

Enhancing Energy Efficiency in IoT Based Application

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By
Utkarsha Singh
(Roll No.: 801731014)

Under the supervision of
Dr. Inderveer Chana
(Professor & Associate Head, CSED)



THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA - 147004

JULY 2019

Certificate

I hereby certify that the work which is being represented in the thesis entitled “*Enhancing Energy Efficiency in IoT Based Application*”, in partial fulfillment of the requirements for the award of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Inderveer Chana* and refers other researcher’s work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other university.



Utkarsha Singh
(801731014)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.



Dr. Inderveer Chana
(Professor, CSED)

Acknowledgement

First of all, I would like to thank God for blessing me with all the strength and resources required to complete this task. I would like to express my deepest gratitude towards my respected supervisor **Dr. Inderveer Chana**, Professor and Associate Head, Computer Science and Engineering Department for guiding and encouraging me to accomplish this task successfully. I am very thankful to my supervisor for her valuable time, patience and sharing experiences as well as knowledge with me which has made this journey rewardable and enjoyable.

I am also very thankful to **Dr. Maninder Singh**, Professor and Head, Computer Science and Engineering Department for providing uncanny guidance and support throughout the preparation of the thesis report.

I am also thankful to the entire faculty team and staff members of Computer Science and Engineering Department for their assistance, cooperation, love and affection.

Last but not the least, I would like to thank my family for their wonderful support and encouragement without which none of this would have been accomplished.



Utkarsha Singh
(801731014)

Internet of Thing (IoT) has been recognized as one of the favorable networking paragons to bridge the space between the physical and internet world. With the emergence of IoT, a gigantic rise has been seen in smart applications. In upcoming years, IoT is going to be ubiquitous. Many of the countries are trying to progress towards the smart cities. For such initiation, billions of sensors will be installed for the implementation of IoT applications which will generate massive amount of data. Such massive amount of sensors, data and devices would cost huge amount of money. In addition to cost of installation, energy consumption by the IoT devices is a prominent area of concern. Although IoT applications in themselves are considered to be very energy efficient, however their own energy consumption is very high. Energy efficiency of IoT would make it the long term technology in the coming years.

Components which are installed for the working of IoT devices require energy in the form of electricity. Most of the IoT applications are battery operated and the power dissipation is mainly dependent on the working and interaction between different components of IoT. Charging of batteries from direct electricity is not a good option. This problem has been resolved by energy harvesting ideas like solar energy. But to serve billions of sensors and other components as well as managing such big data need huge amount of energy and dissipate battery's energy at higher pace. So, it is very important to focus on controlling the growing demand of energy by IoT applications.

This thesis studies the gaps in existing techniques as well as in previously proposed IoT based applications and analyzes them to list different solutions for saving energy in IoT applications. Scheduling of Data Transmission is one of the solutions which has been listed and proposed in this work to enhance the energy efficiency of IoT based application. This solution mainly focuses on the energy consumption during continuous data transmission in IoT systems which keeps the networking device or gateway active all the time and results in drainage of energy continuously. In order to implement Scheduled Data Transmission solution, an IoT based application, named Smart Street Lighting System has been studied as a case study and also implemented. Additional feature Gmail notification on fault detection in the system has also been added in the application to make it more efficient in terms of its usability. The proposed solution aims to reduce the energy consumption in IoT application by transmitting data to the cloud in an energy efficient way.

Table of Contents

Certificate	i
Acknowledgement	ii
Abstract	iii
List of Figures	vi
List of Tables	vii
Chapter 1	Introduction..... 1
1.1.	Overview of IoT..... 1
1.2.	Application Areas of IoT 2
1.3.	Benefits of IoT 4
1.4.	Challenges faced by IoT 5
1.5.	Research Motivation 6
1.6.	Indispensable Energy Stages of IoT..... 7
1.7.	Organization of Thesis..... 9
Chapter 2	Literature Survey..... 11
2.1.	Energy Efficient Techniques for IoT 11
2.2.	IoT Based Applications..... 17
2.1.	Conclusion 24
Chapter 3	Research Gaps and Problem Statement 25
3.1.	Research Gaps..... 25
3.2.	Problem Statement 26
3.3.	Objectives..... 26
3.4.	Methodology 27
3.5.	Conclusion 27
Chapter 4	Proposed Solution 28
4.1.	Workflow 28
4.2.	Energy Demanding Factors in IoT Applications..... 28
4.3.	Energy Efficient Solutions for IoT Applications 31
4.4.	Proposed Solution for Enhancing Energy Efficiency 34
4.4.1.	Data Transmission in IoT..... 34
4.4.2.	Scheduling of Data Transmission in IoT Based Applications 36
4.5.	Conclusion 37
Chapter 5	Case Study: Smart Street Lighting System 38
5.1.	Existing Model..... 38
5.2.	Description of Proposed System..... 39

5.3.	Design of Proposed System	40
5.4.	Conclusion	42
Chapter 6	Implementation and Result.....	43
6.1.	Tools Used	43
6.2.	Experimental Setup	47
6.3.	Implementation of Proposed System	48
6.4.	Result	56
6.5.	Comparative Analysis	61
6.6.	Conclusion	63
Chapter 7	Conclusion and Future Scope	64
7.1.	Conclusion	64
7.2.	Thesis Contribution.....	64
7.3.	Future Scope	65
	List of Publications	66
	References.....	67

List of Figures

Fig. 1.1. The 4 Stage IoT Architecture	1
Fig. 1.2. Application Areas of IoT	3
Fig. 1.3. Stages of IoT.....	7
Fig. 1.4. Thesis Organization.....	10
Fig. 2.1. Phases of Literature Survey	11
Fig. 2.2. Energy Efficient Solutions at Different Stages of IoT.....	14
Fig. 2.3. Ranking of IoT Based Applications	17
Fig. 4.1. Workflow of Proposed Work.....	28
Fig. 4.2. Data Transmission through Gateway.....	35
Fig. 4.3. Scheduling of Data Transmission in IoT Based System.....	37
Fig. 5.1. Design of Proposed System	51
Fig. 6.1. Arduino Uno	43
Fig. 6.2. LDR	44
Fig. 6.3. ESP8266	44
Fig. 6.4. Symbol of Arduino IDE.....	44
Fig. 6.5. LED	45
Fig. 6.6. ThingSpeak- Cloud Platform.....	45
Fig. 6.7. Linking React with ThingHTTP.....	46
Fig. 6.8. Digital Multimeter	46
Fig. 6.9. Initialization of Variables	48
Fig. 6.10. Setting ESP8266 and Wi-Fi Connection	49
Fig. 6.11. Sending Instruction to Establish Wi-Fi and LEDs Output.....	49
Fig. 6.12. Switching Lights On.....	50
Fig. 6.13. Switching Lights Off	51
Fig. 6.14. Converting Integer Data Type into String Data Type.....	51
Fig. 6.15. Scheduling of Data Transmission.....	52
Fig. 6.16. Sending Data on Cloud.....	53
Fig. 6.17. Channel Creation on Cloud Platform.....	53
Fig. 6.18. API Creation on IFTTT	54
Fig. 6.19. ThingHTTP Creation.....	54
Fig. 6.20. React Creation	55
Fig. 6.21. Induction of Fault	55
Fig. 6.22. On State of Lights During Night	56
Fig. 6.23. Off State of Lights During Day	56
Fig. 6.24. Gmail Notification for Fault Detection.....	57
Fig. 6.25. Pattern of Continuously Sent Data on Cloud.....	58
Fig. 6.26. Pattern of Scheduled Transmitted Data on Cloud	59
Fig. 6.27. Comparative Analysis of Energy Consumption in Proposed System.....	62

List of Tables

Table 1.1. Energy Demanding Factors at Different Stages of IoT.....	9
Table 2.1. Comparison of Energy Efficient Techniques in IoT.....	16
Table 2.2. Energy Efficient Techniques for IoT Based Applications.....	23
Table 4.1. Working Stages and Energy Demanding Factors of IoT Applications.....	29
Table 4.2. Energy Efficient Solutions for IoT.....	34
Table 4.3. Power Consumption by Gateway.....	35
Table 6.1. Comparison of Energy Consumption in Proposed System.....	62

With the advent of IoT smart living is no longer a day dream. It advantages people by saving time, cost as well as energy. It has become popular with the instant development of low cost sensors, wireless technologies and new Internet techniques. It is one of the best solutions for automating as well as monitoring and controlling applications in real time. This chapter introduces IoT, its application areas, advantages as well as challenges faced by IoT, motivation for carrying this research and also discusses the indispensable energy stages of IoT.

1.1. Overview of IoT

IoT is an advance form of Machine to Machine (M2M) communication where each object links with another object with no human interference and form a large connected ecosystem of devices [1]. IoT has several definitions. The famous author David Hanes defined IoT as “Connecting the unconnected” [2]. ENISA (European Union Agency for Network and Information and Security) defined IoT as “a cyber-physical ecosystem of interconnected sensors and actuators, which enables intelligent decision making” [3]. IoT is a combination of two words “Internet” and “Thing”. In simpler words it can be defined as thing (device/system) which has some computational intelligence to act accordingly and connect with other things as well as people through internet is an IoT.

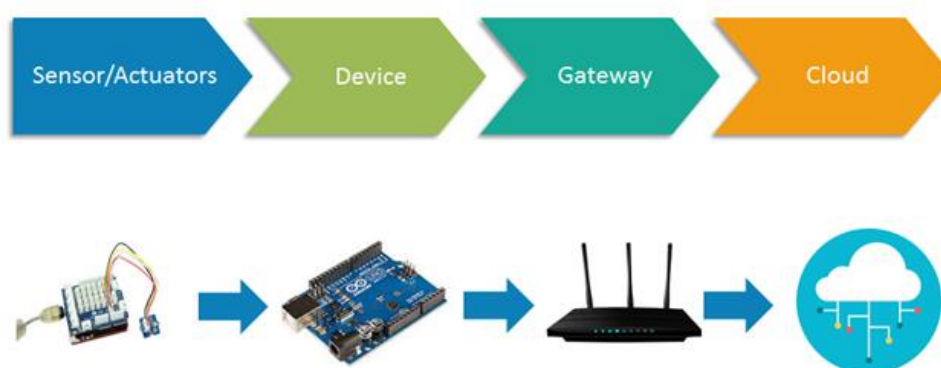


Fig. 1.1. The 4 Stage IoT Architecture [4]

Fig. 1.1 depicts the working of IoT. It represents the intermediate stages of IoT which constitute a complete IoT application. Description of each stage is discussed in subsequent section.

Real time monitoring and controlling of devices has become possible because of IoT. The remote monitoring and controlling of applications not only save time and cost but also saves energy. For example, smart Wi-Fi enabled bulb launched by Syska [5] has the capability to connect with Wi-Fi and enables user to control every aspect of lights from his/her mobile application. In case if user forgets to switch off lights while leaving his home, then he can easily switch them off via mobile. This smart bulb has also the capability to automatically get switch on/off with the sunrise/sunset. There are many more features which are provided in this single bulb. Here the important point is that this smart bulb's automatic and remote monitoring and controlling features helps in reducing energy wastage. The concept of IoT has bestowed many energy efficient applications.

1.2. Application Areas of IoT

IoT has become part of daily life. It is responsible for making things smart in various ways. It is benefitting people at both selective and collective levels. For example, wearables like smart watch which tracks sports activities, sleep, steps etc. in real time helps individual to track his/her fitness [6] at former level. In the latter, admin can monitor and control street lights remotely which light on/off according to sunlight's intensity hence, aid benefits to whole society [7]. IoT has marked its presence in every sphere of life. Fig. 1.2 represents the applications areas of IoT.

Some major application areas of IoT are discussed as follows:

i. Smart Home

It is the most common and popular application among common people. It enables user to control and monitor his home remotely. Its main advantage is reduced physical efforts as well as reduced energy wastage due to its remote monitoring and controlling features. Examples of smart home are smart lightbulb, smart refrigerator, smart lock, smart thermostat etc.

ii. Wearables

These are the smart electronic devices which can be worn on the body as an accessory. The most common wearable is the smart watch. It not only tracks the activity of user but also helps him in managing calls, messages and so on through his watch.

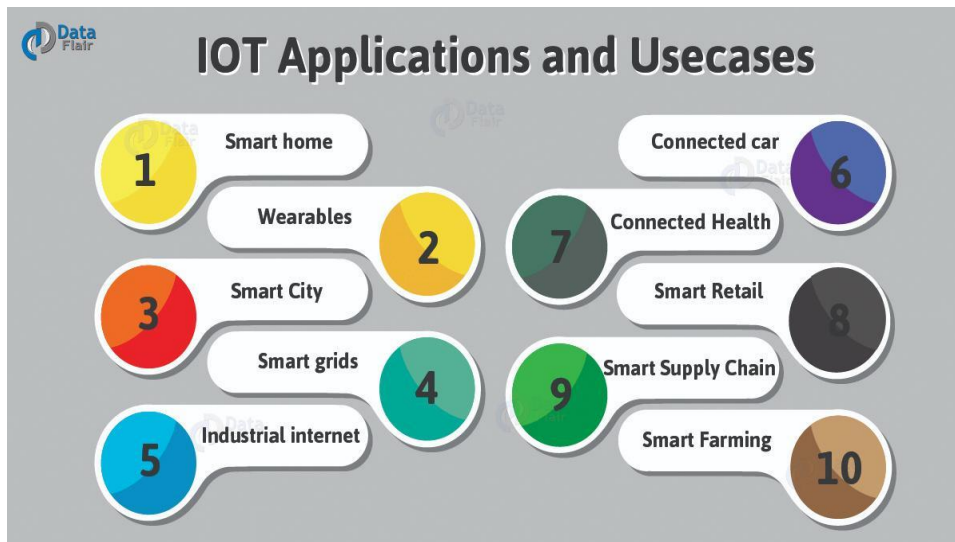


Fig. 1.2. Application Areas of IoT [8]

iii. Smart City

It is the most popular application across the world as most of the countries are focussing to adapt this application area. There are many spheres of city where manual efforts are wasting time, cost and energy. In smart city different day-to-day areas are automated in order to ease life of citizens. It has become an important project of major countries with the aim to automate most of the aspects of city which are being used by its citizens. Examples of smart city applications are smart street lighting system, smart parking system, smart waste management system, smart traffic management system etc.

iv. Smart Grids

It plays an important role in saving electricity. It works with electrical grid in such a way that it responds automatically to the changing electric demand of consumer in order to achieve economically efficient sustainable power system. Distribution of electricity based on consumer's demand plays an important role in saving energy. Smart meter is one of the application of smart grid.

v. Industrial Internet

It incorporates the idea of smart machines with embedded technology and IoT. It connects devices which helps in exchanging data, analysis of data that helps in improving productivity, efficiency as well as economic benefits. CPS (Cyber Physical System), AI (Artificial Intelligence), Machine learning, Big data, Cloud computing plays an important role in achieving Industrial IoT (IIoT).

vi. Connected Health

It is a healthcare management system where healthcare services are provided remotely through internet. Devices based on patient's needs are designed and health related data is shared in such a way that patient can receive care and advices remotely from his doctor. It helps both patient and doctor to track health at each and every time. Examples of connected health are telehealth, sleep therapy devices etc.

1.3. Benefits of IoT

Following are the major advantages of IoT:

i. Reduces manual intervention

The intelligent automation feature of IoT reduces dependency on its users to operate applications. The computational intelligence of IoT applications able to handle the complete IoT device without human intervention. It reduces manual efforts.

ii. Remote monitoring and controlling

IoT applications has the capability to communicate with other devices as well as with people through internet. The sharing of information over the network helps the user to monitor applications remotely. Remote monitoring helps the user to know the status of application and helps in controlling applications accordingly.

iii. Saves time and money

Automation as well as remote monitoring and controlling features of IoT save both time and money. Automation results in reduced man force which saves money as well as time. Remote monitoring and controlling results in accessing application in an effective way which also saves time and money. For example, manual street lights require man power to switch lights on/off which needs money to pay that man power and manual operation of street lights also results in electricity wastage due to the lights left on mistakenly during day time. The automated and remote monitoring and controlling features enable street light to switch on/off automatically which saves manual efforts as well as cost by saving electricity.

iv. Energy saving

Energy efficiency is the most important aspect of IoT. Initially there were applications in the market which were automated because of programmed instructions uploaded onto them and working accordingly without any human intervention. But due to the absence of remote monitoring and controlling part it was resulting in energy wastage too. Primarily automated lighting system was developed to light bulbs based on the presence of people in room, then connected lighting system was introduced to switch on lights for security purposes. If these lights were left on then resulted in energy wastage. Now smart lighting system like LumiFi [9] has introduced the concept of energy efficient IoT applications. It advances the system by offering remote configuring, monitoring, controlling and optimizing lighting environment remotely through a mobile application. If the lights are left on then can be switched off by just a click on phone. Such IoT enabled devices are giving us the experience of smart living. Hence, IoT applications play an important role in saving energy.

1.4. Challenges faced by IoT

There has been growing trend towards the use of smart and connected devices which has made the IoT widespread across the world. Even the industries and government are accepting IoT as a source of saving time, cost and ultimately energy. Many countries have headed towards smart city by deploying IoT based applications to save operational costs as well as energy [10] [11] [12]. However, there are some issues exist with the growing IoT which must be handled effectively in order to make IoT applications more reliable, secure and cost effective. Some of the open challenges are:

i. Security and privacy

The security of massive data generated by billions of IoT devices is one of the greatest concerns. These generated data can be industrial, personal, enterprise or consumer which needs to be secured while transmitting over the network. In IoT data is transmitted from one stage to another which can be tampered by external unwanted efforts. Hence, to maintain the privacy of data It is very important to secure data.

ii. Energy efficiency

It is the highly concerned area of IoT. One of the major benefits of IoT is its energy efficient applications. However, the energy efficient IoT applications also need energy to work. According to IEEE Consumer Electronics Magazine Editor Peter Corcoran

“energy is one of the three determinants of the long term sustainability of Internet of Things (the other two being privacy and security)” [13]. Areas where IoT based systems are developed on larger scale demand huge amount of energy. Energy consumption by increasing IoT devices will adversely impact the energy consumption growth. Hence, it is very important to focus on energy efficiency of IoT itself.

iii. Storing and processing of data

According to Cisco report “500 billion devices are expected to be connected to the Internet by 2030” [14]. It clearly indicates that these devices result in massive data generation i.e. big data and processing and storage of such big data is a challenging task. Data generated every second by IoT devices which is not utilized goes into waste which is of no use. To make the generated data utilized and decisive in different context is the main goal of IoT [15]. Hence, the processing and storage of generated data is an area to focus.

iv. Interoperability and Standardization

In IoT different devices are connected and share information with each other. These devices are designed by different vendors based on different technologies and may not be accessible to other devices. To bridge this gap standardization is required to support interoperability among different IoT devices.

1.5. Research Motivation

Popularity of IoT is growing day by day with the aim of connecting people with devices as well as devices with devices. Each IoT device consists sensors for sensing environment, gateway for transmitting collected data and cloud for managing and storing data. IoT applications aggregate, analyze and deliver insights based on data generated by these connected IoT devices. Both automation and remote access of IoT applications reduce energy wastage. To be energy efficient these applications also need energy for their own operation. From sensing to sharing and storing data requires energy and feeding such billions of devices [14] need enormous amount of energy which is an area of concern. Energy resources are inadequate in nature and using it at such pace can be hazardous in future. At one hand where IoT has been proved as a way to save energy by its energy efficient applications on the other hand its own energy efficiency is at risk.

