Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)

Кафедра Інфокомунікаційної інженерії імені В.В. Поповського
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Моделі та методи безпечної маршрутизації в IoT
(Models and Methods of Secure Routing in IoT)
(тема)

Виконав:
студент 2 курсу, групи ТСМім-19-1
Егбе Годсон Агбара
(прізвище, ініціали)

Спеціальність: 172 Телекомунікації та радіотехніка
(код і повна назва спеціальності)

Тип програми: освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма: Телекомунікаційні системи та мережі
(повна назва освітньої програми)

Керівник: проф. каф. ІКІ імені В.В. Поповського
Єременко О.С.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ Лемешко О.В.
(підпис) (прізвище, ініціали)

2020 р.

Не містить відомостей, заборонених до відкритого опублікування

Студент _____ Егбе Годсон Агбара
(підпис) (ініціали, прізвище)

Керівник _____ О.С. Єременко
(підпис) (ініціали, прізвище)

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)
Кафедра Інфокомунікаційної інженерії імені В.В. Поповського
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 172 Телекомунікації та радіотехніка
(код і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Телекомунікаційні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ
Зав. кафедри_____
(підпис)
«____»_____2020 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Егбе Годсон Агбара
(прізвище, і'мя, по батькові)

1.Тема роботи : Моделі та методи безпечної маршрутизації в IoT (Models and Methods of Secure Routing in IoT)
затверджена наказом по університету від 27.10.2020 р. №1447 Ст
2.Термін подання студентом роботи до екзаменаційної комісії: __.__.2020 р.
3.Вихідні дані до роботи: провести аналіз сучасних тенденцій та застосувань телекомунікаційних мереж на базі технології Інтернету речей – IoT; зробити детальний огляд протоколів безпечної маршрутизації в IoT; засобів моделювання процесів безпечної маршрутизації в IoT.
4.Перелік питань, які потрібно опрацювати в роботі: 1. Аналіз IoT архітектур, проблемних питань і перспективних рішень. 2. Огляд протоколів безпечної маршрутизації в IoT. 3. Вибір математичної моделі безпечної маршрутизації в IoT і проведення чисельного дослідження на фрагменті безпроводової сенсорної мережі. 4. Використання середовища MATLAB для аналітичного моделювання безпечної маршрутизації та відповідних протоколів у мережах IoT. 5. Аналіз результатів аналітичного моделювання. 6. Пропозиції щодо стратегії безпечної маршрутизації в рамках обраних моделей.
5.Перелік графічного матеріалу із зазначенням креслень, плакатів, комп'ютерних ілюстрацій:

1. Титульний. 2. Опис проблеми. Об’єкт, предмет та мета дослідження. 3-6. Аналіз IoT архітектур. 7-10. Огляд протоколів безпечної маршрутизації в IoT. 11-13. Математичне моделювання безпечної маршрутизації в IoT. 14. Моделювання протоколів маршрутизації IoT у середовищі MATLAB. 15. Висновки. 16. Апробація результатів дослідження.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	професор Єременко Олександра Сергіївна		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Збір матеріалів для дослідження	03.11.2020 р.	Виконано
2	Розробка розділу 1	16.11.2020 р.	Виконано
3	Розробка розділу 2	23.11.2020 р.	Виконано
4	Розробка розділу 3	07.12.2020 р.	Виконано
5	Розробка розділу 4	14.12.2020 р.	Виконано
6	Оформлення роботи	21.12.2020 р.	Виконано

Дата видачі завдання 27 жовтня 2020 р.

Студент _____ Егбе Годсон Агбара
(підпис) (прізвище та ініціали)

Керівник роботи _____ професор Єременко О.С.
(підпис) (посада, прізвище, ініціали)

№ з/п	Позначення	Найменування	Додаткові Відомості
		Текстові документи	
1	ГЮІК.ХХХХХХ.001ПЗ	Пояснювальна записка	87 с.
		Графічні документи	
2	Слайд 1	Титульний	Л.1, ф.А4
3	Слайд 2	Опис проблеми. Об’єкт, предмет та мета дослідження	Л.1, ф.А4
4-7	Слайд 3-6	Аналіз IoT архітектур	Л.1, ф.А4
8-11	Слайд 7-10	Огляд протоколів безпечної маршрутизації в IoT	Л.1, ф.А4
12-14	Слайд 11-13	Математичне моделювання безпечної маршрутизації в IoT	Л.1, ф.А4
15	Слайд 14	Моделювання протоколів маршрутизації IoT у середовищі MATLAB	Л.1, ф.А4
16	Слайд 15	Висновки	Л.1, ф.А4
17	Слайд 16	Апробація результатів дослідження	Л.1, ф.А4

					ГЮИК.ХХХХХХ.002.Д1	
Изм.	Лист		Підп.	Дата		
Розроб		Егбе Годсон Агбара			Моделі та методи безпечної маршрутизації в IoT	
Перев.		Єременко О.С.				ХНУРЕ
Норм.		Єременко О.С.				Кафедра ІКІ ім. В.В. Поповського
Затв.		Лемешко О.В.				
					Відомість атестаційної роботи магістра	

РЕФЕРАТ

Пояснювальна записка: 87 с., 21 рис., 5 табл., 35 джерел.

БЕЗПЕЧНА МАРШРУТИЗАЦІЯ, IoT, ПРОТОКОЛИ, ЗАСНОВАНІ НА ДОВІРІ, МОДЕЛЮВАННЯ

Об’єкт дослідження – процес безпечної маршрутизації у мережах IoT.

Предмет дослідження – моделі, методи та протоколи безпечної маршрутизації у мережах IoT.

Мета роботи – аналіз моделей, методів і протоколів безпечної маршрутизації в IoT з метою підвищення мережної безпеки.

Методи досліджень – аналітичне моделювання, симуляція, формалізація та порівняння.

Завдяки стрімкому зростанню технологій та важливості даних, ми прямуємо у світ, де все і всі будуть пов’язані, а IoT стане основною технологією, яка робить це можливим. Тоді як ядром функціональних можливостей IoT є маршрутизація та засоби обміну інформацією, передавання пакетів та маршрутних даних між малопотужними сенсорними пристроями або вузлами, що самоорганізуються.

У роботі було проведено аналіз архітектур IoT, завдань, пов’язаних з маршрутизацією, та проблем у IoT, а також поточного стану протоколів безпечної маршрутизації в IoT. Особлива увага приділяється моделі безпечної маршрутизації в IoT та її дослідженню. Також було представлено моделювання протоколів маршрутизації IoT у середовищі MATLAB.

Було представлено математичну модель безпечної маршрутизації в IoT та чисельне дослідження на фрагменті безпроводової сенсорної мережі.

ABSTRACT

This thesis contains 87 pages, 21 figures, 5 tables, and 35 sources or references.

SECURE ROUTING, IoT, TRUST BASED PROTOCOLS, MODELING

The object of research is a process of secure routing in the IoT networks.

The subject of research is the models, methods, and protocols of secure routing in IoT networks.

The purpose of the work is the analysis of models, methods, and protocols of secure routing in IoT in order to increase network security.

Research methods are analytical modeling, simulation, formalization, and comparison.

With the rapid growth in technology and the importance of data, we are heading into a world where everything and everyone will be connected and IoT will be the underlying technology that makes this possible. While the core to the functionality of IoT is its routing and the way low-powered sensory devices or nodes self-organize and share information either data packets information or routes in between themselves.

The analysis of IoT architectures, routing-related issues, and challenges in IoT, as well as the current state of secure routing protocols in IoT, was carried out in the work. Particular attention is paid to the model of secure routing in IoT and its investigation. The modeling IoT routing protocols with the MATLAB environment was also presented.

A mathematical model of secure routing in IoT and a numerical study on a fragment of a wireless sensor network were presented.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS, SYMBOLS, UNITS AND TERMS

API – Application Programming Interface

AODV – Ad Hoc on Demand Distance Vector Routing

APs – Access Points

ANSGA – An Adapted non dominated sorting Based Generic Algorithm

BC- Border Controller

CORE- Constrained Restful Environments

CoAP – Constrained Application Protocol

CPU- Central Processor Unit

CH – Cluster Head

CBLR – Cluster-Based Location Routing

CBR– Cluster-Based Routing

CDMA- Code-division Multiple Access

DSDN- Distributed Software Defined Networks

DTLS- Distributed Transport Layer security

DNS – Domain Name System

DSRC – Dedicated Short-Range Communications

DAGS- Directed Acyclic Graphs

DOS- Denial Of Service

ERP- Enterprise Resource Planning

EPC- Electronic Product Code

FC – Fog Computing

GPRS- General Packets Radio Service

HTTP- Hyper Text Transfer protocol

IoT – Internet of Things

IEEE 802.11 – Institute of Electrical and Electronics Engineers 802.11

IBFO - Improved Bacterial Foraging Optimization

LEACH- Low Energy Adaptive Clustering Hierarchy

LoWPAN - Low-power Wireless Personal Area Network

LLN- Low Power Lossy Networks

LBR- LLN Border Router

LTE – Long Term Evolution

LPWA- Low Power Wide Area

MANETs – Mobile Ad-hoc Networks

MAC- Media Access Control

M2M – machine To Machine

MILP- Mixed Integer Linear Programming

NIC – Network Interface Card

NBBTE- Node Behavioral Strategies Branding Belief Theory of Thrust Evaluation

OSAP - Optimal Secured Energy Aware Protocol

OWASP – Open Web Application Security Project

OSLR - Optimized Link State Routing Protocol

ONS – Object Naming Service

QoS – Quality of Service

RFID – Radio-frequency Identification

RPL- Routing Protocol For Low-power and Lossy networks

REST API- Representational State Transfer Application Transfer protocol

SDN – Software-Defined Networking

SEAP- Secure Energy Aware Routing Protocol

SSL – Secure Socket Layer

TERP- Trust and Energy Aware Routing Protocol

TLS – Transport Layer Security

TDMA- Time Division Multiple Access

WAN – Wide Area Network

WSN- Wireless Sensor Networks

WIFI – Wireless Fidelity

WLAN – Wireless Local Area Network

WPAN- Wireless Personal Area Network

5G NR – Standard New Radio

6TISCH – IPV6 Time Slotted Channel Hopping

6LOWPAN- IPV6 Based low Power Personal Area Network

INTRODUCTION

With the rapid growth in technology and the importance of data we are heading into a world where everything and everyone will be connected and IoT will be the underlying technology that makes this possible. Its main idea is to have independent and self-sufficient connections that allows exchange of data among real world physical devices and real applications.

The fundamental characteristics of IoT include but are not limited to the following:

1. Interconnectivity: where anyone or anything can be connected.
2. Heterogeneity: that is has to be platform and hardware independent.
3. Dynamic state: where it can be operating in an active or a passive state.
4. Enormous scale: that it can accommodate lots of devices and data.

It is worth mentioning that IoT is not a single technology but a combination of different software and hardware devices and together these two categories ensure the effective operational and functional capability of IoT. Some examples of these devices that makes IoT what it is include sensors, microprocessors, microcontrollers, GPS, Wi-Fi, GSM, RFID, GPRS, 2G/3G/4G, etc.

An IoT architecture consist of different layers of technologies and this different layers work hand in hand to ensure efficient and effective communication and data processing. These layers are classified into smart device/sensor layer, gateway/network layer, management service layer and application layer.

Future technological developments for IoT are mainly concerned with improvements in the field of science in areas such as semiconductor, smart phones/devices, cloud computing and networking, network virtualization and sensors etc. It is safe to state that IoT will be tremendously affected by any or all of the aforementioned improvements and these improvements will likely result in better sensing, data transfer, data processing and runtime, as well as connectivity.

While to the future of a functional human will depend on IoT it is also important to acknowledge its drawbacks and some of these concern and have to deal with issues regarding privacy and security, interoperability, data management and different device energy levels.

IoT technology has become increasingly popular in the last 5 years and with advances in computing communication and routing it will be what will shape the future of humans and interaction with technological environment that will be found in all facets of human life ranging from agriculture, healthcare, manufacturing, transportation and even insurance.

Internet of Things (IoT) technology is the next technological leap that will introduce a significant improvement to people by connecting devices, people and networks together and no facet of the human environment. IoT devices are by nature highly connected hence it provides broad attack platform for hackers to exploit, consequently it needs a robust security model to support resource constrained IoT devices and end-to-end security. The attack vectors and security requirements for routing in the IoT system and organizational approach towards security were investigated in the work.

CHAPTER 1

ANALYSIS OF IOT ARCHITECTURES: CHALLENGES AND PERSPECTIVE SOLUTIONS

1.1. Architectures of the IoT

The past few years have been a windfall towards the validation and massive growth of the IoT tech sector and it's about time we took a look at some architectural themes that have risen to the top [1-4]. In support of healthy industry growth, there have been parallel efforts towards the standardization of particular protocols, security practices and even system architectures. However, considering the multitude of bits and pieces that make up the IoT stack, it is a real challenge to know where to start and how to "future-proof" your particular system architecture [5-9]. And if you are looking for one take away just remember that there is no single unified IoT architecture that is agreed on.

There are essentially three major types of IoT architectural contexts: application specific, open platform and Network as a Service (NaaS) [5-11]. This section summarizes the leading trends in end-to-end, open platform IoT architectures where scalability and interoperability are major driving factors.

1.1.1. IoT Architecture Basics

So what are we looking for in an "end-to-end" or complete IoT architecture anyway? Here are some important requirements [9]:

- Concurrent Data Collection – support for collection, analysis and control from a large number of sensors or actuators
- Efficient Data Handling – minimize raw data and maximize actionable information

- Connectivity and Communications – provide network connectivity and flexible, robust protocols support between sensors/actuators and the cloud
- Scalable – scale individual elements in the system using the same architecture
- Security – end to end encryption and monitoring
- Availability and Quality of Service – minimal latencies and fault tolerant
- Modular, Flexible and Platform-independent – each layer should allow for features, hardware or cloud infrastructure to be sourced from different suppliers
- Open Standards and Interoperable – communication between the layers should be based on open standards to ensure interoperability
- Device Management – enable automated/remote device management and updates
- Defined APIs – each layer should have defined APIs that allow for easy integration with existing applications and integration with other IoT solutions

1.1.2. Common IoT Architectures

While we can't cover all of the possibilities and permutations, the following group of architectures should give you a greater understanding of the core design considerations and typical primary functional layers in an end-to-end IoT stack.

1.1.2.1. Three Layer (Tier) IoT Architecture

While there are myriad bits that build a complete end-to-end IoT architecture, this architecture simplifies it down to three fundamental building blocks [9, 11]:

1. Perception layer – Sensors, actuators and edge devices that interact with the environment
2. Network Layer – Discovers, connects and translates devices over a network and in coordination with the application layer

3. Application Layer – Data processing and storage with specialized services and functionality for users

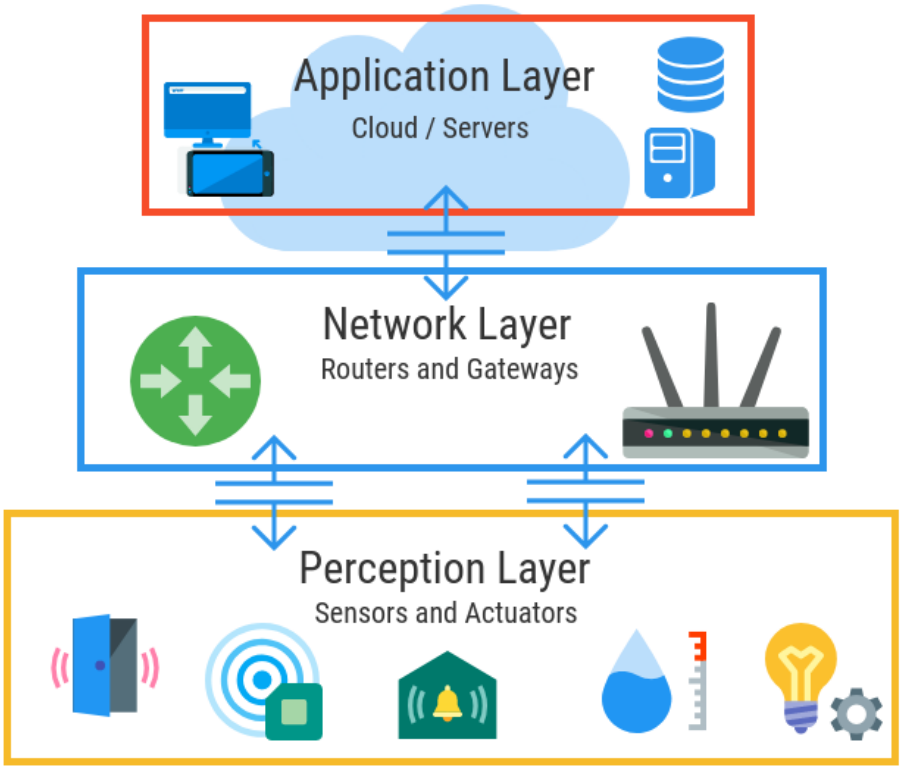


Fig. 1.1. The fundamental Three Layer IoT Architecture [9]

Devices make up a physical or perceptual IoT layer and typically include sensors, actuators and other smart devices. One might call these the “Things” in the Internet of Things. Devices, in turn, interface and communicate to the cloud via wire or localized Radio Frequency (RF) networks. This is typically done through gateways. Often times IoT devices are said to be at the “edge” of the IoT network and are referred to as “edge nodes”. When selecting a device, it is important to consider requirements for specific I/O protocols and potential latency, wired or RF interfaces, power, ruggedness and the device’s overall sensitivity. It is critical to determine how much device flexibility your architecture should have.

Many newer devices are IoT ready right out of the box (e.g. are sold with low power Bluetooth or are Ethernet enabled). However, most sensors, actuators and legacy devices

still interface via conventional “pre-IoT” methods such as analog or serial connections. It is common practice to connect one or more of these conventional devices to microcontrollers, systems on modules (SOMs) or single-board computers (SBCs) with the necessary peripherals (e.g. Arduino, NetBurner, or Raspberry Pi). At a minimum, such collectors provide network connectivity between the edge nodes and a master gateway. In some instances, they may be capable of being configured as a gateway as well.

IoT Gateways are are an important middleman element that serves as the messenger and translator between the cloud and clusters of smart devices. They are physical devices or software programs that typically run from the field in close proximity to the edge sensors and other devices. Large IoT systems might use a multitude of gateways to serve high volumes of edge nodes. They can provide a range of functionality, but most importantly they normalize, connect and transfer data between the physical device layer and the cloud. In fact, all data moving between the cloud and the physical device layer goes through a gateway. IoT gateways are sometimes called “intelligent gateways” or “control tiers” [9, 11].

Today, gateways also support additional computing and peripheral functionality such as telemetry, multiple protocol translation, artificial intelligence, pre-processing and filtering massive raw sensor data sets, provisioning and device management. It is becoming common practice to implement data encryption and security monitoring on the intelligent gateway so as to prevent malicious man-in-the-middle attacks against otherwise vulnerable IoT systems. NetBurner devices can be used as robust IoT Gateways, as well as IoT Device Collectors, as mentioned above.

Certain gateways offer an operating system that is specialized for use in embedded and IoT systems along with optimized low-level support for different hardware interfaces, such as NetBurner’s SOMs with our custom Real Time Operating System (RTOS) and interface libraries. Managing memory, I/O, timing and interface is not a trivial task. According to Google Cloud, “Generally these abstractions are not easy to use directly, and frequently the OS does not provide abstractions for the wide range of sensor and

actuator modules you might encounter in building IoT solutions."[5] Libraries are typically available based on standard protocols. Oftentimes, the most optimized libraries will be part of commercially available development kits and SDKs (as is the case with NetBurner for a multitude of protocols and hardware types).

