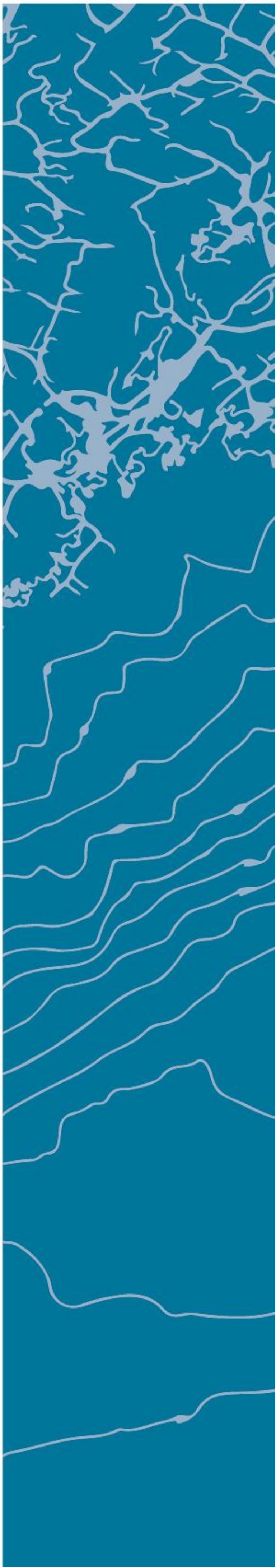
UNIVERSITETET I AGDER

Relevance- and Aggregation-based Scheduling for Data Transmission in IEEE 802.15.4e IoT Networks



MATRIKA SUBEDI
NONSO EZENNABUIKE

SUPERVISOR
Prof. Frank Yong Li

University of Agder, 2017
Faculty of Engineering and Science
Department of ICT

Relevance- and Aggregation-based Scheduling for Data Transmission in IEEE 802.15.4e IoT Networks

By

Matrika Subedi and Nonso Ezennabuike

Supervisor: Professor Frank Yong Li

Master's Thesis
IKT 590

Department of Information and Communication Technology
Faculty of Engineering and Science
University of Agder

Grimstad, 21 May 2017

Abstract

Internet of thing (IoT) is regarded as a new communicating paradigm with Internet connectivity enabling embedded devices to interact with each other on a global scale. IoT has the potential to become the largest producer of information because of a massive number of connected devices with diverse applications ranging from environmental monitoring, home, and building automation. This ubiquitous connectivity requires reliability, efficiency, and sustainability of access to information.

As an enabling technology, wireless sensor networks (WSNs) have opened new opportunity with recent technological developments in making miniaturized smart connected devices. With an increase in the activity of these smart devices, there are challenges in maintaining their limited energy, lifetime, and reliability required for IoT applications. The reason is that these devices are mostly battery powered. In this respect, an insight into the activities of sensing devices produced by different vendors with interoperability based on industrial standards is needed.

As an enhancement of IEEE 802.15.4 MAC sublayer, the ratification of IEEE 802.15.4e standard makes a step towards IoT medium access control (MAC) for industrial applications. One of the significant enhancements in IEEE 802.15.4e is different MAC modes. However, IEEE 802.15.4e does not specify standardized scheduling policy for network building and data transmission maintenance. It is basically application specific. In general, activities performed at the MAC sublayer contribute to sensor energy consumption. Therefore, an efficient MAC scheme is needed to utilize network resources more efficiently, minimize energy consumption level and at the same time improve data transmission of the network. In this thesis work, we focus on proposing transmission schemes for improving energy consumption for data transmission in IoT networks and as well as increasing average packet delivery ratio (PDR). Our target is to improve time slotted channel hopping (TSCH) mode that enables deterministic access and robust network. The focus is on dedicated and shared slots in TSCH.

More specifically, we propose two MAC schemes; relevance- and aggregation-based scheduling for data transmission in IEEE 802.15.4e IoT networks. With relevance-based scheduling, the coordinator node builds and maintains communication in the network based on a historical data value of member nodes. On the other hand, aggregation-based scheduling enables the coordinator node to build and maintain communication by integrating multiple data inside a single frame payload at the source node before transmission.

Further, the proposed schemes are implemented using network simulator version 3 (ns-3). We use Ubuntu 16.04.2 as the operating system for our implementation and performance evaluation. Numerical results for a few performance metrics including PDR, collision probability, delay, and energy consumption are obtained through extensive simulations. The superiority of the proposed schemes is demonstrated by comparing the simulation results with that of IEEE 802.15.4e TSCH standard under varies network scenarios.

Keywords: IoT and WSNs, IEEE 802.15.4e, TSCH, hybrid slots, scheduling, implementation, and simulation.

Preface

This report summarizes the Master thesis work taken at the Department of Information and Communication Technology (ICT), University of Agder (UiA), Campus Grimstad, Norway, from 2 January 2017 to 21 May 2017. The workload is equivalent to 30 ECTS.

Our supervisor has been Professor Frank Y. Li, University of Agder. We would like to express our gratitude to him for his invaluable and profound guidance throughout the thesis period. Frequent meetings and regular discussions have been the foundation on which completeness of this thesis work is built. We wish to thank everyone who has directly or indirectly motivated us for the successful completion of this thesis. Your love, care, and most especially, ability to share knowledge with us are priceless. Finally, we thank our families, and friends for their very special place in our lives. Our classmates have been so fantastic and thank you for all the nice moments together.

Production note: We use Microsoft Word as the tool for writing this thesis. Simulation is done in ns-3 while graphical results are obtained from Matrix Laboratory (MATLAB).

Matrika Subedi & Nonso Ezennabuike
21 May 2017
Grimstad, Norway

Contents

List of Figures

List of Tables

List of Abbreviations

6LoWPAN	Internet Protocol version 6 Over Low-power Wireless Personal Area Network
6TiSCH	Internet Protocol version 6 Over TSCH IEEE 802.15.4e
ABI	Allied Business Intelligence
ABS	Aggregation-based Scheduling
ACK	Acknowledgement
AGT	Acknowledgement Guard Time
AMCA	Asynchronous Multi-Channel Adaptation
API	Application Programming Interface
AppIoT	Application Platform for Internet of Things
ASN	Absolute Slot Number
BE	Backoff Exponent
CFDS	Completely Fair Distributed Protocol Scheduler
CoAP	Constrained Application Protocol
CPAN	Personal Area Network Coordinator
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
DBA	Deterministic Beacon Advertising
DES	Discrete-event Simulation
DevIoT	Development IoT
DiSCA	Distributed Scheduling for Convergecast
DMSE	Deterministic and Synchronous Multi-Channel Extension
DTMC	Discrete Time Markov Chain
EBs	Enhanced Beacons
FastA	Fast Association
FFD	Full Function Device
GMPLS	Generalized Multi-Protocol Label Switching
HART	Highway-Addressable Remote Transducer
IBSG	Internet Business Solution Group
IDC	International Data Corporation
IE	Information Element
IEEE	Institute of Electrical and Electronic Engineers
IERC	European Research Cluster on Internet of Things
IETF	Internet Engineering Task Force

IoT	Internet of Things
IP	Internet Protocol
IPv6	IP version 6
ISA	International Society of Automation
ISM	Industrial, Science, and Medical
LE	Low Energy
LLDN	Low Latency Deterministic Network
LLNs	Low-power and Lossy Networks
M2M	Machine-To-Machine
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MATLAB	Matrix Laboratory
MBS	Model-Based Beacon Scheduling
METIS	Mobile and Communication Enablers for 2020 Information Society
MIT	Massachusetts Institute of Technology
MODESA	Multi-Channel Optimized Delay Timeslot Assignment
MUSIKA	Multi-Channel Multi-Sink Data Gathering Algorithm
NFC	Near Field Communication
ns-2	Network Simulator Version 2
ns-3	Network Simulator version 3
PAN	Personal Area Network
PC	Personal Computer
PDR	Packet Delivery Ratio
PGT	Packet Guard Time
RBS	Relevance-based Scheduling
REST	Representational State Transfer
RFD	Reduced Function Device
RFID	Radio Frequency Identification
RFID	Radio Frequency Identification Blink
RH	Random Horizontal
ROLL	Routing Over Low-power Lossy Network
RPL	Routing Protocol for Low-power and Lossy Networks
RSVP-TE	Resource Reservation Protocol-Traffic Engineering
RV	Random Vertical

TDMA	Time Division Multiple Access
TSCH	Time slotted Channel Hopping
TSMP	Time Synchronized Mesh Protocol
Wi-Fi	Wireless Fidelity
WPAN	Wireless Personal Area Network
WSNs	Wireless Sensor Networks

Chapter 1

Introduction

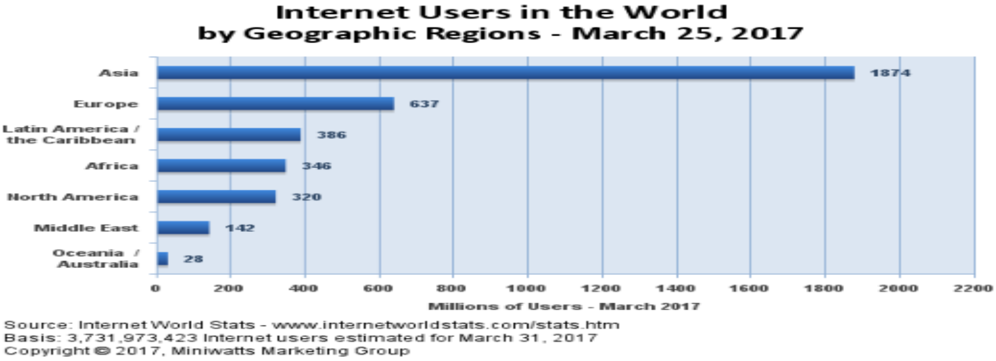
In this chapter, we give an introduction on Internet of Things (IoT) and present wireless sensor networks (WSNs) as the building block. The motivation of the thesis along with brief information about the research topic is also introduced. Besides, the objectives are explained, and thesis organization is outlined.

1.1 Background Information on IoT and WSNs

IoT is defined as "a *dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network*" [1]. It is the most discussed topic today among researchers in technology industries. The concept of IoT has been put forward in early 2000's by Kevin Ashton at the Massachusetts Institute of Technology (MIT) Auto-ID Centre as he was searching for ways to improve Proctor & Gamble business by linking radio-frequency identification (RFID) information to the Internet [2].

In general, IoT signifies a scenario in which physical devices and sensors are connected through wired or wireless to the Internet allowing interaction among them on a global scale with the aim of improving accuracy, effectiveness, and overall economic benefits. It has accounted for an improved environmental monitoring, building and home automation, and smart grids interconnections giving systems better awareness in reliability and sustainability. The early deployment of IoT networks necessitated the connection of industry apparatus. Currently, IoT is no longer a futuristic idea but an increasingly commercial pragmatism [3]. The vision has expanded to connect everything at anytime and anyplace providing ubiquitous services to industrial equipment as well as everyday objects. IoT is transforming today's Internet where objects can make themselves identifiable and attain intelligence because of the fact of the exchange of information and as well access information from other devices. For example, Bigbelly smart waste and recycling system is used in smart cities for waste management, Fitbit Charge heart rate is a wearable IoT device used to track heart rate as well as sitting activities, the smart metering solution offered by Landis+Gyr consumers to better understand their energy needs as well as load management, etc.

The use of Internet is growing over the years. Recent statistics shows that the number of the Internet users at the end of March 2017 exceeded 3.7 billion [4]. Figure 1.1 depicts the geographical distribution of users. Initially, providing Internet facility used to be costly, but today, the size and cost of wireless radios have reduced immensely. The digital world is enjoying tremendous growth due to IoT revolution, and hence, businesses are experiencing an ever-increasing number of smart things interacting and exchanging information explosively.



Source: Internet World Stats - www.internetworldstats.com/stats.htm
Basis: 3,731,973,423 Internet users estimated for March 31, 2017
Copyright © 2017, Miniwatts Marketing Group

Figure 1.1: Internet users in the world by geographical regions [4].

It necessitates an insight into the intelligent of IoT for technological industries. Allied Business Intelligence (ABI) Research estimated above five billion wireless connected chips in 2013 [5]. Automated corporations are building wireless fidelity (Wi-Fi) and cellular connectivity into an extensive variety of devices to improve networks broadband speeds. Still, there are a lot of challenges in creating a smart world where physical, digital, and computer-generated worlds will rather converge than diverge. Therefore, the explosion of the IoT is hugely dependent on how its component will interact. More importantly, how these devices will communicate, coordinate, and collaborate to achieve IoT goal.

WSNs provide the building block for the IoT proliferation by leveraging the flexibility and harmonization of interaction between the connected devices. WSNs transmit sensed data and mainly operate on battery power. In general, IoT will connect these sensing devices, and machine-to-machine (M2M) communication to enhance instantaneous sensing services. Autonomous sensing systems have expressively gained an improved consideration over the years because of reduced power consumption of the electronic devices and advance of wireless communication technologies [6]. With the progress made on fused circuits, wireless communication technologies, nanotechnologies and low cost miniaturized processors have necessitated the development of smart devices to monitor human activities [7]. It has enabled the gathering, analysis, and processing of data thus optimizing an overall performance and effectiveness of a system. Also, IoT as a network of physical objects has its components developed and produced by different companies with enabling wireless protocols, improved sensors, and low-priced processors with compatible management software. The international data corporation (IDC), a technical analyst, has estimated growth of the IoT installed base to approximate 212 billion devices by 2020. IDC sees this growth driven largely by intelligent systems that will be installed and collect data across both consumer and enterprise application [8]. There is a need for interoperability of all IoT components produced by different proprietaries to enable continuous information exchange. Thanks to the successful standardization of the institute of electrical and electronic engineers (IEEE) 802 family.