The motivation for writing this thesis is to address the problem area of high energy consumption with the increasing IoT systems all over the world as well as how energy consumption can be reduced.

1.6. Indispensable Energy Stages of IoT

This section covers the framework of IoT with the objective to represent the requirement of energy at each stages of IoT for its working. Like IoT definition, its framework is also defined in different ways but the base of IoT is same in all those proposed frameworks. Though studying different architectures of IoT [16] four staged framework of IoT is proposed in Fig. 1.3.

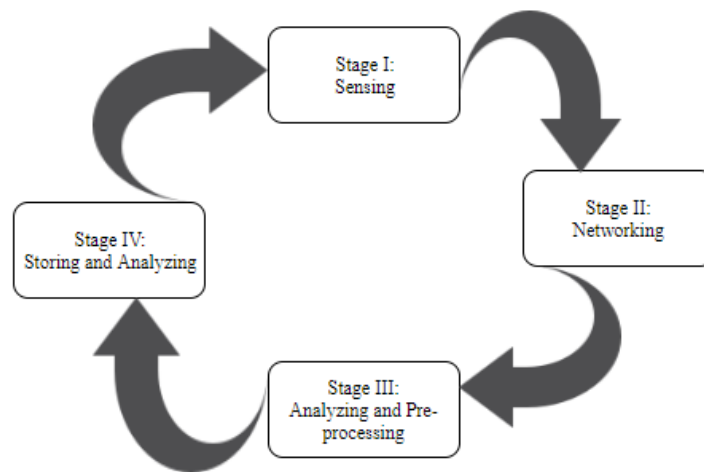


Fig. 1.3. Stages of IoT

This framework represents the workflow of IoT and also explains the energy consumption in IoT at its different stages as follows:

Stage I: This stage is the foundation of IoT. It bridges the physical world and information world. It is responsible for sensing the environment, collecting the sensed data and assisting the devices to act according to the application. It uses different technologies like sensors, radio frequency identification to sense environment and collect data in the form of numbers or images. The components like RFID, sensors, actuators which are used for these purposes are connected to the microcontroller or say embedded system. This embedded system has the uploaded application program written by programmer which automates the devices to act accordingly.

Continuous sensing which makes the IoT system to react and act on sensed data requires energy in continuous manner. Thus, this stage requires 24x7 energy in the form of power to run its components.

Stage II: This layer has two functions: accessing and transmitting the collected data. The collected data from stage I is sent to base station node of next stage. Once base station node receives data it connects to the internet through gateway. Connection can be done either through LAN (Local Area Network) or WAN (Wide Area Network). After establishing connection with internet data can be send to further stages.

Gateway or router which is used for communicating generated data to further stages requires energy. At this stage energy is consumed in powering of gateways or routers as well as transmitting data to further stages.

Stage III: Initially, this stage was not introduced in IoT. This stage is mainly responsible for pre-processing and analysis of collected data to make real time decisions. Computation is performed locally i.e. near the edge of devices through which data is generated or near the edge of network through which data is transmitted. Edge servers make the data-driven decisions faster [17]. This stage does not send all generated data i.e. big data directly to cloud. It keeps pre-processed data with itself for real time analysis and communicate rest of the data to cloud. Once the pre-proceeded data is utilised for required purpose they are send on cloud for further analysis and storage. However, edge computing is an emerging and extended concept of IoT that's why research is going on in this area.

Like stage II edge servers also require energy to work, pre-process and transmit data.

Stage IV: It is last stage where the collected data is stored and analysed. At this stage big data which is generated by IoT devices are analysed (in case of absence of edge computing) for real time decisions as well as for predictions, business values etc. Remote monitoring and controlling of IoT applications are possible because of the cloud where the collected data is stored and accessible to user through mobile applications.

For storing and managing such big data collected from billions of IoT devices data centres (cloud) require huge amount of electricity.

The above mentioned stages are responsible for working of IoT successfully. However, powering of each component of these stages is an important task because without power supply IoT devices are unable to work. Battery is the most commonly source for powering IoT applications. There are many low powered IoT devices which run for longer duration without

charging them on regular basis [6]. Apart from this there are many high powered IoT applications like smart city which require continuous power supply for their working. Energy consumption in such applications is very high which results in power dissipation of battery at higher rate. Charging of batteries on regular basis at such scale consume lots of energy. Energy consumption of IoT applications is very high and going to rise in future which will be perilous.

Table 1.1. Energy Demanding Factors at Different Stages of IoT

Stages	Components	Energy Demanding Factors
Stage I: Sensing	Sensors/Actuators, RFID, Camera, Embedded System	<ul style="list-style-type: none"> • Charging of batteries. • Continuous sensing.
Stage II: Networking	Gateway, LAN (Wi-Fi/Ethernet) / WAN (GSM/5G)	<ul style="list-style-type: none"> • Requires power for working of gateways/LAN/WAN. • Continuous sending of data on cloud. • Sending huge amount of data on cloud. • Transmission of data for prolonged time because of inefficient routing and protocols.
Stage III: Analysing and Pre-processing	Edge nodes in the form of servers, gateway	<ul style="list-style-type: none"> • Powering of edge servers.
Stage IV: Storing and Analysing	Data Centre i.e. cloud	<ul style="list-style-type: none"> • Storing and processing of big data requires lots of energy.

Many research work has been done to reduce energy consumption of IoT systems. However, implementation of those proposed ideas and solutions are rare in real time. It has been noticed that energy is not only consumed by the sensing stage of IoT for powering its components and running of IoT devices but also consumed at other stages while transmitting, processing, storing, analysing data. Table 1.1 represents the energy demanding factors responsible for energy consumption at each stage of IoT. These factors are very common which are mostly not considered while implementing IoT applications. If any of the above mentioned factors is controlled in IoT systems huge amount of energy can be saved.

1.7. Organization of Thesis

Rest of the thesis is organized in the following order:

Chapter 2: This chapter includes a detailed survey of energy efficient solutions proposed for IoT as well as applications based on IoT. Since this study has focused on IoT based

applications, it analyzes the techniques which have been implemented to save energy in those applications. Further it groups the techniques into different stages of IoT on the basis of the stages where they are saving energy.

Chapter 3: This chapter covers the research gaps in the surveyed work on the basis of which problem statement is defined. Objectives and methodologies are also covered under this chapter.

Chapter 4: This chapter proposes solutions to solve the defined problem. Workflow which has been followed to implement the solution is also presented.

Chapter 5: This chapter contains the case study of an IoT application- Smart Street Lighting System. This case study is used for implementation purpose.

Chapter 6: This chapter mainly contains the implementation details as well as results for showing energy saving by proposed solution in implemented system.

Chapter 7: This chapter gives conclusion and future scope for the proposed work.

Fig 1.4 represents the topics covered in each chapter of thesis.

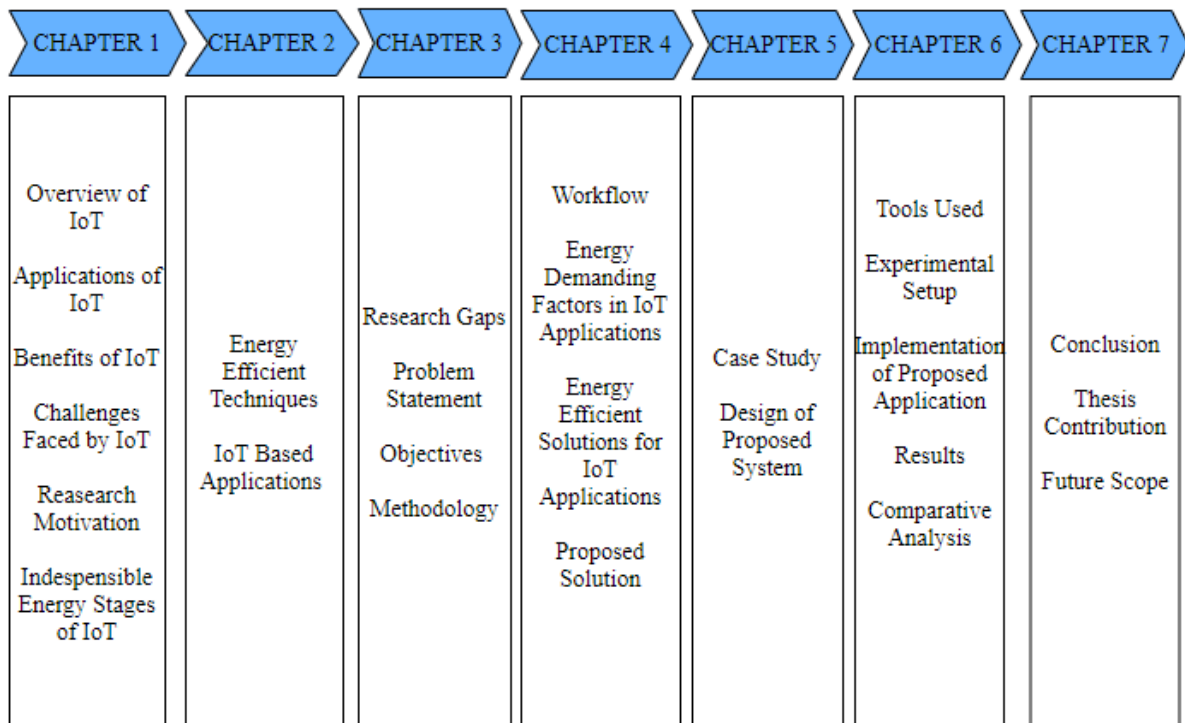


Fig. 1.4. Thesis Organization

The main aim of this literature survey is to study the previous work done in the direction of energy efficiency of IoT and also to realize the future work that needs to be done by gaining insights from it. Literature Survey is done in the following two phases:

Phase 1: In this phase the papers related to ideas as well as techniques which have been implemented in order to save energy in IoT are surveyed.

Phase 2: In this phase IoT application related papers are surveyed in order to find how many techniques or ideas have been implemented in IoT applications to make them more energy efficient.

Fig. 2.1 represents the phases of literature survey.

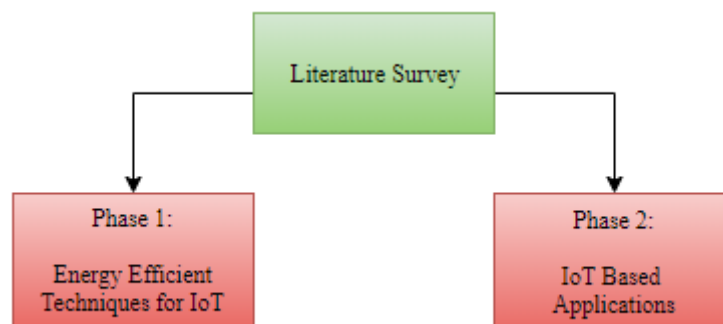


Fig. 2.1. Phases of Literature Survey

2.1. Energy Efficient Techniques for IoT

IoT is a combination of both hardware and software and its energy efficiency depends on both. Various advancements have been done at hardware part to make it energy efficient [18] [19] [20] [21]. This section mainly covers previously proposed software techniques/solutions to save energy in IoT. Fig. 2.2 depicts the proposed techniques at different stages of IoT.

One of the solutions proposed is sleep scheduling algorithm. The authors in [22] proposed a system model in which hardware components like sensors, appliances are linked with embedded web server which in turn hosts RESTful web service to interact with cloud server for virtualization of objects. Also embedded web server is responsible for scheduling sensor devices. They have introduced an energy efficient sleep scheduling algorithm to make sensor

devices in sleep mode when they are in idle state and react only when some request occurs. The three core stages: On-duty, Pre-off duty and Off duty are designed to schedule duty cycle of sensors also proved that Off-duty is most energy saving state by practically evaluating each stage's performance. This improves the energy consumption of sensor nodes.

The theory of compressed sensing is used for data aggregation in Wireless Sensor Networks (WSNs)- an internal part of IoT so that redundant data is never acquired in the network, hence reduces energy dissipation. A method presented in [23] utilizes compressed sensing technique and dictionary learning to achieve energy efficient data aggregation in WSN. In this paper algorithms are proposed for multiplexing and demultiplexing of multiple and different sensor data. The idea is to reduce the multiplexed compressed data vector length of sensors by using dictionary learning which results in low transmission data length.

The authors in [24] integrated proposed sleeping scheduling scheme [25] using a tree and energy aware routing protocol. Broadcast tree is constructed and each node in the network is able to determine its parent. If a node detects any event it turns into active node, otherwise in sleep mode. Periodic reconstruction of broadcast tree ensures balanced energy consumption by all the nodes.

Problems related to limited energy, battery power and processing power of IoT devices and how they are affecting communication among them are mentioned in [26]. Also an energy efficient communication protocol is proposed at network layer for IoTs and comparison is made with distributed cluster computing energy efficient routing (DCEER) to prove its efficiency in terms of residual energy and number of routing rounds.

In [27] an overview of green IoT is provided along with the challenges faced due to extreme usage of energy thirsting IoT devices. Five principles: Reduced Network size, Use Selective Sensing, Use Hybrid Architecture, Policy Making and Intelligent Trade Offs are proposed to achieve Green IoT. Case study is done on smart phones which is an important facet of IoT and provided brief overview for enhancing the ongoing actions to make IoT greener in near future.

The authors in [28] presented a concise outline of energy management and problems in smart cities. Classified the energy management in smart cities into two types: energy efficient solutions and energy harvesting operations. Also provided combined architecture for energy efficient optimization and scheduling of smart cities. Two detailed case studies are also done and their simulation results signify the effect of energy efficient scheduling optimization and wireless power transfer on the performance of IoT in smart cities.

A survey is done in [29] and its main objective was to impart insights for network designers and policy makers in deploying energy efficient IoT applications. In this survey importance of fog computing and how it is beneficial for achieving energy efficient IoT is discussed in detailed manner. Architecture of fog and cloud computing is presented to understand the working of both fog and cloud computing in IoT. Authors have also discussed the parameters that are affecting the energy consumption of IoT network and devices. Parameters for greening IoT are also listed as well as work done in this area is also mentioned. In fog computing, computation is performed near the edge of the devices. For this local computation nano data centers are distributed at local server. These distributed nano data centers consume less energy in comparison to centralized data center [30]. New energy models are developed for shared and unshared networks and experiments are carried out to show better energy efficiency of fog computing in comparison to cloud computing.

An energy efficient architecture is presented for Industrial IoT (IIoT) in [31]. This is a three-layer architecture containing sense entities layer, network layer and cloud server layer. The sense entities layer is further classified into three layers i.e. sense layer, gateway layer and control layer. The goal of further classified layers is to save energy and balance traffic loads by disallowing direct interaction between sense nodes and making gateway nodes to act as relay nodes. Data from sense nodes is sent to gateway nodes and from gateway nodes to control nodes. Control nodes send aggregated data to the network and also responsible for allocating sense nodes to specific gateway nodes for sending data. Also proposed a novel sleep scheduling and wake up protocols to improve energy efficiency of presented architecture. The idea is to switch sense nodes to sleep mode when they are at low energy state and continues in sleep mode until a wake up signal sent by the gateway node is received.

Multi-hop and cooperative multi-hop routing are proposed in [32] for optimizing energy in WSN (Wireless Sensor Network) communication to base stations. These routing methodologies reduce the energy consumed in extensive path transmission by routing the generated data in hop-to-hop level. By using Pruned Adaptive Routing protocol to prune network requests and making parallel communication among the sensors in [33] routes are optimized to the target node which results in reduced energy consumption in IoT network.

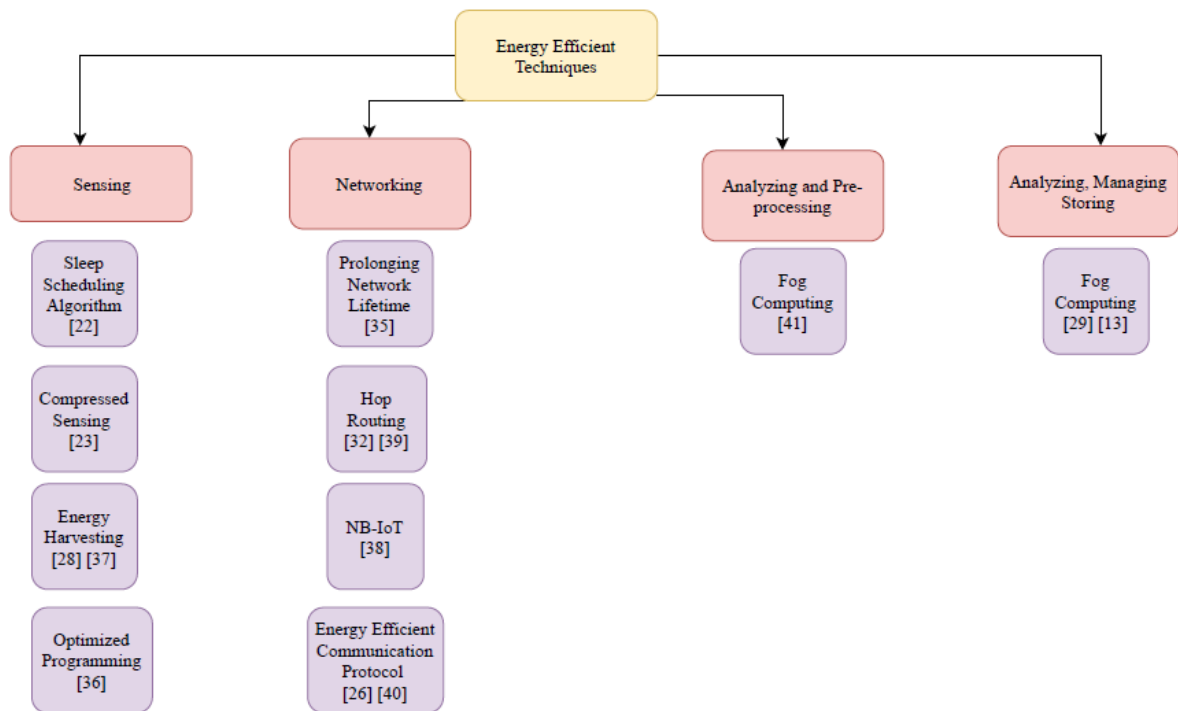


Fig. 2.2. Energy Efficient Solutions at Different Stages of IoT

The challenges related to big data which is generated by IoT devices are discussed in [34]. IoT and big data concepts are studied and presented in this paper with the main attention on data velocity and volume reduction without affecting the value and variety of data. How big data is generated at each stage of IoT and consuming energy are explained. Existing energy efficient approaches are discussed for data velocity and volume reduction. Few techniques like data filtering, compression, dynamic monitoring frequency and their tradeoffs are also reviewed.

Deployment of WSNs in IoT is a complex task and most of the deployment schemes cannot be implanted directly in IoT. Hence implementing green deployment schemes for IoT is a challenging task. To address this issue a hierarchical framework for general IoT deployment is presented in [35]. In this framework network lifetime is prolonged by disallowing direct communication between sensing nodes and allowing direct communication between relay nodes which shift the traffic load from sensing nodes to relay nodes. An optimization problem is created on the basis of energy consumption, link flow balance and system budget to model green IoT. This problem is solved by using clustering principle i.e. MECA and tree algorithm Steiner which facilitates green deployment of IoT. Lastly, a numerical experiment is done to present the better performance of proposed work with other deployment schemes for WSNs in IoT.

As discussed earlier that the program which is written and uploaded on embedded system is responsible for running the IoT application and has the acme command over hardware. Hence, its role is very crucial in optimizing energy of system. In [36] the problem of how software distresses the energy consumption of a system is discussed. Need and realization of how energy transparency in software development can handle the IoT energy challenge is explained. Also the requirements for the energy aware software development as well as the existing techniques and their limitations are listed.