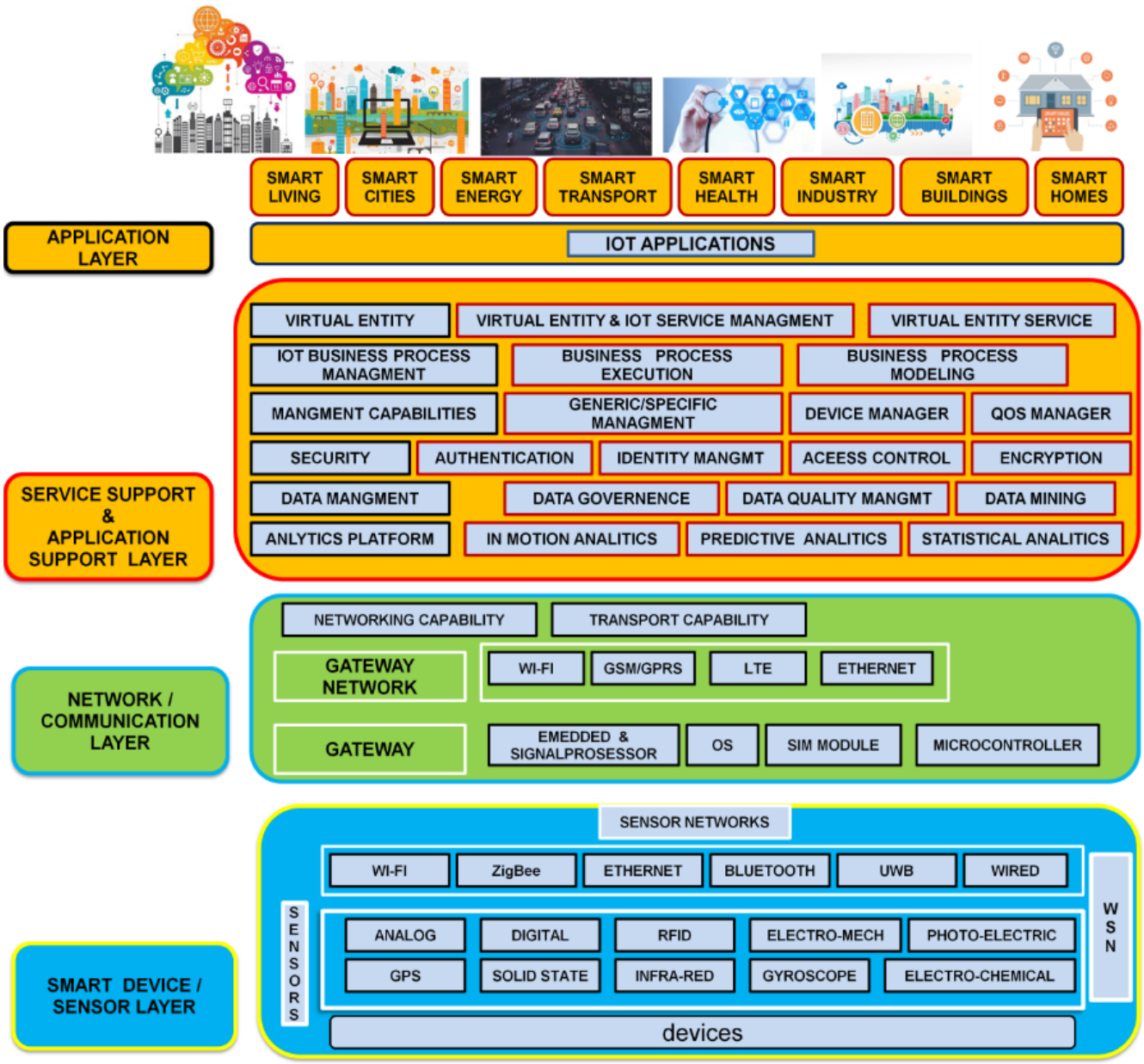


Fig. 1.2. Example of the IoT Architecture [5]

The Cloud is the application layer. It communicates with the gateway, typically over wired or cellular internet. The “Cloud” might be anything from services like AWS or Google Cloud, server farms, or even a company’s on-premises remote server. It provides powerful servers and databases that enable robust IoT applications and integrate services such as data storage, big data processing, filtering, analytics, 3rd party APIs, business logic, alerts, monitoring and user interfaces. In a Three Layer IoT Architecture, the “Cloud” is also used to control, configure, and trigger events at the gateway, and ultimately the edge devices.

1.1.2.2. Five Layer (Tier) IoT Architecture

The Five Layer IoT Architecture essentially builds upon the three layer approach (see figure above). This is still primarily a cloud-centric IoT architecture, where almost all of the IoT data processing is done on the cloud or a remote server. The difference between the Five Layer IoT Architecture and the Three Layer IoT Architecture are the addition of the following: Business Layer and Processing Layer

Table 1.1. Five Layer IoT Architecture [9]

Business Layer	Manages the entire IoT system, its functionality, applications, and business models.
Applications Layer	Provides application specific services to users.
Processing Layer	Analyses, stores, and processes large data sets. Might use databases, cloud computing, big data processing resources.
Transport Layer	Convert and transfers sensor data between layers and through networks such as 3G, LAN, Bluetooth, LoRaWAN etc. A typical IoT gateway.
Perception or physical layer	Sensors gather data from the environment; actuators turn things on or off, or set values.

The top-level Business Layer highlights the various management, business logic, and top-level requirements that need to be coordinated for a sustainable and successful architecture that is able to provide consistent value to the business and end users. It also reinforces the idea that IoT applications may be just a part of an organization's portfolio of interconnected technology and business areas.

The use of a Processing Layer unveils an important element of many IoT systems which need to incorporate numerous layers of processing in their architecture; oftentimes to filter down massive data sets and thus conserve resources. Using the processing layer at more than one point within in a specific architecture may be required for particular systems.

1.1.2.3. Fog Computing IoT Architecture

Fog computing is a newer convention that moves certain IoT services, like monitoring and pre-processing, closer to the edge to enable faster local decision making and automation. The Fog Layer resides between the Physical Layer (sensors and devices) and Transportation Layer (gateway) in what might be called the local network for that IoT cluster. In Fog architectures, computational and storage resources are typically provided by what is called a "Smart IoT Gateway" or "Fog Node", which can also be laterally networked to other fog nodes.

According to NIST's 2018 Fog Computing Definition Draft, "Fog nodes may be either physical or virtual elements and are tightly coupled with the smart end-devices or access networks. Fog nodes typically provide some form of data management and communication service between the peripheral layer where smart end-devices reside and the Cloud. Fog nodes, especially virtual ones, also referred as cloudlets, can be federated to provide horizontal expansion of the functionality over disperse geolocations." [6]

Fog architectures can have many benefits. By pre-processing sensor data, they can reduce bandwidth requirements between the gateway and the cloud while reducing the

resource consumption on the cloud. They also can lead to significantly improved real-time performance. Using the Fog Architecture makes a lot of sense for use cases where the above reasons hold value. It is sort of like refining a raw material closer to the source and only transporting the refined product to market. Fog nodes may even be configured to communicate directly with other fog nodes to create a mesh that can bypass the cloud completely.

Table 1.2. Fog Computing Architecture [9]

Business Layer		Manages the entire IoT system, it's functionality, applications, and business models.
Applications Layer		Provides application specific services to users
Processing Layer		Analyses, stores, and processes large data sets. Might use databases, cloud computing, big data processing resources
Transport Layer		Transfers sensor data between layers and through networks such as 3G, LAN, Bluetooth, LoRaWAN etc. A typical IoT typical gateway.
Fog Layer – Smart IoT Gateway	Security Layer	Encrypt / decrypt data.
	Storage Layer	Store files or data with localized relevance.
	Pre-Processing	Filter, processes, analyze and reduce edge data or process commands or subscriptions from the cloud.
	Monitoring	Monitor power, resources, responses, and services, access.
Perception or Physical Layer		Sensors gather data from the environment; actuators turn things on or off, or set values.

Fog computing architectures attempt to address requirements surrounding real-time performance, security and efficiency. The figure below illustrates the added layers entailed with the Fog Architecture [9]. You can see how four new layers sit on the margin between the Physical and Transport Layers and the value that these new layers can provide. That said, this also increases the complexity of the architecture, and the new layers introduce additional steps and data conversions which can lead to more points of failure.

1.1.2.4. Edge Computing Architecture

Edge computing is closely related to fog computing, where the goal is to keep certain processing capabilities and functionality closer to the edge nodes. It can be particularly useful in reducing sense and respond latencies for applications that require real-time performance. In edge computing, processing takes place at the physical perception layer directly on a smart device or on an IoT device collector as discussed earlier in this article. It can work independently or in any combination with fog and cloud computing [9].

If the edge node's processing unit were powerful enough, we could even extend layers for transport, security, storage, pre-processing, and monitoring, thereby reducing the work and functionality required between it and the cloud. According to info.opto22.com, Edge computing saves time and money by streamlining IoT communication, reducing system and network architecture complexity, and decreasing the number of potential failure points in an IoT application. Conversely, there are many reasons why having too much computing at the edge can be inefficient and overkill.

In this modality, edge computing provides similar benefits as fog computing but with an even greater capability to reduce localized latencies. Edge computing potentially allows for decentralization of computing, even greater data privacy, and allows for mesh networking off of the cloud. We go into more detail on edge computing in our piece called, "Computing from the Edge." Whether or not the benefits of this architecture outweigh its

potential cons will depend on the specific requirements of the system and applications in question.

1.1.2.5. Hybrid Cloud-Fog-Edge Architecture

As hinted to earlier, fog and edge architectures can be hybridized with cloud-centric IoT architectures if deemed to be a good fit for a project's requirements and business objectives. The below diagram portrays one combination which uses a nested configuration.

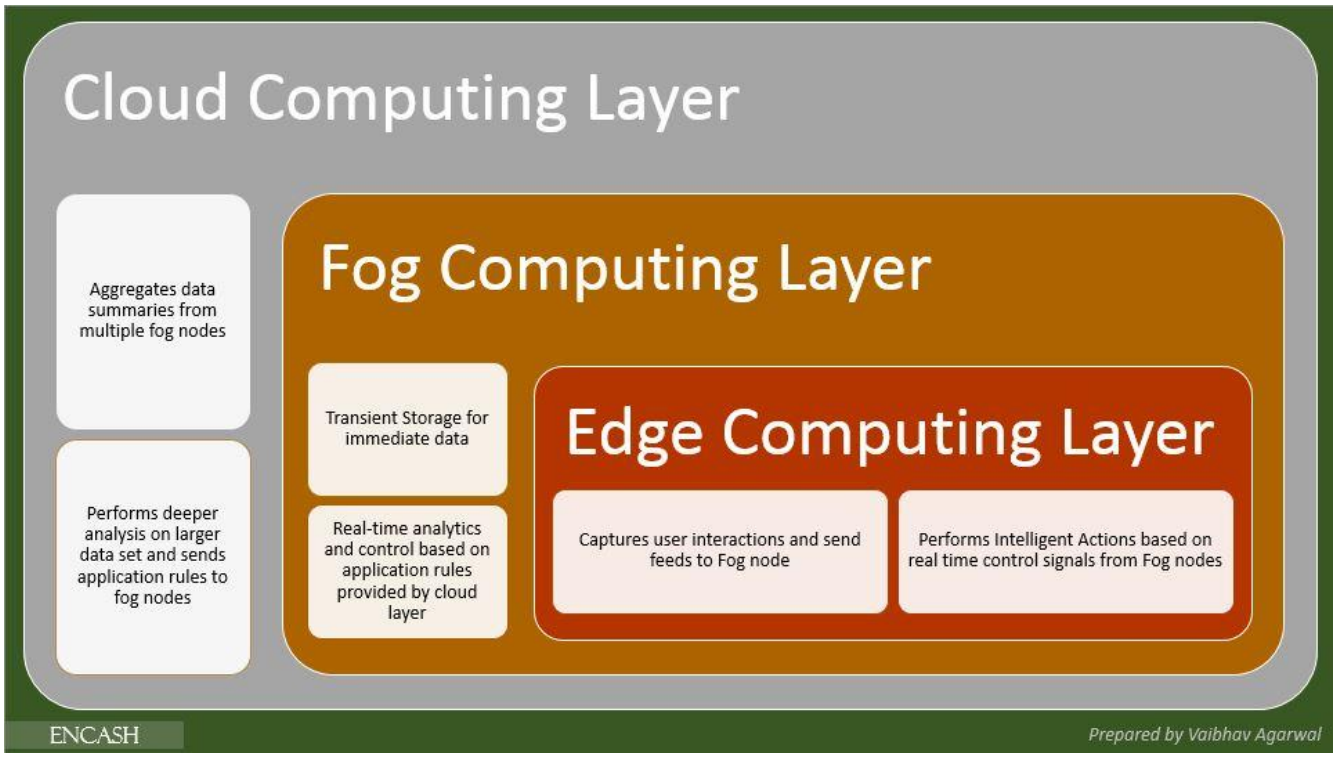


Fig. 1.3. Example of hybridizing IoT Architectures [9]

1.1.3. The hierarchy of the Fog and the Cloud

The technologies of Edge, Fog and Cloud are somewhat similar in the sense that they all act as servers. It is important to note that the heterogeneity of IoT devices also means a heterogeneity of edge and fog computing resources. While cloud resources are

expected to be homogenous, it is fair to expect that in many cases both edge and fog resources will use different operating systems, have different CPU and data storage capabilities, and have different energy consumption profiles. but in relation to IOT, all three technologies complement each other and in many cases really depend on the co-operation in between different layers for them to be functional. Edge and fog thus require an abstraction layer that allows applications to communicate with one another. The abstraction layer exposes a common set of APIs for monitoring, provisioning, and controlling the physical resources in a standardized way. The edge and fog computing layers simply act as a first line of defense for filtering, analyzing, and otherwise managing data endpoints. This saves the cloud from being queried by each and every node for each event. This model suggests a hierarchical organization of network, compute, and data storage resources. At each stage, data is collected, analyzed, and responded to when necessary, according to the capabilities of the resources at each layer. As data needs to be sent to the cloud, the latency becomes higher. The advantage of this hierarchy is that a response to events from resources close to the end device is fast and can result in immediate benefits, while still having deeper compute resources available in the cloud when necessary.

From an architectural standpoint, fog nodes closest to the network edge receive the data from IoT devices. The fog IoT application then directs different types of data to the optimal place for analysis:

- The most time-sensitive data is analyzed on the edge or fog node closest to the things generating the data.
- Data that can wait seconds or minutes for action is passed along to an aggregation node for analysis and action.
- Data that is less time sensitive is sent to the cloud for historical analysis, big data analytics, and long-term storage. For example, each of thousands or hundreds of

thousands of fog nodes might send periodic summaries of data to the cloud for historical analysis and storage.

The figure below shows the relationships existing between the three different technologies

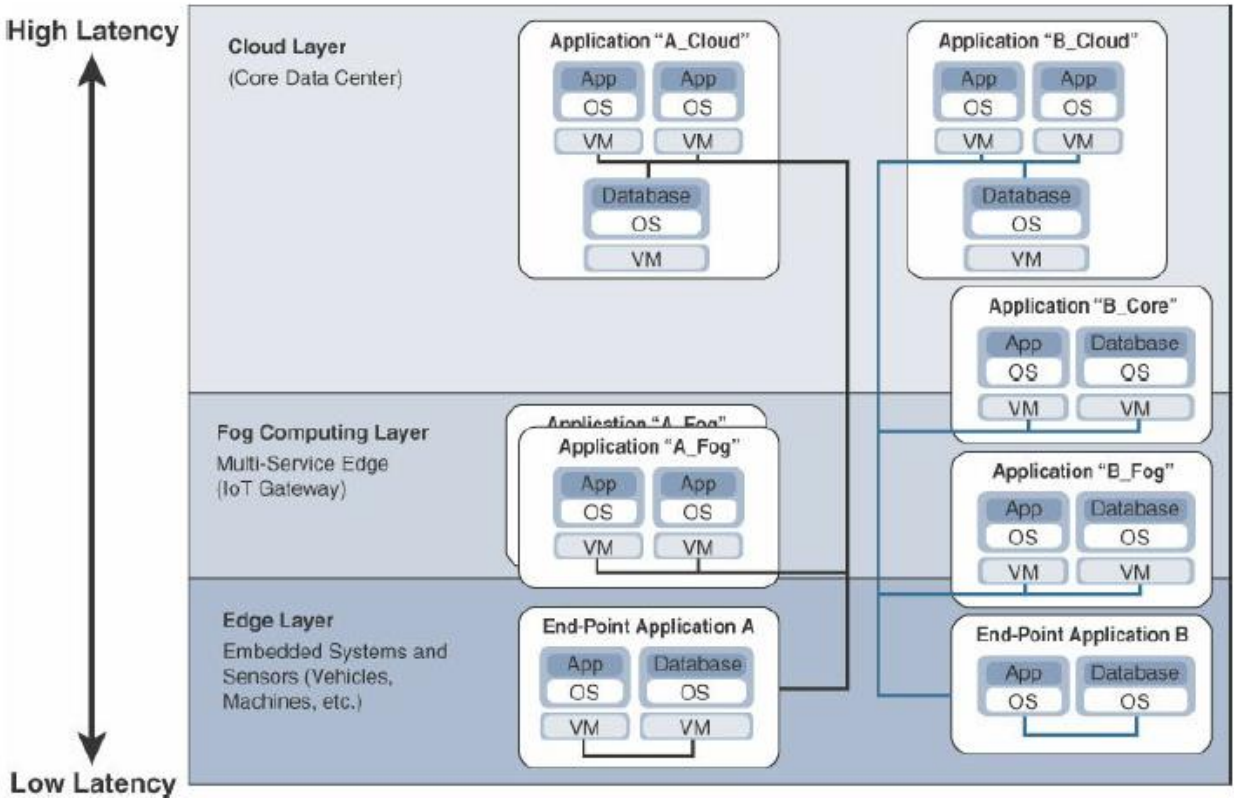


Fig 1.4. Distributed Computing and Data Management Across an IoT System [11]

1.2. IoT Reference Model

1.2.1. IoT Reference Model and Sub-models

Today, there is no standard way of understanding or describing models for the IoT, as a result, it is difficult to differentiate between IoT devices and systems and non-IoT devices and systems. The fact is, not every network is an IoT network, nor does it need to be, and not every application is an IoT application. Network, compute, application, and data management architectures that are IoT-ready require a different communication and

processing mode at different times and dependent on the kind of IOT service it wants to render. Hence, the IoT Reference Model aims at establishing a common ground and a common language for IoT architectures and IoT systems. They are several models used in IOT reference such as: The Domain model, the information model, the functional model, the communication model and lastly the security model. However, it is worth noting that the Domain model is the primary model from which all other model base their concepts on. The figure below illustrates the interaction between the different models listed above.

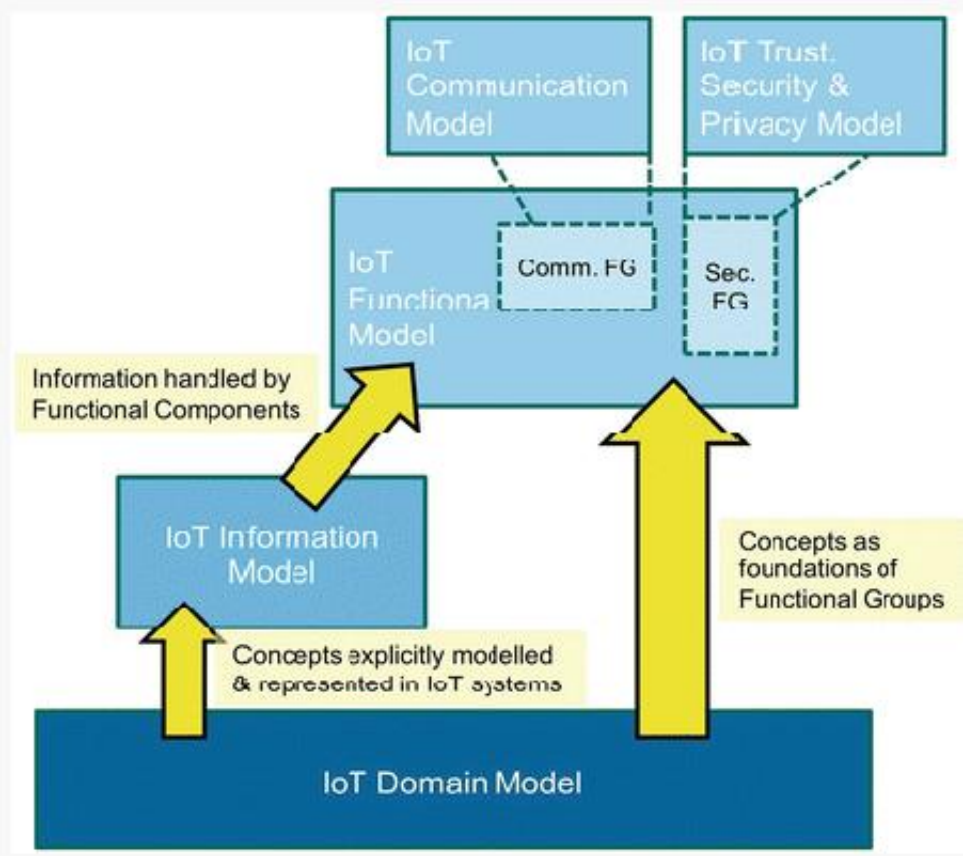


Fig. 1.5. Interaction of all sub-models in the IoT Reference Model [11]

The IoT Reference Model does not restrict the scope or locality of its components. For example, from a physical perspective, every element could reside in a single rack of equipment or it could be distributed across the world, it also allows the processing occurring at each level to range from trivial to complex, depending on the situation, it

describes how tasks at each level should be handled to maintain simplicity, allow high scalability, and ensure supportability and lastly, the model defines the functions required for an IoT system to be complete.