The standard specified by IEEE 802.15 allows wireless personal area networks (WPANs) to convey information in small distances which enables small, power-efficient, inexpensive solutions to be applied for a variation of devices. Further, it is designed for providing wireless connections to devices while optimizing energy consumption as well as maintaining low-data-rate and relaxed throughput requirements [9]. Most viable solutions of WSNs are built on the IEEE 802.15.4 standard which defines the physical and MAC layer for low-power and lossy networks (LLNs) in WPANs [10]. However, the standard limits the requirements needed for industrial, health applications and home automation with regards to reliability, timeliness, and robustness. To overcome these limitations, IEEE 802.15.4e amendment is designed with additional functionality to the IEEE 802.15.4 MAC to improve and sustain these stringent requirements [11]. Thus, IEEE 802.15.4e is regarded as MAC standard for IoT [12].

1.2 Motivation and Problem Statement

The extensive study within the past years in autonomous sensing systems with much emphasis on the reduction of power consumption, the cost-effectiveness of the devices, and spread of wireless communication technologies has improved lifetime of WSNs. It has revolutionized sensing systems into more compact, smart and intelligent sensors with wireless interfaces to enhance network formation. Research into this area has necessitated the integration of new apparatus and develop the structures to make sensors smarter and perform added logical functions in making decisions.

Most sensor nodes use a battery as the primary source of energy in addition to the power supply through energy harvests from the atmosphere by way of solar panels. Still, there is the need for energy efficient in sensor networks via new design concepts, improving current protocols and evolving new schemes. MAC schemes coordinate energy consumption activities, consequently playing an important role in the lifetime of the network. Thus, an alternative to the IEEE802.15.4 MAC is the development of IEEE 802.15.4e MAC scheme that combines time-slotted access with multichannel to prevent multipath fading and external interference which is more power efficient and reliable. Nevertheless, there are still open issues towards developing energy efficient IEEE 802.15.4e MAC schemes to meet the needs of an IoT applications. Primarily, lightweight scheduling solutions for quick node establishment should be designed and, an improved data communication among others. Motivated by tremendous research in MAC layer which is very crucial in WSN for IoT applications, this Master thesis tries to discourse some of these challenges. In this regard, the following research questions are considered:

- Q1: Considering that industrial applications demand stringent requirements, how can we use a hybrid scheme to integrate both dedicated and random access for data transmission in IoT networks?

- Q2: Giving the fact that there is no standardized scheduling scheme, how can we propose a scheduling scheme to reflect historical characteristics of data?

- Q3: Considering the size of frame transmission, how can we propose a scheduling scheme that considers small data transmission in IoT networks?

- Q4: How can we evaluate the performance of the designed scheduling schemes?

1.3 Objectives and Methodology

IEEE 802.15.4e is redesigned from the existing IEEE 802.15.4 MAC scheme with to overcome its limits. Further, the set goal defines a low-power, reliable, robust and multi-hop MAC scheme capable of sustaining the requirements for attaining IoT application needs. Achieving energy efficiency and reliability are not limited to MAC layer only but also to activities performed across all layers. The scope of this thesis mainly focused to the MAC layer with assumptions of the star topology, single channel, and static nodes in a small cluster that are extensible. Correspondingly, the goal of this Master thesis is to propose MAC schemes that extend IEEE 802.15.4e ideas by reducing energy consumption at the same time, increase packet delivery ratio (PDR), and addresses the mentioned research questions. More precisely,

- Goal1: To develop suitable slot scheme that considers both time division multiple access (TDMA) and carrier sense multiple access with collision avoidance (CSMA-CA) for IoT applications. Thus, it requires the knowledge of device application scenario with unique requirements from other application scenarios, thus making the device more suitable for realizing IoT goals.

- Goal2: To develop scheduling scheme that considers packet transmission based on the historical data value in IoT networks. In doing this, knowledge of timeslots allocation aimed at dedicated and shared in the slot frame are required for efficient scheduling.

- Goal3: To develop scheduling scheme that considers aggregation of small data in IoT networks intending to increase PDR and minimize energy consumption.

- Goal4: The developed schemes are implemented in ns-3. Besides, Matrix Laboratory (MATLAB) environment is used to generate result graphics for performance metrics comparisons.

1.4 Document Structure

The remainder of this thesis report is as follows.

- Chapter 2 describes related work and enabling technology with an emphasis on academic efforts and industrial/standardization activities.

- Chapter 3 introduces reference model, relevance, and aggregation schemes in IoT networks. All key factors necessary for a reliable and robust network are discussed.

- Chapter 4, implementation of the reference model, relevance-based scheduling (RBS) and aggregation-based scheduling (ABS) schemes using discrete-event simulation (DES) are presented.

- Chapter 5 presents an analysis of reference model, RBS, and ABS schemes.

- Chapter 6 concludes the thesis highlighting our contributions and potential future work.

Chapter 2

Related Work and Enabling Technologies

In this chapter, we present details of various efforts in both academia with an emphasis on fundamental approaches in IEEE 802.15.4e networks and industries/ standardization activities towards IoT goals. Additionally, the enabling technologies with specifications are discussed. The abundant smart environments, equipped with affordable, ease of establishing WSNs and ubiquitous mobile ad hoc networks (MANETs) are opening innovative opportunities in the development of new IoT applications in diverse domains. This phenomenon must do with the collection of data by several communicating sensors which is still a thought-provoking task because of some open technical topics, for instance, bandwidth, energy, delay, throughput, etc., limiting the practical deployment of IoT scenarios. Also, as the number of connected and communicating devices gradually increase, the size of conveyed data also grow exponentially.

Thus, to lessen the strain on infrastructure and increasing latency due to large data, future IoT devices will need to handle data generation and transmission with small data solution. As such, future IoT devices would be appraised on their capability to deliver real-time results making small data very effective and efficient in information transfer. With the ever-growing popularity of IoT, an enormous amount of research and developmental work are on-going both in academia and industries around the world to realize the objectives of IoT. Based on the survey of relevant literature, there have been definitions, illustrations and as well as predictions of IoT in business environment, industrial, and home automation applications.

2.1 Academic Efforts on IEEE 802.15.4e

Most IoT applications are deployed in locations where energy is scarcely available and hence, require energy to operate. Therefore, the policies employed in designing such applications should consider a power efficient mechanism for both sensing and transmission of data. Many research works have been done and are ongoing in this regard for the development of full system solutions to support IEEE 802.15.4e TSCH networks. The approaches followed in designing new scheduling schemes especially for this, are not limited either a centralized or distributed method but also, issues of network formation, synchronization and performance are equally considered.

2.1.1 Centralized scheduling

In centralized scheduling scheme, a central node creates and informs scheduling updates based on information supplied by other nodes in the network. In [13], the authors propose a multichannel technique to provide a small gathering cycle and a higher throughput. Hence, the paper tries to tackle slot assignment model that reduces the number of assigned slots but also ensures that no conflicting nodes will transmit at the same time using the same channel. Nodes in multichannel optimized delay timeslot assignment (MODESA) assume a dynamic priority calculated based on the number of packets it must transmit. Following this, consideration is given to nodes having packets to send to a parent with many descendants. The authors in [14], majorly focus on a multi-sink multichannel context with dedicated traffic per sink. Multichannel multi-sink data gathering algorithm (MUSIKA) is a slight modification of MODESA with more emphasis on multiple sinks to improve the robustness of data collection and different application functionalities distribution. Through linear programming formulation, they aimed at realizing the shortest cycle length to propose MUSIKA algorithm with a possibility of achieving collision-free schedule through smallest frame length.

Traffic Aware Scheduling Algorithm that builds a schedule based on the network topology and the traffic load generated by the nodes has been proposed in [15]. The authors consider tree topology network which could be represented in an oriented graph. (i.e., G = (V, E), V represents nodes while E is the connecting links, all links are dedicated). More, the paper builds its schedule avoiding conflicts due to duplex or interference by employing *matching* and *colouring* procedure. In [16], a centralized adaptive multi-hop scheduling was proposed by reserving communication resources for each set of end-to-end communication links. Virtual resources are assigned to weak links as preventive procedures to prioritize those with more traffic demand or less link quality with the aim of allocating more resources to them and thus, improve communication reliability and achieve ultra-low latency. The authors used the idea of tentative schedule in a cell that allocates backup cells for every active cell during scheduling.

2.1.2 Distributed scheduling

Distributed Scheduling must do with individual nodes performing link scheduling upon receiving its neighbour status in TSCH networks. In [17], the authors propose the use of generalized multi-protocol label switching (GMPLS) and the resource reservation protocol-traffic engineering (RSVP-TE) to schedule TSCH in a distributed fashion to match nodes requirement in a single sink network topology. By using GMPLS, a label is assigned to each node making it unique thereby aiding information transportation per switching rule while RSVP layer establishes a path towards the sink with described requirements of the track. These requirements are passed to GMPLS for upwards transmission to the sink. Completely fair distributed scheduler (CFDS) provides timeslot and channel offset selection at every node. Decentralized traffic-aware scheduling algorithm which constructs optimum multi-hop schedules in a distributed fashion is proposed in [18]. The authors aimed at building a conflict-free TSCH schedule where every node can send its data to one precise sink. The authors in [19], propose a distributed interference aware joint channel and timeslot assignment (DiSCA) for a traffic-aware converge cast in multichannel WSNs that consider two cases of transmissions (instant acknowledgment (ACK) and without ACK) bound on a minimum number of the timeslot and a sink with multiple radios. The simulation performance shows that DiSCA is optimal when the radio interfaces of the sink are equal to the minimum between the sink children and existing channels.

2.1.3 Network formation

In TSCH, the network is formed when the network coordinator begins advertising network by sending enhance beacons (EBs) where other network nodes join by listening to a valid EB. The authors in [20], propose two algorithms: *Random vertical (RV) filling and Random horizontal (RH) filling*. Both algorithms allow each node to send EB in the first timeslot (advertising slot) of the multi-slot frame while the coordinator always transmits EB in channel offset 0. In RV, other nodes in the network send their EB using the same timeslot, but different channel offset and in RH, other nodes transmit EB on the same channel offset but different timeslots. Their results from performance analysis show that both algorithms show comparable performance. A *model-based beacon scheduling (MBS)* algorithm is proposed in [21] to approximate the optional EB schedule in the actual situation. Through *discrete time Markov chain (DTMC*), they derive an analytical expression for average joining time. In [22], *deterministic beacon advertising (DBA)* is proposed to ensure that beacons are transmitted on all frequencies used in TSCH network void of collision with performance analysis in ns-3.

2.1.4 Network synchronization

TSCH adopts either data frame or an ACK synchronization approach. It requires TSCH node to synchronize to it time source device. The authors in [23] through their research work have shown that a factor of 10 can minimize a synchronization period through measurement in a real testbed. The proposed adaptive synchronization allows neighbour nodes to track their drift rate and use it to increase keep-alive messages. In [24], adaptive synchronization in the multi-hop network is proposed. The paper considers a synchronization procedure where a node studies and predicts its neighbour relative clock drift. The solution allows construction of a single node as a coordinator while others synchronize to it thereby building up a tree topology via routing protocol for low-power and lossy networks (RPL) [25], with each child node using its parent node as a time source.

2.1.5 Network performance

Performance analysis in TSCH has been carried out to estimate its performance parameters, i.e., latency, power/ energy consumption and throughput with regards to industrial applications. Detail of the energy consumption model in TSCH networks is presented in [26]. Through slot based step-by-step, the authors obtain an analysis of energy consumed by using different slot types. A model that estimates the performance of a complete network has been presented in [27], with the assumption that topology, wireless quality, and traffic demands are known. The model has been applied to s*mart mesh Internet Protocol (IP)*, a commercial TSCH product for validation and captured in a tool referred to as *smart mesh power and performance estimator.*

2.2 Industrial/Standardization Activities

According to European research cluster on the IoT (IERC) definition stated in the introduction, the word *dynamic* emphasizes how the network is developing, varying, adopting, scaling and reconfiguring. Further, *self-configuring capabilities* illustrate the attributes needed for IoT applications for example; self-healing, self-organizing, auto-configuration, context aware, etc. Consequently, it depicts essential physical features identified with "Things" prompting sensing, actuating, interaction, and communication. With *virtual personalities* signifying that identity of the item may alter as it travels through the atmosphere and *use intelligent interfaces, and are seamlessly integrated into the information* network as it will be incorporated and implemented in everyday life [1]. Figure 2.1 below illustrates IERC definition.

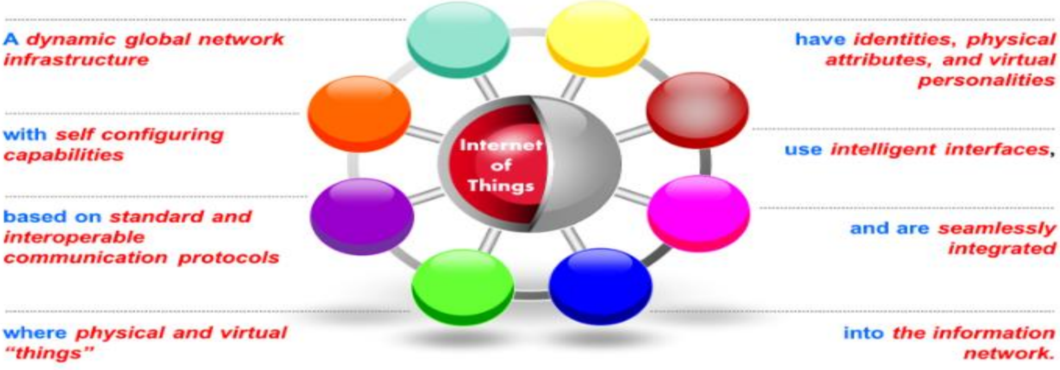


Figure 2.1: IERC IoT definition [3].