Charging of batteries for running IoT applications is a big issue in IoT because it requires huge amount of electricity. Apart from electricity energy harvesting is a good option. Authors in [37] represented the concept of “Internet of Energy Harvesting Things”. This is a two staged energy maximization process. In first stage energy from different sources are collected by harvesters and in second stage all the collected energy from different sources are combined so that overall energy is utilized by the sensor nodes to work. This concept can solve the problem of dependency on single energy source which is utilized for harvesting.

The concept of NB-IoT, its architecture and its energy efficiency features was discussed in [38]. NB-IoT is proved as one of the green initiatives for IoT. Its LPWA technology can save a large amount of energy. It discusses how LTE networks are useful for deploying NB-IoT in networks. It also represents three options for deploying NB-IoT i.e. standalone, guard band and in-band based on spectrum’s range.

Authors in [39] introduced long hop scheduling algorithm to optimize energy usage of WSNs. This algorithm prevents re-transmission of messages coming from far distances by routing them at first prior. The prevention in re-transmission increases the network lifetime. This reduces energy consumption during transmitting packets. It also reduces the number of re-transmitted packets which helps in effective data transmission.

Network protocols play very important role in saving energy. A node level energy efficient routing protocol was proposed in [40]. This protocol solves the problem of high energy consumption by intermediate nodes while routing packets from sensing nodes to head node because of limited transmission range. Proposed protocol uses the residual energy of its nearest hop as well as average value of residual energy of all nodes present in the network. It helps in forwarding packets by using minimum number of hops which reduces energy wastage.

Fog computing is gaining popularity in terms of energy efficiency in IoT. In [41] two technologies: fog computing for local computation and microgrids for renewable power

resources were used to save energy in IoT. An IoT gateway was developed to make decisions about where to run IoT applications- cloud or fog based on local power and weather forecast awareness. Energy efficiency of fog computing is also presented in [42]. Quality of fog and cloud computing as well as networking orchestration services in 5G mobile was evaluated in terms of energy efficiency. During evaluation it was proved that more distant smart devices requires more energy from the 3G and 2G serving RAN in comparison to 5G RAN. It also suggested that cloud in 5G networks can be diffused into fog to enhance virtual network mobility.

The different solutions discussed above have different objectives. Table 2.1 compares and contrasts the previously proposed techniques.

Table 2.1. Comparison of Energy Efficient Techniques in IoT

Author	Year	Technique	Improved Parameters	Application Specific
P. Bai, K. Kumar et al. [26]	2018	Energy efficient communication protocol	Residual energy, Number of routing rounds	No
O. B. Akan, O. Centinkaya et al. [37]	2018	Energy harvesting	Reduces dependency on single source for charging batteries	No
F. Jalali, S. Khodadustan et al. [29]	2017	Fog computing	Power consumption of servers	Yes
P. Sethi and S. R. Sarangi [32]	2017	Multi-hop and cooperative multi-hop routing	Energy consumption of WSN (Wireless sensor network)	No
L. Farhan, A. E. Alissa et al. [39]	2017	Long hop scheduling algorithm	Network lifetime, energy consumption during transmission of data	No
S. Kitanov and T. Janevski [42]	2017	Fog computing	Distant communication of smart devices	No
F. Jalali, K. Hinton et al. [30]	2016	Fog computing	Energy consumption centralized data center	No
K. Wang, Y. Wang et al. [31]	2016	Sleep scheduling and wakeup protocol	Energy consumption of sensor nodes	Yes
M. Amarlingam, P. K. Mishra [23]	2016	Compressed sensing, Dictionary learning	Data aggregation	No
V. M, K. SPR et al. [40]	2016	Node level energy efficient routing protocol	Energy consumption of intermediate nodes	No
F. Jalali, A. Vishwanath et al. [41]	2016	Fog computing, microgrids	Decision making	Yes
S. F. Abedin, M. G. R. Alam et al. [22]	2015	Sleep scheduling algorithm	Energy consumption of sensor nodes	No
C. Estevez and J. Wu [28]	2015	Energy efficient scheduling optimization, wireless power transfer	Battery life	Yes

J. Huang, Y. Meng et al. [35]	2014	Prolonging of network lifetime, clustering principle	Green deployment scheme for WSN	No
S. D. Dwivedi and P. Kaushik [24]	2012	Sleep scheduling scheme [25], Energy aware routing protocol	Energy consumption of network nodes	No
S. M. A. Otefy, F. M. Al-Turjman et al. [33]	2012	Pruned adaptive routing protocol	Energy consumption of routing routes to target nodes	No

2.2. IoT Based Applications

IoT has wide variety of applications in different areas like smart city, smart energy, smart home and so on. In a survey IoT applications are ranked on the basis of real time projects [43]. According to survey smart city has topped the list followed by industrial IoT (IIoT) and smart buildings. Similarly, IoT applications are also ranked on the basis of people’s interest [44] in which smart home has topped the list followed by smart wearables and smart city. Fig. 2.3 depicts the top five ranked applications based on both real-time projects as well as user’s interest. On analyzing these rankings, it has been noticed that smart city is always among the top rankings which clearly signifies that smart city applications are growing at larger scale. This will ultimately result in high energy consumption for running such huge number of IoT devices. However, these rankings can differ with the change in the number of real time projects and people’ interest.

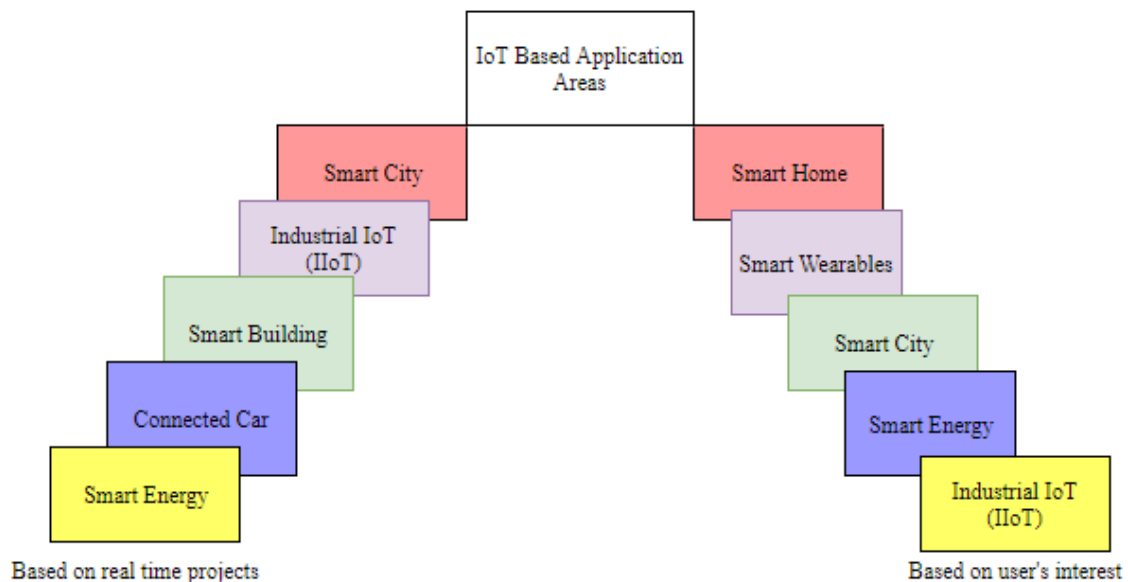


Fig. 2.3. Ranking of IoT Based Applications

Based on Fig. 2.3 smart city and smart home are at top, so in this section these two applications are surveyed. Smart city and smart home are huge areas of IoT applications therefore their sub

applications are surveyed. An extensive survey is done on smart street lighting system, smart parking system and smart refrigerator with the motive to discover their energy efficiency.

Smart Street Lighting System

In [45] intelligent street lighting system is developed for smart cities. By the use of light and IR (infrared) sensors street lights are automatically switched on/off. Dimming of streets lights is done in night time according to traffic intensity based on dimming scheduling to save energy. Also programmable street signs are used to notify public about real time traffic through Xbee module. Similarly, automated street light system is developed in [46] which get switched on/off according to sunlight intensity detected by LDR (light dependent resistor) sensor. This system also senses temperature and humidity of surrounding with the help of DHT11 (temperature humidity) sensor. Data like status of lights, temperature, humidity are sent to cloud through ESP8266 (Wi-Fi module) in real time which helps in remote monitoring of lights.

Like [45] street lighting system based on IoT was proposed [47]. In this system street light controller are placed on street lights. Lights are switched on/off automatically due to LDR sensor. During night time the lights are switched on/off according to object's movement on roads detected by IR sensor. The light data is sent to base station through Wi-Fi to monitor the system. This complete system can be controlled either manually or automatically.

The idea of dynamic control of street lights through controllers is implemented in [48]. In this street lights are deployed with sensors which sense environment data and controllers which collect data from different sensors and send them on server. Web application can monitor lights remotely and access the sensed data. It dynamically controls the lighting by sending command to controllers to either switch on/off the street lights. Authors in [49] developed an energy efficient smart street lighting system using Zigbee technology because of its low power transmission. Each street light is assigned with unique identity so that they can be monitored remotely in an easy way. Sensors are attached to lights which sense the data and send to MCU to take appropriate actions. The data is also send to control centre for remote monitoring and fault detection.

Cost Analysis of smart street lights for smart cities is done in [50] on the basis of proposed heuristics. Efficiency of each heuristic is defined in terms of high, medium and low. Among all heuristic best is considered for the implementation of street lights. Lights work on the basis of occupancy control strategy i.e. lights light with full intensity if the presence of object in a particular radius of lights is detected during night time, otherwise dim for rest of the

time. SE 10 PIR motion sensor is used to detect object's presence during night. Cost analysis in terms of capital and operation is done to show comparison between different heuristics.

The concept of NB-IoT is used in [51] for low power wide area communication as it has the lower terminal power consumption. The single streetlight controller and monitor is acted as terminal node which controls switching on/off of street lights using sensors and collecting and sending data to the cloud through NB-IoT network. Management platform is used for analysis of real time data for light controlling purposes, fault detection, locating street lights, big data analysis and so on.

Fog computing is an emerging paradigm which is preferred as one of the energy efficient solutions for IoT [29] [30]. By using the concept of fog computing and NB-IoT (Narrowband IoT) SSL system is implemented [52]. Intelligent street lamps are developed which adjust its brightness and communicate with server based on fog computing in real-time through NB-IoT and also communicate with managers about the breakage of lamps information using Wi-Fi automatically. Managers who are monitoring street lamps can send commands to servicemen using mobile application or web browser to rectify the problem of breakage of lamps.

Recharging of batteries to run such huge number of IoT devices is challenging. Using electricity directly to recharge those batteries is not a good option with the view point on energy. This problem is resolved in [53]. Solar panel is used to utilize solar energy to charge batteries for running street lighting system. Street lights are made to switch automatically on during night time due to low intensity of environment light which is detected by LDR sensor and off during day time due to high intensity of light. By the use of IR sensor lights are dimmed during night according to presence of traffic. Also the data from Arduino is displayed on the web browser using internet.

Smart Parking System

In [54], mobile application was designed for users to use smart parking system. Sign-in is required for the first time users who are using that application. After sign-in user is redirected to Google OAuth flow to select Google Account so that the information of user like username, email, google ID is stored for further reference. After sign-in and authorization of user nearest parking location is returned to user based on user's current location. At parking locations ultrasonic sensors are embedded and RFID tags with Arduino and Arduino is connected to Raspberry Pi. These sensors sense the obstacles at every 5 seconds and indicates whether the

parking slot is free or not. Raspberry Pi sends free parking slots at every 5 seconds in Parking Logs collection. If parking slot is free a response is sent to user where user is navigated using Google Maps, otherwise “Not found” response is sent to user. For the first time user RFID card is assigned to him so that his entry/exit timings can be stored in cloud to calculate the bill for parking and mail the invoice and bill to the user.

In [55], Ultrasonic sensors are used for sensing free parking slot in parking lot. ESP32 LoRa transmitters are embedded near parking lots. The sensed data i.e. flag = 1 indicating parked and flag = 0 indicating vacant are sent to ESP32 LoRa receiver through multiple transmitters. The information related to status of parking like slot name and slot status is sent to cloud i.e. IBM Watson as well as local Wi-Fi server to access information on android phone by user.

A three layered architecture is proposed for smart parking system’s working in smart cities [56]. The first layer is sensing layer which senses the parking slot occupancy using ultrasonic sensors. These sensors are connected to ESP8266 which is used as Wi-Fi module to access web services for acquiring IDs for new parking slots and communicating with context broker changing the occupancy status in agreement with the sensor readings. In middleware layer, processing and storage are performed. Registration of new spots are done by receiving introduction messages on web services. NoSQL database is used for storage purposes. New ID is generated if there is new parking spot and that ID is used to communicate with context broker. Orion Context Broker is used for real time updates. The last layer i.e. application layer through which users can have access to parking system using their phone. Wirecloud is used as application mashup through which different widgets can be combined and accessed at a time. User can view parking occupancy in form of Google Maps showing different colours for different parking status using android application.

For advanced driver assistance smart car parking is implemented [57]. IR sensors are connected to raspberry pi which detect movement of vehicles and send information of cloud accordingly. User can access cloud to get parking lot information by sign-in/login to mobile application. After login parking status is provided to user according to which user can book free parking slot by entering user ID, slot number, date and time which is automatically updated in cloud. Once user books the slot no other user is allowed to book that slot. Booking ID is generated during booking slot which is used for authentication of user during parking vehicle. During sign-out two options are available to know charges of parking: temporary charges to

know estimated cost of parking before leaving and actual charges of parking which is only generated after leaving parking area. I-SPARK an IoT based smart parking system [58] is developed to access parking slot information online when user opens the webpage designed for parking system. Webpage displays both vacant (represented by green) and occupied (represented by red) slots because of ultrasonic sensors embedded at parking slots. These ultrasonic sensors are connected with Arduino and with the help of internet shield data regarding slot occupancy is sent on cloud in real time. Also information about ambience and temperature of parking area sensed by LMR 32 sensor is sent on cloud which is only accessible by admin who can accordingly adjust light and sprinklers.

A prototype of smart parking system is presented in [59]. It uses CCTV nodes to detect the vacant parking slots. Existing CCTV cameras are used as IoT cams to cover parking area and by using algorithm like blob detection and techniques like transform Matrix, image processing etc. to monitor and report changes in defined grid of polygons (i.e. retrieves the coordinates/frames from streaming video). Retrieved data indicates vacant or occupied areas which is sent on parking area server which aggregates data and updates database accordingly in real time. This parking area information is accessible by user who has registered himself on the mobile application to view free parking area. A complaint button is also available on app for user to get another parking area if previously booked area is mistakenly accessed by some other person.

Smart Refrigerator

Now-a-days smart refrigerator is very common. Many smart fridges have been introduced with multiple features. Refrigerator in itself needs power 24x7 for their operation and adding features including IoT results in more power consumption.

Smart refrigerator was developed with the aim to detect rotten food and notify user accordingly [60]. Sensing module consists of MQ3 sensor to detect gases produced by the foods kept inside the refrigerator and DHT11 sensor to monitor temperature and humidity of the refrigerator. Sensing module is connected with control module which consists microcontroller i.e. WeMos D1 R2 connects all sensors and makes them work accordingly. Also it connects them to WiFi so that data can be send on cloud. Transmission module sends data like gas readings, tempearture and humidity on cloud i.e. Thingspeak as well as push notifications to notify user about the rotten food detected by MQ3 sensor and also about temperature and humidity.

In [61], smart fridge is developed which is controlled by the smart intelligent system. DHT11 sensor senses temperature and humidity inside the fridge, FSR-402 sensor senses the weight of the product kept inside the fridge and LDR sensor senses light intensity of fridge. LDR activates horn if the fridge door is opened for longer duration (more than one minute). DHT11 sends temperature and humidity data to user to help him to keep track of power cut as well as number of times fridge is opened. FSR-402 adds the food item into the shopping cart as per change in their weight indicating out of stock. All these data are sent on cloud through Wi-Fi at every 5 minutes. User can access these data through mobile application. User can view last recorded temperature and humidity data as well their historic data also. Graphical representation of temperature indicated power cut in last hour. Shopping cart is visible to user. A low cost refrigerator was proposed in [62]. Two web cameras are installed inside and outside the fridge. Both are triggered by light sensor and distance measuring sensor respectively. Inner image of fridge i.e. food inside the fridge is captured when door of fridge is opened which results in change of light intensity which is sensed by light sensor. Outer image of the fridge i.e. food input by the user is based on user's choice. If user wants the status of food which he is going to keep inside it then he simply keeps his food in front of distance-measuring camera which helps in preventing accidental triggering of outside camera. All the food data in the form of images captured by both cameras are sent on server in real time. User can access those images and latest update time by using mobile application.

In [63], RFID technology is used to detect the presence of food kept inside the fridge. RFID kit is triggered at every 15 minutes to scan previously and newly stored food items. It fetches item ID and checks in global database to find its details like expiration date so that updates are stored in database as well as necessary notifications can be send to user accordingly. Essential Inventory list is designed which is accessed by user to add essential items which he wants to buy on frequent basis. If the essential food item is marked as removed item by RFID then it is accordingly added to shopping list. User can access inventory and shopping list using his mobile phone.

Table 2.2. lists those techniques which have been particularly implemented in proposed IoT applications to reduce energy consumption.

Table 2.2. Energy Efficient Techniques for IoT Based Applications

IoT Based Applications	Proposed Applications	Year	Techniques for Improving Energy Efficiency
Smart Street Lighting System	Intelligent Street Lighting for Smart Cities [45]	2018	Dimming scheduling in night time
	IOT Based Street Light Monitoring System [49]	2018	Zigbee
	The Smart Street Lighting System Based on NB-IoT [51]	2018	NB-IoT
	IoT Based Solar Smart LED Street Lighting System [53]	2018	Solar Energy
	IOT Based Smart Street Light Management [46]	2017	None
	Smart Street Lighting System using IoT [47]	2017	Switching of lights on/off based on object's movement during night
	IOT Based Dynamic Control of Street Lights for Smart City [48]	2017	None
	Cost Analysis of Smart Lighting Solutions for Smart Cities [50]	2017	Occupancy control strategy
Smart Parking System	Smart Parking System using Cloud based Computation and Raspberry Pi [54]	2018	None
	An IoT based Smart Parking System using LoRa [55]	2018	None
	I-SPARK: IoT based Smart Parking System [58]	2018	None
	IoT-Based Smart Parking for Smart Cities [56]	2017	None
	IoT Based Sensor Enabled Smart Car Parking for Advanced Driver Assistance System [57]	2017	None
Smart Refrigerator	The Implementation of IoT Based Smart Refrigerator System [60]	2018	None
	SmartFridge: The Intelligent System that Controls your Fridge [61]	2018	None
	Low-Cost Smart Refrigerator [62]	2017	None
	Smart Refrigerator: A next generation refrigerator connected to the IoT [63]	2016	Sensing Scheduling

With time, many IoT systems have been developed with the aim to automate as well as monitor and control applications remotely to save time, efforts and energy. From the above survey, it is clear that IoT reduces manual efforts on large scale which in turn helps in saving energy. However, the energy efficiency of these applications is not discussed in terms of power consumption. Each of the components of IoT needs energy for their operation and according to report by Cisco [14] IoT devices are going to be increased in billions which will stress the energy resources. Hence, it is very important to consider energy efficiency prior while implementing IoT devices at larger scale where energy consumption is sky-high.

