KARADENİZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

CONTEXT-AWARE CACHING IN WIRELESS IOT NETWORKS

MASTER THESIS

AKHTARI ZAMEEL

JANUARY 2020
TRABZON

KARADENİZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING

CONTEXT-AWARE CACHING IN WIRELESS IOT NETWORKS

Akhtari ZAMEEL

This thesis is accepted to give the degree of
"MASTER OF SCIENCE"
By
The Graduate School of Natural and Applied Sciences at
Karadeniz Technical University

The Date of Submission : 23 / 12 / 2019
The Date of Examination : 22 / 01 / 2020

Supervisor : Asst. Prof. Dr. Sedat GÖRMÜŞ

Trabzon 2020

KARADENİZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING
Akhtari ZAMEEL

CONTEXT-AWARE CACHING IN WIRELESS IOT NETWORKS

Has been accepted as a thesis of
MASTER OF SCIENCE
after the Examination by the Jury Assigned by the Administrative Board of the Graduate School of Natural and Applied Sciences with the Decision Number 1834 dated 31 / 12 / 2019

Approved By

Chairman	:	Assoc. Prof. Dr. Tolga OVATMAN	....
Member	:	Asst. Prof. Dr. Sedat GÖRMÜŞ	....
Member	:	Assoc. Prof. Dr. Güzin ULUTAŞ	....

Prof. Dr. Asim KADIOĞLU
Director of Graduate School

ACKNOWLEDGMENT

This thesis is submitted for completing Master Program to the Karadeniz Technical University, Institute of Science. First of all, I would like to give my special thanks to my respected thesis supervisor, Asst. Prof. Dr. Sedat GÖRMÜŞ. He always guided me with his valuable discussions, comments and suggestions throughout the realization of this thesis. This thesis I could not able to complete without his support.

I would like to express my deep gratitude to Turks Abroad and Related Communities (YTB) for giving me the scholarship to study in Turkey. I would like to express my deep gratitude to Mustafa Najmuldeen, Masters student and a good friend, for keeping me on the track when I got lost and supporting me during my thesis work.

I would like to give my most deep thanks to my husband and my family who gave me so much love, support, encouragement during my entire studies. And finally, I would like to thank my friends for inspiring me at the time of my frustration regarding thesis completion.

Akhtari ZAMEEL
Trabzon 2020

THESIS STATEMENT

I declare that the research work in this thesis entitled “Context-Aware Caching in IoT for Wireless Network” is a record of an authentic research work carried out by me, Akhtari ZAMEEL under the guidance of my supervisor Asst. Prof. Dr. Sedat GÖRMÜŞ. All data used in this master thesis are obtained by simulation and experiment work, which I obtained by following all research rules. For the award of any degree or diploma, neither this thesis nor any part of it has been submitted to any other University of Institute, except where due reference acknowledgement has been given in the text. 22/01/2020

Akhtari ZAMEEL

CONTENTS

Page No

Master Thesis

SUMMARY

CONTEXT-AWARE CACHING IN WIRELESS IoT NETWORKS

Akhtari ZAMEEL

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Asst. Prof. Dr. Sedat GORMUS
2020, 59 Pages

Wireless communication has been increasingly utilized to connect a plethora of services over the Internet backbone. Caching is considered to be a way forward to efficiently deliver contents over the Internet. However, with the introduction of the Internet of Things (IoT) concept, new requirements emerge. In contrast to traditional Internet traffic, IoT data are ordinarily transient and occasionally refreshed by the producer. Furthermore, IoT devices can be resource-constrained with limitations in terms of energy, storage, and processing capabilities. In this thesis, we propose a new context-aware caching approach for IoT networks which pro-actively learn the caching need in a particular instant and modify the cached data to improve the cache hit rate. This approach can enable application to achieve a better user experience since the user will not have to wait for the data to be retrieved from a central entity.

Keywords: Caching, cache content placement, ARIMA, context-aware, wireless networks.

Yüksek Lisans Tezi

ÖZET

KABLOSUZ IOT AĞLARINDA İÇERİĞE DUYARLI ÖNBELLEKLEME

Akhtari ZAMEEL

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Dr. Öğr. Üye. Sedat GÖRMÜŞ
2020, 59 Sayfa

Kablosuz haberleşme, İnternet omurgası üzerinden çok sayıda hizmet bağlamak için giderek daha fazla kullanılmaktadır. Önbelleğe alma, içerikleri Internet üzerinden verimli bir şekilde sunmak için bir yol olarak kabul edilir. Ancak, Nesnelerin İnterneti (IoT) kavramının tanıtılmasıyla birlikte yeni gereksinimler ortaya çıkmaktadır. Geleneksel İnternet trafiğinin aksine, IoT verileri geçicidir ve bazen üretici tarafından yenilenir. Ayrıca IoT cihazları, enerji, depolama ve işlem yetenekleri bakımından kısıtlı kaynaklara sahip olabilir. Bu çalışmada, IoT ağlarında belirli bir anda önbellekleme ihtiyacını proaktif olarak öğrenen ve önbellek isabet oranını iyileştirmek için önbellek verilerini değiştiren yeni bir önbellekleme yaklaşımı sunuyoruz. Bu yaklaşım, kullanıcının merkezi bir yapıdan veri almasını beklemediğinden uygulamaya daha iyi bir kullanıcı deneyimi sunar.

Anahtar Kelimeler: Önbelleğe alma, önbellek içeriği yerleştirme, ARIMA, içeriğe duyarlı, kablosuz ağlar.

LIST OF FIGURES

Page No

LIST OF TABLES

Page No

ABBREVIATIONS

IoT	Internet of Things
WSN	Wireless Sensor Network
RFID	Radio Frequency Identification
IoE	Internet of Everything
IBM	International Business Machines Corporation
ITU	International Telecommunication Union
ICT	Information and Communication Technology
OSI	Open System Interconnection
RPL	Routing Protocol for LLN
IP	Internet Protocol
DODAG	Destination-Oriented Directed-Acyclic Graphs
DIO	DODAG Information Object
DIS	DODAG Information Solicitation
DAO	DODAG Destination Advertisement Object
DAO-AKC	DAO Acknowledgment
LLN	Low-power Lossy Network
6LOWPAN	IPv6 over low power wireless personal area network
IPv6	Internet Protocol version 6
IETF	Internet Engineering Task Force
UDP	User Datagram Protocol
MAC	Message Authentication Code
6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4
ETSI	The European Telecommunications Standards Institute
TSCH	Time Slotted Channel Hopping
CoAP	Constrained Application Protocol
ICN	Information Centric Network
CCN	Content Centric Network
CR	Content Router

CS	Content Store
PIT	Pending Interest Table
FIB	Forwarding Information Base
LFU	Least Frequently Used
LRU	Least Recently Used
ARIMA	Auto Regressive Integrated Moving Average
AR	Autoregressive
MA	Moving Average
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion

1. INTRODUCTION

1.1. Thesis Motivation and Goal

In the early 1970s, the Internet emerged with the simple objective of linking two devices situated in different geographical locations. With the advent of technology over time, the vision has broadened to make things of the physical world "smart” and link every smart thing globally over the Internet backbone which is called the Internet of Things. The ultimate vision of such technologies is to enable instant sharing of information on the web. This led to the new requirements such as enormous growth of data consumption, high internet traffic, slow loading time, the ability for storage and processing in the low powered devices of IoT Networks. Despite recent developments in wireless mobile radio networks, these networks formed by low power and low throughput devices cannot keep up with the huge increase in mobile data traffic [1].

The traditional solution to this increasing demand for wireless data consumption is content caching or mirroring in different locations in the network. Caching is regarded to be an appropriate approach to increase the efficiency of content delivery in order to better control the expanding demand for information ferried over the Internet. Caching also allow to reduce disk storage cost. Generally, the large proportion of data traffic accounts for a limited number of extremely popular content. Due to these facts, caching is envisioned to increase wireless content delivery performance [2].

Again, caching can be a way forward to ensure an improvement in the quality of service perceived by the user. A fundamental limitation of caching is that there is a huge amount of information available in the Internet and it is not possible to place all this information in cache entity when the cache size of the caching entity is limited. Intelligent approaches are therefore necessary for the placement of cache data.

The main objective of this study is to exploit the properties of low-powered IoT devices, embed caching in the growing wireless IoT network. Another goal is designing an extremely stable cache methodology that could allow a joint solution to the problem of wireless content delivery and limited storage of IoT devices.

1.2. Work Phase and Methodology

This thesis is devoted to the proactive content caching in IoT Networks. As IoT devices have limitations in terms of storage, the main challenge is to efficiently utilize these low storage devices while optimizing the content retrieval time. This challenge motivates us to come up with an innovative solution to this limitation. To address these issues, a novel approach named “context-aware caching” has been proposed here which proactively learns the user behavior in a specific IoT context to cache user data probabilistically in a particular instant. Due to storage limitations in such low power devices, it is only feasible to cache content depending on the context. This approach can enable proactive content caching where the cache decision and cache replacement strategy is made after the context of the users is learned by observing the usage patterns of IoT services.

1.3. Structure of The Thesis

This thesis report is made up of six chapters. The reporting of this thesis begins with a brief discussion on the thesis motivation and goal. Then this chapter includes the work phase and methodology used in this thesis to implement this proposed algorithm. Finally, this chapter concludes the structure of this thesis.

In chapter 2, general context of this thesis is given which is start with the introduction of the Internet of Things (IoT). Then the reference model and three-layer structure of IoT are explained. Chapter 2 especially describes the important technologies and protocols of IoT. Here, some real applications of IoT are also discussed. This chapter concludes with a brief discussion on motivation and goal, work phase and methodology, and structure of this thesis.

In chapter 3, the fundamental theory of caching including strategy and techniques are discussed. A brief discussion on the Information-Centric network, its architecture and working procedure are given in section 3.1.2. A review of the caching approaches is discussed in section 3.3.

Chapter 4 elaborates on the system model which begins with the description of the proposed system in section 4.1 and continues describing the designed system in section 4.2, implementation in section 4.3 and proposed algorithm in section 4.4.

Chapter 5 discusses the performance evaluation of the proposed system where simulation setups are mentioned and results are analyzed

Chapter 6 summarizes the whole research including limitations of this study. Further development of this research discussed in section 6.2.

2. TECHNOLOGIES AND PROTOCOLS OF IOT

2.1. Internet of Things (IoT)

The wireless communication is one of the major innovations that have changed human life. It has increasingly been used to connect a plethora of services through a set of standard protocols and connect different heterogeneous networks-academic, industry, government, etc. In the last decades, the technology of wireless communication advance very fast. It has been introduced as a part of many different aspects of our lives providing variety of services and applications.

Today, it has become indispensable to use communication equipment to make life easier for living in societies. Most of these communication devices operate independently of one another. These communication devices interact with each other and generate a smart communication environment thanks to the Internet of Things (IoT) technology.

The term “Internet of Things (IoT)” is new as well as an old one. Weiser presented "Ubiquitous Computing" back in 1988. He proposed the following types of omnipresent computer systems that could provide end-user services at any time or location: tab, pads, and panels. Since then, a great deal has changed in terms of computational strength and computing device integrity. "Ubiquitous Computing" changes the emphasis in almost every "thing" around us today are interconnected and able to exchange information [3]. The concept of the Internet of Things was first used by Kevin Ashton in 1999 in a presentation on the benefits of Radio Frequency Identification (RFID) technology for the company Proctor & Gamble [4]. To connect the concept of radio frequency identification (RFID) with the then-new Internet subject he used the term. However, in 1991 [5] at Cambridge University, the first application of the Internet of Things in history was the sharing of images of a coffee machine and a camera system on the internet by a group of academicians. This system was used until August 22, 2001 [6].

Moving the focus towards today, defining the word IoT can be a bit difficult, as there are many definitions depending on who is defining it [7]. The fundamental concept of the Internet of Things is to connect a wide range of things, where things include everything from sensors to smart devices, which are associated with the Internet. Cisco describes IoT as the idea of connecting more and more things to the Internet to facilitate the daily life of people. Moreover, as we connect more things, there will be a rise in the need for IPv6, big data, and cloud computing, and the theory of IoT will be transformed into an Internet of everything [8].

The definition of IoT given by IBM is split into two parts: "One is to be more efficient, be less destructive, to connect different aspects of life which do affect each other in more conscious, deliberate and intelligent ways. But the other is also to generate fundamentally new insights, new activity, new forms of social relations" [9].

International Telecommunication Union (ITU) is a United Nations agency that specializes in ICT (Information and Communication Technology) proposed a definition of IoT: "… a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies"[10].

2.1.1. Reference Model

A comprehensive reference model for IoT has provided by ITU. For the definition of the IoT Architectural Reference Model, the IoT Reference Model presents the highest level of abstraction. Like the internet OSI reference model, the architecture of IoT is composed of six layers: four horizontal layers and two vertical layers. The four horizontal layers are application layer, service support and application support layer, network layer and the device layer. Each of these layers is associated with vertical layers: management and security capabilities [8].

At the bottom of the stack, it has the device layer which also includes gateway capabilities by providing various interfaces to support linked devices through various kinds of wired and wireless systems. This layer contains the physical things necessary for sensing or controlling the physical world and acquiring data. In some circumstances, conversion of the protocol is required to promote communication between devices using the distinct device and network layer protocols [10].

The network layer contains networking and transport capabilities. The networking capabilities delivers appropriate network connectivity control features such as authentication, authorization, and accounting. And the transport capabilities focus on transportation of IoT services and applications along with the transportation of information related to IoT control and management [10].

Above the network layer, it has the service support and application support layer which consists of the generic support and specific support capabilities. The generic support capabilities are common capabilities i.e. information processing or data storage, that various IoT applications can use. The specific support capabilities are particular capabilities that meet the needs of skills to support diversified applications.

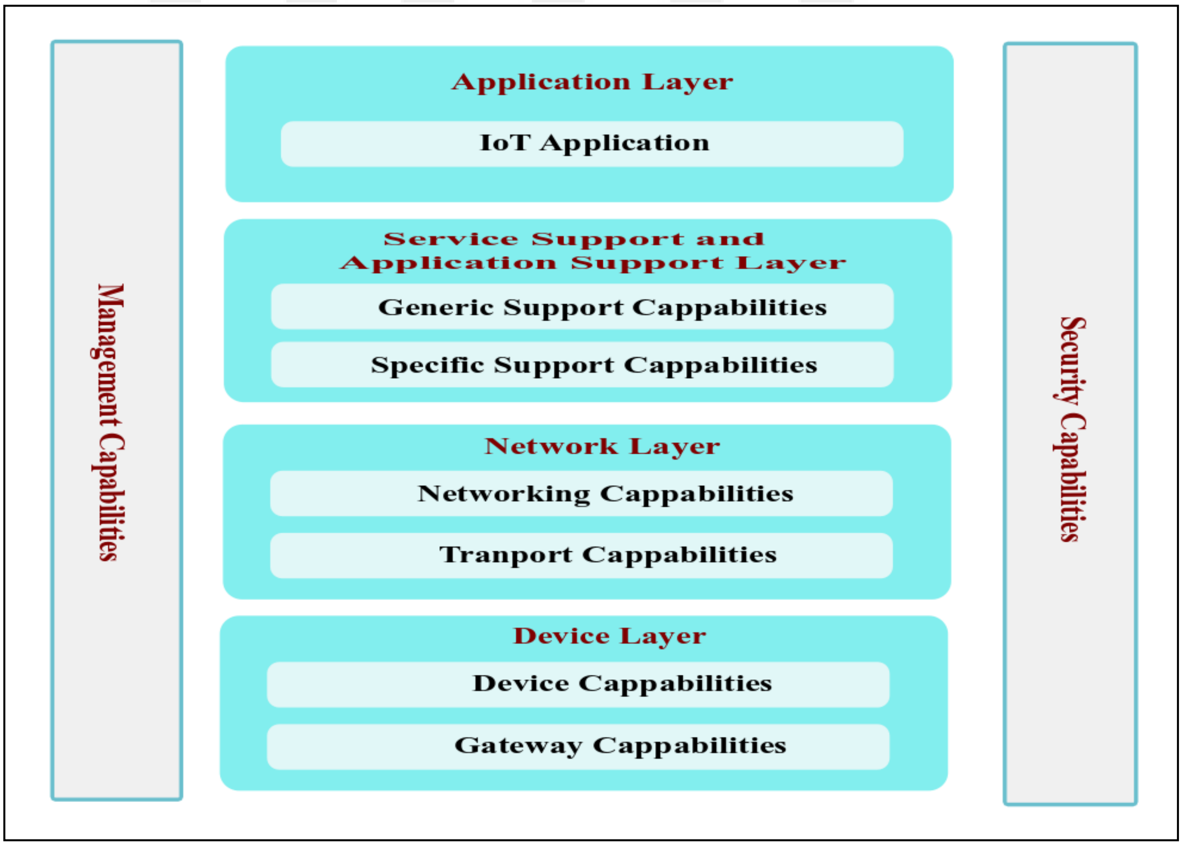

Figure 1. IoT reference model recommended by ITU

At the top of the stack, it has the application layer which contains IoT applications. To function these IoT applications require certain support capabilities from the underlying layer.