The IERC vision is to realize the aims of IoT in the creation of the smart world and self-aware things aimed at digital society, buildings, health applications, etc. It, in turn, will propel the development of innovative technologies that are desirable in addressing the evolving challenges around the world. Besides, the wireless band enhances transportation of generated data with service infrastructure to take care of designing, connecting, monitoring and examining IoT deployment. The level of generated data would increase tremendously with IoT developing into the more stable smart environment. Thus, expanding the nature of business models into more dynamic networks of establishments with massive data communication among smart technologies across different borders. Given this analysis from several leading industries (Acatech, Cisco, Ericsson, IDC, Forbes, etc.) have shown that IoT paradigm will be making big impressions to support changes and technical advance giving rise to an annual estimation of about 20% business growth [9].

In this regard, Cisco has established development IoT (*DevIoT)* which is an IoT development environment to aid design and deployment of complex IoT solutions [28]. More, according to the Cisco Internet business solution group (IBSG), there will be about twenty-five billion connected devices by 2015 and fifty billion by 2020 [38], as shown in Figure 2.2

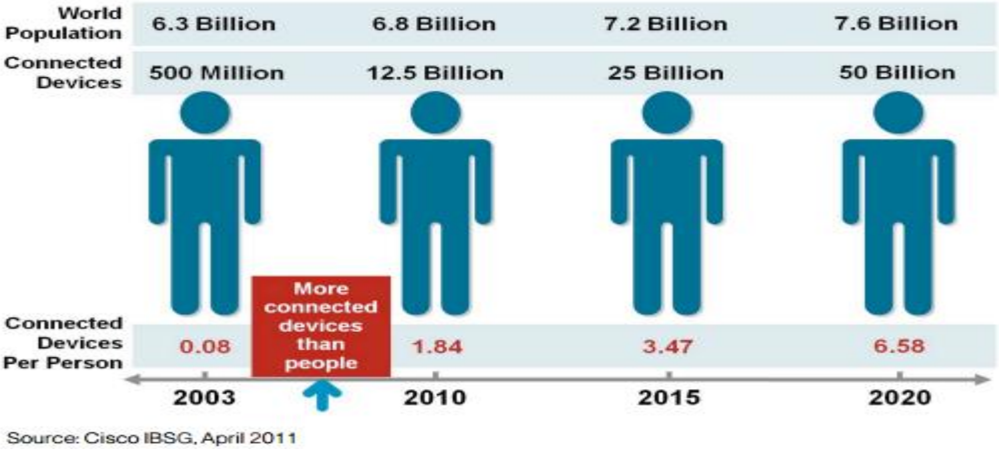


Figure 2.2: Cisco connected devices prediction [38].

It is reported that the value of Internet-connected devices exceeded the number of human beings around the globe in 2011. Cisco has an opinion of IoT as a point in the period when much “things or objects” are linked to the internet more than a human being. Further, with the explosion of affordable sensors allowing infinite sensing of every type of objects, thus signifying the Internet of everything where the Internet could be a robust strength to individuals, earth, and government. In recent years, there has been a shift in the development of smart technologies, as these devices are now less dependent on people with millions performing inbuilt functions autonomously. As recorded by Cisco with all the advances, in 2013 only one-third of the global population has access to the Internet, meaning that more connections are still ongoing [29]. From Cisco, mobile data network analyses, traffic data grew by 60% in 2016 and reached 7.2 exabytes per month from previous 4.4 exabytes per month in 2015. Equally, nearly half a billion (429) portable devices and connections have been added in 2016 with the majority being smartphones seconded by M2M components. Thus, from statistics, smart objects represent 46% of the whole mobile devices around the globe in 2016, hence, account for 89% of the total data traffic with a network speed of 6.8 megabits per second (Mbps) up from 2.0 Mbps in 2015. Therefore, their calculations for 2021 among others include; monthly global data traffic of about 49 exabytes while annually, it will exceed half a zettabyte, average connection speed to be more than 20 Mbps, etc., [30].

According to Mobile and wireless communications Enablers for 2020 information society (METIS), mobile and wireless communication systems will influence societal development leading to the propagation of critical services for examples, e-banking, e-learning, and e-health to turn out to be more mobile thereby making life more comfortable and safer. Also, beyond 2020, mobile and wireless communication must respond to the traffic explosion through capacity and sufficient energy enhancement as well as bandwidth utilization [31].

The total transformation of the smart networked world is defining a new trend for the emerging smart grid, sustainable mobility strategies, smart health and in a manufacturing setting. Thus, this trend in making versatile network of devices, end on engineering integration across entire web is creating the fourth stage of industrialization "*Industrie 4.0*". *Industrie 4.0* is creating smart products, measures, and processes capable of working in a more efficient manner to minimize complexities that surround performance [32]. With an increasing demand for IoT solutions, Ericsson has established the Ericsson application platform for IoT (AppIoT) for rapid IoT application development. AppIoT uses standardized open representational state transfer (REST) application programming interfaces (APIs) where any operating system (Linux, Window, etc.) can implement code written in any programming language [33]. Also, Telenor has created Telenor Connexion which is an IoT dedicated company with the aim to design and run IoT solution on an international scale. The company expertise extends to automobile, security, fleet and asset management, manufacturing industry, etc., implementing her connexion for optimal and innovative IoT solutions [34]. In line to create a smart grid, Soria consortium has signed an intelligent metering contract with Eltel's power distribution for an estimated 282, 000 meters in Norway [35]. An Internet protocol and other standard metering protocol-based solutions are being used. The trend is opening more smart environments around the globe.

Gartner emphasized that over 50% of Internet connections are readily in place. Consequently, in 2011, there were more than 15 billion things on the web, with 50 billion+ intermittent connections. Analysis of the growth trend shows that by 2020, there will be over 30 billion connected things, with more than 200 billion intermittent connections. The driving technologies are embedded sensors, image recognition and near field communication (NFC) [36]. Further, he projects tablet and smartphones to rise to 26 billion units without personal computers (PCs) in 2020. These will generate huge revenue to the tune of more than $300 billion, comprising services in 2020 resulting to $1.9 trillion global market recuperation in sales. The trend will see a reduction in component costs, to the level that even processors will cost less than $1 making connectivity a standard feature in so doing create the possibility of connecting almost anything [37].

With the increasing smart environments, IoT must be supported with standards to enhance interoperability of devices produced from different vendors. Consequently, these rules will be applied to all spectrum and allow interoperability to consist of policy, protocols, and architecture [9]. Regardless of the possible benefits, full-scale IoT technology will generate, it still faces several problems. These challenges will limit proper transmission and reception of data from embedded processors and sensors as IoT depends on Internet connections. Three major initial setbacks are the deployment of IP version 6 (IPv6), power for sensors and accepted standards. Currently, the public routable IPv4 address is no longer feasible since February 2010. Sensor energy remains one of the most concern issues for IoT, so, potential sensors need to be self-sustaining [38]. However, with emerging new technologies, and several ongoing research across the globe, IoT network is in progress. Other wireless technologies that will aid IoT include near field communication, ZigBee, Z-Wave, Blue tooth, Wi-Fi, etc.

2.2.1 Standardization

There have been several standards to support the development of WSNs over the years by international bodies in different application fields. In 2007, the highway addressable remote transducer (HART) communication foundation published WHART standard [39]. Furthermore, there are the international society of automation (ISA-100.11a) [40] to address wireless manufacturing and control systems, Bluetooth [41], ZigBee [42], and IEEE 802.15.4 [9]. Similarly, the Internet engineering task force (IETF) has developed several protocols to enable seamless communication between the smart devices with the Internet [43]. These protocols are necessary for cross-layer communication in IoT applications. They are, IPv6 over low- power WPAN (6LoWPAN) [44] which forms an adaptation layer protocol for packets, the RPL [25], and the constrained application protocol (CoAP) [45] which allows web transmission on smart constrained devices. IPv6 provides ample address space for every IoT device; besides, CoAP, 6LoWPAN, and Routing Over Low-power Lossy Network (ROLL) provide enabling complete stack for IoT applications. With IETF protocols, the TSCH MAC schemes can be seamlessly integrated with the Internet Protocol version 6 Over TSCH IEEE 802.15.4e (6TiSCH) protocol stack for optimal reliability and energy consumption [46], as shown in Figure 2.3.

CoAP	
UDP	
IPv6	RPL
6LowPAN	
6TiSCH 6top	
IEEE 802.15.4e TSCH	
IEEE 802.15.4 PHY	

Figure 2.3: 6TiSCH protocol stack [41].

2.3 IEEE 802.15.4 Slotted CSMA-CA: MAC in WSNs

The IEEE 802.15.4 standard [9] is designed to accommodate constrained devices for WPANs defining the physical and MAC layer protocol stack and thus viewed as the key standard for commercial WSNs. There are several compliant applications referenced to the performance and analysis based on this standard, for example; [47], [48], [49], [50], etc. However, IEEE 802.15.4 is limited and cannot fulfil many critical requirements such as communication reliability, latency, and interference due to multi-path for industrial applications and home automation. Therefore, to overcome these deficiencies, IEEE set up a working group with the name *802.15 Task Group 4e* [10] with the mandate to define the MAC amendment to the existing 802.15.4-2006 standard and add functionality to suit the emerging needs of the smart devices. The work of this task group resulted in what we now referred to as IEEE 802.15.4e standard [11], MAC amendment, 2012. It is only a MAC protocol change and does not require any modification in the physical layer. The general enhancement introduced in IEEE 802.15.4e standard specify that the added functionality is not tied to a particular domain and different MAC modes to support application areas. This thesis work mainly focuses on IEEE 802.15.4e standard, and specifically, it will pay more attention to the TSCH mode.

2.4 IEEE 802.15.4e: MAC in IoT

The IEEE 802.15.4e standard adopts its concept from existing standards. In 2006, Dust Networks introduced time synchronized mesh protocol (TSMP) [51] with WHART and ISA100.11a protocols taking it as their basis for operation. These concepts are slotted access, shared and dedicated slots, frequency hopping, and multi-channel communication. Importantly, 802.15.4e MAC amendment introduces different MAC modes specifically, to allow a design of MAC protocol to support single application areas and overall functional enhancement not tied to any specific application domains. This exclusive MAC layer enhancement permits IEEE 802.15.4e to work with an IPv6 enabled upper protocol stack for data exchange.

2.4.1 MAC enhancements in IEEE 802.15.4e in brief:

- *Low Energy (LE).* LE mechanism is not unique to any application domain but designed for applications that can trade latency for efficient energy. Thus, it permits smart objects to function at 1% or less duty cycle yet appears to be on to the upper layers.

- *Information Element (IE)*. It provides an extensible prospect to include information at the MAC sublayer.

- *Enhanced Beacons (EB).* It is an extension of the 802.15.4 beacon frames but with greater flexibility. EB has its version frame field set to 0b10 and provides application-specific frames using relevant IEs.

- *MAC Multipurpose Frame*. It provides an extensible and flexible frame format to address several MAC operations based upon IEs.

- *MAC Performance Metrics*. It is used to provide adequate and necessary information about channel status to accommodate data frame transmission and reception.

- *Fast Association (FastA)*. FastA is an improvement in 802.15.4 association process to save a significant amount of energy. Used for applications that prioritize time over energy consumption.

2.4.2 IEEE 802.15.4e MAC modes

The five new MAC modes behaviour are listed below.

- *Deterministic and synchronous multi-channel extension (DMSE).* The mode is designed for both commercial, industrial, and healthcare applications with the primary requirement relating to flexibility, deterministic latency, and reliability. It is very suitable for multi-hop and mesh networks with contention based and time-division medium access.

- *Low latency deterministic Network (LLDN).* LLDN is typically modelled for factory automation with a star topology and single channel. The applications in LLDN require very low latency and medium access through TDMA scheme defined by super-frame.

- *Asynchronous multi-channel adaptation (AMCA).* It is targeted to large infrastructure where using the common channel to connect to the same PAN may be unrealistic. Thus, the variance of channel quality is enormous prompting asymmetric transmission between neighbouring nodes (improper transmission and reception between neighbouring nodes). AMCA uses asynchronous multi-channel adaptation with non-beacon-enabled PANs. The devices in AMCA network opt for the channel with the best local link quality. Cases of AMCA applications include smart utility networks, infrastructural monitoring networks, and process control networks.

- *Radio frequency identification blink (RFID).* It provides a mechanism for an object to its ID to others without first association and prior ACK. Typically designed for the item and people identification, location and tracking. Its packet is mostly on ‘transmit only’ through Aloha protocol.

- *TSCH.* TSCH behaviour mode is designed for industrial application domains including process automation facilities and control. The mode facilitates time-synchronized communication and multi- channel to offer network timeliness and reliability, with the multi-hop operation to lessen the effect caused by interference and multipath fading.

The thesis will concentrate more on TSCH behaviour mode; thus, its application domain and functions are presented in the next section.

2.5 Time Slotted Channel Hopping

TSCH application domain is targeted to oil and gas industry, food and beverage products, chemical products, green energy production, pharmaceutical products, water/waste treatments and climate control [11]. It uses *time slotted access* along *multi-channel and channel hopping capabilities.* The power consumption of the TSCH devices can differ from microseconds to milliseconds with an accurate *time synchronization* depending on the solution employed [52]. Through the combination of time-synchronized communication and channel hoping, it offers network robustness via different frequencies.