2.1. Conclusion

This chapter has compared and contrasted the previously proposed energy efficient solutions for IoT and also analyzed that very few of the techniques are implemented in IoT based applications.

Research Gaps and Problem Statement

The above literature survey consists of various energy efficient techniques for IoT. However, there are some gaps exist in previously proposed work which are addressed in this section. On the basis of research gaps, problem statement is formulated in this chapter.

3.1. Research Gaps

Many of the researchers have proposed and implemented their ideas, framework and algorithms for the energy efficiency of IoT, however, there exists some gaps in their work:

- There is a need to develop an adaptive adjusting period of tree reconstruction in [24] on the basis of input data rate which can further increase the network lifetime and ensure decreased energy consumption.
- By directing straight communication between two sensor nodes and using gateway nodes as relay nodes, and the use of sleep and wake up protocols the proposed system [31] saves energy, however further work needs to be done in the direction of continuous sensing by gateways which results in energy wastage.
- Transmission of data over the network plays an important role in energy consumption. Efficient routing of data [32] [33] [39] [40] to save energy in IoT by efficient data transmission saves energy, though no discussion is made on the energy consumption by continuous data transfer over the network. Energy consumed by continuous data transfer is missed which needs to be focused.
- Instead of generating a methodology according to application domain, only a single technique is applied for all kinds of applications in the existing work.

It has been clear that automation and real time monitoring and controlling of IoT applications make them energy efficient. However, there exists some gaps in previously proposed IoT applications as mentioned below:

- None of the IoT applications focus on the problem of continuous sensing and continuous transmission of data on cloud which results in high energy consumption.
- Directly sending huge amount of data on cloud consumes lots of energy which is missed in proposed applications and need to be focussed in IoT based applications to save energy.

3.2. Problem Statement

Based on literature survey it is clear that energy efficiency of IoT is one of the area of utmost concern and unavoidable in upcoming years. To make IoT more energy efficient it is very essential to know which part or phase of IoT is consuming energy and what are the factors responsible for the increased energy usage at both hardware and software level. In above survey of IoT applications, very few energy efficient techniques are implemented to save energy in IoT [45] [47] [49] [51] [53] apart from automation. From sensing to sending data on cloud IoT needs a lot of energy which results in high power dissipation. Thus, it is very important to control energy consumption at intermediate steps of IoT so that the energy dissipation of power source can be reduced.

Applications which are deployed at larger application areas require enormous amount of energy for their working. Automation saves huge amount of energy but the factors like continuous sensing, continuous sending of data, storing and processing of big data also consumes a lot of energy at the same time. At present energy harvesting is utilized as one of the solutions for saving energy in IoT based applications. To further reduce the energy consumption, there is a need to investigate the intermediate stages of IoT applications where energy consumption is high and can be controlled by proposing and implementing energy efficient solutions.

3.3. Objectives

Following are the primary objectives of carrying this thesis:

- To study the energy consumption pattern of IoT based applications.
- To analyze the existing energy efficient techniques of IoT based applications.
- To propose and design an energy efficient technique for IoT application.

3.4. Methodology

To accomplish the above mentioned objectives following steps are followed:

- Understand the IoT framework to identify the energy demanding stages of IoT.
- Study various research papers of proposed techniques to save energy in IoT based applications.
- Propose energy efficient solution for enhancing the energy efficiency of IoT applications.
- Implement ‘Smart Street Lighting System’ to find its energy efficiency and also implement proposed solution in that application to enhance its energy efficiency by saving energy in it.

3.5. Conclusion

In this chapter research gaps inferred from exhaustive literature survey have helped in formulation of problem statement. Next chapter discusses the possible solution to solve the problem.

As discussed earlier automation as well as remote monitoring and controlling are obligatory features of IoT which helps in saving energy. With the demand of increasing IoT devices to develop large scale applications like smart city the demand of energy is also increasing. To enable such large applications with reduced energy consumption is an important task. Therefore, energy efficiency of IoT applications in itself is an area of concern. Hence, It is very important to focus on the solutions through which energy consumption can be reduced in IoT applications. This chapter proposes solution for increasing the energy efficiency of IoT applications by reducing energy consumption.

4.1. Workflow

Based on problem statement workflow is presented in Fig. 4.1 to provide solution to the defined problem. This workflow represents the step by step process which have been followed to find the energy consumption areas in IoT applications as well as the solutions to reduce the energy consumption.

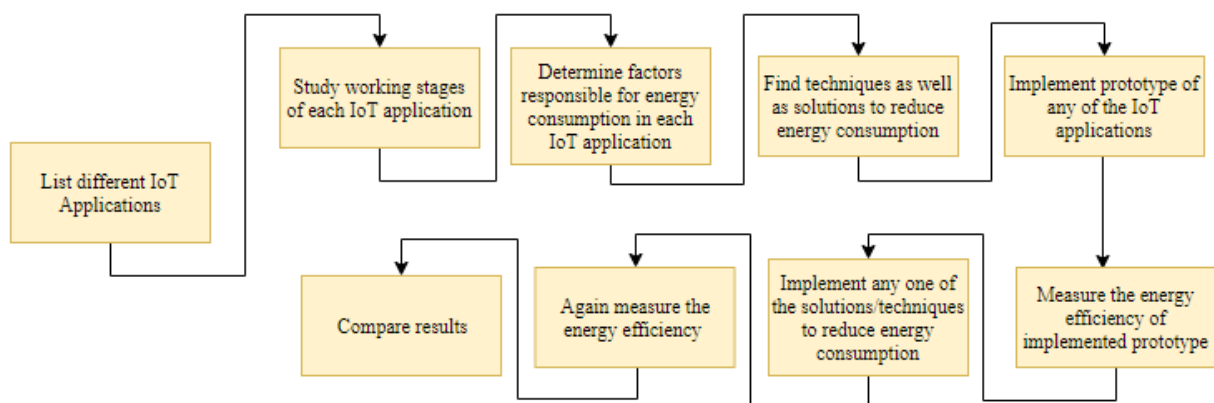


Fig. 4.1. Workflow of Proposed Work

4.2. Energy Demanding Factors in IoT Applications

After studying framework of IoT in chapter 1 factors responsible for draining energy in it are discovered. Previously proposed IoT based applications have also those energy drainage factors which have been observed from studying their working stages. Table 4.1 enumerates the working stages of proposed IoT applications which results in energy consumption.

Table 4.1. Working Stages and Energy Demanding Factors of IoT Applications

IoT Applications	Paper	Working Stages	Energy Demanding Factors
Smart Street Lighting System	Intelligent Street Lighting for Smart Cities [45]	<ul style="list-style-type: none"> • By the use of different sensors street lights are automatically switched on/off. • Dimming of streets lights is done in night time according to traffic intensity based on dimming scheduling to save energy. • Also programmable street signs are used to notify public about real time traffic through Xbee module. 	<ul style="list-style-type: none"> • Continuous sensing
	IOT Based Smart Street Light Management [46]	<ul style="list-style-type: none"> • Automated street lights are developed which get switched on/off according to sunlight intensity detected by LDR. • Also sense temperature and humidity of surrounding. Data like status of lights, temperature, humidity are sent to cloud which helps in remote monitoring of lights. 	<ul style="list-style-type: none"> • Continuous sensing • Continuous transmission of data on cloud
	Smart Street Lighting System using IoT [47]	<ul style="list-style-type: none"> • Street light controller are placed on street lights. Lights are switched on/off automatically due to LDR sensor. • During night time the lights are switched on/off according to object's movement on roads detected by IR sensor. • The light data is sent to base station through Wi-Fi to monitor the system. This complete system can be controlled either manually or automatically. 	<ul style="list-style-type: none"> • Continuous sensing • Continuous transmission of data on cloud
	IoT Based Dynamic Control of Street Lights for Smart City [48]	<ul style="list-style-type: none"> • Street Lights are deployed with sensors which sense environment data and controllers which collect data from different sensors and send them on server. • Web application can monitor lights remotely and access the sensed data. It dynamically controls the lighting by sending command to controllers to either switch on/off the street lights. 	<ul style="list-style-type: none"> • Continuous sensing
	IOT based street light monitoring system [49]	<ul style="list-style-type: none"> • Energy efficient smart street lighting system is developed by using ZigBee technology. • Each street light is assigned with unique identity so that they can be monitored remotely in an easy way. • Sensors are attached to lights which sense the data and send to MCU to take appropriate actions. • The data is also send to control centre for remote monitoring and fault detection. 	<ul style="list-style-type: none"> • Continuous sensing • Continuous transmission of data on cloud
	Smart Parking System using Cloud based Computation and Raspberry Pi [54]	<ul style="list-style-type: none"> • After sign-in and authorization of user nearest parking location is returned to user based on user's current location. • At parking locations ultrasonic sensors are embedded and RFID tags with Arduino and Arduino is connected to Raspberry Pi. • These sensors sense the obstacles at every 5 seconds and indicates whether the parking slot is free or not. Raspberry Pi sends free 	<ul style="list-style-type: none"> • Continuous transmission of data on cloud

		parking slots at every 5 seconds in Parking Logs collection. If parking slot is free a response is sent to user where user is navigated using Google Maps, otherwise “Not found” response is sent to user.	
Smart Parking System	An IoT based Smart Parking System using LoRa [55]	<ul style="list-style-type: none"> • Ultrasonic sensors are used for sensing free parking slot in parking lot. ESP32 LoRa transmitters are embedded near parking lots. • The sensed data i.e. flag = 1 indicating parked and flag = 0 indicating vacant are sent to ESP32 LoRa receiver through multiple transmitters. The information related to status of parking like slot name and slot status is sent to cloud i.e. IBM Watson as well as local Wi-Fi server to access information on android phone by user. 	<ul style="list-style-type: none"> • Continuous sensing • Continuous transmission of data on cloud
	IoT-Based Smart Parking for Smart Cities [56]	<p>Working stages are developed in a three layered architecture:</p> <ul style="list-style-type: none"> • Sensing layer senses the parking slot occupancy using ultrasonic sensors. These sensors are connected to ESP8266 which is used as Wi-Fi module to access web services for acquiring IDs for new parking slots and communicating with context broker changing the occupancy status in agreement with the sensor readings. • In middleware layer, processing and storage are performed. Registration of new spots are done by receiving introduction messages on web services. NoSQL database is used for storage purposes. • In application layer users can have access to parking system using their phone. Wirecloud is used as application mashup through which different widgets can be combined and accessed at a time. 	<ul style="list-style-type: none"> • Continuous sensing • Continuous transmission of data on cloud
	IoT Based Sensor Enabled Smart Car Parking for advanced driver assistance system [57]	<ul style="list-style-type: none"> • IR sensors are connected to raspberry pi which detect movement of vehicles and send information on cloud accordingly. User can access cloud to get parking lot information by sign-in/login to mobile application. • After login parking status is provided to user according to which user can book free parking slot by entering user ID, slot number, date and time which is automatically updated in cloud. • Once user books the slot no other user is allowed to book that slot. Booking ID is generated during booking slot which is used for authentication of user during parking vehicle. 	<ul style="list-style-type: none"> • Continuous sensing
	The Implementation of IoT Based Smart Refrigerator System [60]	<ul style="list-style-type: none"> • Sensing module consists of MQ3 sensor to detect gases produced by the foods kept inside the refrigerator and DHT11 sensor to monitor temperature and humidity of the refrigerator. • Sensing module is connected with control module which consists microcontroller i.e. WeMos D1 R2 connects all sensors and 	<ul style="list-style-type: none"> • Continuous transmission of data on cloud

Smart Refrigerator		<p>makes them work accordingly. Also it connects them to Wi-Fi so that data can be send on cloud.</p> <ul style="list-style-type: none"> • Transmission module sends data like gas readings, temperature and humidity on cloud i.e. Thingspeak as well as push notifications to notify user about the rotten food detected by MQ3 sensor and also about temperature and humidity. 	
	<p>SmartFridge: The Intelligent System that Controls your Fridge [61]</p>	<ul style="list-style-type: none"> • DHT11 sensor senses temperature and humidity inside the fridge, FSR-402 sensor senses the weight of the product kept inside the fridge and LDR sensor senses light intensity of fridge. • LDR activates horn if the fridge door is opened for longer duration (more than one minute). DHT11 sends temperature and humidity data to user to help him to keep track of power cut as well as number of times fridge is opened. FSR-402 adds the food item into the shopping cart as per change in their weight indicating out of stock. • All these data are sent on cloud through Wi-Fi at every 5 minutes. • User can access these data through mobile application. User can view last recorded temperature and humidity data as well their historic data also. Graphical representation of temperature indicated power cut in last hour. Shopping cart is visible to user. 	<ul style="list-style-type: none"> • Continuous sensing

4.3. Energy Efficient Solutions for IoT Applications

The suggested solutions are inferred from the factors which are responsible for draining energy in IoT. These solutions mainly focus on the basic aspects of IoT which are often missed while implementing IoT applications and results in energy wastage. Following are the solutions which can be implemented in IoT based applications to reduce their energy consumption:

A. Combination of multiple sensors

Sensor is an elemental part of IoT. Different types of sensors are available in market which are utilized in different IoT applications. Depending on applications sensors vary. For example, LDR sensor is used where application is based on light, IR sensor is used where application is based on tracking objects and so on. It is possible to deploy different types of sensors in same application to make it more efficient in terms of energy. If one sensor is controlling application, then other sensor can be scheduled for some particular time depending on application in such a way that it can control the already controlled application to save more energy. For example, LDR sensor is used in street lights to automatically switch lights on/off based on sunlight's intensity. Also

IR sensor is used to switch lights on during night only when movement is detected on roads. Thus, combination of sensors saves more energy. Furthermore the deployed sensors can be scheduled by using previously proposed techniques [22] [31] in order to reduce energy consumption by continuous sensing.

B. Scheduling of data transmission

Sending sensed data on cloud helps in remote monitoring of IoT based systems. Though sending data continuously on cloud drains lots of energy. Transmitting data continuously through gateway requires energy as gateway consumes more energy while transmitting data. When the amount of data is huge then energy consumption by gateway increases further. Applications where continuous transmission of data as well continuous monitoring is not required periodic data transfer can be a good way to save energy. In periodic data transfer data transmission is scheduled at some particular time interval so that data can be send on cloud periodically i.e. at fixed time interval. Hence, energy consumption of gateway is lowered during the time when no data is sent on cloud through the gateway.

C. Data filtration

Data redundancy is most common among the areas where big data is generated. IoT devices generate such massive amount of data which is a combination of both redundant and essential data. Directing all the data on cloud directly consumes lots of energy for their transmission. Data filtration can be proved to be useful technique to filter redundant data and send only useful data on cloud. Consequently, the data size will be reduced and transmission of reduced data can save energy at some extent. This technique can be implemented at different stages of IoT. The sensed data can be filtered at sensing stage while programming the application. Some filters and library have been created to filter noisy data [64]. Also the data can be filtered at IoT gateway i.e edge. Gateway is not only about communication between devices and cloud. Installed software in gateway plays different roles including filtering data which helps to decide which data is redundant or noisy and accordingly filter that data and send them on cloud [65]. Hence the size of filtered data is less in comparison to original data which requires less energy while transmission. Also, it saves energy at cloud stage by storing and processing only filtered data.

D. Efficient programming

This technique is applicable at each and every stage of IoT to make it energy efficient. All other mentioned techniques are based on this approach. From the very first stage of IoT efficient programming plays an important role in saving energy. While working on hardware part i.e. embedded system (microcontroller) which controls the whole IoT application developers have the ability to control energy consumption of the whole system. Based on the software on which they write code they can save energy by utilizing energy saving modes, interrupt, sleep-wake up protocols and so on efficiently.

E. Local Computation

Its relevance is already discussed in chapter 1. Beside this edge computing has also the capability to save energy in IoT. It doesn't transfer all the data received from device to the cloud at a time. It keeps the data along with it i.e. local device for processing and analytics purposes which results in reduced backhaul data traffic to central data centre [66]. The above mentioned data filtration technique is a part of local computation. Only portion of data can be send on cloud which requires storage and analysis in long run. This helps in saving energy while sending massive data directly to the cloud.

Fog computing is the extended form of edge computing which helps in saving energy in IoT [41] [42].

F. Application based real and no-real time analysis and processing of data

IoT helps in taking real time actions based on real time access of real time data by user. Real time processing and analysis of data drains both energy and time. It may be possible that real time processing and analysis of data is not required in IoT. Other features of application can help user in real time decisions. Based on application non-real time analysis on real time data can be performed weekly or monthly to observe patterns basically in case of cause analysis to take appropriate actions. This non-real time analysis saves energy in real time.

Table 4.2 lists the above mentioned energy efficient solutions as well as solutions inferenced from the literature survey.

Table 4.2. Energy Efficient Solutions for IoT

Stages	Energy Demanding Factors	Energy Efficient Solutions
Stage I: Sensing	<ul style="list-style-type: none"> • Charging of batteries • Continuous sensing • Generation of redundant data 	<ul style="list-style-type: none"> • Efficient programming • Combination of multiple sensors • Data filtration • Energy harvesting [37] • Scheduling algorithm [22]
Stage II: Networking	<ul style="list-style-type: none"> • Continuous sending of data on cloud. • Sending huge amount of data on cloud directly. • Transmission of data for prolonged time because of inefficient routing and protocols. 	<ul style="list-style-type: none"> • Efficient communication protocol [40] • Efficient routing [39] • Scheduling of data transmission • Data filtration • NB-IoT [38]
Stage III: Analysing and Pre-processing	<ul style="list-style-type: none"> • Requires energy for running edge servers. 	<ul style="list-style-type: none"> • Local computation with Fog computing [41]
Stage IV: Storing and Analysing	<ul style="list-style-type: none"> • Storing and processing of such big data requires lots of energy. 	<ul style="list-style-type: none"> • Fog computing [42] • Application based real and non-real time analysis

With the deployment of billions of IoT devices number of sensors as well as data has also been increased at high extent. Handling of data is one of the major concerns in IoT applications because generated data helps in gaining insights which helps in real time monitoring and controlling. Big data generated from IoT devices need more energy while transmitting, processing and storing them to cloud. Even a byte of data requires energy while transmitting from gateway to cloud and continuous sending of data in IoT devices on cloud consumes energy for transmission [67]. Hence, It is very crucial to understand that how transmitting of data can be done so that energy consumption by IoT applications will be reduced.

4.4. Proposed Solution for Enhancing Energy Efficiency

As discussed earlier that energy consumption of IoT application also depends on the way data is sent on cloud. If transmission of data is done in such a way that IoT system consumes less energy when no data is sent on cloud, then energy consumption can be reduced. But it is very important to understand that how energy is consumed while transmission of data in IoT.

4.4.1. Data Transmission in IoT

Internet is the foremost part of IoT which helps in connecting device with device as well as device with people. IoT systems use medium to transmit data generated from IoT devices to people and that medium is the gateway. Gateway is the medium which acts as an entry and exit

point for a network which is responsible for passing all the data through it prior to being routed. Fig. 4.2 illustrates the data transmission in IoT applications through gateway. In IoT applications, number of sensors are deployed for the working of IoT devices. These sensors generate data which is transmitted to cloud where it gets processed. That processed data is accessed by user through an application or cloud based on the type of application. Each data should be transmitted through gateway to reach cloud. Thus, each IoT system uses gateway as a medium to transfer data to cloud.