Both the management and security capabilities layer cover generic and specific capabilities. The generic management capabilities deal with device management, local network management, and traffic and congestion management. Being independent of the applications the generic security capabilities include authorization, authentication, application data confidentiality, and integrity protection, privacy protection, security audit, anti-virus and etc.

With application-explicit requirements, both the specific management and security capabilities are closely coupled.

2.1.2. Structure of IoT

IoT has three issues [9] : comprehensive sensing, knowledgeable processing and efficient communication. Therefore, structure of IoT mainly consists of three-layer: sensing, network/transport, and application layer.

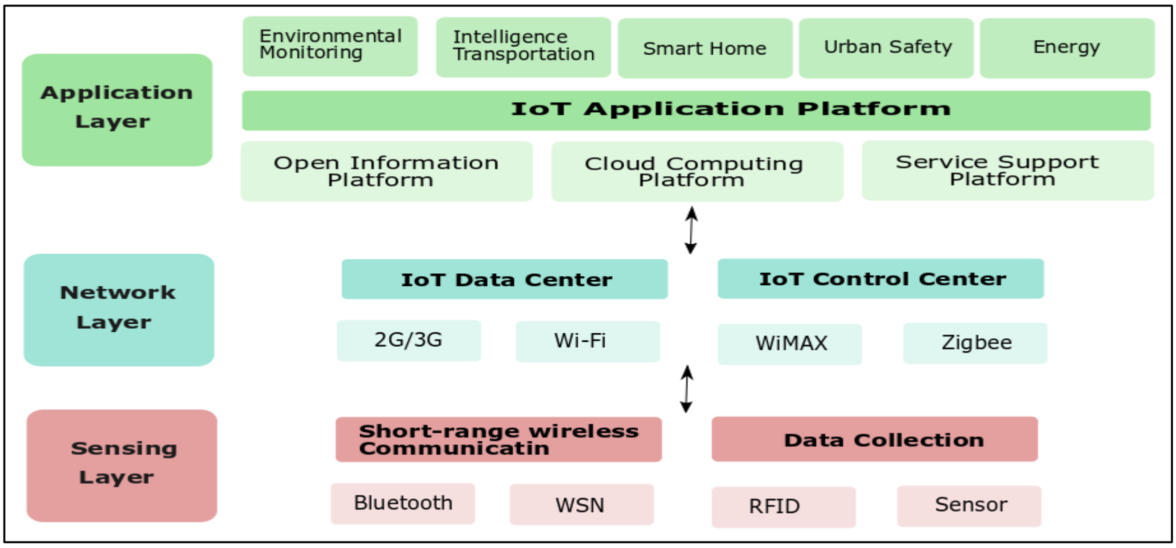

Figure 2. Structure of IoT [11].

- Sensing Layer:

In IoT, the most fundamental and core layer is the sensing layer. The layer of sensing comprises a set of Internet-enabled devices capable of perceiving, detecting objects, gathering information, and exchanging information via Internet communication networks with other devices. This layer mainly has two parts: basic sensors and sensor network. Basic sensors such as RFID tags and readers, cameras, GPS, different sensors, two-dimensional code labels, and readers, as well as other fundamental identification and sensor elements. The primary technology is engaged in this layer are RFID technology, sensing and control technology, short-range wireless communication technology. There are also sensors, sensor nodes, electronic tags, wireless gateways and wireless routers in the main products. Principally the sensing layer solves the issue of information gathering from these devices. For data forwarding sensor nodes use wireless communication technology.

- Network Layer:

To a certain extent, the role of the network layer is sensing and transmitting data from the perception layer to the application layer under the limitations of the capacities of devices, network limitations and the limitations of the applications. By using separate access devices, IoT will connect to the mobile communication network and the Internet as it builds on the current network of mobile communications and the Internet.

Since the cloud computing system could analyze and store large data, it is an essential part of the network layer. IoT solutions use a combination of short-range communications networks like Bluetooth and ZigBee to transfer information from perception devices to a local gateway based on communication parties ' capabilities [12].

- Application Layer:

The application layer is the last implementation layer. The main purpose is to address the issue of information processing and human-machine communication. It is also called the processing layer.

From the network layer, it processes the data then transfers the data to all forms of information system. Basically, the information will be exchanged through other machines with humans.

The cloud computing platform in the network layer is also the basis of several applications in the application layer. It also includes various platforms, such as system support platform, network management platform, information security platform, information processing platform, etc.

In this layer, automation and virtualization are the core technologies. Automation technology can manage user requests and automatically arrange the resource [11]. In addition, virtualization technology can significantly enhance resource use to lower costs.

Since applications are aimed at creating smart homes, smart cities, distributed power storage coordination, power system control, and renewable energy generator integration and so on. Therefore, better strategies for power distribution and management are able to design [13].

2.1.3. Application Area

IoT's technical architecture allows many new applications to be developed and implemented to enhance the quality of every part of our life. We are addressing IoT-enabled applications in this segment.

- Smart Home:

To provide comfortable living, IoT is regarded in the form of remote control and management of various home appliances by its owners through smartphones, tablets or connected computers. For example, the doors of garage and home can be controlled automatically by RFID communication, room lighting varies in accordance to the day time, the air condition of the rooms is automatically controlled in accordance with our preferences, use an automated watering system to keep plants alive in the garden, etc. The survey of smart home or office is discussed in [14], [15] and [16] from which some of these technologies are already aimed at realizing to the public.

- Smart Cities:

In correlation to cities, IoT will allow better systems that effectively gather and process information by enabling real-time monitoring and control of various city services. Sensors can be used to control city services monitoring centers for traffic lights, street lights [17], car parks, water, gas, power grid, and camera surveillance. These systems offer the potential to improve the transport systems, personal and social safety.

- Smart Health:

In the healthcare industry, the Internet of Things will have several applications with the aim of preventing severe emergencies and monitoring patients with chronic diseases remotely and acting swiftly if needed. In-person visits are reduced by communication via smart devices and patients can handle their treatment from home. The IoT enables healthcare professionals to monitor parameters such as temperature, heart rate, sleep, physical activity, blood sugar, and blood pressure [18]. A concise overview of several monitoring systems already deployed based on the “The Internet of Medical Things" is provided in [19] The IoT is quickly enhancing modern medicine, alleviating the stress on both doctor and patient.

- Industry:

Another region that the IoT will revolutionize is the industrial sector. Operational optimizations, productivity-boosting, resource savings, and cost reduction are typically the primary objectives of industrial IoT solutions. In industrial manufacturing systems, the amount of RFID tags, wireless sensor devices, and integrated controllers is increasing considerably. For instance, this technology enables enterprises to keep track of company assets, enhance environmental safety, and preserve consistency and quality in a manufacturing system and to provide a profitable after-sales service.

- Military Application:

The military services may become more effective and efficient by incorporating sensors, actuator and control systems into existing military infrastructure and services. The linked sensors and digital analytics offered by IoT technology can be used in terms of military implementation for monitoring enemy force, supplies and machinery from their origin to where they are required in the field of defense. A military force appears to use specialized networking and communications infrastructure. Cloud infrastructures can be used by transmitting to the Internet, which can give operational flexibility [20].

2.2. Wireless Communication Technologies in IoT

Communication is the most important component of achieving IoT because they need to be able to communicate in order to interconnect different IoT devices. Fast technological developments in wireless communication frameworks allow multi-functional low powered sensor devices to recognize and respond to adjustments in their surroundings. Among several technologies, we represent the technologies that are either playing major role in the IoT field or closely connected to our research.

2.2.1. Radio-Frequency Identification

Among all the technologies Radio-Frequency Identification (RFID) technology is one of the most commonly implemented technology. To offer users a better solution, it serves as an information carrier and helps users collect such data in an efficient way.

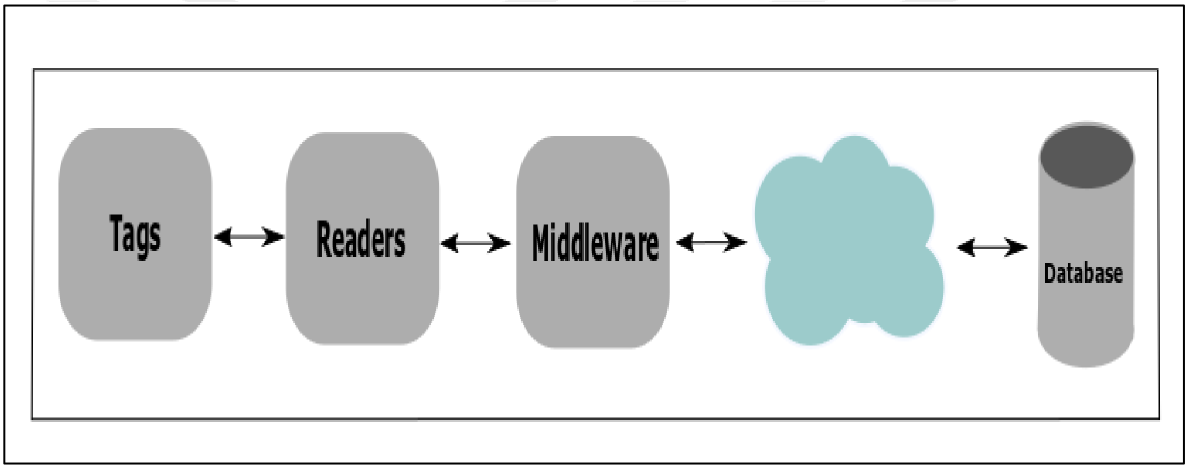

Figure 3. RFID system architecture.

As shown in Figure 3, RFID system consists of several parts: RFID tag, reader, middleware, network, and database. Tag or transponder carries the information. RFID reader or interrogator will identify the tag information and then transfer it to the middleware section. The information sent from the reader is processed and managed in the middleware section. Afterwards all the information will be transferred to the databases through the network gateway so that all the information can be stored there.

Generally, RFID tags are made of two fundamental elements: a built-in circuit to store and process information, modulate and demodulate a radio frequency (RF) signal, and alternatively to extract energy from the RF signal of a reader, an antenna where RF signals are received and transmitted [21].

An RFID system's operational theory works as follows. Through transmitting a query signal, the RFID reader activates data transfers. By sending their stored information such as EPC code multiple RFID tags will respond to the reader. Two primary methods are used to prevent collisions. In a slotted approach based on ALOHA protocol [22], the query signal contains a parameter used by RFID tags to extend their response spontaneously. The adaptive binary tree methodology [23], the RFID reader sequentially broadcasts a sequence of bits 0 and 1 every time a bit is added to the string. It enables a reader to discover all RFID tags within their range of transmission effectively by only allowing those RFID tags to response whose ID matched with the bit sequence.

According to the type of tags, RFID systems are classified as passive and active. The reader activates the transmission of passive RFID through the produced RF signal. Active RFID systems have their own on-board power source to activate its transmission. Active tags are larger in size compared to passive tags, have higher production costs, can be read from a wider range up to 100 meters, and are therefore reserved for expensive products such as cars, transponder linked to an aircraft [24], containers. Passive tags are read from short-range and mostly included in less expensive items like in retail industry. However, RFID has risks in information safety measures as RFID chips cannot be encrypted.

2.2.2. Wireless Personal Area Networks (WPAN)

Wireless Personal Area Networks (WPAN) is a type of network inter-connective devices by using short-distance wireless communication. It is driven over low cost, low powered consumption, short communication range wireless network of IoT protocols such as Bluetooth, wireless USB, ZigBee, etc. For effective and efficient interaction between things, systematized physical signal processing such as modulation, coding and wireless media access such as scheduling approaches are necessary because IoT devices consist of a variety of hardware.

- Bluetooth

Bluetooth uses short-range communication such as WPAN which requires low power and low cost. It is also known as a piconet, several Bluetooth devices are connected together in a master-slave relationship to build a star topology network. In the piconet, the master is the first Bluetooth device, and every single other device is slaves interacting with the master. Only the master or any of the slaves can transmit at each time slot at a specific frequency selected according to the technique of adaptive frequency hopping (AFH) [21]. In the same geographical region, the interaction between the simultaneous transmissions of different Piconets is reduced by AFH. Scatternet is an interconnection between two or more piconets. Bluetooth technology is applied in devices such as keyboards, printers, headsets, a mouse that connects to computers or cell phones.

- ZigBee

Zigbee is a standard based on IEEE 802.15.4 for a set of high-level communication protocols that are used for building personal area networks (PAN) [25]. Zigbee is designed for low data transfer, low power consumption. To extend the range and increase network accessibility ZigBee often supports meshed network architectures. It is commonly used in low data rate applications but requires longer battery life and reliable communication.

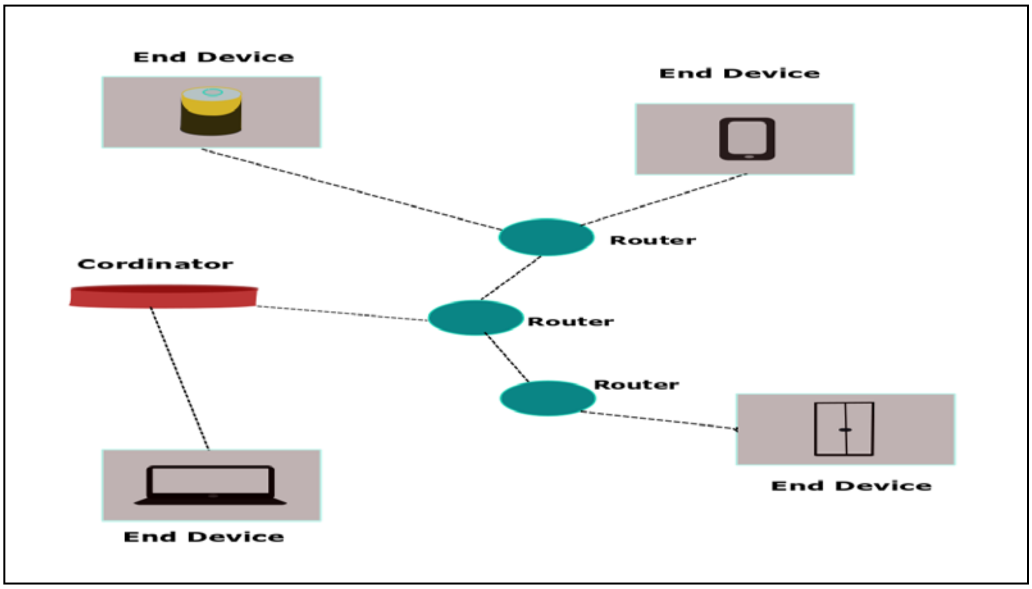

Figure 4. ZigBee network architecture

Typically, the ZigBee network is composed of three-node groups. They are ZigBee coordinator, ZigBee Router, ZigBee End Device [26]. ZigBee coordinator is the root of the network which is a bridge of the network. It is the responsibility of the supervisor to manage and store the information when transmitting and receiving data. Each Zigbee network has only one coordinator. Zigbee routers serve as intermediary devices allowing data to travel to and from other devices through them. Also, gets involved in messages multi-hop routing. End devices have minimal functionality which allows interacting only with parent nodes. This approach helps to improve battery life as the node can sleep for a considerable amount of time. It is less expensive as it takes less memory. ZigBee technology is applied in home automation, health care applications, telecom service as ZigBee SIM [27] and more on.

2.2.3. Wireless Sensor Networks (WSN)

In the early stage of the IoT developments, wireless sensor network is the core building foundation of IoT. Additionally, WSN is a technology that really helps users to accomplish the true significance of IoT. In the IoT, the fundamental thought of WSN is to link sensing and network layer. The purpose of wireless sensor network is to transform the physical data into a digital signal and then transmit data to the sink or root node for further data processing.