Multi-channel communication is one of the significant enhancement in TSCH thereby using frequency diversity to ease the effect of interference and multi-path fading. It leads to network capacity increment as nodes can transmit using different channel offset even with the same timeslot. In TSCH, a link definition is associated with a pair of a timeslot on a specified channel offset [11]. Therefore, for communicating nodes, their link can be detailed as a pair of the timeslot and channel offset (*chOf*) used at a given time. It can be denoted as a set; [*t*, *chOf*] and it is related to the frequency (*f*) used in communication with the following function

$$f = F\{(ASN + chOf) mod n_{ch}\} \qquad (2.1)$$

Where the absolute slot number (ASN), is the total number of slots that pass from the start of the network. ASN increases on a global scale in the network. Function F is implemented as a lookup table where n_{ch} is the size of the table.

In detail, $ASN = (K * S + t)$

Where k is the slot frame cycle, S is the slot frame size, and t is the slot offset. ASN is initialised to 0 and increases by 1 at each timeslot. IEEE 802.15.4e network specifies 16 channels, but some channels may be blacklisted to allow co-occurrence with others and Equation (2.1) yields a different frequency for the same link, thanks to channel hopping technique.

TSCH is not limited to a particular topology as it is well adapted to different network topologies (e.g., star, mesh, tree). TSCH MAC, therefore, is well suited to multi-hop networks inter-relating with higher networking protocols to form power-saving and strong networks with Internet connectivity and seamless transport to support IoT applications. Additionally, its technique helps to minimize duty cycle of a network to less than 1% by keeping the radio off most times [12]. There is the existence of some industrial TSCH technology products for consumer usage [53]; still, many studies are ongoing to finding an ideal strategy for the ubiquitous IoT application deployment.

Chapter 3

Relevance-based and Aggregation-based Scheduling Schemes in IoT: Concept and Design

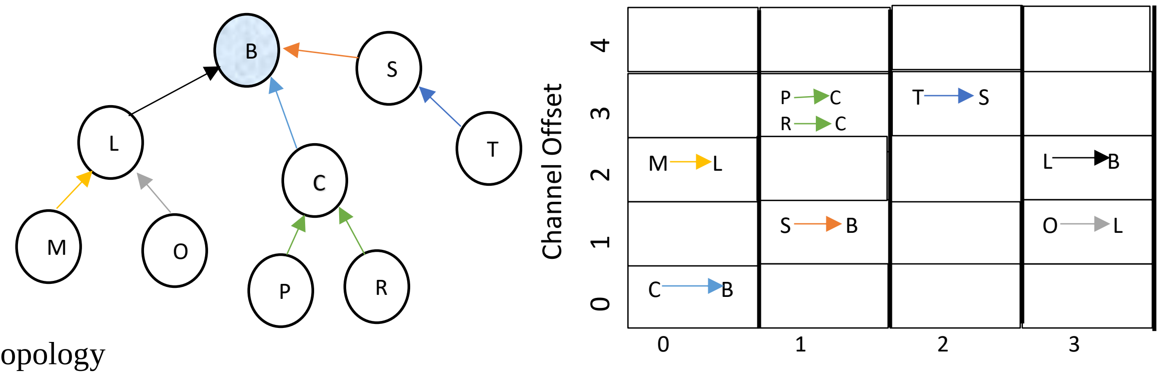
In this chapter, the thesis will explain the scheduling techniques adopted for a reliable and robust TSCH mode schemes. More precisely, we propose two scheduling schemes with emphasis on historical data value and data aggregation. Also, two versions of timeslot assignment are considered; fixed and variable (more advanced) timeslots.

3.1 Preliminaries

Several research works in TSCH mode have been necessitated by the fact that there is no standardized policy to build and maintain communications. So, scheduling schemes are developed based on application domain requirements.

3.1.1 Slot frame construction and synchronization

TSCH mode does not use the super frame. It adopts the use of slot frame concept where nodes synchronize on a periodic slot frame with several defined timeslots. A timeslot is adequate to accommodate a MAC frame of a maximum size to send a packet and receive an ACK from one node to another. These defined periods regarding slots could either be guaranteed or shared among the devices. Figure 3.1 shows an example of topology and slot frame with 4 timeslots and 5 channel offsets in TSCH mode. The availability of multi-channel approach used in TSCH guarantees multiple transmission within a slot (i.e., 8 transmissions in 4 timeslots in this example). Each node concentrates in the cell it is attached. A cell is a single element in TSCH identified by a *SlotOffset* and a *chOf* [54]. For instance, node C uses slot 0 to communicate with node B in channel 0. Most nodes assume the similar position in the dedicated slot, while other nodes share the same slot as seen for node P to C, and R to C, in order words, link [1, 3]. Also, if node O has a packet for node B, the packet is initially sent to node L where it is buffered before sending it to node B in the next frame. To avoid a collision on packet re-transmission in the shared slot, the standard has defined backoff algorithm to be enabled if an ACK is not received within a predefined time. The retransmission is differed to next cycle in the same timeslot between sender/receiver.



(a) Topology

(b) slot frame

Figure 3.1: Tree topology and slot frame with four timeslots and five channel offset [12].

Figure 3.2 shows typical slot frame with 4 timeslots. A slot frame automatically repeats based on the number of slots. Very importantly, the number of timeslots in a slot frame regulates its repetitions.

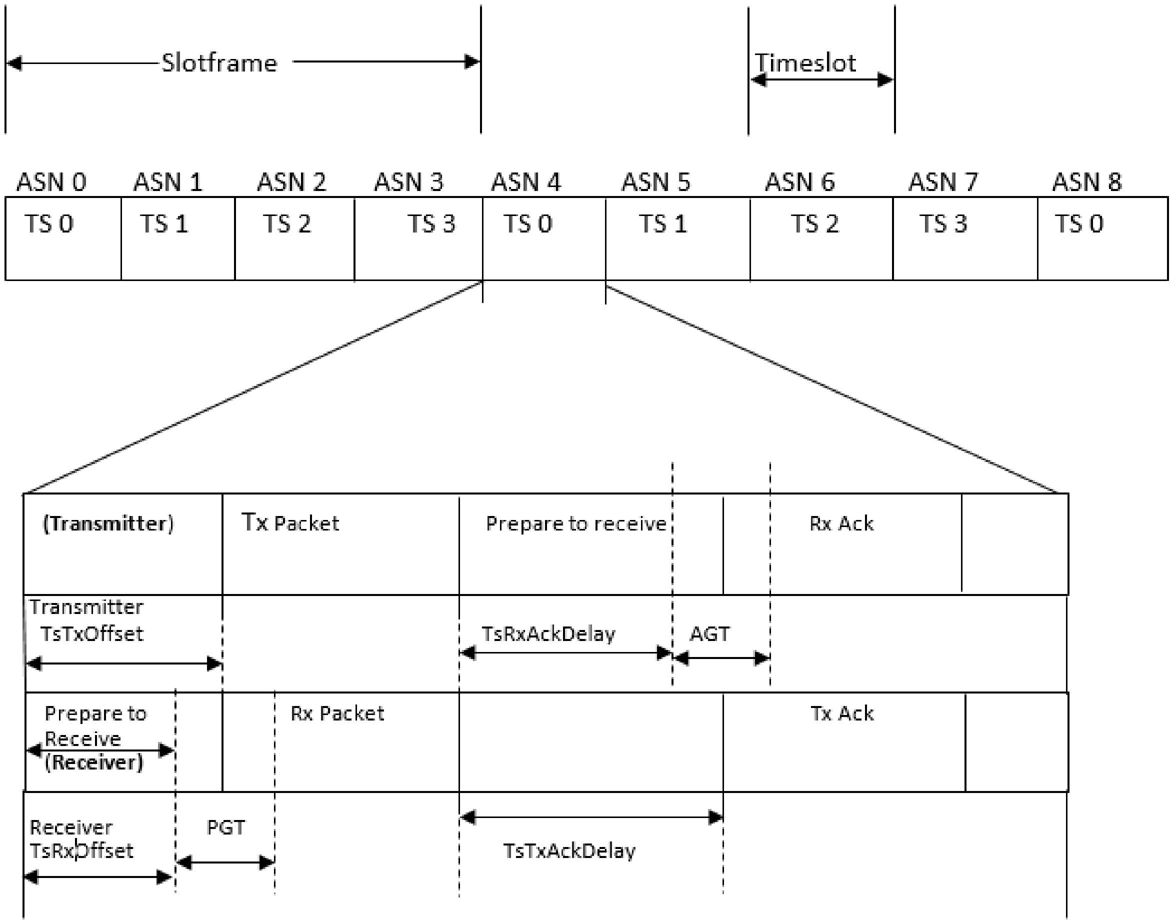


Figure 3.2: Illustration of slot frame and timeslot for transmitter/receiver, showing transmission activities within a single slot [12].

The figure highlights what happens between sending and receiving devices. The transmitter turns its radio on and starts transmitting its packet within *macTsTxOffset ms*, while the receiving device turns its radio to listen for *macTsRxOffset ms* and after *packet guard time* (PGT), if no packet is received, it turns off its radio. Then, if a packet is received, the receiver node waits for *macTsTxAckDelay ms* and prepare to send an ACK. While at the transmitter after sending, the packet will wait for *macTsRxAckDelay ms* to prepare for an ACK and if after *ACK guard time* (AGT), no ACK is received, it turns its radio off. *Guardtimes ms* give room for desynchronization and as well used to avoid energy depletion. In addition, it is required that desynchronization cannot be greater than *guardtime ms* to enable communication.

Each node in TSCH adapt to schedule which informs it what to do in each slot; either to transmit, receive or sleep. While a sleeping node does not turn its radio on, every active node is scheduled with which neighbour to transmit or receive within a specific channel offset. The duration of a slot is per application scenario, but the *10ms* value is suggested in the draft [52]. Furthermore, TSCH adopts two methods for node resynchronization to its time source neighbour, i.e., *Frame-* and *ACK-based synchronization*. The former comprises receiving node to calculate the difference between the expected time of a frame arrival and its actual arrival, then, align its clock by the difference. The later should do with the receiver calculating the difference between the expected time of arrival and the actual time of arrival and insert the information in the ACK frame to the sender. However, if there exist periods (approximately 30 s) when no communication has happened in the network, nodes must exchange empty data frame (keep-alive message) to re-synchronize.

3.1.2 Link scheduling

TSCH defines how the MAC layer can execute a schedule; however, it does not specify how this schedule is built [52]. Notably, link scheduling is a significant part in TSCH as it must do with the assignment of links to nodes for data transfer and therefore, require to be constructed carefully so that pairing nodes will know when to listen for packet or sleep. More, considering changing networks, scheduling should be thoroughly built and regularly refreshed for new nodes joining and leaving the network. Scheduling algorithm in TSCH can either be *centralized* or *distributed.*

In the centralized approach, a particular node (network coordinator) in the network assumes the responsibility of building and maintaining the network based on the information supplied by all nodes in the network. The coordinator sets the connectivity with the information at its disposal and assigns slots per data generation demands from the member nodes. This scheduling approach is not very suitable for dynamic nodes and vast network because it must be re-built and re-distributed with any slight change in the functioning conditions. But, centralized solution builds very efficient schedules and have been readily implemented and deployed with TSMP industrial applications. While in a distributed approach, nodes separately create link schedule based on the exchange of local information with its neighbours. The scheme is more adaptable for mobile or large networks.

3.1.3 Network formation

During network formation, EB frame is sent periodically by the network coordinator to advertise the network. Any node that wants to join will turn on its radio to scan for EBs. Upon receiving a valid EB, the node synchronizes to the network, sets slot frame and starts sending EBs. EB frame contains all the information necessary for link scheduling, synchronization, channel hopping, and timeslot requirements of all the TSCH devices. IEEE 802.15.4e standard does not only specify the EBs advertising policy but also does not indicate the possible link in the slot frame for sending EBs. Thus, it is vital to optimize network formation procedure in TSCH. Additionally, TSCH network is compliant with a set of devices considered in IEEE 802.15.4 standards [9], i.e., full function device (FFD) and reduced function device (RFD). FFD can be utilized both as a node or network coordinator while RFD can ordinarily perform node function but not functions allotted to the coordinator.

3.1.4 CSMA-CA algorithm

In TSCH, a slot can be assigned to several nodes for transmission and may result in a possibility of packet collision on a shared link. Thus, to avoid packet retransmission collision, CSMA-CA is enabled upon failure of the sending node to receive an ACK after ACK timeout. Nevertheless, retransmission backoff is only applicable in a shared link as there is no delay in transmission on dedicated link. Ideally, the sending node will perform the following retransmission backoff upon transmission failure;

- *The sending node will initialize the backoffexponent (BE) to macMinBE viz, (NB = 0) and (BE = macMinBE).*

- *A random number in the range of* $w \in [0, 2^{BE}\text{-}1]$ *delay on the shared links before trying to retransmit again is generated.*

- *Retransmission is deferred for w shared links until a dedicated link is encountered.*

- *In shared link, if retransmission is successful, the backoffexponent is reset to macMinBE, and it terminates the algorithm. Otherwise, the variable is reorganised i.e., NB = NB + 1, BE = min (BE + 1, macMaxBE). Still, if an acknowledgement is not received after the maximum allowed value, (NB > macMaxFrameRetries), the MAC sublayer on the assumption that the frame has failed will notify the upper layer about the failure.*

It is represented in Figure 3.3. The TSCH CSMA-CA backoff mechanism is only enabled after the sending node has experienced a collision, in order words, it is used to prevent repeated collision different from CSMA-CA algorithm in other standards. The IEEE 802.15.4e standard specifies minimum backoff exponent (*macMinBE*) ranging from 0-*macmaxBE* and maximum backoff exponent (*macMaxBE*) as 3-8 integer values. It is expressed in the number of skipped shared links (i.e., window increases in each failed transmission in shared link while it is unchanged when there is a failure in the dedicated link) [11].