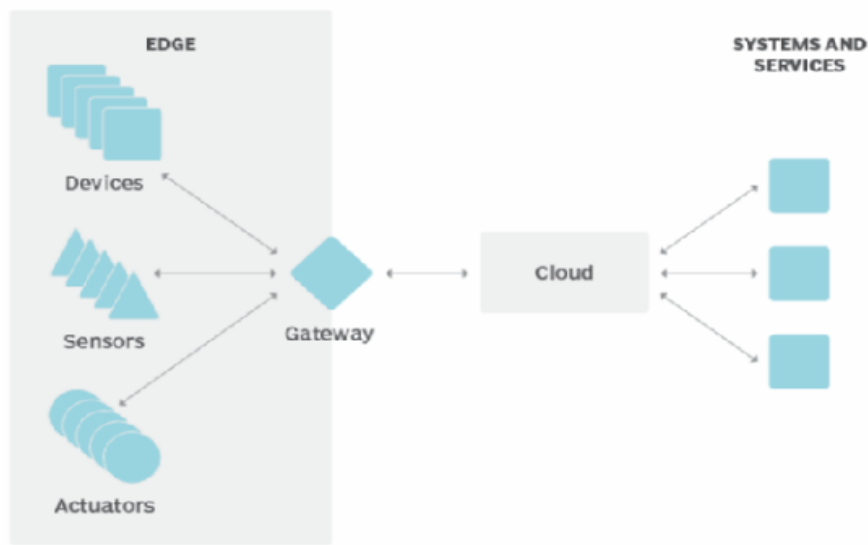


Fig. 4.2. Data Transmission through Gateway [68]

Wi-Fi is also one of the mediums in IoT systems to transfer data from IoT devices to cloud. As discussed earlier in chapter 1 that each component of IoT requires power for their working. Wi-Fi module also requires power for their working. Batteries supply power to the whole IoT system. But the amount of current varies when data starts transmitting from the gateway or router to cloud. Gateway device draws less current i.e. consumes less power when no data is transmitted through it. But when gateway device is in active state i.e. transmitting data through it, then it draws more current i.e. consumes high power as shown in Table 4.3.

Table 4.3 Power Consumption by Gateway

	Mode	Data Transmission	Current Drawn (Less/More)	Power Consumption
Gateway	Active	Yes	More	High
	Ideal	No	Less	Low

It is clear that when data is transmitted from routing device to cloud it is consuming more energy in comparison to when no data is transmitted through it. Continuous sending of data in IoT systems deployed at larger scale drains lots of energy while transmission.

4.4.2. Scheduling of Data Transmission in IoT Based Applications

Many techniques have been proposed in IoT for the efficient transmission of data through the network to reduce energy consumption. However, scheduling of data transfer i.e. at what time data should be transmitted has not been discussed in previous work. Data transfer also depends on the type of IoT application. If there is a life critical application, then data needs to be monitored continuously. But if there is an application where continuous monitoring is not required and additional features make the monitoring and controlling more easy then scheduling of data transfer can be a good strategy to save energy in IoT applications.

Scheduling of data transfer schedules the transmission of data at a particular time interval. This means that the data generated from the sensors should not be transmitted continuously on cloud hence, make the routing device ideal for some time which results in low power consumption. When the scheduled time interval arrives then only generated data at that time gets transmitted through routing device to cloud which makes the routing device active for a short period of time. Only for that period power consumption will be high and rest of the time it will remain low.

Fig. 4.3 represents the flow of data in an IoT system at defined time interval. Defined time is the time when data needs to be transmitted to cloud depending on programmer's and application's choice. If the condition $\text{Time_interval} = \text{Defined time}$ is met, then only data is transmitted through the gateway to cloud.

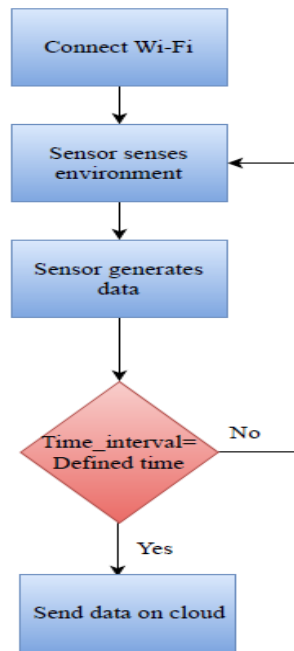


Fig. 4.3. Scheduling of Data Transmission in IoT based System

4.5. Conclusion

The proposed scheduling of data transmission solution is specifically for IoT based applications where continuous monitoring of data is not required. Hence, to evidence the energy saving through scheduled data transmission in IoT an IoT application has been implemented. Next chapter discusses the case study of smart street lighting system as well as design of the proposed system.

Case Study: Smart Street Lighting System

This chapter discusses case study which is based on smart city application i.e. smart street lighting system. The aim of this case study is to depict the existing systems of smart street light in order to represent the existing solutions being implemented to save energy in them. In contrast to existing systems, prototype of smart street lighting system is proposed with the aim to save more energy by implementing proposed scheduling data transmission solution into it.

5.1. Existing Model

In real time, automated street lights play vital role in the development of smart city. Few countries have entered in the generation of smart city by implementing real time smart applications at large scale [69] [70] [10] [71]. Street lights are one of those applications which have been deployed at such scale. Automation of street lights results in huge energy savings. To further save energy dimming of street lights during night [10] and solar energy is used as a solution to charge batteries in small application areas [72]. Following are some real time smart street lighting system that exist in real world:

i. Twilight

It is a Dutch startup company which is using the concept of IoT to make city lighting more energy and cost efficient. It deploys motion sensors in the existing street lights which results in on-demand motion-based illumination. The streetlights automatically adjust their brightness on detecting the presence of vehicles or pedestrians. For real time monitoring and management of lights web based platform is used. This system has witnessed 50 to 70 percent energy saving and cut down electricity bill by 2 billion EUR per year. Currently some trial projects at Amsterdam's Schiphol airport, Seoul and some cities of Germany are going on to adopt this system [73].

ii. Comlight

It is a Norwegian company which has developed world's first patented control system "Motion Sensing Street Lighting System" for reducing energy wastage [10]. Street lighting control system is installed with Eagle Eye radar which senses all those activities which require optimal lighting on the road. When movement is detected by the nearest street light, it gets lit and the detection is communicated to other nearest street lights

wirelessly by radio for providing full light and also get dim when no movement is detected. Eagle Eye Gateway Unit is used for remote access of street lights for monitoring purposes along with GPS [74].

iii. Telensa PLANet

It is the world's number one outdoor lighting control system which has deployed nearly 1.5 million street lights around the world. It is a connected street lighting system consisting of wireless control nodes, UNB (Ultra Narrow Band) wireless network and CMS (Central Management System). It saves energy by controlling amount of light on the basis of usage and by accurately measuring every watt. It helps in saving huge amount of energy [75].

iv. Motwane

It is an India based R&D and IoT company. Its IoT based smart street lighting solutions as JUSCO's project in Jamshedpur is the biggest smart street lighting deployment project in India where 300 smart street lights were deployed. In partnership with TATA Communications Motwane is paving way to deploy 15,000 smart street lights in India. Street lights get switched on/off or dimmed remotely from control center. Lights can also be adjusted based on the location which results in reduction of energy consumption and manpower costs [76].

The above mentioned real time applications are saving energy by adding the features of IoT i.e. automation as well as remote monitoring and controlling of lights. Also, dimming of lights based on detection of vehicles or pedestrians is one of the enhanced feature to save more energy in street lights. However, none of the applications focuses on the way the street lights sharing data on cloud. Huge amount of data is generated from these street lights and continuously sending them on cloud for monitoring purpose drains lots of energy which is not required. Hence, to witness more energy saving in IoT based street lighting system scheduling of data transmission can be a proficient solution.

5.2. Description of Proposed System

Smart street lighting system which has been implemented during thesis is an automated system where lights switched on/off according to sunlight's intensity. Status of lights are send on cloud platform for remote monitoring and controlling purposes. Automation along with remote monitoring and controlling makes the whole system an IoT system. This proposed system is

based on the previous proposed work however the difference lies in the way that how data is sent on cloud. In previous work status of street lights are send on cloud continuously for remote monitoring purpose. But in real time, application area is wide due to which huge number of sensors are deployed at large scale which generates huge amount of data results in energy drainage during continuous transmission.

In the proposed system, the proposed solution i.e. scheduling of data transmission has been implemented which is discussed in chapter 4. Transmission of data is scheduled at an interval of 1 hour so that during that duration only gateway remain active consuming more power and rest of the time it remains ideal consuming less power. Code for scheduling of data transmission is done just before the code of sending the data on cloud so that only last sensed data can be send on cloud in order to reduce data size and congestion while transmission. Also, to make the more monitoring easier Gmail notifications are send to admin if fault is detected. By mutation testing fault is induced at software level i.e. code is changed in such a way that lights switch on during day time and off during night time. To identify fault Light_status is set as 50. If this kind of fault is detected Gmail notification is sent to admin to inform about fault detection. This feature reduces the need of continuous monitoring of lights and indirectly helps in saving energy by helping admin to take real time decisions as soon as possible.

5.3. Design of Proposed System

Fig. 5.1 represents the design of proposed street lighting system in which scheduling of data transmission and fault detection mechanisms have been implemented to save energy as well as efforts.

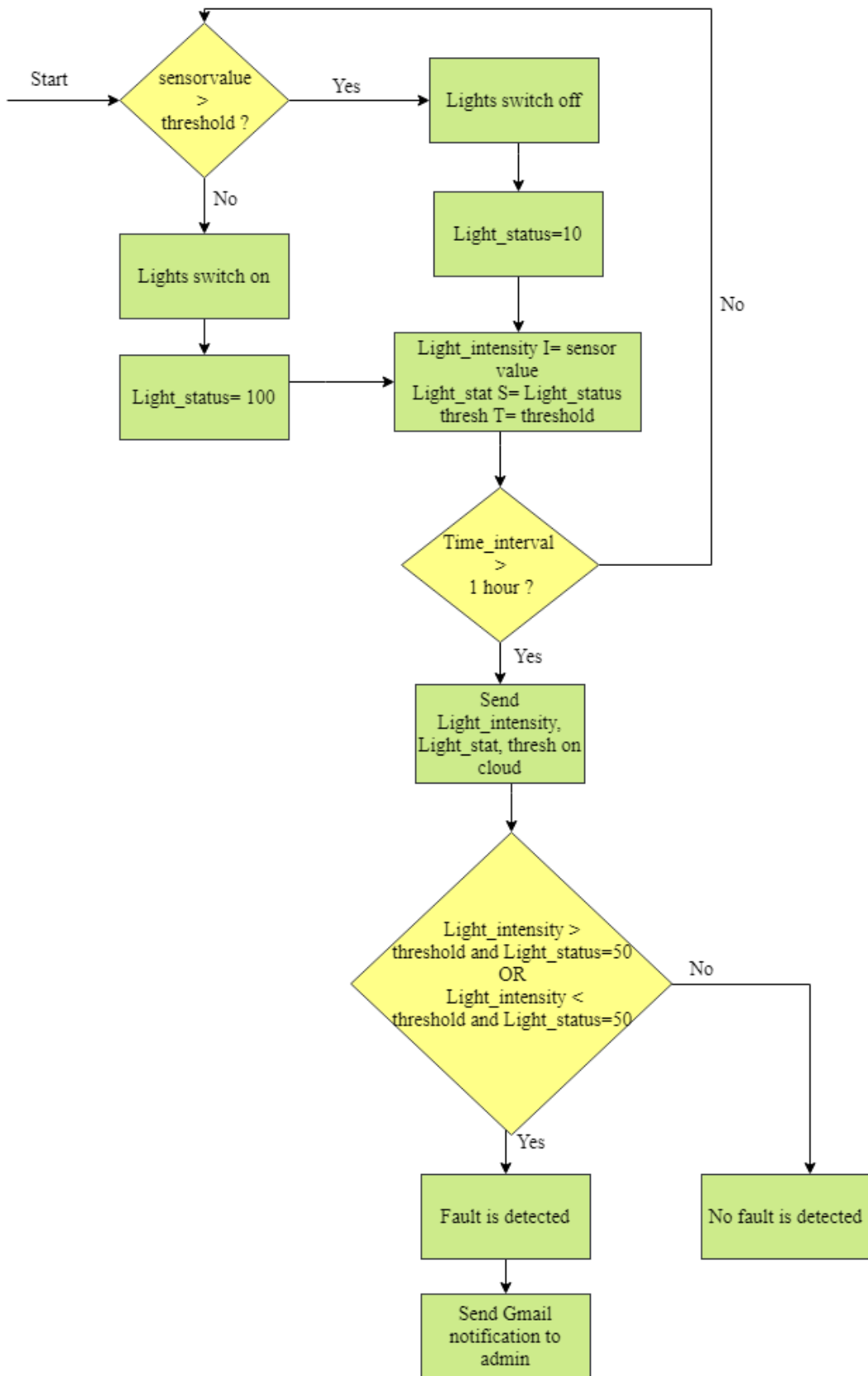


Fig. 5.1. Design of Proposed System

5.4. Conclusion

This chapter has elaborated the features and working of proposed system “Smart street lighting system”. The next chapter discusses the implementation and results of the proposed system.

This chapter discusses about the tools that has been used for implementation of proposed system, implementation in detailed manner as well as the results obtained from it.

6.1. Tools Used

Following are the tools which have been used during the implementation of proposed system:

i. Arduino UNO

Arduino is an embedded system which is used in development of IoT devices. It is highly efficient in its working. It looks like a rectangular board which consists of pins, microcontroller, reset button, USB connector, power jack. It has some computational complexity in the form of uploaded program. When it is connected to internet and performs some task, it becomes part of IoT [77].

Arduino Uno is one of the types of Arduino board. It consists of ATmega328 microcontroller on which written code is uploaded through USB cable. It consists of 14 input/output pins, 6 analog pins, USB connection, power jack. It is easy to connect with computers using USB cable [78].



Fig. 6.1. Arduino Uno [78]

ii. LDR (Light Dependent Resistor)

LDR is a light sensitive device most often used to indicate the presence or absence of light, or to measure light intensity. It also called photoresistor. It works on the principle of photoconductivity. i.e. “electrical conductivity increases on receiving photons results in generation of electrons”. In simpler words, resistance decreases with exposure to light (increase in illumination) [79]. It is used as sensor in systems to sense environment’s light intensity.



Fig. 6.2. LDR [79]

iii. **ESP8266**

ESP8266 is a Wi-Fi hardware module and leading platform for IoT. It enables internet in the embedded system. It has dual functionality: self-contained to host entire application and Wi-Fi adapter to microcontroller. It comes with 32-bit microcontroller and also contains in built TCP/IP stack. The working power range varies from 3.0 to 3.6 volts. It helps in transmission of data generated from sensor to cloud by connecting to Wi-Fi connection. During implementation most basic model of ESP8266 i.e. ESP8266-01 is used.



Fig. 6.3. ESP8266 [80]

iv. **Arduino IDE (1.8.6)**

It is a cross platform application for writing codes and uploading them to Arduino board. Code is uploaded to microcontroller of Arduino by connecting to system where code is written using USB cable. It supports the languages C and C++ using special rules of code structuring. Code related to sensing, scheduling and sending data on cloud is written on Arduino IDE.



Fig. 6.4. Symbol of Arduino IDE [77]

v. LED (Light Emitting Diode)

LED is a semiconductor device. When electrical current passes through it, it radiates light. LED comes in varying sizes. In proposed system four white light LED of size 10 mm are used in order to represent four different locations of street lights. LED has longer life span than sodium vapour lights. Also the power consumption of LED is low in comparison to sodium vapour lights [81] which makes it suitable for saving more energy in proposed system.



Fig. 6.5. LED [82]

vi. ThingSpeak

ThingSpeak is an IoT analytics platform. It aggregates, visualizes and analyses live data streams. It records the data related to lights send from street lighting system and displays information of lights in graphical form. Graphical representation of lights helps admin for monitoring lights status remotely. ThingSpeak has also provided the feature to develop MATLAB codes for analytics purposes based on the application.

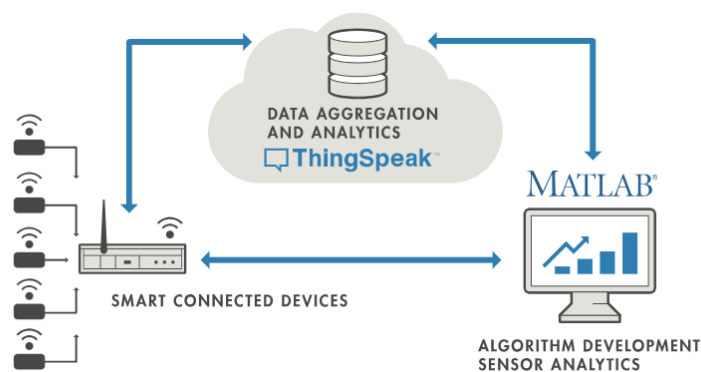


Fig. 6.6. ThingSpeak- Cloud Platform [83]

vii. ThingHTTP

It is an inbuilt feature of ThingSpeak. In ThingHTTP there is no need to implement protocols on the device level. It facilitates communication among IoT devices. Actions

are specified in ThingHTTP, which are triggered using other ThingSpeak apps such as TweetControl, TimeControl, and React.

viii. React

It is used to react to the data in a channel when specific conditions are met and allows to trigger ThingHTTP request. It is linked to ThingHTTP in such a way that when the light status is becoming 50 it is triggering an action in the form of emails indicating fault is detected.

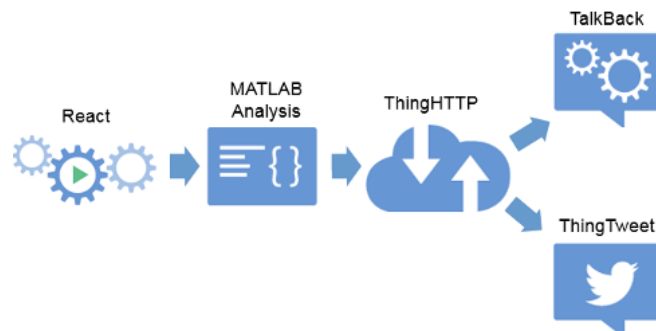


Fig. 6.7. Linking React with ThingHTTP [84]

ix. IFTTT: It is an open source web based service which is used to create applets called webhooks. Conditions are put inside applets. Once an applet is created, a unique API key is generated which is linked to ThingHTTP. An action is triggered in the form of applet by using various web services such as Gmail, Instagram, Pinterest etc. We are using Gmail web service to trigger required action using IFTTT.

x. Multimeter

It is a tool used to measure electrical values like voltage, current and resistance in the circuit. Digital multimeter has more accuracy than analog multimeter hence, commonly used for measuring electrical values [85]. For measuring power consumption in the proposed system Fluke 87V multimeter is used to measure current in both on and off states of lights while transmitting data continuously as well as periodically.