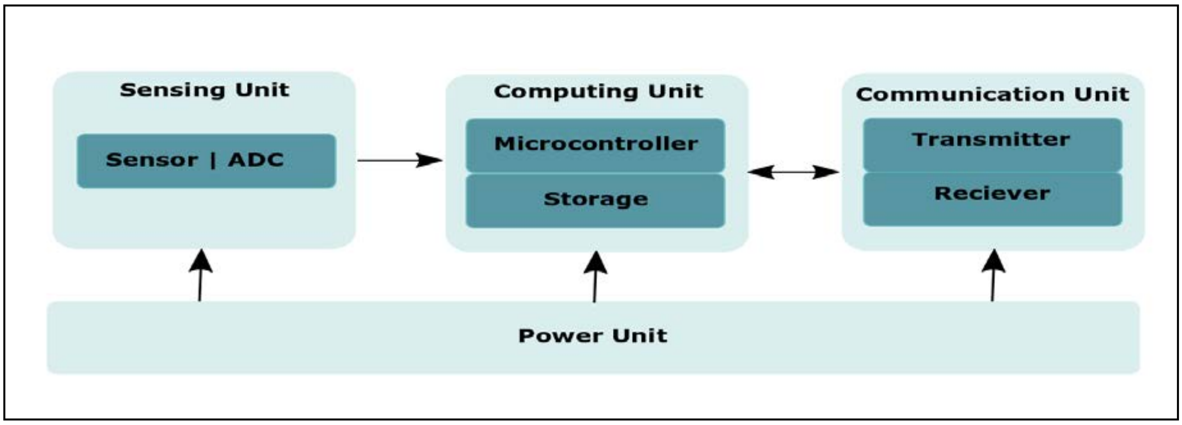

Figure 5. Block diagram of Wireless Sensor Network Node [28].

Figure 5. represents the structure of the wireless sensor network. The wireless sensor network is usually comprising of four units: sensing, computing, communication, and power unit.

Sensing sub-system is made up of sensors and an ADC converter. The sensor device implements necessary functions to identify and convert environmental parameters to analog signals. Then the ADC samples and converts the analog signal to digital signal.

The computing unit conducts different functions. Other sensor units are also controlled by it. The microcontroller in the computing unit is in charge of receiving and processing the sensing unit's analog or digital data.

The communication system enables the computing unit to transmit sensor data through wired or wireless medium to the central servers located in the backbone network. The communication unit (radio) is used for transmitting and receiving data packets.

The power unit provides the required power to the other unit i.e. sensing unit, computing unit, and the communication unit. The battery power of the sensor nodes must be utilized efficiently as its limited.

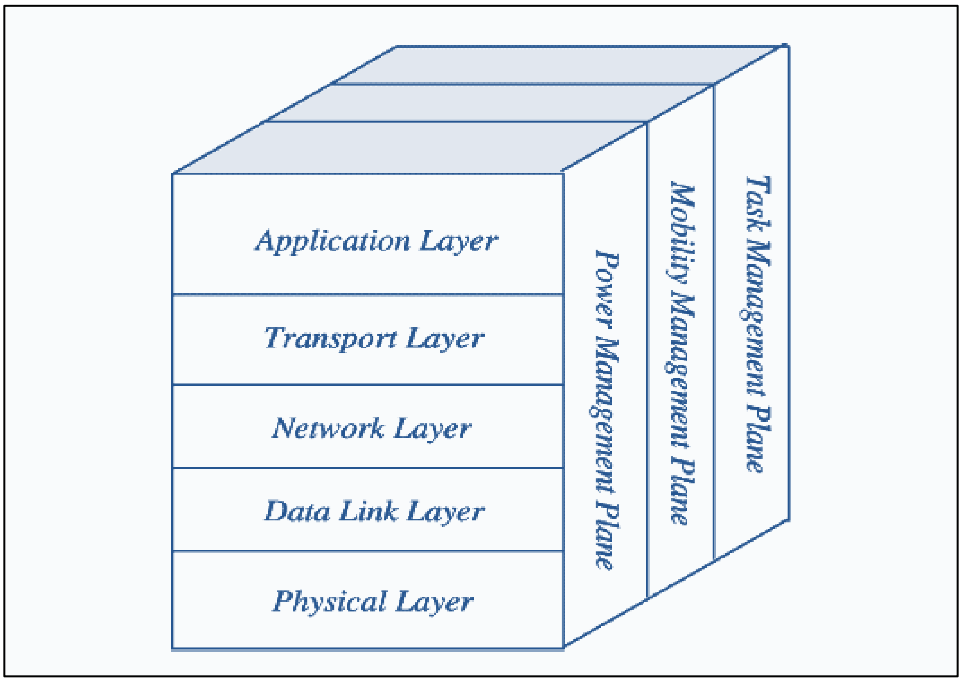

Figure 6. Architecture of wireless sensor network.

The architecture of wireless sensor networks is based upon the well-known OSI model. This architecture has five main layers. Three cross layers are also added to it as shown in Figure 6. The three cross layers are; power management plane, mobility management plane and task management plane [29] [30]. These layers are utilized to monitor the network and make the sensors cooperate to improve the overall network performance. The five layers in the architecture are- physical layer, data link layer, network layer, transport layer, and application layer.

- Physical layer

The lowest layer in the protocol stack is the physical layer which facilitates modulation, frequency selection, radio technology and manages transmission[29]. IEEE 802.15.4 [31]: Offered as the communication standard of the low-cost personal area and low-cost WSN. The standard facilitates power management to put the radio on the duty cycle to extend battery lifetime. It also ensures low complexity, low power consumption. Extended versions of IEEE 802.15.4 adapted for different application scenarios are available.

- Data Link Layer

The Data Link Layer is responsible for multiplexing data streams, data frame identification, Medium Access Control (MAC), and error control [29]. It also enables the nodes to be able to communicate in a point–point and point– multipoint manner in an efficient way.

- Network Layer

The primary function of Network Layer is to route the information provided by the transport layer towards the intended receiver. It has a lot of application-based tasks, but the main responsibilities are to be power conservation, limited memory, buffers, and self-organized sensor because a unique global ID for each sensor may not be provided. The basic concept of the routing protocol is to create a secure and redundant path based on a metric scale that is unique from the protocol to the protocol [30].

- Transport Layer

The two main objectives of the transport layer are to ensure reliability and prevent congestion [29] [30]. Specifically, this layer is required when assembling a framework to access other external or internal networks. TCP is not ideal for WSN since it requires end-to-end confirmation of data delivery via three-way handshaking mechanism. This mechanism requires a large protocol overhead which may be detrimental to sensors with limited energy sources.

- Application Layer:

The application layer is accountable for traffic management and provides software to find positive information for numerous applications that convert the data in a clear context [29]. The application layer contains different protocols that provide multiple services to the sensor network.

With the advent of many significant WSN technologies, there have been increased efforts to incorporate WSNs with the Internet. Internet ready WSNs can be used in several new environments. Such Internet enabled IoT application can be applied, for instance, in military surveillance, transportation, environmental, health care, smart home, entertainment, and other business areas [32].

2.3. Emerging IoT Protocols

Due to the presence of a number of networks, applications, and devices in an IoT setting, several standards are being used, involving many standardizations [33]. In this section, the IoT routing protocols and standards are presented as two sublayers: Routing layer managing packet transmission from source to destination i.e. RPL, and encapsulation layer which structures the packets i.e. 6LOWPAN, 6TiSCH [30].

2.3.1. RPL

Routing protocol for low-power and lossy networks (RPL) is an energy-efficient protocol specifically designed for routing in the IoT networks. It is a distance-vector protocol based on destination-oriented directed-acyclic graphs (DODAG). In DODAG, the nodes are organized as a tree topology with one or more roots serving as sinks. Each of the leaf nodes has a rank and from each leaf there is only one path to the root as shown in figure 7.

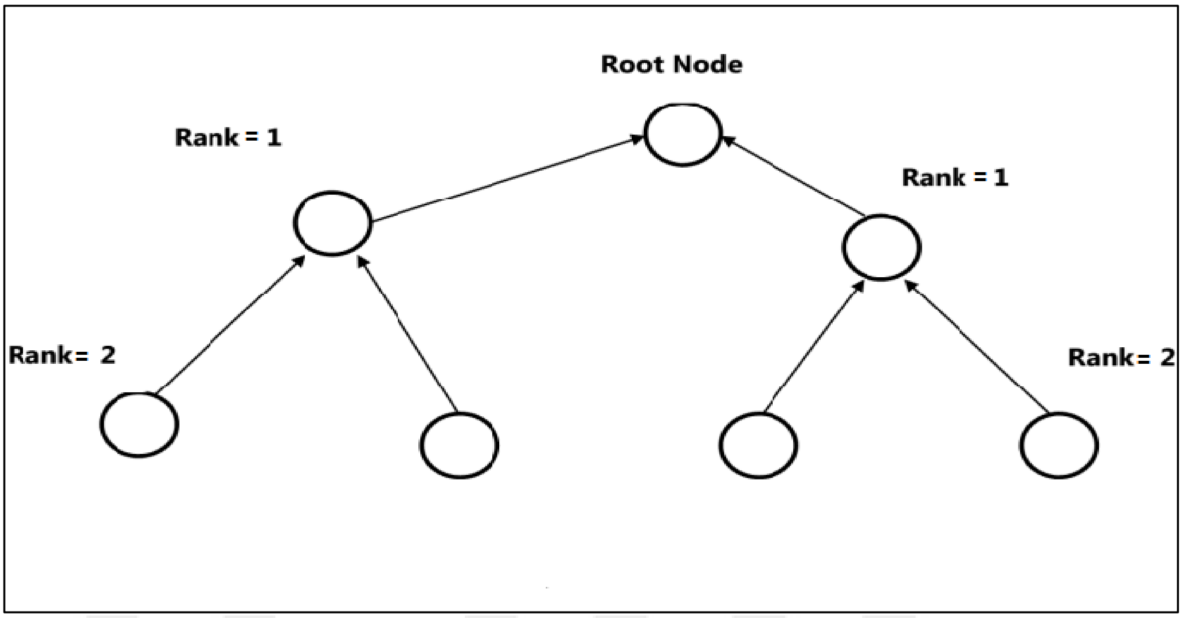

Figure 7. RPL DODAG.

To maintain the RPL topology, there are three different types of control packets in RPL. They are DIO - DODAG Information Object, DIS - DODAG Information Solicitation, DAO - DODAG Destination Advertisement Object. To form and maintain the DODAG, each node sends a DIO and propagate their ranks on the network [34]. DIO messages contain DODAG configuration data which a node to identify instances of RPL. DODAG information solicitation (DIS) requests are sent by new nodes for joining the network to solicit a DIO from the node which are already part of a valid DODAG [35]. To ensure confidentiality and integrity, RPL supports three security modes named unsecured, preinstalled and authenticated respectively.

2.3.2. 6LOWPAN

IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) is a standard designed by the Internet Engineering Task Force (IETF). This protocol serves as an additional layer to securely encapsulate IPv6 frames in IEEE802.15.4 PHY data packets [35]. To provide maximum performance, 6LoWPAN specification defines header compression and fragmentation. Header compression: describes how IP and UDP headers can be compressed to reduce transmission payload. Fragmentation specifies an algorithm for fragmentation of packets into many link-layer frames, then the collection of IP packets and supports multi-hop packet distribution service.

6LoWPAN is initially designed to support low-power 2.4 GHz wireless networks based on IEEE 802.15.4, but a unique aspect of this specification is now being introduced and utilized in many other systems, namely wireless networks in bands under 1 GHz, Power Line (PLC) transmission, Smart Bluetooth, and low-power Wi-Fi networks [36] [37]. Many methods such as caching, load balancing can be efficiently executed on the IP-based NAT features.

2.3.3. 6TISCH

IETF 6TiSCH protocol incorporates Time Slotted Channel Hopping (TSCH) MAC with IPv6 addressing to create low power, high performance, and deterministic WSNs. Based on the system specifications, this protocol dynamically assigns bandwidth resources to the network nodes. 6TiSCH is based on the IEEE std 802.15.4e - TSCH MAC.

IETF 6TiSCH protocol incorporates Time Slotted Channel Hopping (TSCH) MAC with IPv6 addressing to create low power, high performance, and deterministic WSNs. Based on the system specifications, this protocol dynamically assigns bandwidth resources to the network nodes. 6TiSCH is based on the IEEE std 802.15.4e - TSCH MAC.

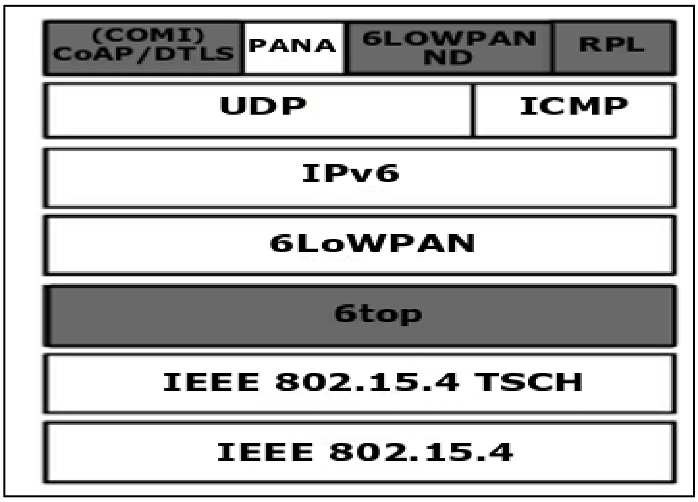

Figure 8. Architecture of 6TiSCH

Generally, for applications and management, UDP and the CoAP are used. RPL is the Low Power and Lossy Networks (LLN) routing protocol. The protocol stack extends the standard IoT protocol stack with a 6top sublayer specification defining the protocol to negotiate for adding and removing cells (i.e. time slots in IEEE 802.15.4e - TSCH MAC) for adjacent nodes. 6top can execute one or more scheduling functions that determine how and when to include and eliminate cells for each node [38].

6TiSCH's objective is to create IPv6 multi-hop connectivity across via nodes running TSCH MAC layer. IEEE802.15.4e TSCH, IETF 6LoWPAN, RPL and CoAP are included in the 6TiSCH protocol stack [39] [38].

3. OVERVIEW AND LITERATURE REVIEW

3.1. Caching

Caching is the mechanism with the aid of which information is stored or mirrored in a cache entity. A cache is a place of temporary storage nearer to the individual needs it rather than the actual source of the information. The technique of caching is focused on preserving content that duplicates original values stored somewhere else that will be requested in a small duration of time.

Caching is one of the most significant attributes of IoT networks. In the Internet or wireless networks, this idea is not recently developed. With the increasing demand to serve enormous requests for data, particularly for various types of contents, it introduces a major challenge over the Internet [40]. From the center to the edge of the network, the same redundant information travels through in most situations which lead to congestions and delays in the network backbone. To overcome this problem and to improve the user experience popular contents are cached in the caching entity.

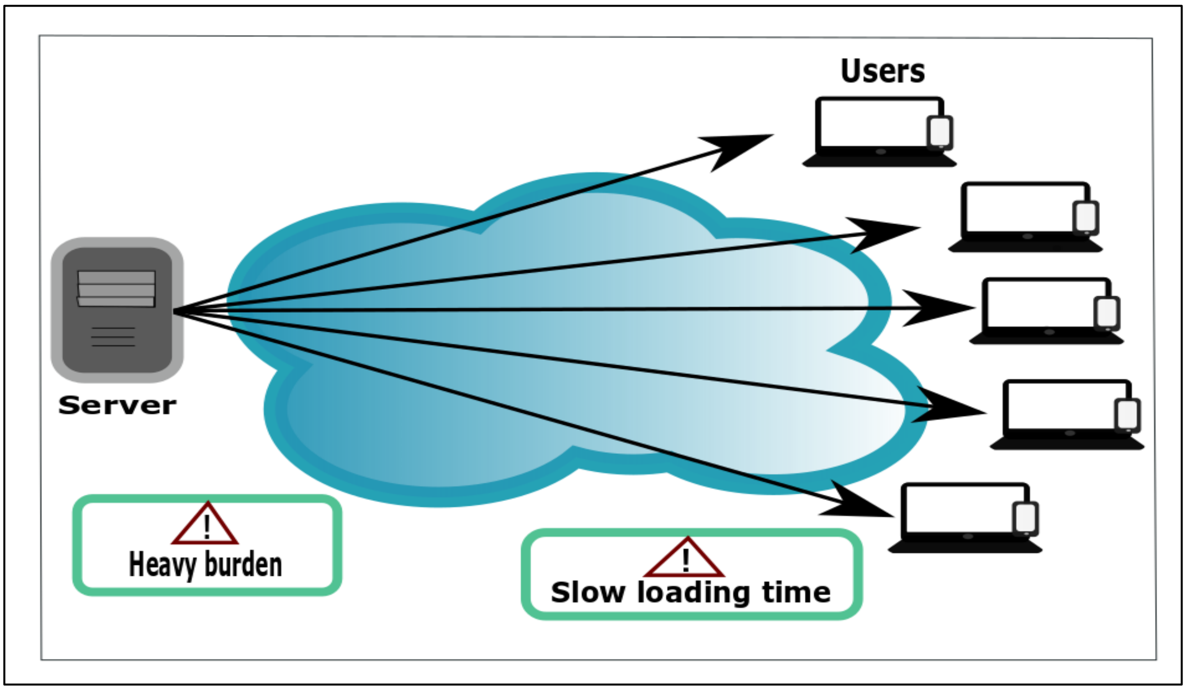

Figure 9. System before having a caching technique [41]

Figure 9 represents the state of the network prior to implementing a caching strategy. Due to huge demand, the server has a significant burden and it takes a lot of time to load in the user data. Caching methodology makes it easier to retrieve data since the data is retrieved from the caching entity not from the source server. Caching has the benefit of reducing database burdens by decreasing access and allow optimal delivery of information as illustrated in Figure 10.