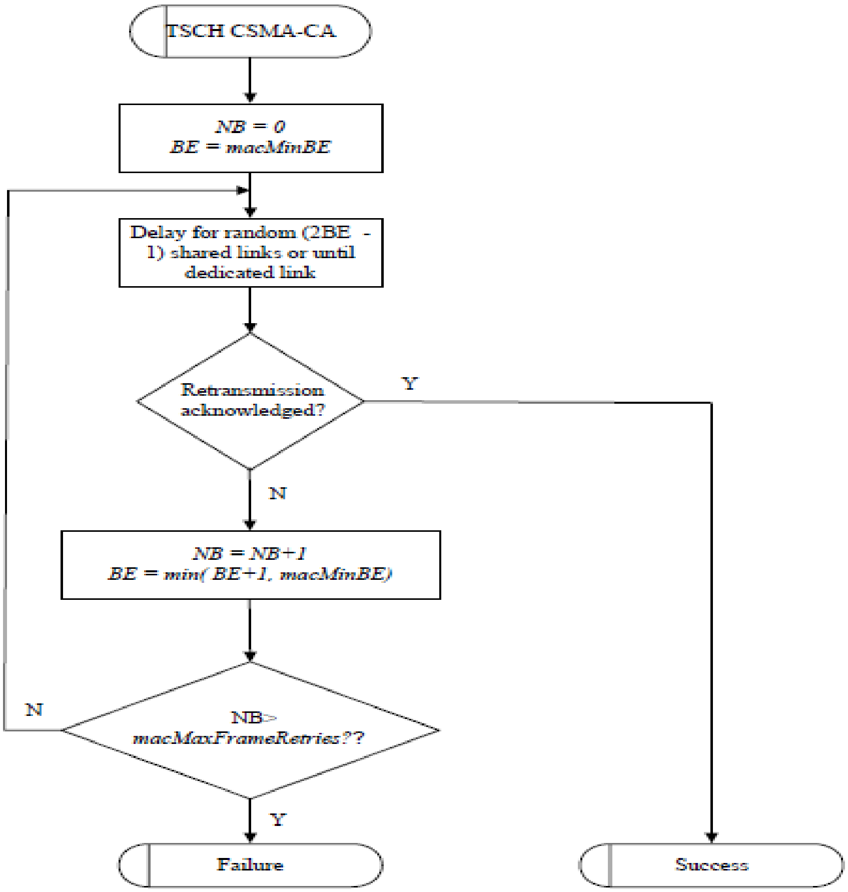


Figure 3.3: Illustrating TSCH CSMA-CA backoff algorithm flow [11].

In general, timeslot features of the TSCH standard including TDMA and CMSA-CA techniques are considered in our design. Time is divided into slots which are long enough for a MAC frame of a maximum size to be transmitted, and an ACK is received between TSCH nodes. Each timeslot has a duration of *10ms* in a slot frame which repeats according to several timeslots. The network supports multipoint-to-point traffic where every active node generates its packet randomly and transmits to the personal area network coordinator (CPAN) or cluster-head within the assigned slot forming a star topology. Nodes in the network communicate only to the cluster-head. Reception of an ACK signifies a successful packet transmission. A centralized scheduling approach is adopted which is effective and deterministic with regards to static networks [16]. A packet is assumed to be the amount of data to be transmitted in both TDMA and CSMA-CA slots. As a major factor in TSCH, this thesis is proposing a scheduling scheme with an emphasis on historical data value and data size. Sequel to many research documents available to us, no earlier paper has considered packet historical data and size in the time domain. Thus, to this, we propose the following scheduling schemes; relevance- and aggregation-based scheduling for data transmission in IEEE 802.15.4e IoT networks.

3.2 Relevance-based Scheduling (RBS)

In this section, we present the proposed RBS scheme designed for IoT networks. In IoT networks, the core functions of network devices are to sense and collect data from their location and then, share the sensed data to Internet for utilization of application purposes in real time. For instance, sensors work directly to monitor temperature, body functions, humidity, etc., in both remote, emergency and hazardous areas. Since the majority of IoT devices operate in constrained environments where power is derived from batteries, RBS scheme is focused on device energy efficiency and network reliability issues.

3.2.1 The concept of RBS

Energy consumption in WSNs is the sum of the energy consumed by devices in different states (i.e., active, sleep, and wake-up periods). In many application domains, efficient energy for network devices is usually the target in the design of WSNs. The concern is because the replacement of devices batteries can be tedious and expensive or even not feasible [55]. In addition, some application domains demand more requirements, i.e., timeliness, scalability, and reliability. Considering all these, sensor networks need to have an extended lifetime to fulfil application requirements. In finding a solution to fit into IoT networks, it is important to answer the fundamental question: “how can network lifetime be prolonged?” Therefore, one of the ways is designing network scheme capable of minimizing the energy of the network. In general, TSCH standard support EBs to provide energy management through duty cycle where each node can be active and inactive at certain periods. Similarly, we will consider energy consumed during transmission for comparative analysis. Specifically, RBS scheme focuses on a small cluster of the network where the cluster-head assigns communication links to other nodes. Through RBS scheme, the devices know what to do at any time to maximize network lifetime. Additionally, RBS scheme considers the amount of data transmitted in the network very crucial. Thus, data reduction scheme to reduce energy spent on packet transmission is much valued.

RBS scheme functions in the source nodes where samples of sensed data are compared to ensure scheme effectiveness. The scheme considers set of historical values obtained randomly in time succession. In order words, the internal structure of sensed data is explicitly considered. Based on the RBS set parameters, the sensed historical data values are checked against stored value and if found to fall within application tolerance, sensed data is regarded valid. The source node will transmit sensed data and update otherwise, source node goes to sleep, and the historical data value is increased. Insight into network architecture will help to elaborate more on the concept of RBS scheme. In this respect, we present network component and topology subsequently.

3.2.2 Network components and topologies

The WSNs components enable wireless connectivity with application platform and sensors in a network. In this, the data path is created between physical world and application platform. The sensed data travel from the leaf node to gateway in a single-hop transmission link. In some cases, the network is extended to include a relay node between the gateway and the leaf node resulting to multi-hop communication. A gateway forms an interface between application platform and wireless nodes while relay node is used to increase network coverage area, route around hindrances and offer backup in case of device failure.

Since a WSN contain several network devices, network topologies should be considered in its design. Common topologies in WSNs are point-to-point, star, tree, mesh or hybrid. Several network attributes (i.e., latency, throughput, scalability, hops, fault resilience, range) are considered in selecting appropriate topology. Understanding their characteristics will be of great benefit. Figure 3.4 shows general TSCH network scenario with different network clusters connecting sink. The cluster-based approach separates sensor nodes into different clusters for data transmission. Cluster-head collects information from member nodes and sends to sink node. RBS considers a cluster of the small network representing star topology in dash line of Figure 3.4. This topology allows a single-hop bi-directional communication with the cluster-head making it accessible for connection within the small network. Also, failure of one node does not affect another as far as it is not the cluster-head and addition of more nodes can be easier as all nodes only communicate to cluster head. Thus, topology is a function of network scheme, so, it is important to consider a scheme that will fit into the constrained nature of IoT applications.

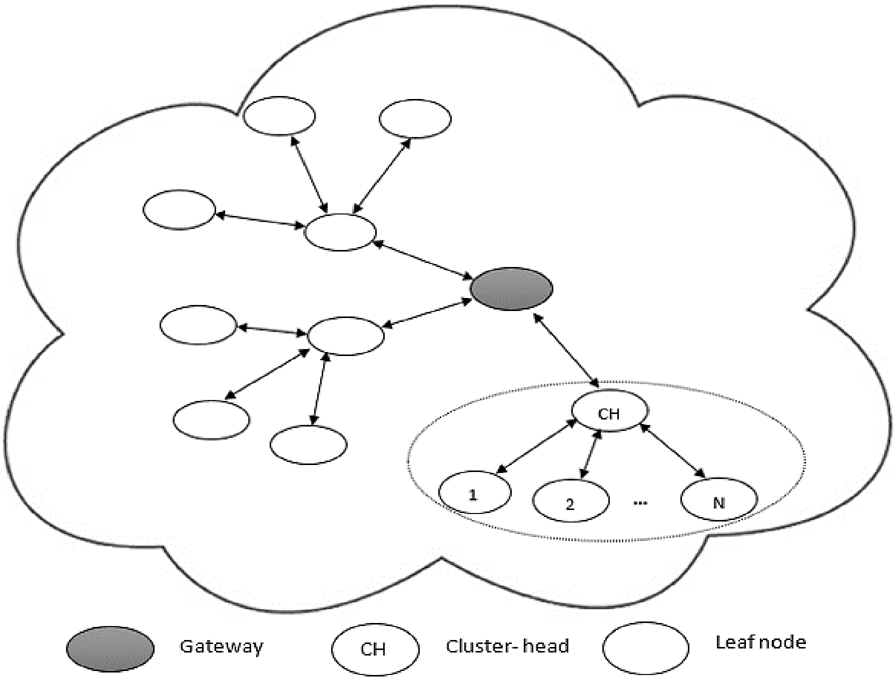


Figure 3.4: Illustration of clusters in tree topology [11].

Correspondingly, the characteristic of sensed data is very necessary as it describes the interaction between sensors and the application scenario. Consideration must be given if the monitored data is by periodic sampling where the exceptional conditions (e.g., temperature, pressure, etc.), depends on how fast they change. In this respect, monitoring conditions vary over time making devices to gain insight into historical data pattern of the sensed data and help to translate it into decisive actions. Also, event-driven where required variables are monitored following a specific event. For example, fire alarm, pressure alarm, etc., but in this, consideration must be given to power consumption used by applications to wake-up in the absence of any occurring event. Thus, in RBS scheme, the range of relevance values will affect the data transmission. It will depend on the set of values between the minimum and maximum limits.

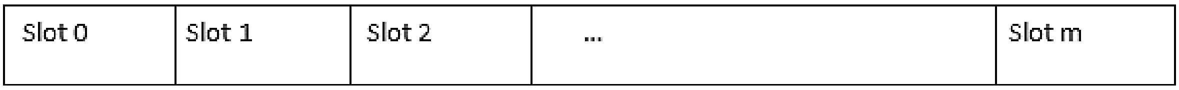
3.2.3 Designing RBS scheme in IoT networks

In this subsection, we present the design principles of RBS scheme in IoT networks. WSNs are characterized with several challenging factors due to its nature. For instance, sensed data flow from all leaf node to cluster head or gateway, and there is redundancy in the generated data since several nodes generate the same data within the same cluster. Moreover, these devices are constrained regarding of energy so, an efficient utilization of power is necessary for the network devices to last for a long time.

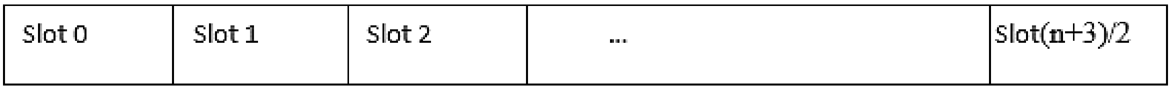
Correspondingly, an extensive survey of the state-of-art where related work, proposed technologies and solutions are studied. It leads to the terminology used in this thesis work with analysis of all the survey as the foundation. Following thesis time frame, we clearly outline design patterns as highlighted in motivation and objectives leading us to IoT conceptual architecture and topology. Then, mapping these IoT components with stringent requirements lead to implementation and subsequent performance evaluation with anticipation to reduce the volume of data transmission. Therefore, RBS strategy is built to perform scheduling function in IoT TSCH networks, where extraction of useful information about historical data content is considered and hence, eliminate redundant data traffic. It ensures that less data traffic occurs in the network, less energy is spent on network devices as nodes with the same value will go to sleep allowing another node with the different data value to send. RBS scheme is compared to reference model under set scenarios and configurable scenarios to check its strength or weakness. It is expected that the designed scheme will optimize various performance metric considered in this thesis and at the same time, respond to the randomness of data generation.

3.2.4 Timeslot allocation and data generation in RBS scheme

In RBS, we assume that all nodes have joined the network and thus, all network statistics have been collected by the cluster-head (i.e. master node has comprehensive topology information). As soon as all the required information has been collected, the network will then change to RBS scheme allocation. The cluster-head uses the first timeslot for EBs and link scheduling, and assign the remaining slots to active nodes with traffic to transmit on the same channel offset. Then, the scheduling information is forwarded to all nodes to know at what time to transmit. The below Figures 3.5 show slot frame. Link allocation is given as a set of [*timeslot, chOft*]. Here, we have two varieties of slot allocation (i.e., a1-fixed and a2-variable).



(a1) Fixed timeslot



(a2) Variable timeslot

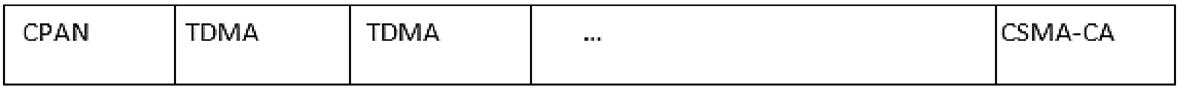


Figure 3.5: Illustration of different slot frames [11].