Fig. 6.8. Digital Multimeter [85]

6.2. Experimental Setup

Following are the steps which have been followed to run the whole application:

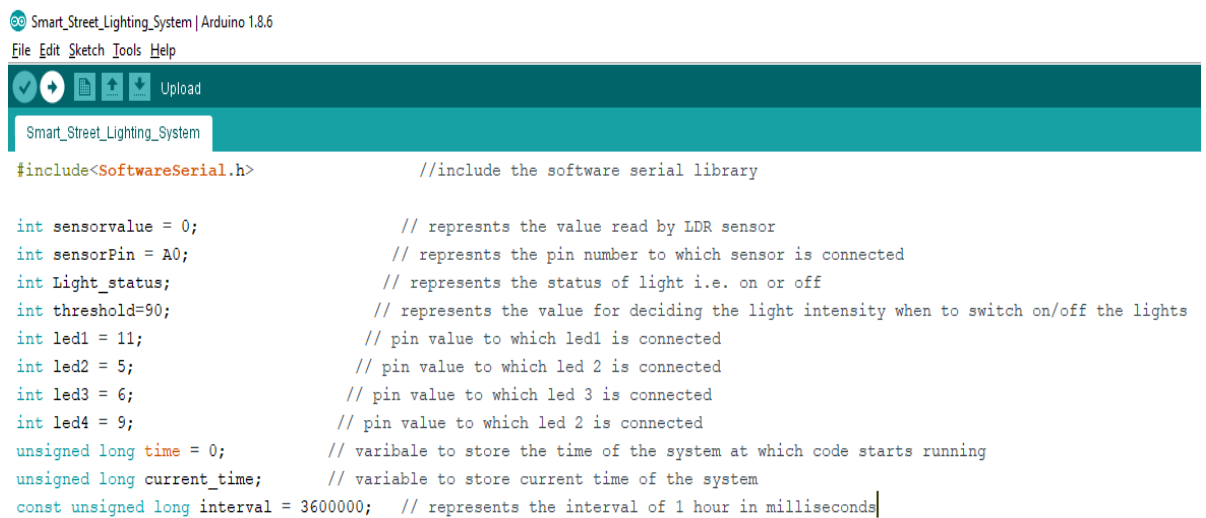
- i. LDR and ESP8266 are connected to Arduino UNO board using Male to Male and Male to Female Connectors.
- ii. Four LEDs are connected with four resistors in parallel.
- iii. User account is made on cloud platform ThingSpeak.
- iv. Channels are created on ThingSpeak and API keys are generated.
- v. User Account is made on IFTTT to make applet.
- vi. Code is written on Arduino IDE which includes:
 - Switching of lights on when sunlight intensity is less than threshold value i.e. 90 and off when sunlight intensity is more than threshold value. Light_status is set as 100 when lights are switched on, 10 when lights are switched off and 50 when there is fault.
 - Setting ESP8266 to connect with Wi-Fi.
 - Calculation to get current time of the system to send data at an interval of 1 hour.
 - Sending of generated data on created channels of cloud.
- vii. Once Wi-Fi is connected, lights status of each LED is sent on ThingSpeak at an interval of 1 hour.
- viii. Link applet with ThingHTTP and React on ThingSpeak which is used for fault detection part.

- ix. To calculate energy consumption, first power consumption of both on and off states have been calculated, then energy is calculated.

6.3. Implementation of Proposed System

This subsection presents the software implementation steps:

- i. At initial step, Arduino IDE was downloaded from <https://www.arduino.cc/> on which related code was written to compile, run and upload on embedded system i.e. Arduino. After downloading Arduino IDE, the very first step of code was to include the library as well as initialize the variables as shown in Fig. 6.9.



```
Smart_Street_Lighting_System | Arduino 1.8.6
File Edit Sketch Tools Help
Upload
Smart_Street_Lighting_System
#include<SoftwareSerial.h> //include the software serial library

int sensorvalue = 0; // represents the value read by LDR sensor
int sensorPin = A0; // represents the pin number to which sensor is connected
int Light_status; // represents the status of light i.e. on or off
int threshold=90; // represents the value for deciding the light intensity when to switch on/off the lights
int led1 = 11; // pin value to which led1 is connected
int led2 = 5; // pin value to which led 2 is connected
int led3 = 6; // pin value to which led 3 is connected
int led4 = 9; // pin value to which led 2 is connected
unsigned long time = 0; // varibale to store the time of the system at which code starts running
unsigned long current_time; // variable to store current time of the system
const unsigned long interval = 3600000; // represents the interval of 1 hour in milliseconds
```

Fig. 6.9. Initialization of Variables

For scheduling of data transmission at an interval of 1 hour “interval” variable is defined and initialized by 3600000 milliseconds which is equivalent to 1 hour. To schedule the transmission, it is important to know the current time of the system which is defined by variable “current_time”. This variable schedule the transmission of data by reading the current time of system.

- ii. To establish internet connection Wi-Fi module i.e. ESP8266 was connected with available network. AT commands CWMOD and CWJAP were used to set Wi-Fi mode as station mode and fix the Wi-Fi connection with the given SSID and password.

```

SoftwareSerial esp8266(3,4); //set the software serial pins RX=4,TX=3

#define SSID "Utkarsha"           // Wifi network name
#define PASS "internetofthings24" // Wifi password

String sendAT(String command,const int timeout)
{
  String response="";
  esp8266.print(command);
  long int time = millis();
  while((time + timeout)>millis())
  {
    while(esp8266.available())
    {
      char c = esp8266.read();
      response += c;
    }
  }
  Serial.print(response);
  return response;
}

void connectwifi()
{
  sendAT("AT\r\n",1000);
  sendAT("AT+CWMOD=1\r\n",1000); //calls send AT function to set esp8266 to station mode
  sendAT("AT+CWJAP=\"SSID\", \"PASS\"\r\n",2000); //AT command to connect with wifi network
  while(!esp8266.find("OK"))
  {
    //wait for connection
  }
  sendAT("AT+CIFSR\r\n",1000); //ATcommand to print IP address on serial monitor
  sendAT("AT+CIPUMX=0\r\n",1000); //AT command to set esp8266 to single connection
}

```

Fig. 6.10. Setting ESP8266 and Wi-Fi Connection

- iii. To establish Wi-Fi connection once as well as to get the output of each LED setup() function of Arduino IDE was used as shown in Fig 6.11.

```

void setup()
{
  Serial.begin(9600); // begin the serial communication with baud rate 9600
  esp8266.begin(9600); //begin the software serial communication with baud rate 9600
  sendAT("AT+RST\r\n",2000); //call sendAT function to send reset AT command
  connectwifi();
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
}

```

Fig. 6.11. Sending Instruction to Establish Wi-Fi and LEDs Output

iv. The main code which enabled the switching of lights on/off on the basis environment's light intensity and responsible for sending data on cloud was written under loop() function. This loop function is responsible to run code repeatedly. Hence, the main code was written under this function.

If the environment's light intensity which was read by sensor is less than threshold value i.e. 90, then there is night otherwise day. The Light_status = 100 represents that light is on and 10 represents that light is off. Similarly, the outputs HIGH and LOW at different LEDs informs the embedded system to switch on and off lights respectively.

```
void loop()
{
  // put your main code here, to run repeatedly:
  sensorvalue = analogRead(sensorPin);

  Serial.println(sensorvalue);
  if (sensorvalue < threshold)
  {
    Serial.println("LED light on");
    digitalWrite(led1,HIGH);
    Light_status=100;
    digitalWrite(led2,HIGH);
    Light_status=100;
    digitalWrite(led3,HIGH);
    Light_status=100;
    digitalWrite(led4,HIGH);
    Light_status=100;
  }
}
```

Fig. 6.12. Switching Lights On

```

else
{
    digitalWrite(led1,LOW);
    Light_status=10;
    digitalWrite(led2,LOW);
    Light_status=10;
    digitalWrite(led3,LOW);
    Light_status=10;
    digitalWrite(led4,LOW);
    Light_status=10;
}

```

Fig. 6.13. Switching Lights Off

- v. Lights related integer values- sensorvalue, Light_status and threshold were converted into string data type to send them on cloud.

```

String Light_intensity = String(sensorvalue); //convert integer to string data type
Serial.print("Light intensity LED:");
Serial.println(sensorvalue); //print light intensity on serial monitor

String Light_stat = String(Light_status); //convert integer to string data type
Serial.print("Light status LED:");
Serial.println(Light_status); //print light status on serial monitor

String thresh = String(threshold); //convert integer to string data type
Serial.print("Threshold LED:");
Serial.println(threshold); //print threshold on serial monitor

```

Fig. 6.14. Converting Integer Data Type into String Data Type

- vi. millis() function was used to get the current time of the system which has helped in scheduling the data transfer at an interval of 1 hour. updateTS function was called at an interval of 1 hour to send the data on cloud. Firstly, variable “current_time” stores the system’s current time by using millis() function. It keeps on checking for the completion of 1 hour. As soon as 1 hour is passed with the last calculated current time of system data is sent on cloud by calling updateTS function. After sending data on cloud variable “time” is updated with the value of “current_time” so that in next iteration the time at which data is sent is added with an interval of 1 hour in order to send data at an interval of 1 hour.

```

current_time = millis();
Serial.print("Current_Time is:");
Serial.println(current_time);
if (current_time > time + interval)
{
    updateTS(Light_intensity,Light_status1,threshold1); //call the function to update ThingSpeak channel
    time = current_time;
}
delay(1000);
}

```

Fig. 6.15. Scheduling of Data Transmission

- vii. Under updateTS function link of cloud platform i.e. ThingSpeak as well as the API keys generated during creating channels for each led on ThingSpeak was passed. The generated API key distinguished the LEDs located at different locations.

```

void updateTS(String I,String S,String T)
{
    Serial.println("");

    //channel 1(LED1)

    sendAT("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", 80\r\n", 1000);
    delay(2000);
    String cmdlen1, cmdlen2, cmdlen3, cmdlen4;
    String cmd1 = "GET /update?key=0YFU16JTGMGQ2E71&field1="+I+"&field2="+S+"&field3="+T+"\r\n";
    cmdlen1= cmd1.length();
    sendAT("AT+CIPSEND="+cmdlen1+"\r\n", 1000);
    esp8266.print(cmd1);
    Serial.print("");
    sendAT("AT+CIPCLOSE=\r\n", 2000);

    //channel 2(LED2)

    sendAT("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", 80\r\n", 1000);
    delay(2000);
    String cmd2 = "GET /update?key=DTI4LYOWF2UGFEBJ&field1="+I+"&field2="+S+"&field3="+T+"\r\n";
    cmdlen2= cmd2.length();
    sendAT("AT+CIPSEND="+cmdlen2+"\r\n", 1000);
    esp8266.print(cmd2);
    Serial.print("");
    sendAT("AT+CIPCLOSE=\r\n", 2000);

    //channel 3(LED3)

    sendAT("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", 80\r\n", 1000);
    delay(2000);
    String cmd3 = "GET /update?key=XDOLHTQRHIFCRLDY&field1="+I+"&field2="+S+"&field3="+T+"\r\n";
    cmdlen3= cmd3.length();
    sendAT("AT+CIPSEND="+cmdlen3+"\r\n", 2000);
    esp8266.print(cmd3);
    Serial.print("");
    sendAT("AT+CIPCLOSE=\r\n", 2000);
}

```

```

//channel 4 (LED4)

sendAT("AT+CIPSTART=\\"TCP\\",\\"api.thingspeak.com\\",80\r\n",1000);
delay(2000);
String cmd4 = "GET /update?key=230D3RG0K59VY8BP&field1="+I+"&field2="+S+"&field3="+T+"\r\n";
cmdlen4= cmd4.length();
sendAT("AT+CIPSEND="+cmdlen4+"\r\n",2000);
esp8266.print(cmd4);
Serial.print("");
sendAT("AT+CIPCLOSE=\r\n",2000);

Serial.println("");
delay(1000);
}

```

Fig. 6.16. Sending Data on Cloud

- viii. Four channels were created in order to monitor lights remotely which were located at different locations. Four locations of college were taken for reference and named accordingly. On clicking channel information related to particular light gets displayed.

Name	Created	Updated
COS_LED1 Private Public Settings Sharing API Keys Data Import / Export	2018-11-01	2018-11-23 16:16
Hostel K_LED2 Private Public Settings Sharing API Keys Data Import / Export	2018-11-14	2018-11-23 16:17
TAN_LED3 Private Public Settings Sharing API Keys Data Import / Export	2018-11-14	2018-11-14 07:34
Thapar R&D center_LED4 Private Public Settings Sharing API Keys Data Import / Export	2018-11-14	2018-11-14 07:37

Fig. 6.17. Channel Creation on Cloud Platform

- ix. To send Gmail notifications to admin on detecting fault in LEDs, ThingHTTP and React features of ThingSpeak are used. Firstly, applet was created on IFTTT to trigger the event and then it was linked with ThingHTTP and React so that if any event occurs i.e. fault then Gmail notification is sent to admin.



Your key is: **guvtkTJAMkXcrhZHNzhqWsr_rug9RcT4R-ueuOpMJ8R**

[◀ Back to service](#)

To trigger an Event

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/{event}/with/key/guvtkTJAMkXcrhZHNzhqWsr_rug9RcT4R-ueuOpMJ8R
```

With an optional JSON body of:

```
{ "value1" : " ", "value2" : " ", "value3" : " " }
```

The data is completely optional, and you can also pass `value1`, `value2`, and `value3` as query parameters or form variables. This content will be passed on to the Action in your Recipe.

You can also try it with `curl` from a command line.

```
curl -X POST https://maker.ifttt.com/trigger/{event}/with/key/guvtkTJAMkXcrhZHNzhqWsr_rug9RcT4R-ueuOpMJ8R
```

Fig. 6.18. API Creation on IFTTT

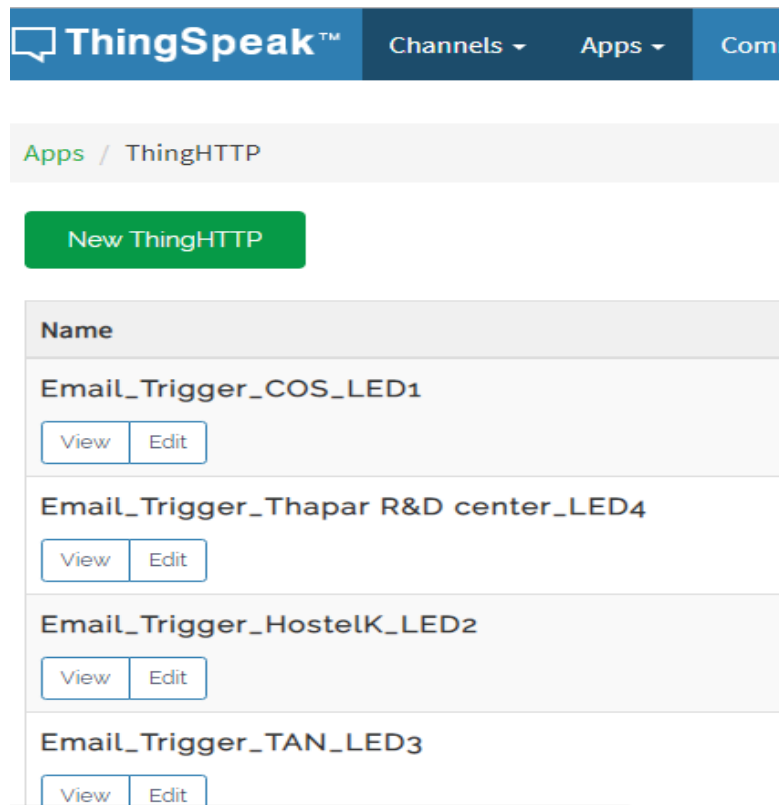


Fig. 6.19. ThingHTTP Creation

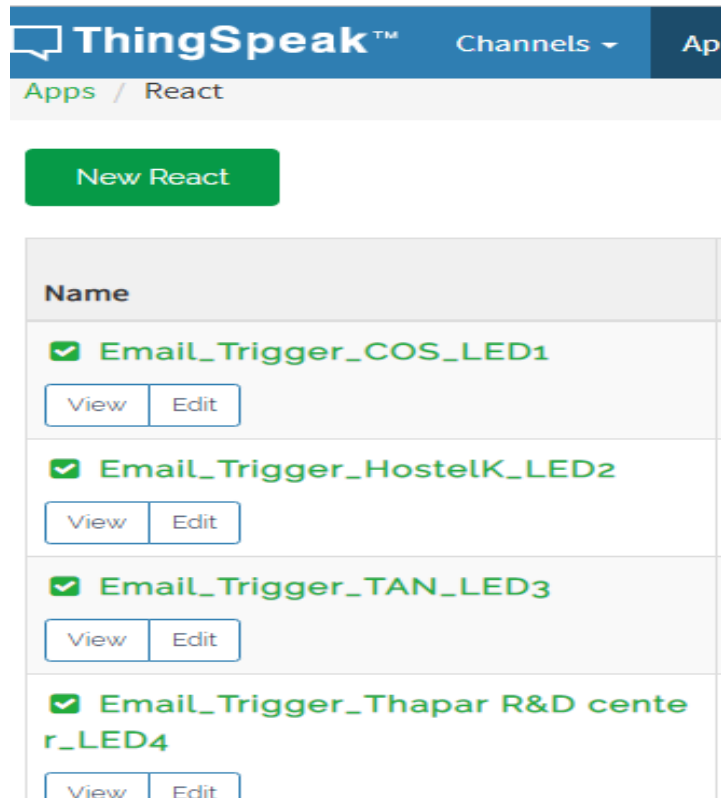


Fig. 6.20. React Creation

- x. Mutation testing was done to check the system reaction on detecting fault in the system. Fault was induced in the code in such a way that lights do not light on during night time and off during day time. In this case, Light_status value becomes 50 indicating fault in the system.

