. Through each frame, only a single time slot will be selected by the pseudorandom code generator to transmit a modulated data packet. Once the signal reaches the receiver side, it will be passed through electronically controllable switch, to be demodulated later on with the help of a bit synchronizer that is responsible for controlling the PN code generator to keep up synchronization with the received signal. Finally, the processed signal will be sent out through the storage and relocked unit.

[7] 'n' symbol represents the number of the transmitted bits per time slot in one frame.

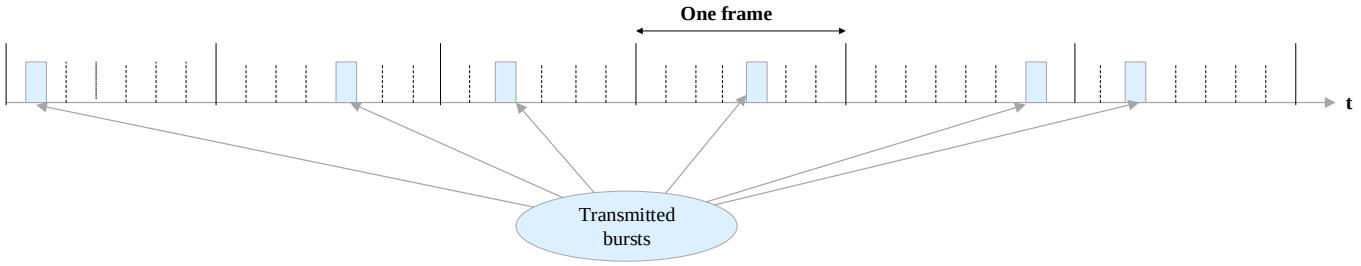


Figure 10: Time hopping spread spectrum system

5. IoT Stack Protocols (Origin, Recent, Future Enhancements)

Communication protocols are the proper descriptions of transmission, design, and rules of any digital message [71]. These protocols form the backbone of IoT networks as they enable them to be coupled and connected to smart services and applications. In addition, they allow smart things and devices to exchange their sensed data through these networks. The major functions of communication protocols are defining the following features; different addressing schemes of smart devices, transmitted data formats, data encoding, flow control, retransmission of lost packets ways, and routing process of IoT packets from source nodes toward destination nodes [27] [72].

IoT field is widely and swiftly spreading, where it comprises of a massive number of heterogeneous smart objects and power-constrained devices that are connected to IoT network with minimal storage and computing resources [73]. Based on that, IoT communication protocols face many challenges that should be taken into consideration while designing an IoT application, which are indicated as follows [74]:

- **Identification and addressing**: As billions of smart devices will be connected to the internet, each device must be identified via a unique address that permits it to communicate with other objects. Based on that, a large addressing space is required.
- **Low power communication**: Data exchanging process through devices is a power-consuming operation, especially in a wireless medium. Hence, a solution that facilitates communication among smart things with minimal power consumption is required.
- **Routing protocol** with minimum memory requirements and efficacious communication patterns.
- **Non-Lossy and high-speed communication**.
- **The mobility of the smart objects**.

Many classifications have been proposed to overcome the aforementioned challenges of IoT protocols. In this article, we follow the well-known classification that refers to the OSI model to describe the original, recent and future enhancements of each IoT layer protocols as given below:

5.1 Application Layer IoT Protocols

The application layer of IoT is in charge of determining suitable protocols and providing services that are required for message passing at the application level. Many factors should be taken into consideration when selecting proper communication protocol for a specific application, which are power consumption, required bandwidth, transfer and connection time, delivery guarantee, data security, and packet size. The following sections discuss IoT application layer protocols along with their recent and future enhancements, where Table 3 compares these protocols from different aspects and clarifies their advantages and disadvantages.

5.1.1 Original Application Layer IoT Protocols

(1) Message Queue Telemetry Transport: It is a lightweight protocol that was developed by Andy Stanford-Clark and Arlen Nipper in 1990. It enables the communication process between IoT devices and the network with middleware and applications in many forms such as M2M, server to server and machine to the server, and it works over the top of Transmission Control Protocol/Internet Protocol (TCP/IP) [75]. It supports also the communication over limited bandwidth and unreliable links. Hence, MQTT is used for publishing and subscribing operations to exchange lightweight messages, with a packet size that does not exceed 256MB, between clients and servers [76] [77]. Moreover, MQTT is suitable for operation in constrained devices with limited power and processing capabilities.

(2) HyperText Transfer Protocol: It is a web messaging and text-based protocol that was designed by Tim Berners-Lee in 1997, it also supports request/response Representational State Transfer Protocol (RESTful) functions, where the client transmits HTTP request message to the server [78]. HTTP depends on TCP as a transport protocol and Transport Layer Security/ Secure Sockets Layer (TLS/SSL) as a security protocol, which makes the communication between the server and the client connection-oriented. However, IoT communication over HTTP protocol causes the consumption of network resources and serious overhead as it requires transferring a lot of small packets [79].

(3) Extensible Messaging and Presence Protocol: It was developed by Homonym open source community in 1999 and was standardized by the Internet Engineering Task Force (IETF). XMPP supports low latency communication and small message transmission, which makes it suitable for many services such as video and voice calls, instant messaging, chats, publish-subscribe systems, gaming, and IoT applications. This protocol permits communication among heterogeneous applications due to its simplicity and flexibility. Nevertheless, it consumes the network bandwidth, needs high CPU capabilities, allows only the transmission of simple data type and there is no guarantee on the QoS [80] [81].

(4) Representational State Transfer Protocol: REST protocol is a set of best practices, rules, and constraints, where it was designed by Roy Fielding to provide web services that permit data exchange and communication among different devices and to build distributed hypermedia systems and provide them with desirable features such as modifiability and scalability. RESTful is based on HTTP protocol to support request-response and client-server models, which will allow the client to access server resources on IoT environments. However, RESTful Application Programming Interfaces (APIs) are considered to be a good choice for multiple IoT applications because they are lightweight and simple protocols [81] [82].

(5) Constrained Application Protocol: This protocol was proposed by IETF [83], to suit the communication among resource-constrained and unsynchronized devices, provide flow control, reliable delivery, and simple congestion control for IoT applications. It supports also the publish/subscribe communication model that is based on multicast and unicast requests. CoAP runs over User Datagram Protocol (UDP) because of its simplicity, having a small message size and a low code footprint, to manage resources, to reduce bandwidth requirements, and eliminate the cost of TCP handshake overhead before the beginning of transmission [84]. However, this protocol has many shortcomings as it increases communication latency, packet delivery corruption and it fails to transmit complex data [81].

(6) Advanced Message Queuing Protocol: It was developed by John O'Hara in 2003 to support a publish/subscribe architecture based on an efficient and reliable messaging queue. AMQP is widely used in commercial and business fields, as it supports reliable and secure communication between heterogeneous devices. Also, it runs over TCP protocol to guarantee more reliability. The process of transmitting data over AMQP consists of two steps which are; message queue and exchange queue. In the message queue model, the messages will be stored until they are transmitted to the receiver, while in exchange queue form the message will be routed in a suitable order [85].

(7) Data Distribution Service: It was developed by Object Management Group (OMG) and run over TCP/UDP transport protocols to achieve high performance, real-time, interoperable, scalable and dependable data communication based on publish/subscribe model. DSS is based on Peer-to-Peer (P2P) and decentralized communication, by a data-bus to enable asynchronous data transmission, which makes it a significant solution for IoT applications.

5.1.2 Recent Enhancements of Application Layer IoT Protocols

- Novel enhancements were applied on MQTT to launch MQTT v5.0, which has considerable amendments compared with the previous versions as the following [86]:
  - ✓ Allowing discovery functions: Inform the client with the maximum packet size and the maximum number of packets it can transmit at the beginning of the connection.
  - ✓ Better error reporting: Reason code has the responsibility of warning users if data is not transmitted successfully.
  - ✓ Shared subscriptions: By distributing messages evenly among the receivers for the sake of load balancing purposes, when the message rate of subscription is high.
  - ✓ Message properties: Define packets' properties and features through metadata at the header of the message.
  - ✓ Message expiry: It is an option to discard a message if it cannot be received within a predefined time.
  - ✓ Session expiry: Terminate client session, if it cannot be connected after a period of time.
  - ✓ Appropriate delay: Publish a message to the client, if it is disconnected more than predefined time. Also, notify clients about disconnections of their applications.
  - ✓ Topic alias: Representing messages topics by a single number, which will reduce message sizes.

- Designing RESTful IoT systems have many commonalities with other web applications, even though the primary characteristics that should be considered when building these systems are:
  - ✓ Interaction patterns, data formats and other approaches that avoid or reduce the need for human intervention.
  - ✓ Preferring simple and compact data formats to ease the transmission and processing over constrained networks.

  However, many aspects of RESTful protocol need to be improved to enhance its capabilities as follows [87] :
  - ✓ 3-way commit, because of robust and unreliable communication in high packet loss networks.
  - ✓ Sharing knowledge methods between system components, such as media types, well-known locations, uniform resource identifiers schemes, and relation types.
  - ✓ Further information on choosing what is modeled as a resource and how to select resources.

- The main objectives behind enhancing CoAP capabilities to run over TLS and TCP, are that some enterprise networks face connectivity issues compelling them to block UDP packets, while the second target is to gain many desirable features of TCP protocol such as [88]:
  - ✓ Nating over TCP lasts for a long period compared with UDP, as it provides additional information regarding session lifecycle. Thus, timeout binding for TCP is 386 minutes, while it does not exceed 160 seconds for UDP protocol. Generally, the shorter timeout of UDP requires to transmit the keepalive messages more frequently compared with TCP protocol.
  - ✓ TCP uses techniques for flow control and congestion control that are more advanced than those provided by UDP, which allow CoAP to transmit larger payloads.

  However, there are numerous hindrances of using CoAP over TCP, as it requires more round trips, large code and packet sizes, and more RAM requirements.

- AMQP v2.5.0 has added a new platform to the previous version of the protocol, dropped Python 3.4, and fixed numerous bugs. In addition to the above, the motivation behind launching a novel AMQP v.2.5.0 protocol is the need of scaling hundreds to thousands of subscribers and publishers in a reliable and flexible manner [89].

- DDS protocol specifies the communication semantics (QoS and behavior) and APIs that permit robust and efficient data transmission to the right place at the right time. Therefore, it is important to design the interfaces in a way that meet the above requirements as follows [90]:
  - ✓ Permitting the middleware to dynamic pre-allocate resources to be at the minimum.
  - ✓ Evading features that require using of hard-to-predict or unbounded resources.
  - ✓ Reducing the need for making copies of data.

5.1.3 Future Research Directions of Application Layer IoT Protocols

- HTTP protocol supports a wide range of internet services. A novel version (HTTP/3) is proposed to suit running over Quick UDP Internet Connections (QUIC) protocol. QUIC tries to enhance HTTP performance by incorporating TLS v1.3 security procedure. HTTP/1.1 runs over TCP protocol and utilizes whitespace-delimited fields to transmit HTTP texts, where multiple TCP connections are required since one HTTP response or request can be transferred at a time in each direction. HTTP/2 presents a new layer multiplexing and binary framing layer in order to enhance network latency without any modification in the transport layer. Nevertheless, the parallel multiplexing nature of HTTP/2 makes it prone to packet reordering or loss. HTTP/3 is intended to support transporting over QUIC protocol and internal framing layer to benefit from their features [91].
- XEP-0128 is a service discovery extension for XEP-0030 protocol which does not have an option that allows users to add a service description attribute. Adding an additional attribute to service discovery schema does not solve this issue, so it is better to include additional information that provides a method to resiliently specify data structured formats [92].

Table 3: Comparison between application layer IoT protocols

Protocol	**MQTT**	**HTTP**	**XMPP**	**RESTful**
Year	1999	1997	1999	2000
Standard	OASIS[8], Eclipse Foundations	IETF, W3C[9]	(RFC 3920-RFC 3923) RFC 4622, RFC 4854, RFC 4979, RFC 6122	IETF
Latest Version/year	MQTT version 5.0 (2018) [93]	HTTP version 3.0 (2018) [91]	XMPP v 1.0.1 XEP-0128 (2019) [92]	RESTFUL (2018) [87]
UDP/TCP	TCP	TCP	TCP	TCP
Architecture	Publish/Subscribe	Request/Response	Publish/Subscribe Request/Response	Request/Response
Semantics/ Methods	Connect, Disconnect, Publish, Subscribe, Unsubscribe, Close	Get, Post, Head, Put, Patch, Options, Connect, Delete	Get, Post, Put, Set, Result	Post, Put, Delete, Get
Security	TLS/SSL	TLS/SSL	TLS/SASL[10]	TLS/SSL
QoS options	Yes	Limited	No	Yes
Caching	Yes	Yes	Yes	API calls can be cached
Performance	Needs low power requirements	High latency	Traffic overhead	Requires fewer resources
Message format	Plain-text	Plain-text, Textual information encoded in ASCII	Chatting, message exchanging.	Plain-text , XML[11], HTML YAML[12], JSON
Merits	• Suitable for resource-constrained devices • Suitable for high latency and low bandwidth networks • Simplicity • Very small message header	• Persistent connections • Request pipelining • Chunked transfer encoding • High interpretability on the web	• Decentralization can be run by anyone on any server and there is no central master server • Open standards Flexibility (Custom functionality can be built on top of XMPP)	• Scalability • Easy implementation and interaction • Browser-friendliness • Flexibility • Independence of programming language and platforms
Demerits	• It does not support encryption • Needs more efforts in security	• Requires high power and resources • Increases communication latency • Consumes network bandwidth • Does not include reliability	• Does not support QoS • High network overhead • In-band binary data transfer is limited	• Less security • Not suitable for distributed environments
References	[75] [76] [77] [94] [95] [96]	[78] [79]	[81] [92]	[35] [81] [82] [97] [98] [99]

Table 3: Comparison between application layer IoT protocols (Cont.)

Protocol	**AMQP**	**DDS**	**CoAP**
Year	2003	2001	2010

[8] OASIS: Organization for the Advancement of Structured Information Standards
[9] W3C: World Wide Web Consortium
[10] SASL: Simple Authentication and Security Layer
[11] XML: Extensible Markup Language
[12] YAML: Yet Another Markup Language

Standard	OASIS, ISO/IEC[13]	OMG	IETF, Eclipse Foundation
Latest Version/year	AMQP v 2.5.0 (2019) [89]	DDS v.1.4 (2015) [90]	RFC 8323 (2018) [88]
UDP/TCP	TCP	TCP/UDP	UDP
Architecture	Publish/Subscribe	Publish/Subscribe Request/Response	Publish/Subscribe
Semantics/ Methods	Consume, Deliver, Publish, Get, Select, Ack, Delete, Nack, Recover, Reject, Open, Close	Write, Read, Take, Dispose, Wait	Post, Put, Delete, Get CON(Confirmable), NON (non-confirmable), ACK (Acknowledgement), RST (reset)
Security	TLS/SSL, IPSec[14], SASL	TLS/ DTLS	DTLS[15], IPSec
QoS options	Yes	Yes	QoS by 4 types of messages: Confirmable, Non-Confirmable, Acknowledge, Reset
Caching	Yes	Yes	Yes
Performance	Efficient in the environment that does not have any restriction in network bandwidth, power, latency, and processing capabilities	Efficient in the application that requires low latency and high bandwidth	Sufficient for constrained environment and networks
Message format	Binary encoded	ASCII characters, Binary encoded	Binary encoded
Merits	• Scalable • Supports the communication between heterogeneous devices • Supports reliability, security, and performance	• Supports durability, security, and priority QoS standards • Achieves high performance, real-time, interoperable, scalable and dependable data communication	• Reliability • Retransmitting lost packets • Multicast support • Resources monitoring • Low overhead • Simplicity for constrained environments
Demerits	• It is not suitable for real-time and resource-constrained environments • It does not support automation discovery procedure • Heavy protocol as it requires memory and power resources	• Consumes high bandwidth	• Multicast communications are less secure, as there are no suitable key management procedures • An end to end security is not supported • It does not contain built-in security characteristics
References	[35] [84] [85] [94] [100]	[35] [94] [101]	[35] [83] [84] [94] [100] [102]

5.2 Transport Layer IoT Protocols

This layer is also known as routing layer since it is responsible for routing data packets through the network area, where its protocols are in charge of ordering packets, error detection, and correction [103]. The following sections describe the main transport protocols that are utilized in IoT environments with their enhancements and future works, where Table 4 provides a comparison among these protocols from different characteristics.

5.2.1 Original Transport Layer IoT Protocols:

(1) Transmission Control Protocol (TCP): It is a heavyweight and connection-oriented protocol, which means that the connection must be established until all the required data have finished exchanged between each end device. This makes TCP suitable for reliable communications, as it needs acknowledgment message to guarantee each sending and receiving process, supports retransmission of lost or corrupted packets and provides a flow-control mechanism. Consequently, the packet overhead will be very large in this protocol, which will lead to more power consumption from devices and hence, making it not suitable to operate in power-constrained devices. TCP breaks down the data packet into multiple packets, where each packet has an ordering number with source and destination IPs [104].

(2) User Datagram Protocol (UDP): It is a connectionless protocol that aims to provide unreliable, minimal message queueing, message passing and best-effort transport to protocols and applications which operate over IP. There is no need to establish end to end connection between the communicating entities, which in turn will offer a very efficient communication for some applications that require real-time performance with low latency such as video and voice [105]. Moreover, there is no guarantee on data packets ordering, duplicate, delivery or protection. On the other hand, UDP provides a port number attribute to address functions of source and destination, as well it provides a checksum for data integrity.

(3) Datagram Congestion Control Protocol (DCCP): It provides unicast bidirectional connections of unreliable dynamic congestion-controlled datagram. These features make DCCP suitable for the applications that transmit massive amounts of data and the applications that tradeoff between reliability and timeliness, such as Voice over Internet Protocol (VoIP) and media streaming [106]. The flow rate in DCCP can be adjusted gradually since it is unreliable and lacks a receiving window [107].

(4) Stream Control Transmission Protocol (SCTP): It is a connectionless, message-oriented and IP transport layer protocol like UDP that was designed and launched in 2007 by Stewart [108].On the other hand, SCTP provides connection-oriented P2P, reliable transmitting for the applications that are communicating over an IP. Thus, it inherits most of the TCP features including the recovery of the lost packets and congestion control [109].

(5) Transport Layer Security: It runs on the top of many transport layer protocols, as it was designed to provide secure channels among the communicating peers and to provide authentication, data confidentiality, data integrity and encryption to the

[13] ISO/IEC: International Organization for Standardization/International Electrotechnical Commission

[14] IPSec: Internet Protocol Security

[15] DTLS: Datagram Transport Layer Security

applications, by preventing eavesdropping, message forgery, and tampering. It consists of two components, where the first component is the handshaking protocol that has the responsibility of authenticating the communication ends, agreeing on shared keys and negotiating the cryptographic parameters and modes, where the second component is the record protocol that splits the traffic into many records and protects them by utilizing the traffic keys [110] [111].

(6) **Datagram Transport Layer Security:** It was designed to provide security for datagram applications that do not require or provide in-order or reliable data delivery such as datagram online gaming, internet telephony and media streaming, which are considered to be delay-sensitive applications. DTLS is an extension of TLS protocol, where it provides the same security functionalities but for data stream transmission by preventing message forgery, tampering, and eavesdropping. Thus, it should deal with and solve many datagram issues, such as loss of datagram packets, packet reordering and delay [112].

(7) **Resource Reservation Protocol (RSVP):** It is multicast and unicast control transmission protocol that was designed to provide flexible, robust, scalable and heterogeneous resources reservation setup at each router for data stream transmission. RSVP organizes message formats, hosts and routers mechanisms, also it can operate over IPv4 or IPv6 [113]. It also supports many functionalities such as resource reservations in each node along the data path, multipoint to multipoint communication paradigm, cache (state) management routers and receiver-initiated reservation [114] [115].

(8) **Quick UDP Internet Connections:** It is a general-purpose, secure and multiplexed transport protocol. Quick was built on the top of UDP protocol by google to provide reliability, security, multiplexing, flow control per-stream, congestion control per connection, low latency for data stream transmission, and connection migration to NAT rebinding [116]. This protocol aims to improve the performance of connection applications, which are based on TCP protocol through established multiplexed connections over UDP [117].

(9) **Aeron**: It is an open-source connection-oriented communication protocol that was proposed by Martin Thompson to run over unreliable media such as InfiniBand and UDP, as well to provide in order transmission with optional reliability through retransmission of dropped packets. Aeron tries to provide the highest throughput with the lowest latency, which makes it ideal for the communication of real-time applications, VoIP, fast-paced networked multiplayer games, video streaming, and high-frequency financial trading. However, implementing this protocol by java language will reflect on reducing resource requirements such as memory and CPU [118] [119].

5.2.2 Recent Enhancements of Transport Layer IoT Protocols

- TLS v1.3 has improved the major specifications of the original protocol (TLS) as the following [120] :
  - ✓ New encryption techniques were proposed and work only with the newest versions of TLS.
  - ✓ A zero round trip mode was proposed, so the data transfer session cannot be started until the handshaking process is completed.
  - ✓ After receiving the ServerHello packet all of the handshaking messages have to be encrypted.
  - ✓ The handshake state machine was reconstructed to eliminate unnecessary messages and to be more consistent.
  - ✓ Prevent renegotiation when the connection in TLS v1.3 has been established.
  - ✓ Using RSA[16] probabilistic signature scheme instead of RSA padding, besides removing DH crypto groups and digital signature algorithms.

- RSVP protocol was proposed to transform unidirectional Label Switch Path (LSP) connection into a bidirectional connection, either by single-sided or by double-sided method, by following the same path. RSVP-Extended (RFC 8537) amends single-sided and double-sided methods to support fast reroute and co-routed procedures. Fast reroute methods make sure that the traffic of LSP flows smoothly via co-routed paths in both directions after it transmits through the fast route. However, to implement RFC 8537 standard successfully, all the nodes on the LSP path should support this protocol [121].

5.2.3 Future Research Directions of Transport Layer IoT Protocols

- TCP is a significant transport protocol that has been continuously improved since 1981. Over this time, many piecemeal changes have been done to suit tremendous numbers of internet applications and to fix many errors and hindrances in many aspects such as performance and security. TCP provides byte stream service, in-order and reliable delivery of data segments over the network as IP datagram. Achieving data reliability is done by detecting packets errors through segment checksum, or detecting packet loses through sequence number. Also, it supports a connection-oriented unicast or anycast transmissions. Many issues should be considered to be solved in the future, such as IP security precedence and compartment, validation of sequence number, Nagle algorithm (small packets buffering) modification and low watermark function usage.

- Many transport protocols extend their capabilities by dedicating an area for header options, which will adapt the protocol to be used in particular environments or in unexpected conditions that have not been seen by the developers. UDP is one of the popular transport layer protocols that lack this feature. Thus, UDP-Options-07 comes to extend UDP header to locate a trailer space for options after the data payload field [122].

- Transmission over SCTP has faced many issues and hindrances from the first launching till now. RFC 8540 presented the improvements that have been made to handle these issues, such as path error, counter threshold, shutdown request of the upper-layer protocols, new chunk types registration, detection of endpoint failure, identifying the rules of data transmission, miscellaneous typos, etc [123].

[16] RSA: Ron Rivest, Adi Shamir and Leonard Adleman

- Communication through DCCP is currently limited on one path per connection, even though multipath connection only exists among peers. Improves DCCP capabilities to support the use of simultaneous multipath communications, will reflect positively on enhancing network resources usage through applying load balancing techniques, providing flexibility to face the network failure and improving the network throughput [124].

- DTLS v1.3 has been evolved to allow a secure client/server communication over the internet by implementing the following [125]:
  - ✓ A new handshaking form has to be proposed to support short message exchange.
  - ✓ Legacy and weaker cryptographic algorithms ought to be removed.
  - ✓ Supporting authenticated encryption with associated data ciphers.
  - ✓ Encrypting sequence numbers.
  - ✓ Adding connection ID functionality.
  - ✓ Optimizing sizing and encoding of the record layer.
  - ✓ Providing elastic cryptography method negotiation.
  - ✓ Redefining a new method for phase-shift keying authentication.
  - ✓ Proposing a new session resumption procedure.

- QUIC v.1 is an enhanced version of QUIC protocol that aims to be utilized over UDP, which will evade the need to change the middleboxes and the operating systems of clients by applying data encryption and headers authentication techniques [126].

Table 4: Comparison between transport layer protocols considering different aspects

Protocols	**DTLS**	**RSVP**	**QUIC**	**Aeron**
Standard	RFC4347	RFC 2205	gQUIC	Aeron
Latest version of protocol\ Year	DTLS v.1.3 (2019) [125]	RFC 8537 (2019) [121]	QUIC v.1 (2019) [126]	*
Flow control	*	Yes	Yes	Yes
Congestion control	*	Yes	Yes	Yes
Packet size	2^{24}-1 bytes (handshake message)	16 bits header	2- and 19-bytes header for wire connection	32 bytes
Transport packet entity	Datagram	Datagram	QUIC packet	Frame
Error detection	Yes	Yes	Yes	Yes
Reordering and sequence numbering	Yes	Yes	Yes	Yes
Reliability	Yes	Yes	Yes	Yes
Port Numbering	*	Yes	Yes	*
Merits	• Provides security for datagram transmission • Provides reliability for handshake • uses retransmission timer to reduce the probability of packet loss • Queues unordered messages	• Data Integrity • Error reporting • Permits multicast communications among heterogeneous devices • QoS routing can be deployed separately from data	• Built-in performance and security, as it has many security functions such as encryption and authentication • Processing many requests and transmission concurrently with one handshaking • Low packet loss • Minimize bandwidth consumption	• Tries to attain high throughput with low latency for both unicast and multicast communications • Affords reliable multicast operation • Provides different QoS degree based on data stream type
Demerits	• Cannot provide protection for SCTP control chunks • DTLS over SCTP is slower • When the collision occurs, DTLS will process only the packets from the first source and discards the others	• Requires a lot of work on the router's side to manage resources reservations • Puts heavy processing load on routers especially in a heavy traffic case, which will degrade their performance • Soft state requires many refreshments • Scalability issue	• Performance problem of the data transmitting and receiving • Information exposure when using long header	This protocol on its infancy stages
References	[112] [127] [128]	[114] [129] [130] [131]	[117] [116] [132] [133]	[118] [119] [134]

Table 4: Comparison between transport layer protocols considering different aspects (Cont.)

Protocols	**TCP**	**UDP**	**DCCP**	**SCTP**	**TLS**
Standard	RFC793	RFC768	RFC4340	RFC4960	TLS 1.0 (RFC2246)
Latest version of protocol\ Year	RFC793bis-14 (2019) [135]	Transport Options for UDP (2019) [122]	Multipath DCCP (2019) [124]	RFC8540 (2019) [123]	TLS v1.3 (RFC8446) (2018) [120]
Flow control	Yes	No	No	Yes	*
Congestion control	Yes	No	Yes	Yes	*
Packet size	20-40 bytes header	8-bytes header	12 or 16-bytes header	12-bytes header	5-byte header
Transport packet entity	Segment	Datagram	Datagram	Datagram	Runs over Segment
Error detection	Yes	No	Yes	Yes	Yes

Reordering and sequence numbering	Yes	No	Yes/No	yes	Yes, by MAC
Reliability	Yes	No	No	Yes	Yes
Port numbering	Yes	Yes	Yes	Yes	Yes
Merits	• Supports most of the applications that run over the internet • Improves the performance and robustness of varying quality and capacity networks	• There is no guarantee on packets delivery • Packets may arrive out of order • No flow-control • High packet loss • There is no startup latency	• Eliminates the delay that can occur when waiting packets arrive out of order • Supports various delivery modes such as strict, partial and unordered delivery • Multi-homing support as it can send a message to the same destination, but it can reroute it to another IP, if the previous IP is unreachable • Enables congestion control techniques	• Provides flexibility for VoIP applications that need reliable message data transmission • Supports Multihoming method • Supports additional security features, which minimize denial of service attacks	• Prevent tampering by intruders • Prohibit passively listening by attackers
Demerits	• It is not suitable for real-time and synchronous applications • It gives strict order of the data that is delivered between hosts • It cannot continue a transmission if a specific sequenced packet has not been received and acknowledged yet	• All broadcast and multicast transmissions are unreliable in UDP • Retransmission is required when there is a corrupted data	• Unreliable transport protocol, which affects visual quality of the video streaming and QoS performance • It lacks receiving window	• Network address translation problem when using multi-homing function • Dynamic IP addressing issue, especially in multi-homing function • The transmission process of the line will be blocked until the head of the stream is received and acknowledged	• Adding more latency • Handshaking process consumes resources • Complicates the configuration managements
References	[109] [136]	[137]	[107] [138]	[109] [136] [139] [140] [141]	[110] [142] [143]

5.3 Network Layer IoT Protocols

This layer has the responsibility of forming, addressing and routing data packets, as it receives the datagram packets from the transport layer and transfers them into data packets form to be then transmitted to the destination side. The following subsections discuss the common routing protocols that are broadly utilized in data packets transmission along with their future improvements, where Table 5 compares these protocols from distinctive aspects.