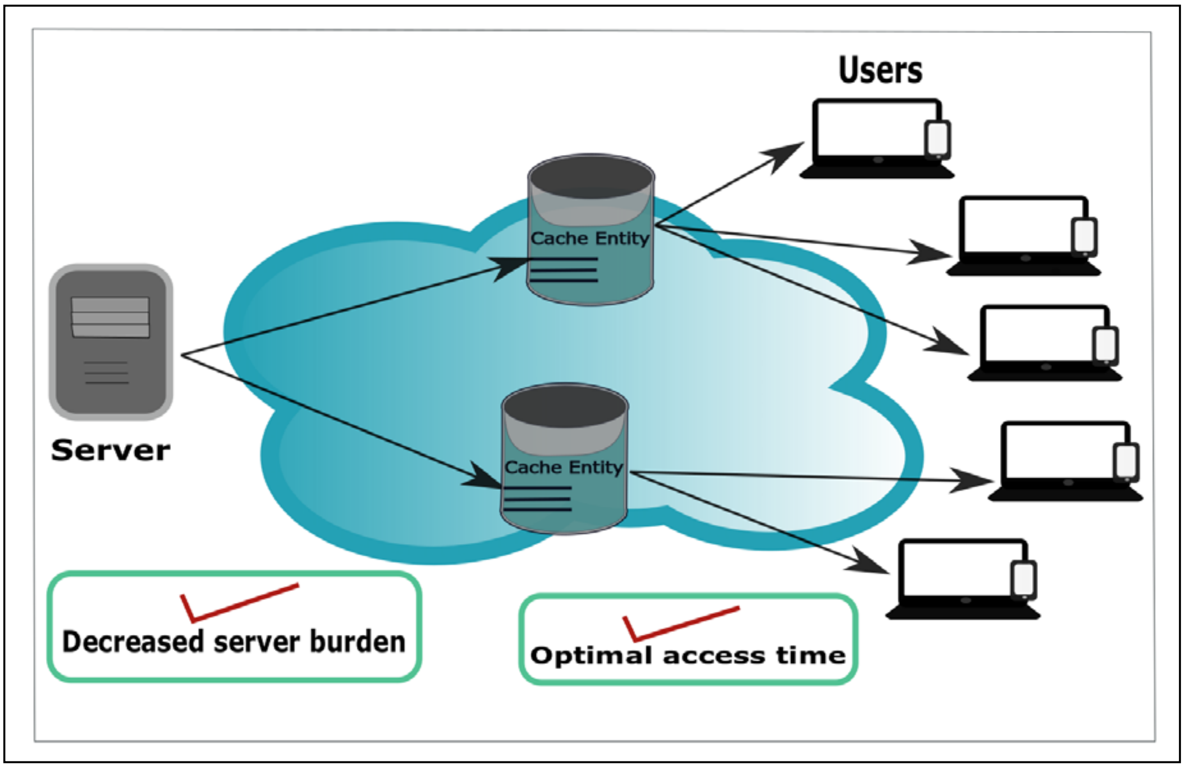

Figure 10. System after having a caching technique [41]

3.1.1. Caching Techniques

This section shortly introduces basic caching techniques and summarizes some of the existing energy-efficient caching techniques. Nowadays caching techniques are used as a way of offloading content delivery of base stations in a cellular wireless network setting. Users can retrieve popular data from caches that are located in the closest nodes. Caching techniques include approaches like proactive, predictive and anticipatory networking[42].

The proactive caching strategy depends on keeping a copy of the information in the local cache whenever it becomes available. The spatial and social formation of the network could be utilized by the proactive caching. To expand the proficiency of future networks paradigm the proactive approach uses the current heterogeneous cellular network and includes the design of predictive radio resource management methods by limiting the peak load of cellular networks and increasing the transfer [40].

Predictive caching techniques exploitly utilize the most recent improvement in storage, context-awareness, social networking and the development of smartphones that have turned out to be extremely refined gadgets with upgraded storage capabilities and the trend to online social networks which become a current popular approach to distribute information. Subsequently, organize hubs track, learn and assemble clients' request profiles to predict future solicitations, utilizing gadgets' abilities and the tremendous measure of accessible information.

Anticipatory caching proposes predictive approaches where the peak traffic activity requests can be significantly reduced by proactively serving client requests, utilizing the cached content at the time of the off-peak traffic to relieve the backhaul congestion [40].

3.1.2. Caching Strategies

A crucial caching challenge is the limited caching space which is especially prominent in IoT networks. The limited storage space on IoT devices is primarily attributed to the factors of cost and energy. To manage cache resources properly and enhance the cache performance, cache management involves two main strategies, the cache decision and the cache replacement strategy.

Cache decision or placement is the process by which decision of where is it best to put cached data. While it is important, there is not much study on strategies for caching decisions [43]. To make this decision different types of policies have been introduced. The first policy is called Leave Copy Everywhere (LCE) policy [44] [45], which leave cache content everywhere on path. Another policy is Leave Copy Down (LCD) [46], which performs caching on a single level downstream router and edge caching. Based on the stretch LCD duplicates the demanded information on a single router [47]. There are algorithms which are proposed based on the following strategies. The first strategy is the

'Leaving Copies with Probability (LCProb)' which leaves a copy of the data in the cache with a pre-determined probability. When the cache replacement depends on a uniform probability, the caching strategy is named as 'Leaving Copies with Uniform Probability (LCUniP)'. Cache placement strategies can be classified as content popularity, content partitioning, selfishness and miscellaneous as shown in Figure 11.

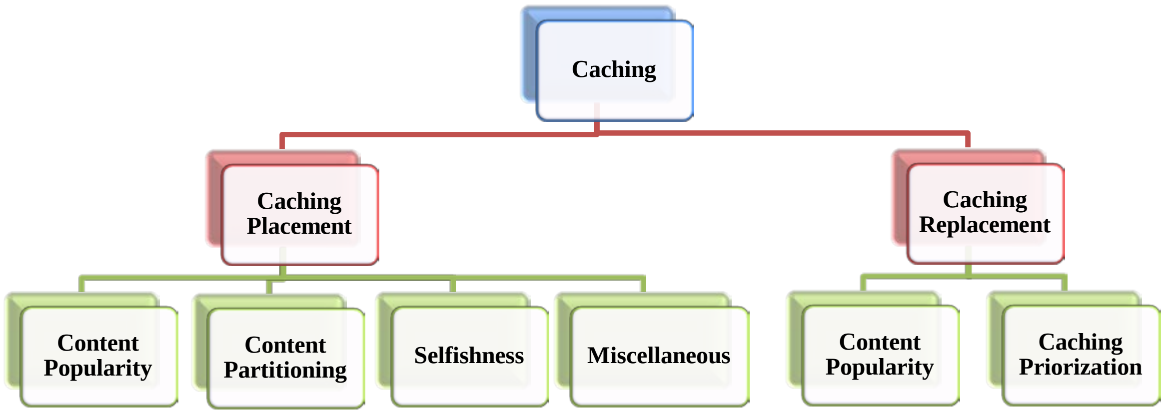

Figure 11. Classification of various strategies for caching.

Cache replacement is the process to decide which content should go where. Replacement strategies for the cache are required if a new data object's size is larger than the amount of available space remaining in the cache. Therefore, the data of the cache must be replaced. Since this is not generally known ahead of time, items are replaced on the basis of a prediction of use [48]. The most commonly used policies are Least Recently Used (LRU) and Least Frequently Used (LFU) [49]. We can categorize cache replacement as content popularity and content prioritization. The summarization of the caching strategies according to various researchers is shown in Figure 11.

3.1.3. Context-Aware Caching

Context is any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves [50]

[51]. The user context can be a collection of different entities For example, identity, behavior, place, and mood; social context: the essence of their relationship with other individuals (such as parents, friends, associates) and their physical context may involve (for example) the level of lighting of their location [41].

It is possible to implement context-aware systems in several ways. The strategy to implementing context-aware systems relies on specific criteria and conditions like the location of sensors (local or remote), the number of possible users (one user or many), the available resources of the devices (high-end-PCs or small mobile devices) used or the facility to expand the system further [51]

3.2. Information-Centric Network

To better cope with the Internet usage Information-Centric Network (ICN) architectures has been proposed. This new approach of communication changes the paradigm from a host-centric network to a data-centric one. Because clients need their intended content irrespective of content location information. In this framework, data or information is considered the primary object rather than the host. The emphasis is thus on WHAT to communicate with, instead of WHO. In ICN caching strategies have been developed by naming the content. And intelligently caching the content across the Internet network, rather than having always the real source for access to content at the time of need [52]. Thus ICN addresses the drawbacks of IP-based solutions and also demonstrates impressive Internet-of-Things (IoT) performance such as faster response time, better portability, better security and so on [53] [54].

3.2.1. Content-centric Networking (CCN)

Many researchers have proposed a range of architecture for Information-Centric Network (ICN) [55] of which one of the most popular is a Content-Centric Networking (CCN) [44]. In CCN, content is sent in response to interest packets that address named chunks of data, which are cached back to the originator of the request. To prevent redundant internet network traffic: interest packets match cached data in per packet, for every request. As with CCN, each node must cache the data, and as several paths can be used, the cache network is therefore not structured as a tree, but rather as an arbitrary graph [43].

3.2.2. The architecture of CCN Based ICN Approach

Subsequently, CCN has become one of the most promising techniques for ICN. The architecture of CCN based ICN is shown in figure 12. CCN communication protocol is a receiver-driven data-centric protocol. CCN implements two types of packets: interest and data packets. These packets carry a name to uniquely identify the carried data. Each content router (CR) has three tables: a content store (CS) for temporary caching of received data packets, a pending interest table (PIT) to contain the list of interest packets and a set of interfaces from which the matching interest packets have been received, and a forwarding information base (FIB) to forward interest packets [56].

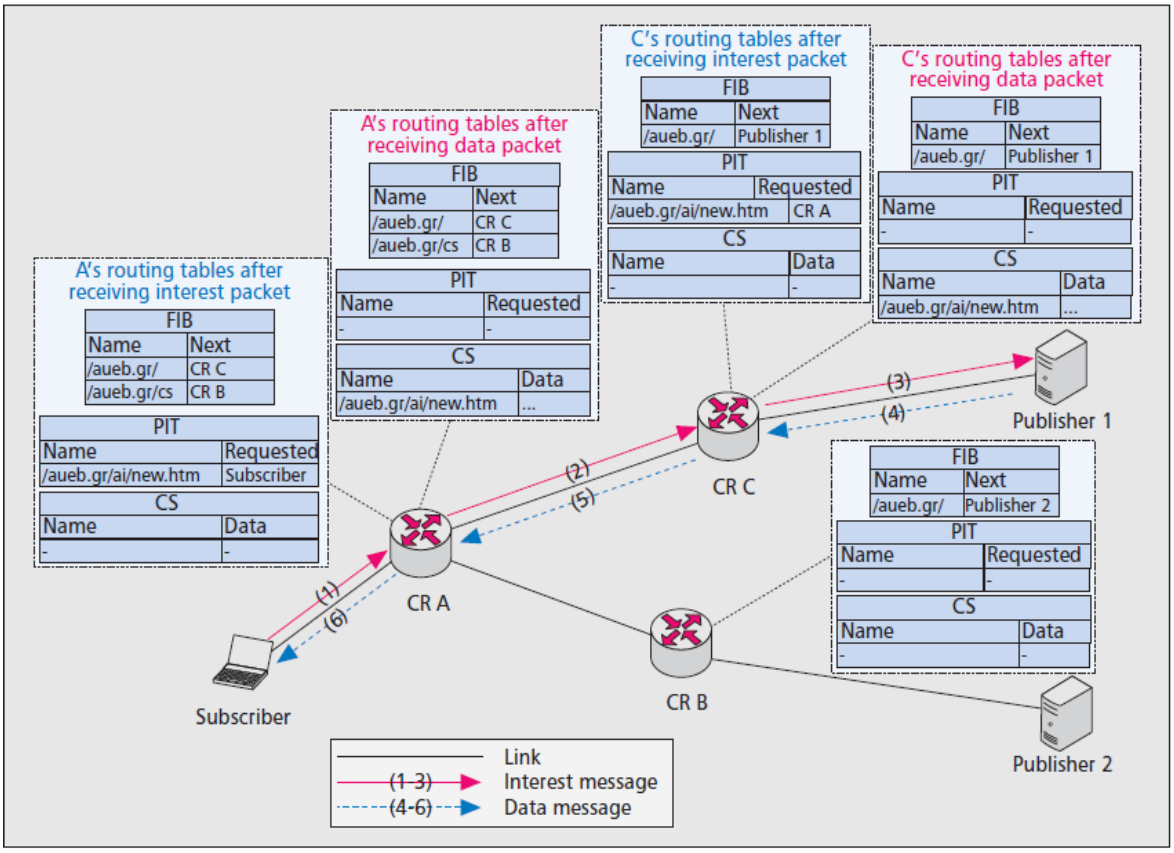

Figure 12. CCN based ICN architecture and workflow [56].

3.2.3. The Workflow of CCN Based ICN Architecture

The working procedure of the CCN-based ICN architecture is given in Figure 12. Suppose subscriber requests a particular content (/aueb.gr/ai/new.htm) by issuing interest packets to its neighbors. When the interest packet arrives at CR A, the CR searches in its CS. If any CR can locate the requested information in the CS, the content is directly delivered from the device in a data message. Else, the router will conduct a longer prefix match on its FIB to determine the exact way of direction. Then the packet is forwarded to the directed path and store information for each forwarded interest in the PIT. When the information found at publisher node, it returns the information as a data message following the reverse path towards the subscriber based on the information stored in their respective PITs. and also, on strategic caching policy. In figure 12. Interest packets path shown as arrow (1-3) and data packet is shown as arrow (4-6) [56].

3.2.4. Content Caching Procedure

In this section, ICN context-aware caching system is explained. According to the example given in Figure.13, user 1 retrieves the content by submitting a packet of interest. While there are no other interest packets, and intermediate ICN routers do not have the requested data in their buffers [54]. The interest packet is forwarded to the server through the shortest path (i.e., H, C, B, and A). The requested content is then delivered by traversing the reverse path (i.e., A, B, C, and H). The forwarded content is kept in the router PIT by each intermediate router. If user 2 needs the same content, there will be a match at terminal B for the interest packet. Then the information requested will be distributed from that node B directly.