While various IoT reference models exist, the one put forth by the IoT World Forum offers a clean, simplified perspective on IoT and includes edge computing, data storage, and access. It provides a concise way of visualizing IoT from a technical perspective. Each of the seven layers is broken down into specific functions, and security encompasses the entire model.

The figure below shows the reference model set up by the IoTWF body.

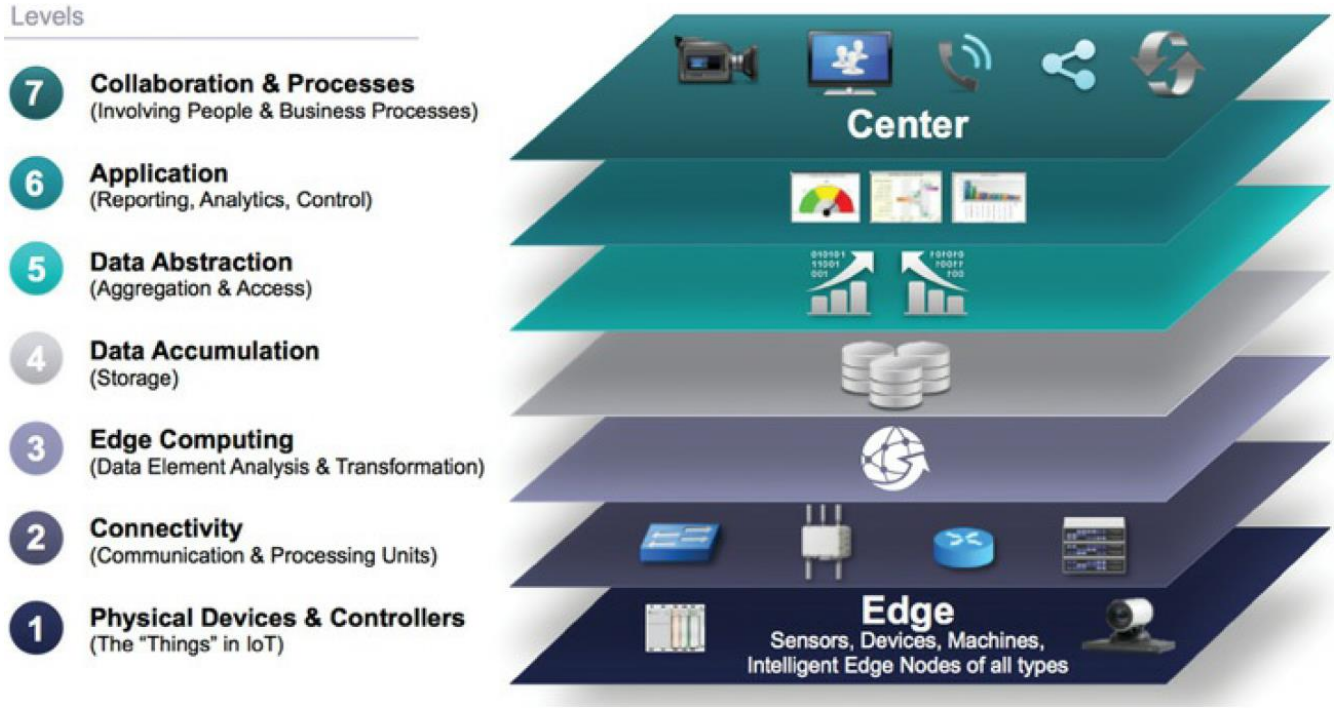


Fig. 1.6. IoT Reference Model Published by the IoT World Forum [11]

As shown in figure 1.6 above, the IoT Reference Model defines a set of levels with control flowing from the center (this could be either a cloud service or a dedicated data center), to the edge, which includes sensors, devices, machines, and other types of intelligent end nodes. In general, data travels up the stack, originating from the edge, and

goes northbound to the center. Using this reference model, we are able to achieve the following:

- Decompose the IoT problem into smaller parts.
- Identify different technologies at each layer and how they relate to one another.
- Define a system in which different parts can be provided by different vendors.
- Have a process of defining interfaces that leads to interoperability.
- Define a tiered security model that is enforced at the transition points between levels.

Communications and connectivity are concentrated in one level which is Level 2. The most important function of Level 2 is reliable, timely information transmission. This includes transmissions:

- Between devices (Level 1) and the network
- Across networks (east-west)
- Between the network (Level 2) and low-level information processing occurring at Level 3.

The IoT Reference Model does not require or indicate creation of a different network – it relies on existing networks. However, some legacy devices aren't IP-enabled, which will require introducing communication gateways. Other devices will require proprietary controllers to serve the communication function. However, over time, standardization will increase. As Level 1 devices proliferate, the ways in which they interact with Level 2 connectivity equipment may change. Regardless of the details, Level 1 devices communicate through the IoT system by interacting with Level 2 connectivity equipment, as shown in Figure 1.7 below.

Another level/layer worth mentioning as regards our topic is the layer three which is needed to convert network data flows into information that is suitable for storage and higher level processing at Level 4. The Figure 1.8 below shows the level three.

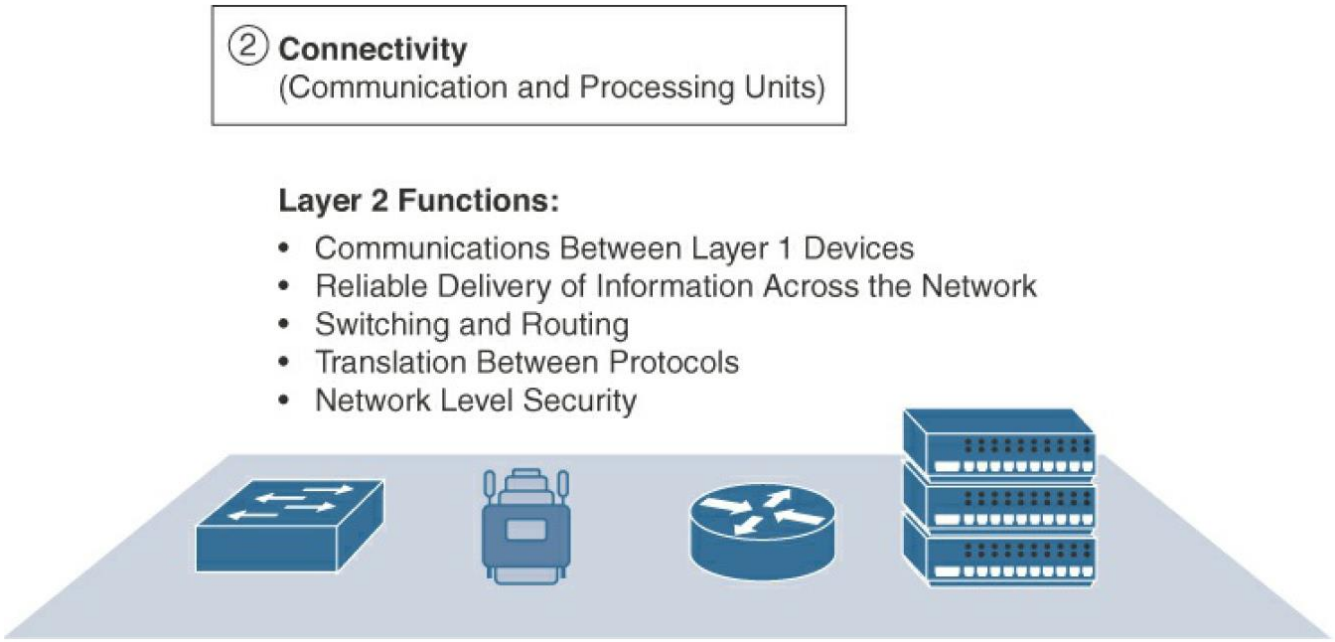


Fig. 1.7. IoT Reference Model Connectivity Layer Functions [11]

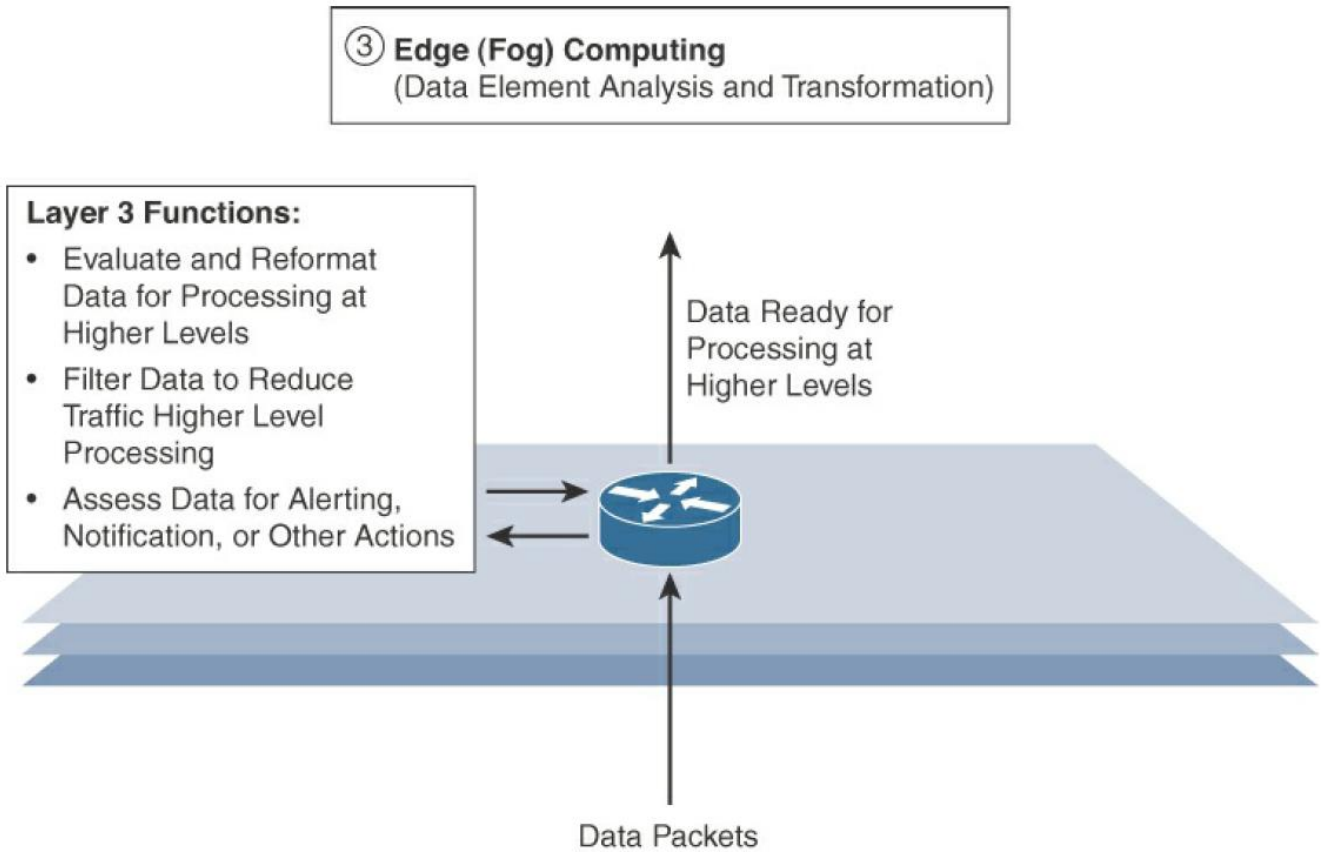


Fig. 1.8. IoT Reference Model Layer 3 Functions [11]

Level 3 processing is performed on a packet-by-packet basis. This processing is limited, because there is only awareness of data units and not sessions or transactions. Level 3 processing can encompass many examples, such as:

- Evaluation: Evaluating data for criteria as to whether it should be processed at a higher level.
- Formatting: Reformatting data for consistent higher-level processing
- Expanding/decoding: Handling cryptic data with additional context (such as the origin)
- Distillation/reduction: Reducing and/or summarizing data to minimize the impact of data and traffic on the network and higher-level processing systems
- Assessment: Determining whether data represents a threshold or alert; this could include redirecting data to additional destinations.

1.2.2. A Simplified IoT Architecture

In as much as they are differences which exists between all the aforementioned reference models, one common factor found in all them is that they all have a layered approach to IOT and they all recognize the interconnection of the IoT endpoint devices to a network that transports the data where it is then ultimately used by applications, whether at the data center, in the cloud, or at various management points throughout the stack. Lastly, all the other frameworks of the reference model are all derived from the Domain reference model.

This work highlights the fundamental building blocks that are common to most IoT systems and which is intended to help you in designing an IoT network. This framework is presented as two parallel stacks: The IoT Data Management and Compute Stack and the Core IoT Functional Stack. Reducing the framework down to a pair of three-layer stacks in no way suggests that the model lacks the detail necessary to develop a

sophisticated IoT strategy. Rather, the intention is to simplify the IoT architecture into its most basic building blocks and then to use it as a foundation to understand key design and deployment principles that are applied to industry-specific use cases.

The Figure 1.9 below shows the simplified IoT architecture.

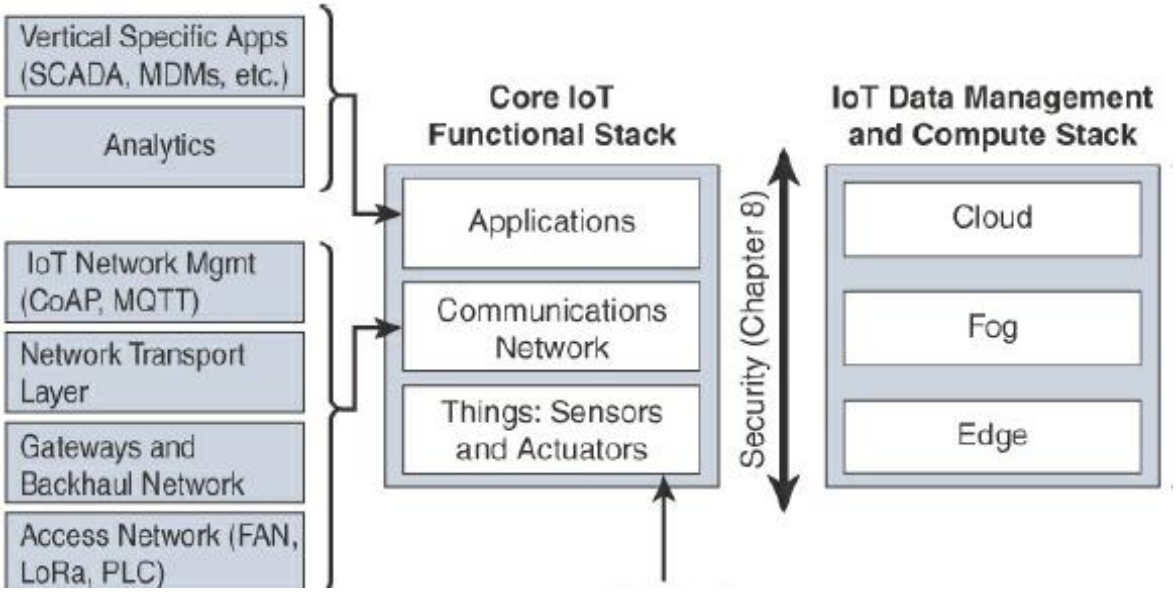


Fig. 1.9. Expanded view of the simplified IOT architecture [11]

1.2.3. The Core IoT Functional Stack

IoT networks are designed in such a way in which nodes or objects are independent and intelligent in terms of functionality whereby passive elements in the network are somewhat no existent. The nodes/objects are smart because they use a combination of contextual information and configured goals to perform actions. These actions can be self-contained (that is, the smart object does not rely on external systems for its actions); however, in most cases, the "thing" interacts with an external system to report information that the smart object collects, to exchange with other objects, or to interact with a management platform. In this case, the management platform can be used to process data

collected from the smart object and also guide the behavior of the smart object. several components have to work together for an IoT network to be operational, there are:

- Things layer;
- Communications network layer;
- Access network sublayer;
- Gateways and backhaul network sub layer;
- Network and transport sub layer;
- IOT network management sublayer;
- Application and Analytics layer.

Most IoT networks start from the object, or things that needs to be connected. From an architectural and network view point, the variety of smart object types, shapes, and needs drive the variety of IoT protocols and architectures also from network point of view consideration will be given to the throughputs and mobility requirements. The figures below show both the architectural and network considerations.