5.3.1 Original Network Layer IoT Protocols

(1) Routing Protocol for Low-Power and Lossy Network (RPL): It is a tree-based, IPv6 proactive, distance vector routing protocol that was designed by routing-over-low-power-and-lossy-networks working group in 2012 to run over lossy and low power commercial appliances networks, where their interconnections are characterized by instability, low data rates, and high loss rates [144]. RPL structs the network topology into Destination Oriented Graph (DAG) that consists of Destination-Oriented Directed Acyclic Graph (DODAG). Every DODAG represents a routing tree that is constructed by a root (sink) node. To create optimal routes of DODAG, RPL utilizes an Objective Function (OF), which is calculated based on routing metrics [145]. The first step of creating DODAG begins by transmitting the DODAG Information Object (DIO) message, which consists of node rank, OF, mode of operation and metric, by the root node to all other neighbors. Consequently, nodes that receive DIO messages will decide to join DODAG or not based on OF. The joining nodes will compute their rank, determine the upward route toward the root node, opt their preferred parents and refresh their neighbor tables. If the node sends a DIO message it will become a router, else it will be a leaf [146] [147] [148] [149].

(2) Cognitive Routing Protocol for Low-Power and Lossy Network (CORPL): It is an extension of the RPL protocol, where it was designed to suit cognitive network and is based on DODAG topology in routing generation with novel modifications. CORPL uses an opportunistic forwarding mechanism allowing it to select the optimal forwarder from a set of eligible neighbors to be the next hop for data transmission. In this approach, each node maintains a set of forwarders instead of one parent and updates its set based on the receiving DIO messages [150].

(3) Channel-Aware Routing Protocol (CARP): It is a distributed protocol that was designed for underwater and IoT applications because of its lightweight data packet. CARP considers link quality to opt the forwarder nodes, according to the successful data transmission that occurred by neighboring sensors. The routing operation of CARP consists of network initialization step and data forwarding step. In the first step, the sink node broadcasts hello messages containing its ID along with the hop count to enable the receiving node from updating its distance toward the sink node. In the data transmission step, the sender broadcasts a ping message to its neighbors to choose the optimal relaying node based on the link quality and the information that it receives from pong messages by them, in order to forward data through the optimal node [151].

(4) Collection Tree Protocol (CTP): It is a tree-based routing protocol that was designed to provide the best effort for anycast communication in low energy demands networks. In the beginning, some nodes advertise themselves as root nodes (sink nodes),

where data is delivered to the root with minimum cost. Other nodes will connect to the root tree through beacon advertisements, then send their collected data to the next hop toward the sink node based on the minimum Expected Transmission Count (ETX) cost of their neighbors. However, CTP does not support reverse routing from the sink node to sensors [152] [153].

(5) Lightweight On-Demand Ad Hoc Distance-Vector Routing Protocol-Next Generation LOADng: It is a lightweight distance-vector and reactive protocol that is derived from On-demand Distance vector (AODV) protocol to enable secure, scalable and efficient routing in lossy and low power networks. As a reactive protocol, there is no routing table for the routes to all destinations. Thus, LOADng generates on-demand route requests to discover a path to the destination node, when there is generated data required to be sent, until receiving unicast reply hop by hop from the destination node back to the sender node. If broken is detected in the route, then attempts to repair is applied or an error message will be directed to the requested node [154].

(6) An Efficient Routing Protocol for Emergency Response Internet (ERGID): It aims to provide reliable data transmission and efficient emergency response for IoT applications. ERGID selects the optimal route toward destination considering global latency estimation and the residual energy of the candidate route nodes. The first procedure is called delay iterative method, and it tries to alleviate the problem of disregarding valid routes, update routing tables periodically and ensure real-time communication for the emergency-response-applications. Whereas, the second procedure is called residual energy probability choice [148] [155].

(7) Parent Aware Objective Function (PAOF): It is an objective function proactive protocol that tries to achieve load balancing by employing parent count and ETX metrics in route selection for data transmission. To select the desired route, PAOF first computes the difference between the ETX of the candidate nodes, in case if it is smaller than predefined value called MinHopRankIncrease[17], then it will compare between the parents count and consequently select the least value as the preferable route [148] [156].

(8) Geographic Routing Approach for The Ipv6-Enabled Large-Scale Low-Power and Lossy Networks (GeoRank): It is a hybrid approach that integrates the rank-based behavior of RPL protocol with geometric-based behavior of greedy other adaptive face routing protocol, to be implemented in large-scale networks that have a non-uniform link density, in order to enhance P2P communication over 6LowPAN and to minimize the number of control packets. In this protocol, each node in the network area is declared by its position and must be aware of other nodes' positions. Initially, GeoRank computes the distance between the source node and the destination according to the list of DODAG root, to choose the anchor root that gains the lowest absolute angle difference between the source and destination path. Then, the protocol tries to forward the data to the neighbor that is located one hop from the destination based on a greedy forwarding algorithm. If there is no node achieves this condition, then GeoRank mode will be applied to forward the data to the preferred parent in the path to be sent then to the anchor node until it reaches its destination. The proposed algorithm is only applied for down routes, where RPL is performed to discover and reach upward routes [148] [157].

(9) Ad-Hoc On-Demand Multipath Distance Vector for IoT (AOMDV-IoT): It aims to discover and create a connection among nodes and the internet. AOMDV-IoT creates two routing tables for each node, which are Internet Connecting Table (ICT) and routing table. Also, it transforms IP address into Internet Linking Address (ILA). Once a node requests to be connected to the internet, the required IP will be converted into ILA in order to facilitate searching through ICT, which provides the source node with a suitable internet node. In case, if there is no internet node in ICT, then the source node will broadcast a requested packet to update both tables until it finds the optimal route toward an internet node [158].

5.3.2 Recent Enhancements of Network Layer IoT Protocols

- RPL routing protocol is not applicable for Mobile Nodes (MNs) of dynamic networks, as it cannot deal efficiently with MNs disconnections, data losses, routes reliability and real-time applications. Applying mobility detection using RPL is based on the absence/reception of DIO messages and that means more control overhead, which will consequently lead to more delay, more power consumption, more collisions, and data losses. Energy and Mobility Aware Routing protocol (EMA-RPL) was proposed by Bouaziz et al to deal with the aforementioned issues of real-time IoT applications, wherein the data is transmitted via MNs. In EMA-RPL protocol MNs must join the DODAG tree by choosing the Preferred Parent (PP) depending on specific OF, while static nodes are connected to PP using a proactive process. This will reflect on reducing or avoiding the data loss and the disconnection time during the network recovery process. To avoid route interruption, EMA-RPL excludes MNs from the route path selection and prevents them from sending periodic DIO messages to preserve their energy. A new node role is proposed by EMA-RPL protocol to preserve network resources and to achieve load balancing among nodes, called Associated Node (AN) to be connected with MN. AN is in charge of detecting any movement of a MN using the Received Signal Strength Indicator (RSSI), data transmission from or to the MN and looking for a new AN for the MN. Future refinements are required because using RSSI in the prediction process is not efficient especially in the presence of obstacles or in closed environments [159].

- Zhou et al proposed an enhanced version of CARP (E-CARP) protocol, which aims to provide an efficient energy routing protocol in the underwater wireless sensor networks. To achieve this end, E-CARP employs many techniques as follows:
  - ✓ Instead of transmitting the sensed data toward the sink node by the same sensor each timepoint, E-CARP just permits caching the received data to reuse it by the sink when needed. Precisely, if the bias in data is within a certain range, the sensor node transmits only small (INFORM) control packets rather than large data packets, which consequently improves the network capacity and reduces the energy consumption.
  - ✓ There is no need to periodically select a relay node for each source node if the network topology is stable, this will improve the network lifetime by reducing the number of control overheads.

[17] MinHopRankIncrease: It is a parameter defined in the DIO control message of RPL protocol [144].

However, E-CARP distinguishes and prioritizes data based on its importance, as the data with the high priority should firstly be routed to the base station. Moreover, sensed data may change based on temporal or/and spatial discipline. The sensed data that are gathered at earlier time points by some nodes might be used in some applications, instead of fetching instantaneous data [160].

- Extend Collection Tree Protocol (XCTP) was proposed as an extension of CTP. CTP maintains a routing tree that affords paths in one direction from sensor nodes toward root (base station) node only, while XCTP solves this issue through allowing communication in both ways from node to root and root to node requiring low overhead and few memory storages. Finding routes to the reverse path (from root to nodes) requires transmitting acknowledgment packets and feedback commands to guarantee reliable data delivery [152].

- Expected Life Time of Energy-Aware Parent Routing (ELT-EAPR) protocol tries to select the optimal route to the base station node based on parent event rate and residual energy through utilizing a sigmoid neural network predictor, which will enhance the network lifetime [161].

- LOADng protocol requires many enhancements as it faces many issues such as determining all the nodes that are responsible for providing internet connections to other network nodes, also the creation of the on-demand routes leads to a massive number of control overheads. As a result, LOADng-IoT protocol tries to improve the route discovery process, enhance the network QoS, and minimize the number of control overheads by employing the following amendments [162]:
  - ✓ Finding Internet Connected Nodes (INs) without any prior knowledge of their addresses in the local network, by broadcasting a special RREQ-IoT, so any intermediate node knows an IN will send unicast RREP message to the originator node. However, the prior knowledge of INs causes several issues such as the INs can be overloaded by the messages from other network nodes, network nodes may be configured in long paths toward INs, and packets may be lost if INs are disconnected from the internet.
  - ✓ Internet route cache is responsible for storing information about the routes toward INs, which will reduce both delay time and control overhead packets. It is worth mentioning that this procedure is optional and based on device capabilities.
  - ✓ A novel error code to evade the loss of data by informing the other nodes about any internet connection loss, which will allow them to find a new IN they can relay their data through in order to increase the successful delivery ratio.

Table 5: Comparison between network layer protocols considering distinctive aspects

Protocol	**RPL**	**CORPL**	**CARP**	**CTP**	**LOADng**
Standard	RFC6550	*	*	*	*
Recent protocol (year)	EM-ARPL (2019) [159]	[163]/ 2019	E-CARP (2015) [160]	XCTP (2016) [152]	LOADng-IoT (2019) [162]
Network topology	Mesh, hierarchical based on DAG	Cognitive M2M networks, mesh	*	Tree-based topology, Mesh	Grid
Scalability	Yes	Yes	Yes	Yes, by beacon message	Yes
Applications	Building automation, home, industrial, Smart Grid, Smart Cities	Smart grid	Underwater WSNs applications	Commercial products, industrial WSNs, teaching, research	Home applications, industrial applications
Routing metrics	Bandwidth, reliability, hop count, number of transmissions, connectivity, link quality	Expected Transmission (ETX) value, reliability, collision risk, delay	End-to-end packet latency, energy consumption per bit, buffer spaces, packet delivery ratio	ETX	Hop-count
Multi-hop routing	Yes	Yes	Yes	Yes	Yes
Consider link quality	No	Yes	Yes	Yes	Yes
Traffic flows	MP2P[18], P2MP[19] or P2P	MP2P, P2P, P2MP	MP2P, P2MP, P2P	MP2P, P2MP	P2P
Algorithm	Distance vector	Distance vector	Link state	Distance vector	Distance vector
Data rates	Low data rates	Low data rate	low data rate	Low traffic rates	
Mobility of Network	No	No	Supported	Yes	Yes
Proactive [20] or Reactive [21]	Proactive	Proactive	Reactive	Both	Reactive
Security	Not supported	Not supported	Not supported	Not supported	It uses integrity check value, timestamp
Buffering	Limited buffer size	Yes	Yes	Yes	Limited buffer size
Latency	High latency	Supports the delay of sensitive applications	Low latency	High latency	High latency
Simulation tool	Contiki/Cooja	Cooja	Real-Time Test-bed, NS2	nesC C, Java, C++, TOSSIM	NS2, Tmote Sk, Cooja
OS to implement a protocol	Contiki, LiteOS, TinyOS, T-Kernel, EyeOS, RIOT	Contiki OS	SUNSET	TinyOS, Mantis OS, Sun SPOTs, Contiki OS, Linux/Click	Linux kernel, Contiki

[18] MP2P: Multipoint-to-Point communication
[19] P2MP: Point-to-Multipoint communication
[20] **proactive protocol**: Each node builds its routing table based on the entire topology of the network, and updates it regularly to get up-to-date routing paths to other nodes.
[21] **Reactive protocol:** The routes are created when source node wants to communicate with a destination, it recalls route discovery technique to look for a path toward destination.

Merits	• Supports routing in limited resources environments • Supports storing and non-storing mode to reduce memory requirements • Avoids loops	• Achieves good packet delivery ratio • Minimum collisions • Improves the performance in spectrum sensing state	• Considers residual energy, link quality and buffer space when choosing relaying node	• Achieves high delivery data ratio when transmitting from sensors to sink node	• Generates control traffic to construct a route, when there is data transmission only • Finds a bi-directional path for any destination in the network
Demerits	• Susceptible to high packet loss due to congestions • High delay • Susceptible to attacks as it does not support end-to-end encryption • Floods the network with control over had packets	• Takes a long time for DAG convergence in high node density networks • Retransmissions of duplicate data packets	• No security • No reusability of previously collected data • Control packets increase communication cost, which will consequently increase the consumed energy of the network	• Adaptive beacons consume more bandwidth and energy • Does not support routing from sink toward sensors • There is no guarantee on data delivery • Routing changes could lead to loops	• Prone to data packets loss due to collisions • There is no policy to protect the network confidentiality • Data transmissions consume a lot of energy, which will reduce the lifetime of nodes • Route discovery delay • It does not consider the constraints of the nodes, which will reduce the network's lifetime.
References	[144] [148] [164] [165] [166] [167]	[168] [169] [170]	[151]	[171] [172] [173] [174]	[175] [176] [177]

Table 5: Comparison between network layer protocols considering distinctive aspects (Cont.)

Protocol	**ERGID**	**PAOF**	**GeoRank**	**AOMDV-IoT**
Standard	*	*	*	
Recent protocol (year)	*	ELT-EAPR (2018) [161]	*	EAOMDV (2018) [178]
Network topology	Mesh, hierarchical based on DAG	Mesh, hierarchical based on DAG	Geographical greedy networks	Dynamic IoT network
Scalability	Yes	Yes	Yes	Yes
Applications	Emergency response applications	*	Smart street lights application and urban IoT applications	Mobile IoT applications
Routing metrics	Residual energy, transmission delay	ETX, the number of candidate parents	Distance from node to root (rank)	Lifetime hop count
Multi-hop routing	Yes	Yes	Yes	Yes
Consider link quality	No	No	No	No
Traffic flows	MP2P, P2P, P2MP	MP2P, P2P, P2MP	P2P	P2P, P2MP
Algorithm	Dijkstra algorithm	Distance vector	Distance vector, greedy-forwarding	Distance vector
Data rates	High	*	Low data rate	
Mobility of Network	No	No	Yes, but restricting the mobility of node to be one hop from the static node	Yes
Proactive or Reactive	Proactive	Proactive	Reactive	Reactive
Security	No	No	No	No
Buffering	Yes	Limited buffer size	Yes	Yes
Latency	Low latency	Low latency	*	Low latency
Simulation tool	NS2	Cooja	Simulation supports the implementation of open street map data set	NS2
OS to implement protocol	Linux	Contiki OS	*	Linux
Merits	• Achieves load balancing among routes • Minimizes delay, packet loss, and energy consumption	• Achieves load balancing among routes • Reduces end to end delay • Minimizes collision rates • Increases network lifetime	• Reduces control overhead in P2P communication • Improves scalability routing performance • Reduces memory utilization • Adaptive protocol that supports varying link densities • Avoids using DAO control messages	• Decreases end to end delay • Reduces packet loss rate
Demerits	• On large scale networks, energy consumption is not validated • High transmission rate will increase network congestion, which will lead to the increase of data loss rate • Uses a high number of control overheads • Requires a frequent update of routing tables	• It does not consider parents node energy • Large number of control packets	• It suits static network or requires embedding GPS into mobile nodes that should be one hop away from static nodes	• In data routing, there is no security technique applied • Requires more memory size to maintain ICT • It does not consider the residual energy of the node in selecting data route • It chooses the path with minimum hop count, but it may not be an optimal path • High latency and failure data delivery when link failure, as it stores information of one route only
References	[148] [155] [179]	[148] [156] [179]	[148] [157] [179]	[158] [180]

5.4 Data Link Layer IoT Protocols

This section handles the most popular IoT protocols in the data link layer and gives a brief description of their main specifications and future improvements as displayed in Figure 11, whereas Table 6 compares between them from different features.

5.4.1 Original Data Link Layer IoT Protocols

- **NFC protocol:** The range of this protocol is very short, so mobile objects that utilize it can communicate with each other over a few centimeters. All varieties of data can be transmitted in seconds between NFC devices if they are very close to each other. This protocol depends on RFID, as it utilizes the alteration in the magnetic field to allow devices to communicate with each other. NFC devices can operate in two modes, active and passive. In the active mode, all the communicating devices should create magnetic fields, wherein the passive mode one of these devices creates a magnetic field and the others utilize load modulation to transmit their data. The passive mode is very useful when power-constrained devices communicate with each other as it conserves the energy, which makes it widely used in all smartphones today [181] [182] [183].
- **Low-power Wireless Personal Area Network (6LowPAN) protocol:** 6LowPAN permits smart devices to connect to the internet using IPV6 protocol, takes into consideration the nature of wireless IoT networks through constructing very compact header message format [184]. Moreover, it breaks down hindrances to utilize IPV6 addressing protocol in limited processing capabilities, low data-rate, and restricted power IoT objects over the limited bandwidth of wireless networks [28] [185] [186].
- **Bluetooth Low Energy (BLE) protocol:** This communication technology was developed by Bluetooth Special Interest Group, as a low-power solution for short-range communication between controlling and monitoring applications [187]. Moreover, it supports quick transmission process of data packets with data rates up to 2Mbps in the ISM band. Devices that implement BLE protocol are classified into two types; master and slave where master devices act as a prime device that can connect to several slaves. To comprehend that, let us assume an IoT scenario in which a PC or a phone act as a master, where other devices as smartwatch, fitness tracker and thermostat are considered to be slaves. In such a scenario, slaves ought to be in a sleep mode until they receive packets from the master device to preserve their energy [28].
- **ZigBee:** It was designed in order to provide a scalable, low cost and low power wireless connectivity for a wide variety of controlling and monitoring applications. This protocol builds over IEEE 802.15.4 and extends its features through providing expandable and flexible wireless network topologies by employing intelligent routing and setup procedures to enable high resilience for failure and easy installation. Moreover, it is very efficient when working with other wireless communication technologies, as it incorporates rigorous security and listening techniques [188]. Based on the above, ZigBee will be utilized in a vast range of applications and products across commercial, government, consumer and industrial markets in the near future [189].
- **Radio Frequency Identification protocol**: RFID is a low cost and low power wireless communication protocol that is implemented on totally passive chips or battery-assisted passive (BAP) chips, which are embedded with antennas named tags [28]. These tags can send data only when they are powered through an electromagnetic field created by a reader [190]. The lifetime of RFID tags can be measured in decades, as they do not depend on an internal source of energy to operate, which makes this technology suitable in many IoT applications [191]. Nonetheless, the primary hurdle of this technology is that RFID tags operate only under a reader coverage domain, which is not more than 10 m in fully passive tags, while its range reaches up to 50 m in BAP tags [192].
- **Low Power Wide-Area-Networks (LPWAN) protocols**: LPWAN protocols are low-power, low-bandwidth, and low-cost protocols, especially in the communications over long distances areas. Moreover, the devices that implement these protocols transmit over sub-GHz radio frequencies from 433MHz to 868 MHz in Europe and up to 915 MHz in the USA, with transmission ranges from 1m up to 50Km [193]. Since many industrial, civil and other IoT applications operate over 2.4GHz or 5GHz ISM frequency bands, a number of low power wide-domain networking protocols have arisen. The following are the general characteristics of LPWAN protocols, followed by a brief discussion about the characteristics of each protocol:
  - ✓ The devices that implement these protocols have very low power consumption.
  - ✓ These protocols support the transmission process of small packets only, commonly 100 bytes or less.
  - ✓ The devices that implement LPWAN protocols consist of very low-cost units, so they usually cost less than a few dollars.
  - ✓ These devices are designed to have good coverage inside and outside their domains.

  i. **Long Range Wide-Area-Networks (LoRaWAN) protocol**: It is a physical layer communication protocol, with low power consumption and long battery lifetime that reaches up to 10 years. LoRaWAN is employed in wide Area Network (WAN) services and applications, such as M2M, industrial applications and smart cities [193], that require long communication distances ranging from (2-5) Km in urban territories and up to 15 km in suburban areas [194]. It also supports the communication process over large networks that contain billions of smart devices, thus the data rate of this protocol varies from 0.3 kbps to 50 kbps in the full-duplex wireless medium.

  ii. **Low Power WiFi (WiFi HaLow) protocol**: It is a wireless communication MAC and physical layers protocol. WiFi HaLow was developed to enable wireless sensors to communicate with each other over long distances with low power consumption.

  iii. **WiSUN protocol:** This protocol operates in both sub-GHz bands and 2.4GHz bands and it also supports data transmitting rates within (40 -1000) kbps for data packet size starts from1500 bytes and above. Furthermore, WiSUN enables IP packets to be delivered without fragmentation [195].

iv. **Narrowband Internet of Things (NB-IoT):** It is a narrowband radio technology that was standardized and developed by the 3rd Generation Partnership Project (3GPP) in June 2016 to support low data rates and complexity IoT applications. It introduces a novel radio access technology based on Long-Term Evolution (LTE) standards but with minimal features in order to reduce the power consumption of resource-constrained IoT devices. It operates on (180-200) kHz and also employs QPSK modulation.

v. **SigFox protocol:** A narrowband or ultra-narrowband technology was developed to connect a massive number of power-constrained devices. This protocol operates on an 868MHz frequency band, where the spectrum is divided into 400 channels of 100Hz. IoT devices can transmit up to140 packets each a day with a data rate of up to 100 bps and its signal can reach distances from (30-50) km in rural territories wherein urban territories it reaches from (3-10) km [196].

- **Z-Wave:** A low power wireless communication technology is designed for domestic automation products like smart light controller and other sensors inside home devices. This technology aims to provide reliable communication of small data packets with low latency transmissions and small data rates that reach up to 200kbps and operate over 900MHz ISM bands. Moreover, the Z-Wave protocol enables controlling of up to 232 smart devices [197].
- **Cellular**: Any IoT service that demands to operate over long distances can benefit from deploying Global System for Mobile Communication (GSM) technologies such as 3G, 4G, and 5G cellular communication protocols, as they have abilities to transmit large quantities of data packets, particularly in 4G and 5G technologies. Based on that, communication through cellular protocols is very expensive and extremely power-consuming for many applications [198].
- **Telensa:** This communication protocol transmits over Ultra Narrowband technology and sub 1GHz unlicensed ISM bands. Besides, it completely supports bi-directional communications (full-duplex technology). Consequently, it is convenient for monitoring and controlling the operations of IoT applications. A Telensa sink node could connect up to 5000 devices and its communication range can reach up to 2km in urban territories and 4 km in rural environments. The lifetime of A Telensa node can reach up to 20 years [199], which makes is applicable for many applications such as smart lighting, smart parking, and other smart city applications that are required long lifetime sensors [200] [201].

5.4.2 Recent Enhancements of Data Link Layer IoT Protocols

- Considering the exponential expansion in the number of heterogeneous air interface technologies that have their different characteristics and require to communicate with each other, NFC is deemed to be one of the most used air interfaces technologies for short distances. NFC has many properties such as protecting the privacy and the security of communication from attacks, low power consumption, and acceptable overhead. IPv6 considers to be an ideal internet protocol solution, as it provides large address space for a huge amount of network devices. Thus, it is necessary to enhance the characteristics of NFC protocol to support transmission over IPV6 protocol utilizing 6LowPAN techniques to produce a novel version of the NFC protocol called IPv6-over-NFC. However, this new technology is not suitable to transmit large data size or multimedia streaming over long-lived distances [202].
- Given the essential role of BLE technology in IoT fields, many communities such as IETF and Bluetooth adapt 6LoWPAN technology to enable IPV6 over BLE mesh networks (6Lo-BLEMesh). Nevertheless, 6Lo-BLEMesh technology finds the desired route by using a routing protocol, which makes the network prone to many threats and attacks [203].

1. Leonardi et al proposed a connection Multi-hop Real-Time BLE (MRT-BLE) protocol to provide higher throughput and bounded packet delays compared with the connectionless origin version of BLE [204]. Moreover, it permits data to hop over 37 channels instead of 3 connectionless channels. The basic idea of MRT-BLE is to divide the WSN into many sub-networks, where each one of them is managed by a master node and two sub-networks are connected through a master\slave device or a slave device that acts as a bridge between them. However, MRT-BLE does not suit mobile networks.

- ZigBee 3.0 is built over Zigbee PRO to support monitoring and controlling of heterogeneous networks that connect IP based devices from different vendors and markets such as smartphones, tablets or computers by adding security layer and mesh networking to the application framework. This will make heterogeneous IoT networks certifiable, green low-power, more reliable and robust, interoperable and full-stack solutions [205].
- LoRaWAN technology is developed to support fixed battery-powered or mobile star networks, where the gateway node has the responsibility of relaying data between the central server and end devices. Gateways are connected to the central servers via IP connection standards, while the end devices communicate with one or more gateways through FSK communication or single-hop LoRa. All the communications between the gateways and devices are bidirectional and spread over different data rates and channels. An adaptive data rate technique is utilized by LoRaWAN to maximize the network lifetime. LoRaWAN technology is classified into three categories, which are (class A) bi-directional end-devices, (class B) bi-directional end-devices with scheduled receive slots, and (class C) bi-directional end-devices with maximal receive slots. LoRaWAN v1.0.3 supports both unicast and multicast transmissions over class B end devices, whereas (DeviceTimeRequest) a new MAC command is utilized to synchronize the time clock of both class A and class B end devices [206].
- IEEE 802.11ah-2016 technology was proposed by the IEEE standards association to extend the transmission range of Wireless Local Area Network (WLAN) to sub 1 GHz band, providing alternative bands rather than heavily overcrowded 2.4GHz and 5GHz bands. 1 GHz band used nowadays to minimize the propagation loss through obstructions, walls, and free spaces. Moreover, IEEE 802.11ah-2016 provides multiple data rate modes, based on the application's requirements starting from 150kbps up to 347Mbps. Low data rates options are appropriate for IoT applications, as it can provide full home coverage for the transmission of battery-powered devices, whereas the high data rates modes are suitable for power amplifier devices. Briefly, IEEE 802.11ah-2016 aims to improve WLAN lifetime, provide more network scalability, and support single-hop and multi-hop operations [207].

- Z-Wave Plus is a novel version of Z-Wave protocol, where it was designed to enhance smart home users' experience and make installation and setup of this protocol easier and faster. It extends Z-Wave capabilities by increasing battery life 50%, allowing devices to communicate with each other up to 60m, and permitting automatic installation of new devices. Moreover, Z-Wave plus improves network bandwidth to be more than 250%, as it offers 3 new radio frequency channels, which will improve IoT devices connections, noise immunity, and increase bandwidth. Lastly, it enables devices to address issues with the explorer frame feature in order to enhance fault tolerance and self-healing [208].