```

if (sensorvalue < threshold)
{
  Serial.println("LED light on");
  digitalWrite(led1, LOW);
  Light_status=50;
  //digitalWrite(led2, HIGH);
  //Light_status=100;
  digitalWrite(led3, LOW);
  Light_status=50;
  //digitalWrite(led4, HIGH);
  //Light_status=100;
}

else
{
  digitalWrite(led1, HIGH);
  Light_status=50;
  //digitalWrite(led2, LOW);
  //Light_status=10;
  digitalWrite(led3, HIGH);
  Light_status=50;
  //digitalWrite(led4, LOW);
  //Light_status=10;
}

```

Fig. 6.21. Induction of Fault

6.4. Result

The proposed work provides a solution to the problem of high energy consumption in the existing smart street lighting systems. In order to save energy lights were automated on the basis of surrounding's light intensity. Fig. 6.22 represents the on state of street lights when surrounding light intensity is less than 90 i.e. indicating night time. Similarly, Fig. 6.23 represents the off state of street lights when surrounding light intensity is greater than 90 i.e. indicating day time.

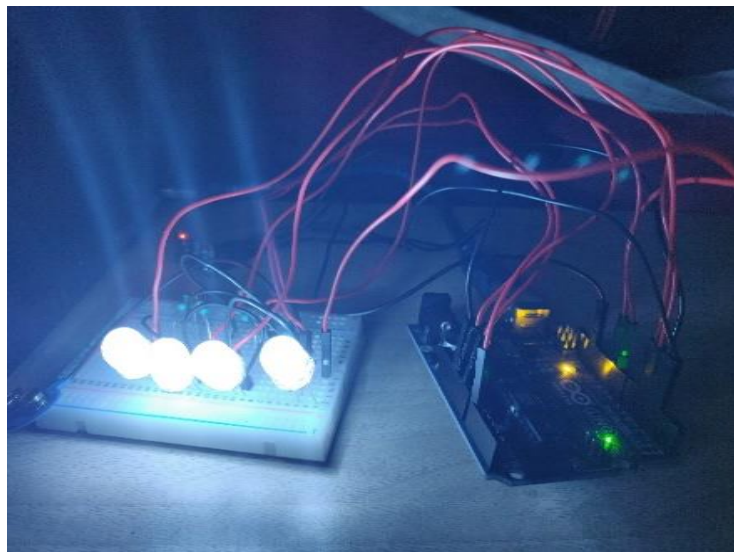


Fig. 6.22. On State of Lights During Night

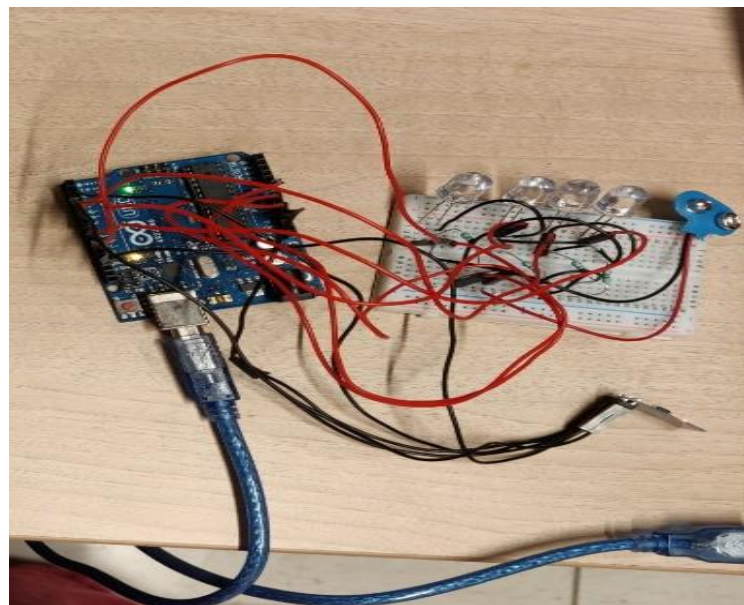


Fig. 6.23. Off State of Lights During Day

Also Gmail notification is sent to admin when fault is detected. Light_status = 50 indicates that there is fault in the light. In notification location of light is also send to admin so that quick action can be taken without wasting any time.

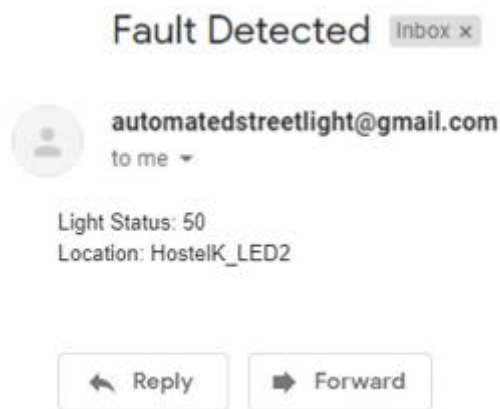


Fig. 6.24. Gmail Notification for Fault Detection

Two cases are created to show the difference between the energy consumption of whole system when data is transmitted continuously through Wi-Fi to cloud and when transmission is scheduled at an interval of 1 hour. Following are the two cases:

Case I: Wi-Fi (ESP8266) sending data continuously on cloud

Fig. 6.25 illustrates the presentation of data send continuously on cloud which is visible to admin for monitoring purpose. Light_intensity with values less than 90 indicates that there is night and accordingly Light_stat with values 100 indicates that lights are switched on during night time. Hike in upward direction of Light_intensity indicates that environment's light intensity is increased i.e. day time and at the same time falling of values from 100 to 10 in Light_stat indicates that light gets switched off during day time. Threshold value 90 is constant throughout the time.

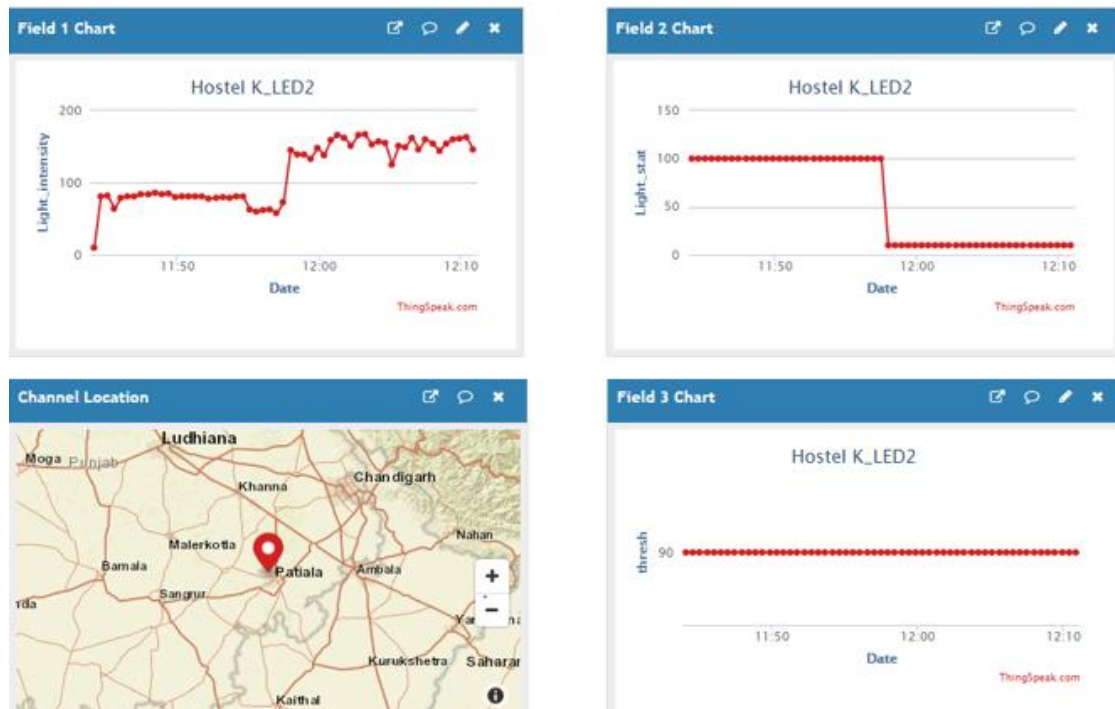


Fig. 6.25. Pattern of Continuously Sent Data on Cloud

For powering lights voltage of 8.31 V is supplied through DC source which remain constant throughout the working of system. Current is measured using Multimeter which varies on the basis of lights either on or off and the state of ESP8266 either active or ideal.

a) Lights are off i.e. during day time

Voltage, $V = 8.31\text{V}$
 Current, $I = 400\text{ mA}$

$$\text{Power}_{Ia} = V * I = (8.31 * 400) / 1000 = 3.324\text{ Watt}$$

b) Lights are on i.e. during night time

Voltage, $V = 8.31\text{V}$
 Current, $I = 480\text{ mA}$

$$\text{Power}_{Ib} = V * I = (8.31 * 480) / 1000 = 3.9888\text{ Watt}$$

Case II: Wi-Fi (ESP8266) sending data periodically i.e. on an interval of 1 hour

Fig. 6.26 illustrates the presentation of scheduled data transmission on cloud which is visible to admin for monitoring purpose.

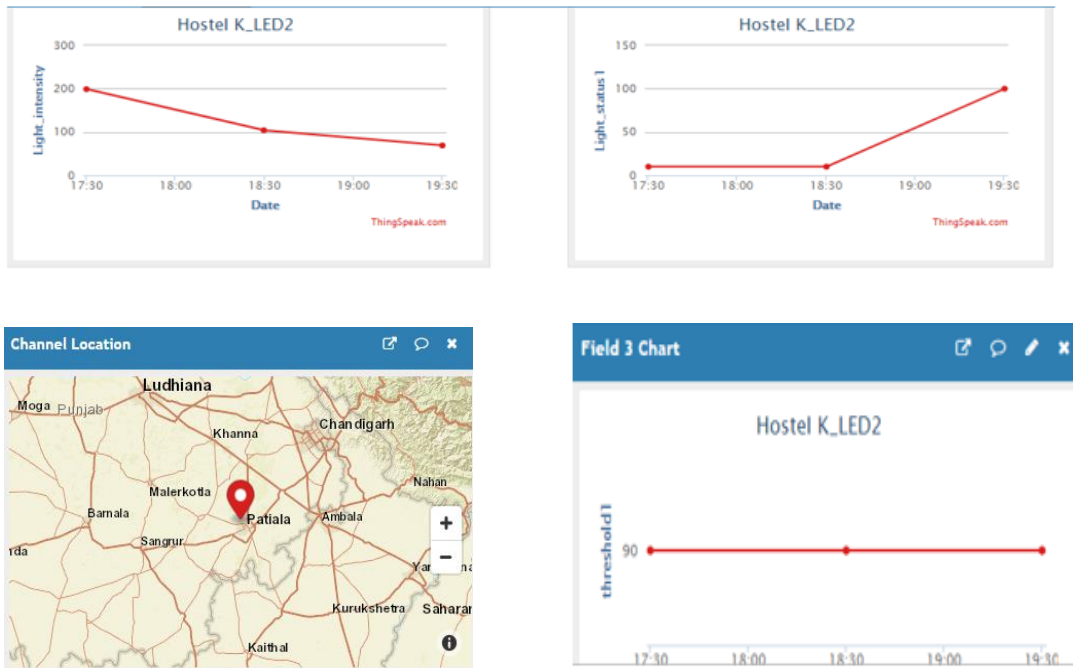


Fig. 6.26. Pattern of Scheduled Transmitted Data on Cloud

Like Case I values of Light_intensity, Light_stat, thresh are showing accordingly. The only difference is that the time at which data is sent on cloud. There is a difference of 1 hour between the transmission of data on cloud.

a) Lights are off i.e. during day time

Voltage, $V = 8.31V$

Current, $I = 325 \text{ mA}$

$$\text{Power}_{IIc} = V * I = (8.31 * 325) / 1000 = 2.70075 \text{ Watt}$$

b) Lights are on i.e. during night time

Voltage, $V = 8.31V$

Current, $I = 405 \text{ mA}$

$$\text{Power}_{II d} = V * I = (8.31 * 405) / 1000 = 3.36555 \text{ Watt}$$

Calculation of Energy Consumption

To calculate the energy consumption of lights in real time standard timing of switching on and off street lights are considered.

Duration of lights off = 6 A.M. to 7 P.M. = 13 hours

Duration of lights on = 7 P.M. to 6 A.M. = 11 hours

Case I: Wi-Fi (ESP8266) sending data continuously on cloud

a) Power Consumption per month I_a

$$\begin{aligned} & \text{Power } I_a * \text{Duration of lights off} * \text{Number of days in a month} \\ &= 3.324 * 13 * 30 \\ &= 1296.36 \text{ Wh} \\ &= 1.29636 \text{ KWh} \end{aligned}$$

Energy Consumption I_a (E_{I_a})

$$\begin{aligned} &= \text{Power Consumption per month } I_a * 3600000 \text{ J} \\ &= 1.29636 * 3600000 \\ &= 4666896 \text{ J} \\ &= 4666.896 \text{ kJ} \end{aligned}$$

b) Power Consumption per month I_b

$$\begin{aligned} & \text{Power } I_b * \text{Duration of lights on} * \text{Number of days in a month} \\ &= 3.9888 * 11 * 30 \\ &= 1316.304 \text{ Watt-hour} \\ &= 1.316304 \text{ KWh} \end{aligned}$$

Energy Consumption I_b (E_{I_b})

$$\begin{aligned} &= \text{Power Consumption per month } I_b * 3600000 \text{ J} \\ &= 1.316304 * 3600000 \\ &= 4738694.4 \text{ J} \\ &= 4738.6944 \text{ kJ} \end{aligned}$$

Case II: Wi-Fi (ESP8266) sending data periodically i.e. on an interval of 1 hour

c) Power Consumption per month P_{IIc}

$$\begin{aligned} & \text{Power}_{IIc} * \text{Duration of lights off} * \text{Number of days in a month} \\ &= 2.70075 * 13 * 30 \\ &= 1053.2925 \text{ Watt-hour} \\ &= 1.0532925 \text{ KWh} \end{aligned}$$

Energy Consumption P_{IIc} (E_{IIc})

$$\begin{aligned} &= \text{Power Consumption per month}_{IIc} * 3600000 \text{ J} \\ &= 1.0532925 * 3600000 \\ &= 3791853 \text{ J} \\ &= 3791.853 \text{ kJ} \end{aligned}$$

d) Power Consumption per month $P_{II d}$

$$\begin{aligned} & \text{Power}_{II d} * \text{Duration of lights on} * \text{Number of days in a month} \\ &= 3.36555 * 11 * 30 \\ &= 1110.6315 \text{ Watt-hour} \\ &= 1.1106315 \text{ KWh} \end{aligned}$$

Energy Consumption $P_{II d}$ ($E_{II d}$)

$$\begin{aligned} &= \text{Power Consumption per month}_{II d} * 3600000 \text{ J} \\ &= 1.1106315 * 3600000 \\ &= 3998273.4 \text{ J} \\ &= 3998.2734 \text{ kJ} \end{aligned}$$

6.5. Comparative Analysis

On the basis of calculation of energy consumption, it has been observed that the proposed solution scheduling of data transmission consumes less energy in both on and off states of lights in comparison to continuous sending of data in both on and off states of lights. Table 6.1 summarizes the result as well as compares energy consumption of both cases.

Table 6.1. Comparison of Energy Consumption in Proposed System

Case I: Sending data continuously on cloud	ESP8266 Mode	Light Status	Energy Consumption	Analysis
Case I: Sending data continuously on cloud	Active	ON	$E_{Ia} = 4666.896 \text{ kJ}$	$E_{IIc} < E_{Ia}$ $E_{IIId} < E_{Ib}$
	Active	OFF	$E_{Ib} = 4738.6944 \text{ kJ}$	
Case II: Scheduled data transmission	Ideal	ON	$E_{IIc} = 3791.853 \text{ kJ}$	
	Ideal	OFF	$E_{IIId} = 3998.2734 \text{ kJ}$	

Fig. 6.27 illustrates the energy consumption of both on and off states of lights of the proposed system in both the above mentioned cases. Y- axis represents the energy consumption in kJ (kilo joules).

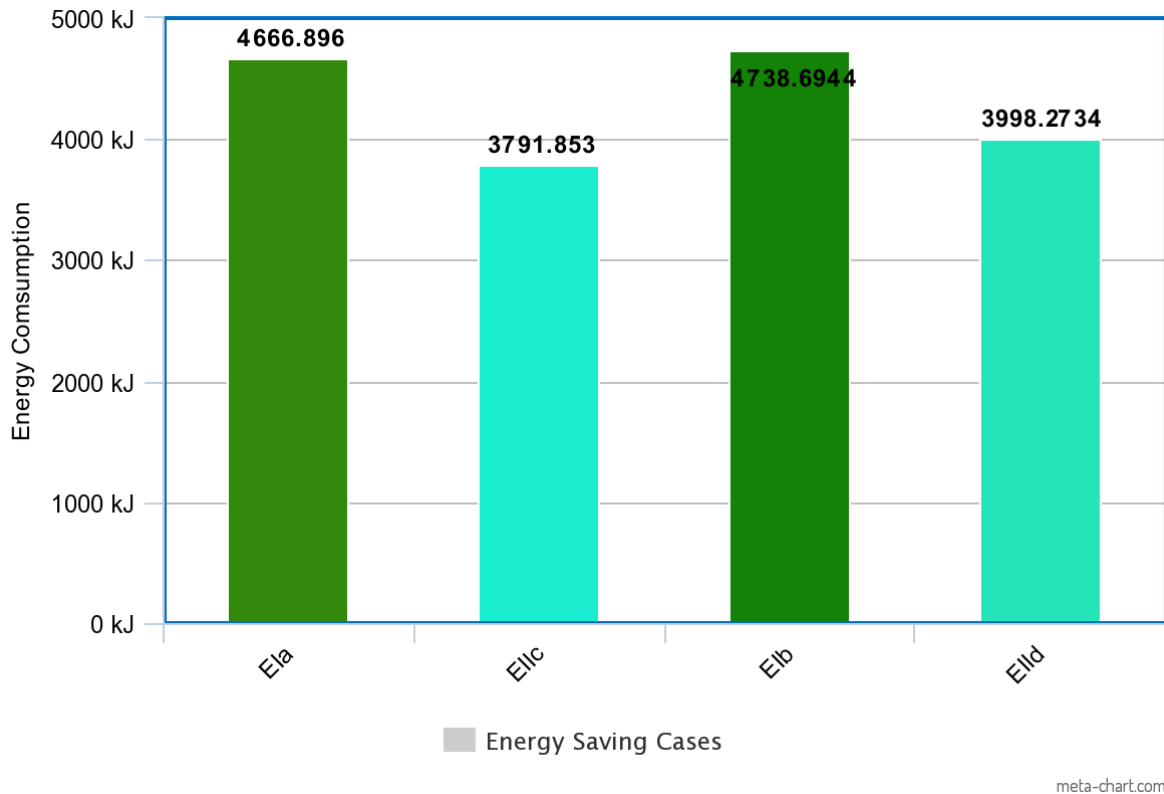


Fig. 6.27. Comparative Analysis of Energy Consumption in Proposed System

X - axis represents the energy consumption of implemented IoT based application in different cases as discussed above. It represents that the energy consumption of lights in off state when data is send continuously on cloud (E_{Ia}) is more than when data is send at scheduled interval of time (E_{IIc}). Similarly, energy consumption of lights in on state when data is send continuously on cloud (E_{Ib}) is more than when data is send at scheduled interval of time (E_{IIId}).

Energy Efficiency off state

$$\begin{aligned} &= (\text{Energy Saving during off state} / E_{1a}) * 100 \\ &= (875.043/4666.896) * 100 \\ &= 18.75 \% \end{aligned}$$

Energy efficiency of lights in off state is increased by 18.75% after implementation of proposed solution.

Energy Efficiency on state

$$\begin{aligned} &= (\text{Energy Saving during on state} / E_{1b}) * 100 \\ &= (740.421/4738.6944) * 100 \\ &= 15.625 \% \end{aligned}$$

Similarly, energy efficiency of lights in on state is increased by 15.625 % after implementation of proposed solution.

6.6. Conclusion

This chapter elaborates the implementation details of the proposed solution in the proposed system. In the results, it has been observed that when Wi-Fi module is in active state i.e. sending data through it, it consumes more power and when it is sending data continuously, then power consumption becomes high throughout the working of the system. And when Wi-Fi module is in ideal state i.e. not sending data through it, it consumes less power. Scheduling of data transmission in smart street lighting system saves energy by reducing energy consumption during transmission of data.

7.1. Conclusion

The main goal of this research work is to propose an energy efficient solution for IoT based application. The proposed solution satisfies the research gap found in the literature survey of energy efficient solutions for IoT as well as in IoT based applications. The proposed solution handles the data transmission of data generated from IoT device to cloud such that transmission is scheduled at an interval of 1 hour which avoids continuous data transfer and results in low energy consumption. The projected work assures the energy efficiency of proposed IoT system by verifying that periodic data transfer consumes less energy than continuous data transfer.

7.2. Thesis Contribution

This thesis has contributed in following ways:

- This work provides an efficient solution for saving energy in IoT environment for smart street lighting systems by considering energy consumption while transmitting data from IoT device to cloud.
- It presents a solution to reduce energy consumption of IoT based system while transmission of data.
- A case study application, named Smart street lighting system has been proposed and designed that is used for implementing the proposed scheduling solution and validating the results.
- This scheduled data transfer makes the Wi-Fi module ideal for the duration when no data is sent due to which energy consumption is low during that period, thus reduces energy consumption of whole system.
- Fault is induced in the proposed system by using the concept of mutation testing so that Gmail notification is sent to admin on detection of fault in order to reduce continuous monitoring.

7.3. Future Scope

The proposed work can be enhanced using the following future work:

- In future the proposed work can also be enhanced with more energy efficient solutions like scheduling of sensors, deployment of fog servers, combination of other sensors etc.
- Prototype of proposed system can be installed in real time application area with more street lights.
- The proposed solution can be used in other IoT based applications to make them more energy efficient.
- In future especially designed IoT gateway can be used to run application on larger scale.

List of Publications

[1] Utkarsha Singh and Inderveer Chana, “Energy Efficient Solutions for IoT Based Applications”, *2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT 2019)*.

[Accepted]

[2] Utkarsha Singh and Inderveer Chana, “Enhancing Energy Efficiency in IoT (Internet of Thing) Based Application”, in *Springer 4th International Conference on Inventive Computing (ICICT 2019)*.

[Accepted]

[3] Utkarsha Singh and Inderveer Chana, “Comparative Analysis of Energy Efficient Techniques in IoT”, *IEEE Global Conference for Advancement in Technology (GCAT 2019)*.

[Communicated]

References

- [1] M. Rouse, "What is machine-machine-to-machine(M2M)? - Definition from WhatIs.com," TechTarget, [Online]. Available: <https://internetofthingsagenda.techtarget.com/definition/machine-to-machine-M2M>. [Accessed 02 10 2018].
- [2] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton and J. Henry, IoT Fundamentals:Networking Technologies,Protocols,and Use Cases for the Internet of Things, Indianapolis: Cisco Press, 2017.
- [3] "Defining and securing the Internet of Things: ENISA publishes a study on how to face cyber threats in critical information infrastructures - ENISA," European Union Agency for Network and Information Security, 20 November 2017. [Online]. Available: <https://www.enisa.europa.eu/news/enisa-news/defining-and-securing-the-internet-of-things>. [Accessed 02 May 2019].
- [4] JavaTpoint, "IoT Solution Architecture Models - JavaTpoint," JavaTpoint, [Online]. Available: <https://www.javatpoint.com/iot-architecture-models>. [Accessed 4 January 2019].
- [5] "Wifi enabled Syska SmartLight compatible with Aazon Alexa," Syska, 2018. [Online]. Available: <https://syska.co.in/smartlight-led-bulb.html>. [Accessed 01 May 2019].
- [6] "Mi Band 3 | All New-Large OLED touchscreen - Mi India," Xiaomi, [Online]. Available: <https://www.mi.com/in/mi-band-3/>. [Accessed 20 February 2018].
- [7] A. Jalan, "A Survey on Automated Street Lighting System on Indian Streets Using Arduino," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 6, no. 3, pp. 4139-4144, 2017.
- [8] D. TEAM, "IoT Applications | Top 10 Uses of Internet of Things - Data Flair," Data Flair, 15 September 2018. [Online]. Available: <https://data-flair.training/blogs/iot-applications/>. [Accessed 27 November 2018].
- [9] L. Experience, "Wireless Lighting Control - LumiFi Experience," Shopify, [Online]. Available: <https://lumifi.com/>. [Accessed 14 January 2019].
- [10] J. Page, "A Town in Norway Installs Auto Dimming Street Lights to Cut Down on Energy Costs," Laughing Squid, 5 1 2018. [Online]. Available: <https://laughingsquid.com/auto-dimming-street-lights>. [Accessed 23 September 2018].

- [11] D. Peeples, “Smart Cities Council | Smart street lighting: Where do we go from here?,” 02 February 2017. [Online]. Available: <https://na.smartcitiescouncil.com/article/smart-street-lighting-where-do-we-go-here>. [Accessed 27 October 2018].
- [12] G. Electric, “How to Gain a Competitive Advantage in a Tough Economy,” Noria Corporation, [Online]. Available: <https://www.reliableplant.com/Read/27393/Nations-smart-grid-investment>. [Accessed 07 May 2019].
- [13] “Energy Consumption from Internet of Things and Wireless Technology,” 5G ? IoT, [Online]. Available: <https://whatis5g.info/energy-consumption/>. [Accessed 18 December 2018].