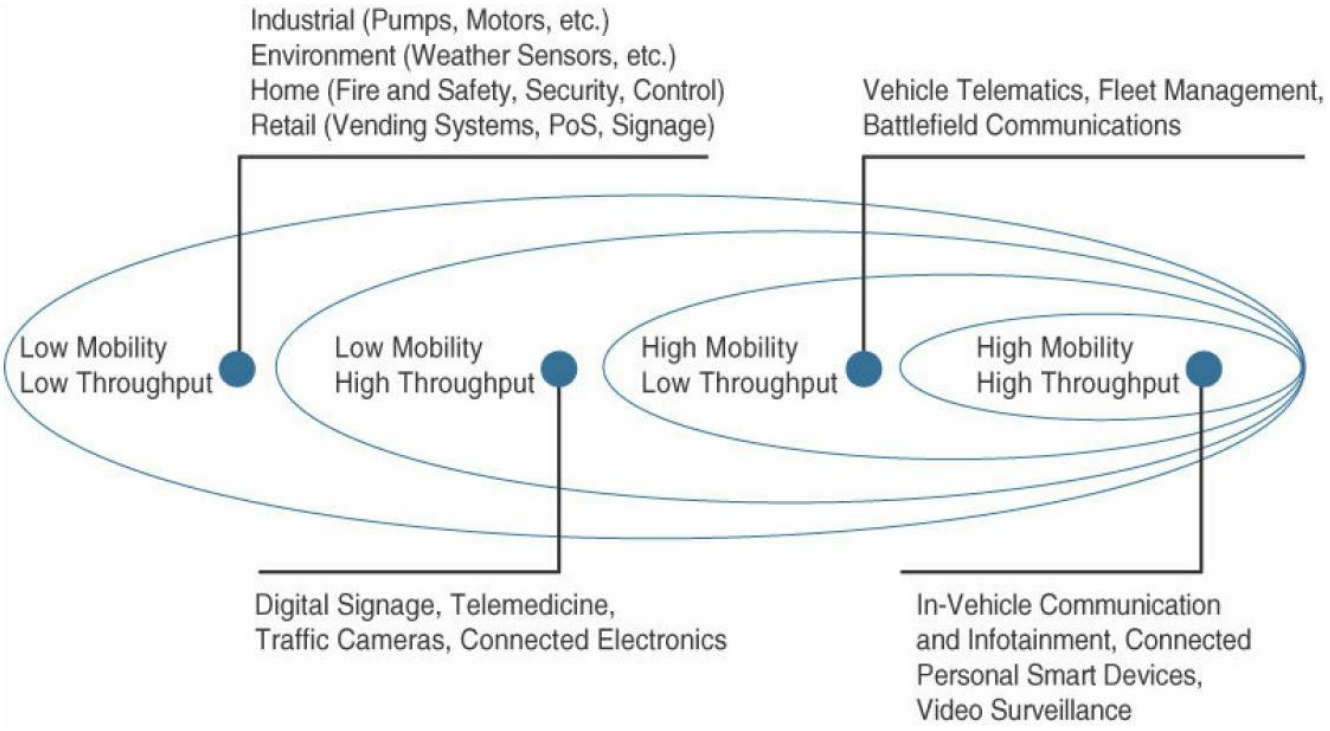


Fig 1.10. Sensory applications based on mobility and throughput [11]

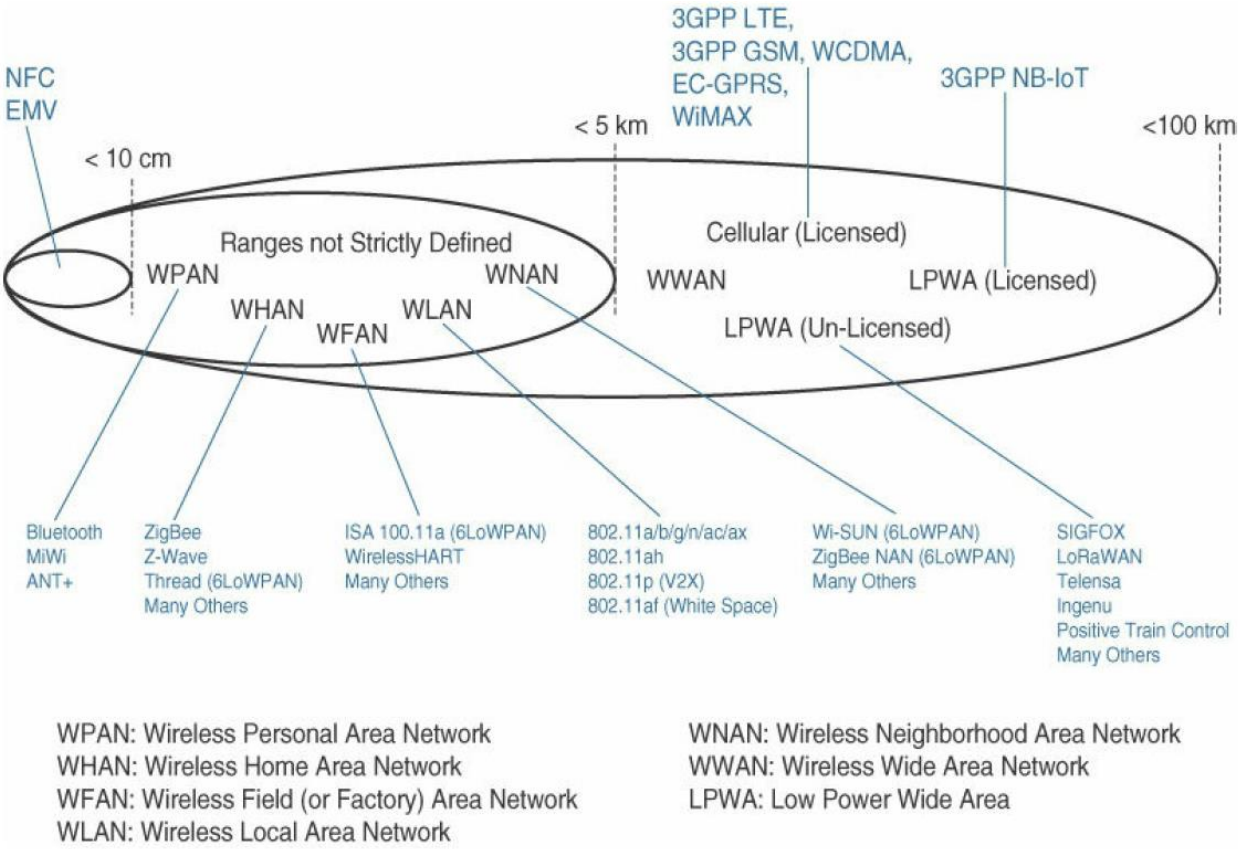


Fig. 1.11. Access technologies and distances [11]

1.3. Routing Related Issues and Challenges in IoT

Internet-of-Things refers to a loosely coupled, decentralized system of devices augmented with sensing, processing, and network capabilities. The devices senses, log, and interpret what's occurring in their proximity, intercommunicate with each other and exchange information. Due to this inherent heterogeneous nature of IoT, it is expected to have varying ranges of issues surrounding it, but its standout issues that are apparent are its interconnectivity and security of the IoT devices. However, as technologies evolves and IoT gains far deeper integration into the fabric of human day to day activity, new issues and challenges will also emerge extending the information security risks far more widely than the Internet has to date. IoT devices carrying RFID tags with a unique Electronic Product Code (EPC) and with the help of Object Naming Service (ONS) has

been proposed to address the security issues for IoT. However, these solutions inherit the traditionally weakness of DNS and also subjected to single point of failure.

Furthermore, despite the heterogeneous nature of IoT introducing complexity, it is also worth stating that it can also be its greatest assert as well. The most valuable part in IoT is a large number of devices are connected and linked to the Internet, each of which one is transmitting data, but its major challenge will be seen in the IoT network's ability to store the data, understand the large data and to finding path to transmit the data. Interconnectivity amongst the IoT devices requires new peer discovery methods, physical and MAC layer procedures that are different from traditional wired and wireless architectures.

Finally, it is also worth mentioning that aside routing, there are also other issues and challenges that accompany any heterogeneous network and IoT is not an exception. Stated below are some of the issues and challenges will be expected to find in a IoT network; access control, authentication, standardization and interoperability, security, privacy, identity management, Big data, scalability, software complexity, storage volume, data interpretation, fault-tolerance, ubiquitous data exchange through wireless technologies, energy-optimized solutions, service orchestration, things to cloud: computation and communication gateways.

Due to the inherent heterogeneous nature of IoT, it is expected to have varying ranges of issues surrounding it, but its standout issues that are apparent are its interconnectivity and security of the IoT devices. However, as technologies evolves and IoT gains far deeper integration into the fabric of human day to day activity, new issues and challenges will also emerge extending the information security risks far more widely than the Internet has to date. The table below shows some of the most significant challenges facing IoT.

Table 1.3. IoT Challenges [11]

Challenge	Description
Scale	IOT devices should have the ability to upscale and down scale automatically to meet the specific network and node requirement at a particular time.
Security	As more devices are connected and transmitting packets, ways to protect nouvelle nodes and networks needs to take into consideration as traditional methods of ensuring information technology will seem really inadequate for IOT.
Privacy	As sensors become more prolific in our everyday lives much of the data they gather will be specific to individuals and their activities. This data can range from health information to monetary transitions. Hence privacy needs to be thought of as regards with IoT.
Big data and data analysis	IoT and its large number if sensors is going to trigger a deluge of data that must be handled. This data will provide critical information and insights if it can be processed in an efficient manner. The challenge however is evaluating massive amounts of data arriving from different sources in various forms and doing so in a timely manner.
Interoperability	As with any other nascent technology, various protocols and architectures are jockeying for market share and standardization within IoT. Some of these protocols and architectures are based on proprietary elements and others are open.

1.4. SDN Based Architecture for IoT

Security as related to has always being a major concern for information technology and this concern pre-dates the current internet era we are in. in correlation with IoT, it is very easy to realize lots of security vulnerabilities across the various nodes or devices that makes up IoT system [12]. Traditional security mechanisms like Firewalling, Intrusion Detection and Prevention Systems are deployed at the Internet edge. Those mechanisms are used to protect the network from external attacks. Such mechanisms are no longer enough to secure the next generation Internet. The borderless architecture of the IoT raises additional concerns over network access control and software verification. In Ad-hoc network for IoT does not exist simple solution to control the exchanges between each node. For instance, if one thing is corrupted by a virus, this treat can propagate itself in the network without any control. Moreover, anyone can connect his things on the network. Details of network access control implementation based for IoT devices can be found in.

As a solution to this problem face by IoT, a nouvelle networking concept called software defined networking SDN was introduced, this methodology has the ability to protect the network and also cover the flaws that is present in traditional security systems. In SDN architectures, network devices do not make forwarding decisions.

Instead of that, network devices communicate with a special node, called the SDN controller, in order to provide them with the appropriate forwarding decisions. To communicate with the Controller, the network devices can use different protocols. The most used protocol for the communication between the SDN controller and the network devices is the OpenFlow. OpenFlow is what defines control messages that enable the SDN controller to establish a secure connection with the network devices, read their current state, and install forwarding instructions.

1.4.1. SDN Based Ad-Hoc Architecture

SDN based architecture for Ad-Hoc Network can be visualized as a combination of: legacy interfaces which is the physical layer, the programmable layer which is the SDN compatible virtual switch and an SDN controller and lastly an operating system and their applications which is the OS layer. The figure below shows the representation of how the three parts are set up.

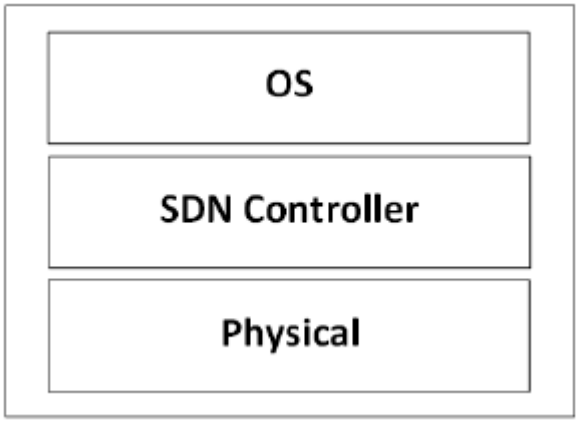


Fig 1.12. Node in an Ad-Hoc network [12]

In this architecture, all legacy interfaces are connected to a virtual switch, and this switch is controlled by an SDN controller, integrated on the node. Since all controllers of each node operate in equal interaction, they will have no need to be concerned about nodes liability for the misbehaving users connecting through them. Ad-Hoc users will connect with other nodes through their embedded SDN-compatible switch. At the same time, the SDN controller, in equal interaction, can enhance the security and connectivity between the nodes [12].

One of the feature of the Ad-Hoc architecture is backward compatibility, since each node in the Ad-Hoc network has an embedded SDN-compatible switch and an SDN controller, we can interconnect the Ad-Hoc network to the legacy network to construct an

Extended SDN domain as shown in the figure 1.13 below. Moreover, as all controllers of the extended SDN domain are in equal interaction, all rules will be synchronized.

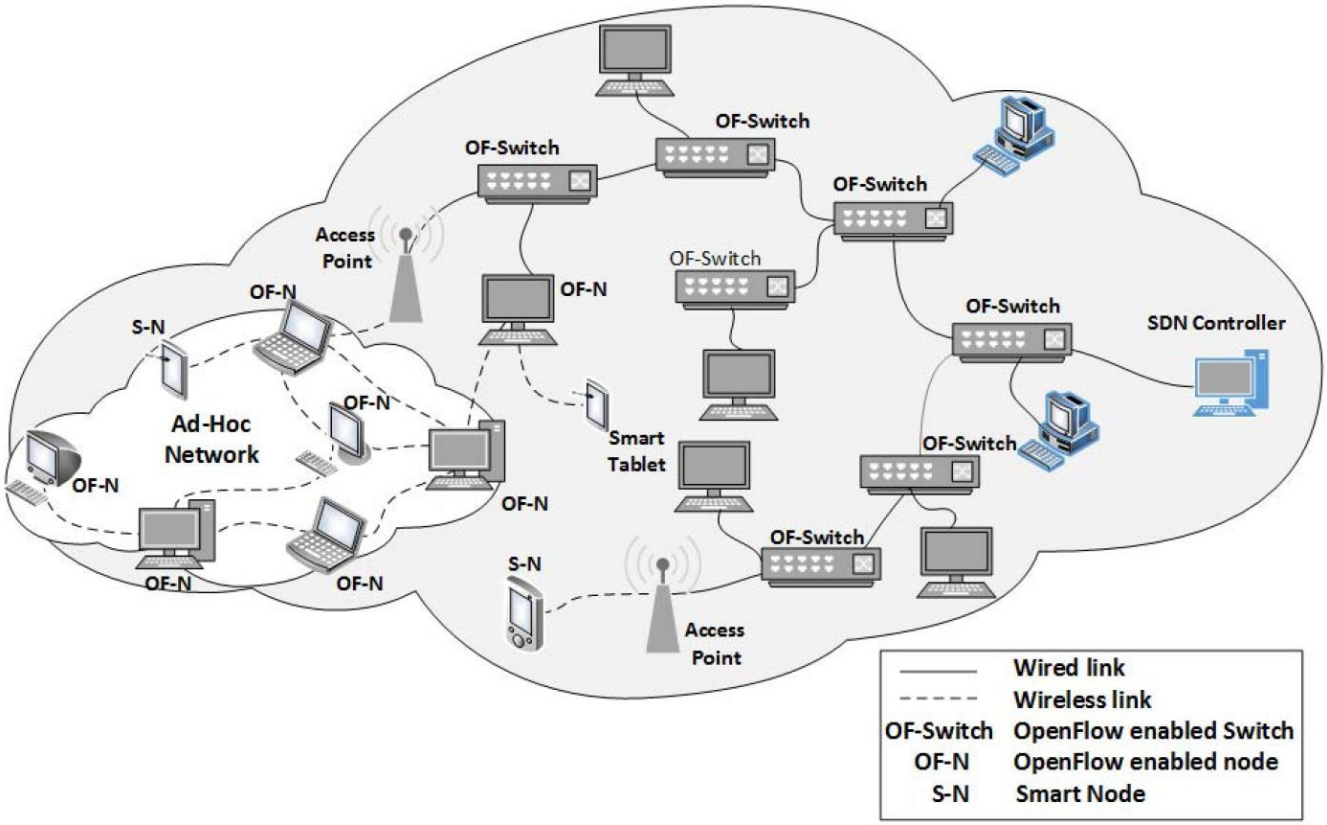


Fig. 1.13. Extended SDN domain [12]

As each Ad-Hoc node has its own SDN controller, the SDN control plane has to manage the evolution of each SDN virtual switch on each Ad-Hoc device. When a new Ad-Hoc device connects itself or leaves the network, we can have many exchanged messages in order to synchronize all the rules. In order to ensure scalability and fault tolerance, a distributed SDN architecture is preferred, with multiple controllers. To ensure that, we dynamically add new controllers to the Ad-Hoc network area and authorize special nodes to run control operation, as shown in the Figure 1.14 below.

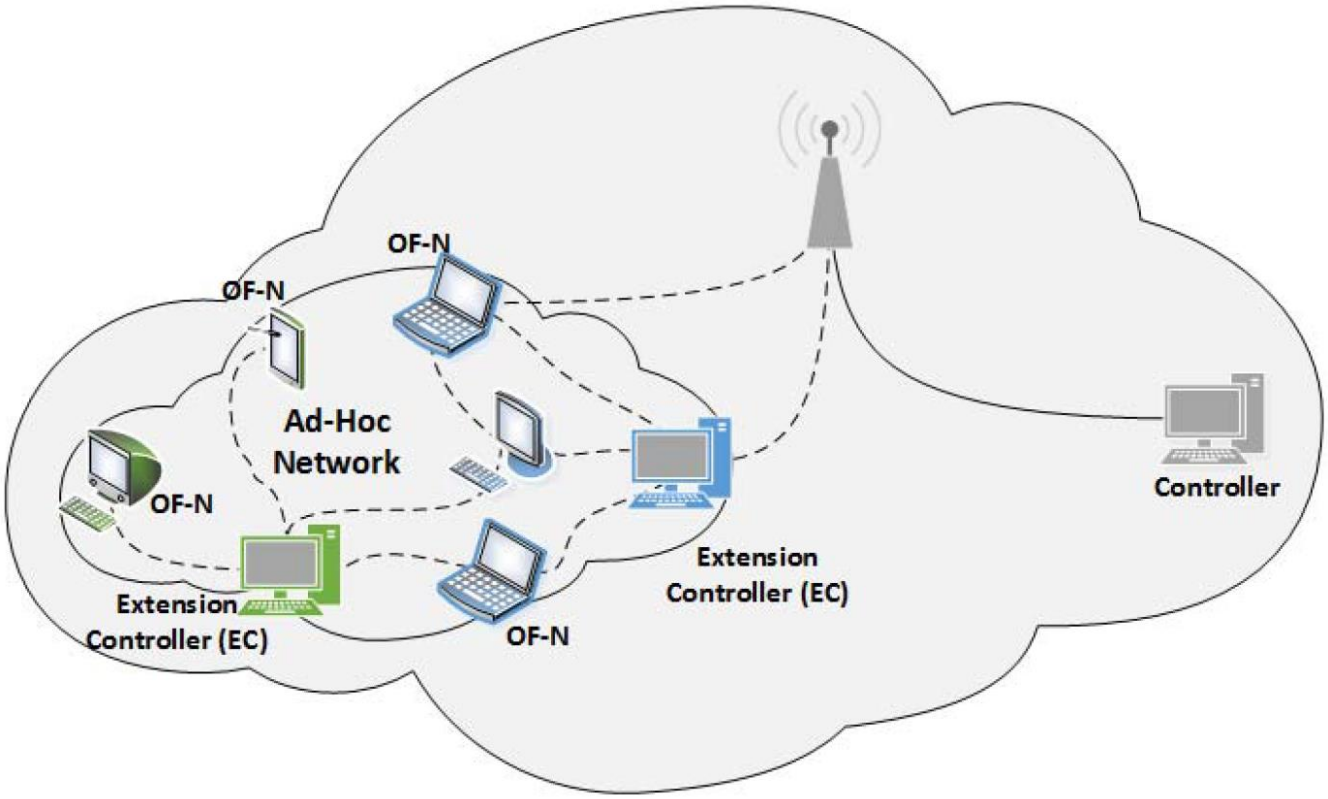


Fig. 1.14. Distributed Ad-Hoc control plane [12]

The distributed network access control architecture enables faster response to network changes. Moreover, it reacts to attacks occurring in the SDN extended domain, while sharing the traffic load management with the root controller.

1.4.2 SDN Based Architecture for IoT

An SDN-based architecture for the IoT requires heterogeneous interconnection with large number of SDN domains. In order to achieve such large scale interconnection, they need to be a new type of controller in each domain. The development and success of this architecture is based on equal interaction between controllers. These controllers are what are responsible for establishing and exchanging information with other SDN border controllers using existing security mechanisms. Furthermore, each SDN domain has its own security policies and management strategy [12].

1.4.3 Distributed SDN Security Solution

As it was stated previously, traditional Ad-Hoc architecture does not provide network access control or global traffic monitoring, due to the absence of the network infrastructure. Hence the adoption of the concept of distributed SDN security and management solution. In this concept, the controller can not only manage the network, but also monitors and efficiently secures the network against outside and inside attacks whilst providing other services such as authenticating network devices, users and objects connecting to users using both wired and wireless technologies. It also has the ability to achieve maximum synchronization of SDN Controllers in a security perimeter enabling a granular control over network access and continuous monitoring of network endpoints, thereby providing a more secure model for IoT. The figure below gives a visual representation of this model interconnection [12].