5.4.3 Future Research Directions of Data Link Layer IoT Protocols

- 5G is the latest cellular communication protocol that replaces the current 4G technology by providing a number of enhancements in scalability, reliability, connectivity, speed, energy, and efficiency of the network. The main reason for developing a new communication technology is to support the tremendous growth of devices connected to the internet and controlled the operations of critical commercial machines and appliances, which creates the need for zero delays, more bandwidth, and less energy consumption communication. 5G supports the connection of 1 million devices per square km their speeds may reach up to 500km/h, allowing them to communicate with uplink speeds at least 10Gbps and 20Gbps for downlink, while it reaches 100Mbps for download and 50Mbps for upload per user. This cutting-edge technology is very efficient in energy conservation as it alternates between sleep mode with zero energy consumption and applies energy efficiency mode in loaded circumstances. Moreover, the maximum latency caused by utilizing 5G technology should reach 4ms compared with 20ms by the 4G network. 5G supports multi-layer spectrums to meet different requirements, through utilizing large-scale antenna, which are sub 1GHz for low-band spectrum, 1GHz and 6GHz for mid-band spectrum and 24-40 GHz for high-band spectrum [209] [210].

- 3GPP-Release17 technology concentrates on enhancing 5G system capabilities to be launched in 2021. This release will enhance and cover many aspects, such as 5G IoT, high precision positioning, improving low latency and ultra-reliable communications, asset tracking, application layer support for 5G factories, unmanned aerial communication systems, audio/visual service production, communication services for critical medical applications, and architectural enhancements for 5G multicast-broadcast services [211].

- Telensa 5th generation has released "urban data project" with the partnership of Qualcomm, Kainos, and Microsoft Azure to protect the data generated from street light sensors by applying city-data guardian method in the cloud with safeguard in data usage and privacy, which will improve and leverage city services [212].

Table 6: Comparison between data link layer protocols considering different aspects

Wireless communication Protocol	**NFC**	**6LowPAN**	**Bluetooth Low Energy (BLE)**	**Zigbee**	**RFID**	**LoRaWAN**	**Low Power Wi-Fi WiFi HaLow**
Network standard	ISO/IEC 13157, ISO/IEC 18000-3	IEEE 802.15.4	802.15.1	IEEE 802.15.4	ISO 18000 v1 – ISO 18000 v7 ISO 10536, ISO 11784, ISO 11785, etc.	*	IEEE 802.11ah
Recent version of the protocol (year)	IPv6-over-NFC (2019) [202]	6Lo-BLEMesh (2019) [203]	6Lo-BLEMesh (2019) [203] MRT-BLE (2018) [204]	Zigbee 3.0 (2018) [213]	RFC 8371 (2018) [214]	LoRaWAN v1.0.3 (2018) [206]	IEEE 802.11ah-2016 (2017) [207]
Network type	P2P	Star, mesh	Star	Star, tree cluster, mesh, hybrid	P2P network, mesh	Star-of-stars, mesh	Mesh, star, tree
Frequency Band	13.56MHz	2.4GHz	(2.402 – 2.481) GHz	2.4GHz, 915Mhz, 868Mhz	(125–134) KHz (13.56, 865-60) MHz (902-928) MHz	(100Hz, 869 MHz) for Europe 915 MHz for North America	(1, 2, 4, 8, 16) MHz (902 -928) MHz USA (863- 868) MHz Europe (775-787) MHz China. 1 GHz
Transmission range	10 cm	(10-100) m	up to 100 m	(10-100) m Sub-GHz up to 1km	(1-10) cm (1 -30) m	(2-5) km urban environment, 15km suburban environment	1 km
Power consumption	15 mA	*	15 mA	30 mA	*	up to ~50mW	2 µA- 8 mA
Number of nodes per network	2 nodes	65000 nodes	65535 nodes	65000	*	Thousands of nodes	8191

Applications	Service initiation applications, payment, and ticketing applications, P2P data transferring	Smart home, smart agriculture, industrial IoT, structural monitoring, healthcare applications	Mobile phones, gaming, smart homes, wearables, PCs, security, proximity, healthcare, sports and fitness, Industrial, etc.	Smart home, medical monitoring, environment AI sensors, consumer electronics	Retail sector, warehouse management, inventory management, supply chain management and logistics, library systems, traceability management medicine smart spaces, smart parking, environmental monitoring	Smart city, industrial applications, real-time monitoring, metering, smart logistics and transportation, video surveillance.	Smart home, digital healthcare, smart city, agriculture, retail
Data rate	106 kbit/s -424 kbit/s	(20, 40, 250) kbps	125 Kbps, (1, 2) Mbps	250kbps	700 kbps - 4 Mbps	250 bps– (5.5, 11, 50) kbps	347 Mbps
Spreading technique	*	DSSS	FHSS	DSSS	DSSS, FHSS	FHSS, CSS	DSSS, FHSS
Applicable routing protocols	NFC includes routing features	RPL, AODV	RPL, 6LoWPAN	Zigbee, RPL, AODV, ZBR[22], ZBR-M	OLCMR[23] OLSR[24]	AODV, HWMP[25]	AODV, OLSR, DSDV[26]
Mobility	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cryptography	No	AES[27]-128 bit	AES-128 bit	AES-128 bit, ACLs[28]	Present, Hummingbird, Photon, DES, Hight	AES-128 bit	WPA3,[29] Morse micro, OTA[30]
References	[215] [216] [217]	[24] [28] [186] [218] [219]	[24] [220] [221]	[222] [223] [224]	[225] [226] [227] [228] [229]	[230] [231]	[207] [232] [233] [234]

Table 6: Comparison between data link layer protocols considering different aspects (Cont.)

Wireless communication Protocol	**Wi-SUN**	**NB-IoT**	**SigFox**	**Z-Wave**	**Cellular 1G, 2G, 3G, 4G**	**Telensa**
Network standard	IEEE 802.15.4g	3GPP	ETSI EN [31]300 220-1, ETSI EN 300 220-2	IEEE 802.11 IEEE 802.15 IEEE 802.16	MTS [32], AMTS [33], PTT[34] (1G) GSM, iDEN[35], GPRS, HSCSD[36] (2G) UTMS[37], IMT[38]-2000 (3G) LTE, LTE -A [39], IMT-Advanced (4G)	*
Recent version of the protocol (year)	*	3GPP-Release 17 (2019) [211]	Sigfox v. 2.6.0 (2018) [235]	Z-Wave plus (5th Generation Z-Wave) (2015) [208]	5G (2018)	Telensa 5G (2019-2028) [212]
Network type	Mesh, star, hybrid star/mesh	Star	Star	Mesh	Mobile network or cellular network	Mesh, star
Frequency Band	920 MH 863–870 MHz	3.75 kHz, 15 kHz, 180-200 kHz, 850-900 MHz	200 kHz 868 - 869 MHz 902 -928 MHz	868 MHz (Europe) 908 MHz (United States) 900MHz (ISM)	30 KHz (1G) 200 kHz (2G) (1800- 2400 MHz)3G (2-8 GHz) 4G	60MHz, 200MHz, 433Mhz, 470MHz, 868Mhz, 915MHz

[22] ZBR: ZigBee Network Routing
[23] OLCMR: Optimal Link Cost Multipath Routing
[24] OLSR: Optimum Link State Routing
[25] HWMP : Hybrid Wireless Mesh Protocol
[26] DSDV: Destination Sequenced Distance Vector
[27] AES: Advanced Encryption Standard
[28] ACLs: Access Control Lists
[29] WPA3: Wi-Fi Protected Access 3
[30] OTA :Over-the-Air
[31] ETSI EN: European Telecommunications Standards Institute, European Standard
[32] MTS: Mobile Telephone System
[33] AMTS: Advanced Mobile Telephone System
[34] PTT: Push to Talk
[35] iDEN : integrated Digital Enhanced Network
[36] HSCSD: High-Speed Circuit-Switched Data
[37] UTMS: Universal Mobile Telecommunications System
[38] IMT: International Mobile Telecommunications
[39] LTE-A: Long Term Evolution Advanced

Transmission range	500m -1 km	1 km (urban) 10 km (rural)	(30–50) km (rural) (3–10) km (urban)	30 m	(2- 20) km 1G (35-200) km 2G Rural: 500 km/h *t, suburban: 120 km/h *t, 10 km/h *t (3G) 500 km/h *t (4G)	20km (rural) 3km (urban)
Power\ current consumption	2 µA- 8 mA	(3-50) µA	500 mW - 4W/ (19-49) mA	~5mW	1800mA (2G) 800mA (3G) (1,000–3,500) mW 4G	100µW
Number of nodes per network	5000	55000, 100 K devices per cell	*	232 nodes	4,000 devices /km^2 (4G)	5000 lights per base station
Applications	Smart meters, smart city, smart agriculture	Electric metering manufacturing automation, retail point of sale terminals, smart city	Smart farming, status monitoring, asset tracking, smart building, pallet tracking for logistics	Smart home	Voice Calls (1G) Voice calls, browsing and short messages (2G) Video conferencing, GPS and mobile TV (3G) Wearable devices, high-speed applications and mobile TV (4G)	Street lighting, smart city, air quality, traffic monitoring, smart waste bin management, and smart meter
Data rate	50 kbps- 1 Mbps	(30-60) kbps 200 kbps	(10-100) bps	(9.6, 40, 200) kbps	2.4 kbps (1G) 64 Kbps (2G) 144 kbps-2 Mbps (3G) 100 Mbps - 1 Gbps (4G)	500bps downlink 62.5 bps uplink
Spreading technique	DSSS	DSSS	FHSS	DSSS	FHSS, DSSS, CDMA[40]	*
Applicable routing protocols	RPL	*	*	AODV, DSR[41]	AODV, DSR, GPSR[42]	RPL
Mobility	Yes	Yes	Yes	Yes	Yes	Yes
Cryptography	AES, certificates, HMAC[43]	AES, LTE encryption	AES-128	AES-128	Voice scrambling (1G) Authentication and 128-bit key per subscriber (2G) SNOW3G cipher, Rijndael cipher, KASUMI cipher and AES-128 (3G) EPS integrity algorithm (4G)	City-data guardian
References	[236] [237]	[238] [239]	[238] [240]	[24] [241]	[242] [243] [244] [245]	[212] [246]

6. Middleware

It is anticipated that the number of IoT devices will reach around 50 billion in 2020 [247]. This massive number of smart things that are connected to the internet, represents the so-called IoTs, aims to make the surrounding environment more intelligent [248]. Based on the above, the amount of the collected data in the IoT environment will be immense and will create considerable defiance for both industries and researches domains. One of the major challenges that IoT paradigm confronts is machine-to-machine communication, where this challenge forms a big concern in IoT systems because of an abundant number of the existing smart devices that do not follow the same protocols, as most vendors do not care about the compatibility of their products with other competitors' brands. One of the proposed solutions to solve this issue is to enforce universal standards, which is very hard to be applied, while another proposed solution is to implement middleware software to facilitate the communication process among these devices. Middleware can be defined as a software that offers interoperability between incompatible applications and devices, also it hides all the details of smart objects [249] [250]. Hence, it acts as a software bridge between the applications and the things, as it enables IoT systems to work efficiently with each other [12] [20] [24] [251]. There are numerous middleware solutions, either a proprietary or an open-source provided through companies, where most of these solutions are similar to each other. However, there are no guidelines or performance metrics that enable us to compare these solutions to each other [249]. According to that, many challenges face IoTs middleware as described below [28]:

i. **Programming abstractions and interoperability**: To facilitate collaboration and data exchange among heterogeneous devices, IoT middleware aids to permit distinct sorts of smart devices to interact easily with each other.

ii. **Device management and discovery:** This property allows IoT devices to discover all other devices and services that are located in their network domain. The infrastructure of the IoT environment is mostly dynamic since all newly joined devices must announce their existence and the services they provide. Therefore, IoT middleware requires being scalable and provides APIs in order to list all IoT devices, their capabilities, and their services. In addition, APIs have to provide the users with abilities to categorize the devices based on their capabilities, manage devices depending on their remaining energy, report problems in IoT devices to the users and perform load-balancing procedures among them.

iii. **Big data and analytics**: IoT sensors collect an enormous amount of data that requires to be analyzed by specific algorithms based on a data type. Also, some of the sensed data may be incomplete because of the flimsy nature of wireless sensor networks. Thus, middleware should consider this issue and extrapolate incomplete data by using a suitable machine-learning algorithm.

[40] CDMA: Code Division Multiple Access
[41] DSR: Dynamic Source Routing
[42] GPSR: General Packet Radio Service
[43] HMAC: Hash based Message Authentication Code

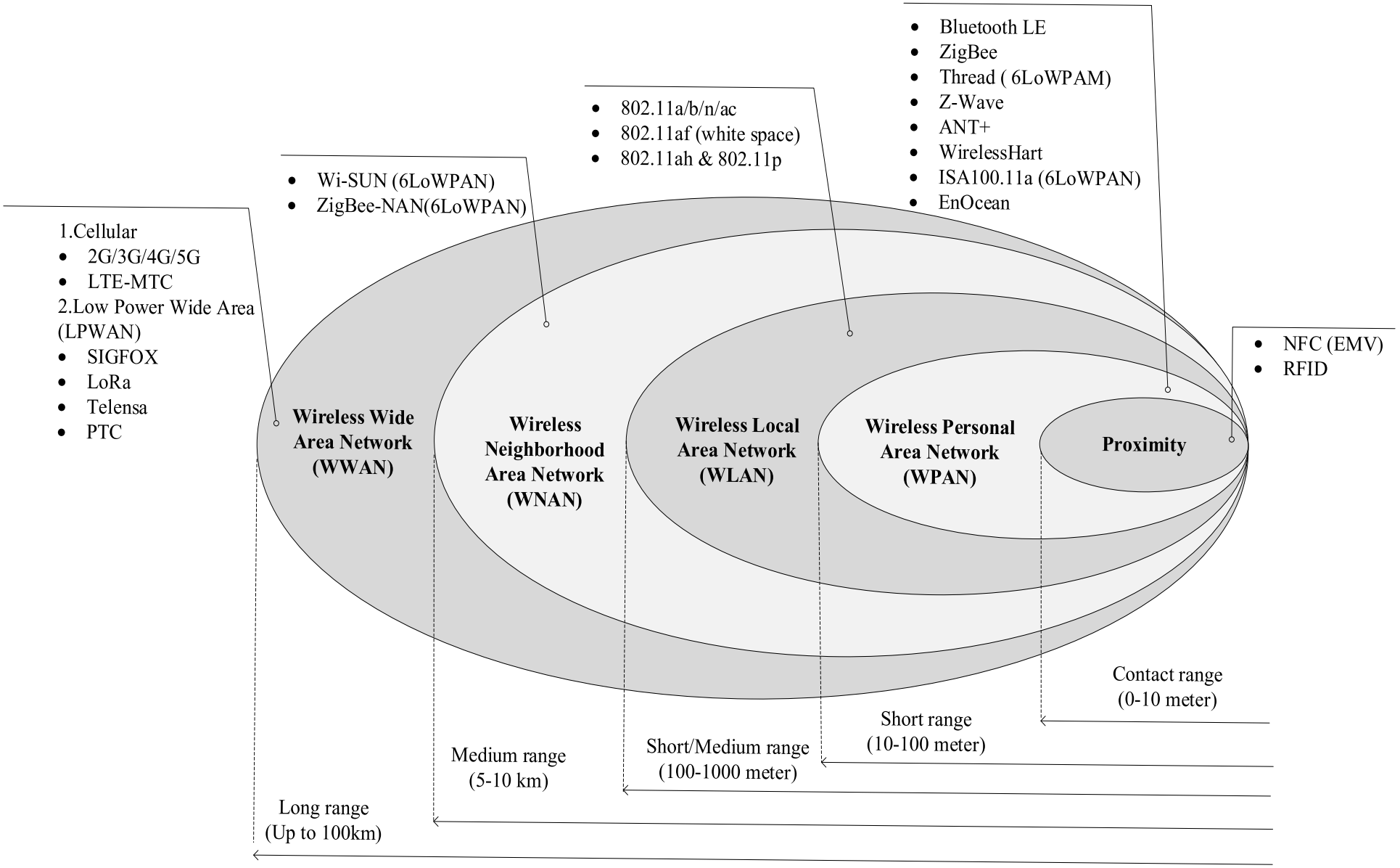


Figure 11: Wireless IoT connectivity technologies

iv. **Privacy:** Most data that comes from IoT applications and services are related to human personal life. Thus, security and privacy issues have to be considered when transferring and processing them, which is required to build mechanisms that address these issues by middleware.

v. **Cloud services:** Cloud computing part is the most important layer of any IoT system because all of the sensed data will be stored and analyzed in a centralized cloud. Therefore, IoT middleware should be run smoothly in distinctive types of clouds and enables IoT users to gain the most benefits from the data collected through smart sensors.

vi. **Context detection:** IoT applications are classified into two types, which are ambient data collection applications and real-time reactive applications. In the first type, sensors collect data that will be processed later on offline to get reasonable information that will be used for the same scenarios in the future, while in the second type systems should make a real-time decision based on the sensed data.

6.1 Architecture of IoT Middleware

The current architecture of IoT middleware is classified into three types based on the services they provide as follows [252]:

1. **Service-Oriented Architecture (SOA) or Service-Based Solution:** In SOA users and developers are allowed to employ or add different types of IoT devices to be utilized as services [30] [253]. Figure 12 represents the architecture of SOA middleware, which consists of three layers: The Physical layer that contains actuators and sensors, the Virtualized layer, which consists of cloud and infrastructure servers that are responsible for performing different computational operations, and the Application layer that composes of all services and utilities. SOA is deemed to be a heavyweight and a very high performing middleware, where it can be implemented on the nodes that communicate with the cloud servers or on a powerful gateway that is placed between the cloud layer and IoT devices layer. Based on that, this type of middleware is not suitable to be implemented on resource-constrained devices and it does not permit device-to-device communication.

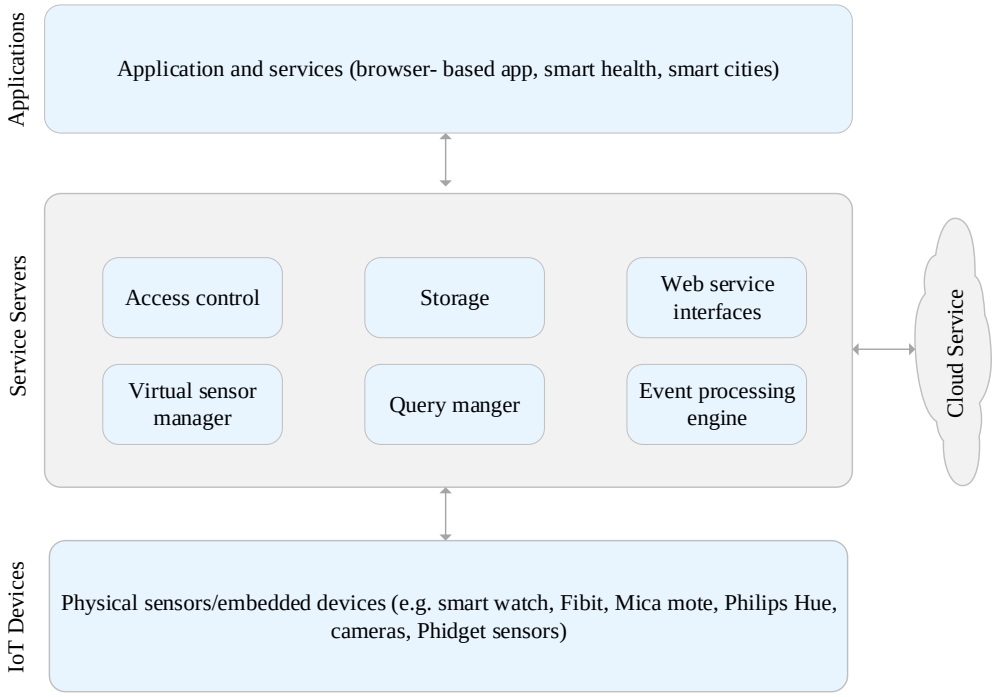


Figure 12: Service-based IoT Middleware

2. **Cloud-Based Solution:** In this type, users are constrained by the number and types of smart devices that can be connected to IoT applications. In addition, the sensed data can be easily collected and interpreted, because different used cases can be programmed and then determined in advance [30]. The resources of the cloud-computing environment restrict the operational components of this middleware. These functions such as storage system or computation engine are represented and managed by APIs, where IoT services are controlled and accessed by either cloud bolster RESTful APIs or by the applications provided by vendors as shown in Figure 13.
3. **Actor-Based Framework**: It is a lightweight middleware that can be implemented in Sensory, Gateway and Cloud Computing layers. The computational operations of this middleware are distributed in both sensory layer and mobile access layer as shown in Figure 14 [24].
4. **Event-Based Framework:** This type of middleware aims to improve the development of distributed systems by supporting the implementation of the publish\subscribe paradigm as shown in Figure 15. This paradigm is considered to be a communication infrastructure that aims to provides clients with general-purpose services, as it helps them to cope with the heterogeneity and complexity of large-scale and distributed environments. In event-based middleware, distributed application complexity is partially hidden from the programmer, which will, in turn, simplify the development and programming of many functionalities.

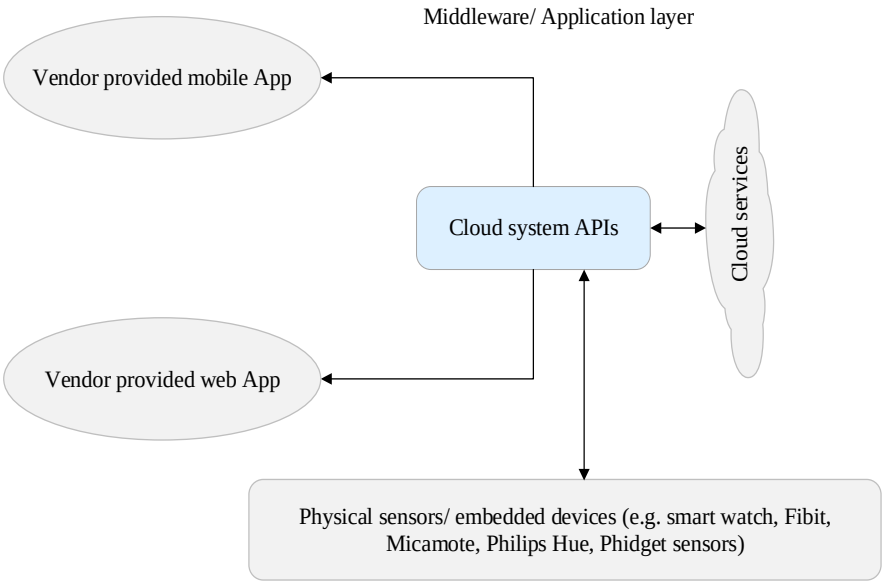


Figure 13: Cloud-Based IoT middleware

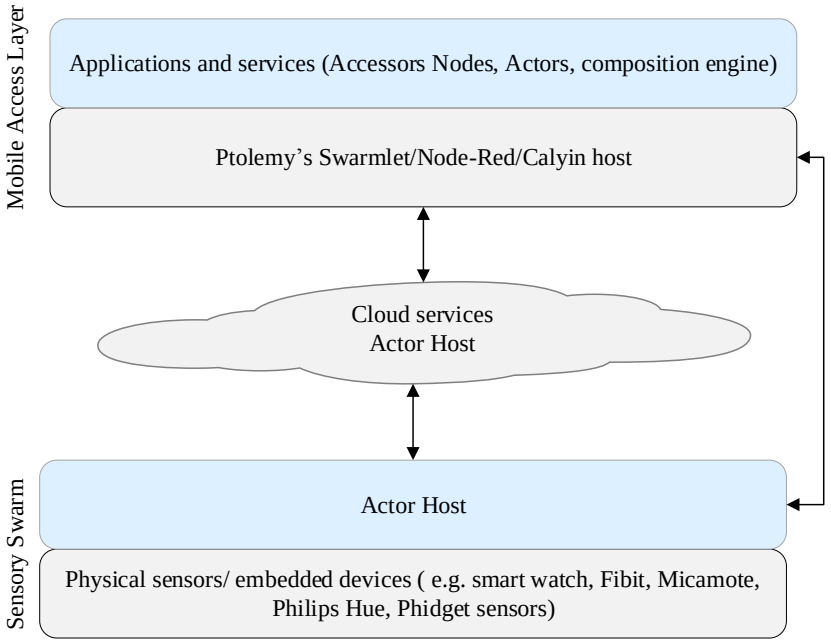


Figure 14: Actor-based IoT Middleware

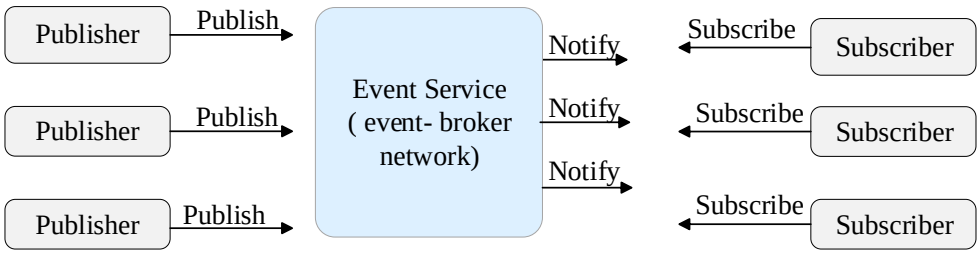


Figure 15: Event-Based IoT Middleware

6.2 Existing IoT Middleware Platforms

The following subsections summarize different solutions of IoT middleware based on its type, where Table 7 compares between IoT middleware platforms from different aspects.

6.2.1 Cloud-Based IoT Middleware

1. **AWS IoT:** This platform was developed by Amazon to manage cloud services, such as permitting millions of connected devices to interact securely and easily with other devices and cloud applications. AWS IoT allows customers to build their IoT applications in order to collect, process and analyze the sensed data to take a suitable decision without any need to manage

infrastructure by using AWS services like Amazon Kinesis and Amazon CloudWatch. Also, AWS IoT customers can keep track of all the devices that are communicating with their applications all the time [254].

2. **Azure IoT Hub:** It is a central platform that was released by Microsoft to manage bidirectional communication between IoT applications and their connected devices. Due to the high capabilities of Azure, it allows clients to construct full-featured, scalable IoT solutions with secure and reliable communications among the hosted cloud and a massive number of IoT devices. Azure IoT Hub supports various messaging patterns to control IoT connected devices, such as request-reply, file upload from devices and device-to-cloud telemetry [255].
3. **IBM Watson IoT:** This platform is built on the top of IBM Cloud to connect and control different IoT appliances, sensors industries, and home applications. IBM Watson provides its clients with an enormous set of adds-on, built-in tools, and Blockchain service that enable them to build their own IoT applications, manage their appliances, extract key performance indicators from their data, connect their tools and applications and process their collected data using historical and real-time analytics.
4. **Google Cloud IoT**: It is a fully managed service, which consists of a set of tools that provide a complete solution to securely and easily connect, process, manage, store, analyze, and visualize the generated data from dispersed devices, both in the cloud and at the edge of the network. Google Cloud IoT aims to have the ability to build models that can efficiently optimize and describe a client's business, anticipate problems, and improve operational efficiency [256].
5. **Xively**: It is a public cloud-based IoT middleware that provides a Platform as a Service (PaaS) [257]. This software aims to help companies and developers to connect, monitor and control distinctive types of IoT sensors [258]. Furthermore, it offers a web-based application that allows IoT devices to quickly connect and transmit data to its cloud servers. Also, it allows clients to retrieve their data from the cloud easily at any time and from anywhere, as it provides a time-series database that enables swift storage and retrieval of data [30].
6. **Oracle IoT:** It is a cloud-based service platform that enables users to build a real-time IoT solution, which can be integrated with enterprise applications, using robust security cloud capabilities, innovative and powerful edge analytics. Moreover, it processes the streaming of IoT data to merge insights into customer business easily and quickly. Oracle IoT permits clients to connect their devices to the cloud, which will help them in taking critical strategies and decisions [259].