In general, the principle of slot allocation is similar considering the two versions but only differs in how slot changes. Fixed timeslots constitute a specific number of timeslots which are used by active nodes (e.g., m can be any specific number, with regards to network size). While in dynamic, the number of active nodes determines the available timeslots in the slot frame. For instance, let n denotes the number of network devices then, the available timeslots would be given as;

$$(n+3)/2 \tag{3.1}$$

Further, in these slots, slot 0 is used by the CPAN, the last slot is used to implement shared link while the remaining slots are used for dedicated link assignment. Each node in the network generates data randomly and send the generated data to the cluster-head node. At the nodes, the generated data value is stored in buffer prior transmission. Figures 3.6/3.7 below show data generation and sending principle with regards to the data value.

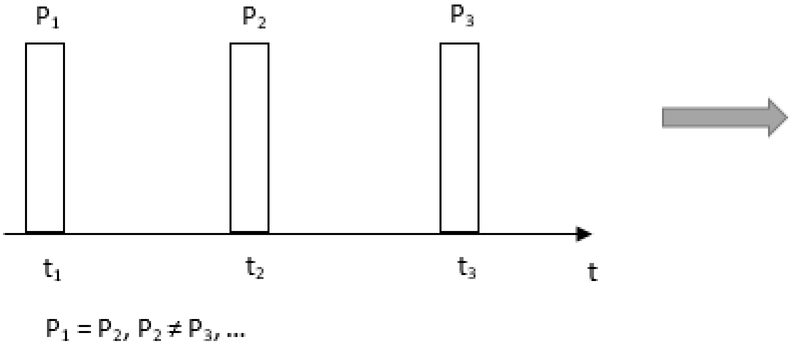


Figure 3.6: Data generation at the source node.

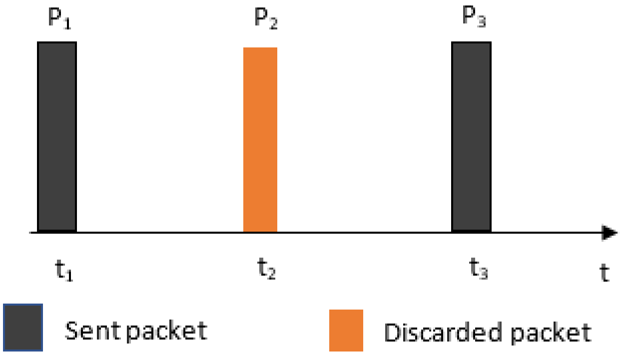


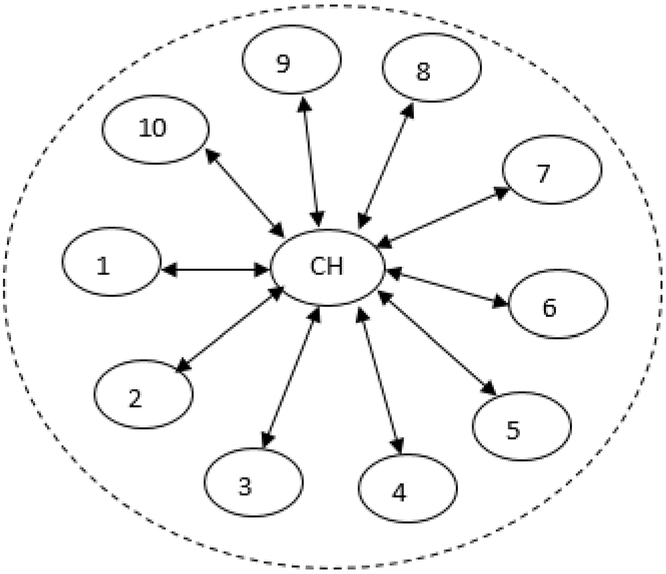
Figure 3.7: Illustrating sends/drops generated data based on data value principle.

The scheduling principle in RBS follows a simple process where the cluster-head uses historical data information at each source node. It scheduled each node to a specific timeslot and shared timeslot based on slot allocation available and node population. In this regard, the first generated data is sent to cluster-head but following generated data value is checked at the source node and if found to be the same, the data is dropped, otherwise, it is sent. In the case where it is dropped, the system should allow another node to transmit while former node goes to sleep. The collision may occur at the shared link and to avoid repeated collision for data re-transmission in next cycle within the same slot, CSMA-CA backoff algorithm is activated in the shared link. In the case where data is successfully transmitted on a dedicated link from a sender to receiver, an ACK is received by the sending node, and no delay is anticipated then. All other source nodes follow the same procedure and thus conserve their energy during sleeping moments and energy for transmission. More, packet transmission accounts for high energy consumption in WSNs, thus applying relevance technique will allow network devices to save some amount of energy and in general, will increase the life cycle of the entire network.

3.2.5 An example of the RBS operation

In this subsection, we give an example of RBS application scenario in monitoring an environment to report temperature change. Maintaining a particular temperature and accurately reporting is essential in areas prone to high climate gradients for quality control. All values in this example are for illustration. The range scale must reflect characteristics of the change because it can be random. Thus, different or similar values of temperature could be reported by different devices at different periods. Based on the control logic of the application, the minimum and maximum values are an indication of acceptable temperature values and above will signify danger. In this example, we consider the following.

A number of nodes = 10, 8 timeslots, cluster network, and star topology.



0	1	2	3	4	5	6	7

Figure 3.8: Topology & 8 timeslots for the RBS scheme example.

Assuming temperature operating range are from 22-68 °C for the sensing nodes. According to RBS design, CH uses slot 0, allocates slots 1-6 to corresponding nodes for dedicated transmission, while the last slot is used for the remaining nodes for shared access. If node1 reports 25 °C at t_1 and t_2, first it will send, but in the second period, generated data value 25 °C is compared with initial stored value which is the same, then it will allow i.e. node2 with a different value to send (t = different arrival time). At t_3, it generates 29 °C and will be sent. As stated, values are random; however, the current data value is compared to the previous value before transmission. This same logic applies to all nodes as shown in Figure 3.9 below. Nodes within the last slot may experience a collision.

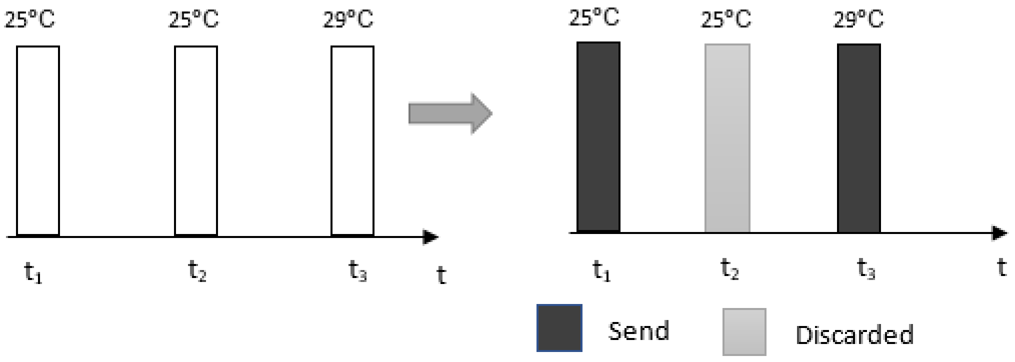


Figure 3.9: Illustrating an example of the RBS scheme operation.

3.3 Aggregation-based Scheduling (ABS)

The proposed ABS scheme is designed for IoT networks. Recent advancement in technology has resulted in an explosive growth in WSNs extending its applications to different IoT monitoring scenarios. So, an efficient network resource management scheme is essential to maintaining a lifetime of the entire network. Hence it is appropriate to reduce the amount of data transmission from sensors to sink. In this regard, data aggregation is introduced to improve network lifetime, increase PDR and reduce communication cost. Data aggregation can either be in time domain where a single node periodically aggregates its sensed data or space domain where multiple nodes perform aggregation before sending to cluster-head. ABS scheme considers time domain aggregation.

3.3.1 The concept of ABS scheme

In ABS scheme, raw data sensed at the source node must be aggregated and send to the cluster-head. The aim is to minimize average energy consumption and increase PDR in IoT networks subject to energy constraints of sensor nodes. One way of reducing the energy consumption of IoT networks is to reduce redundant data transmission. Similarly, it is important to Checkmate packet holding or waiting time before it is delivered to the cluster-head to avoid introduction of extra packet delivery delay [56]. Thus, converting this holding time into aggregation will help to optimize network lifetime. At the MAC layer, intra-cluster aggregation achieves improved energy saving in the more accurate way and thus suits better for data aggregation. Temporal (performed at the source node) and spatial (performed at the relay node) [57] can occur in WSNs. Dynamic aggregation scheme has been found to improve network performance on all metrics [58].

Based on IEEE 802.15.4e features but not limited to these, data rate 250kbps, 2.4GHz industrial, science, and medical (ISM) band, [10] and the maximum data length of 128 bytes in wireless LLNs, data aggregation can be achieved per timeslot in TSCH networks. TSCH MAC mode basis is to achieve reliable, scalable and flexible networks [11] and thus, high PDR could be traded for additional packet delivery delay. Therefore, we also apply temporal aggregation on active nodes based on *10ms per slot* time available for a node to send a packet and receive an ACK. In this regard, it follows that a node can send a packet and receive an anticipated ACK with an approximation of *4ms*. Thus, a slot can accommodate transmitting an aggregated data in IEEE 802.15.4e TSCH networks if carefully implemented. Further, regarding packet size of 20 bytes, merging data is applicable in ABS scheme where small data occurrence is synonymous to sensing activity of the network devices.

3.3.2 Designing ABS in IoT networks

In the energy-constrained IoT network environments, it will not be suitable for nodes to perform data generation and transmission to sink continuously due to battery power issue. As transmission consumes a lot of energy, thus transmitting data separately in each node per data generation will consume bandwidth and deplete energy faster in the whole network. In general, data aggregation integrates multiple bundles of sensed information into a single packet which is useful in achieving ABS strategy and in extension, leverage IoT networks. Therefore, as ABS scheme is applied at the leaf node in the time domain, it will remove enormous redundant information accrued to the sensed data and save energy.

In comparison, the performance of data aggregation is related to the network topology, and thus, star topology also forms network topology in the cluster as represented in Fig. 8. In clustering and data aggregation, it is inefficient for leaf node to transmit directly to sink node so, its alternative is to send to the cluster-head which is one-hop away. However, in the process of data aggregation, initially arrived data must wait for another data before data transfer process will take effect and will likely increase average latency of the network. It leads to less data transmission but with the trade-off of introduced delay because of the aggregation function. It may also depend on the slot configuration, a number of hops between the transmitting and receiving nodes. In IoT networks, while certain applications require the successful transfer of data between communication nodes, other applications may need more guarantees. In this regard, scheme evaluation metrics used to determine network lifetime should be application dependent. Thus, it is expected that in ABS, data is sensed, stored, aggregated, and forwarded to the cluster-head in the one-hop link.

Additionally, smart sensing devices deployed for monitoring purposes in LLNs are designed to collect small datasets. These small data tell about location, temperature, pressure, etc., in real time and part, provide a historical view. For example, smart sensors deployed in an environment to monitor temperature, vibration, and other desired attributes are automatically designed to work with the changing condition and generate instant information by small data. Thus, to report the accurate and current state of events, small data tend to provide a solution in LLNs. In this regard, small datasets are well suited for IoT TSCH application scenarios.

3.3.3 Timeslot allocation and data generation in ABS scheme

The ABS slot frame allocation follows a similar design as explained in subsection 3.2.3 where cluster-head uses the first slot for EBs/link scheduling, last slot for shared link, and other slots are for dedicated nodes. Network information is forwarded to all node to know when to transmit, receive, or sleep. Specifically, in ABS, data generation and transmission follow the principle as shown in the Figures 3.10/3.11 below.

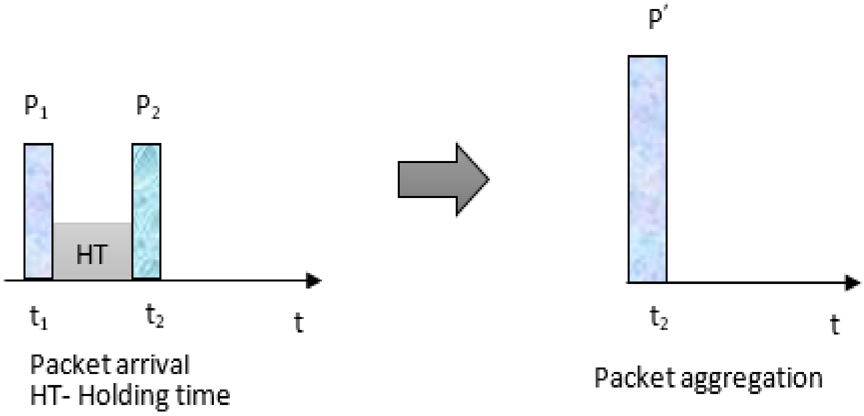


Figure 3.10: Illustration of data aggregation principle.