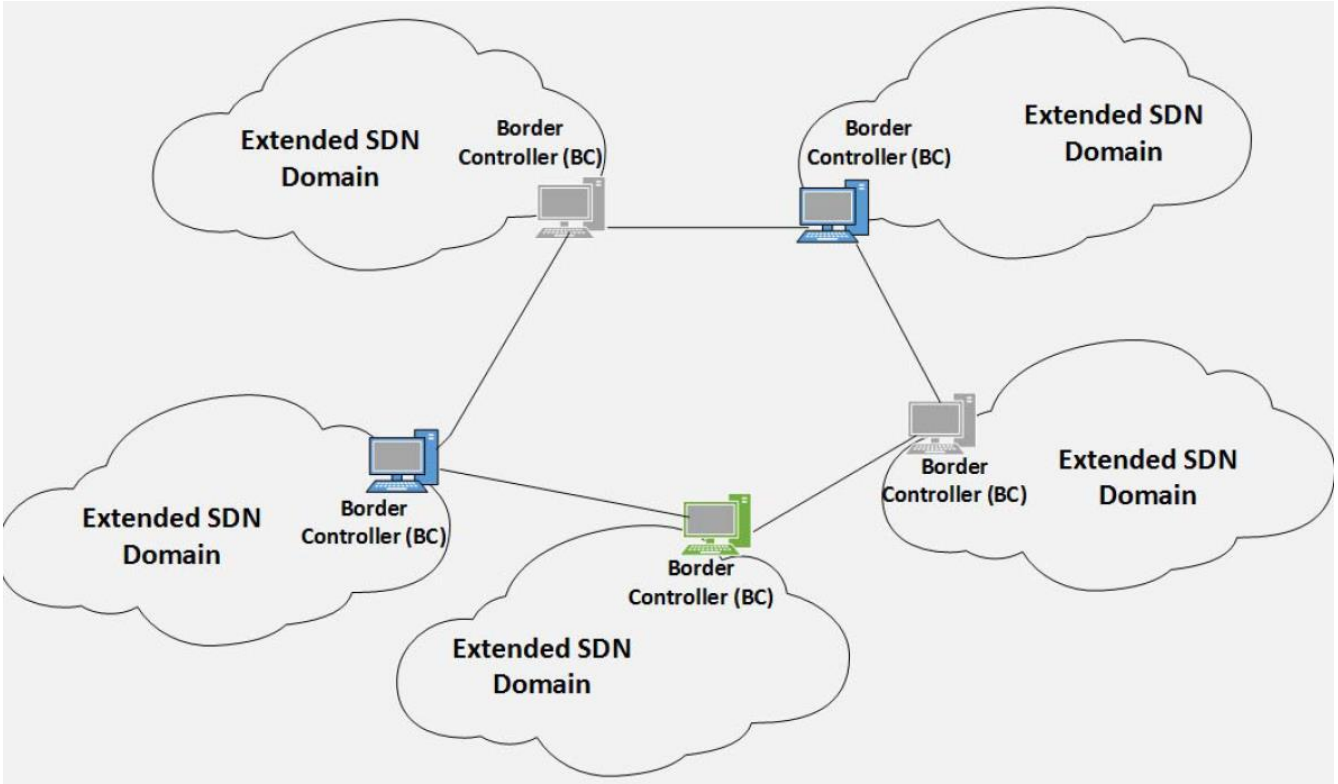


Fig. 1.15. SDN domain interconnection [12]

In order to secure network access and network resources, the SDN controllers begin by authenticating the network devices. Once the OpenFlow secure connection between the switch and the controller is established, the controller blocks switch ports directly connected to the users. After that, the controller authorizes only user's authentication traffic. Once the user is authenticated, and based on the authorization level of the user, the controller will push the appropriate flow entries to the software or the hardware access switch. In IoT, we extend this authentication process to devices. Each device has to associate itself with an OpenFlow enable node, each of which is connected to one controller in their domain [12].

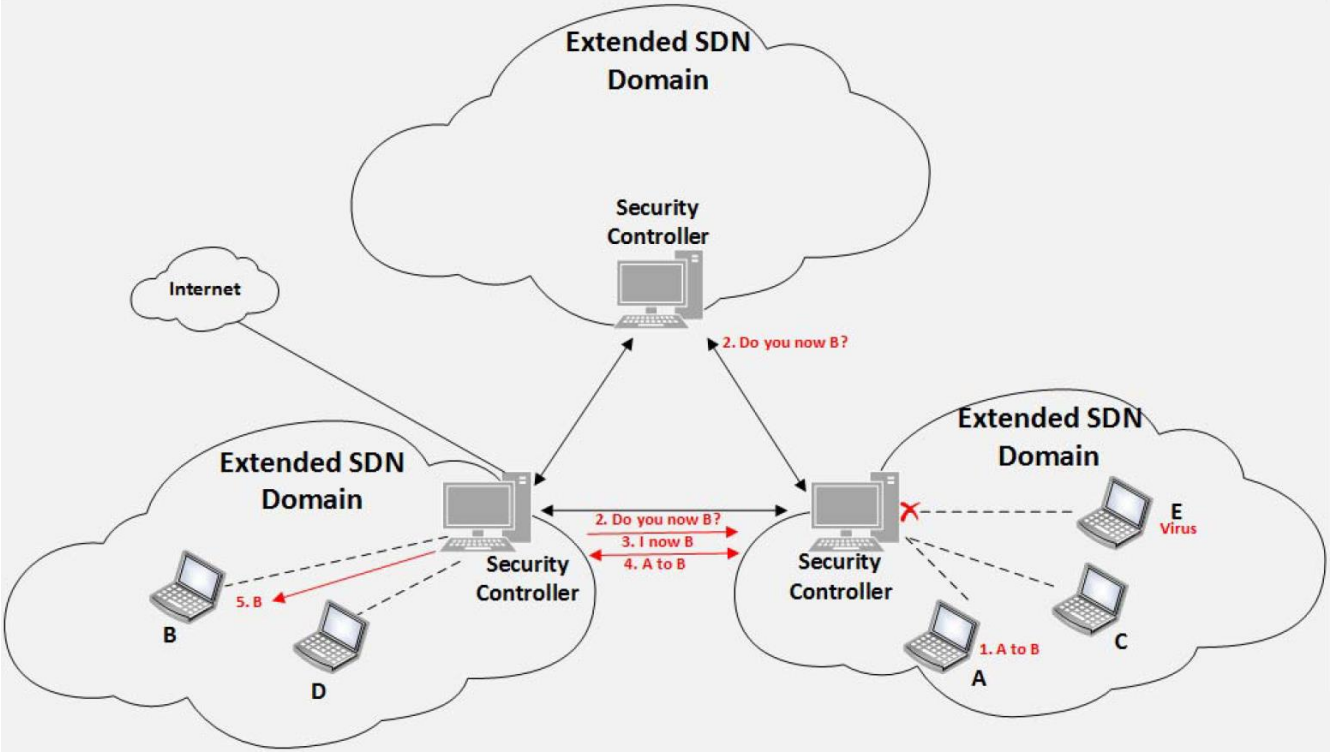


Fig. 1.16. Grid of security in SDN domain [12]

The whole concept of the grid of security network is to extend the SDN domain concept to multiple domains and each controller of each domain exchanges their security rules. Some of the SDN controllers will behave as security guards on the edge of the extended SDN Domain to ensure the network safety. Safety connections between domains

could be provisioned and only added to SDN Controllers then only recognized traffic could be accepted. The controllers know policies in their domain but they don't know policies of the other domains. So, when a node wants to communicate with another node of another domain, the flow has to be forward to the Security Controller, also called the Border Controller. The security controller asks each neighbor controller if it knows the destination of the information. In addition, the extended SDN controllers periodically monitor and check the flow table entries of the software switches because they are deployed on the user side. In the proposed architecture, we deployed software switches on the user's side to enhance the forwarding capabilities of Ad-Hoc devices. Moreover, the deployed software switches allow the SDN controller to apply and enforce the security policies inside the Ad-Hoc area. The figure below shows a visual representation of this set up.

CHAPTER 2

SECURE ROUTING SURVEY IN IOT

2.1. SECURITY IN IOT

IoT security is the act of securing devices and the networks they are connected to which in most cases these devices are constrained devices (Constrained devices are devices with limited resources such as CPU, memory (ROM and RAM), and battery life [13-21]. These devices often function as sensors collecting information, machine to machine (M2M) or smart devices controlling electrical appliances and services). IoT devices are being connected to the internet to allow for the collection and exchange of data with web servers and cloud data centers. Efforts are being made to standardize IoT devices and how they communicate with the web. At present, data communication in the web is primarily conducted using HTTP which was not designed for constrained environments and carries a lot of overhead. Furthermore, current securities measures found in traditional networks have proven to be ineffective when employed to IoT networks. Other protocols tailored for IoT such as CoAP has been developed by Constrained RESTful environments (CORE) as part of the Internet Engineering Task Force (IETF) and is a specialized application layer, web transfer protocol designed to be used with devices such as resource constrained IoT [13-21]. Unfortunately, the aforementioned protocols do not provide secure data transmissions by default on their own hence appropriate security measures must be applied such as cryptography to ensure the greater security of data. In 2014 the Open Web Applications Security Project (OWASP) ranked the top ten security issues facing IoT devices and placed the lack of encryption in data transmissions as number four on the list. To solve this problem security protocols are needed. Transport Layer Security protocols are added to HTTP and CoAP in an effort to secure communication. HTTP is secured using TLS and CoAP using DTLS. While much

work has been done by IETF to minimize the resource requirement, DTLS is still a heavy weight protocol and many IoT devices fall short of the minimum resources needed to support it. The use of cryptographic functions in DTLS adds much complexity and demand for resources on an IoT device. As a result, devices with system resources below the minimum of 10KB Random Access Memory (RAM) and 100KB Read Only Memory (ROM) are considered to be too constrained to effectively support the transport layer 5 security mechanisms needed to provide secure communications over the internet. These devices are known as class-0 devices. Other classes been class-1 and class-2 devices.

"Things," in the IoT world consists of wide variety of devices, for example: heart monitoring implants, biochip transponders on farm animals, electric clams in coastal waters, automobiles with built-in sensors, or field operation devices that assist firefighters in search and rescue operations. These devices collect required information with the help of various existing technologies and then autonomously share the information between other devices. Current market examples include smart thermostat systems and washer/dryers that use Wi-Fi for remote monitoring.

Besides providing the infrastructure for the tremendous number of new devices and application areas for Internet connected automation to exchange the information, IoT is also expected to generate large amounts of data from diverse locations that is aggregated very quickly, thereby increasing the need to better index, store and process such data is an avoidable concern to flourish this technology. But The IoT's anywhere, anything, anytime nature could easily change these advantages into disadvantages, if privacy aspects would not be provided enough. For example, if anyone can have access to any personal services and information, or if the information of a wide range of people can be reached by the environment automatically, the IoT would not have a reliable environment. There is not any sufficient backbone to define control and information asymmetry policies for interaction among any different users and devices. Controlling the flow with the traditional tools will cause a huge amount of traffic that is hard to guarantee the privacy and protection for elements. Also, solutions for different security requirements have direct

impact on the cost and time to market. Moreover, every solution has its own business requirements which may or may not be as strict. Another important issue in IoT is the quality of the user's satisfaction. IoT should provide a better service by avoiding and rejecting certain services that may happen by current classic mechanisms used to obtain user's consent. Hence, IoT should provide different methods such as implementing consent mechanisms through the devices themselves as privacy proxies and policies for each device, which includes conditions and constraints attached to data that describe how it should be treated.

2.2 Factors Influencing Routing in IoT

Routing is the process of selecting a path for traffic in a network or between or across multiple networks [14]. Broadly, routing is performed in many types of networks, including circuit-switched networks, such as the public switched telephone network, and computer networks, such as the Internet and routing is also a big part when it comes to Internet of things architecture. One of the major hurdle to expansion and growth of IoT was issue of proper and efficient routing as well as other issues, such as secure management, cooperation and coordination within the devices and hubs. Devices in network have to communicate with one another for sensed/collected data, process the data, sharing data and transmission of the data in multi-hop approach. IoT system generates massive data frequently; this leads to the formation of an intellectual environment the received data have to be transmitted into intelligence. This intellectual environment may play an important role while routing the information in the IoT system.

The transmission of data and routing in IoT network is a difficult issue, because large amount of data collection and accumulating can be estimated. The leading protocols for routing in IoT network are RPL (Routing Protocol for Low- power and lossy networks) and 6LoWPAN for IoT networks. The routing issues become more challenging in

dynamicity of the IoT network due to low-power and lossy radio-links, constrained power, multiple hops mesh topology and battery operated devices.

Routing, in general, answers the question of "how an entity is brought from an origin to a destination. In the context of the IoT, the entity is a data packet, and the origin and destination of the data packet are two computing devices and are called the source and the destination respectively. However, there are some factors which may inhibit the smooth process of routing. Some of this factors are shown in the table below.

Table 2.1. Factors affecting routing in IoT [14]

Factors	Description
Devices	Devices may have identical or dissimilar types
Network System	The origin of source and to the target devices may be present in a network or in another network
Longevity	Connectivity/longevity between the IOT nodes may be continuous or discontinuous
Resources	Inadequate availability of resources
Co-operation in data relaying	Non- corporation of the devices mainly due to limitation of the resource
Communication process	Varying the mode of data communication(single or multi hop)
Network topology	Changing of network topology due to mobility of the devices
Communication range	Communication ranges of the devices mainly depend upon the manufacturers and by vendors
Intolerant environmental solutions	Intolerant environmental situations like flood, cyclone, humidity, low/high temperature

Addressing mechanisms	These devices should have unique addressing mode and universally acceptable for the creation of devices to devices communication.

2.3 Routing Challenges

There are many challenges that can affect routing in the IoT [14]. The challenges can come from the routing layer itself, or from the layers underneath it such as physical and medium access control (MAC) layers. Although a cross-layer approach can be employed to take advantage from the properties of the lower layers in the design of a routing protocol, such approach would tighten the routing protocol to a (set of) specific MAC and physical designs. This would limit the use of the routing protocol to only a few types of IoT devices, whereas IoT devices are known to be extremely heterogeneous. In this work, our focus is the routing layer. In the following, we present the major challenges that directly affect routing in the IoT.

SCALABILITY: The IoT will be large in scale, both in terms of number of nodes and geographically. As routing means to decide over which routing path the data packet should be sent, the more candidate relay nodes to be evaluated for inclusion in a routing path, the more complex routing is. This complexity is many fold, including what cost function to be used, how to decide which of the neighbors of a node is the relay node, what is the cost to setup and maintain a routing path, how to setup new a routing path when another one is broken, etc. Such complexity will quickly grow unmanageable if the routing protocol was not carefully designed with scalability challenge being taken into account.

DYNAMIC ROUTING TOPOLOGY: The cause of the dynamicity of the routing topology is many fold. Firstly, due to energy constraint, IoT devices are usually scheduled to be idle or working (e.g., by turning the wireless radio on/off) to minimize energy consumption, making the routing topology dynamic. Secondly, since users deploy or

remove their IoT devices at will, routing nodes will be connected to and disconnected from the IoT at unknown rate, which adds the unpredictability to the dynamicity of the routing topology. Thirdly, node failures are common in the IoT. The causes of a failure include hardware malfunctioning (e.g., antenna damage), exhausted energy supply (e.g., depleted batteries), and environmental impact (e.g., the air humidity level is unexpectedly high causing shortcuts). Fourthly, node mobility causes the wireless links between the mobile nodes and other nodes in their proximity to be reconfigured. Finally, the low-power wireless links in networks of IoT devices (e.g., WPAN, WSN) are unreliable and transitional, which also contributes to the dynamicity of the topology. The routing protocols, hence, must be flexible enough to deal with such dynamicity of the IoT's topology.

LIMITED RESOURCES: Another major challenge to routing in the IoT is the presence of network partitions and voids in the network. A partition is a disconnected part of the network, such that nodes inside a partition cannot communicate with nodes in the other parts of the network, because there is no routing path to exchange data packets. A void is an area that is not covered by the network. Since there is no node located inside the void that is connected to node(s) outside it, data packets can only be forwarded around the void to reach to their destination. For example, a WSN has been deployed by randomly scattering a large number of sensor nodes over a geographical area. Due to the structure of the area, there may be lakes that cause voids, or rivers that cause partitions in the WSN.

PARTITIONS AND VOIDS: Partitioning of the network based on applications and voids are also the challenges of routing in IoT. Partition, here means separated from a part of the network. The devices within the partition cannot make communication with other network of the devices, since there is no routing path exists between the partitioned networks to transmit the data. While, Void means, it does not contain any devices or network (no devices is present inside the void) [14].

2.4 Routing Attacks in IoT

Internet of Things (IoT) could be described as the pervasive and global network where real-world entities augmented with computing devices, sensors and actuators are connected to the Internet, enabling them to publish their generated data [20, 22]. Thus, an efficient and secure routing service is required to enable efficient communication of information among IoT nodes. This sophisticated, dynamic, and ultra-large-scale network requires the use of contextual information, attention to security issues and the consideration of service quality to make proper routing decisions.

Security threats against routing and IOT in general can be classified into two types: passive or active attacks The purpose of a passive attack is usually to eavesdrop on routing communication and to retrieve information from monitored data packets. In a passive attack a malicious network node tries to identify communication parties and the contents of their communication. This can open up possibilities to launch further security attacks. The attack is passive since the normal network communication is not altered. In an active attack a malicious node tries to interrupt, disturb, and/or change the routing functionality.

Furthermore, attacks in IOT can be grouped into four different categories and there are:

- Passive attacks;
- Active attacks;
- External attacks;
- Internal attacks.

Below is a list of some of these attacks related to IoT:

- **Spoofed, altered or replayed routing**: In IoT, with the help of the routing protocol communication is possible between a source node to destination node with their Unique ID. When a particular device is wanted to send data packets to the destination node and then an attacker targets the information that going between the

source node and destination node. The attacker also creates routing loop, extend or shorten a source path, generate false message and besides the partition the network.

- **Selective forwarding attack**: In this selective forwarding attack, the attacker may introduce malicious node in the network. It node may refuse or not forward data packets, or it may simply the drop the packets, ensuring that they are not propagated any further. However, such an attacker runs the risks that neighboring nodes will conclude that it has failed and decides to seek another route.
- **Worm hole attack**: In the Network, if the source node wants to send the data to the destination node. Then it passes from one node to another node called as hop and to reach the destination with the help of multiple hops. The attacker could convince nodes they have one or two hop to reach a destination via the wormhole node. The attacker adds the node between two legitimate nodes.
- **Sniffing attack**: It is a good example of interception or listen-in channel attack. In this attack, an attacker is placed in the proximity (neighbor) of the sensor grid to capture data. Collected data is transferred to the intruder by some means for further processing. Not affect the normal functionality of the routing protocols. An attacker can launch this attack for gather valuable data from the sensors.
- **Node replication attack**: This is an attack where an attacker tries to mount several nodes with same identity at different places of existing network. There are two methods for mounting this attack.
  - In first method, the attacker captures one node from the network and creates a clone of a captured node. It is mounts in different places in the network.
  - In this method, an attacker may generate a false identification of a node. Then this attacker creates a duplicate of node and tries to generate insecure data to change the network setting.
- **Black hole attack**: In this, an attacker places a node in range of the sink and attracts the entire traffic to be routed through it. This attacker node is advertising itself as

the shortest path route. In the network, an attacker drops the packet coming from specific source. This attack can isolate certain nodes from the base station and creates a discontinuity in the network connectivity. This attack generally targets the flooding based protocols.

- **Energy drain attack**: Energy is important part of the Internet of Things (IoT). The aim of this attack is to crush the sensor nodes in IoT, degrade the performance of the network and ultimately split the network grid and consequently, take control of the sensor network by inserting a new sink node. The energy of a sensor node is down because of the limited amount of energy available. The attacker may use compromised nodes to inject fabricated reports into the network or generate the large amount of traffic in the network.
- **Homing attacks**: The attacker looks at network traffic to guess the geographic location of critical nodes such as cluster heads or neighbor of the base station. The attacker physically disables these nodes. This attack aims to block the whole traffic to the sink and provide a better grouped for launching another attack like data integrity.

2.5 Secure Routing Protocols in IoT and Issues

Core to functionality of IoT is its routing and the way low powered sensory devices or nodes self-organize and share information either data packets information or routes in between themselves even if the sensory devices are energy constrained, they however perform storage and computation functions while communicating over lossy channels [18-25]. The nodes also work in tandem and can join or leave the network at any given moment when it service is no longer needed. Therefore, it is upmost importance that wireless routing solution for these scalable and autonomous while being energy efficient. The devices used in these low power lossy networks are nothing but sensors and actuators

embedded with routing capabilities. Some of these sensor nodes acts as border routers (called LBR) and hence connect the low power lossy networks to the internet or to a closely local area network.

The internet Engineering task force (IETF) body has developed some standards for IoT routing protocols. Examples of these routing protocols are following.