6.2.2 Service-Based IoT Middleware

1. **LinkSmart (Hydra):** It is a web service platform that aims to eliminate the heterogeneity of distinctive devices and entities in the IoT environment [260] [261]. Furthermore, it enables controlling all types of smart devices regardless of their communication protocols, such as ZigBee, RF, RFID, Wi-Fi, Bluetooth, etc. LinkSmart distributes social trust computation and security units through middleware to make IoT devices and services more secure and trustworthy. A prime novelty of this middleware is supporting the utilization of IoT devices as services by embedding the required services in these devices. LinkSmart can also be used to manage specific IoT applications such as healthcare, agriculture and home automation. Also, it supports the self-configuration of devices and service discovery [262]. There are no local aggregation or processing units for the sensed data on IoT devices that implement LinkSmart, so it will be sent to the cloud to be processed and archived [263].
2. **Kaa:** It is an open-source platform that is managed by Cybervision Inc and KaaIoT technologies to enable building IoT solutions. Using web page Graphical User Interface (GUI), based on the Apache platform, enables the creation of data delivery schema, supporting multi-tenancy on servers and generation of endpoint Software Development Kit (SDK). Kaa enables interaction with endpoint devices directly or via gateways, while it secures their data by AES and RSA encryption methods.
3. **Global Sensor Networks (GSN**): It aims to provide a uniform platform that supports adaptable deployment, sharing, and integration of heterogeneous IoT objects [20] [264]. This platform is built to meet the requirements of smart objects whether they are physical or virtual sensors or actuators. GSN is a Java platform that is deployed either on IoT cloud or servers, where a set of wrappers are permitted to feed the system with a collected live data, which will be processed later on based on XML specification files.
4. **ThingSpeak IoT:** It is an analytical open-source platform service that is developed by Matlab to enable communications between people and things. ThingSpeak provides users with tools that permit them to collect, visualize and analyze real-data streams in the cloud. Developers can easily store and retrieve data from devices and sensors by utilizing HTTP protocol over the internet [265].
5. **Aura:** This middleware is designed to ease the development of pervasive mobile IoT applications, by abstracting the differences among heterogeneous devices and permitting them to communicate with each other without any hindrances. Aura tries to optimize screen backlight and CPU to improve the performance level and reduce power consumption. Aura applies two concepts in interacting with events, where system layers reply directly to the upper layer in a proactive concept, while in a reactive concept all layers adjust their resources and performance based on demand [266].

6.2.3 Actor-Based IoT Middleware

1. **Calvin:** It is an open-source IoT platform that was developed by Ericsson to be implemented on the energy-constrained smart devices since it provides a portable and light-weight unified programming model, where its interfaces are defined via its input and output ports [267]. In Calvin, all low-level communication protocols of IoT devices are hidden as the communication between devices is performed through smart things' ports [30]. Moreover, Calvin can be implemented at the edge of IoT environments to reduce long-distance communications, which will minimize the latency and power consumption of IoT devices. The major merit of this middleware is its ability to migrate from one environment to another.

2. **Node-RED**: It is an open-source IoT platform that was developed by IBM and is based on node.js[44] [268]. This platform can be implemented at the edge of IoT network, because of its light footprint, whereas on the server-side, a JavaScript platform based on an event-driven module and a non-blocking I/O is implemented. The node-red interface permits users to build their IoT applications easily through dragging and dropping the connected blocks that represent IoT components. The disadvantages of this platform are that it does not support service discovery and enables security by password authentication only [30].
3. **Ptolemy Accessor Host:** This open-source platform was developed by Professor Edward Lee in 1996 to design, simulate, and model embedded and real-time devices [269]. The main concept of this platform is that an IoT system is built based on the software components that interact and communicate with each other via messages sent through interconnected ports [30].
4. **Akka:** It is a set of open-source libraries and free actor-based platform that was designed to build distributed and run-time applications using Java or Scala language. It permits users to meet business requirements without the need for writing large low-level codes, which will provide them with high performance, fault tolerance, and reliable behavior. Akka also supports multi-threading behavior, abstracts the communication among applications and their devices and provides high-availability and clustered architecture [270] [271].

6.2.4 Event-based IoT Middleware:

1. **Hermes:** It is an event-based and scalable middleware that aims to ease the construction of distributed and large-scale applications. Hermes creates self-managed event brokers based on P2P routing layer, to handle large scale and dynamic environments. It introduces a resilient solution against failures via automatic adaptation of event brokers routing states and overlay broker network. Hermes middleware released two versions that share most of the codebase, which are the implementation in distributed and large-scale applications and communications, besides the implementation among event brokers [272].
2. **Gryphon**: It is a patronizable publish\subscribe and highly scalable middleware that aims to distribute a large amount of real-time data over the network. Gryphon is developed by Java interface to support web applications and to build a robust, redundant, publish\subscribe, and content-based multi-broker. This middleware contains robust security features, scalable routing algorithms, and an effective event matching engine. Also, it is based on an information flow paradigm for messaging (BKS+99) to specify the communication between the publisher and the subscriber.
3. **Rebeca:** This middleware is based on publish\subscribe technology to implement largescale business applications, by emphasizing on the design of efficient routing algorithms and employing professional software engineering methodologies. Rebeca aims to prevent and reduce flooding the network by events by utilizing advanced routing techniques. It integrates interoperability and subscription merging features with its services to support location mobility and to reduce routing table size. Event scope function hides the details of service implementation, as transmission policies, security, data transmission methods, interfaces among external and internal, and notification representation [272].
4. **FiWare***:* It enables efficient, flexible, secure and scalable communications among distributed IoT devices and applications. It was designed to support the control and monitoring of many IoT applications such as logistics, shopping floor and smart city [28]. This platform consists of many components such as APIs, reusable modules and huge codes, which allow an IoT user to build his IoT application. A set of sensed data collected by IoT sensors (context) is captured through REST API, to be sent later on to a specific server called the broker. FiWare has developed API to query and store different IoT contexts, so any application is registered as a context consumer can retrieve the required data from the broker. There is a specific component in this platform called an adapter, where it is responsible for transmitting a particular type of context to the subscriber applications [273].

6.3 Open Research challenges of IoT Middleware:

Even though the IoT middleware field has handled many requirements and issues that face the development of IoT applications, there are still some open challenges that require to be covered and solved. The following bullet points summarize some of these issues:

- **Non-autonomous or semi-autonomous devices and services registration and discovery:** Human intervention in IoT components through registration and discovery, makes these applications non-scalable and prone to error. This issue makes middleware unsuitable for self-adaptive applications, as M2M communication systems.
- **Unscalable device and services registrations and discovery:** The time consumed on devices and services registration or discovery may make middleware an improper solution even in small IoT systems.
- **Heterogeneous environments**: This issue is considered to be a key challenge that needs to be addressed since most of the middlewares support only one or two types of heterogeneous IoT components. Non-autonomous and inflexible services and devices registration and discovery limit the support of IoT applications. Subsequently, it is highly recommended that new approaches should handle and resolve the heterogeneity of IoT environments, especially in large-scale networks.
- **Leakage of device interpretability**: An abstraction layer is required between middleware and resources to solve this challenge. Also, extensive researches to abstract the heterogeneity of the resources of IoT systems should attract more attention from the researchers.
- **Service Level Agreement (SLA):** To afford an agreed level of service to customers, three components should be taken into consideration: A model that precisely defines all functional and non-functional services that are required by consumers, automatic service to guarantee a high level of QoS and adaptation, and monitoring tool for SLA services. Human intervention in current SLA middleware should be replaced and considered by middleware development.
- **QoS level:** There is no mechanism that guarantees a specific level of QoS for non-functional services of IoT. So, middleware researches should find procedures for optimizing and monitoring QoS levels.

[44] **Node.js** : It is an open-source, cross-platform and run-time environment for executing JavaScript code on the server-side.

Table 7: Comparison between IoT middleware considering different aspects

IoT middleware	AWS IoT	Azure IoT Hub	IBM Watson IoT	Google Cloud IoT	Xively	Oracle IoT	LinkSmart (Hydra)	Kaa	GSN	ThingSpeak
Middleware architecture	Cloud-Based	Cloud-Based	Cloud-Based	Cloud-Based	Cloud-Based	Cloud-Based	Service-Based	Service-Based	Service-Based	Service-Based
Open source SDK\ open API	Open source SDK	Open Source API	Open source SDK	Open API	Open API	Open source SDK	Open API	Open source SDK	*	Open source
Device abstraction\ Interoperability	Web services	Azure IoT SDK for C	Through MQTT	gRPC, REST APIs	Web services, MQTT, board support package	Oracle service bus	Web services	Apache NiFi, Apache ZooKeeper	XML-RPC, protothreads, token machine language	Libelium, AllJoyn, Beckhoff, Senet
Deployment type	IaaS, PaaS	IaaS	IaaS, PaaS	IaaS, PaaS	PaaS	PaaS	PaaS, SaaS	IaaS	PaaS	PaaS
Network connectivity	MQTT, HTTP, WebSocket	HTTP, AMQP, AMQP over WebSocket, MQTT, MQTT over WebSocket	MQTT, HTTP, TLS	MQTT, HTTP	HTTP, MQTT, WebSockets MQTT	MQTT, HTTPS	HTTP, REST, MQTT	MQTT, CoAP	HTTP	MQTT, REST API
Data format supported	JSON	JSON	CSV, JSON	JSON	CSV, JSON, REST API	CSV, REST API	JSON	REST, JSON, API	JSON, SenML	XML, CSV, ThingSpeak API, JSON
Programming languages supported	SDK for Arduino, Java, NodeJS, C, JavaScript, Python, iOS, Android	Node.js, Python, Java, Android, iOS, C, C#	NodeJS, Java, Python, C#, C	Java, Node.js, .NET, Python, Ruby, PHP	SDK for Arduino, Python, Clojure Android, Arm mbed, Ruby, C, JavaScript	Android, Java, JavaScript, iOS, C	PHP, Java, C#, Python, .NET, JavaScript,	C, C++, Java	Ruby, Java, C	Matlab
Application development functionalities	Real-time analytics, analytics, artificial intelligence, machine learning, event reporting, visualization	Real-time analytics, analytics, machine learning, event reporting, visualization	Real-time analytics, analytics, machine learning, event reporting, visualization	Real-time analytics, analytics, machine learning, event reporting, visualization	Real-time messaging, file and firmware deployments, device provisioning, device logs, rules and orchestrations	Real-time analytics, analytics, event reporting, visualization	Device abstraction, stream mining, live data management, data storage, online machine learning	Analytics, machine learning, event reporting, visualization	visualizing the network structure, data stream processing, plotting data	Real-time analytics, analytics, event reporting, visualization
Technologies used for application development	AWS Cloud-Trail, AWS Lambda, Kenisis, Amazon, Amazon Dynamo DB, Amazon CloudWatch Amazon machine learning	SQL database, Azure tables, Azur CosmosDB,	Cloudant, NOSQL DB	Firebase, Google's BigData tool, BigQuery, Go, Riptide IO, PubSub	Connected Product Management	NoSQL Database	Semantic model-driven architecture, Symfony2, URSA, hydra-py, Hydrus, Levanzo, Argolis, hydra-core, Go	Hadoop, goDB Cassandra NoSQ	GSN-WRAPPERS, Generic serial wrapper, Generic UDP wrapper, TI-RFID wrapper, USB camera wrapper, TinyOS wrapper, HTTP generic wrapper	MATLAB dashboard
Service discovery	Discovery API, ECS Event Stream, AWS Lambda, Amazon Route 53, Netflix Heureka, etcd,	Azure container service with kubernetes, Zookeeper Netflix Eureka, Consul, Eureka	Discovery Knowledge Graph, Watson Discovery	Consul, etcd, ZooKeeper	Cloudera Navigator	Java WSDP	REST API	MQTT with Kaa protocol v1	REST HTTP query, sbt 0.13+, Java JDK 1.7, Scala 2.11	*

	HashiCorp Consul, AWS App Mesh									
Security and privacy	Auditing, encryption, authorization, authentication	Encryption, authorization, authentication	Authorization, authentication	Authentication	Encryption, authorization	Authenticati on, authorization	Authentication authorization, encryption	Encryption	Authentication, access control mode	Encryption
Pricing	Executing customers functions requires payment	Payment based on messages per day and number of devices	Payment based on data storage, data traffic and number of connected devices	Per MB	Per device	Based subscription	Free	Per device	*	Free or based on standard license
Persistency (Session Persistence)	Persistent sessions based on MQTT 3.1.1 features	CmdKey, Azure Storage Persistence	Persistent iSCSI, JPA 2.0 persistence, WSJPA, OpenJPA, EclipseLink	MQTT v3.1.1 brokers, CloudMQTT, DIoTY, IBM Bluemix, ThingStudio	MQTT 3.1.1 broker	Load balancer	Machine learning algorithms	*	*	Using MQTT
Stream processing	AWS Lambda	SQL query language, JavaScript, C#	IBM Streams toolkits	Semios, GCP Console, Firebase SDK, ImageMagick	Semios, GCP Console, Firebase SDK, ImageMagick	Oracle event processing, oracle continuous query language	CEP queries, Esper EPL	*	SQL queries.	MATLAB
References	[254]	[255]	[256]	[274]	[275]	[259]	[263]	[276]	[277]	[265]

Table 7: Comparison between IoT middleware considering different aspects (Cont.)

IoT middleware	**Aura**	**Calvin**	**NODE-RED**	**Ptolemy Accessor Host**	**Akka**	**Hermes**	**Gryphon**	**Rebeca**	**Fiware**
Middleware architecture	Service-Based	Actor-Based	Actor-Based	Actor-Based	Actor-Based	Event-Based	Event-Based	Event-Based	Context-Aware Event-Based
Open source \ open API	Open source	Open source	Open source JS	Open source	Open source	Open source	Open source	Open source	Open source
Device abstraction\ Interoperability	Connectors, Task abstraction	Actor model (event-driven)	Web services	Accessor	Aggregate programming	Active message abstraction, 5-layers architecture by Fenix, Pegasus	Information flow graph between devices, broker	HTTP, SNMP, RMI	IoT Agent framework library
Deployment type	IaaS, SaaS	IaaS	PaaS, SaaS	*	*	PaaS	SaaS	PaaS	PaaS
Network connectivity	MQTT, HTML	MQTT	HTTP, MQTT	HTTP, HTML	HTTP, HTML	KQML, Fipa ACL, HTML, XML	HTML, HTTP	HTTP, SNMP, Java RMI	MQTT, WebSocket, HTTP
Data format supported	RESTful API, JSON	JSON	JSON	JSON, XML	JSON	JSON, Hermes XML	NASDAQ, NYSE, JSON	XML	HTTP, JSON-LD

Programming languages supported	JavaScript, PHP, C++, python	C, python	JavaScript, Node.js	JavaScript, C++, C	Java, Scala	Java, Python, C, UML	Python, Java	.NET, C#, Java	C++, Java
Application development functionalities	Real-time applications, connecting GUI to a real-time application, online video services, billing systems, consoles, and mobile devices, smart TVs	Distributed applications, runtime applications	For connecting to IoT, connecting and binding to databases, collecting and storing IoT data for processing and in event-driven applications	Finite state machine applications, web applications	Real-Time streaming applications, building powerful and concurrent, web applications	Internet-based distributed applications, large-scale ubiquitous applications, web service	Exchange connections, ledger accuracy guarantees, state tracking, fault tolerance, monitoring, machine learning, quantitative analysis	Monitoring and management, fault Tolerance, publishing methods	Collecting and processing data, visualization, and analysis of data, data access control, monetization or publication, publisher-subscriber communications
Technologies used for application development	OWL, ZMQ, SPARQL, MongoDB	MicroPython	Bluemix, MongoDB	CapeCode, Nashorn, TDL, AJAX, Vert.x, XMLHttpRequest, Simulink/Stateflow, LabVIEW, SCADE	Spray, play framework, apache-spark, socko web server, event-sourced library, Gatling stress test tool, Scalatra, Vaadin, apache flink	Type-based routing algorithm, type, and attribute-based routing algorithm, service agents, AIXO, WS^{2}A, OMSA, lightTS-SA	Heartbeats, RabbitMQ, Java Message Service (JMS), BKS+99, information flow graph, publisher-hosting broker,	Java management extensions, object-oriented API, IMyPub, SetCurrency,	FIWARE Context Broker, eProsima Fast-RTPS
Service discovery	Environment manager	Calvin control APIs	Bonjour / Avahi	Discovery.js discovery function	Akka discovery method, Kubernetes API, AWS, Consul, Marathon API	Service agents, yellow page service, discovery component, matchmaker service agent	*	Publish\subscribe mechanism	selection component, FIWARE NGSI RESTful API, eProsima Fast-RTPS, eProsima Micro-RTPS
Security and privacy	Authentication, authorization	Authorization, authentication	Authentication, encryption	Authentication, encryption.	Authentication, authorization encryption	Authentication, encryption	Authentication, auditing	Authentication, authorization, encryption	Authentication, authorization,
Pricing	Free	Free	Per hour or one invoice per month	Free	Free	Free	*	*	Free
Persistency (Session Persistence)	Aura-session	Distributed hash table	MQTT	Local file system	Akka persistence library	Java Persistence	Buffered stream, JMS persistent events	Fault tolerance plugins, sliding window scheme	Apache flume, MySQL, MongoDB, PostgreSQL
Stream processing	Aura library	Data flow processing	node-red-contrib-cep	Discrete event director	Akka HTTP, Akka stream library and Apache Flink, Lagom	Open RTSP	Relational subscriptions service	*	FIWARE Kurento, WebRTC
References	[266]	[278]	[279]	[280] [281]	[270] [271]	[282] [272]	[272] [283]	[284] [272]	[285]

- **Privacy and security**: Most of the middleware solutions restrict the application of security mechanisms in authentication and authorization, this is due to the resource-constrained devices in the IoT environment. Thus, privacy and security issues need to be end-to-end and lightweight to suit the communication between cloud, gateway, and sensors.

7. Simulation tools of IoT Networks

Simulations are utilized to model system behavior at a certain time, where the simulation environment mimics and evaluates a realistic scenario before building or implementing it in a real-life environment. Simulations are commonly used to estimate easily the performance and cost effects on complicated systems. Using simulation tools to emulate IoT context is indispensable as it supports assessing efficiently the performance of any application, because of the accuracy and the reliability of the results that are provided. Diverse simulators have been built and proposed to mimic the behavior of mobile and distributed applications with several approaches, by making them compatible with many operating systems as Linux and Windows. However, every simulator has its particular configuration requirements, which permits distinctive application aspects to be simulated. In general, any IoTs simulator should offer high reliability when it simulates the scenarios that include heterogeneous sensors, provides computation or energy efficiency estimation, supports scalability, and be able to support new requirements such as any new protocol [286]. Specifying a suitable tool to simulate the IoTs environment is a challenging task since there are only a few simulation tools that have been designed for IoT applications. IoT simulators are classified according to the level of architectural layer and to the scope, they cover into three categories [287]:

1. **Full Stack Simulators:** These simulators have been developed as a consequence of IoT revolution to provide users with the ability to simulate IoT elements and devices. The main simulators in this category are Devices Profile for Web Services Simulator (DPWSim) and iFogSim [288] [289].

   - **DPWSim:** It is a cross-platform simulator that enables the development and the simulation of different IoT applications, where the essential role of this platform is to create virtual IoT devices that can be discovered on IoT networks and can also communicate with each other through DPWS protocols [288]. Besides that, this simulator has a management tool that allows users to create, load, store, and manage their applications with high flexibility. The graphical user interface of DPWSim is designed by Java language, which permits IoT users of interacting with their virtual environments smoothly. Finally, this toolkit helps in developing, prototyping and testing the DPWSim functionalities, but the main drawback of this simulator that it has no support for new technologies and protocols [286].
   - **IFogSim:** This platform was emerged through upgrading and extending the capabilities of the CloudSim simulator [290]. It allows the simulation of different IoT applications and the management of diverse resources that are distributed across the cloud and the edge of the network under various conditions and scenarios [289]. IFogSim permits users to evaluate different resources management that is applicable in Fog environments according to their influence on energy consumption, latency, operational cost, and network congestion. Furthermore, it supports the simulation process of different types of actuators and sensors by enabling the developer to build realistic network topologies.

2. **Big Data Processing Simulators:** These simulators concentrate on processing big data and evaluating the performance of cloud resources, where the main simulators in this category are CloudSim [291], SimIoT [292], and IoTSim [293].

   Cloudsim: It is a toolbox utilized for modeling, experimenting, and simulating a cloud-computing environment. Developers and researchers can design a particular cloud system via this toolkit without any concern about low-level details of the cloud environments and the services they provide [291]. The library functions of the cloudsim is written using Java programming language and it consists of the main classes that are needed to mimic virtual machines, servers, and clients to perform computational assets and to build applications. Furthermore, in order to set up a cloud environment, designers must utilize many simulation components such as virtual machines, data-centers, cloudlets, cloud coordinators, and data center brokers [294].
   - **SimIoT:** It is derived from SimIC simulator and has been developed to mimic large-scale resources management [292] [295]. SimIoT is used to estimate the time needed for processing data that is submitted either by IoT users or sensors to a particular cloud, which is done by using numerous methods to simulate the communication between the cloud and IoT sensors [296].
   - **IoTSim:** This simulator was developed by [297] to simulate the behavior of IoT applications that are responsible for processing big data that is produced from various devices using the MapReduce framework. The vital contributions of this simulator lie in allowing simulation and modeling of a network using virtual machines, permitting the processing of IoT data through using big data framework (MapReduce), and supporting the IoT applications model.

3. **Network Simulators:** The growing of interest toward the field of WSNs has led to the booming of current simulators [298]. The election process of a suitable simulator is a critical and time-consuming mission, particularly in the WSNs domain, since there are many complicated scenarios and numerous protocols utilized in this domain that need specific features to exist in a network simulator. Particular requirements of WSNs and the availability of a vast number of simulators make it difficult to select a suitable simulator. Numerous WSNs simulators have been adapted to suit the simulation process of IoT environments such as Cooja [299], QualNet [300], CupCarbon [301], OMNeT++ [302], and NS-3 [303].
   - **Cooja:** It is a discrete event and a flexible simulator, since several parts of Cooja functions can be extended or replaced by new functionalities such as OS, sensor node platforms, radio transmission models, and radio transceivers [298] [299]. Cooja is developed and written in java language and runs over the Contiki operating system. However, this simulator is not very efficient for many reasons as it requires a lot of calculations to deal with cross-level simulations, there is no GUI interface, and the simulation process supports up to 10000 nodes only.

- **QualNet**: It is a tool that allows network designers to create a virtual scenario of all forms of video, data and voice networks. Any network scenario consists of nodes that represent WSNs elements and endpoints (switches, routers, ground stations, access points, mobile phones, satellites, firewalls, radios, servers, sensors, and other security equipment) and links that connect these nodes (Wi-Fi signals, internet circuits, LAN segments, LTE connections radio transmissions, etc.) [300]. The graphical user interface permits network designers to build their projects in 2D and 3D environments. Also, it allows the analysis of statistical data and packet tracing for debugging purposes [298].
- **CupCarbon:** It is an IoTs WSN and smart city simulator that aims to visualize, design, compile and validate the algorithms that are required for monitoring and collecting environmental data [304]. Furthermore, this simulator helps the researchers to test their wireless models and protocols. CupCarbon provides two simulation environments; the first one permits the generation of natural events like fires and it also supports the simulation of mobile entities such as flying objects and vehicles. On the other hand, the second simulation environment allows designers to represent discrete event scenarios of WSNs. Also, it grants WSNs designers the ability to simulate scenarios and algorithms in many steps as the following; a step for specifying designated nodes, another step to determine the communication types between these nodes, and finally determining routing to the base station. This simulator supports many IoTs communication protocols such as Lora, ZigBee, and WiFi.
- **OMNeT++:** It is a discrete event network simulator that is developed using C++ language by OpenSim company [302]. This simulator consists of GUI libraries for tracing, debugging and animating any network scenario. It also has graphical tools that enable building simulations and performing results computations. OMNet++ permits the hierarchical organization of any simulation scenario, because the number of layers is not restricted. The processes inside the virtual network such as drawing data flow charts, illustrating network graphics and displaying variables or objects during simulation are visualized through a graphical user interface [305]. The structure of the scenario is defined by using network description files (NED) that can be modified by the user via a graphical interface or a text file, where NED files are separated from the simulator to efficiently support the simulation of large topologies. Further, OMNeT++ is distinguished from other simulators in its ability to modify topologies in run time.
- **NS-3:** It is a discrete event simulator that is developed by C++ and Python language [286]. NS-3 permits researchers to analyze large-scale systems and different internet protocols in a controlled environment. This simulator has been improved to provide an open-source and an enormous network simulation platform, for the sake of supporting the education and the research in wireless networks. Concisely, NS-3 provides users with a simulation engine to conduct their simulation experiments and provide them with models that show how data packets perform and work. Furthermore, this simulator supports having multiple radio interfaces and channels for the same node [306]. Many wireless communication protocols can be implemented via NS-3 such as 802.15.4 and 6LoWPAN, but it does not support the protocols of the application layer [287].

To the best of our knowledge, there is no simulator that can be used to build a fully detailed representation of any IoT project until now. Consequently, to simulate a complete IoT project, multiple simulators should be used together such as data generation, big data processing, and packet tracing simulators. Table 8 shows a comparison between different IoT simulators based on popular IoT criteria and features, where the justification for each selected criterion is explained as follows:

- **Scope:** This criterion specifies the level of coverage for different architectural layers of IoT, where (IoT) means that the simulator has full coverage.
- **Last update:** It represents the time of the last maintenance or upgrading that is performed on the simulator.
- **Language:** It refers to the programming language of the simulator and reflects the portability degree of the simulated primitives to be used in subsequent hardware models.
- **Type:** It illustrates basic assumptions regarding the simulated entities and the relationships among them.
- **Layer of IoT architecture:** Represents the architectural layer(s) components, standards, and parameters that are supported by a specific simulator.
- **Evaluated scale:** The maximum network scale that can be simulated and provided through performing simulator evaluations.
- **Mobility:** Determines whether the simulator supports objects mobility or not.
- **Built-in IoT standards:** Specifies different protocols that are supported by a simulator.
- **Overall practicality:** It is a specific measure to indicate the utility behind simulating all components and services in the IoT environment.
- **Target domain:** Indicates specialization degree.
- **Cyberattack simulation:** It indicates if the simulator supports security simulations.

Table 8: Comparison between different IoT simulators

Simulator	Scope	Last Update	Language	Type	Layer(s) of IoT Architecture	Evaluated Scale	Mobility	Built-in IoT Standards	Overall Practicality	Target Domain	Cyber Attack Simulation
DPWSim [288]	IoT	2016	Java	Event-driven scenarios, resource-constrained environments|	Application	Small scale	No	Devices Profile for Web Services (DPWS)	Medium	Generic	No

				and service-oriented [307]							
iFogSim [289]	IoT	2018	Java	Discrete event	Perceptual Network/ Application	Large scale	No	No	Medium	Generic	No
Cloudsim [291]	Data analysis	2016	Java	Discrete event	Application	large scale	Yes	Yes	High	Cloud Analyst	Yes
SimIoT [292]	Data analysis	2014	Java	Discrete event	Application	Small scale	No	No	Medium	Generic	No
IoTSim [297]	Data analysis	2017	Java	MapReduce model	Application	Large scale	No	No	Medium	Generic	No
Cooja [298]	Network	2018	C /C++	Discrete event	Perceptual Network	Small scale	Yes	Supports all IoT protocols	High	Generic with Focus on power constrained sensors	Using custom extension-ns
QualNet [300]	Network	2017	C /C++	Discrete event	Perceptual Network	Large scale	Yes	Zigbee /802.15.4	Medium	Smart city	Yes
CupCarbon [304]	Network	2017	Sen Script	Discrete event and agent-based	Perceptual Network	Large scale	Yes	LoRaWAN/ 802.15.4	High	Generic	No
OMNeT++ [302]	Network	2018	C++	Discrete event	Perceptual Network	Large scale	Yes	Manual extension	Medium	Generic	Using custom extension-ns
NS-3 [286]	Network	2018	C++/ Python	Discrete event	Perceptual Network	Large scale	Yes	LoRaWAN 802.15.4 6LoWPAN	High	Generic	No

8. IoT Applications

The Internet of Things is a modern communication model that envisions a close future, where devices of everyday life will be equipped with transceivers, microcontrollers, sensors, actuators, and appropriate communication protocols that will allow them to communicate with each other and with other clients [308] [309] [310] [311]. IoT aims to make the internet immersive and pervasive through enabling easy access and interaction with a wide diversity of IoT devices as surveillance cameras, monitoring sensors, and home appliances. IoT will promote the development of several applications that utilize the gigantic and diverse amount of data, which is generated by smart devices to provide modern services for companies, citizens and organizations [312] [313].