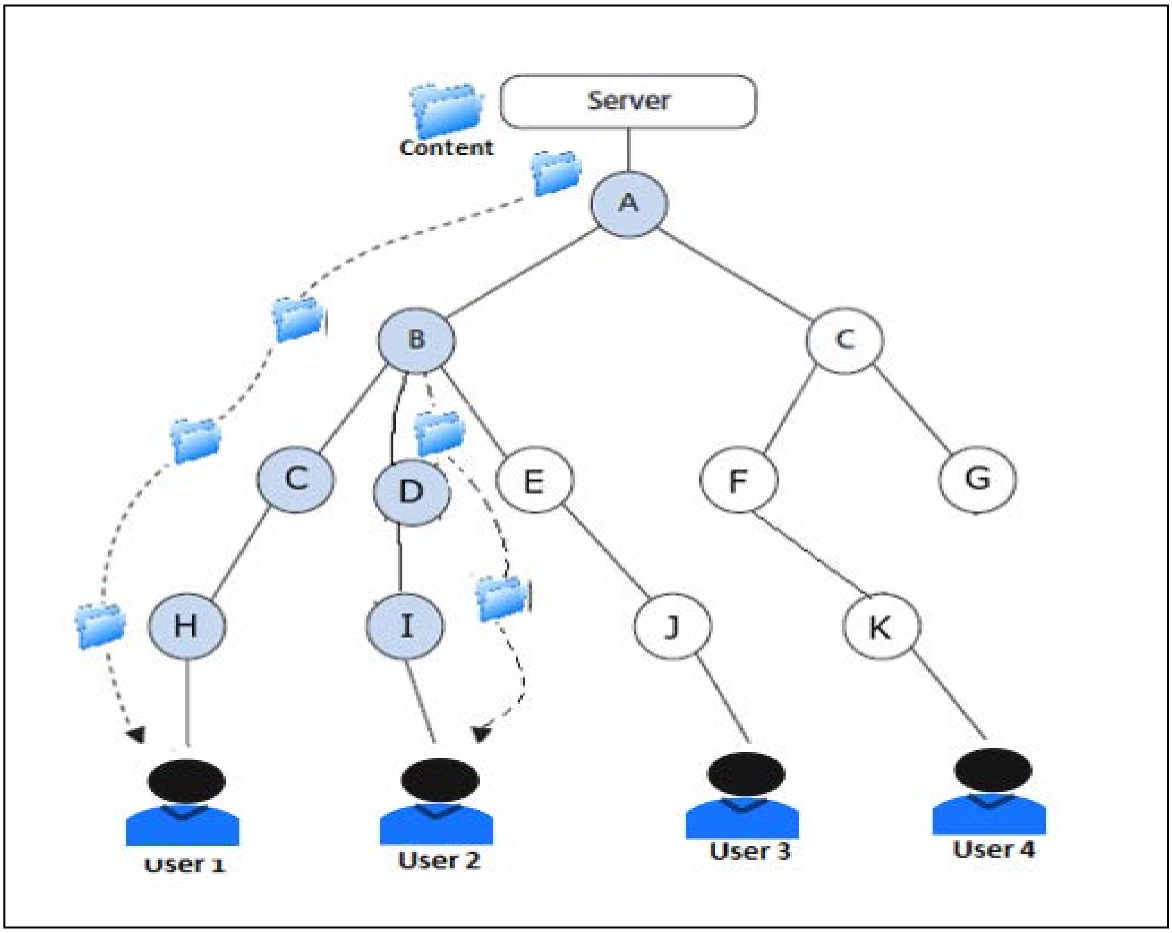

Figure 13. Content caching procedure

3.3. Literature Review

In the recent years, enormous data consumption is observed in wireless networks. Content Distribution Networks (CDNs) is the conventional method for dealing with this increased demand in the wired network. In this approach, content is replicated or mirrored at different locations of the network. Though it is not sufficient to apply the solution of wired Internet to the wireless network as CDN solution has the most benefits if the local communication link is not the bottleneck [57] [58]. As wireless network link may become limited due to interference and contention, CDN approach may not be able to solve the content delivery problem in wireless networks.

.

The main constraints of existing caching research are that all items are typically cached. To solve this limitation, a random caching is proposed in which random contents are selected to cache until the cache capacity is full. But this random caching approach doesn't efficiently solve the limitation and also unable to provide smart solutions to the content delivery problem.

With respect to cache policies, the majority of authors dedicate their attention to cache replacement algorithms rather than caching policies [43] [59]. Generally, in the ICN context, during the delivery period, these cache replacement algorithms upgrade the cache generally relentlessly according to the Least Recently Used (LRU) or Least Frequently Used (LFU) algorithms [60] [55]. Just limited studies include other strategies in ICN, such as Most Recently Used (MRU) and Most Frequently Used (MFU) [61]. As indicated, many of the established decision-making and replacement policies are not of practical significance, since CCN caching procedures need to take place at line speed [43].

When the cache capacity is high, in a backhaul-constrained small cell network the use of caching utilizing optimization methods is analyzed in [62]. In order to enhance the cache hit ratio effectively, instead of implementing traditional caching strategies such as least recently used (LRU), least frequently used (LFU), and first-in-first-out (FIFO), modeling cooperative caching method is challenging.

With an emphasis on energy efficiency, the authors of the [63], for an ICN-based IoT framework, look at resource constraint mobile devices. Similarly, for sensor networks, the authors of the [64] emphasis on energy efficiency. However, none of them focused on the approach of context-awareness of the users.

The research presented in this study focuses on a cooperative context-aware proactive caching strategy which effectively increases the cache hit ratio compared to standard caching approaches. The proposed algorithm's cache replacement is based on context-awareness which is obtained by studying the user's past data in each atomic time slot.

For web caching, some researchers utilize proactive caching which improves the cache decision strategy by prefetching data [65]. In the IoT network, placing content in the cache has a lot of issues. One of them is the optimal placement of cache content depends primarily on the distribution of content popularity. Nevertheless, when caching content at a specific point in time, it is not clear which content will be requested in the future. There may not even be an assessment of the distribution of content popularity at hand. As a consequence, it has to be determined by the caching entity on its own [66] [67], from an overhead point of view which is not valid. For solving this issue, we developed a forecasting model to accurately predict the distribution of content popularity at a specific point in time.

Another concern is that users ' priorities regarding the content consumed that change depending on their location [68], personal characteristics [69] [70] [71], or the features of their appliances. Thus, the placement of cache content should be context-aware since the popularity of content relies upon the user’s context [41]. A caching entity may gain knowledge of user proclivities from various contexts. We modeled the best caching approach by obtaining more practically appropriate approach. Through achieving a more technically efficient approach, we modeled the optimal caching strategy. In our proposed system, we learn the user context in a high degree of granularity by evaluating the instantaneous demands for each atomic time (i.e. per hour). Considering stationary file popularity, it is basically based on the past information of a user which takes into account different locations such as who is most likely to use a particular service library, dormitory, laboratory, etc. [41].

4. CONTEXT-AWARE CACHING SYSTEM MODEL

4.1. Proposed Work

In this section, context-aware caching, a caching strategy that a caching technique is proposed which defines the optimized collection of content to be cached based on the user context. This caching technique is proposed with the goal of maximizing the cache hit rate by intelligently utilizing the limited storage space of cache content. The context-aware caching technique of the cache is the main focus of this study. This caching approach is designed to manage the cached content and enhance the performance of the limited storage capacity of the cache entities.

This caching system mainly has two phases: the training phase and the exploitation phase. In the training phase, the caching entity learns the context of the user by observing the usage pattern of the IoT devices in the network. In the exploitation phase, it caches content that is most popular on average when stored throughout slots of previous times. In this way, the algorithm can learn user context with time. After caching the sets of files, according to the observation, this algorithm predicts future probabilities to cache user data for the given time slot and location. From the predicted data set, the system selects caches content until cache memory is full. Finally, the cache hit ratio is determined.

Our context-aware caching algorithm is based on a proactive caching approach. This is an excellent concept to improve the efficiency of caching in an IoT system when it involves the near-real-time analysis of data. In the proactive cache mechanism [72], the fresh data is queried from the server on a regular basis and updated in the cache instead of the traditional way, where information is retrieved at the time data was requested from the cache. Figure 14. represents the traditional and proactive caching procedure. This enables the user to read information from the cache nearly in real-time.

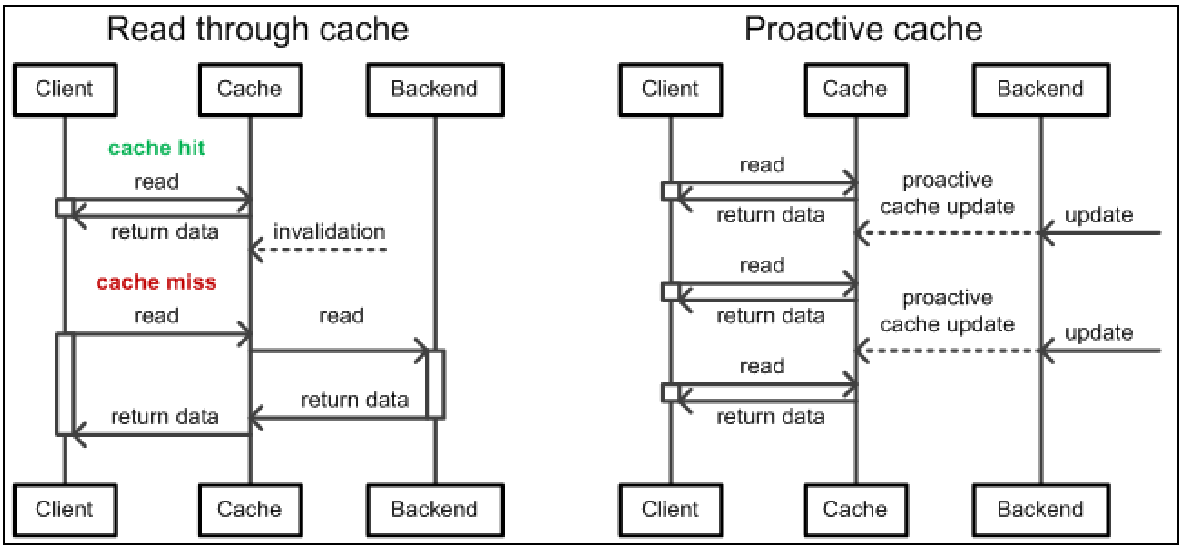

Figure 14. Traditional and Proactive caching pattern [72].

Context-awareness is most significant when the environment is exceedingly dynamic. Users preferences in terms of consumed data may vary depending on their contexts, such as their location, personal characteristics or their devices attribute. Consequently, cache content placement ought to be context-aware by considering that content popularity relies upon a user's context. In this manner, a caching entity can learn the preferences of users with various contexts. Also, content popularity may differ over a user population since different users may lean toward various content. A user's preferences may be connected to different factors.

In Figure 15, we showed overview of how our context-aware system works. The wireless system of a university campus is regarded for the implementation of this research. Limited storage capacity and a weak backhaul connection to the main network distinguish this wireless network. Students use their ID cards in the university network system to verify themselves to access the dormitory, canteen, library, laboratory, etc. Since IoT devices that verify the ID cards have space limitations, it is not possible for devices to preserve the cache of all student information.

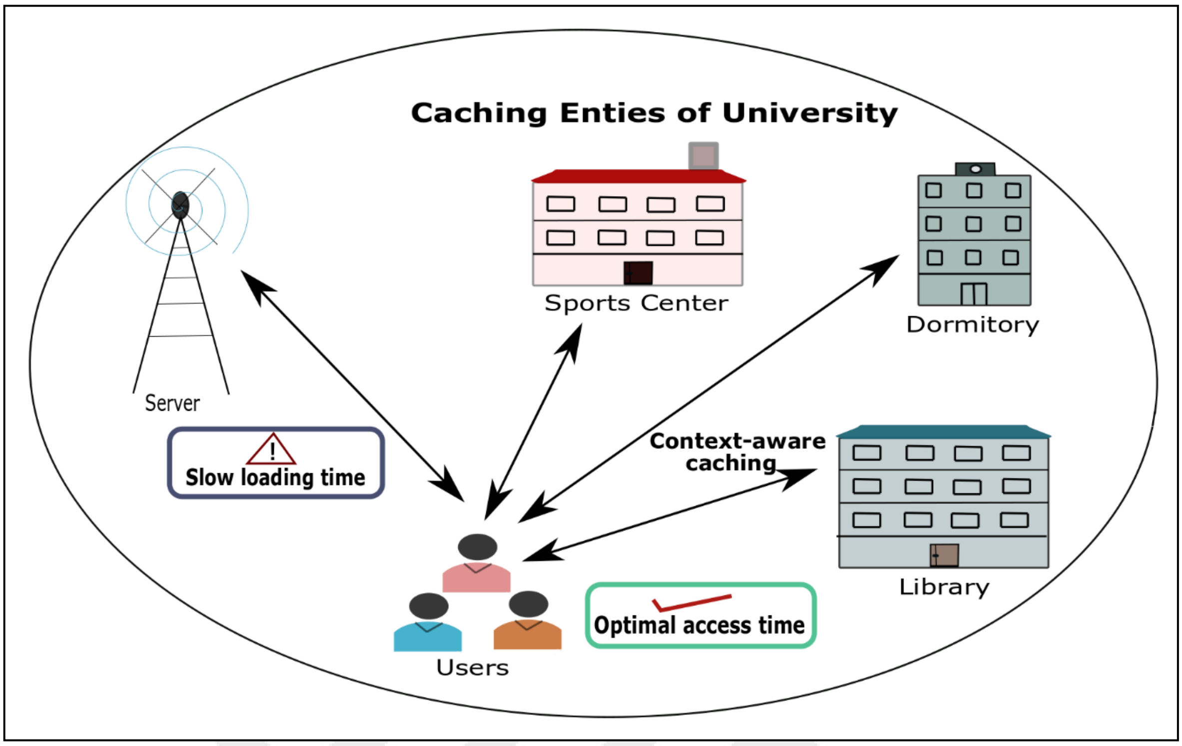

Figure 15. Overview of Context-Aware Caching system model.

If user demand for a file which is cached in the caching entity, then the IoT device requests the file from the caching entity. For this instant, no extra load is put on neither the microcellular network nor the backhaul network. When the data is not available in the cache, it will lead to some delay in data delivery as devices need to collect the information of the student from the main server. The caching entity gathers the file from the content provider by means of its backhaul connection, for this situation, the load is put on both the microcellular and on the backhaul network which will incur slow loading time.

A context-aware approach is used to minimize delays in the authentication process, which learns user behavior from previous knowledge and caches user data based on this acquired context to increase the cache hit rate. As IoT devices are low powered, our proposed context-aware caching approach will optimally place the user data on to the limited storage of the caching entities and also provide optimal access time.

Our proposed system instead of caching randomly all student info in these devices will learn the context of the users'. This is done by observing user, which is going to access these devices in the training phase. Poisson Distribution is employed for modeling user behavior. And according to the probability given by Poisson arrival distribution, the Auto-Regressive Integrated Moving Average (ARIMA) method is used to estimate the probability of user arrival for caching content in the future. Finally, in the limited storage of the device, authentication information for the student data are proactively cached in the local storage for only those specific time and day for each user. The solution maximizes the cache hit by efficiently utilizing the limited storage.

4.2. System Modeling

This section, we represent the modeling of the proposed system and mechanisms employed in the solution. And we also explain the implementation of the designed system which is visualized in Figure 16. The implementation of the proposed method can be visualized as having different steps. Modeling this system is based on three main mechanisms: generating arrival data sets. ARIMA modeling and forecasting, and context-aware caching.

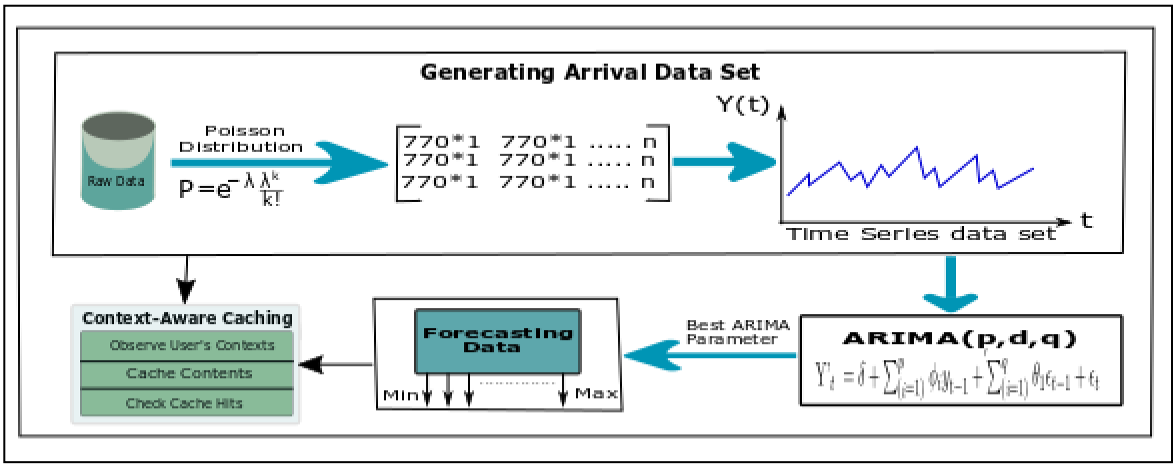

Figure 16. Block diagram of the designed system

4.2.1. Generating Arrival Data Set

This algorithm has two phases: training phase and exploitation phase. At the beginning, this algorithm is in training phase where data set is generated for learning arrival probability of the students. For the purpose of this study, the data sets of 500 student's arrival time during each days of the week at different location of the university is modelled. These datasets include students' ID, locations, students' arrival time and days of the week. In Table 1. represents the list of the main attributes and their details used in the datasets.