6LoWPANs (IPv6-based Low-Power Personal Area Networks) are formulated by devices that are compatible with the IEEE 802.15.4 standard. To moderate the effects of network mobility, the Internet Protocol (IP) does not calculate routes; it is left to a routing protocol, which maintains routing tables in the routers. 6LowPAN uses an adaptation layer between the network (IPv6) and data link layer (IEEE802.15.4 MAC) to fragment and reassemble IPv6 packets. The routing in 6LoWPAN is primarily divided on the basis of routing decision taken on adaptation or network layer.

6LoWPANs are formed by devices that are compatible with the IEEE 802.15.4. However, ZigBee uses the IEEE 802.15.4 standard as its communication protocol for Medium Access Control (MAC) layer and Physical (PHY) layer. IEEE 802.15.4 devices are characterized by low computational power, scarce memory capacity, lower bit rate, short range, and low cost. 6LoWPAN have devices that work together and connect the physical working environment to real-world applications like sensors with wireless application. Some protocols exist in sensor networks that have a non-IP network layer protocol such as ZigBee, where the TCP/IP protocol is not used.

RPL was developed by the IETF as functionalities in another routing protocol 6LoWPAN was very challenging due to the resource constrained nature of the nodes. RPL operates at the network layer making it capable to quickly build up routes and distribute route information among other nodes in an efficient manner. RPL is a distance vector IPV6 routing protocol for LLN, hence network path information is organized as a set of directed Acyclic graphs(DAGs)

IPv6 over the time slotted channel hopping mode of IEEE 802.15.4e (6TiSCH): The development of this IoT protocol is currently ongoing and has not being deployed yet.

It will be based on IPv6's multi-link subnet spanning over high speed IEEE 802.15.4e TiSCH wireless mesh networks linked to the backbone via synchronized backbone routers. The new protocol will include details about how packets, belonging to a deterministicIPv6 flow, may be treated while issues such as classification, routing and forwarding of packets over the mesh network can bead dressed. Other areas to be addressed will include security, link management for the IPv6 network layer, neighbor discovery and routing [6tiSCHesIoT]. More electronic systems are being ported from closed systems (such as Modbus, SCADA) into IP-based systems and this will further increase the risk of more attacks.

2.6 Secure Routing in IoT: Limiting Factors

Routing protocols enables nodes acting as routers to exchange routes details to create routes between nodes. These route details could become a target for malicious nodes who intend to cause harm to the network. From the standard protocols been set by IETF body, it can be seen that none of them is immune to attacks in whatever form those attacks may come, hence implementing a secure routing technique can be considered as an herculean task, but nonetheless if one does have an idea of some malicious attack, one may find a way to protect itself from it. Below are list of some of the limiting factors that needs to be taken into account when working towards achieving a secure routing system for IOT networks.

- Energy levels of the IOT nodes:
- Scalability of the IOT networks
- Heterogeneity of the IOT networks
- Unavailability of intermediate nodes
- Memory and CPU/GPU/TPU capacity
- Open wireless medium

- Mobility of the IOT networks

2.7 Trust Based Protocols

Trust can be defined as the affiliation between two parties in which one party is willing to count on the actions to be performed by the other party. Trust can be classified according to time and conditions [15]. In the domain of computer science, and in particular sensor networks, trust is a complex term, it is based on computational perspective as the level of doubt and optimism regarding an outcome with the consideration and perspective of the individual. This is the aggregation of positive direct experiences among nodes. it refers to the confidence and belief about the reliability, integrity, security, dependability and character of cumulative trust value of a node is used in defining the reputation of a node which is a quantifiable limit of the observable experience a node has with its neighbor either directly or indirectly. Trust level is maintained on the basis of trust value which is calculated from algorithms. This is used in the decision of the limit of trust that a node will have towards its neighbor. Trust value can be measured on a continuous scale 0 to 1, where 0 is means no trust and 1 means complete trust. Two methods have been proposed: centralized and distributed methods. There are two methods proposed in distributed approach: direct trust computation and indirect trust computation. In direct trust computation, trust value is calculated on the basis of its own experience and in indirect one, it is calculated from the recommendation received from neighbors. In distributed methods, nodes are independent.

There are factors which can be highlighted as reasons why nodes can be compromised and rendered vulnerable to attacks. Examples of these factors can be: mutual interference of wireless links, battlefield applications and nodes without good physical protection.

There are various trust models been employed in the analysis of trust within an IOT system. Examples of these trust modelling is the practice of using trust in the evaluation

of a system. It shows the concerns that affect the trust of a system while helping to identify areas where a low value of trust could degrade a system operational efficiency and functionality [15]. Examples are:

a) Bayesian trust model
b) Fuzzy trust model
c) Probability trust model
d) Swarm intelligence model
e) Neural network model

The trust based security methods provide security to sensor networks without using cryptographic approach. Trust based solutions help in predicting future actions of the nodes in accordance with their past behavior. The existing routing protocols have some limitations. First, importance to dynamic detection of faulty or damaged nodes is not given which prevents to forward packet successfully. Second, they do not optimize the end to end route across the network. Third, they incur high routing and computational overhead. Fourth, they target only powerful hardware platforms but cannot be directly applied to wireless sensor networks.

- Node Behavioral Strategies Banding Belief Theory of Trust Evaluation Algorithm: In order to solve this network security problem, an algorithm was proposed by NBBTE (Node Behavioral Strategies Banding Belief Theory of Trust Evaluation Algorithm) which combines nodes behavioral strategies and modified evidence theory. This algorithm analyses the behavior of sensor nodes and establishes a variety of trust factors and coefficients related to the network application which in turn are used to obtain direct and indirect trust values through calculating weighted average of trust factors. On the other hand, the fuzzy set method is used to form the basic input vector of evidence. The fuzzy set theory is used to determine the trustworthiness levels. Using this, evidence difference is calculated between the direct and indirect trust values, and Dempster evidence combination rule is used to produce integrated trust value of nodes.

- Trust and Energy aware Routing Protocol (TERP): which makes use of a distributed trust model for detection and removal of faulty nodes and addresses the existing limitations of the trust based routing protocols. This algorithm uses trust, residual-energy and hop counts of neighbor nodes in making routing decisions. Keeping resource-constrained characteristics of WSN in mind, this protocol is cantered in trustworthiness and energy efficiency. This protocol includes mechanisms which ensure selection of end-to-end routes keeping in mind energy levels of intermediate nodes, ensuring longer lifetime of the network. This algorithm also forwards packets through shortest of paths that consist of trusted and energy efficient nodes that lead to balancing the energy consumption among the trusted nodes. The simulation results reveal improved performance of TERP algorithm proposed as compared to the existing ones.
- Another proposed algorithm for optimizing the fault tolerance and minimizing the communication delay in virtualization. The algorithms proposed earlier did not give importance to the communication failure. The most common issue that occur in virtual networks is fault toleration due to changing wireless channel based connectivity. The use of bandwidth reservation for removing fault tolerance decreases resource use and hence is not that efficient. The algorithm proposed in this paper minimizes fault tolerance and communication delay for different virtual networks in IOT. In this paper, the fault tolerance removal problem is made considering fault tolerance and communication delay as two incompatible factors. Also sorting based genetic algorithm is used to solve optimization problem. The proposed framework is more effective as compared to the earlier ones. Simulation results have shown that better optimization results are obtained and they are also achieved in shorter time as compared to the earlier proposed algorithms. This proves the efficiency of the proposed algorithm.

- A trust management algorithm is proposed based on DS evidence theory. This proposed algorithm maintains the unknowing and transitive features of trust. Also, time as a factor is used to keep track of past weight information with the current weight information in real time which increases the accuracy of the trust calculation. To keep the proposed algorithm green i.e. energy efficient and cost efficient, a synthesis method is used to acquire recommended trust values if no direct trust values are present. Also this algorithm is better to fight against attacks such as on-off attack and mouth attack and also detection of faulty nodes than the previous proposed algorithms. The proposed algorithm is again more energy efficient as compared to the previous proposed algorithms. Simulation results show that the proposed algorithm can effectively fight attacks like on-off attack and also can effectively calculate trust values of various sensor nodes present in the network. Furthermore, the flexible synthesis method is very energy efficient and thus increases the life time of the network.
- This algorithm deals with the identification of these types of malicious nodes in a hierarchical architecture of WSN. Hierarchical network architecture is used because it implements clustered methodology that minimizes network overhead. The sensor nodes are located at the lowest layer which directly communicate with the environment. The nodes in second layer are of high computational capability which receive information from lower layer and forward them to the base station where further calculation is done. This paper has introduced the concept of weight based nodes. The sensor nodes are provided with certain weightage. The forwarding layer computes the aggregation of all the information received from lower nodes. This result is used to evaluate the malfunctioning. The malicious node sends fluctuating data i.e. weightage is also unsteady. Finally, the weight value is supposed to be decreasing. A lower limit is set so that the nodes are detected as malfunctioning.

The practical results show that this approach is efficient in finding the affected nodes.

- This algorithm introduces secure process of creating new network or adding devices to existing network. The process involves different layers of authentication in order to make sure that secure devices are entered into the network. The process involves two steps. At first, the device is searched for getting it into the network. Then the new device is proceeded for authentication and verification. The authentication process involves identifying the device and matching it with existing applications and addresses. The approach introduced here uses a tricky technique (i.e. by altering the confidential information frequently) to resist external attacks. When a device wants to join the network or wants to create a new network, its security information is checked by specific method and if the test passes then the devices are authorized to enter the network. This paper has experimented the topology practically. The results showed that the network able to maintain its security without any threats during the experimentation period.
- This algorithm is based on how to minimize the consumption of energy when number of connected devices in IOT are increasing day by day. Energy conservation has become a great challenge in today's world. As there is a great increase in the number of connected devices in a network, the nodes in the network consume more energy to send and receive signals. So, this paper is proposing an effective approach for economic use of energy in a wireless sensor network. It has proposed a hierarchical design of network where nodes are divided into different levels, from top to bottom. All the nodes (i.e Sensor nodes, cluster coordinates, relay nodes and normal nodes) present at the lower level are not allowed to among each other. The sensor nodes collect information from the surrounding and send them to the relay nodes which passes the information to the cluster head. The cluster head sends signals to the cluster coordinates of upper layers and finally to the

topmost layer. The top layer is known as base station. This design provides ascendible approach that is flexible as well. The next topic this paper addresses is communication algorithm that tends to reduce the energy consumption while transmitting signal among nodes. The nodes in lower layers are supposed to be active more than the nodes at upper layers. So, minimizing the object distance can be a sustainable mode of reducing the energy consumption. Through numerical verification, this paper has proposed an effective model for WSN.

- Finally, a proposed model for securing the sensor network based on trust mechanism. The model analyses the communication trust based on the how much packets are correctly forwarded by the nodes. The Direct trust for the node is evaluated based on direct observation by the node for neighbor behavior which is defined as using the equation:

$$x_b^a = \frac{(packets\ forwarded)}{(packet\ dropped + packet\ forwared)}. \quad (2.1)$$

Based on the behavior, whether it is sending or dropping the punishment or appreciation is added. Further, indirect trust is obtained by evaluating the recommendation from neighbors. Though, all the recommendations are not trustworthy for which the recommendation credibility of the node sending the value is to be consider. Credibility is achieved after finding the similarity of behavior between nodes. The error will determine how different node acts differently. Similarity between node will be determine. Further dempster Schaffer theorem will be used for finding the Indirect trust value.

$$Er_b^a = \sqrt{\frac{D_a^{node} - D_b^{node}}{neigbour^{ab}}}. \quad (2.2)$$

Below is a table showing these methods.

Table 2.2. Trust based security models [15]

Secure Routing Protocol	Technique Applied	Advantages	Shortcomings
A Trust Evaluation Algorithm for Wireless Sensor Networks Based on Node Behaviors and D-S Evidence Theory	Fuzzy Set Theory, NBBTE Algorithm	Able to efficiently identify the malicious node	Energy consumption is more and it is costly
TERP: A Trust and Energy Aware Routing Protocol For wireless Sensor Network	Trust Estimation and Trust-Energy aware Routing Algorithms	Efficient and reliable in obstinate environment. energy efficient	Less resistant to attacks such as Sybil, wormhole
Virtualization in Wireless Sensor Networks: Fault Tolerant Embedding for Internet of Things	An adapted non-dominated sorting based genetic algorithm (ANSGA)	Based on heterogeneous network environment, more capable of tolerating fragile network, reduces the delay in data transmission	Less network parameters considered
A Novel Deployment	Minimal Energy Consumption	Energy efficient scheme, longer	Not available

Scheme for Green Internet of Thing	Algorithm (MECA)	lifespan of network	
Trust Management Scheme Based on D-S Evidence Theory for Wireless Sensor Networks	Dempster-Shafer evidence theory, TMS algorithm	Can withstand on-off attack and bad mouthing attack	Not available
Malicious Node Detection in Wireless Sensor networks using Weighted Trust Evaluation	Weightage based Analysis	Relevant to both wired and wireless networks	May not perform well in high traffic network
A Secure Multi-Hop Routing for IOT communi-cation	Single multi-hop routing protocol	More efficient than OSLR protocol, routing and authentication coalesced providing minimum overhead to the network	Not available
A Secure Multi-Hop Routing for IOT communication	Secure multi-hop routing protocol	More efficient than OSLR protocol, routing and authentication coalesced providing minimum overhead to the network	Not available

A Novel Scheme for an Energy Efficient Internet of Things Based on Wireless Sensor Networks	Minimum Energy Consumption Chain Based Algorithm (ME-CBCCP),	Energy Consumption Chain Based Algorithm (ME-CBCCP), Expends less energy than LEACH and ERP	May not support dynamic environment and small networks
An Evolutionary Secure Energy Efficient Routing Protocol in Internet of Things	Fuzzy c-means (FCM) algorithm, Improved Bacterial Foraging Optimization (IBFO) algorithm, Optimal Secured Energy Aware Protocol (OSEAP)	More lifespan of devices	Not available

CHAPTER 3

MODEL OF SECURE ROUTING IN IOT AND ITS INVESTIGATION

Compared to wired networks, ensuring of information security in IoT is associated with the detection and prevention of many existing vulnerabilities and attacks [26-28]. Firstly, wireless channels are more susceptible to attacks such as passive listening (eavesdropping), active interference of signals and jamming. Secondly, the majority of routing protocols in IoT imply trusted interaction between participating nodes for packet transmission. Dependence on such interaction makes data more vulnerable to unauthorized access, data substitution, and attacks such as "Denial of Service" (DoS). Thirdly, the absence of fixed infrastructure and centralized management makes it difficult to apply many of the traditional solutions to ensure information security.

3.1. Threshold Message Sharing Mechanism

One of the approaches of ensuring the specified level of information security in communication networks is the implementation of SPREAD mechanism [26], based on the multipath message routing after its fragmentation to parts in accordance with the Shamir`s scheme (Fig. 4.1) [26, 27]. Shamir's Secret Sharing is an algorithm in cryptography created by Adi Shamir. It is a form of secret sharing, where a secret is divided into parts, giving each participant its own unique part. To reconstruct the original secret, a minimum number of parts is required. In the threshold scheme this number is less than the total number of parts. Otherwise, all participants are needed to reconstruct the original secret.

As a result of using SPREAD mechanism, it is possible to reduce the probability of compromise of the transmitted message, because it complicates the adversary`s task: it must compromise not only one path that passed undivided message, but all paths

transmitting its fragments. A message is compromised in case of unauthorized access to its content, i.e. in order to compromise the message, transmitted using SPREAD mechanism, all the paths used to deliver message fragments must be compromised. Thus, the fact of a compromised path is adversary access to all message fragments, transmitted over this path. It should be noted that probability of compromise of individual path depends on the number of nodes and links it consists of and their security parameters, i.e. each element of the path (node, link) can be compromised with a certain probability. In general, various paths used to transmit the message fragments obtained in accordance with the Shamir`s scheme [26] can have different values of the probability of compromise. Unfortunately, under the well-known mathematical models [26, 27] devoted to realization of SPREAD in message fragments allocation over the non-overlapping paths security parameters (such as the probability of compromise) of these are not considered explicitly.

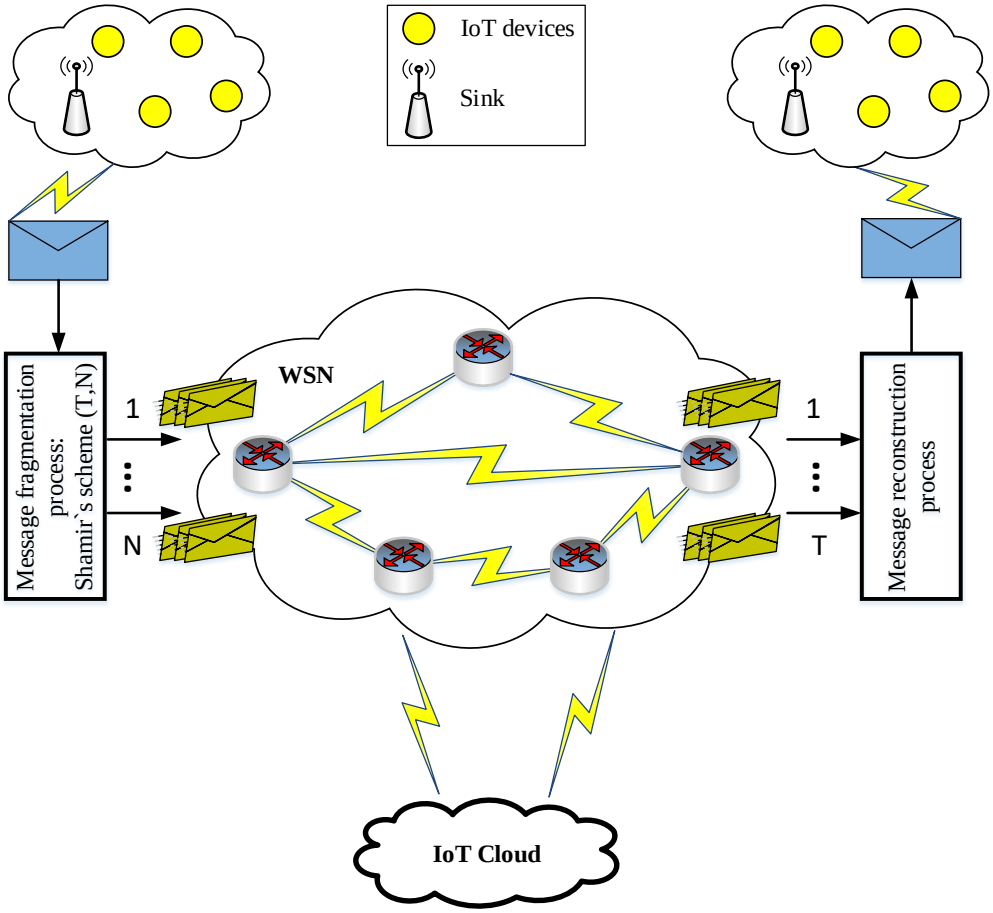


Fig. 3.1. Message fragmentation according to Shamir`s scheme [26]

Thus, the actual problem seems related to the improvement of the mathematical model of secure routing message transmitted over the network based on the optimal allocation of its fragments over non-overlapping paths resulting from the use of applying Shamir`s scheme, and comprehensive address to the security parameters of available paths.

3.2. Secure Routing Model

In information security and IoT in particular, when implementing security measures, both passive and active approaches are used. While the proactive means are used at the stage of preventing the compromise of transmitted messages or minimizing the probability of its occurrence, the reactive means are used in those cases when the security of the transmitted data is violated and, for example, it is important for the means of routing to quickly restore the required level of security. When it comes to security, it is largely known that its solutions are usually based on quality of the mathematical models and methods for calculating the desired paths, so therefore in this work, theoretical solutions presented by mathematical models and methods of secure routing and combine the capabilities of both proactive and reactive approaches when implementing secure routing of messages in the network.