8.1 Sensors in IoT Applications

An IoT network can commonly be described as an area that is occupied by smart sensors, which sense and control the IoT environment [314] [315] [316] [317]. A sensor node is defined from an engineering point of view as an object that converts chemical, biological, physical or mechanical parameters into an electrical signal. These sensors are used to measure different parameters like wind speed (an anemometer), solar radiation or temperature (thermometer), where an IoT application requires to include at least one type of sensors to collect data from the IoT environment [314]. Sensor technology is continually improving, accordingly these devices become cheaper, smaller, more energy-efficient, and more intelligent. This will enable more applications to be implemented and disseminated such as; environmental monitoring, disaster management, domestic, human health, public security and early warning systems. Van Laerhoven and Schmidt provided an overview of diverse types of sensors that can be utilized in constructing IoT applications. The following section provides a concise preview of these sensors:

1. **Light sensor:** It is an electronic device used to detect light. The main function of these sensors is to provide information about the light density, intensity, type (artificial, sunlight), color temperature (wavelength), and light reflection. There are many types of light sensors like photodiode, UV-sensors, color sensors, IR sensors, etc. The light sensor is considered to be a rich source of data at a very low cost, as it has low energy consumption.
2. **Audio and microphone sensor:** It provides information about various sound types (noise, music, speaking) with minimum processing capabilities.
3. **Accelerometer sensor:** It provides information about the motion, the acceleration or the inclination of any mobile device, where angular sensors, accelerometers, and mercury switches are examples of this type of sensors.
4. **Location sensors:** These sensors provide important information about collocation, location, proximity, and position of devices, users or environment. Many applications can be applied using this type of sensors such as GPS, GSM, and active badge systems [4] [318] [319].
5. **Touch sensors:** Smart devices, which are handled by users, could profit from this type of sensors, as it can be implemented directly with a specific conductive surface, such as skin conductance or indirectly via temperature sensors or light sensors. These sensors tend to reduce energy consumption significantly, particularly for devices that operate in the user's hand.

6. **Temperature sensors:** These sensors are distinctive as they are easy to use and very cheap. Thus, they can be implemented in many applications such as temperature measurement, fumes and flue gases, body heat detection, and applications of rubber and plastic manufacturing processes, etc.
7. **Pressure sensor:** It is utilized to measure many parameters such as the pressure of liquids or gases, altitude or water level.
8. **Medical Sensors**: Improving the efficiency of biomedical systems and the healthcare infrastructure is one of the most challenging objectives in this era, due to the need of offering quality care to patients with low costs, as well as tackling the shortage problem in nursing staff. IoT sensors can be utilized to resolve the aforementioned issues through monitoring and measuring several medical parameters like blood glucose levels, heart rate, blood pressure, respiration rate, pulse rate and body temperature in the patient's body without any human interference. Medical applications aim to remotely monitor a patient's health and consequently, transfer the sensed data directly to the doctors to take a proper decision [28].
9. **Neural Sensors:** Nowadays, it is easy to comprehend neural signals that come from the human brain, deduce the brain state and train it for a better focus and attention. These operations are known as neuron feedback, while the technology utilized in this operation is called Electroencephalography or also known as a brain-computer interface and totally depends on the electromagnetic field that surrounds humans' brains. This field is generated as a result of the communication between the neurons of the human brain and it is measured in terms of frequencies. Human brain signals can be classified according to their frequencies into gamma, theta, beta, delta, and alpha. Depending on the signal type, it can be concluded whether the brain is wandering in thoughts, calm, etc. in order to train the brain later on to be more focused, have better mental well-being, manage stress and to pay better attention towards things [27] [28].
10. **Environmental and Chemical Sensors:** These sensors are utilized to detect physical, biological, and chemical environmental parameters such as pressure, temperature, humidity, air pollution, and water pollution [320]. A barometer and a thermometer measure the pressure and the temperature parameters, while the air quality is measured through sensors that detect the presence of gases and other polluters in the air. Chemical sensors comprise of transducer and recognition part, where electronic tongue (e-tongue) and electronic nose (e-nose) are examples of applications that are developed depending on this technology [321]. Both of e-nose and e-tongue applications are based on the data generated by chemical sensors, which will be then analyzed by different pattern recognition to identify the stimulus. Furthermore, environmental and chemical sensors play a major role in monitoring the level of pollution in smart city applications [28].
11. **Mobile Phone-Based Sensors:** Today smartphones not only serve as a means of communications and computing operations, but they also provide a valuable set of embedded sensors [322]. These sensors enable the deployment of many applications in various domains, such as accelerometer, camera and microphone, magnetometer, GPS and light sensor.

8.2 Recent IoT Applications

This paradigm finds applications in many distinctive aspects such as medical aids, home automation, mobile healthcare, industrial automation, elderly assistance, smart city, smart grid and many other applications [42]. In this section, some of these applications will be summarized as follows:

8.2.1 Smart Cities

The application of the IoT field toward urban domains is of particular interest. This is coming from a strong motivation of numerous national governments to adopt information and communication technology (ICT) in the management of public affairs, hence realizing the so-called Smart City concept [323]. Smart city aims to make superior utilization of public resources as shown in Figure 16 and to decrease operational costs of public management on many traditional public services such as lighting, transporting and parking. Also, it supports the surveillance of public areas and garbage collection, while it increases the QoS that is offered to the citizens. Furthermore, the collected information from urban environments could be used to improve the awareness of the citizens about the status of their city. Despite the aforementioned benefits, the smart city market has not truly taken off yet, for a number of technical, political, and budgetary obstacles.

Table 9 shows different types of services with their appropriate communication protocols, expected traffic, maximum acceptable delay, source of energy for each service and finally an estimation on the feasibility of each service based on the technology it implements. The following subsections explain different services that can be deployed in a smart city.

Table 9: Services specification for the smart city project [323]

Service	Network types(S)	Traffic rate	Tolerable delay	Energy Source	Feasibility
Structural health	802.15.4; Wi-Fi and Ethernet	1 packet every 10 min per device	30 min for data;10 sec for alarms	Mostly battery powered	Easy to achieve but seismograph could be difficult to integrate
Waste management	Wi-Fi;3G and 4G	1 packet every hour per device	30 min for data	Battery-powered or energy harvesters	Possible to achieve but needs smart bins
Air quality monitoring	802.15.4; Bluetooth and Wi-Fi	1 packet t every 30 min per device	5 min for data	Photovoltaic panels for each device	Easy to realize however greenhouse sensors may be from the cost wise expensive
Noise monitoring	802.15.4 and Ethernet	1 packet every 10 min per device	5 min for data;10 sec for alarms	Battery-powered or energy harvesters	Sound pattern recognition is difficult to be implemented on resource-constrained devices
Traffic congestion	802.15.4; Bluetooth; Wi-Fi and Ethernet	1 packet every 10 min per device	5 min for data	Battery-powered or energy harvesters	Needs the realization of both noise monitoring and air quality
City energy consumption	PLC and Ethernet	1 packet every 10 min per device	5 min for data; tighter requirements for control	Mains powered	Simple to achieve, but requires the permission of power operators

Smart parking	802.15.4 and Ethernet	On-demand	1 min	energy harvester	Smart car parking systems are available on the markets, so these projects are easy to be implemented
Smart lighting	802.15.4; Wi-Fi and Ethernet	On-demand	1 min	Mains powered	Requires upgrading the existing infrastructure
Automation and salubrity of public buildings	802.15.4; Bluetooth; Wi-Fi and Ethernet	1 packet every 10 min for remote monitoring,1 packet kt every 30 min for local control	5 min for remote monitoring, few seconds for local control	Mains powered, and battery-powered	Needs intervention on the existing infrastructure

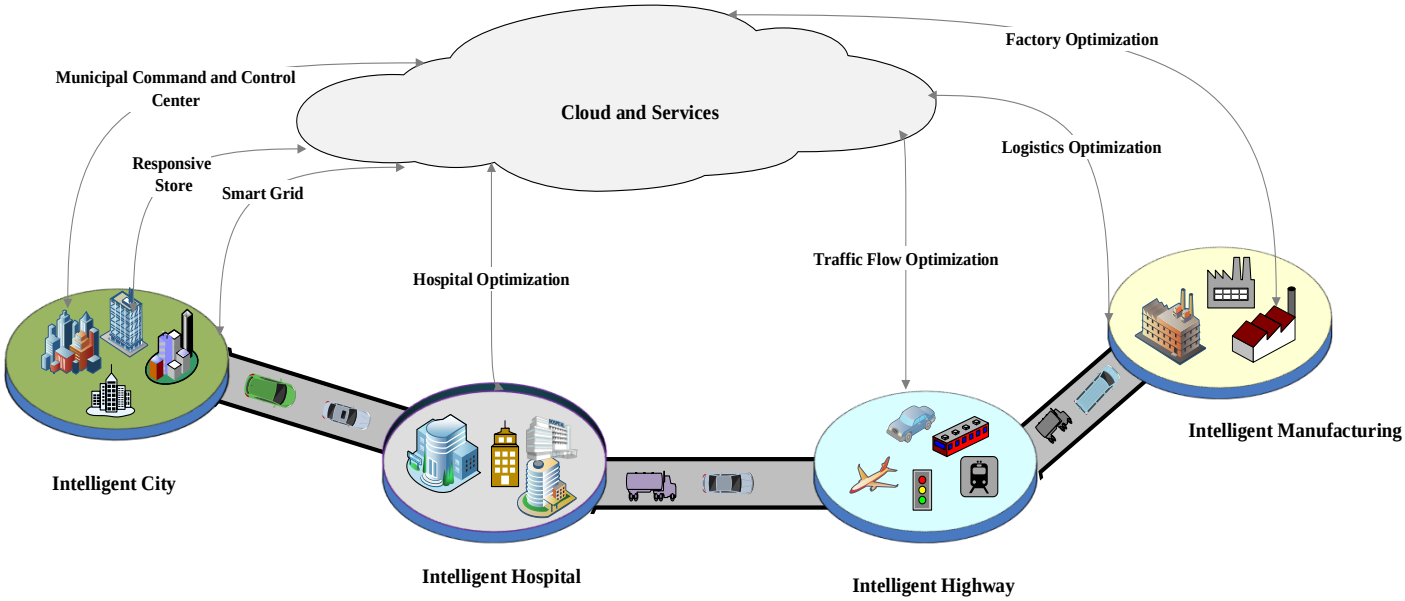


Figure 16: Smart city applications

8.2.1.1 Structural Health of Buildings

This service requires continual monitoring of the specifications of the areas that are prone to the effects of outside agents and the conditions of every building. IoT sensors that are deployed in these buildings should construct a database containing information about the measurement of building structural integrity [324]. There are many types of IoT sensors that can be deployed in this area such as distortion and vibration, which are responsible for measuring buildings stress, atmospheric sensors for sensing pollution level of the surrounding area and the sensors that are responsible for measuring the temperature and the humidity of the environment [323]. Employing IoT technology in this field reduces the cost of human periodic checking on building health through deploying a number of wireless sensors on the building and the surrounding area.

8.2.1.2 Waste Management

Waste disposal is an essential problem in many modern cities, because of both the storage constraints of garbage in landfills and the cost of this service. Applying IoT in this domain will lead to significant ecological advantages and significant cost savings. For example, the utilization of smart garbage collection to detect waste level and to optimize the garbage truck route will decrease the cost of the garbage collection process and will enhance the quality of recycling. To attain these objectives, IoT must connect the smart garbage collectors with a control center that processes the sensed data by an optimization software and then determine the best management of this operation [323].

8.2.1.3 Air Quality and Noise Monitoring

Sound and air pollution are escalating problems nowadays. It is important to monitor air quality and keep it within acceptable limits for a healthy living and a better future for all organisms. Air quality monitoring gives estimations of gases and toxic concentrations to be then analyzed and interpreted, allowing authorities to monitor air pollution in distinctive zones, consequently taking action against any pollution. In such a way, individuals can find the healthiest places to practice outdoor exercises, also they can access their favoured training applications that are connected to IoT infrastructure [325] [326].

8.2.1.4 Traffic Congestion

Traffic management is an issue that most cities confront today. Investing in smart traffic solutions makes sense, as more than half of the world's population were reported living in cities in 2012 [28]. Hence, many cities try to improve transportation by deploying smart services like smart traffic signals and developing applications for smart parking. Furthermore, improving smart transportation systems will increase transportation capacity and make traveling safer, efficient, and secure [327].

Embedding IoT sensors in smart traffic areas will alleviate congestion, respond rapidly to any accident or incident, and manage the daily traffic in smart transportation environments. The major objectives of smart transportation systems are to minimize traffic congestion and provide the individual with hassle-free and easy parking. Furthermore, it will help to avoid accidents by properly routing the traffic and informing the drivers about other bad drivers [28]. Sensors technologies that control these applications are

accelerometers for speed, RFIDs for vehicle identification, GPS sensors for location, gyroscopes for direction, and cameras for recording traffic and vehicle movements. The aforementioned sensors are utilized in the following applications:

1. Traffic monitoring and management applications: Each vehicle in a smart city that is connected to other vehicles and a cloud in a wireless network must be occupied with at least one IoT sensor such as RFID, GPS, cameras to assess traffic conditions in distinctive areas. Traffic congestion is detected using smartphone sensors like GPS and accelerometers, where these sensors are used to detect vehicle movement patterns while the individual is driving. The sensed data then will be sent to map applications in order to be analyzed and subsequently guides the drivers to select the best-uncrowded paths [28] [328].
2. Applications to ensure safety: Several IoT applications have been created to help drivers become safer through monitoring their driving behavior and subsequently guiding them to drive safely, this is done through determining when they are feeling tired or drowsy and aiding them to deal with such situations or suggesting taking a rest. There are many IoT applications that monitor drivers' behaviors such as eye movement recognition, face detection, and pressure detection on the steering wheel [329] [330]. Such applications can be deployed on a smartphone that is occupied by a gyroscope, GPS and accelerometer sensors, which allow analyzing the sensed data to take a suitable decision for safer driving.
3. Parking Guidance and Information (PGI) systems: In order to solve vehicle parking issues, different kinds of PGI applications have been proposed. These systems supply vehicle drivers with the data that help them in finding available parking places in their controlled zones, via virtual message signs on the street or via web applications [331] [332]. PGI systems try to reduce traffic congestion by helping drivers find free parking places without squandering time in looking for a vehicle park. The parking application consists of four primary components, which are parking space information distribution, parking surveillance instrument, control center, and communications network. Also, it uses bar-code machines and barriers to count the number of vehicles that are entering and departing specific park region. Thus, by using PGI applications neither parking supervisor nor drivers are required to know the occupancy status of a particular parking space. Furthermore, PGI systems set up cameras and sensors near the parking zone for vehicle detection and monitoring. These sensors are classified into two types; the on-Roadway and the off-Roadway sensors, where on-Roadway sensors are implemented on the road surface, while the other type sensors are distributed above the road as shown in Figure 17 [333].
4. Smart traffic lights: These sensors are prepared with sensing, processing, and communication abilities, which allow them to sense the traffic jam and the amount of activity going on a specific way. The sensed data will be analyzed and then will be transmitted to a contiguous traffic light or a central controller in order to take suitable action, for instance in an emergency circumstance the traffic lights allocate a lane to an ambulance. Briefly, technologies that are required to build smart traffic lights applications are communication protocols, cameras, and data analysis systems [28].
5. Accident detection applications: Smartphone applications can detect the occurrence of any road incident with the assistance of acoustic information and accelerometer sensors. It instantly transmits the data with additional circumstantial data, such as onsite and area images to the closest hospital. Subsequently, the first responder will know about the whole situation and the degree of medical assistance that is required to present an appropriate degree of help**.**

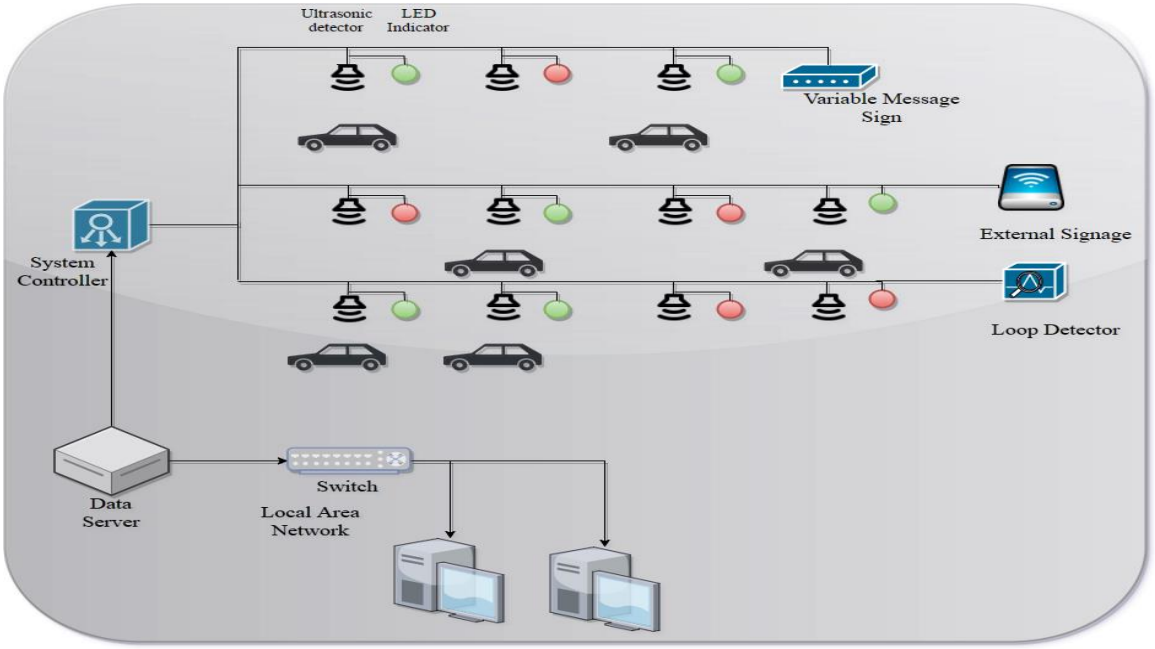


Figure 17: PGI system architecture

8.2.1 5 Smart Grid

It is an electrical network that smartly connects and integrates the activities of many users, whether they are producers or consumers or those that do both, to effectively afford economic, sustainable and secure power resources. A Smart Grid employs inventive services and products with intelligent surveillance, self-healing technologies, communication, and control to accomplish the following purposes [334]:

1. Permitting customers to optimize the operations of the smart grid system.

2. Providing customers with more data and choices of power supply.
3. Significantly diminishing the environmental effect on the entire power supply systems.
4. Providing enhanced levels of security and reliability on power supply systems.
5. Enabling distribution of the generation and utilization processes of renewable energy resources.

8.2.1.6 Automation and Salubrity of Public Buildings

This significant application aims to achieve salubrity of the environment and to alleviate energy consumption problem in public buildings such as museums, administration offices and schools [323], which improves the level of comfort for the individuals, enhances the efficiency, while it decreases the costs of heating and cooling [335]. This is accomplished by utilizing appropriate types of actuators and sensors that control humidity, temperature, and lights.

8.2.1.7 Smart Water Systems

It develops a modern approach that promotes water security from significant future risks such as rapid urbanization, population growth, weak policies, aging infrastructure and climate changes, where these factors will increase the burdens on water resources. Water is delivered to consumers through complex distribution systems. Thus, these systems should supply potable and safe water with adequate pressure. Nevertheless, any failure that infects these systems will lead to waste and declination of the quality of water. Hence, a novel water management procedure is robustly required to carefully control water distribution network and to detect any deficiencies promptly. The primary objectives of Smart Water Networks (SWNS) are to construct a complete surveillance system, data acquisition, integrating sensors technology, securing the gathered information, information analysis, and take decisions in real-time [336]. SWN operation comprises of many steps, where the first step is to have a schematic visualization to collect full information of water network, like pipes, tanks, air valves, pumps, and stabilizers, in order to group them in the next step in geographic information framework. After that, a set of sensors will be deployed to continuously sense many water parameters such as pressure, quality, and flow. Finally, the sensed data will be transmitted through communication channels to be analyzed by an information system to take a suitable action [337].

8.2.2 Medical and Healthcare Applications

Wireless body area networks and WSNs that are utilized in both healthcare and medical applications have received an important interest, as they have major roles in remote monitoring of a patient's situation in real-time, life quality enhancement of the elderly via smart environment, drugs and medical database administrator, avoidance of critical patient situations, welfare services, etc. According to that, it is clear that applying IoT in medical applications will improve radically medical environments [313]. For example, smart health applications allow elderly and patients who are suffering from serious health conditions to live independently apart from hospital restrictions, through utilizing IoT sensors, which continually monitor and record different parameters of their health conditions. Subsequently, delivering warnings in case of finding any unusual indicator. Smart sensors, which are dedicated, for healthcare can measure, monitor and analyze different health status conditions such as heart rate, blood pressure, oxygen saturation in the blood and glucose levels. After measuring the aforementioned parameters, the sensed data will be transmitted to a specific database in order to be analyzed and accordingly to take a proper action, which will enhance the patient's health as shown in Figure 18 [28] [313].

Stress recognition is another healthcare application that is based on sensors of smartphones, which sense the stress level of an individual. This can be achieved through measuring physiological and behavioral data such as blood pressure, skin conductance, heart rate, pupil diameter, and cortisol level to identify whether the person is feeling stress or not [338].

8.2.3 Agricultural Applications

Agriculture plays an important role in any country's economy as it provides extensive employment opportunities for individuals. However, numerous factors affect this field such as soil moisture, carbon dioxide and changes in temperature, which affect the crops, yield. Thus, it is vital to have surveillance systems on these factors to manage harvest growth and to raise agricultural production yield by deploying IoT sensors in agricultural areas [339]. These sensors are able to monitor different environmental parameters such as humidity, temperature, barometric pressure, and luminosity. Any agricultural smart application comprises of two sides, the transmitter side and the gateway receiver side. The transmitter side consists of many sensors that are connected to a wireless network in order to sense different agricultural parameters, while the receiver side monitors and analyzes the sensed parameters, which will be displayed by a user through a web interface as shown in Figure 19.

8.2.4 Smart Home (SH)

SH technology has changed individual life by providing connectivity between everyone and everything regardless of the place and the time. This application changes a traditional home into an automated building with installed and controlled smart devices such as heating, air conditioning, ventilation, security systems and lightings as shown in Figure 20. These systems, which consist of sensors and switches that are sometimes called gateways, communicate with a central station that can be controlled through a user interface installed in a mobile phone, tablet or computer and managed by IoT technology [340]. Smart home system aims to improve domestic comfort, security, leisure, and convenience, while minifying energy consumption through optimizing domestic energy management techniques [341]. SH applications are characterized by the following features:

1. Compatibility with distinctive communication protocols: It can merge numerous heterogeneous communication techniques through installing different communication interfaces on a home gateway.
2. Widespread services: With the utilization of widespread access networks, real-time smart home data can be obtained easily regardless of where the clients are.
3. Comprehensive perception: Real-time surveillance of domestic and comprehensive perception can be attained by deploying an assortment of physical and logical sensors.

4. Easily to be controlled: Since SH applications can be managed via mobile phones, PCs and other communication devices.

8.2.5 Smart Manufacturing System (SMS)

Maintainable manufacturing competitiveness relies on its capabilities with respect to quality, cost, delivery, and flexibility [342]. SMS tries to maximize those capabilities through utilizing advanced technologies, which promote quick flow and widespread utilization of digital data inside and among manufacturing systems. Also, it integrates information and communication technologies with smart software applications to:

1. Enhance the utilization of material, energy, and labor to produce high quality and customized items to be delivered on time.
2. Rapidly respond to changes in supply chains and mart demands.

Smart manufacturing model is distinctive from other manufacturing paradigms as it determines a vision of the next manufacturing generation with improved capabilities [343]. SMS adapts to any new circumstances by utilizing real-time information for intelligent decision-making and by predicting and preventing any failure proactively.

8.2.6 Internet of Robotics Things (IoRT)

In diverse industries or even in offices or homes, robotics come in all sizes and shapes from greeting robotics in restaurants, retail stores or hotels to heavy robot arms in factories. The internet of robotics is an emerging technology that integrates robots as an object into an IoT environment to enable connections through different protocols. IoRT integrates smart robots through the internet to perform personal activities or different professional operations as monitoring activities and events, controlling objects in the real-world and manufacturing. In IoRT application, multiple intelligent sensors and smaller robots are connected and collaborated in an orchestrated manner to achieve the goal of large robotic. There are many applications that are implemented through IoRT such as a self-driving vehicle, software robots to avoid human errors and save time, Smart Manufacturing (Industry 4.0), adaptive digital factory and automated IT processing applications

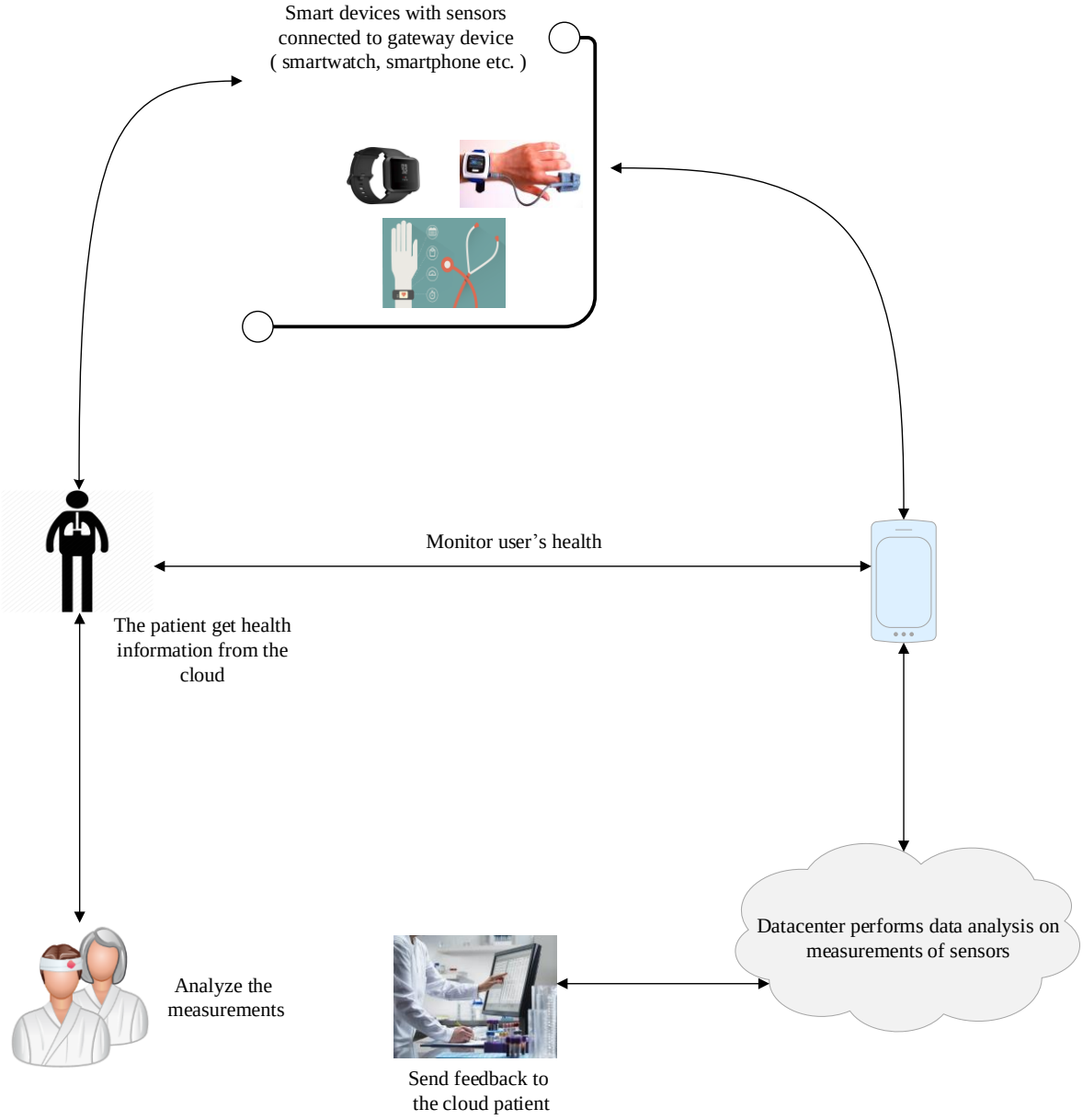


Figure 18: Smart healthcare system

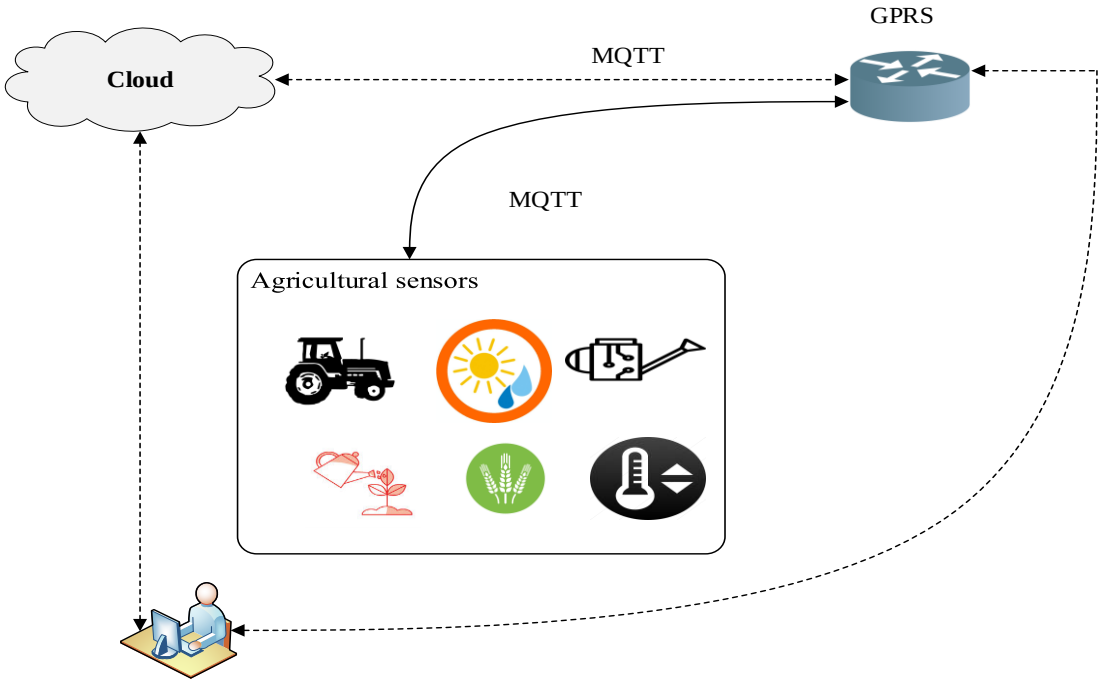


Figure 19: IoT agricultural application

Figure 20: Applications of smart home technology [344]

8.2.7 Oil and Gas

IoT paradigm has found its way through the oil and gas domain. As of now, many IoT companies help factory managers, field staff and machine operators to improve production, protect the safety of employees in work environments and predict the time at which machines require maintenance. IoT technology permits machines, devices, and equipment to collaborate and communicate with each other, which will enable oil and gas companies to create applications, manage and store data and utilize suitable security protocols based on scientific methodologies.