Table 1. Dataset Information

Attributes	**Attributes' Details**
Student ID	1,2,3,4,5 …, 500
Location	Laboratory, Lab, Library…
Days of the week	Sunday, Monday, Tuesday, Wednesday, Thursday…
Time	8.00 a.m. – 22.00 p.m.

In order to implement the proposed algorithm more effectively, the system is assumed to have five different locations named Laboratory, Canteen, Library, Sports Center, and Dormitory. The arrival of students in these locations follows the Poisson process which may vary depending on some external factors. These external factors could be considered as context dimensions such as time of the day, days of the week, location, etc. may have an impact on a user's preferences. As the proposed scheme focuses on context-awareness, the system selects different locations with different context dimensions. Each location has different contexts in terms of working hours. For example, location laboratory is active from 8.00 a.m. to 16.00 p.m. whereas active hours of location Library are 8.00 a.m. to 9.0 p.m.

The arrival data set is generated by using generalized Poisson arrival distribution method. The Poisson distribution is a discrete probability distribution which is used to model the probability of occurring number of events within a specific unit of time period [73] [74]. Here, the arrival of an event is independent of the previous event.

For a random variable K, Poisson probability function can be defined as:

$$P(K = k) = \frac{e^{-\lambda}\lambda^{k}}{k!} \tag{1}$$

Where, λ is the expected average number of events per interval $for\ k = 0,1,2$ .... And e is the base of the logarithm.

For finding the Poisson distribution of students arriving in a particular location of the university, first of all, we determine the value of λ. In our case, λ is the average number of student's arrivals per hour. The value of λ can be estimated by assuming that the number of students that arrive during each hour is independent. This implies that each arrival does not influence the probability of other future arrivals. Again, in real-world situations, there could be "seasonality" in the arrival rate. This means student arriving rate in the morning hour could differ from the arriving rate at noon hours. To avoid the "seasonality" issue we considered each atomic instant (i.e. hour of the day).

For example, we have 4 weeks' data on the number of student arrivals at a particular facility of the university in every working hour of the day. The arrivals of the students are at random. In these 4 weeks of data, we observed which student came frequently and which students are not coming. For example, in one month, every Monday at 9.00 am how many students arrived are observed and then the average of this value is taken. After finding this average number of events λ for Monday at 9.00 am, we used it to estimate the probability of that instant.

Following this procedure, the arrival data set for all students is generated for every working hour of each specific service (library, laboratory, etc.). Next, exploratory data analysis is conducted in order to advance with the Auto-Regressive Integrated Moving Average (ARIMA) method and finally, time series training data set is prepared.

4.2.2. ARIMA Modelling and Forecasting

Auto Regressive Integrated Moving Average (ARIMA) is a standard statistical model for analyzing and forecasting time series data. It is one of the most commonly employed strategy to time series forecasting. The model is developed with univariate model which is capable of changing non-stationary data to stationary data by applying finite differentiation of the data points as explained in [75].

This approach combines the moving average (MA) model with the autoregressive (AR) model. Then this method is integrated means an initial differencing step is applied. This model is expressed by the following standard notation:

$$ARIMA(p, d, q) \tag{2}$$

Here, p is the number of the autoregressive term describing the number of lags; d is a means of non-seasonal difference by which current and previous values are subtracted d times and used for stabilizing non-stationary data; q is a moving average showing lags of prediction.

An autoregressive (AR) model is a type of random process which determines that the output variable depends linearly on its own preceding values. An autoregressive (AR) model of order p can be defined as:

$$Y_t = c + \Phi_1 Y_{t-1} + \cdots \Phi_p Y_{t-p} + \varepsilon_t$$
$$Y_t = c + \sum_{i=1}^{p} \Phi_i Y_{t-i} + \varepsilon_t \tag{3}$$

Here, c is the constant, $\Phi_1, \ldots \Phi_p$ the parameters of the model, and ε_t is the white noise.

The moving-average (MA) method states that the output parameter depends linearly on the current and different past values of a stochastic term [76]. The moving-average (MA) model of order q can be defined as:

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots \theta_p Y_{t-q}$$

$$Y_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{4}$$

Here, μ is the mean of the series, $\theta_1, \dots \theta_p$ the parameters of the model, and ε_t is the white noise.

To obtain a non-seasonal ARIMA model, this study combines differencing with autoregression and a moving average model. This model can be mathematically represented as:

$$Y'_t = c + \sum_{i=1}^{p} \Phi_i Y'_{t-i} + \varepsilon_t + \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$
$$Y'_t = c + \Phi_1 Y'_{t-1} + \cdots + \Phi_p Y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots \theta_p Y'_{t-q} + \varepsilon_t \tag{5}$$

In equation 5. Y'_t is the difference of the series, the mean of the series μ is zero. This can be rearranged as:

$$Y'_t - \Phi_1 Y'_{t-1} \dots - \Phi_p Y'_{t-p} = c + \theta_1 \varepsilon_{t-1} + \cdots \theta_p Y'_{t-q} + \varepsilon_t$$
$$\left(1 - \Phi_1 \mathrm{B} - \cdots - \Phi_p B^p\right) (1 - B)^d Y_t = c + \left(1 + \theta_1 \mathrm{B} - \cdots - \theta_p B^p\right) \varepsilon_t \tag{6}$$

where B is backshift operator. Equation 6. represent $ARIMA(p, d, q)$ with drift.

First, Algorithm 1. selects a few weeks of data-sets as a training data for example 10 weeks of training data sets. This data sets represent some challenges such as growing trend and being nonstationary. As this noisy data can lead to inaccurate results, to achieve precise results data set cleaning is an important part. In order to progress with the $ARIMA$ model exploratory data analysis is conducted and data is converted into stationary data. The first order of differentiation provides sufficient precision to convert the non-stationary data into stationary. After the time series data set has been analyzed, the $ARIMA$ model is used for prediction.

To get the best parameter for $ARIMA$ model this study performed the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). AIC and BIC are both penalized-likelihood criteria which are used for choosing best predictor subsets in regression. The analysis indicates that $ARIMA(2,1,2)$ is the best parameter for the proposed model. As $ARIMA(2,1,2)$ gave the lowest value for AIC and BIC. Lower AIC and BIC implies that a model is assumed to be closer to the truth [75].

In the $ARIMA(2,1,2)$, the value of parameter $p = 2,\ d = 1,\ q = 2.$ Then from equation 6. its mathematical structure can be shown as:

$$(1 - \varphi_1 B - \varphi_2 B^2)(1 - B)Y_t = (1 - \theta_1 B - \theta_2 B^2)\varepsilon_t \tag{7}$$

This study simplified the equation 7. As follows:

$$Y_t = \pi_1 Y_{t-1} + \pi_2 Y_{t-2} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} \tag{8}$$

Where, $\pi_1 = (\varphi_1 - 1)$ and $\pi_2 = \varphi_2 - \varphi_1$.

$ARIMA(2,1,2)$ is used to forecast future arrival data of the students. For example, one week of arrival data is predicted by analyzing 10 weeks of training data set. Then, the data is used to run under Monte Carlo Simulation. Monte Carlo simulation approach used to evaluate unknown situations and to analyze probabilistic conditions by running a random sample of the series of input parameters. The probabilistic scenario is provided by each outcome of the Monte Carlo simulation run. The system considered simulation results to determine the cache hit rate for computing efficiency of the system.

4.2.3. Context-Aware Caching

The proposed system first observed the user context information which is the usage pattern of the IoT devices in the network. In accordance with the observed past information, future arrival data is forecasted. After forecasting the arrival probabilities of the active user, according to the cache capacity of the cached entity, a number of users are selected to cache. The selection of the cached content is based on the criteria which maximize the sum of expected numbers of cache hits. Every available caching entity in the network selects a collection of files to cache that display cached user authentication data for the particular time instance. The IoT devices are characterized by a limited storage space to store the cache. It implies that the cache entity can keep up to m number of files in its storage from a set of files F. This file F contains an entire set of active users' information. The system observes each individual who is most likely to use these devices in order to increase the cache hit rate. Moreover, to prioritize certain users, each arrival of a user is annotated with a weight. For each location, if within the same time slot, a user arrives twice at the same location, its weight is maximized. Over time the caching entity learns which users are most popular. For each time slot, it caches the most popular m number of user data.

4.3. Proposed Algorithm

In this study, a context-aware caching algorithm is proposed. This generalized algorithm provides an innovative way to proactively cache the most appropriate user's context information files with maximum probabilities from all the connected users for finding cache hit. For developing this algorithm, the usage pattern of IoT devices in the university network is assumed as the context information of the user. The principal motivation for implementing this algorithm is to efficiently cache content in order to maximize the cache hit rate while intelligently utilizing the system's insufficient cache storage resources.

As discussed before, proposed algorithm first activate its training phase. In the training phase, this algorithm begins by generating students arrival data set which is considered as training data set D_T. This training data set D_T, for arrival probability of the students is generated by using a generalized Poisson Arrival Distribution method. To get Poisson Arrival Distribution data $D_{Poisson}$, it is essential to estimate the value of λ for all user's ID. As one of the main interest of this study is to accurately forecast hourly arrivals. Therefore, the value of λ is estimated for all locations, over the entire cycle of the training data set i.e. working hours of the day and the working days of the week. If selected training data set D_T is of 10 weeks, to get the value of k, user's hourly device accesses are observed for each week for the whole training period. After obtaining Poisson distributed arrival data $D_{Poisson}$, the size of the training data set for a single facility e.g. Library will be 770, which is shown in the matrix of the Figure. 16. In this matrix, column represents all the active student and rows represents the locations of the university network system. For the purpose of this study, in a day it is assumed that location Library has 11 working hours and 10 weeks of training data set is stored in the system. So, $10\, weeks * 7\, days * 11\, hours = 770$. Eventually, arrival data set is modeled and it is sorted in descending order by their probability. At the end of the training phase while user's contexts are learned, C_T selects a set of user's IDs with the highest probability based on the cache memory capacity of the IoT devices.

Algorithm 1 Context-Aware Caching Algorithm

procedure $Week_T, Week_F$ & $Cache_{Capacity}$
$D_T \leftarrow [\,]$ ▷ Make a variable for training week
for $1st_{trainWeek} : last_{trainWeek}$ **do**
    $Lambda \leftarrow Lambda_{AllID}$ ▷ Get lambdas for IDs
    $Data \leftarrow Data_{AllID}$ ▷ Get data for IDs
    $D_{Poisson} \leftarrow Poisson(Lambda, Data)$ ▷ Find Poisson distribution for all data
    $D_T \leftarrow Sort(D_{Poisson})$ ▷ Sort data as a descend
    $C_T \leftarrow Select(D_T, Cache_{Capacity})$ ▷ Select first higher ID probabilities according to length of $Cache_{Capacity}$
end for
$Hour_T \leftarrow Hour(C_T)$ ▷ Observe training data hourly
$D_F \leftarrow [\,]$ ▷ Make a variable for forecasting data
for $1st_{forecastWeek} : last_{forecastWeek}$ **do**
    $dataSet \leftarrow Data_{AllID}$
    $D \leftarrow ARIMA(2, 1, 2)$ ▷ Run ARIMA model
    $D_R \leftarrow MCSim(D)$ ▷ Run Monte Carlo Simulations for forecasting data
    $D_F \leftarrow Sort(D_F)$
    $C_F \leftarrow Select(D_F, Cache_{Capacity})$
end for
$Hour_F \leftarrow Hour(C_F)$ ▷ Change data to hours
$HitC \leftarrow HitCheck(Hour_T, Hour_F)$ ▷ Find hit number between training and forecasting weeks

Figure 17.Context-Aware Caching Algorithm

To progress with the Box–Jenkins Arima modeling, the first step is to determine if the time series is stationary. An Augmented Dickey-Fuller (ADF) test is performed to test the stationarity of the data set. Then, this study performs first-order differentiation to make the data set stationary with the help of exploratory data analysis. Next, $ARIMA$ is applied to the data set. $ARIMA(2,1,2)$ is found to be the best model. To check the fitness of the regression, the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) techniques are used. Both the AIC and BIC gives the lowest value for $ARIMA(2,1,2)$. So, for this study $ARIMA(2,1,2)$ is found to be the best parameter. The Monte Carlo simulations are run 500 times over the output data set of $ARIMA(2,1,2)$ to accurately model the future arrival probabilities of the students. The model forecasted the student's arrival probability for every hour of the days of the week. The forecasted data set is sorted in descending order based on their probabilities, in order to select the student's data with the highest probability to maximize the cache hit. Finally, the system selects m number of users according to the cache capacity of the entity for each time slot.

5. PERFORMANCE EVALUATION

5.1. Simulation Setup

For designing, implementation and simulation of the proposed Context-Aware caching system MATLAB version 9.2 (R2018b) is used. The simulations ran on a 16 GB RAM Intel CORE i5 processor with 2.7 GHz. In order to better analyze the efficiency of the proposed system, many simulations were conducted with different system conditions and parameters. Experimentally a deeper knowledge of the functionality of the proposed algorithm could be obtained by carrying out comprehensive experiments with different parameter values. To test the impact of the context-aware caching system, results are represented for parameters such as the number of users, the cache memory size and five different device locations of the university network. The performance of the proposed scheme is compared with a random caching algorithm and Least Frequently Used (LFU) algorithm. Least Frequently Used (LFU) is a caching algorithm in which, when the cache is overflowed, it removes the least frequently used files from the cache. To evaluate the performance of the proposed context-aware caching strategy, these two algorithms are used as a benchmark. In Table 2, summarization of system parameters used in the simulation is given.

Table 2. System Parameters

Parameter	Meaning
$\boldsymbol{C_T}$	Cache Capacity
F	Total number of users
T	Time
P	Probability
CH	Cache hit rate
$\boldsymbol{CH_a}$	Average cache hit
$\boldsymbol{CH_{mx}}$	Maximum cache hit rate
$\boldsymbol{CH_{mn}}$	Minimum cache hit rate

5.2. Simulation Results

In order to figure out the effect of the proposed algorithm, the system is assumed to have five different locations named Laboratory, Canteen, Library, Sports Center, and Dormitory. Each location has different contexts in terms of working hours. The arrival of students in these locations follows the Poisson process which may vary depending on some external factors. Context dimensions such as time of the day, days of the week, location, etc. may have an impact on a user's preferences. Figure 18. shows the effects of different context dimensions at different locations of the network.

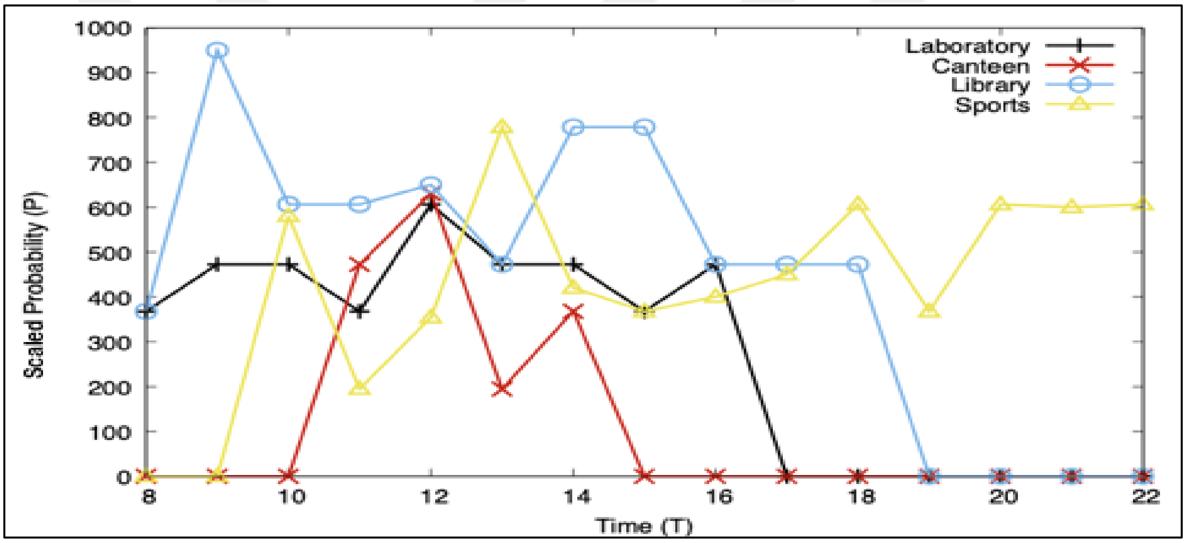

Figure 18. Effects of different context dimension

In figure 18. X axis represents the active hours of the day and the Y axis represents the scale of arrival probabilities of the students which is mapped into [0, 1000]. In this study, at location Laboratory students have classes from time T, 8.00 a.m. to 16 p.m. The system is assumed to have the highest probability of students' arrival at the location Canteen from 11.00 a.m. to 14.00 p.m. The active hours of the location Library are 8.00 a.m. to 9.0 p.m. So, it is visible from the above figure that depending on the context dimensions of individual location, the arrival probabilities of students are varying.