Within the model let it be assumed that the following inputs are known [26, 29]:

- n – number of links in the network;
- m – number of nodes in the network;
- S_{msg} – sender of a transmitted message (source node);
- D_{msg} – receiver of a transmitted message (destination node);
- M – number of used non-overlapping paths in routing message fragments;
- (T, N) – Shamir`s scheme parameters;

- N – total number of fragments, obtained by applying the Shamir`s scheme;
- T – minimum number of fragments ($T \le N$) needed for the message reconstruction;
- p_i^j – probability of compromise j-th element (node, link) of i-th path;
- M_i – number of elements in the i-th path that can be compromised.

During the solving of the secure routing problem, the following parameters should be calculated:

- p_i – probability of compromise the i-th path;
- order of distribution of the number of fragments of the transmitted message by paths taking into account the selected Shamir`s scheme (T, N);
- x_i – number of fragments, transmitted over the i-th path ($i = \overline{1, M}$);
- P_{msg} – probability of compromise for the whole message during its transmission by fragments over the network.

The number of paths used in the network (M) determines the size of the vector $\vec{x}$, the coordinates x_i of which characterize the number of fragments transmitted in the i-th path between the sending node and the receiving node. Based on the physical meaning of the variables x_i, they are subject to restrictions of the form:

$$x_i \in \mathrm{N}_0 \ (i = \overline{1, M}) \tag{3.1}$$

where N_0 is the extended natural number, i.e. variables x_i can take only non-negative integer values.

Besides, during the calculation of the control variables x_i ($i = \overline{1, M}$) regulating the allocation of the message fragments over the non-overlapping paths the following condition [26, 29] must be met:

$$N = \sum_{i=1}^{M} x_i \,. \tag{3.2}$$

It is assumed that the sender and the receiver are trusted, i.e. probability of compromise of the sender and receiver nodes is equal to zero. Furthermore, within the proposed solution (as in [26]), it is supposed that if the element (node, link) is compromised, all fragments transmitted through the element will also be compromised. Then the probability of compromise of the *i*-th path consisting of the M_i elements can be calculated by the expression [29]

$$p_i = 1-(1-p_i^1)(1-p_i^2)\ldots(1-p_i^{M_i}) = 1-\prod_{j=1}^{M_i}(1-p_i^j)\,. \tag{3.3}$$

In the case of Shamir`s scheme with redundancy when $T < N$ the condition below must be satisfied

$$N - x_i < T\,,\ (i = \overline{1,M}\,). \tag{3.4}$$

while when $T = N$ the following conditions must be met in the non-redundant sharing scheme

$$1 \le x_i \le T-1,\ (i = \overline{1,M}\,). \tag{3.5}$$

Conditions (3.5) and (3.6) ensure that in the case of compromising all the paths except *i*-th path an adversary cannot reconstruct the whole message. While the probability of message compromise divided into the *N* fragments using Shamir`s scheme transmitted over the *M* paths determined by the expression [29]

$$P_{msg} = \prod_{i=1}^{M} p_i \,. \tag{3.6}$$

The solution to the problem of secure routing is determining the order of distribution of the number of fragments of the message to be transmitted, along paths that do not overlap. In turn, this task can be formalized as a Mixed Integer Linear Programming (MILP) problem, which was solved by the MATLAB Optimization Toolbox, presented by the `intlinprog` subroutine [29-32].

When solving MILP problems, it is necessary to minimize the objective function represented by the linear form

$$\min_x \vec{f}^{\,t}\vec{x}\,,$$

when a number of conditions are met, which are presented in the form of constraints on equations and inequalities

$$A\cdot\vec{x}\le\vec{b}\,;\ Aeq\cdot\vec{x}=\vec{b}eq\,;\ \overrightarrow{lb}\le\vec{x}\le\overrightarrow{ub}\,,$$

where $\vec{f}$, $\vec{x}$, $\vec{b}$, $\vec{b}eq$ are vectors;

A and *Aeq* are matrices of the corresponding size;

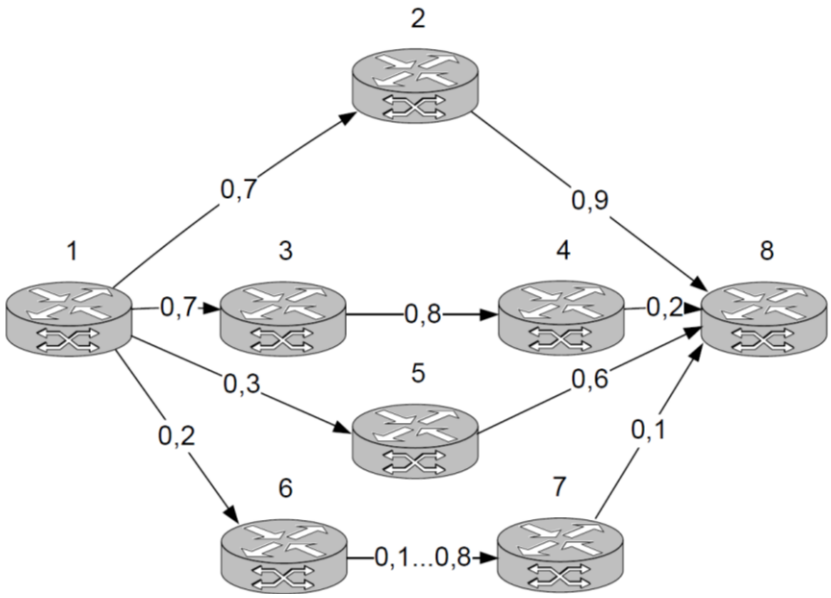
$\overrightarrow{lb}$ and $\overrightarrow{ub}$ are vectors-columns of size *M*.

3.3. Numerical example of describing and solving a secure routing problem in a MATLAB environment

Let the structure of the network and the probabilities of compromise of its

communication links be presented in Fig. 3.2. Then the total number of nodes in the network is eight (m = 8), and the number of links is ten (n = 10). Let also the source node of packets be the node 1, and the destination node is the node 8. The total number of message fragments N = 10. Let the number of used non-overlapping paths, according to the network structure, be equal to four (M = 4) that are:

- 1→2→3;
- 1→3→4→8;
- 1→5→8;
- 1→6→7→8.



Fug. 3.2. The WSN fragment for secure transmission of the message under investigation [29]

The investigated parameters of the Shamir`s scheme (T, N): without redundancy (11, 11) at $T = N$ = 11, with redundancy (10, 11) at T = 10, N = 11.

Then we form the desired vector $\vec{x}$. Within the model represented by expressions (3.1)-(3.4), it has the form:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \tag{3.7}$$

The size of the metric vector $\vec{f}$ corresponds to the number of paths used in the network *M*, the coordinates f_i of which characterize the probability of compromising the *i*-th path (3.3). Then the vector $\vec{f}$ for example (3.7) has the form:

$$\vec{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}. \tag{3.8}$$

Then we formalize the condition of the integrity of the message, consisting of *N* fragments:

$$x_1 + x_2 + x_3 + x_4 = 10. \tag{3.9}$$

Next in accordance with (3.8) vectors *Aeq* and $\overrightarrow{beq}$ are as follows:

$$Aeq = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}; \ \overrightarrow{beq} = 10. \tag{3.10}$$

The text of the program is edited in the M-file Editor window in the MATLAB environment. For the example described above, the source codes are following.

```
% Results of calculations using the Shamir's scheme without
redundancy (11, 11)
clc;
clear all;
N=11;
T=11;
% probabilities of compromise of network links
p_link=[0.3;0.2;0.2;0.3;0.2;0.4;0.3;0.2;0.1;0.3];
% calculation the probability of compromise the i-th path
p_path(1)=1-(1-p_link(1))*(1-p_link(2));
p_path(2)=1-(1-p_link(3))*(1-p_link(4))*(1-p_link(5));
p_path(3)=1-(1-p_link(6))*(1-p_link(7));
p_path(4)=1-(1-p_link(8))*(1-p_link(9))*(1-p_link(10));
Aeq=ones(1,4);
beq=N;
lb=ones(4,1);
ub=(N-1)*ones(4,1);
f=p_path;
intcon=[1 2 3 4];
[x,fval]=intlinprog(f,intcon,[],[],Aeq,beq,lb,ub)
% calculation the probability of message compromise
p_msg=p_path(1)*p_path(2)*p_path(3)*p_path(4)
p_path
```

Based on the results of the obtained calculations, the order of distribution of the number of message fragments by paths was determined, namely:

- eight fragments are transmitted by the first paths;
- one fragment is transmitted by the second path;
- one fragment is transmitted by the third path;
- one fragment is transmitted by the fourth path.

Taking into account the given probabilities of compromise of communication links of the modeled network, the values of probabilities of compromise of the non-overlapping paths are received:

- $1 \rightarrow 2 \rightarrow 3$ – p_1=0.44;
- $1 \rightarrow 3 \rightarrow 4 \rightarrow 8$ – p_2=0.552;

- 1→5→8 – p_3=0.58;
- 1→6→7→8 – p_4=0.496.

The probability of compromising the message as a whole, which is equal to $P_{msg} = 0.0699$, was also determined.

Source code for the Shamir`s scheme with redundancy:

```
% Results of calculations using the Shamir's scheme with
redundancy (10, 11)
clc;
clear all;
N=11;
T=10;
% probabilities of compromise of network links
p_link=[0.3;0.2;0.2;0.3;0.2;0.4;0.3;0.2;0.1;0.3];
% calculation the probability of compromise the i-th path
p_path(1)=1-(1-p_link(1))*(1-p_link(2));
p_path(2)=1-(1-p_link(3))*(1-p_link(4))*(1-p_link(5));
p_path(3)=1-(1-p_link(6))*(1-p_link(7));
p_path(4)=1-(1-p_link(8))*(1-p_link(9))*(1-p_link(10));
Aeq=ones(1,4);
beq=N;
% conditions of maximum security under Shamir`s scheme with
redundancy
lb=(N-T+1)*ones(4,1);
ub=N*ones(4,1);
f=p_path;
intcon=[1 2 3 4];
[x,fval]=intlinprog(f,intcon,[],[],Aeq,beq,lb,ub)
% calculation the probability of message compromise
p_msg=p_path(1)*p_path(2)*p_path(3)*p_path(4)
p_path
```

While the order of distribution of the number of message fragments by paths under Shamir`s scheme with redundancy was as shown below:

- five fragments are transmitted by the first paths;

- two fragments are transmitted by the second path;
- two fragments are transmitted by the third path;
- two fragments are transmitted by the fourth path.

Graph of the dependence of the probability of compromising the message as a whole on the changing compromise probability, for example, the ninth link (fourth path) is presented in Fig. 3.3.

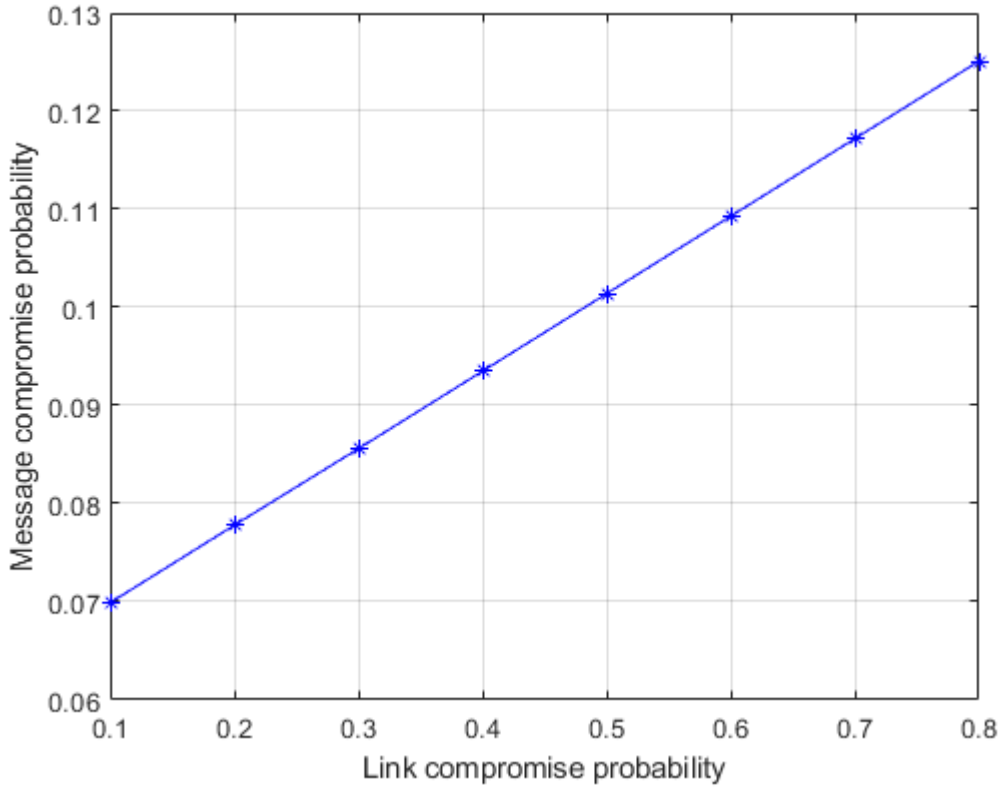


Fig. 3.3. Dependence of the probability of compromising the message on the changing compromise probability the ninth link

CHAPTER 4

MODELLING IOT ROUTING PROTOCOLS WITH MATLAB

Wireless Sensor Network (WSN) comprises a large number of tiny sensor nodes. These nodes are intelligent of sensing and monitoring the environmental or physical condition like temperature, sound, pressure, motion, etc. and communicating with other nodes. Sensor nodes in WSN have limited power and energy constraint hence it becomes necessary to efficiently use these resources. Energy consumption is the major problem of WSN to overcome this problem there are many solutions introduced, it consists flat, hierarchical and location based routing. The basic hierarchical routing protocol for WSN is Low Energy Adaptive Clustering Hierarchy (LEACH). LEACH works in two different phases viz. set-up and steady-state phase. LEACH periodically do the rotations among cluster-head nodes in such a way that every node gets a chance to become cluster head and distributes energy consumption between the nodes in the network which reduces the power utilization, increasing the lifetime of the network. In this chapter the working of LEACH is discussed using MATLAB [31-37].

4.1. Low Energy Adaptive Clustering Hierarchy (LEACH) Protocol

It is a cluster-based protocol [31, 32], which includes distributed cluster arrangement. In this technique it selects some sensor nodes as a cluster head (CH) and distributes energy to all other sensor nodes in the network. The clusters are formed based on applications and parameter of devices. Each cluster consists of CH, the function of this in LEACH is to collect the data from different nodes and to do the compression of data. Finally, it sends the aggregated data to base station.

To overcome intra and inter-cluster collusion in network, the LEACH uses MAC, CDMA and TDMA techniques. The collection of data in leach technique is centralised

and will be performed at regular intervals. This protocol is well suitable for when there is required for continuous monitoring of the network. By implementing this observer may not need to monitor all the data instantly and regularly. Since, the constrained energy of the sensor devices may drain due to period transmission.

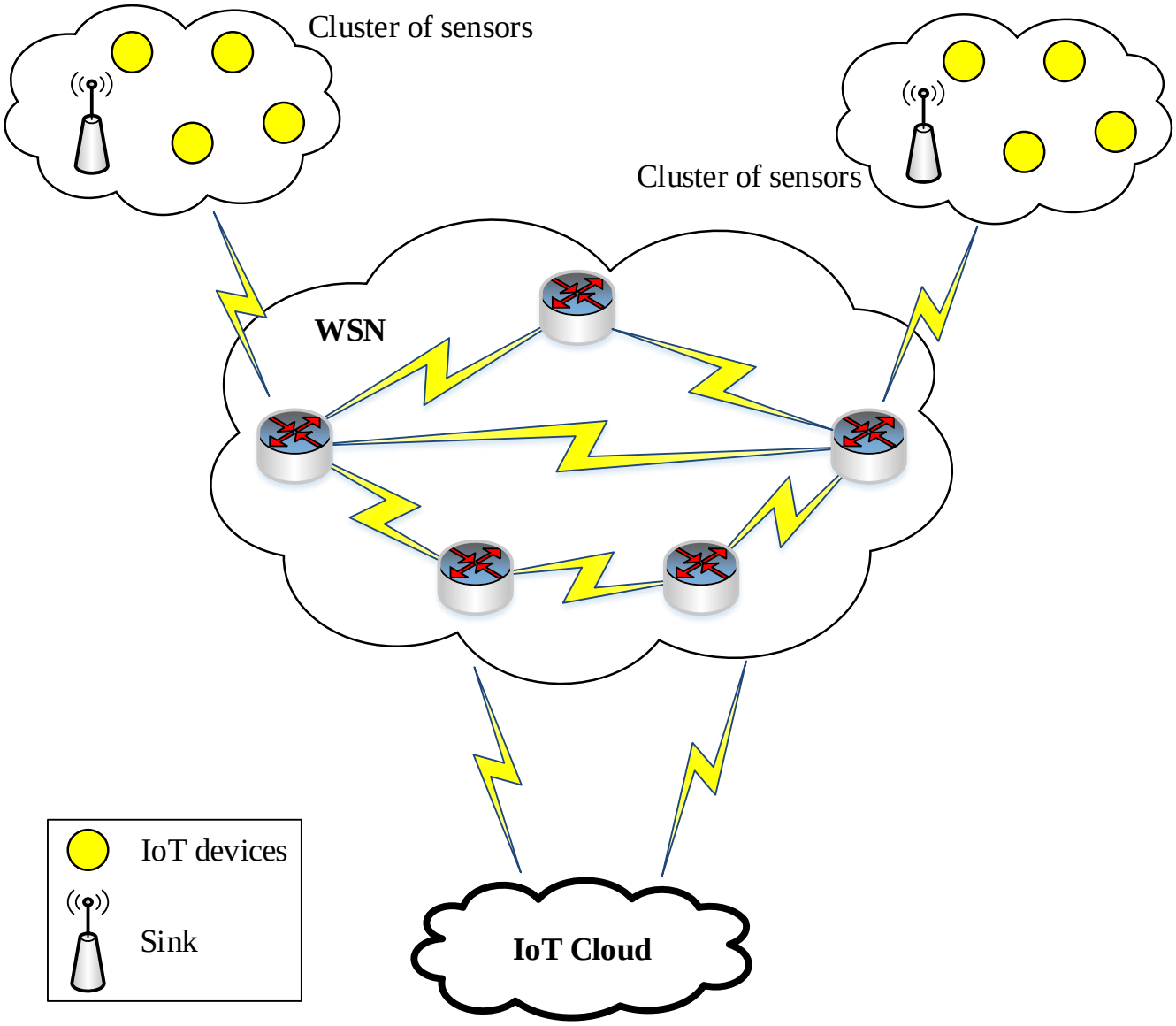


Fig. 4.1. Wireless connectivity of IoT devices to IoT cloud

LEACH [31, 32] uses hierarchical routing. LEACH provides data fusion, the basic significance of LEACH protocol is better utilization power consumption of sensor nodes

and maximizing the life expectancy of the network. Nodes in the network assign roles to themselves as one of them is assigned as cluster head and others as leaf nodes also known as non-cluster heads. Leaf nodes are responsible for sensing data. Sensed data from leaf nodes is collected at a cluster head, signal processing functions are performed. The processed data is aggregated and then send to the sink node/base station. All the leaf nodes of the cluster consume less power than the cluster head nodes because they do not have to perform any signal processing function on data. Leaf nodes losses communication when the cluster head nodes die. The main emphasis of LEACH is on reducing the power consumption as well as to increase the network lifetime. LEACH minimizes the energy which is consumed by the cluster head nodes and the leaf nodes of a cluster as much as 8 times when compared with other routing techniques. LEACH periodically do the rotations among cluster-head nodes in such a way that each node in the network will be assigned as cluster-head to avoid drainage of any particular sensor node. In this way, the power consumption related to cluster head will be uniformly distributed within all nodes. As cluster head nodes identify all the cluster members which are belonging to that cluster, it will create a time division multiple access [2] (TDMA) schedule will help nodes to decide when the data is transmitted to the cluster head. There are no intra-cluster collisions in LEACH because it uses TDMA scheduling. LEACH operates in two phases viz.