9. Broad and Open Research Challenges

IoT is a fabulous technology concept that for a long period was merely a dream. Nowadays, IoT has taken the world by the storm and it is expanding with an unbelievable rate. Morgan Stanley predicted that the number of smart and heterogeneous devices that will be connected to the internet would exceed 75 billion devices in 2020 [41] [345]. However, IoT services, applications, and devices face numerous issues and challenges that are deemed to be a primary hindrance in the implementation of IoT from different

aspects, such as coverage and protocols, communication technology, energy-saving, bandwidth efficiency, interoperability and integration, memory management, signal acquisition and processing, scalability, deployment, security, fog computing, and computational limitations [346]. Further details are provided below [12] [21] [347]:

1. **Building smart environments based on IoT paradigm**: The first defiance in creating an intelligent environment is to embed countless smart devices, sensors and supplementary technologies in that environment and setting up communication between them. Another issue is to gather and transform massive amounts of data between smart things, which leads to medium contentions and collisions issues.
2. **Privacy and security of IoT applications**: The heterogeneity of IoT communication technologies and the diversity of its applications and services will lead to various sets of security challenges. Protecting the security of IoT architecture from different attacks and potentially malicious software necessitates utilizing many security measures. Those measures are relevant to protection laws, privacy enhancement technologies, privacy tools and standards to control individual information (data privacy), personal physical location movement (location privacy), and various security methodologies.
3. **Compatibility**: Devices from diverse vendors will be connected and embedded to the IoT network, so issues regarding monitoring and tagging will arise. These issues can be solved under the condition that all manufacturers agree on the same standards, which is impossible to be attained [348].
4. **Scalability***:* IoT is expected to face a lot of challenges associated with the probable abundant number of co-operating entities besides the major differences in the interaction behaviors and patterns. Thus, current IoT architecture requires to be scaled up in order to accommodate the rising of intelligent devices number.
5. **Energy Efficiency**: Small smart devices that compose IoT systems, suffer from limited battery power that is impossible to be replaced, which will lead to ultimately global energy crisis and heavy power consumption, memory and processing capabilities. Based on that, routing processes and compute-intensive applications cannot run appropriately on these devices. Keeping in mind the constrained energy of smart devices is not adequate to suit the utilization of WSN routing protocols. Even though some routing protocols support low-power communication, but they are at the infancy stage of development.
6. **Mobility Management**: Mobile nodes in IoT environments can create many confrontations in terms of the efficiency of routing protocols and IoT networks. The existing mobility protocols of sensor networks, mobile ad hoc networks, and vehicular ad hoc networks cannot cope with different routing issues, due to the limited processing and energy capabilities of these sensors.
7. **Cost of maintenance and services:** IoT environments contain an extensive number of connected devices, which will, in turn, increase the cost required for maintenance and servicing. One solution to mitigate this problem is to produce devices and sensors in such a way that they require less maintenance.
8. **Internet disconnection problem:** Since the internet connection is the soul of IoT, thereby the disconnections in internet services will degrade the performance of IoT devices and lead to poor QoS. Also, a limited number of concurrent devices that can communicate with the base station will reduce the number of service recipients.

10. Conclusions

The emerging notion of IoTs technology has swiftly disseminated throughout our contemporary life, where it aims to optimize the quality of our life by embedding smart things, applications, and technologies to automate all things in the environment that surrounds us. What distinguishes our survey paper from other works is that it covers the most important sides of the IoT paradigm, with a concentration on what has been done and what has required more research. Specifically, this paper presents an overview of IoT evolution, its stack's protocols, technologies, applications, and the research challenges facing the implementation of this technology. This, in turn, provides a good ground for the researchers who are whether interested in designing realistic IoT projects or developing novel theoretical approaches in the IoT field by acquiring deep knowledge in different IoT aspects. Furthermore, some of the prevalent issues and challenges that face the deployment and the design of IoT applications were discussed. Future research directions have been further described considering IoT stack and middleware architectures. Additionally, this paper presents the interaction between different IoT network components, which are smart nodes, fog nodes, and cloud computing nodes. Lastly, details of IoT application domains were demonstrated followed by not only open research issues, but also rigorous analysis of the research history along with efficient recommendations.

10.1 Research History Analysis and Recommendations

The motivation behind this research is to inform the researcher's community with depth and breadth of recent and future works in different IoT domains. A massive number of researches in different IoT fields have been published in different conferences and journals. The explosive expansion of IoT technology has opened many scientific and engineering opportunities and issues, which will require huge research efforts from different sectors such as industries, communications, academics, etc. The collaborative efforts and works of these sectors will create novel services, technologies, architectures and protocols, which are necessary to face the challenges of IoT. To demonstrate the ongoing research work in IoT stack architecture and middleware, we used Scopus database to extract the number of publications from 2011 until 2020. Figure 21 displays the distribution of ongoing researches and developments in various layers of IoT architecture, where it denotes a trend towards research in the network layer as it has the largest number of researches compared to other layers with 1183 publications. Network layer developments and enhancements have taken the attention of the researchers' community since it confronts many focal emerging design challenges that require to be tackled and ameliorated swiftly, as specifying the optimal route that guarantees the security and accuracy of IoT data transmission. Other issues that may encounter this layer include finding the best procedures to control network bottlenecks and congestions, appropriate management for different IoT QoS metrics (i.e. transmit time, throughput and efficiency, delay, availability, jitter, etc.), and overcoming issues caused by networks heterogeneity such as various routing protocols, distinct identity techniques, etc. In other words, coping with the network layer security challenges is as important as solving the above issues, where these challenges can be

a denial of service attack, a man in the middle attack, a storage attack, and an exploit attack. The number of researches and publications of data link layer and communication protocols was 931, where numerous traditional enterprise communication technologies, such as Ethernet and Wi-Fi, have been adopted or evolved to be utilized in IoT environments. Simultaneously, many new communication protocols have been developed to face the challenges and requirements of harsh IoT environments, where devices, distances, and bandwidth challenges have to be considered. They also, find new mechanisms to detect and correct the corrupted data, control medium access for broadcast networks, and keep both transmitter and receiver synchronous in data transmission to avoid overwhelming the receiving side with data (i.e. flow control). Similar to a network layer, the data link layer is prone to many security problems that need to be faced and solved by researchers such as address resolution protocol spoofing that permits an attacker to masquerade as a legal host and subsequently intercept, alter or stop data, in addition to MAC flooding and dynamic host configuration protocol attacks. The application layer occupied the third rank in the number of researches with 561 publications. In fact, the bulk of the responsibility for the development and improvement of this layer lies on the programmers and developers. As they are in charge of ensuring that all IoT devices present a consistent interface that abstracts their internal and heterogeneous details, which will guarantee to organize and transfer data smoothly among these devices. On the other hand, they should continually improve and provide users with applications interfaces (i.e. API) that permit them to control, calibrate, and diagnose their devices, which will promote the integrity of control applications. However, the application layer is exposed to many attacks, that endeavor to adversely affect the normal operation of the system, such as physical attacks by overwhelming devices with dummy stimuli, eavesdropping, reprogramming, denial of service attacks and physical capture. Hence, more research efforts are needed to produce new methods that detect and mitigate such attacks. The transport layer came in last in the number of researches as it got 231 publications. This layer is responsible for describing the nature and the quality of delivering data, as well ensuring that messages are transmitted in-order, error-free with no duplications or losses, and establishing an end to end connection. Thus, it requires more attention to fabricate novel methods and procedures that conquer many transport layer issues such as high packet loss, low bandwidth requirement, error control, flow control, congestion control, low power, low memory availability alongside the prevalent security issues.

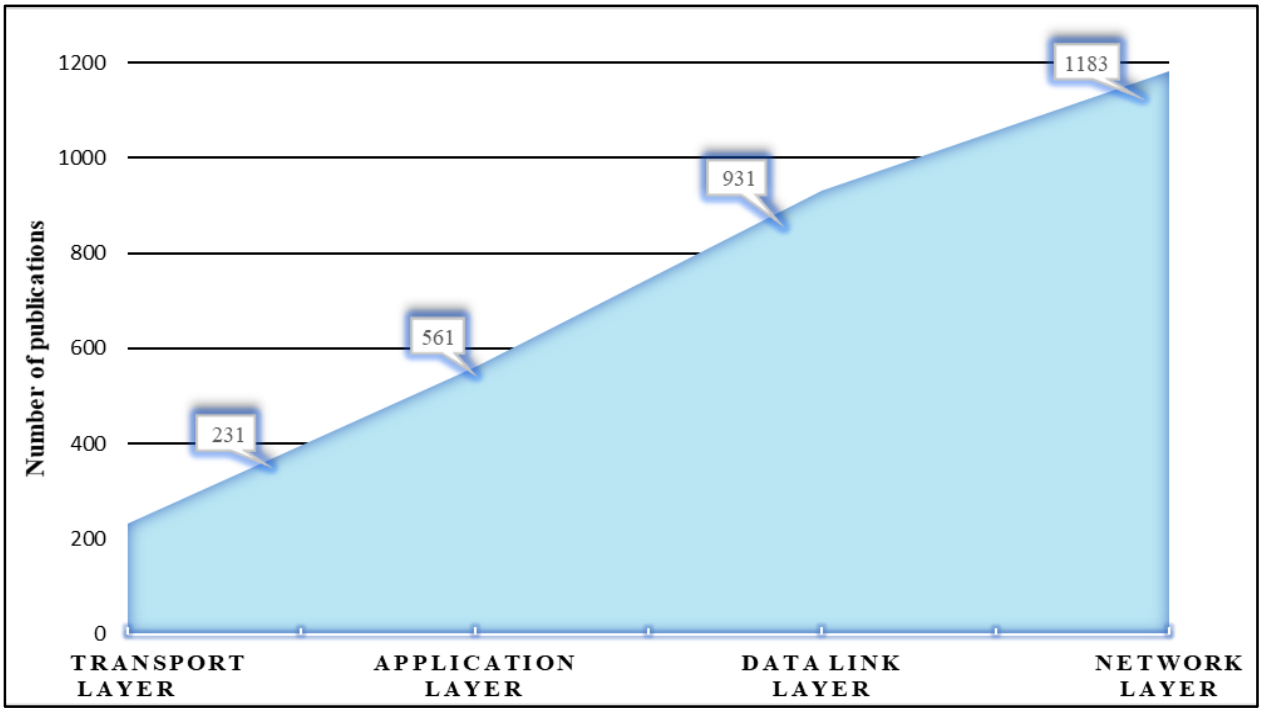


Figure 21: Number of publications in IoT layers from 2011 until 2020

As previously mentioned in section 6, middleware can be defined as a set of sublayers or a software layer interposed between the application and the technological layers. Interestingly, the essential role of this model is to hide the different technologies of IoT assets, which will consequently keep programmers away from problems that are not pertinent to their concern. It also prevents them from having to be aware of rigorous details related to the heterogeneous technologies in the lower layer. Middleware acquires more prominence and attention owing to its primary role in simplifying the creation of services and applications as well as integrating conventional technologies into new ones. However, the nature of the IoT environment makes the role of middleware challenging and difficult since the services that are provided by smart things are usually device-dependent, less reliable, mobile and dynamic. Moreover, middleware solutions have to address functional components such as service composition, registration and discovery and non-functional needs, such as ease of deployment, privacy, security, availability, reliability, timeliness, and scalability. Furthermore, IoT middleware must include architectural properties that offer programming distributiveness, autonomy, context awareness, adaptability, interoperability, and abstraction. It can be seen from Figure 22 that a few contemporary studies have been qualitatively evaluated and surveyed in different architectures of IoT middleware, especially in actor-based architecture which only has 5 publications regarding it. This model was proposed to cope with parallel programming and processing (i.e concurrent programming) in high-performance environments. Despite the widespread availability of multi-core processors with high capabilities, minimal research was found in this field due to the fact that the concurrent programming used in this model is error-prone, complicated to implement, and exhibits indecisive behaviors that make it difficult to predict and address.

Based on the above, the major issues need to be addressed by the research community are the lack of progress (i.e. deadlocks, livelocks) and message protocol violations (i.e. message order violation, bad message interleaving, memory inconsistency). Event-based and cloud-based architectures are not much better in this regard, with the former having 21 publications and the latter having 75. The event-based model, as we stated in section 6, presents interesting features to build highly decoupled and distributed applications, where each of them assumes a specific structure of notifications, application scalability degree and on the way that permit consumers to announce their interest regarding some event. In spite of that, this architecture faces many challenges such as events delivery guarantee, lack of operational tools, and data and transaction management along with processing events in order, particularly when the same consumer runs over multiple instances. As illustrated previously, cloud-based architecture was proposed to meet several requirements of complex analytical services. Today, the emergence of new services and applications that implement time-critical control loops and cannot be performed in the cloud because of insufficient bandwidth or unpredictable delays, creates new challenges that need to be solved. Furthermore, one of the most critical issues that requires further research efforts is the limited security support provided by the cloud-based architecture, since it cannot be applied in resource-constrained IoT devices. Service-based architecture is considered to be one of the most efficient designing styles, as it provides many interesting features for applications and users such as availability, scalability, reusability, and platform independence, which is why the number of publications in this field reached a high of 421 published papers as compared to the lesser amount of research done regarding its brethren subjects. Despite the above features, this architecture endures many open research issues such as delays, service identification, service discovery, complex service management, and it does not suit GUI applications that require heavy data traffic besides homogeneous applications.

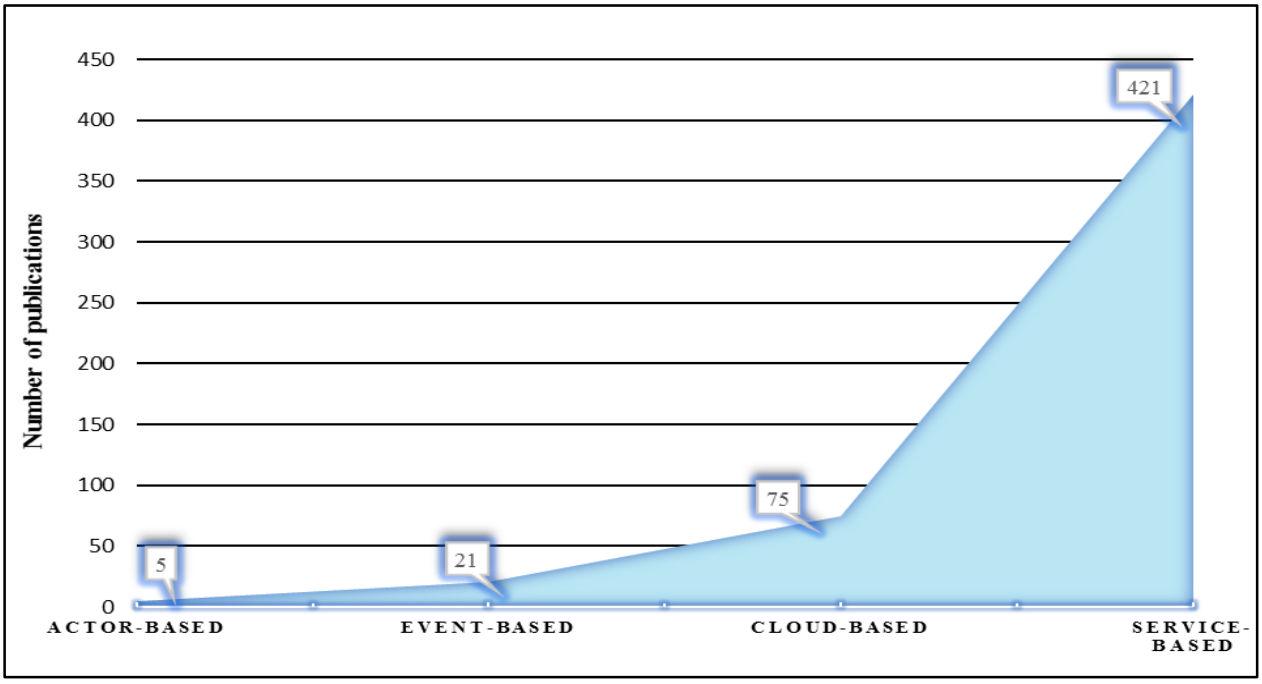


Figure 22: Number of publications in different middleware architectures from 2011 to 2020

References

[1] L. Yan, Y. Zhang and L. T. Yang, The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems, New York: Auerbach publications,Taylor and Francis Group, 2008.

[2] K. Anshton, "That 'internet of things' thing in the real world, things matter," 26 June 2009. [Online]. Available: https://www.rfidjournal.com/articles/view?4986&fbclid=IwAR1r5XFF5_2gcuze1fdcPrN6ryzRc68xhR1XyRD2aGulYFYy1QRzr_G4oas. [Accessed 17 February 2019].

[3] K. Darabkh, W. Albtoush and I. Jafar, "Improved Clustering Algorithms for Target Tracking in Wireless Sensor Networks," The Journal of Supercomputing, vol. 73, no. 5, p. 1952–1977, 2017.

[4] K. Darabkh, M. El-Yabroudi and A. El-Mousa, "BPA-CRP: A Balanced Power-Aware Clustering and Routing Protocol for Wireless Sensor Networks," Ad Hoc Networks, vol. 82, pp. 155-171, 2018.

[5] K. Darabkh, N. Al-Maaitah, I. Jafar and A. Khalifeh, "Energy Efficient Clustering Algorithm for Wireless Sensor Networks," in International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2017.

[6] H. Steven, "The Internet of Things How the world will be connected in 2025," Master thesis,Utrecht University, 2016.

[7] R. Hamidouche, Z. Aliouat, A. Ari and M. Gueroui, "An efficient clustering strategy avoiding buffer overflow in IoT sensors: a bio-inspired based approach," IEEE Access, vol. 7, pp. 156733 - 156751, 2019.

[8] K. Darabkh and M. Al-Yabrodi, "A Reliable Relaying Protocol in Wireless Sensor Networks," in European Conference on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, 2017.

[9] M. Alhasanat, S. Althunibat, K. Darabkh, A. Alhasanat and M. Alsafasfeh, "A Physical-Layer Key Distribution Mechanism for IoT Networks," Mobile Networks and Applications, p. 1–6, 2019.

[10] K. Darabkh, S. Odetallah, Z. Al-qudah and A. Khalifeh, "A New Density-Based Relaying Protocol for Wireless Sensor Networks," in 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, 2018.

[11] D. Uckelmann, M. Harrison and F. Michahelles, "An Architectural Approach Towards the Future Internet of Things," in Architecting the Internet of Things, Berlin, Heidelberg, pringer-Verlag, 2011, pp. 1-24.

[12] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems,Elsevier, vol. 29, no. 7, pp. 1645–1660,, 2013.

[13] L. Tan and W. Neng, "Future Internet: The Internet of Things," in International Conference on Advanced Computer Theory and Engineering(ICACTE), Chengdu, China, 2010.

[14] K. Darabkh, J. Zomot and Z. Al-qudah, "EDB-CHS-BOF: energy and distance-based cluster head selection with balanced objective function protocol," IET Communications , vol. 13, no. 19, pp. 3168 - 3180, 2019.

[15] K. Darabkh, S. Odetallah, Z. Al-qudah, A. Khalifeh and M. Shurman, "Energy–aware and density-based clustering and relaying protocol (EA-DB-CRP) for gathering data in wireless sensor networks," Applied Soft Computing, vol. 80, pp. 154-166, 2019.

[16] K. Darabkh, M. Alfawares and S. Althunibat, "MDRMA: Multi-data rate mobility-aware AODV-based protocol for flying ad-hoc networks," Vehicular Communications, vol. 18, 2019.

[17] K. Darabkh, M. .Judeh, H. BanySalameh and S. Althunibat, "Mobility aware and dual phase AODV protocol with adaptive hello messages over vehicular ad hoc networks," AEU- International Journal of Electronics and Communications, vol. 2018, pp. 277-292, 2018.

[18] K. Darabkh and O. Alsukour, "Novel Protocols for Improving the Performance of ODMRP and EODMRP over Mobile Ad Hoc Networks," International Journal of Distributed Sensor Networks (IJDSN) , vol. 11, no. 10, 2015 .

[19] Lu Tan and N. Wang, "Future Internet: The Internet of Things," in International Conference on Advanced Computer Theory and Engineering(ICACTE), Chengdu, China, 2010.

[20] L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A survey," Computer Networks, Elsevier, vol. 54, no. 15, p. 2787–2805, 2010.

[21] M. Daniele, S. Sabrina, P. Francesco and C. Imrich, "Internet of things: vision, applications and research challenges," Ad Hoc Networks, Elsevier, vol. 10, no. 7, p. 1497–1516, 2012.

[22] O. Said and M. Masud, "Towards internet of things: survey and future vision," International Journal of Computer Networks (IJCN), vol. 5, no. 1, 2013.

[23] A. Whitmore, A. Agarwal and L. Da Xu, "The Internet of Things-A survey of topics and trends," Information Systems Frontiers, vol. 17, no. 2, 2015.

[24] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communications Surveys and Tutorials , vol. 17, no. 4, p. 2347–2376, 2015.

[25] S. Kraijak and P. Tuwanut, "A survey on internet of things architecture, protocols, possible applications, security, privacy, real-world implementation and future trends," in International Conference on Communication Technology (ICCT), Hangzhou, China, 2016.

[26] P. Masek, J. Hosek, K. Zeman, M. Stusek, D. Kovac, P. Cika, J. Masek, S. Andreev and F. Kröpfl, "International Journal of Distributed Sensor Networks," Implementation of True IoT Vision: Survey on Enabling Protocols and Hands-On Experience, vol. 2016, 2016.

[27] R. Pratim, "A survey on Internet of Things architectures," Journal of King Saud University - Computer and Information Sciences, vol. 30, no. 3, pp. 291-319, 2016.

[28] P. Sethi and S. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," Hindawi Journal of Electrical and Computer Engineering, vol. 2017, pp. 1-25, 2017.

[29] M. Burhanuddin, A. Mohammed, R. Ismail and H. Basiron, "Internet of Things Architecture: Current Challenges and Future Direction," International Journal of Applied Engineering Research, vol. 12, no. 21, pp. 11055-11061 , 2017.

[30] A. Ngu, M. Gutierrez, V. Metsis, S. Nepal and M. Sheng, "IoT Middleware: A Survey on Issues and Enabling Technologies," IEEE Internet of Things Journal , vol. 4, no. 1, pp. 1-20, 2016.

[31] B. N. Silva, M. Khan and K. Han, "Internet of Things: A Comprehensive Review of Enabling Technologies, Architecture, and Challenges," IETE Technical Review, vol. 35, no. 2, pp. 205-220, 2017.

[32] M. Burhan, R. Rehman, B. Khan and B.-S. Kim, "IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey," Sensors, vol. 18, no. 9, 2018.

[33] H. Atlam, R. Walters and G. Wills, "Internet of Things: State-of-the-art, Challenges, Applications, and Open," International Journal of Intelligent Computing Research, vol. 9, no. 3, 2018.

[34] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A Review of Enabling Technologies, Challenges, and Open Research Issues," Computer Networks, vol. 144, pp. 17-39, 2018.

[35] J. Dizdarević, F. Carpio, A. Jukan and X. Masip-Bruin, "A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration," ACM Computing Surveys, vol. 51, no. 6, 2019.

[36] B. Subramanian, K. Nathani and S. Kumar, "IoT Technology, Applications and Challenges: A Contemporary Survey," Wireless Personal Communications, vol. 108, no. 1, p. 363–388, 2019.

[37] I. Yaqoob, I. Hashem, T. Abaker, A. Ahmed and H. Kazmi, "Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges," Future Generation Computer Systems, vol. 92, p. 265–275, 2019.

[38] V. Balas, K. Solanki, R. Kumar and M. Khari, "The History, Present and Future with IoT," in Internet of Things and Big Data Analytics for Smart Generation, Springer, Cham, 2019, pp. 27-51.

[39] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati and M. Rossi, "Secure Communication for Smart IoT Objects:Protocol Stacks, Use Cases and Practical Examples.," in IEEE Thirteenth International Symposium on "A World of Wireless, Mobile and Multimedia Networks", San Francisco, 2012.

[40] V. Aleksandrovičs, E. Filičevs and J. Kampars, "Internet of Things: Structure, Features and Management," Information Technology and Management Science,Walter de Gruyter Gmbh , vol. 19, no. 1, p. 78–84, 2016.

[41] "Internet of Things in 2020 Roadmap for the Future," 27 May 2008. [Online]. Available: https://docbox.etsi.org/erm/Open/CERP%2020080609-10/Internet-of-Things_in_2020_EC-EPoSS_Workshop_Report_2008_v1-1.pdf. [Accessed 17 February 2019].

[42] G. Matthew and K. Simon, "Internet of Things: Services and Applications Categorization," Advances in Internet of Things, Scientific Research, vol. 1, no. 2, pp. 27-31, 2011.

[43] L. Atzori, A. Iera and G. Morabito, "SIoT: Giving a Social Structure to the Internet of Things," IEEE Communications Letters, vol. 15, no. 11, pp. 1193-1195, 2011.

[44] G. Gan, Z. Lu and J. Jiang, "Internet of Things Security Analysis," in Internet Technology and Applications (iTAP), Wuhan, China, 2011.

[45] R. Khan, U. Khan, R. Zaheer and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," in International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 2012.