Figure 19. represents an example scenario about how the forecasted user's arrival probabilities P are used to estimate the cache hits. In this figure, the column represents the scale of probability and rows represent the days of the training and forecasted weeks.

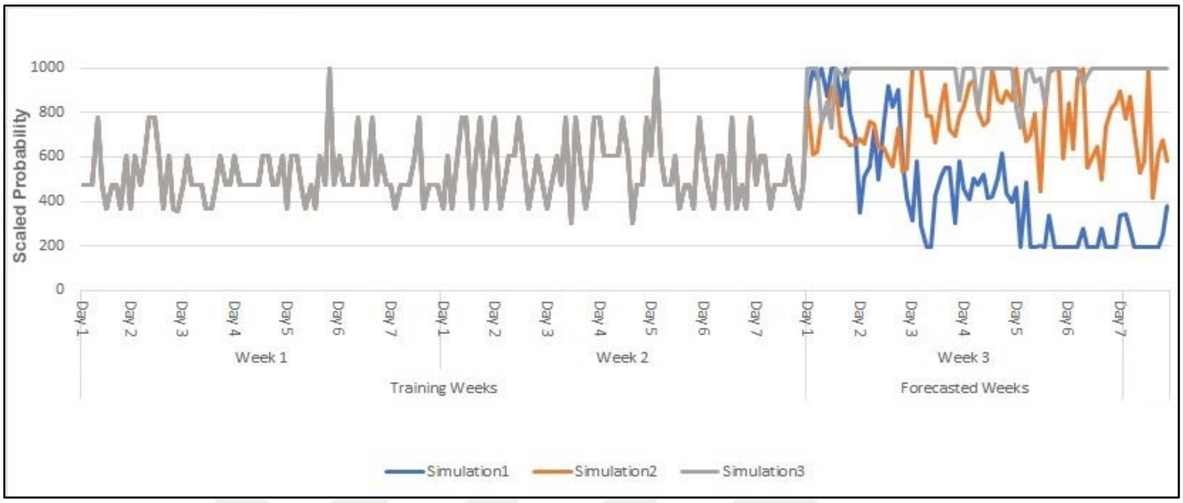

Figure 19. Outcomes of Monte Carlo simulations.

The outcomes of three separate simulations are presented to give an insight into the forecasting procedure. In these three simulations, the outcome of Simulation1 in the figure 19. gives the lowest user arrival probability among the other two simulations. For this particular forecasting simulation, the cache hit ratio is found to be 77%. The probability trends in Simulation2 follows the training data set closely. So, this particular simulation achieves the maximum cache hit ratio of 95%. The results of Simulation3 differ greatly from the input training data set. As a result, it's cache hit ratio is found to be only 52%. The average cache hit $\boldsymbol{CH_a}$ ratio is obtained by averaging these outcomes which is 75% percent for this example. To get more accurate data the results here are calculated by averaging over 500 runs of the algorithms with only the first 10 weeks of the training data set. From the 500 runs of the simulation, the minimum, the maximum and the average values of all simulation are used to analyze the performance of the proposed algorithm.

Table 3. presents the performance evaluation of the context-aware caching system. Three different cache capacity limits $\boldsymbol{C_T}$ of IoT devices of a single facility with different

numbers of users are used to evaluate the context-aware caching algorithm. For this study, it is presumed that the network has five different locations.

Table 3.Performance evaluation of context-aware caching algorithm

No. of Users	**Cache Capacity, C_T**	**Locations**	**Cache Hits (Context-Aware Caching)**		
			Avg Hit, CH_a	**Max Hit, CH_{mx}**	**Min Hit, CH_{mn}**
50	10	Laboratory	79.46	98.89	22.22
		Canteen	73.20	100.00	15.00
		Library	94.09	99.09	33.64
		Sports	97.49	97.69	58.46
		Dormitory	72.13	100.00	17.50
	25	Laboratory	81.94	97.34	61.34
		Canteen	79.96	100.00	51.00
		Library	96.42	97.82	75.64
		Sports	95.90	96.00	78.46
		Dormitory	80.69	80.69	80.69
	40	Laboratory	88.56	97.22	80.00
		Canteen	94.13	100.00	82.50
		Library	97.84	98.41	88.64
		Sports	98.85	98.85	98.85
		Dormitory	91.28	91.28	91.28
100	20	Laboratory	66.28	91.11	17.78
		Canteen	72.44	100.00	16.25
		Library	94.02	98.64	34.09
		Sports	97.49	97.69	58.46
		Dormitory	66.28	91.11	17.78
	50	Laboratory	74.47	95.33	47.33
		Canteen	79.70	100.00	52.50
		Library	94.30	97.82	63.45
		Sports	93.88	94.31	80.77
		Dormitory	79.70	100.00	52.50
	80	Laboratory	88.61	96.11	82.08
		Canteen	90.97	100.00	77.81
		Library	95.51	97.16	84.55
		Sports	87.36	87.50	83.56
		Dormitory	88.61	96.11	82.08

Table 2 continues …

No. of Users	Cache Capacity, C_T	Locations	Cache Hits (Context-Aware Caching)		
			Avg Hit, CH_a	Max Hit, CH_{mx}	Min Hit, CH_{mn}
300	60	Laboratory	68.98	97.78	20.19
		Canteen	71.56	100.00	19.17
		Library	92.85	97.73	33.48
		Sports	94.14	94.87	91.67
		Dormitory	68.98	97.78	20.19
	150	Laboratory	75.10	93.85	49.56
		Canteen	76.62	99.83	50.50
		Library	93.98	97.03	58.48
		Sports	92.58	93.03	91.85
		Dormitory	75.10	93.85	49.56
	240	Laboratory	85.65	94.49	79.72
		Canteen	92.15	100.00	79.69
		Library	94.85	96.21	84.96
		Sports	94.06	96.38	94.01
		Dormitory	85.65	94.49	79.72
500	100	Laboratory	66.68	96.44	25.33
		Canteen	78.08	100.00	19.50
		Library	94.77	96.00	92.45
		Sports	92.03	92.31	31.08
		Dormitory	68.98	97.78	20.19
	250	Laboratory	78.93	95.73	51.82
		Canteen	79.19	99.90	50.70
		Library	94.87	97.06	64.35
		Sports	92.54	93.35	62.31
		Dormitory	78.93	95.73	51.82
	400	Laboratory	85.16	94.61	79.67
		Canteen	92.36	100.00	79.19
		Library	93.68	96.91	83.45
		Sports	90.92	91.94	90.42
		Dormitory	92.15	100.00	79.69

The effect of the device's cache capacity $\boldsymbol{C_T}$ on achieving cache hit is analyzed by ranging it's size from 10 to 400, which is 20% to 80% of the total number of active users F as listed in table 1. For simulations of different predicted data sets using the ARIMA method, at Laboratory a minimum $\boldsymbol{CH_{mn}}$ 22.22%, a maximum $\boldsymbol{CH_{mx}}$ of 98.89% and an average $\boldsymbol{CH_{a}}$ 79.46% of the cache hit ratio is observed when there are only 50 users and the device's capacity is only 20 percent of the total number of users.

Table 4. Comparison of Context-Aware Caching with LFU and Random Caching Algorithm

Cache Capacity	**Locations**	**Context-Aware**			**LFU**			**Random**		
		Avg Hit, CH_a	**Max Hit, CH_{mx}**	**Min Hit, CH_{mn}**	**Avg Hit, CH_a**	**Max Hit, CH_{mx}**	**Min Hit, CH_{mn}**	**Avg Hit, CH_a**	**Max Hit, CH_{mx}**	**Min Hit, CH_{mn}**
20%	Lab	70.35	96.33	21.38	23.17	26.69	18.37	20.51	23.19	16.66
	Canteen	73.82	100	17.48	23.19	28.31	14.21	20.16	25.31	14.67
	Lib	93.93	97.86	48.42	24.16	24.73	20.55	17.63	19.37	16.78
	Sports	95.29	95.64	74.92	24.41	24.53	22.19	20.56	20.87	19.06
	Dorm	91.89	94.57	58.08	23.77	23.83	21.00	18.06	19.14	16.78
50%	Lab	74.70	95.23	50.07	52.11	54.89	49.02	48.80	51.36	47.00
	Canteen	78.87	99.93	51.18	50.32	54.44	46.40	50.44	53.13	46.73
	Lib	94.89	97.43	65.48	54.81	55.41	50.15	48.27	49.20	47.97
	Sports	93.73	94.17	85.85	56.49	56.69	55.46	49.09	49.53	48.80
	Dorm	94.33	97.34	65.48	53.11	52.71	50.93	47.83	49.01	45.70
80%	Lab	86.99	95.61	8037	79.21	80.52	78.44	79.35	80.14	78.57
	Canteen	92.40	100	79.80	79.36	80.80	77.84	78.91	80.63	77.74
	Lib	95.47	97.17	85.40	80.24	80.49	78.98	79.41	79.82	79.02
	Sports	92.80	93.67	91.71	80.04	80.20	79.97	79.10	79.18	79.00
	Dorm	95.10	96.05	85.33	81.32	80.54	77.07	78.58	80.21	78.64

In the next simulation, other parameters are kept be constant by only increasing the size of the cache to 50%, a minimum 61.34%, a maximum 97.34%, and an average 81.94% cache hit ratio is achieved. So, in this simulation results, accept cache capacity no other parameters have affect the cache hit ratio. From this analysis, we can find that cache hit is highly dependent on the device's cache storage capacity due to the fact that higher cache capacity enables storing more data.

The impact of the popularity of the locations among the users is also analyzed. When the cache storage capacity C_T is just 20 percent of the total number of users, average cache hit ratio CH_a is achieved 94.09% at the location Library. For the same configuration, the outputs of the location Laboratory greatly differ from these outputs. This is because the popularity of these places differs among the students.

Next, to evaluate the effect of the number of users on system performance, a series of simulations are conducted where the results are presented for increasing the size of the cache. When the parameter F is set to 100, C_T is set to 20% at location Canteen, an average of 72.44% cache hit ratio is observed. By increasing the value of F to 300 and C_T to 50%, an average of 76.62% cache hit ratio is noted. The findings in Table 3 shows that even though the cache size is very small in comparison with the total number of users, the proposed caching approach can provide a significant performance benefit.

The performance of the proposed algorithm is compared against two reference algorithm which is presented in table 4. As a benchmark random caching algorithm and LFU caching algorithm is used. For location Laboratory, the parameter cache capacity C_T is set to 50% of the total number of students, proposed algorithm achieves an average of 76.64% cache hit ratio while random content caching method provides an average hit ratio of 48.2%. On the other hand, LFU algorithm gives an average hit ratio CH_a of 52.33%. These findings suggest that the proposed algorithm offers a higher number of cache hits compared with caching strategies for random and LFU. This is because from historical observations, the context-aware caching algorithm learns the context of the users. While the Least Frequently Used (LFU) algorithm only examines those records in the cache, the Random caching method uses no statistical information to forecast the user's context. To recapitulate, proposed Context-Aware caching method outperforms other approaches.

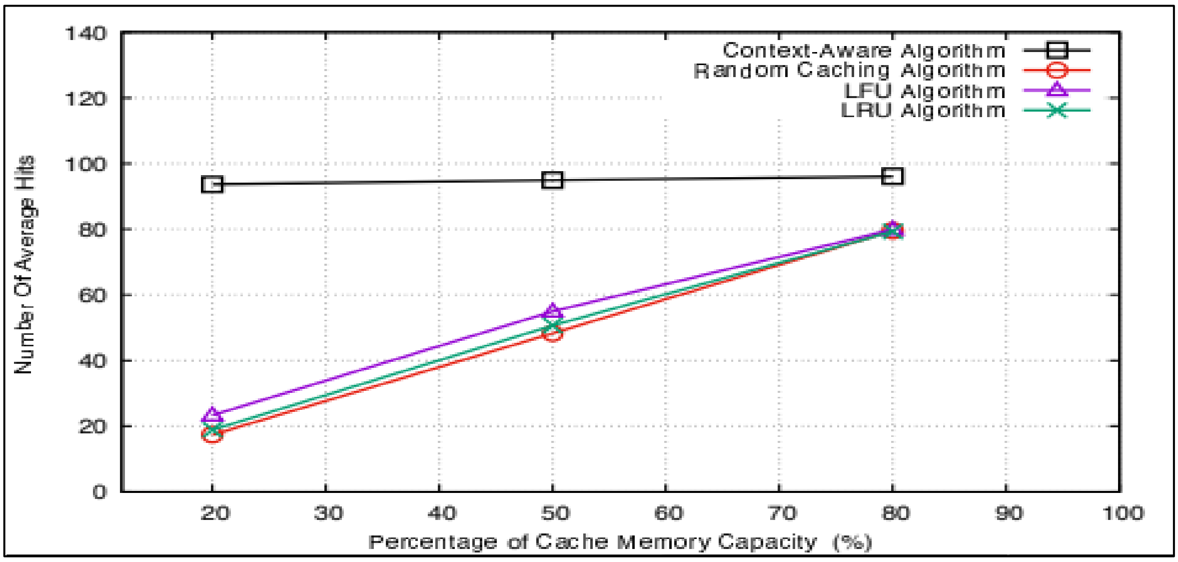

Figure 20. Comparison of the average number of cache hit for location Library

In the figure. 20, for the context-aware caching algorithm the average number of cache hits with varying storage capacities is shown for location Library. Compared to random caching, Least Recently Used (LRU) and LFU caching algorithm, the context-aware caching algorithm gains a higher cache hit ratio for cache capacities which can store only 20% of total user data. The outcomes of LRU are very similar to LFU as expected. From the outcomes, it is visible that the cache hit ratio increases as expected with the growing size of cache storage.

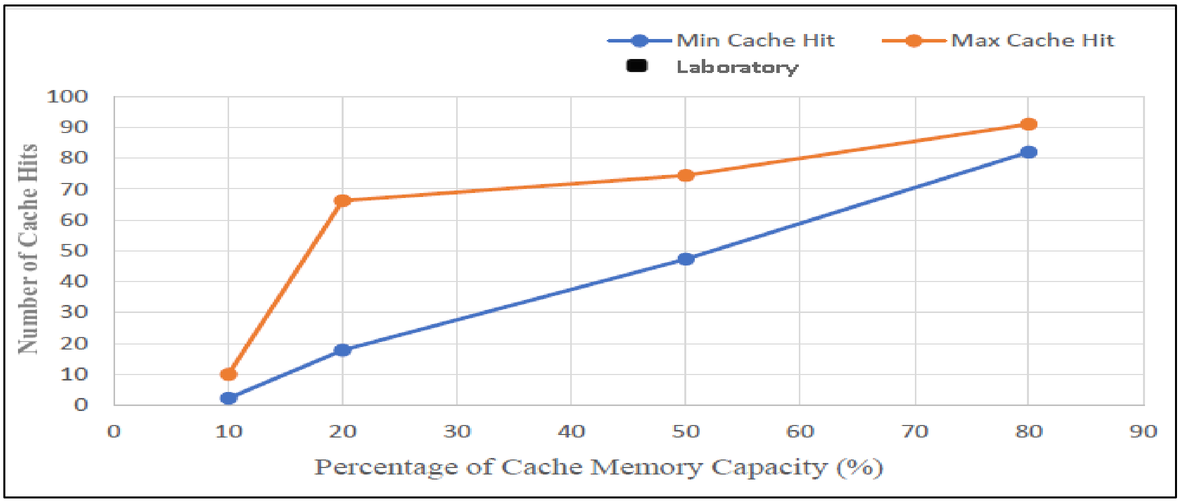

Figure 21. Minimum number of cache hits

In order to evaluate the proposed cache replacement policy, we also consider minimum cache hit ratio. Figure 21. represents the simulations outcomes for location Laboratory. In this figure, the column represents the scale of probability and rows represents the cache capacity. When the system is assumed to have 100 users, for 80% of cache capacity context-aware caching algorithm obtain minimum of 82.02% of cache hit ratio. For 50% of cache capacity context-aware caching algorithm observes minimum of 47.33% of cache hit ratio. When cache capacity is 20%, context-aware caching algorithm observes 17.78% of minimum cache hit ratio. If cache capacity is reduced to only 10%, this proposed algorithm starts to fail. When concerning the maximum cache hit, it is visible from that for only 10% of cache capacity, cache hit line starts to fall down. This is due to the reason that smaller cache storage limit unable to cache appropriate content when the number of users are higher.