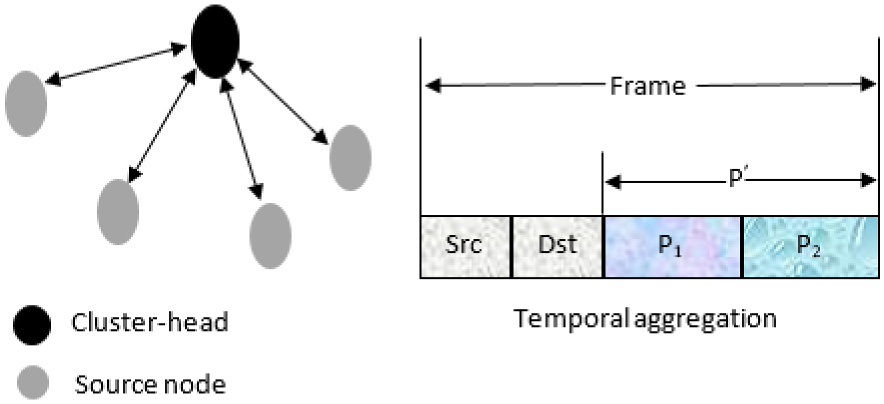


Figure 3.11: Integration of multiple data in a single frame at the source node.

In ABS, it is anticipated that every generated data at the source node is kept in the buffer prior transmission. The initially generated data is counted. Then, upon arrival of another data, both data are aggregated inside a single packet payload before the packet is transmitted to the cluster-head. In this regard, there is an introduction of holding time which accounts for the delay in the arrival of another data. We consider data generation to be normally distributed.

Aggregation at the source node combines multiple data it sensed into a single frame with the same source ID. In so doing, the total generated data at a single node will be halved. In WSNs, more energy is consumed during packet transmission, therefore aggregating multiple data at the same transmission time will help to improve the lifetime of the entire network. Consequently, the cluster-head schedule other network devices according to data aggregation information. In the shared link, less collision is expected to occur as more data are integrated, however, to avoid repeated collision, CSMA-CA backoff algorithm is activated upon collision. Collision occurrence also depends on node population and slot frame construction (i.e., the number of the timeslot in slot frame).

3.3.4 An example of the ABS operation

In this subsection, an example of an ABS application in an environmental monitoring for temperature gradient is briefly discussed. The procedures and parameter specifications are the same as in subsection 3.2.5. Considering the range of temperature in this example, 22-68 °C, applying ABS scheme will result in high data delivery and minimize the number of time an active node transmits to the cluster-head as shown in Figure 3.12.

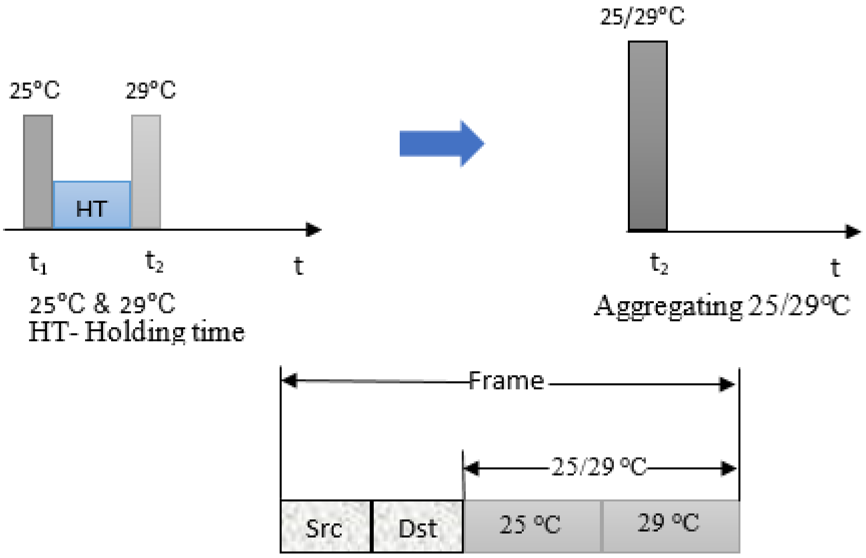


Figure 3.12: Illustration of an example of the ABS operation.

Observe the initially sensed data with 25 °C value must wait in the buffer at t_1, then at $t_{2,}$ data with 29 °C value arrives, and henceforth, aggregation takes place by applying ABS logic. The aggregated data are placed into a single packet payload for onward transmission to cluster-head. This process is applied to all source nodes in the time domain.

3.4 Chapter Summary

This chapter discusses the techniques in designing the envisaged schemes. Initially, TSCH standard approach is explored. Furthermore, following its basic principles, RBS and ABS are considered with an explanation on the concept, design, historical data value and data aggregation.

Chapter 4

Implementation of The RBS and ABS Schemes

This chapter discusses the implementation of RBS and ABS schemes to ensure the realization of technical standards and support IoT application scenario. Thus, it follows formulation of original design ideas into executable computer programs capable of imitating and generating all network activities. Besides, to automate a sequence of instructions with the aim of solving the design task of achieving performance metrics. Correspondingly, flow chart logic and code excerpts from ns-3 implementation codes are included. Design and implementation of MAC scheme must go through series of research cycle before possible deployment.

4.1 Selection of Simulation Tool

Simulation is a powerful tool to aid understanding of complex systems and support decision making. In general, it mimics the physical or abstract behaviour of a system in the view using computer software. Consequently, it is easy to modify network scenarios, make assumptions about traffic situations, topologies, etc., to obtain optimal data output. It has been categorized into discrete-event, continuous and Monte Carlo [59]. These categories differ in their mode of operation. DES employs logics/mathematical model to track changes in a system at a particular simulation time (e.g., customers waiting for service, military combat, etc.). In the continuous simulation, equations are used to describe or model system that changes its state continuously. While Mote Carlo uses the model of uncertainty that needs no representation of the time domain. Its operation is primarily on repetitive trials where numerical relationship results are obtained from one or many input variables provided to the system.

In WSNs, sensed data are kept in buffer for onward transmission to the destination when the opportunity arises. It is synonymous to the customer waiting for service, an example of DES. In this thesis work, DES is a preferred choice as the performance evaluation tool. Several DES simulation tools are in existence for carrying out an assessment in WSNs. These are, but not limited to the following: based on C++, network simulators; ns-2 [60], ns-3 [61], OMNeT++ [62], TOSSIM based on TinyOS [63], etcetera. The introduction of ns-3 has made an impact in performance evaluation. It is not an extension of ns-2 as it is developed from scratch. There are several interesting features (i.e., scalability, cross-layer, real world integration, visualization, several built-in libraries) that make it a preference in project building, implementation, simulation or redesigning new schemes to test efficiency and power consumption. The thesis work considers implementing in ns-3 since it involves building and redesigning MAC schemes to suit nodes with a specific mode of operation. Also, many thanks to MAC enhancement in IEEE 802.15.4e with different MAC modes.

RBS and ABS schemes are targeted to IoT TSCH applications. In general, through an extensive survey of IoT TSCH networks, several implementation trials require modification of procedures and scheme approaches. These adjustments are needed to support an effective implementation while maintaining the integrity of design goal. Thus, we observe the tests encountered as checks and balances to guide towards achieving usefulness of the planned schemes.

4.2 Implementation of The RBS Scheme

This section explains implementation and execution of RBS scheme specifications. In order words, preliminary thinking/idea generation is translated into the logical flow of actions representing network activities. Ubuntu 16.04.2 machine is used as an operating system. The activities of network devices are mimicked and translated into computer programme. Scheduling of these events are used to specify the logical order, duration, and other device attributes in a precise term. Accordingly, the schedule is performed by cluster-head.

4.2.1 Selection of the RBS implementation parameters

ns-3 has frameworks for real application and integrating tools making it a desired tool for MAC scheme design and evaluation preference. A general structure of implementation of the schemes is illustrated in Appendix A. RBS scheme is implemented under MAC module (e.g., lr-wpan-tsch module) with all interconnecting files to other modules written in C++ programme language. Lr-wpan-tsch includes configurable parameters to choose which MAC mode to enable in ns-3. We set the features of TSCH MAC mode by enabling it (i.e., set to true). Then all TSCH attributes are activated. For example, TSCH does not use a super frame, but slot frame is containing several timeslots that repeat accordingly. RBS full function scenario is built and enabled. The source code is in ns-3-lr-wpan-tsch/scratch/scenario.cc and consists of 2-20 nodes, packet size of 20 bytes, etc. The RBS implementation codes are included in Appendix B.

RBS flow chart represents workflow of IoT TSCH network activities based on historical data value. It shows control of how data flow through the network and thus, helps to understand network process as shown in Figure 4.1. Accordingly, the cluster-head builds schedule by using network information to run RBS procedure.

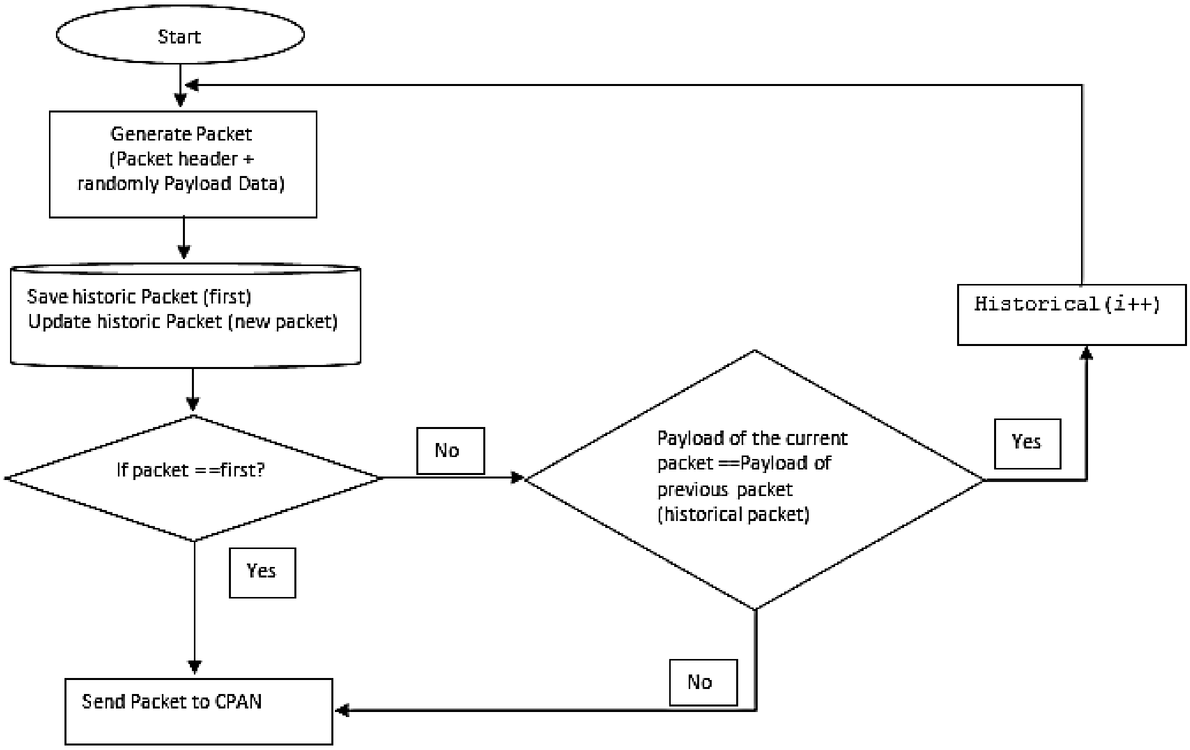


Figure 4.1: Illustration of RBS scheme procedure.

The flowchart depicts how RBS scheme works, when the source node generates initial data, its value is stored in the buffer before sending to cluster-head. As the same node generates more data, the content of the current data must be compared to the previously stored data value in the buffer. If the value of the current data is the same as the historical value, then current data is dropped, and historical value is incremented giving other active node chance to send its generated data. Reception of an ACK by the sending node signifies successful packet transmission. The idea here is to save network constrained resources and improve its entire lifetime. Since historical value is configurable, we can vary the range of minimum and maximum value to observe likely effect on the network with regards to other network parameters.

4.2.2 Timeslot configuration

The implementation RBS scheme follows the fixed timeslot version of slot frame configuration. Both TDMA and CSMA-CA are considered. To facilitate RBS scheme, the frame-based mechanism is used where a cell is a unit of allocation and consist of three type of slots: control slot for EBs, a dedicated slot for TDMA, and a shared slot for CSMA-CA. The coordinator or cluster-head (we use coordinator or cluster-head interchangeably) uses the first timeslot while the last slot is reserved for shared link and other slots are dedicated as shown in the code excerpt for timeslot configuration from the ns-3 implementation codes below. In dedicated link, particular node, for example, node1 communicate with the cluster head in a cell as in Figure 4.2; thus, both nodes have all the required network information attach to that cell. While in the shared link, several nodes communicate with the cluster head within a single cell. For transmission, network follows multipoint-to-point in star topology formation. As a fundamental aspect of RBS, the control packet (EB) is responsible for communicating network information.

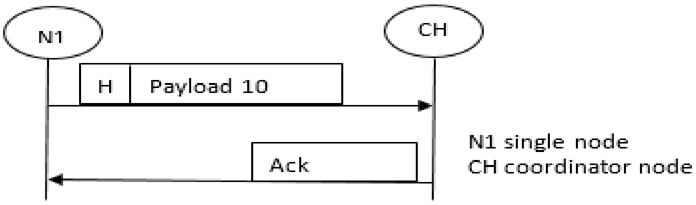


Figure 4.2: Packet transmission between node and coordinator.

Packets from source nodes are delivered to the master node with regards to the data value in the payload. All nodes are within transmission range in a small cluster of the network. As nodes in RBS are configured to different slots, device activities of interest for implementation including but not limited to the following.

- Nodes need to generate data on a periodic basis, and the value of generated data is reserved.

- First, generated data is attached to packet payload for transmission.

- The current data value is compared to previously stored value if it satisfies set condition, data is sent.