1. Set-up phase.
2. Steady-state phase.

4.2. MATLAB Investigation of the LEACH Protocol

MATLAB source code [35]:

```
clear;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Field Dimensions - x and y maximum (in meters)
xm = 100;
```

```
ym = 100;
%x and y Coordinates of the Sink
%sink.x =0.5 * xm;
%sink.y = ym + 50;
sink.x=50;
sink.y=175;
%sink.x=0.5*xm;
%sink.y=0.5*ym;
%Number of Nodes in the field
n = 100
%Optimal Election Probability of a node to become
cluster head
p=0.05;
packetLength =6400;
ctrPacketLength = 200;
%Energy Model (all values in Joules)
%Initial Energy
Eo = 0.5;
%Eelec=Etx=Erx
ETX=50*0.000000001;
ERX=50*0.000000001;
%Transmit Amplifier types
Efs=10*0.000000000001;
Emp=0.0013*0.000000000001;
%Data Aggregation Energy
EDA=5*0.000000001;
INFINITY = 999999999999999;
%maximum number of rounds
rmax=9999
%%%%%%%%%%%%%%%%%%%%%%%%% END OF PARAMETERS
%%%%%%%%%%%%%%%%%%%%%%%%%
%Computation of do
do=sqrt(Efs/Emp);
%Creation of the random Sensor Network
figure(1);
for i=1:1:n
    S(i).xd=rand(1,1)*xm;%Чш±к
    XR(i)=S(i).xd;
    S(i).yd=rand(1,1)*ym;
    YR(i)=S(i).yd;
    S(i).G=0;
    %initially there are no cluster heads only nodes
```

```
    S(i).type='N';
    S(i).E=Eo;
    S(i).ENERGY=0;
    % hold on;
end
S(n+1).xd=sink.x;
S(n+1).yd=sink.y;

%First Iteration
figure(1);
%counter for CHs
countCHs=0;
%counter for CHs per round
rcountCHs=0;
cluster=1;
countCHs;
rcountCHs=rcountCHs+countCHs;
flag_first_dead=0;
for r=0:1:rmax
  r
  %Operation for epoch
  if(mod(r, round(1/p))==0)
     for i=1:1:n
        S(i).G=0;
        S(i).cl=0;
     end
  end
hold off;
%Number of dead nodes
dead=0;
%counter for bit transmitted to Bases Station and to
Cluster Heads
packets_TO_BS=0;
packets_TO_CH=0;
%counter for bit transmitted to Bases Station and to
Cluster Heads per round
PACKETS_TO_CH(r+1)=0;
PACKETS_TO_BS(r+1)=0;
figure(1);
for i=1:1:n
    %checking if there is a dead node
     if (S(i).E<=0)
```

```
        dead=dead+1;
      end

      if (S(i).E>0)
         S(i).type='N';
      end
end
if (dead == n)
   break;
end
STATISTICS(r+1).DEAD=dead;
DEAD(r+1)=dead;
%When the first node dies
if (dead==1)
    if(flag_first_dead==0)
        first_dead=r
        flag_first_dead=1;
    end
end
countCHs=0;
cluster=1;
for i=1:1:n
   if(S(i).E>0)
     temp_rand=rand;
     if ((S(i).G)<=0)
        %Election of Cluster Heads
        if(temp_rand <=(p/(1-p*mod(r,round(1/p)))))
            countCHs = countCHs+1;

            S(i).type = 'C';
            S(i).G = round(1/p)-1;
            C(cluster).xd = S(i).xd;
            C(cluster).yd = S(i).yd;

            distance=sqrt((S(i).xd-(S(n+1).xd))^2 +
(S(i).yd-(S(n+1).yd))^2);

            C(cluster).distance = distance;
            C(cluster).id = i;
            X(cluster)=S(i).xd;
            Y(cluster)=S(i).yd;
            cluster=cluster+1;
```

```
            distanceBroad = sqrt(xm*xm+ym*ym);
            if (distanceBroad >=do)
                S(i).E = S(i).E-(ETX*ctrPacketLength +
Emp*ctrPacketLength*(distanceBroad*distanceBroad*distan
ceBroad*distanceBroad));
            else
                S(i).E = S(i).E-(ETX*ctrPacketLength +
Efs*ctrPacketLength*(distanceBroad*distanceBroad));
            end

            distance;
            if(distance>=do)
                 S(i).E = S(i).E-
((ETX+EDA)*packetLength+
Emp*packetLength*(distance*distance*distance*distance
));
            else
                 S(i).E = S(i).E-
((ETX+EDA)*packetLength+
Efs*packetLength*(distance*distance));
            end
            packets_TO_BS = packets_TO_BS+1;
            PACKETS_TO_BS(r+1) = packets_TO_BS;
        end
     end
   end
end
STATISTICS(r+1).CLUSTERHEADS = cluster-1;
CLUSTERHS(r+1)= cluster-1;
%Election of Associated Cluster Head for Normal Nodes
for i=1:1:n
   if (S(i).type=='N' && S(i).E>0)
    % min_dis = sqrt( (S(i).xd-S(n+1).xd)^2 + (S(i).yd-
S(n+1).yd)^2 );
     min_dis = INFINITY;
     if(cluster-1>=1)
         min_dis_cluster = 1;

         for c = 1:1:cluster-1
            %temp = min(min_dis,sqrt( (S(i).xd -
C(c).xd)^2 + (S(i).yd - C(c).yd)^2 ) );
```

```
            temp = sqrt((S(i).xd - C(c).xd)^2 +
(S(i).yd - C(c).yd)^2);
            if (temp<min_dis)
                min_dis = temp;
                min_dis_cluster = c;
            end
            S(i).E = S(i).E - ETX * ctrPacketLength;
        end

        %Energy dissipated by associated Cluster
        min_dis;
        if (min_dis > do)
            S(i).E = S(i).E - (ETX*(ctrPacketLength) +
Emp * ctrPacketLength*( min_dis * min_dis * min_dis *
min_dis));
            S(i).E = S(i).E - (ETX*(packetLength) +
Emp*packetLength*( min_dis * min_dis * min_dis *
min_dis)); %
        else
            S(i).E = S(i).E -(ETX*(ctrPacketLength) +
Efs*ctrPacketLength*( min_dis * min_dis));
            S(i).E = S(i).E -(ETX*(packetLength) +
Efs*packetLength*( min_dis * min_dis));
        end
        S(i).E = S(i).E - ETX*(ctrPacketLength);

        %Energy dissipated
        if(min_dis > 0)
            S(C(min_dis_cluster).id).E =
S(C(min_dis_cluster).id).E - ((ERX + EDA)*packetLength
);
            S(C(min_dis_cluster).id).E =
S(C(min_dis_cluster).id).E - ERX *ctrPacketLength ;
            if (min_dis > do)
                S(C(min_dis_cluster).id).E =
S(C(min_dis_cluster).id).E - ( ETX*(ctrPacketLength) +
Emp * ctrPacketLength*( min_dis * min_dis * min_dis *
min_dis));
            else
                S(C(min_dis_cluster).id).E =
S(C(min_dis_cluster).id).E - ( ETX*(ctrPacketLength) +
Efs * ctrPacketLength*( min_dis * min_dis));
```

```
            end
          PACKETS_TO_CH(r+1) = n - dead - cluster + 1;
        end

        S(i).min_dis = min_dis;
        S(i).min_dis_cluster = min_dis_cluster;

    end
  end
end
%hold on;
countCHs;
rcountCHs = rcountCHs + countCHs;
end
x=1:1:r;
y=1:1:r;
%z=1:1:r;
for i=1:1:r;
    x(i)=i;
    y(i) = n - STATISTICS(i).DEAD;
    %z(i)=CLUSTERHS(i);
end
%plot(x,y,'r',x,z,'b');
plot(x,y,'r');
hold on;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   STATISTICS
%  DEAD  : a rmax x 1 array of number of dead
nodes/round
%  DEAD_A : a rmax x 1 array of number of dead Advanced
nodes/round
%  DEAD_N : a rmax x 1 array of number of dead Normal
nodes/round
%  CLUSTERHS : a rmax x 1 array of number of Cluster
Heads/round
%  PACKETS_TO_BS : a rmax x 1 array of number packets
send to Base Station/round
%  PACKETS_TO_CH : a rmax x 1 array of number of
packets send to ClusterHeads/round
%  first_dead: the round where the first node died
%
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%ctrPacketLength=200,packetLength=4000,Eo=2J.
```

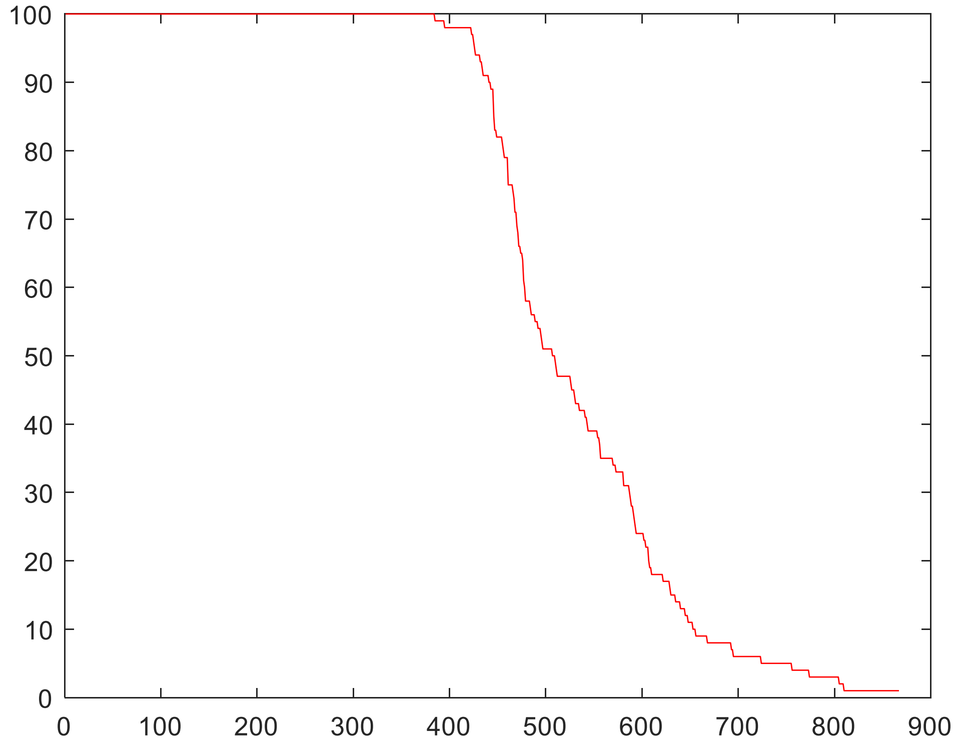


Fig. 4.1. Results of investigation

CONCLUSION

The Internet of Things will continue to grow in both scope and scale and so will its adoption. It is projected that the effect of IoT will permeate every sector of the global society, hence the global task is to ensure that the great advantages this new wave brings shouldn't be turned around and used for malicious or illegal activities.

Hence, this study titled: *Models and methods of secure routing in IoT* aims to shed light on the issues and probable solutions routing has to offer as a means of securing the IoT system. As we have come to realize the backbone of an efficient IoT system is as a result of its routing technique in the sense that if they is a robust and secure routing mechanism, IoT and its embedded devices will function and operate seamlessly and efficiently.

This thesis is divided into four chapters, the first chapter sheds light on some of the different IoT architectures available detailing how those architectures are designed and structured to provide security and also hot they fall short in certain circumstances. Examples of the Architectures covered were: three-layer architecture, the five-layer architecture, the fog architecture and the Hybrid cloud and fog architecture but in all of the aforementioned architectures, they must be underlying conditions and requirements that should be met, such as: Concurrent Data, Efficient Data Handling, Connectivity and Communications, Security and Availability and Quality of Service etc.

A reference model for IoT was also discussed, this reference model formed by the IoT World Forum (just like that of the popular OSI reference model for internet serves the same purpose whereby a standardized structure will be followed to allow for compatibility and connectivity between the various devices that makes up the IoT system.

The second chapter deals with the secure routing in IoT firstly highlighting the factors influencing routing such as: communication process, addressing mechanisms etc. furthermore Routing challenges such as: Scalability, Partitions and Void and Limited resources all which acts as detriments to the full functionality of routing were considered.

Some secure routing protocols and routing protocols limiting factors were also mentioned and discussed. Lastly the concept of trust based protocols was introduced in which the idea is to provide security to sensor networks without using cryptographic approach.

The third chapter dealt with the model of secure routing and its investigation explaining the fact that traditional methods of information security will be ineffective when applied to IoT whilst introducing the concept of Spread and Shamir`s scheme with the aim of providing a more viable solution to issues surrounding secure routing. Then, numerical research and performance analysis were done using MATLAB to ascertain secure routing model for a single path that do not overlap.

Lastly the final chapter focuses on modelling routing protocols for IoT with a focal point on Low Energy Adaptive Clustering Hierarchy (LEACH) which is a preferred routing technology used when dealing with wireless sensor networks. This is due to the fact that LEACH protocol continuously monitors the network and any anomaly detected will be dropped. Hence provided a standardized scheme for ensuring authenticity of packets coming into the network.

REFERENCES

1. Godson Agbara E. Security framework for IoT devices. XIV International Scientific Conference "Modern Challenges in Telecommunications" MCT-2020. Conference proceedings. Kyiv. Igor Sikorsky Kyiv Polytechnic Institute, 2020. p. 382.
2. Godson Agbara E. Modern IoT Architectures, their characteristics and applications analysis. 74-а науково-технічна конференція професорсько-викладацького складу, науковців, аспірантів та студентів. Матеріали конференції. Одеса. Одеська національна академія зв'язку ім. О.С. Попова, 12-14 грудня 2019. С. 58.
3. Персіков М.А., Лемешко В.О., Егбе Годсон А. Аналіз сучасного стану протоколів маршрутизації в мережах IoT. Інформатика, управління та штучний інтелект. Тези сьомої міжнародної науково-технічної конференції. Харків: НТУ "ХПІ", 2020. С. 59.
4. Chopra K., Gupta K., Lambora A. Future Internet: The Internet of Things-A Literature Review. Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon) 2019: Proceedings of the International Conference. Faridabad, India, 14-16 Feb., 2019. IEEE, 2019. P. 135-139. DOI: https://doi.org/10.1109/COMITCon.2019.8862269
5. Patel K.K., Patel S.M. Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. International journal of engineering science and computing, 2016. Vol. 6, Iss. 5. P. 6122-6131.
6. Stallings W., 2015. Foundations of modern networking: SDN, NFV, QoE, IoT, and Cloud. Addison-Wesley Professional. 560 p.
7. Rak J., 2015. Resilient routing in communication networks. Berlin: Springer. 181 p.
8. Sobin C.C., 2020. A Survey on Architecture, Protocols and Challenges in IoT. Wireless Personal Communications, P. 1383–1429. DOI: https://doi.org/10.1007/s11277-020-07108-5
9. Calihman A. Architectures in the IoT Civilization. 2019. URL: https://www.netburner.com/learn/architectural-frameworks-in-the-IoT-civilization/

10. Fremantle P. A reference architecture for the internet of things. WSO2 White paper, 2015. URL: https://wso2.com/whitepapers/a-reference-architecture-for-the-internet-of-things/#21
11. Hanes D., Salgueiro G., Grossetete P., Barton R., Henry J., 2017. IoT fundamentals: Networking technologies, protocols, and use cases for the internet of things. Cisco Press. 577 p.
12. Flauzac O., González C., Hachani A., Nolot F. SDN based architecture for IoT and improvement of the security. In: 2015 ieee 29th international conference on advanced information networking and applications workshops. IEEE (2015). P. 688-693. DOI: 10.1109/WAINA.2015.110
13. Sharma V., You I., Andersson K., Palmieri F., Rehmani M.H., Lim J., 2020. Security, privacy and trust for smart mobile-Internet of Things (M-IoT): A survey. IEEE Access, 8. P.167123-167163.
14. Chandrashekhara B.G., Veena K.N. Routing in Internet of Things: Review. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 790-795, DOI: https://doi.org/10.1109/ICCONS.2018.8663232.
15. Pokharel R., Agarwal S., Khatri A., Singhal S., 2018. A Survey on Secure Routing Protocols Based on Trust Management in IoT. International Journal of Advanced Studies of Scientific Research, 3(12). P. 312-316.
16. Marietta J., Mohan B.C., 2020. A review on routing in internet of things. Wireless Personal Communications, 111(1). P. 209-233. DOI: https://doi.org/10.1007/s11277-019-06853-6
17. Chze, P.L.R. and Leong, K.S., 2014, March. A secure multi-hop routing for IoT communication. In 2014 IEEE World forum on internet of things (WF-IoT) (pp. 428-432). IEEE.

18. Basabi, A.E., He, J. and Hashemi, S.M., 2016, October. Secure routing in IoT with multi-objective simulated annealing. In 2016 2nd IEEE international conference on computer and communications (ICCC). IEEE. P. 2073-2076
19. AlMansour, N. and Alahmadi, S., 2018, April. Secure Ad Hoc On-Demand Distance Vector Routing Protocol in WSN. In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS). IEEE. P. 1-4.
20. Raoof, A., Matrawy, A. and Lung, C.H., 2020. Enhancing Routing Security in IoT: Performance Evaluation of RPL Secure Mode under Attacks. arXiv preprint arXiv:2004.07815.
21. Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C., 2017. SCOTRES: secure routing for IoT and CPS. IEEE Internet of Things Journal, 4(6), P. 2129-2141.
22. Lokulwar, P.P. and Deshmukh, H.R., 2017, February. Threat analysis and attacks modelling in routing towards IoT. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE. P. 721-726.
23. Karlsson, J., Dooley, L.S. and Pulkkis, G., 2018, August. Secure routing for MANET connected Internet of Things systems. In 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE. P. 114-119.
24. Nowak, M., Nowak, S., Domańska, J. and Czachórski, T., 2019, June. Cognitive packet networks for the secure internet of things. In 2019 Global IoT Summit (GIoTS). IEEE. P. 1-4.
25. El Hajjar, A., Roussos, G. and Paterson, M., 2017, June. Secure routing in IoT networks with SISLOF. In 2017 Global Internet of Things Summit (GIoTS). IEEE. P. 1-6
26. Yeremenko O.S., Ali S.A., 2015. Secure Multipath Routing Algorithm with Optimal Balancing Message Fragments in MANET. Radioelectronics and Informatics. 1(68). P. 26-29.
27. Yeremenko O., Yevdokymenko M., Persikov A., 2017. Flow-aware approach of evaluating probability of compromise in combined structure network. Advanced

Information and Communication Technologies (AICT): Proceedings of the 2nd International Conference, Lviv, Ukraine, 4-7 July, 2017. IEEE. P. 258-261.

28. Yeremenko O., Lemeshko O., Persikov A., 2018. Secure Routing in Reliable Networks: Proactive and Reactive Approach. Advances in Intelligent Systems and Computing II, CSIT 2017, Advances in Intelligent Systems and Computing, Springer, Cham. Vol. 689. P. 631–655.
29. Лемешко О.В., Невзорова О.С., Єременко О.С., Євсєєва О.Ю. Методичні вказівки до практичних занять з дисципліни «Управління та маршрутизація в ТКС» для студентів денної форми навчання спеціальності 6.050903 – Телекомунікації. Харків: ХНУРЕ, 2016. 64 с.
30. MATLAB Programming Fundamentals. The MathWorks, Inc. 1720 p.
31. Agarwal, K., Agarwal, K. and Muruganandam, K., 2018, September. Low energy adaptive clustering hierarchy (leach) protocol: Simulation and analysis using matlab. In 2018 International Conference on Computing, Power and Communication Technologies (GUCON). IEEE. P. 60-64.
32. Birajdar, D.M. and Solapure, S.S., 2017, March. LEACH: An energy efficient routing protocol using Omnet++ for Wireless Sensor Network. In 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE. P. 465-470.
33. Kotagi, V.J., Singh, F. and Murthy, C.S.R., 2017, May. Adaptive load balanced routing in heterogeneous IoT networks. In 2017 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE. P. 589-594.
34. http://www.codeforge.com/article/201004
35. Venu Madhav (2020). wireless networks (https://www.mathworks.com/matlabcentral/fileexchange/25853-wireless-networks), MATLAB Central File Exchange. Retrieved December 27, 2020. https://www.mathworks.com/matlabcentral/fileexchange/25853-wireless-networks