[46] A. Luigi, I. Antonio, M. Giacomo and N. Michele, "The social internet of things (siot) when social networks meet the internet of things:Concept, architecture and network characterization.," Computer Networks ,Elsevier, vol. 56, no. 16, 2012.

[47] M. Hassan, B. Song and E. Huh, "A framework of sensor-cloud integration opportunities and challenges," in Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication - ICUIMC, Suwon, Korea, 2009.

[48] A. Khalifeh, K. Rajendiran, K. Darabkh, A. Khasawneh, O. AlMomani and Z. Zinon, "On the Potential of Fuzzy Logic for Solving the Challenges of Cooperative Multi-Robotic Wireless Sensor Network," Electronics, vol. 8, no. 12, 2019.

[49] M. R. Abdmeziem and D. Tandjaoui, "Architecting the Internet of Things: State of the art," in Robotics and Sensor Cloud, Springer, Cham, 2015, pp. 55-75.

[50] S. Chayan, N. A. Uttama, R. P. Venkatesha and R. Abdur, "A scalable distributed architecture towards unifying IoT applications," in 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, 2014.

[51] A. Yousefpour, G. Ishigaki, R. Gour and J. Jue, "On Reducing IoT Service Delay via Fog Offloading," IEEE Internet of Things Journal , vol. 5, no. 1, 2018.

[52] G. Elena, L. María and P. Víctor, "Interacting with Objects in Games Through RFID Technology," in Radio Frequency Identification from System to Applications, London , Intech, 2013, pp. 325-340.

[53] F. Bonomi, R. Milito, P. Natarajan and J. Zhu, "Fog computing: A platform for internet of things and analytics.," in Big Data and Internet of Things: A Roadmap for Smart Environments. Studies in Computational Intelligence, vol. 546, Switzerland, Springer International Publishing , 2014, pp. 169-186.

[54] B. Tang, Z. Chen and G. Hefferman, "A Hierarchical Distributed Fog Computing Architecture for Big Data Analysis in Smart Cities," in Proceedings of the ASE BigData & SocialInformatics, Kaohsiung, Taiwan, October 2015.

[55] P. Adams, 18 August 2017. [Online]. Available: https://knect365.com/cloud-enterprise-tech/article/0fa40de2-6596-4060-901d-8bdddf167cfe/openfog-reference-architecture-for-fog-computing. [Accessed 18 February 2019].

[56] M. Aazam and E. Huh, "Fog Computing and Smart Gateway Based Communication for Cloud of Things," in International Conference on Future Internet of Things and Cloud, Barcelona, Spain, 2014.

[57] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC '12, Helsinki, Finland, 2012.

[58] OpenFog Consortium Architecture Working Group, "OpenFog Reference Architecture for Fog Computing," OpenFog Consortium, USA, 2017.

[59] R. Hamidouche, Z. Aliouat, M. Gueroui, A. Ari and L. Louail, "Classical and bio-inspired mobility in sensor networks for IoT applications," Journal of Network and Computer Applications, vol. 121, pp. 70-88, 2018.

[60] R. Hamidouche, Z. Aliouat, A. M. Gueroui, A. A. A. Ari and L. Louail, "Classical and bio-inspired mobility in sensor networks for IoT applications," Journal of Network and Computer Applications, vol. 121, pp. 70-88, 2018.

[61] A. Munir, P. Kansakar and S. U. Khan, "IFCIoT: Integrated Fog Cloud IoT: A novel architectural paradigm for the future Internet of Things.," IEEE Consumer Electronics Magazine , vol. 6, no. 3, pp. 74-82, 2017.

[62] P. Miguel, P. Octavian and P. Girão, "Spread Spectrum Techniques in Wireless Communication," IEEE Instrumentation and Measurement Society , vol. 12, no. 6, pp. 21-24, 2009.

[63] S. William, Data and computer communications eighth edition, New Jersey: Pearson Education., 2007.

[64] A. Behrouz, Data Communications and Networking 5th edition, United States of America: McGraw-Hill, 2013.

[65] F. Behrouz and C. Sophia, Data Communications and Networking 4th Edition, New York: McGraw-Hill, 2007.

[66] R. Dixon, Spread spectrum systems with commercial applications, 3rd edition, India : Wiley India Pvt. Limited, 1976.

[67] W. Quan, "Non-Linear Chirp Spread Spectrum Communication Systems of Binary Orthogonal Keying Mode thesis," Electronic Thesis and Dissertation Repository, 2015.

[68] P. J and A. Sesay, "Performance of N-ary chirp spread spectrum modulation in the AWGN and broadband multipath channels," Wireless 2004 proceedings, 2004.

[69] C. Quintana, J. Rufo, F. Delgado and R. Jimenez, "Time-Hopping Spread-Spectrum System for Wireless Optical Communications," IEEE Transactions on Consumer Electronics , vol. 55, no. 3, pp. 1083-1088, 2009.

[70] J. Holmes, Spread Spectrum Systems for GNSS and Wireless Communications, Norwood: Artech House, 2007.

[71] K. Saurabh and S. Poddar, "A Review on Communication Protocols Using Internet of Things," in International conference on Microelectronic Devices, Circuits and Systems (ICMDCS) , Vellore, India, 2017.

[72] S. Mahmoud and A. Mahmoud, "A Study of Efficient Power Consumption Wireless Communication Techniques/ Modules for Internet of Things (IoT) Applications," scientific research Advances in Internet of Things, vol. 6, no. 2, pp. 19-29, 2016.

[73] I. Timmins and T. Hazelton, "Self-monitoring cable system". USA Patent US9621262B1, 24 July 2013.

[74] D. Zeng, S. Guo and Z. Cheng, "The web of things: a survey," Journal of Communications, vol. 6, no. 6, pp. 424-438, 2011.

[75] A. A. O. Bahashwan and S. Manickam, "A Brief Review of Messaging Protocol Standards for Internet of Things (IoT)," Journal of Cyber Security and Mobility, vol. 8, no. 1, 2018.

[76] 8 May 2019. [Online]. Available: http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/. [Accessed 15 June 2019].

[77] [Online]. Available: https://1sheeld.com/mqtt-protocol/. [Accessed 14 June 2019].

[78] N. Han, "Semantic service provisioning for 6LoWPAN: powering internet of things applications on web," Institut National des Tel´ ecommunications, Ph.D. dissertation, 2015.

[79] T. Yokotani and Y. Sasaki, "Comparison with HTTP and MQTT on Required Network Resources for IoT," in International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Indonesia , 2016.

[80] P. Saint-Andre, K. Smith and R. Tronçon, XMPP: The Definitive Guide Building Real-Time Applications with Jabber Technologies, United States of America: O'Reilly Media, 2009.

[81] M. Bani Yassein, M. Shatnawi and D. Al-Zoubi, "Application Layer Protocols for the Internet of Things," in IEEE International Conference on Internet of Things and Pervasive Systems, Morocco, 2016.

[82] H. Nguyen and L. Lacono, "RESTful IoT Authentication Protocols," in Mobile Security and Privacy Advances, Challenges and Future Research Directions, Cambridge, United States, Todd Green, 2017, pp. 217-234.

[83] S. Solapure and H. Kenchannavar, "RPL and COAP protocols, experimental analysis for IoT: a case study," International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC), vol. 10, no. 2, 2019.

[84] N. Naik, "Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP," in IEEE International Systems Engineering Symposium (ISSE), Vienna, Austria, 2017.

[85] T. Salman and R. Jain, "A Survey of Protocols and Standards for Internet of Things," Advanced Computing and Communications, vol. 1, no. 1, 2017.

[86] A. Banks, E. Briggs, K. Borgendale and R. Gupta, "MQTT Version 5.0," OASIS Standard, 2019.

[87] K. Ari, K. ETH and H. Klaus, "RESTful Design for Internet of Things Systems," IETF, 2018.

[88] C. Bormann, S. Lemay, H. Tschofenig, K. Hartke and B. Silverajan, "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets," Internet Engineering Task Force (IETF), 2018.

[89] 3 March 2019. [Online]. Available: https://amqp.readthedocs.io/en/latest/changelog.html#version-2-4-2. [Accessed 22 June 2019].

[90] March 2015. [Online]. Available: https://www.omg.org/spec/DDS. [Accessed 22 June 2019].

[91] M. Bishop, "Hypertext Transfer Protocol Version 3 (HTTP/3)," Intended status: Standards Track, 2019.

[92] 19 June 2019. [Online]. Available: https://xmpp.org/extensions/index.html. [Accessed 22 June 2019].

[93] "MQTT v5.0 now an official OASIS standard," 3 April 2019. [Online]. Available: http://mqtt.org/. [Accessed 6 August 2019].

[94] K. Saeed, N. Chaki, B. Pati, S. Bakshi and D. Mohapatra, "Internet of Things: A Survey on IoT Protocol Standards," in Advanced Computing and Intelligent Engineering, Springer, Singapore, 2018.

[95] C. Lesjak, D. Hein, M. Hofmann, M. Maritsch, A. Aldrian, P. Priller, T. Ebner, T. Ruprechter and G. Pregartner, "Securing smart maintenance services: Hardware-security and TLS for MQTT," in 13th International Conference on Industrial Informatics (INDIN), Cambridge, 2015.

[96] A. Tamboli, "What We Built and the Takeaways," in Professional and Applied Computing, Berkeley, Apress,Springer, 2019, pp. 199-209.

[97] L. Silva, "Internet of Things – Pros and cons of CoAP protocol solution for small devices," Master thesis,Mid Sweden University, 2016.

[98] 2 August 2018. [Online]. Available: https://raygun.com/blog/soap-vs-rest-vs-json/. [Accessed 21 June 2019].

[99] [Online]. Available: https://support.kemptechnologies.com/hc/en-us/articles/203863435-RESTful-API. [Accessed 21 June 2019].

[100] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego and J. Alonso-Zarate, "A Survey on Application Layer Protocols for the Internet of Things," Transaction on IoT and Cloud Computing , 2015.

[101] "DDS Security," An OMG DDS Security, Needham, USA, 2018 .

[102] C. Bormann, A. Castellani and Z. Shelby, "CoAP: An Application Protocol for Billions of Tiny Internet Nodes," IEEE Internet Computing, vol. 16, no. 2, 2012.

[103] 2019. [Online]. Available: https://www.engineersgarage.com/Articles/Transport-Layer-Protocols. [Accessed 23 June 2019].

[104] June 2019. [Online]. Available: https://www.iotone.com/term/transmission-control-protocol-tcp/t689. [Accessed 23 June 2019].

[105] L. Eggert, G. Fairhurst and G. Shepherd, "UDP Usage Guidelines," Internet Engineering Task Force (IETF), 2018.

[106] E. Kohler, M. Handley and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," Network Working Group, 2006.

[107] A. A. Ahmed and W. Ali, "A lightweight reliability mechanism proposed for datagram congestion control protocol over wireless multimedia sensor networks," Transactions on Emerging Telecommunications Technologies, vol. 29, no. 3, 2018.

[108] 23 Septemer 2018. [Online]. Available: https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-gprs-sctp.html. [Accessed 25 June 2019].

[109] R. Stewart and C. Metz, "SCTP: new transport protocol for TCP/IP," IEEE Internet Computing, vol. 5, no. 6, 2001.

[110] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," IETF, 2018.

[111] O'Reilly, "Transport Layer Security (TLS)," [Online]. Available: https://hpbn.co/transport-layer-security-tls/. [Accessed 28 June 2019].

[112] "Datagram Transport Layer Security: Revision history," 18 May 2019. [Online]. Available: https://en.wikipedia.org/wiki/Datagram_Transport_Layer_Security. [Accessed 29 June 2019].

[113] F. Baker, A. Weinrib, B. Braden, L. Zhang and S. Bradner, "Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment," Network Working Group, 1997.

[114] R. Braden, D. Estrin, S. Berson, S. Herzog and D. Zappala, "The Design of the RSVP Protocol," USC/Information Sciences Institute, 1995.

[115] M. Rouse, "RSVP (Resource Reservation Protocol)," April 2007. [Online]. Available: https://searchnetworking.techtarget.com/definition/RSVP. [Accessed 28 June 2019].

[116] N. Kortas, "The Performance evaluation of using Quic, TCP and SCTP within Cloud and Cloudlet environments," International Journal of All Research Education and Scientific Methods (IJARESM), vol. 1, no. 1, 2015.

[117] R. Hamilton, J. Iyengar, I. Swett and A. Wilk, "QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2 draft-tsvwg-quic-protocol-02," Network Working Group, 2016.

[118] 17 November 2014. [Online]. Available: http://highscalability.squarespace.com/blog/2014/11/17/aeron-do-we-really-need-another-messaging-system.html. [Accessed 29 June 2019].

[119] M. Thompson, 20 March 2019. [Online]. Available: https://github.com/real-logic/aeron/wiki/Protocol-Specification. [Accessed 29 June 2019].

[120] F. Valsorda, V. Vasiliev, H. Wee, D. Wong, C. Wood, T. Wright, P. Wu and K. Yamamoto, "The Transport Layer Security (TLS) Protocol Version 1.3," Internet Engineering Task Force (IETF), 2018.

[121] R. Gandhi, H. Shah and J. Whittaker, "Updates to the Fast Reroute Procedures for Co-routed Associated Bidirectional Label Switched Paths (LSPs)," Internet Engineering Task Force (IETF) , 2019.

[122] J. Touch, "Transport Options for UDP draft-ietf-tsvwg-udp-options-07," IETF, Manhattan, 2019.

[123] R. Stewart, M. Tuexen and M. Proshin, "Stream Control Transmission Protocol: Errata and Issues in RFC 4960," Internet Engineering Task Force IETF, 2019.

[124] M. Amend, A. Brunstrom, A. Kassler and V. Rakocevic, "DCCP Extensions for Multipath Operation with Multiple Addresses draft-amend-tsvwg-multipath-dccp-02," IETF, 2019.

[125] E. Rescorla, H. Tschofenig and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3 draft-ietf-tls-dtls13-32," IETF, 2019.

[126] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport draft-ietf-quic-transport-22," IETF, 2019.

[127] E. Rescorla, H. Tschofenig and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3draft-ietf-tls-dtls13-01," IETF, 2017.

[128] K. Vignesh, "Master Thesis: Performance analysis of end-to-end DTLS and IPsec-based communication in IoT environments," Blekinge Institute of Technology, Karlskrona, Sweden, 2017.

[129] "[MX] RSVP messages out of sequence," [Online]. Available: https://kb.juniper.net/InfoCenter/index?page=content&id=KB30375&cat=TRAFFIC_ENGINEERING&actp=LIST. [Accessed 28 June 2019].

[130] "MPLS Error Detection," [Online]. Available: https://flylib.com/books/en/2.501.1.43/1/. [Accessed 28 June 2019].

[131] "RSVP ready to manage VoIP, video traffic," 25 January 2007. [Online]. Available: https://searchunifiedcommunications.techtarget.com/news/1242869/RSVP-ready-to-manage-VoIP-video-traffic. [Accessed 28 June 2019].

[132] A. Ghedini, "The Road to QUIC," 26 July 2018. [Online]. Available: https://blog.cloudflare.com/the-road-to-quic/. [Accessed 28 June 2019].

[133] K. Mirja, T. Brian and E. Zurich, "Applicability of the QUIC Transport Protocol draft-ietf-quic-applicability-04," Network Working Group, 2019.

[134] 30 November 2017. [Online]. Available: https://blog.skymind.ai/interview-with-adam-gibson-creator-of-dl4j-why-aeron-matters/. [Accessed 29 June 2019].

[135] W. Eddy, "Transmission Control Protocol Specification draft-ietf-tcpm-rfc793bis-14," Internet Engineering Task Force, 2019.

[136] J. Touch, E. Lear, A. Mankin, M. Kojo, M. Kumiko, S. Lars and M. Eggert, "Service Name and Transport Protocol Port Number Registry," IETF, 2019.

[137] 4 October 2011. [Online]. Available: http://cs-pages.blogspot.com/2011/10/compare-and-contrast-advantages-and.html. [Accessed 27 June 2019].

[138] T. Phelan, G. Fairhurst and C. Perkins, "DCCP-UDP: A Datagram Congestion Control Protocol UDP Encapsulation for NAT Traversal," IETF, 2012.

[139] R. Stewart, M. Tuexen and I. Ruengeler, "Stream Control Transmission Protocol (SCTP) Network Address Translation Support," Network Working Group Internet-Draft, 2019.

[140] [Online]. Available: https://www.ibm.com/support/knowledgecenter/en/ssw_aix_71/com.ibm.aix.networkcomm/sctp_intro.htm. [Accessed 27 June 2019].

[141] J. Mena and R. Rusich, "SCTP: Stream Control Transmission Protocol an analysis," 2006.

[142] D. T and R. E, "The Transport Layer Security (TLS) Protocol Version 1.2," IETF, 2008.

[143] A. Leslie, ""TLS vs. SSL" - 5 Things To Know (Differences, Protocols, & Handshakes)," 8 June 2018. [Online]. Available: https://www.hostingadvice.com/how-to/tls-vs-ssl/. [Accessed 28 June 2019].

[144] P. Levis, K. Pister, R. Struik, J. Vasseur and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," IETF, 2012.

[145] O. Iova, F. Theoleyre and T. Noel, "Using multiparent routing in RPL to increase the stability and the lifetime of the network," Ad Hoc Networks, vol. 29 , 2015.

[146] O. Iova, P. Picco, T. Istomin and C. Kiraly, "RPL: The Routing Standard for the Internet of Things... Or Is It?," IEEE Communications Magazine, vol. 54, no. 12, 2016.

[147] H. Kim, J. Ko, D. Culler and J. Paek, "Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey," IEEE Communications Surveys & Tutorials, vol. 1, no. 1, 2017.

[148] J. Sobral, J. Rodrigues, R. Rabêlo, J. Almuhtadi and V. Korotaev, "Routing Protocols for Low Power and Lossy Networks in Internet of Things Applications," Sensors, vol. 19, no. 9, 2019.

[149] M. Aboubakar, M. Kellil, A. Bouabdallah and P. Roux, "Toward Intelligent Reconfiguration of RPL Networks using Supervised Learning," in Wireless Days (WD), Manchester, United Kingdom, 2019.

[150] T. Salman and R. Jain, "Internet of Things and Data Analytics Handbook," in Networking protocols and standards for internet of things, John Wiley & Sons, 2016, pp. 215-238.

[151] S. Basagni, C. Petrioli, R. Petroccia and D. Spaccini, "CARP: A Channel-aware routing protocol for underwater acoustic wireless networks," Ad Hoc Networks, vol. 34, pp. 92-104, 2015.

[152] B. Santos, M. Vieira and L. Vieira, "eXtend Collection Tree Protocol," in Wireless Communications and Networking Conference (WCNC), New Orleans, LA, USA, 2016.

[153] D. Sharma, "Evaluating and improving collection tree protocol in mobile wireless sensor networ," Thesis, University of Ontario Institute of Technology (UOIT) , Oshawa, Ontario, Canada , 2011.

[154] T. Clausen, J. Yi and U. Herberg, "Lightweight On-demand Ad hoc Distance-vector Routing - Next Generation (LOADng): Protocol, Extension, and Applicability," Computer Networks, vol. 126, pp. 125-140, 2017.

[155] T. Qiu, Y. Lv, F. Xia, N. Chen, J. Wan and A. Tolba, "ERGID: An efficient routing protocol for emergency response Internet of Things," Journal of Network and Computer Applications, vol. 72, pp. 104-112, 2016.

[156] N. Gozuacik and S. Oktug, "Parent-Aware Routing for IoT Networks," in Conference on Internet of Things and Smart Spaces, Balandin , 2015.

[157] C. H. Barriquello, G. W. Denardin and A. Campos, "A geographic routing approach for IPv6 in large-scale low-power and lossy networks," Computers & Electrical Engineering, vol. 45, pp. 182-191, 2015.

[158] Y. Tian and R. Hou, " An Improved AOMDV Routing Protocol for Internet of Things," in International Conference on Computational Intelligence and Software Engineering , Wuhan, China, 2010.

[159] M. Bouaziz, A. Rachedi, A. Belghith, M. Berbineau and S. Al-Ahmadi, "EMA-RPL: Energy and mobility aware routing for the Internet of Mobile Things," Future Generation Computer Systems, vol. 97, p. 247–258, 1 January 2019.

[160] Z. Zhou, B. Yao, R. Xing, L. Shu and S. Bu, "E-CARP: An Energy Efficient Routing Protocol for UWSNs in the Internet of Underwater Things," IEEE Sensors Journal, vol. 16, no. 11, pp. 4072 - 4082, 2015.

[161] S. Sebastian and R. Arockiam, "ELT-EAPR: Expected Life Time of Energy Aware Parent Routing for IoT Networks," International Journal of Pure and Applied Mathematics, vol. 118, no. 8, 2018.

[162] J. Sobral, J. Rodrigues, R. Rabelo, K. Saleem and V. Furtado, "LOADng-IoT: An Enhanced Routing Protocol for Internet of Things Applications over Low Power Networks," Sensors, vol. 19, no. 1, 2019.

[163] S. Hashemian and W. Tabataba, "A Multigate Scheme to Improve CORPL under Traffic Load in Cognitive Radio Based Smart Grids with Mesh Topology," Nashriyyah muhandisi barq va muhandisi kampyutar, vol. 16, no. 4, pp. 229 -238, 2019.

[164] R. Jadhav, R. Sahoo, Y. Wu and D. Zhang, "RPL Observations draft-ietf-roll-rpl-observations-01," IETF, 2019 .

[165] O. Gaddour, A. Koubaa, R. Rangarajan, O. Cheikhrouhou, E. Tovar and M. Abid, "Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism," in IEEE International Symposium on Industrial Embedded Systems , Pisa, Italy, 2014.

[166] S. Abdel Hakeem, A. Hady and H. Kim, "RPL Routing Protocol Performance in Smart Grid Applications Based Wireless Sensors: Experimental and Simulated Analysis," Electronics, vol. 8, no. 2, 2019.

[167] A. witwit and A. Idrees, "A Comprehensive Review for {RPL} Routing Protocol in Low Power and Lossy Networks," in New Trends in Information and Communications Technology Applications, Baghdad, Iraq, Springer International Publishing, 2018.

[168] A. Aijaz and H. Aghvami, "Cognitive Machine-to-Machine Communications for Internet-of-Things: A Protocol Stack Perspective," IEEE Internet of Things Journal, vol. 2, no. 2, 2015.

[169] H. Prasad and S. Babu, "A Survey on Network Routing Protocols in Internet of Things (IoT)," International Journal of Computer Applications, vol. 160, no. 2, 2017.

[170] A. A. Khan, M. H. Rehmani and M. Reisslein, "Requirements, Design Challenges, and Review of Routing and MAC Protocols for CR-Based Smart Grid Systems," IEEE Communications Magazine, vol. 55, no. 5, 2017.

[171] 15 July 2011. [Online]. Available: https://sing.stanford.edu/gnawali/ctp/. [Accessed 3 July 2019].

[172] Z. Aqun, W. Yang, L. Yi and L. Cy, "Research on Dynamic Routing Mechanisms in Wireless Sensor Networks," The Scientific World Journal, vol. 2014, 2014.

[173] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss and P. Levis, "Collection Tree Protocol," in Embedded Networked Sensor Systems, Berkeley, California, USA, 2009.

[174] P. Pecho, P. HanaCek and J. Nagy, "Simulation and Evaluation of CTP and Secure-CTP Protocols," RadioEngineering, vol. 19, no. 1, 2010.

[175] T. Clausen, A. Verdiere and J. Yi, "The Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)," Network Working Group, 2016.

[176] D. Sasidharan and L. Jacob, "Design of Composite Routing Metrics in LOADng Routing Protocol for IoT Applications," in The Sixteenth International Conference on Networks, 2017.

[177] M. Vucinic, B. Tourancheau and A. Duda, "Performance Comparison of the RPL and LOADng Routing Protocols in a Home Automation Scenario," in IEEE Wireless Communications and Networking Conference (WCNC) , Shanghai, China, 2013.

[178] H. Makwana and H. Patel, "Advancement in Performance of Wireless AdHoc Network using AOMDV in MANET," International Journal for Innovative Research in Science & Technology, vol. 4, no. 10, pp. 16-20, 2018.

[179] N. Srinidhi, K. S. Dilip and K. Venugopal, "Network optimizations in the Internet of Things: A review," Engineering Science and Technology, an International Journal, vol. 22, no. 2, pp. 1-21, 2019.

[180] A. Dhumane, R. Prasad and J. Prasad, "Routing Issues in Internet of Things: A Survey," in International MultiConference of Engineers and Computer Scientists, Hong Kong, 2016.

[181] A. P and S. Bhuraria, "Near field communication," SET Labs Bridfings, vol. 10, p. 67–74, 2012.

[182] V. Coskun, B. Ozdenizci and K. Ok, "A Survey on Near Field Communication (NFC) Technology," Wireless Personal Communications,Springer-Verlag , vol. 71, no. 3, pp. 2259-2294, 2013.

[183] K. Curran, A. Millar and C. Garvey, "Near Field Communication," International Journal of Electrical and Computer Engineering (IJECE), vol. 2, no. 3, pp. 371-382, 2012.

[184] Z. Shelby and C. Bormann, 6LoWPAN: The Wireless Embedded the Internet, A John Wiley and Sons, Ltd, Publication, 2010.

[185] J. Hui, David Culler and Samita Chakrabarti, "6LoWPAN: Incorporating IEEE 802.15.4 into the IP architecture," Internet Protocol for Smart Objects (IPSO) Alliance, White paper # 3, USA, 2009.

[186] L. Devasena, "IPv6 Low Power Wireless Personal Area Network (6LoWPAN) for Networking Internet of Things (IoT) – Analyzing its Suitability for IoT," Indian Journal of Science and Technology, vol. 9, no. 30, 2016.

[187] C. Gomez, J. Oller and J. Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology," Sensors, vol. 12, no. 12, pp. 11734-11753, 2012.

[188] L. Mainetti, L. Patrono and A. Vilei, "Evolution of wireless sensor networks towards the Internet of Things: A survey," in SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks, Split, Croatia, 2011.

[189] B. Paolo, P. Prashant, C. Vince, C. Stefano, G. Alberto and H. Fun, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards," Elsevier, vol. 30, no. 7, p. 1655–1695, 2007.

[190] European Commission, "Building radio frequency identification for the global environment," June 2009. [Online]. Available: http://www.bridge-project.eu.. [Accessed 2019 February 2019].

[191] R. Want, "An introduction to RFID technology," IEEE Pervasive Computing, vol. 5, no. 1, p. 25–33, 2006.

[192] C. Luca, D. D. Danilo, M. Luca, P. Luigi, L. S. Maria and T. Luciano, "An IoT-aware Architecture to improve Safety in Sports Environments," Journal of communications software and systems, vol. 13, no. 2, pp. 44-52, 2017.

[193] E. De Poorter, J. Hoebeke, M. Strobbe, I. Moerman, S. Latré, M. Weyn, B. Lannoo and J. Famaey, "Sub-GHz LPWAN Network Coexistence, Management and Virtualization: An Overview and Open Research Challenges," Wireless Personal Communications,Springer-Verlag , vol. 95, no. 1, p. 187–213, 2017.

[194] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui and T. Watteyne, "Understanding the Limits of LoRaWAN," IEEE Communications Magazine, vol. 55, no. 9, pp. 34 - 40, 2017.

[195] C.-S. Sum, G. P. Villardi, M. A. Rahman, T. Baykas, H. N. Tran, Z. Lan, C. Sun, Y. Alemseged, J. Wang, C. Song, C.-W. Pyo, S. Filin and H. Harada, "Cognitive communication in TV white spaces: An overview of regulations, standards, and technology," IEEE Communications Magazine, vol. 51, no. 7, pp. 138 - 145, 2013.

[196] A. Augustin, J. Yi, T. Clausen and W. Townsley, "A Study of LoRa: Long Range and Low Power Networks for the Internet of Things," Sensors, vol. 16, no. 9, 2016.

[197] Z-Wave, "Z-Wave," 2 February 2019. [Online]. Available: https://en.wikipedia.org/wiki/Z-Wave. [Accessed 10 February 2019].

[198] T. Nisha, R. S. Rajat Dwivedi and S. Kapil, "Overview of Technologies Associated with IoT," International Journal of Innovations & Advancement in Computer Science, vol. 6, no. 11, pp. 84-93, 2017.

[199] ACS Wireless, "Telensa Street Light Controls.," 2017. [Online]. Available: https://www.advanced-cx.com/telensa. [Accessed 10 February 2019].