- [14] CISCO, “Internet of Things At a Glance,” CISCO, June 2016. [Online]. Available: <https://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf>. [Accessed 7 December 2018].
- [15] N. Shahid and S. Aneja, “Internet of Things: Vision, application areas and research challenges,” in *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2017.
- [16] J. Guth, U. Breitenbücher, M. Falkenthal, F. Leymann and L. Reinfurt, “Comparison of IoT platform architectures: A field study based on a reference architecture,” in *2016 Cloudification of the Internet of Things (CIoT)*, Paris, France, 2016.
- [17] K. Ismail, “Edge Computing vs Fog Computing: What's the Difference?,” Simpler Media Group, Inc. , 14 August 2018. [Online]. Available: <https://www.cmswire.com/information-management/edge-computing-vs-fog-computing-whats-the-difference/>. [Accessed 14 February 2019].
- [18] R. H. Weber, “Internet of Things - New security and privacy challenges,” *Computer Law & Security Review*, vol. 26, no. 1, pp. 23 - 30, 2010.
- [19] C. Occhiuzzi, S. Caizzone and G. Marrocco, “Passive UHF RFID antennas for sensing applications: Principles, methods, and classifications,” *IEEE Antennas and Propagation Magazine*, vol. 55, no. 6, pp. 14 - 34, 2013.
- [20] Y. Lim, S. Daas, S. J. Hashim, R. M. Sidek, N. A. Kamsani and F. Z. Rokhani, “Reduced hardware architecture for energy-efficient IoT healthcare sensor nodes,” in *IEEE International Circuits and Systems Symposium (ICSyS)*, Langkawi, Malaysia, 2015.
- [21] M.-H. Dao, V.-P. Hoang, V.-L. Dao and X.-T. Tran, “An Energy Efficient AES Encryption Core for Hardware Security Implementation in IoT Systems,” in *International Conference on Advanced Technologies for Communications (ATC)*, Ho Chi Minh City, Vietnam, 2018.
- [22] S. F. Abedin, M. G. R. Alam, R. Haw and C. S. Hong, “A System Model for Energy Efficient Green-IoT,” *International Conference on Information Networking (ICOIN)*, pp. 177 - 182, 2015.

- [23] M. Amarlingam, P. K. Mishra, K. V. V. D. Prasad and P. Rajalakshmi, "Compressed sensing for different sensors: A real scenario for WSN and IoT," *IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 289 - 294, 2016.
- [24] S. D. Dwivedi and P. Kaushik, "Energy Efficient Routing Algorithm with sleep scheduling in Wireless Sensor Network," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 3, no. 3, pp. 4350 - 4353, 2012.
- [25] F. Ye, A. Chen, S. Lu and L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks," *Proceedings Tenth International Conference on Computer Communications and Networks*, pp. 304 - 309, 2001.
- [26] P. Bai, K. kumar and S. kumar, "Energy Efficient Communication Protocol at Network Layer for Internet of Things," *International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 148 - 153, 2018.
- [27] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid and H. Yu, "Green IoT: An Investigation on Energy Saving Saving Practices for 2020 and Beyond," *IEEE Access*, vol. 5, pp. 15667 - 15681, 2017.
- [28] C. Estevez and J. Wu, "Recent advances in Green Internet of Things," *7th IEEE Latin-American Conference on Communications (LATINCOM)*, pp. 1 - 5, 2015.
- [29] F. Jalali, S. Khodadustan, C. Gray, K. Hinton and F. Suits, "Greening IoT with Fog: A Survey," *IEEE International Conference on Edge Computing (EDGE)*, pp. 25 - 31, 2017.
- [30] F. Jalali, K. Hinton, R. Ayre, T. Alpcan and R. S. Tucker, "Fog Computing May Help to Save Energy in Cloud Computing," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1728 - 1739, 2016.
- [31] K. Wang, Y. Wang, Y. Sun, S. Guo and J. Wu, "Green Industrial Internet of Things Architecture: An Energy-Efficient Perspective," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 48 - 54, 2016.
- [32] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *Journal of Electrical and Computer Engineering*, 2017.
- [33] S. M. A. Oteafy, F. M. Al-Turjman and H. S. Hassanein, "Pruned Adaptive Routing in the heterogeneous Internet of Things," *IEEE Global Communications Conference (GLOBECOM)*, pp. 214 - 219, 2012.
- [34] J. Č. Gambiroža and T. Masteli, "Big Data Challenges and Trade-offs in Energy Efficient Internet of Things systems," *26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1 - 6, 2018.
- [35] J. Huang, Y. Meng, X. Gong, Y. Liu and Q. Duan, "A Novel Deployment Scheme for Green Internet of Things," *IEEE Internet of Things Journal*, pp. 196 - 205, 2014.

- [36] K. Georgiou, S. Xavier-de-Souza and K. Eder, "The IoT Energy Challenge: A Software Perspective," *IEEE Embedded Systems Letters*, vol. 10, no. 3, pp. 53 - 56, 2018.
- [37] O. B. Akan, O. Cetinkaya, C. Koca and M. Ozger, "Internet of Hybrid Energy Harvesting Things," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 736 - 746, 2018.
- [38] S. K. Routray and S. K. P., "Green Initiatives in IoT," *3rd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB17)*, pp. 454 - 457, 2017.
- [39] L. Farhan, A. E. Alissa, S. T. Shukur, M. Hammoudeh and R. Kharel, "An Energy Efficient Long Hop (LH) First Scheduling Algorithm for Scalable Internet of Things (IoT) Networks," in *Eleventh International Conference on Sensing Technology (ICST)*, Sydney, NSW, Australia, 2017.
- [40] V. M, K. SPR and R. A, "Node Level Energy Efficiency Protocol for Internet of Things," *Journal of Theoretical & Computational Science*, vol. 3, no. 1, 2016.
- [41] F. Jalali, A. Vishwanath, J. d. Hoog and F. Suits, "Interconnecting Fog Computing and Microgrids for Greening IoT," in *IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, Melbourne, VIC, Australia, 2016.
- [42] S. Kitanov and T. Janevski, "Energy efficiency of Fog Computing and Networking services in 5G networks," in *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, Ohrid, Macedonia, 2017.
- [43] P. Scully, "The top 10 IoT application areas - based on real IoT projects," IoT Analytics GmbH, 22 February 2018. [Online]. Available: <https://iot-analytics.com/top-10-iot-segments-2018-real-iot-projects/>. [Accessed 22 April 2019].
- [44] R. Gour, "Top 10 Applications of IoT - DZone IoT," Dzone, 30 October 2018. [Online]. Available: <https://dzone.com/articles/top-10-uses-of-the-internet-of-things>. [Accessed 23 January 2019].
- [45] C. B. Soh*, J. J. Tan, K. J. Tseng, W. L. Woo and J. W. R. Teo, "Intelligent Street Lighting for Smart Cities," *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, pp. 1027 - 1031, 2018.
- [46] P. P. F. Dheena, G. S. Raj, G. Dutt and S. V. Jinny, "IOT Based Smart Street Light Management," in *2017 IEEE International Conference on Circuits and Systems (ICCS)*, Thiruvananthapuram, India, 2017.
- [47] M. Kokilavani and A. Malathi, "Smart Street Lighting System using IoT," *International Journal of Advanced Research in Applied Science and Technology*, vol. 3, no. 11, pp. 8 - 11, 2017.
- [48] S. Bhosale, K. Gaware, P. Phalke, D. Wadekar and P. Ahire, "IOT BASED DYNAMIC CONTROL OF STREET LIGHTS FOR SMART CITY," *International Research Journal of Engineering and Technology (IRJET)*, vol. 04, no. 05, pp. 1181 - 1183, 2017.

- [49] K. Tamilselvan, K.S. Deepika, A. Gobinath, S. Harhini and S. Gokhulraj, "IoT based street light monitoring system," *International Journal of Intellectual Advancements and Research in Engineering Computations*, vol. 6, no. 1, pp. 71 - 73, 2018.
- [50] G. Cacciatore, C. Fiandrino, D. Kliazovich, F. Granelli and P. Bouvry, "Cost Analysis of Smart Lighting Solutions for Smart Cities," in *2017 IEEE International Conference on Communications (ICC)*, Paris, France, 2017.
- [51] S. Chen, G. Xiong, J. Xu, S. Han, F.-Y. Wang and K. Wang, "The Smart Street Lighting System Based on NB-IoT," *2018 Chinese Automation Congress (CAC)*, pp. 1196 - 1200, 2018.
- [52] G. Jia, G. Han, A. Li and J. Du, "SSL: Smart Street Lamp Based on Fog Computing for Smarter Cities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4995 - 5004, 2018.
- [53] M. Muhamad and M. I. M. Ali, "IoT Based Solar Smart LED Street Lighting System," *TENCON 2018 - 2018 IEEE Region 10 Conference*, pp. 1801 - 1806, 2018.
- [54] A. Gupta, P. Rastogi and S. Jain, "Smart Parking System using Cloud based Computation and Raspberry Pi," in *2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, India, 2018.
- [55] R. K. Kodali, K. Y. Borra, S. S. G. N. and H. J. Domma, "An IoT based Smart Parking System using LoRa," in *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, Zhengzhou, China, 2018.
- [56] A. Araújo, R. Kalebe, G. Girão, I. Filho, K. Gonçalves, A. Melo and B. Neto, "IoT-Based Smart Parking for Smart Cities," *IEEE First Summer School on Smart Cities (S3C)*, pp. 31 - 36, 2017.
- [57] B. M. Mahendra, S. Sonoli, N. Bhat, Raju and T. Raghu, "IoT Based Sensor Enabled Smart Car Parking for Advanced Driver Assistance System," in *2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, 2017.
- [58] P. Chippalkatti, G. Kadam and V. Ichake, "I-SPARK: IoT based Smart Parking System," in *International Conference On Advances in Communication and Computing Technology (ICACCT)*, Sangamner, India, 2018.
- [59] M. Fraifer and M. Fernström, "Smart Car Parking System Prototype Utilizing CCTV Nodes: A proof of concept prototype of a novel approach towards IoT-concept based smart parking," *IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 649 - 654, 2016.
- [60] H. Nasir, W. B. W. Aziz, F. Ali, K. Kadir and S. Khan, "The Implementation of IoT Based Smart Refrigerator System," in *2nd International Conference on Smart Sensors and Application (ICSSA)*, Kuching, Malaysia, 2018.

- [61] J. L. Torralbo-Muñoz, S. Sendra, L. Parra and J. Lloret, “SmartFridge: The Intelligent System that Controls your Fridge,” in *Fifth International Conference on Internet of Things: Systems, Management and Security*, Valencia, Spain, 2018.
- [62] H.-H. Wu and Y.-T. Chuang, “Low-Cost Smart Refrigerator,” in *IEEE International Conference on Edge Computing (EDGE)*, Honolulu, HI, USA, 2017.
- [63] A.-D. Floarea and V. Sgârciu, “Smart refrigerator: A next generation refrigerator connected to the IoT,” in *8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Ploiesti, Romania, 2016.
- [64] A. TEAM, “Arduino Blog >>3 simple filtering techniques to eliminate noise,” ARDUINO, 5 September 2016. [Online]. Available: <https://blog.arduino.cc/2016/09/05/3-simple-filtering-techniques-to-eliminate-noise/>. [Accessed 29 April 2019].
- [65] J. Treadway, “Using an IoT gateway to connect the 'things' to the cloud,” TechTarget, [Online]. Available: <https://internetofthingsagenda.techtarget.com/feature/Using-an-IoT-gateway-to-connect-the-Things-to-the-cloud>. [Accessed 16 April 2019].
- [66] B. Butler, “What is edge computing and how It is changing the network | Network World,” IDG Communications, Inc., 21 September 2017. [Online]. Available: <https://www.networkworld.com/article/3224893/what-is-edge-computing-and-how-it-s-changing-the-network.html>. [Accessed 14 February 2019].
- [67] Nasaruddin, M. Andriani, Melinda and M. Irhamsyah, “Analysis of energy efficiency for Wi-Fi 802.11b multi-hop networks,” in *IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, Yogyakarta, Indonesia, 2013.
- [68] M. Rouse, “What is a gateway? Definition from WhatIs.com,” TechTarget, [Online]. Available: <https://internetofthingsagenda.techtarget.com/definition/gateway>. [Accessed 19 May 2019].
- [69] R. PLUS, “Powering Up Smart Cities with Lithium - ion Batteries,” TOSHIBA, [Online]. Available: <https://www.reuters.com/brandfeatures/road-to-a-new-day/powering-smart-cities-with-lithium-ion-batteries>. [Accessed 02 May 2019].
- [70] I. Group, “Smart Parking - The Power of Smart Technology IPS Group,” IPS Group, Inc, [Online]. Available: <https://www.ipsgroupinc.com/the-power-of-smart-technology/>. [Accessed 01 May 2019].
- [71] M. Luciano, “Top 10 Smartest Cities In The World,” Advantage Business Marketing, 15 Novemebr 2017. [Online]. Available: <https://www.ecnmag.com/blog/2017/11/top-10-smartest-cities-world>. [Accessed 7 June 2019].
- [72] G. i. Technology, “UPS for Smart City Device,” Green ideas Technology, [Online]. Available: http://www.3rgit.com/Solutions/Solution/UPS_for_Smart_City_Device. [Accessed 02 May 2019].

- [73] “Intelligent Lighting | Smart Street Lighting System | Street Light Sensors | Twilight,” TVILIGHT B.V. Amsterdam, 16 April 2018. [Online]. Available: <https://www.tvilight.com/>. [Accessed 22 May 2019].
- [74] Comlight, “Motion Sensing Streetlighting System Solution - Comlight,” Comlight AS, [Online]. Available: <https://www.comlight.no/solution/>. [Accessed 1 June 2019].
- [75] “Smart City Applications | Telensa,” Telensa, [Online]. Available: <https://www.telensa.com/applications#telensa-planet>. [Accessed 10 May 2019].
- [76] “Tata Communications I& Jamshedpur Utilities and Services Company (JUSCO) deliver India's largest deployment of smart street lights,” Tata Communications, 2018. [Online]. Available: <https://www.tatacommunications.com/press-release/tata-communications-lights-the-way-for-jamshedpur-utilities-and-services-company-jusco-with-indias-largest-deployment-of-smart-streetlights/>. [Accessed 10 May 2019].
- [77] “Arduino-Introduction,” Arduino, [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Accessed 15 October 2018].
- [78] “Arduino Uno Rev3,” Arduino, [Online]. Available: <https://store.arduino.cc/usa/arduino-uno-rev3>. [Accessed 16 October 2018].
- [79] R. Guide, “Photo resistor- Light Dependent Resistor(LDR)>>Resistor Guide,” resistorguide.com, [Online]. Available: <http://www.resistorguide.com/photoresistor/>. [Accessed 30 August 2018].
- [80] “ESP8266.net - The Internet of Things with ESP8266,” [Online]. Available: <http://esp8266.net/>. [Accessed 12 February 2019].
- [81] M. Griffith, “Lihghting Comparison: LED vs High Pressure Sodium (HPS) and Low Pressure Sodium (LPS),” STOUCH LIGHTING LED LIGHTING SOLUTIONS, 9 April 2019. [Online]. Available: <https://www.stouchlighting.com/blog/led-vs-hps-lps-high-and-low-pressure-sodium>. [Accessed 2 June 2019].
- [82] S. Electronics, “Super Bright LED,” Electronics Tutorials, [Online]. Available: <https://www.sparkfun.com/products/11118>. [Accessed 5 November 2018].
- [83] “ThingSpeak IoT,” The MathWorks,Inc, 2010. [Online]. Available: https://thingspeak.com/pages/learn_more. [Accessed 16 November 2018].
- [84] “Apps - React - ThingSpeak IoT,” [Online]. Available: <https://thingspeak.com/apps/reacts>.
- [85] “What Is a Digital Multimeter? Fluke,” Fluke Corporation, 2019. [Online]. Available: <https://www.fluke.com/en-in/learn/best-practices/measurement-basics/electricity/what-is-a-digital-multimeter>. [Accessed 25 April 2019].