6. CONCLUSION AND FUTURE WORK

6.1. Conclusion

The IoT model aims to be the Future Internet architecture to cope with its transformation towards a vast and endless world of devices, users and services. In spite of the many advantages of using IoT networks, the requisites determined by IoT applications decelerate the development of the IoT paradigm. IoT is the Internet network that is mostly influenced by mass traffic and the complexity of interaction between the main server and end devices. This is due to the growing number of smart things imposed by their implementations and the rigid requirements. Hence, caching methodology has drawn the IoT community’s awareness.

This dissertation highlighted the capability of caching methodology as a solution for IoT systems support. Nonetheless, the caching approach should make information accessible to users without increasing the volume of data in the device entity while utilizing limited cache capacity of the devices. The main objective of this thesis is to develop a strategy that can serve the clients directly from the cached data to reduce access delay in an IoT network. This approach can be an effective method for balancing an IoT network's traffic load to facilitate low latency applications. The proposed algorithm studies user context data on a regular basis and preemptively stores the appropriate user information in the cache memory of the device.

The simulation tests have been designed to evaluate the efficiency of this proposed method. The cache hit ratio of the system is analyzed. The reason of determining cache hit ratio of the proposed system is that cache hit allows comparing the absolute performance of the algorithms. A number of the cache hit data was collected with different test scenarios after a significant number of experiments.

Once the cache hit is analyzed, findings are summarized on the impact of different conditions. Numerical results are utilized to validate the proposed algorithm, and the findings indicate that the proposed approach maximizes cache hits within an IoT network. After the comparison of simulation results with referenced algorithms, it is found that the Context-Aware caching algorithm outperforms other approaches. To recapitulate, it is proved that the proposed caching strategy improves system performance in terms of cache hit and caching efficiency. Since it makes the cache more efficient by replacing invalid contents with the knowledge of user contexts.

6.2. Future Work

Despite the context-aware solution proposed in this thesis to deal with the challenges of IoT network, our proposed approach can still be addressed with some enhancement. Research on IoT networks based on caching is left with a lack of implementation while the majority of these are evaluated based on simulation. Real platform implementation is necessary to accurately evaluate the performance of these methods. So as future work, the proposed algorithm will be applied in a real-life IoT network.

In this study, the time needed for calculations is not considered. As modeling and implementation are simulated together, it takes a longer time to compute when system parameters are set to a high value. In real-time, the modeling will be done before, so this proposed algorithm will work faster.

7. REFERENCES

1. Wang, X., Chen, M., Taleb, T., Ksentini, A. ve Leung, V.C., Cache in the air: Exploiting content caching and delivery techniques for 5G systems, IEEE Communications Magazine, 52,2 (2014) 131-139.

2. Breslau, L., Cao, P., Fan, L., Phillips, G. ve Shenker, S., Web caching and Zipf-like distributions: Evidence and implications, 1, 126-134, Ieee Infocom 1999.

3. Weiser, M., The Computer for the 21 st Century, Scientific american, 265,3 (1991) 94-105.

4. Ashton, K., That ‘internet of things’ thing, RFID journal, 22,7 (2009) 97-114.

5. López-de-Armentia, J., Casado-Mansilla, D. ve López-de-Ipina, D., Fighting against vampire appliances through eco-aware things, 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2012, 868-873.

6. Fraser, Q.S.-. http://www.cl.cam.ac.uk/coffee/qsf/timeline.html The Story of the Trojan Room Coffee Pot A Timeline. Oct, 10, 2019.

7. Postscapes, http://postscapes.com/internet-of-things-definition Internet of Things Definition- Postscapes. Oct, 10, 2019.

8. http://www.cisco.com/web/solutions/trends/iot/overview.html Cisco Systems Internet of Things (IoT) - Cisco Systems. Oct, 10, 2019.

9. IBM, https://www.youtube.com/watch?v=sfEbMV295Kk The Internet of Things - YouTube. Oct, 11, 2019.

10. http://handle.itu.int/11.1002/1000/11559-en?locatt=format:pdf&auth ITU Telecommunication Standardization Sector ITU-T Recommendation database. Oct, 13, 2019.

11. Wu, M., Wireless Communication Applications in Internet of Things, 2017.

12. Jaradat, M., Jarrah, M., Bousselham, A., Jararweh, Y. ve Al-Ayyoub, M., The internet of energy: smart sensor networks and big data management for smart grid, Procedia Computer Science, 56 (2015) 592-597.

13. Hancke, G.P. ve Hancke Jr, G.P., The role of advanced sensing in smart cities, Sensors, 13,1 (2013) 393-425.

14. Cheng, J. ve Kunz, T., A survey on smart home networking, Carleton University, Systems and Computer Engineering, Technical Report SCE 09-10, (2009).

15. Abowd, G.D. ve Mynatt, E.D., Designing for the human experience in smart environments, Smart environments: technologies, protocols, and applications, (2004) 151-174.

16. Rashvand, H.F. ve Calero, J.M.A., Distributed sensor systems: practice and applications, John Wiley & Sons, 2012.

17. Pantoni, R., Fonseca, C. ve Brandão, D., Energy Efficiency-The Innovative Ways for Smart Energy, the Future Towards Modern Utilities, Street lighting system based on wireless sensor networks, INTECH Open Access Publisher, 337-352, 2012.

18. Baldus, H., Klabunde, K. ve Muesch, G., Reliable set-up of medical body-sensor networks, European Workshop on Wireless Sensor Networks, 2004, 353-363.

19. Schreier, G., Pervasive Healthcare via “The Internet of Medical Things”, Graz, Austria: Austrian Institute of Technology GmbH, (2010).

20. Boumaiza, Y., Authentication in IoT: 6tisch secure join using distributed storage for the key establishment, Karadeniz Technical University, 2019.

21. Truong, C.D., Routing and Sensor Searach in the Internet of Things, University of Lübeck, 2014.

22. Kuo, F.F., The aloha system, ACM SIGCOMM Computer Communication Review, 25,1 (1995) 41-44.

23. Glover, B., Bhatt, H. ve Essentials, R., O'Reilly Media, Inc., Sebastopol, USA, (2006).

24. Finkenzeller, K., RFID handbook [electronic resource]: fundamentals and applications in, (2003).

25. Committee, I.L.M.S., IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Computer Society, (2011).

26. Mohit Yadav, K.K., A Survey of Bluetooth and ZigBee WPAN Standards, IJEDR, (2016).

27. Wang, C., Jiang, T. ve Zhang, Q., ZigBee network protocols and applications, Auerbach Publications, 2016.

28. Jiang, M., Lee, J., Jeong, K., Cui, Z., Kim, B., Hwang, S. ve Choi, Y.J., A data stream-based, integrative approach to reliable and easily manageable real time

environmental monitoring, International Journal of Distributed Sensor Networks, 11,12 (2015) 914612.

29. Koubâa, A., Alves, M. ve Tovar, E., Lower protocol layers for wireless sensor networks: a survey, (2005).

30. Akyidiz, I., Su, W., Sankarasubramaniam, Y. ve Cayirci, E., Wireless sensor networks: A survey computer networks, The International Journal of Computer and Telecommunications Networking, (2002) 393-422.

31. http://www.eee802.org/15/pub/TG4.html. IEEE Task Group 4 (TG4) IEEE 802.15 WPAN Standard. Oct, 16 2019.

32. Feng, Z., Design and implementation of an experimental platform for performance analysis in wireless sensor networks. 2011.

33. Andersen, Ø.L., Security of Internet of Things Protocol Stacks, NTNU, 2016.

34. Görmüş, S., Aydın, H. ve Ulutaş, G., Security for the internet of things: a survey of existing mechanisms, protocols and open research issues, Journal of the Faculty of Engineering and Architecture of Gazi University, 33,4 (2018) 1247-1272.

35. Sheng, Z., Yang, S., Yu, Y., Vasilakos, A.V., McCann, J.A. ve Leung, K.K., A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities, IEEE Wireless Communications, 20,6 (2013) 91-98.

36. Barthel, D., Pister, K., Dejean, N., Vasseur, J. ve Kim, M., Routing metrics used for path calculation in low-power and lossy networks, 2012.

37. Ibrahimova, D., Quality of Service for IETF 6TiSCH Protocol, Karadeniz Tchnical University, 2018.

38. Watteyne, T., Tuset-Peiro, P., Vilajosana, X., Pollin, S. ve Krishnamachari, B., Teaching communication technologies and standards for the industrial IoT? Use 6TiSCH!, IEEE Communications Magazine, 55,5 (2017) 132-137.

39. Mavialp, http://www.mavialp.com/en/product/detail/3/ Alp Protocol Stack A protocol stack for next generation IoT applications. Oct, 20 2019.

40. Najmuldeen, M. ve Gormus, S., A Generic Clique Based Distributed Caching Approach For Wireless IoT Networks, 2019 4th International Conference on Computer Science and Engineering (UBMK) 2019: 1-5.

41. Zameel, A., Najmuldeen, M. ve Gormus, S., Context-Aware Caching in Wireless IoT Networks, 2019 11th International Conference on Electrical and Electronics Engineering (ELECO) 2019: 712-717.

42. Assila, B., Kobbane, A. ve El Koutbi, M., A survey on Caching in 5G Mobile Network, WCOS, 16 21-22.

43. Rossini, G. ve Rossi, D., Evaluating CCN multi-path interest forwarding strategies, Computer Communications, 36,7 (2013) 771-778.

44. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H. ve Braynard, R.L., Networking named content, Proceedings of the 5th international conference on Emerging networking experiments and technologies, 2009, 1-12.

45. Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Crowley, P., Papadopoulos, C., Wang, L. ve Zhang, B., Named data networking, ACM SIGCOMM Computer Communication Review, 44,3 (2014) 66-73.

46. Yan, H., Gao, D., Su, W., Foh, C.H., Zhang, H. ve Vasilakos, A.V., Caching strategy based on hierarchical cluster for named data networking, IEEE Access, 5 (2017) 8433-8443.

47. Naeem, M., Ali, R., Kim, B.-S., Nor, S. ve Hassan, S., A periodic caching strategy solution for the smart city in information-centric Internet of Things, Sustainability, 10,7 (2018) 2576.

48. Pahl, M.-O., Liebald, S. ve Wüstrich, L., Machine-Learning based IoT Data Caching, 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2019, 9-12.

49. Rahel, S., Jamali, A. ve El Kafhali, S., Energy-efficient on caching in named data networking: A survey, 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech) 2017, 1-8.

50. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M. ve Steggles, P., Towards a better understanding of context and context-awareness, International symposium on handheld and ubiquitous computing 1999, 304-307.

51. Baldauf, M., Dustdar, S. ve Rosenberg, F., A survey on context-aware systems, International Journal of Ad Hoc and Ubiquitous Computing, 2,4 (2007) 263-277.

52. Zhang, G., Li, Y. ve Lin, T., Caching in information centric networking: A survey, Computer Networks, 57,16 (2013) 3128-3141.

53. Meddeb, M., Dhraief, A., Belghith, A., Monteil, T. ve Drira, K., How to cache in ICN-based IoT environments?, 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA) 2017, 1117-1124.

54. Xylomenos, G., Ververidis, C.N., Siris, V.A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K.V. ve Polyzos, G.C., A survey of information-centric

networking research, IEEE Communications Surveys & Tutorials, 16,2 (2013) 1024-1049.

55. Choi, J., Han, J., Cho, E., Kwon, T. ve Choi, Y., A survey on content-oriented networking for efficient content delivery, IEEE Communications Magazine, 49,3 (2011) 121-127.

56. Fang, C., Yu, F.R., Huang, T., Liu, J. ve Liu, Y., A survey of energy-efficient caching in information-centric networking, IEEE Communications Magazine, 52,11 (2014) 122-129.

57. Korupolu, M.R., Plaxton, C.G. ve Rajaraman, R., Placement algorithms for hierarchical cooperative caching, Journal of Algorithms, 38,1 (2001) 260-302.

58. Hachem, J., Coded Content Caching for Wireless Networks, UCLA, 2017.

59. Podlipnig, S. ve Böszörmenyi, L., A survey of web cache replacement strategies, ACM Computing Surveys (CSUR), 35,4 (2003) 374-398.

60. Cao, P. ve Irani, S., Cost-aware www proxy caching algorithms, Usenix symposium on internet technologies and systems, 1997, 12, 193-206.

61. Katsaros, K., Xylomenos, G. ve Polyzos, G.C., MultiCache: An overlay architecture for information-centric networking, Computer Networks, 55,4 (2011) 936-947.

62. Shanmugam, K., Golrezaei, N., Dimakis, A.G., Molisch, A.F. ve Caire, G., Femtocaching: Wireless content delivery through distributed caching helpers, IEEE Transactions on Information Theory, 59,12 (2013) 8402-8413.

63. Meddeb, M., Dhraief, A., Belghith, A., Monteil, T., Drira, K. ve AlAhmadi, S., Cache freshness in named data networking for the internet of things, The Computer Journal, 61,10 (2018) 1496-1511.

64. Niyato, D., Kim, D.I., Wang, P. ve Song, L., A novel caching mechanism for Internet of Things (IoT) sensing service with energy harvesting, 2016 IEEE International Conference on Communications (ICC), 2016, 1-6.

65. Ali, W., Shamsuddin, S.M. ve Ismail, A.S., A survey of web caching and prefetching, Int. J. Advance. Soft Comput. Appl, 3,1 (2011) 18-44.

66. ElBamby, M.S., Bennis, M., Saad, W. ve Latva-Aho, M., Content-aware user clustering and caching in wireless small cell networks, 2014 11th International Symposium on Wireless Communications Systems (ISWCS), 2014, 945-949.

67. Blasco, P. ve Gündüz, D., Learning-based optimization of cache content in a small cell base station, 2014 IEEE International Conference on Communications (ICC), 2014, 1897-1903.

68. Brodersen, A., Scellato, S. ve Wattenhofer, M., Youtube around the world: geographic popularity of videos, Proceedings of the 21st international conference on World Wide Web, 2012, 241-250.

69. Mares, M.-L. ve Sun, Y., The multiple meanings of age for television content preferences, Human Communication Research, 36,3 (2010) 372-396.

70. Rentfrow, P.J., Goldberg, L.R. ve Zilca, R., Listening, watching, and reading: The structure and correlates of entertainment preferences, Journal of personality, 79,2 (2011) 223-258.

71. Zillmann, D., Mood management through communication choices, American Behavioral Scientist, 31,3 (1988) 327-340.

72. Ragozin, A., http://blog.ragozin.info/2011/10/grid-pattern-proactive-caching.html Data Grid Pattern - Proactive caching. Nov, 8 2019.

73. Wikipedia, http://en.wikipedia.org/wiki/Poisson_distribution Poisson Distribution. Nov, 13 2019.

74. Doane, D.P. ve Seward, L.W., Applied statistics in business and economics, New York, NY: McGraw-Hill/Irwin, 2011.

75. Box, G.E., Jenkins, G.M., Reinsel, G.C. ve Ljung, G.M., Time series analysis: forecasting and control, John Wiley & Sons, 2015.

76. Bakar, N.A. ve Rosbi, S., Autoregressive integrated moving average (ARIMA) model for forecasting cryptocurrency exchange rate in high volatility environment: A new insight of bitcoin transaction, International Journal of Advanced Engineering Research and Science, 4,11 (2017).

CURRICULUM VIATE

Akhtari ZAMEEL was born on the 9th of January 1991 in Chittagong, Bangladesh. After finishing her secondary education from Girl's College Chittagong, Akhtari enrolled in a Bachelor of Computer Science and Engineering department at International Islamic University Chittagong, Bangladesh in 2011 and graduated in 2015. After graduation from November 2015- October 2016 she worked as a Web Programmer at Xponent Info System (Pvt.) Ltd. In 2016, she won a scholarship from Presidency for Turks Abroad and Related Communities (YTB). From 2016-2017 she completed Turkish language preparatory education and started her master's studies in Computer Engineering at Karadeniz Technical University in Trabzon, Turkey. A. ZAMEEL has successfully published her paper titled "Context-Aware Caching in IoT for Wireless Networks" at 11th International Conference on Electrical and Electronics Engineering (ELECO) 2019 conference. She also started working as a project researcher at TUBITAK from November 2019. Akhtari speaks English, Hindi and Turkish very well as well as her mother language Bengali.