[200] F. Joseph and B. Stephen, "An Analysis of the Energy Consumption of LPWA-based IoT Devices," in International Symposium on Networks, Computers and Communications (ISNCC), Rome, Italy, 2018.

[201] U. Raza, P. Kulkarni and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," IEEE Communications Surveys and Tutorials, vol. 19, no. 2, pp. 855-873, 2017.

[202] Y. Choi, Y.-G. Hong and J.-S. Youn, "Transmission of IPv6 Packets over Near Field Communication draft-ietf-6lo-nfc-11," 6Lo Working Group, 2018.

[203] C. Gomez, S. Darroudi, T. Savolainen and M. Spoerk, "IPv6 Mesh over BLUETOOTH (R) Low Energy using IPSP draft-ietf-6lo-blemesh-05," 6Lo Working Group, 2019.

[204] L. Leonardi, G. Patti and L. Bello, "Multi-hop Real-time Communications over Bluetooth Low Energy Industrial Wireless Mesh Networks," IEEE Access, vol. 6, pp. 26505 - 26519, 2018.

[205] "Zigbee is the only complete loT solution, from the mesh network to the universal language that allows smart objects to work together.," [Online]. Available: https://zigbee.org/zigbee. [Accessed 30 July 2019].

[206] L. A. Technical, "LoRaWAN™ 1.0.3 Specification," LoRa Alliance, 2018.

[207] Y. Seok, "IEEE 802.11AH (WI-FI IN 900 MHZ License-exempt band) for IoT application," 14 August 2016. [Online]. Available: https://www.standardsuniversity.org/e-magazine/august-2016-volume-6/ieee-802-11ah-wi-fi-900-mhz-license-exempt-band-iot-application/. [Accessed 15 July 2019].

[208] "Z-Wave Plus™ Certification," [Online]. Available: https://z-wavealliance.org/z-wave_plus_certification/. [Accessed 19 July 2019].

[209] "5G spectrum: strategies to maximize all bands," [Online]. Available: https://www.ericsson.com/en/networks/trending/hot-topics/5g-spectrum-strategies-to-maximize-all-bands. [Accessed 24 July 2019].

[210] M. Zarri, "Road to 5G: Introduction and Migration," GSMA, 2018.

[211] "Release 17," [Online]. Available: https://www.3gpp.org/release-17. [Accessed 18 July 2019].

[212] "Global Smart Street Lighting & Smart Cities: Market Forecast (2019– 2028)," Northeast group, Washington, 2019.

[213] [Online]. Available: https://www.zigbee.org/zigbee-for-developers/zigbee-3-0/. [Accessed 9 July 2019].

[214] C. Perkins and V. Devarapalli, "Standards Track MN Identifier Types for MIPv6," Internet Engineering Task Force (IETF), 2018.

[215] "How to use NFC on Android," 9 Octobor 2018. [Online]. Available: https://www.androidauthority.com/how-to-use-nfc-android-164644/. [Accessed 8 July 2019].

[216] "Near Field Communication (NFC)," International Journal of Computer Science and Network Security, vol. 12, no. 2, pp. 93-99, 2012.

[217] B. Charrat, "Method for routing incoming and outgoing data in an NFC chipset". USA Patent US 7954,723 B2 , 7 June 2011.

[218] E. Kim, D. Kaspar, N. Chevrollier and J. Vasseur, "Design and Application Spaces for 6LoWPANs draft-ietf-6lowpan-usecases-10," 6LoWPAN Working Group, 2011.

[219] M. Shin, T. Camilo, J. Silva and D. Kaspar, "Mobility Support in 6LoWPAN draft-shin-6lowpan-mobility-01," Network Working Group, 2007.

[220] K. Shahzad and B. Oelmann, "A comparative study of in-sensor processing vs. raw data transmission using ZigBee, BLE and Wi-Fi for data intensive monitoring applications," in 11th International Symposium on Wireless Communications Systems (ISWCS), Barcelona, Spain, 2014.

[221] S. Sirur, P. Juturu, P. Gupta, R. Serikar, K. Reddy, S. Barak and B. Kim, "A mesh network for mobile devices using Bluetooth low energy," in 2015 IEEE Sensors, Busan, South Korea, 2015.

[222] J. Lee, Y. Su and C. Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in 33rd Annual Conference of the IEEE Industrial Electronics Society, Taipei, Taiwan, 2007.

[223] M. Kasraoui, A. Cabani and J. Mouzna, "ZBR-M: a new Zigbee routing protocol," International Journal of Computer Science and Applications, , vol. 10, no. 2, p. 15 – 32, 2013.

[224] R. Singhn, J. Kaur and I. Gill, "Evaluation of Hybrid Topologies under Mobility of ZigBee Devices using Different Trajectories," International Journal of Computer Applications (0975 – 8887), vol. 122, no. 20, 2015.

[225] K. Sattlegger and U. Denk, "Navigating your way through the RFID jungle," Texas Instruments white paper, 2014.

[226] H. Arthaber, T. Faseth and F. Galler, "Spread-Spectrum Based Ranging of Passive UHF EPC RFID Tags," IEEE Communications Letters, vol. 19, no. 10, 2015.

[227] "RFID Standards: ISO, IEC, EPCglobal," [Online]. Available: https://www.electronics-notes.com/articles/connectivity/rfid-radio-frequency-identification/standards-iec-iso-epcglobal.php. [Accessed 10 July 2019].

[228] M. Yarvis, "Mesh networking with RFID publication classification communications". Patent US 2006/0109084 A1 , 25 May 2006.

[229] A. Shah and M. Engineer, "A Survey of Lightweight Cryptographic Algorithms for IoT-Based Applications:," in Smart Innovations in Communication and Computational Sciences, Singapore, springer, 2019, pp. 284-294.

[230] "What is the LoRaWAN® Specification?," [Online]. Available: https://lora-alliance.org/about-lorawan. [Accessed 15 July 2019].

[231] A. Cilfone, L. Davoli, L. Belli and G. Ferrari, "Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies," Future Internet, vol. 11, no. 4, 2019.

[232] [Online]. Available: https://www.wi-fi.org. [Accessed 15 July 2019].

[233] K. Daly, "Why Wi-Fi Halow will revolutionize industrial process controls," 15 October 2018. [Online]. Available: https://www.morsemicro.com/blog/wifi-halow-industrial. [Accessed 15 July 2019].

[234] M. Denatma and D. Perdana, "Simulation and Analysis of Energy Consumption and Performance of Routing Protocol DSDV and OLSR on IEEE 802.11ah Standard," International Journal of simulation, system, science & technology, vol. 1, no. 35, 2016.

[235] "Sigfox Protocol Library for devices," [Online]. Available: https://build.sigfox.com/sigfox-library-for-devices. [Accessed 19 July 2019].

[236] "Comparing IoT Networks at a Glance," [Online]. Available: https://www.wi-sun.org/iot-networks/. [Accessed 16 July 2019].

[237] B. Heile, B. Liu, M. Zhang and C. Perkins, "Wi-SUN FAN Overview," IETF, 2017.

[238] K. Mekki, E. Bajic, F. Chaxel and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," ICT Express, vol. 5, no. 1, pp. 1-7, 2019.

[239] S. Duhovnikov, A. Baltaci, D. Gera and D. Schupke, "Power Consumption Analysis of NB-IoT Technology for Low-Power Aircraft Applications," in IEEE 5th World Forum on Internet , 2019.

[240] S. Farrell, "LPWAN Overview draft-ietf-lpwan-overview-10," IETF, 2018.

[241] 1 October 2014. [Online]. Available: https://z-wavealliance.org/about_z-wave_technology/. [Accessed 8 April 2019].

[242] L. Vora, "Evolution of mobile generation technology: 1G to 5G and review of upcoming wireless technology 5G," International Journal of Modern Trends in Engineering and Research (IJMTER), vol. 2, no. 10, 2015.

[243] "Mobile Communication: From 1G to 4G," 1 February 2019. [Online]. Available: https://electronicsforu.com/technology-trends/mobile-communication-1g-4g. [Accessed 21 July 2019].

[244] "technologies," [Online]. Available: https://www.etsi.org/technologies. [Accessed 21 July 2019].

[245] F. Njoroge and L. Kamau, "A Survey of Cryptographic Methods in Mobile Network Technologies from 1G to 4G," 2018.

[246] "Mass scale smart city technology," [Online]. Available: https://www.telensa.com/. [Accessed 25 July 2019].

[247] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," Cisco Internet Business Solutions Group (IBSG), april 2011.

[248] G. Fortino and P. Trunfio, Internet of Things Based on Smart Objects, Switzerland: Springer International Publishing, 2014.

[249] M. Cruz, J. Rodrigues, A. Sangaiah, J. Al-Muhtadi and V. Korotaev, "Performance evaluation of IoT Middleware," Journal of Network and Computer Applications,Elsiver, vol. 109, p. 53–65, 2018.

[250] G. Kortuem, F. Kawsar, D. Fitton and V. Sundramoorthy, "Smart objects as building blocks for the Internet of things," IEEE Internet Computing , vol. 14, no. 1, pp. 44 - 51, 2010.

[251] A. Farahzadi, P. Shams, J. Rezazadeh and R. Farahbakhsh, "Middleware Technologies for Cloud of Things - a survey," Digital Communications and Networks, Elsevier, vol. 4, no. 3, pp. 176-188, 2017.

[252] N. Peter, "Principled Assuredly Trustworthy Composable Architectures," Principal Scientist, Computer Science Laboratory, California , USA, 2004.

[253] M. Papazoglou and D. Georgakopoulos, "Service-oriented computing: concepts, characteristics and directions," in Proceedings of the Fourth International Conference on Web Information Systems Engineering., Rome, Italy, 2003.

[254] "Service Discovery," [Online]. Available: https://aws.amazon.com. [Accessed 14 August 2019].

[255] [Online]. Available: https://azure.microsoft.com. [Accessed 14 August 2019].

[256] "IBM Watson IoT platform," [Online]. Available: https://www.ibm.com/internet-of-things/solutions/iot-platform/watson-iot-platform. [Accessed 16 August 2019].

[257] xively, 2014. [Online]. Available: http://xively.com. [Accessed 11 February 2019].

[258] N. Sinha, E. Pujitha, J. Alex and R. Sahaya, "Xively Based Sensing and Monitoring System for IoT," in International Conference on Computer Communication and Informatics (ICCCI) - Xively based sensing and monitoring system for IoT, Coimbatore, INDIA, 2015.

[259] [Online]. Available: https://www.oracle.com/internet-of-things/. [Accessed 17 August 2019].

[260] M. Eisenhauer, P. Rosengren and P. Antolin, "A development platform for integrating wireless devices and sensors into ambient intelligence systems," in 2009 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, Rome, Italy, 2009.

[261] HYDRA, 2010. [Online]. Available: http://hydramiddleware.eu. [Accessed 12 March 2018].

[262] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," IEEE, vol. 16, no. 1, pp. 414 - 454, 2013.

[263] "LinkSmart Docs," 16 July 2019. [Online]. Available: https://docs.linksmart.eu. [Accessed 17 August 2019].

[264] Global Sensor Networks, 2004. [Online]. Available: http://lsir.epfl.ch/research/current/gsn/. [Accessed 11 February 2019].

[265] "Understand Your Things The open IoT platform with MATLAB analytics.," [Online]. Available: https://thingspeak.com. [Accessed 18 August 2019].

[266] 3 April 2016. [Online]. Available: https://github.com/AuraMiddleware/aura-middleware. [Accessed 13 August 2019].

[267] P. Persson and O. Angelsmark, "Calvin – merging cloud and IoT," Procedia Computer Science, Elsevier , vol. 52, p. 210 – 217, 2015.

[268] Node-RED, "A visual tool for wiring the Internet of Things," 2015. [Online]. Available: http://nodered.org.

[269] Ptolemy II, 1999 . [Online]. Available: http://ptolemy.eecs.berkeley.edu. [Accessed 11 February 2019].

[270] [Online]. Available: https://doc.akka.io. [Accessed 21 August 2019].

[271] "Akka Part of Lightbend Platform," [Online]. Available: https://www.lightbend.com/akka-part-of-lightbend-platform. [Accessed 21 August 2019].

[272] P. Pietzuch, "Hermes: A scalable event-based middleware," University of Cambridge, Cambridge, 2004.

[273] Z. Theodore, P. Andreas, A. Federico, G. Jose and L. Fernando, "FIWARE lab:managing resources and services in a cloud federation supporting future internet applications," in IEEE/ACM 7th International Conference on Utility and Cloud Computing, London, United Kingdom , 2014 .

[274] [Online]. Available: https://developers.google.com. [Accessed 16 August 2019].

[275] "INTRODUCTION TO XIVELY," [Online]. Available: https://www.developerxively.com/docs. [Accessed 16 August 2019].

[276] [Online]. Available: https://docs.kaaiot.io. [Accessed 17 August 2019].

[277] 29 Septemper 2014. [Online]. Available: https://github.com/LSIR/gsn/wiki/GSN-in-a-nutshell. [Accessed 18 August 2019].

[278] [Online]. Available: https://github.com/EricssonResearch/calvin-base. [Accessed 19 August 2019].

[279] "Node-RED," 16 August 2019. [Online]. Available: https://nodered.org/docs/api/. [Accessed 19 August 2019].

[280] C. Ptolemaeus, System Design, Modeling, and Simulation using Ptolemy II, Ptolemy.org, 2014.

[281] "Ptolemy II," [Online]. Available: https://ptolemy.berkeley.edu/ptolemyII/index.htm. [Accessed 20 August 2019].

[282] M. Bernardo and A. Bogliolo, "Hermes: Agent-Based Middleware for Mobile Computing," in Computer Science, Berlin, Springer, 2005.

[283] "Gryphon Trading Framework 0.12 Documentation," [Online]. Available: https://gryphon.readthedocs.io. [Accessed 23 August 2019].

[284] "REBECA - Publish/Subscribe Middleware," 20 Septemper 2011. [Online]. Available: https://www.ava.uni-rostock.de/en/ava-research/projects/rebeca/. [Accessed 23 August 2019].

[285] "FIWARE step by step," [Online]. Available: https://fiware-tutorials.readthedocs.io. [Accessed 24 August 2019].

[286] M. Sharif and A. Sadeghi-Niaraki, "Ubiquitous Sensor Network Simulation and Emulation Environments: A Survey," Journal of Network and Computer Applications,Elsevier , vol. 93, pp. 150-181, 2017.

[287] M. Chernyshev, Z. Baig, O. Bello and S. Zeadally, "Internet of Things (IoT): Research, Simulators, and Testbeds," IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1637 - 1647, 2018.

[288] S. Han, M. Lee, N. Crespi, K. Heo, N. Van, M. Brut and P. Gatellier, "DPWSim: A Simulation Toolkit for IoT Applications using Devices Profile for Web Services," in 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, South Korea, 2014.

[289] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," Special Issue: Cloud and Fog Computing, Wiley Blackwell (John Wiley & Sons) , vol. 47, no. 9, pp. 1275-1296, 2017.

[290] R. Buyya, R. Ranjan and R. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in 2009 International Conference on High Performance Computing & Simulation, Leipzig, Germany, 2009.

[291] S. Dash, P. Naidu, B. Chandra, R. Bayindir and S. Das, "Analysis of Cloud Environment Using CloudSim," in Artificial Intelligence and Evolutionary Computations in Engineering Systems, Singapore, 2018.

[292] S. Sotiriadis, N. Bessis, E. Asimakopoulou and N. Mustafee, "Towards Simulating the Internet of Things," in 2014 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, Canada, 2014.

[293] X. Zeng, S. K. Garg, P. Strazdins, P. P. Jayaraman, D. Georgakopoulos and R. Ranjan, "IoTSim: A simulator for analysing IoT applications," Elsevier,Journal of Systems Architecture, vol. 72, pp. 93-107, 2017.

[294] "The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne," December 2013. [Online]. Available: http://www.cloudbus.org/cloudsim/. [Accessed 12 February 2019].

[295] S. Stelios, B. Nik, A. Nikos and A. Ashiq, "SimIC: Designing a new inter-cloud simulation platform for integrating large-scale resource management," in IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), Barcelona, Spain, 2013.

[296] A. Markus, G. Kecskemeti and A. Kertesz, "Flexible Representation of IoT Sensors for Cloud Simulators," in 2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), Petersburg, Russia, 2017.

[297] X. Zeng, S. K. Garg, P. Strazdins, P. P. Jayaraman, D. Georgakopoulos and R. Ranjan, "IoTSim: A simulator for analysing IoT applications," Journal of Systems Architecture, Elsevier, vol. 72, pp. 93-107, 2016.

[298] Ž. MIODRAG, N. BOŠKO, P. JELICA and P. RANKO, "A Survey And Classification Of Wireless Sensor Networks Simulators Based On The Domain Of Use," Ad Hoc & Sensor Wireless Networks, Old City Publishing, vol. 20, pp. 245-287, 2014.

[299] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne and T. Voigt, "Cross-level sensor network simulation with Cooja," in Proceedings. 2006 31st IEEE Conference on Local Computer Networks, Tampa, USA, 2006.

[300] "qualnet," 2008. [Online]. Available: https://web.scalable-networks.com/qualnet-network-simulator-software. [Accessed 12 February 2019].

[301] K. Mehdi, M. Lounis, A. Bounceur and T. Kechadi, "CupCarbon: A Multi-Agent and Discrete Event Wireless Sensor Network Design and Simulation Tool," in Institute for Computer Science, Social Informatics and Telecommunications Engineering (ICST), 2014.

[302] C. Mallanda, A. Suri, V. Kunchakarra, S. Iyengar, R. Kannan and A. Durresi, "Simulating Wireless Sensor Networks with OMNeT++," IEEE, 2005.

[303] "NS3," 11 February 2019. [Online]. Available: https://www.nsnam.org/docs/tutorial/html/introduction.html. [Accessed 12 February 2019].

[304] "cupcarbon," 2017. [Online]. Available: http://www.cupcarbon.com/. [Accessed 12 February 2019].

[305] G. Keramidas, N. Voros and M. Hübner, Components and Services for IoT Platforms || Internet of Things Simulation Using OMNeT++ and Hardware in the Loop, Switzerland : Springer International Publishing, 2017.

[306] A. Abdelrahman, H. Mohammad and A. F. O. A. Fayez, "A Survey on Wireless Sensor Networks Simulation Tools and Testbeds," in Sensors, Transducers, Signal Conditioning and Wireless Sensors Networks Advances in Sensors, Barcelona, Spain, International Frequency Sensor Association (IFSA), 2016, pp. 283-302.

[307] S. Han, M. Lee, N. Crespi, V. Luong, K. Heo, M. Brut and P. Gatellier, "DPWSim: A Devices Profile for Web Services (DPWS) Simulator," IEEE Internet of Things Journal, vol. 2, no. 3, pp. 221 - 229, 2015.

[308] K. Darabkh and L. Al-Jdayeh, "AEA-FCP: An Adaptive Energy-aware Fixed Clustering Protocol for Data Dissemination in Wireless Sensor Networks," Personal and Ubiquitous Computing, vol. 23, no. 5, p. 819–837, 2019.

[309] R. Al-Zubi, N. Abedsalam, A. Atieh and K. Darabkh, "LBCH: Load Balancing Cluster Head Protocol for Wireless Sensor Networks," Informatica, vol. 29, no. 4, pp. 633-650, 2018.

[310] K. Darabkh, W. Al-Rawashdeh, M. Hawa and R. Saifan, "MT-CHR: A Modified Threshold-based Cluster Head Replacement Protocol for Wireless Sensor Networks," Computers & Electrical Engineering, vol. 72, pp. 926-938, 2018.

[311] K. Darabkh, N. Al-Maaitah, I. Jafar and A. Khalifeh, "EA-CRP: A Novel Energy-aware Clustering and Routing Protocol in Wireless Sensor Networks," Computers & Electrical Engineering, vol. 72, pp. 702-718, 2018.

[312] F. Petter and V. Ovidui, Internet of Things From Research and Innovation to Market deployment, Denmark: River Publishers, 2014.

[313] G. Gardašević, M. Veletić, N. Maletić, D. Vasiljević, I. Radusinović, S. Tomović and M. Radonjić, "The IoT Architectural Framework, Design Issues and Application Domains," Wireless Personal Communications, Springer-Verlag, vol. 92, no. 1, pp. 127-148, 2016.

[314] A. Bröring, J. Echterhoff, S. Jirka, I. Simonis, T. Everding, C. Stasch, S. Liang and R. Lemmens, "New Generation Sensor Web Enablement," Sensors, vol. 11, no. 12, pp. 2652-2699, 2011.

[315] K. Darabkh and J. Zomot, "An Improved Cluster Head Selection Algorithm for Wireless Sensor Networks," in 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, 2018.

[316] K. Darabkh and L. Al-Jdayeh, "A New Fixed Clustering Based Algorithm for Wireless Sensor Networks," in 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, 2018.

[317] M. Al-Mistarihi, I. Tanash, F. Yaseen and K. Darabkh, "Protecting Source Location Privacy in a Clustered Wireless Sensor Networks Against Local Eavesdroppers," Mobile Networks and Applications, p. 1–13, 2018.

[318] K. Darabkh, W. Al-Rawashdeh, R. Al-Zubi and S. Alnabelsi, "C-DTB-CHR: centralized density- and threshold-based cluster head replacement protocols for wireless sensor networks," The Journal of Supercomputing, vol. 73, no. 12, p. 5332–5353, 2017.

[319] K. Darabkh and N. Alsaraireh, "A Yet Efficient Target Tracking Algorithm in Wireless Sensor Networks," in 15th International Multi-Conference on Systems, Signals & Devices (SSD), Hammamet, Tunisia, 2018.

[320] K. Darabkh and R. Muqat, "An Efficient Protocol for Minimizing Long-distance Communications over Wireless Sensor Networks," in 15th International Multi-Conference on Systems, Signals & Devices (SSD), Hammamet, Tunisia, 2018.

[321] Y. Zou, H. Wan, X. Zhang, D. Ha and P. Wang, "Electronic Nose and Electronic Tongue," Beijing and Springer Science+Business Media Dordrecht, 2015.

[322] N. Lane, E. Miluzzo, H. Lu, D. Peebles, C. Tanzeem, C. Andrew and C. Dartmouth, "A survey of mobile phone sensing," IEEE Communications Magazine, vol. 48, no. 9, pp. 140-150, 2010.

[323] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, "Internet of Things for Smart Cities," IEEE Internet of Things Journal, Institute of Electrical and Electronics Engineers , vol. 1, no. 1, pp. 22-32, 2014.

[324] J. Lynch and L. Kenneth, "A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring," Shock and Vibration Digest, vol. 38, no. 2, p. 91–130, 2006.

[325] N. Maisonneuve, M. Stevens, M. Niessen, P. Hanappe and L. Steels, "Citizen Noise Pollution Monitoring," in The Proceedings of the 10th International Digital Government Research Conference, Puebla, Mexico, 2009.

[326] X. Li, W. Shu, M. Li, H.-Y. Huang, P.-E. Luo and M.-Y. Wu, "Performance Evaluation of Vehicle-Based Mobile Sensor Networks for Traffic Monitoring," IEEE Transactions on Vehicular Technology , vol. 58, no. 4, p. 1647–1653, 2009.

[327] "Smart Cities Are Built On The Internet Of Things," Lopez Research, 2014.

[328] S. Yu, J. Hsieh, Y. Chen and W. Hu, "An Automatic Traffic Surveillance System for Vehicle Tracking and Classification," in Scandinavian Conference on Image Analysis,Springer-Verlag, Berlin, 2003.

[329] W. Hu, X. Hu, J.-q. Deng, C. Zhu, G. Fotopoulos, E. Ngai and V. Leung, "Mood-fatigue analyzer: towards context-aware mobile sensing applications for safe driving," in M4IOT '14 Proceedings of the 1st ACM Workshop on Middleware for Context-Aware Applications in the IoT, Bordeaux, France, 2014.

[330] H. Singh, J. Bhatia and J. Kaur, "Eye tracking based driver fatigue monitoring and warning system," in India International Conference on Power Electronics 2010 (IICPE2010), New Delhi, India, 2011.

[331] A. Kotb, S. Yao-chun and H. Yi, "Smart Parking Guidance, Monitoring and Reservations: A Review," IEEE Intelligent Transportation Systems Magazine, vol. 9, no. 2, pp. 6 - 16, 2017.

[332] Q. Liu, H. Lu, B. Zou and Q. Li, "Design and Development of Parking Guidance Information System Based on Web and GIS Technology," in 2006 6th International Conference on ITS Telecommunications, Chengdu, China, 2006.

[333] L. Mimbela and P. Kent, Summary of vehicle detection and surveillance technologies used in intelligent transportation systems, The Vehicle Detector Clearinghouse, 2007.

[334] V. Tamilmaran and K. Dwarkadas, "Smart Grid: An Overview," Smart Grid and Renewable Energy, vol. 2, no. 4, pp. 305-311, 2011.

[335] W. Kastner, G. Neugschwandtner, S. Soucek and M. Newmann, "Communication systems for building automation and control," Proceedings of the IEEE , vol. 93, no. 6, p. 1178–1203, 2005.

[336] Final Technical Report, "Smart Water Systems," Oxford University, Oxford, 2011.

[337] E. Farah, A. Abdallah and I. Shahrour, "Sunrise: Large-scale demonstrator of the smart water system," International Journal of Sustainable Development and Planning, vol. 12, no. 1, p. 112–121, 2017.

[338] S. Akane and P. Rosalind, "Stress Recognition using Wearable Sensors and Mobile Phones," in Humaine Association Conference on Affective Computing and Intelligent Interaction, Geneva, Switzerland, 2013.

[339] C.-h. Tien and D. Can, "Environment Monitoring System for Agriculture Application Based on Wireless Sensor Network," in Seventh International Conference on Information Science and Technology (ICIST), Da Nang, Vietnam, 2017.

[340] A. Mussab, Z. A, B. Bahaa, M. Talal and L. Kiah, "A Review of Smart Home Applications based on Internet of Things," Journal of Network and Computer Applications, vol. 97, pp. 48-65, 2017.

[341] T. Hargreaves, C. Wilson and R. Hauxwell-Baldwin, "Learning to live in a smart home," Building Research and Information, vol. 46, no. 1, pp. 127-139, 2018.

[342] D. K. Mishra, M. K. Nayak and A. Joshi, "Internet of Things Applications at Urban Spaces (Tel Aviv Smart City: A Case Study)," in Information and Communication Technology for Sustainable Development, Singapore, Springer, 2018, pp. 1-11.

[343] L. Yan, M. Katherine and F. Simon, "Current Standards Landscape for Smart Manufacturing Systems Manufacturing Systems," National Institute of Standards and Technology , USA, 2016.

[344] L. Neweb, 10 September 2019. [Online]. Available: https://www.cnetfrance.fr/news/vie-privee-comment-bien-proteger-vos-donnees-personnelles-39886253.htm. [Accessed 30 January 2020].

[345] D. Tony, "Busniss Insider,Morgan Stanley: 75 Billion Devices Will Be Connected To The Internet Of Things By 2020," 2 October 2013. [Online]. Available: http://www.businessinsider.com/75-billion-devices-will-be-connected-to-the-internet-by-2020-2013-10. [Accessed 17 February 2019].

[346] I. Qusay, "Internet of things: a survey of challenges and issues," International Journal of Internet of Things and Cyber-Assurance , vol. 1, no. 1, pp. 40-75, 2018.

[347] S. Raja, D. Rajkumar and V. Raj, "Internet of Things: Challenges, Issues and Applications," Journal of Circuits, Systems and Computers, vol. 27, no. 12, pp. 1-16, 2018.

[348] M. Shetty and M. D, "Challenges, Issues and Applications of Internet of Things," in nternet of Things: Novel Advances and Envisioned Applications. Studies in Big Data, vol. 25, Cham, Springer International Publishing AG, 2017, pp. 231-243.

Please cite it as: Wafa'a Kassab and Khalid A. Darabkh, "A-Z Survey of Internet of Things: Architectures, Protocols, Applications, Recent Advances, Future Directions and Recommendations," Journal of Network and Computer Applications, Elsevier, vol. 163, p.102663, August 2020. DOI: https://doi.org/10.1016/j.jnca.2020.102663

Published via this link: https://www.sciencedirect.com/science/article/pii/S1084804520301375