- Otherwise, data is dropped, and counter increases historical value.

- ACK is received for successful transmission.

- Collision can occur at CSMA-CA, in this case, a backoff algorithm is enabled.

TDMA implementation has no loss as packets are delivered without delay and nodes correctly synchronized to the cluster-head. All the active nodes know exactly when to perform a function. We assume that the network is not saturated so, nodes periodically generate packets to send but with regards to the content. However, in the case where a node has no packet to send, through network formation policy, cluster-head has topology information and assign its slot to another node.

Let us assume a number of slots (N_s) to be 8 ($N_s = 8$), and a number of active nodes (N_n) are the same then, the packet delivery ratio of the network is expected to be optimal in the dedicated slots as there is no collision expectancy. When the population of the active nodes (N_n) is more than N_s, the rest nodes are configured to the shared link (e.g., $N_n = 10$) and may lead to a collision. Figure 4.3 below shows slot frame with 8 timeslots, slots cycles, and ASN, which increases globally.

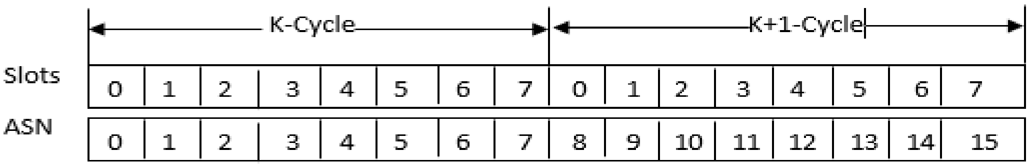


Figure 4.3: Illustration of fixed timeslot and ASN values.

```
//-------------------------AddAdvLink for beacon-----------------------------------------
        AddLinkParams alparams;
        alparams.slotframeHandle = 0;
        alparams.channelOffset = 0;
        alparams.linkHandle = 0;
        alparams.timeslot = 0;
        lrWpanHelper.AddAdvLink(netdev,0, alparams);
        //-------------------------AddLink-----------------------------------------

        uint32_t ASN_TimeSlot = 0;
        for (uint32_t i = 1; i < netdev.GetN() ; i++)
        {
                ASN_TimeSlot++;
                if (ASN_TimeSlot%timeslotnum == 0)
                {
                        ASN_TimeSlot++;
                        alparams.linkHandle = ASN_TimeSlot ;
                        alparams.timeslot = ASN_TimeSlot% timeslotnum ;
                        lrWpanHelper.AddLink(netdev,i,0,alparams,true);
                }
                else if((ASN_TimeSlot+1) % timeslotnum == 0)
                {
                        alparams.linkHandle = ASN_TimeSlot;
                        alparams.timeslot = ASN_TimeSlot% timeslotnum ;
                        lrWpanHelper.AddLink(netdev,i,0,alparams,true);
                        ASN_TimeSlot++;
                }
                else
                {
                        alparams.linkHandle = ASN_TimeSlot;
                        alparams.timeslot = ASN_TimeSlot% timeslotnum ;
                        lrWpanHelper.AddLink(netdev,i,0,alparams,true) ;
                }

DynamicCast<LrWpanTschNetDevice>(netdev.Get(i))->GetCsmaCa()->SetMacMinBE(0);
DynamicCast<LrWpanTschNetDevice>(netdev.Get(i))->GetCsmaCa()->SetSlottedCsmaCa()
;
        }
```

ns-3 code excerpt for timeslot configuration.

For example, considering 8 timeslots and 10 nodes then, after node 7, the addition of more nodes will require them to compete for access to the shared link (CSMA-CA timeslot). In this example, three (3) nodes will be scheduled to access the shared slot, six (6) are dedicated, and cluster-head uses the first slot.

4.2.3 Specification of relevance value

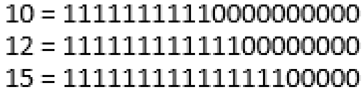
Every generated data contained an assigned value but represented in binary format as shown in Figure 4.4 below. The value is used to determine link allocation to active nodes in RBS scheme. So, an active node communication is dependent on it internal content. With this, nodes communicate only according to schedule time, so they keep their radios off to save energy.

Header	Payload (10-20)

Figure 4.4: Generated packet showing payload value.

An example of payload reading per packet in binary format representation for illustration purpose. Values range from 10 (min. value) to 20 (max. value) but can be adjusted;

For values;



10 = 11111111110000000000
12 = 11111111111100000000
15 = 11111111111111100000

Translating the historical data value is vital in RBS. So, the adopted binary code attribute is used to represent a historical data value. For example, if the value is 10 which corresponds to the length of its equivalent 1s. So, considering the set range of historical value, the system initializes index to 0 and checks if it satisfies the implementation condition. In this case, (m = 10), then it stores 1 in the buffer if the condition is true, otherwise 0, meaning historical value is greater than set condition as shown in code excerpt below.

```
uint32_t rn = x->GetInteger(min_PacketDegree,max_PacketDegree);
                    for (uint32_t index = 0; index < (uint32_t) pktsize ; index++)
                    {
                            if (index < rn)
                                    buffer[index] = 1;
                            else
                                    buffer[index] = 0;
                    }
                    Ptr<Packet> packet= Create<Packet> (buffer,pktsize);
```

Implementing historical value range code excerpt.

So, the cluster-head node uses this information to perform scheduling as highlighted in the flow chart.

4.3 Implementation of The ABS Scheme

Implementation of ABS scheme follows our basic implementation specifications; topology, a number of nodes, and timeslot specification with TDMA and CSMA-CA. Similarly, fixed timeslot is considered with 8 timeslots as explained in section 4.2 but with a difference on schemes logic of operations. Thus, we demonstrate aggregation techniques subsequently.

4.3.1 The ABS implementation flow chart

In ABS flowchart, the sequence of an event representing network scheduling activities based on data aggregation are highlighted. Coordinator node performs aggregation scheduling to other active nodes in the cluster and assigns slot allocation with regards to data information as shown in Figure 4.5 below. Consequently, if we have considered a total number of 200 packets per node and all packets are transmitted, then, the number of the successfully delivered packet will be 100 (i.e., half the total packet per node) with regards to aggregation.

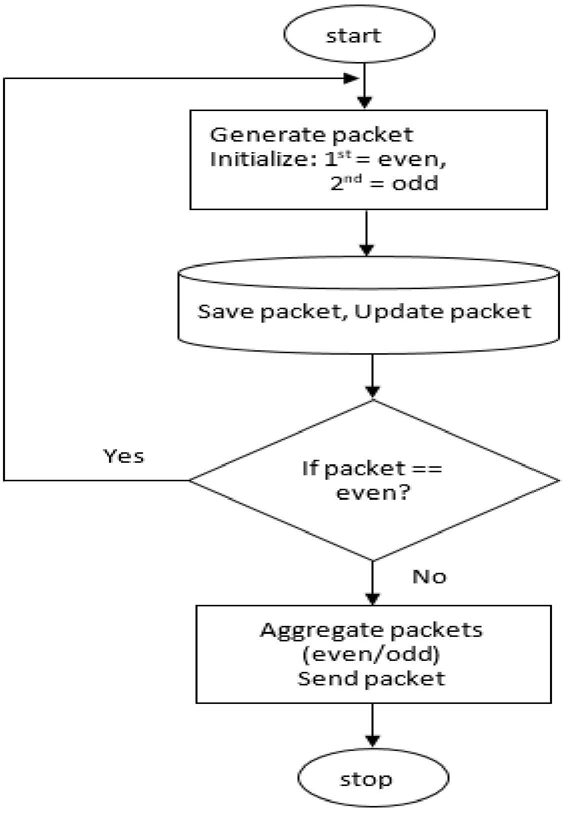


Figure 4.5: ABS flow chart.

ABS scheme precisely considers temporal aggregation in the time domain. This means that a source node upon generating first data puts it in the buffer and waits for second data. Thus, multiple generated data are transmitted in a single packet payload to the coordinator. Below excerpt codes show how this is implemented in ns-3.

4.3.2 Excerpt of the ABS implementation codes

```
if (m_aggregation)
        {
                if (m_counter % 2 == 0)
                {
                        memset(aggregation_buffer,0,sizeof(aggregation_buffer)) ;
                        m_lenght = packet->GetSize() ;
                        packet->CopyData(aggregation_buffer,m_lenght) ;
                }
                else{
                        uint8_t buffer[MAX_PAYLOAD_SIZE] ;
                        uint32_t packet_size = packet->GetSize() ;
                        packet->CopyData(buffer,packet_size) ;
                        for (uint32_t i = 0 ; i < packet_size ; i++)
                        {
                                aggregation_buffer[m_lenght+i] = buffer[i] ;
                        }
                        m_lenght += packet_size;
```

Implementing aggregation code excerpt

In this regard, an initially generated data is stored in the buffer as even data value while the next generated data is processed as odd. The system follows this simple principle and aggregation is performed upon encountering odd data as shown in code excerpt. More, in ABS scheme, packet value specification has a similar pattern in binary format. It is shown below,

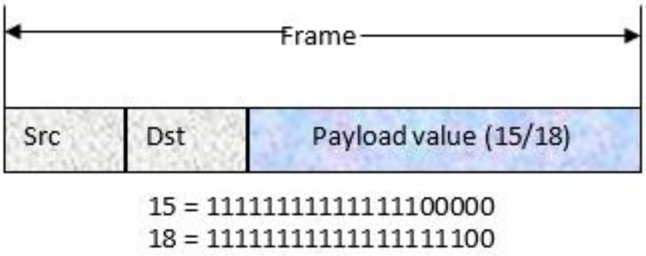


Figure 4.6: Aggregated data in the single packet payload.

ABS code excerpt illustrates the logic of how the generated data are aggregated. The system considers data generated in different sequences as even and odd. For instance, data occurring at (0, 2, 4, 6, …) intervals are considered even while (1, 3, 5, 7, …) are odd. So, aggregation is performed if copied data is odd. Then, aggregated data is wrapped in a single packet payload with same source ID and send to cluster-head. Cluster-head uses this information to schedule as depicted in the flow chart. It is anticipated that both RBS and ABS will improve IoT TSCH networks. In this respect, this thesis work provides designed schemes performance evaluation results in the next chapter.

```
{
    uint32_t size = packet->GetSize() ;
    uint8_t buffer[size] ;
    packet->CopyData(buffer,size) ;
    size = 0 ;
    for (uint32_t i = 0 ;i < packet->GetSize() ; i++)
    {
        if (buffer[i] == 1)
            size++;
    }
    if(size == historicData && !allsend)
    {
        flag = true;
        retval = true ;
    }
```

Aggregating even/odd value code excerpt

4.4 Chapter Summary

This chapter gives the overview of the implementation of RBS and ABS scheduling schemes using DES. Efforts toward achieving design fidelity are highlighted. Furthermore, logics representing RBS and ABS flows that captured IoT TSCH network scenarios including code excerpts are presented.

Chapter 5

Performance Evaluation of The RBS and ABS Schemes

In this chapter, we will analyse ns-3 simulations and their corresponding results. Sequel to an overall scheme design, and implementation, it must be analysed and compared to a reference model to substantiate its effectiveness or weakness. The reference model we used is IEEE 802.15.4e TSCH specification standard, and it is regarded as without scheduling throughout the result graphs.

The performance metrics considered in this thesis include PDR, packet collision probability, average delay, and energy consumption. Different result values are generated with the set parameters. Then, we modify the codes by altering desired parameter(s) while leaving others constant to verify the impact on generated results. The average result values of the reference model, RBS, and ABS schemes, from ns-3 implementation in different scenarios, are imported to MATLAB environment for graphical representations and comparisons.

5.1 Performance Evaluation Methods

In brief, there are three stages through which the theoretical analysis of MAC schemes can be evaluated for validity before deployment in the real-world applications. These may include; analytical model, computer simulation or experimental measurements [64]. In order word, these are done to validate concepts and then develop a suitable structure capable of performing network communication procedures.

- The analytical/mathematical model describes a collection measured and calculated behaviours using mathematical theory over the period.

- Experimental measurement based on testbeds is typically an experimentation on real sensor nodes with various components, (e.g., hardware, software, and networking set up) regarding specific application scenario. Thus, most glitches and failures that could happen in real-world deployments could be possibly noticed and corrected. Thereby, authenticating a theoretical and simulative estimation outcome.

Recently, there has been an increase in the development of testbeds to evaluating specific research works ranging from small-, medium-, or large-scale in WSNs setting [65]. Some examples include: MoteLab [66], TWIST [67], WISEBED [68], Motescope with iCount

[69] etcetera. Most sensor nodes in WSNs are based on Tiny operating system (TinyOS) because of its flexibility in adhering to the constrained resources of the devices in LLNs. Its timeliness, fast execution and simple programming language in NesC place it above other operating systems (e.g., CONTIKI, MANTIS, LiteOS, SOS, and RETOS) [70]. It has been shown that experimental procedures are gaining high ground in wireless networking research than simulations [71]. However, limitations still bound in experimental measurements, thereby making simulation inevitable. We focus on computer-based simulation method of evaluation considering the scope of this thesis and time factor.

5.1.1 Performance metrics

In this subsection, we briefly explain with expressions the considered performance metrics for reference model, RBS and ABS schemes used in result analysis. The metrics are as follows.

- **PDR**: The PDR or reliability is the probability of successful reception of transmitted packets to the cluster-head, and the sending node receives an ACK. In order words, it is the ratio of between the number of a data packets correctly received by the cluster-head to the total number of a data packet transmitted, and an ACK is received. It measures the network reliability. It is given by

$$PDR = \frac{Received\ packets}{Transmitted\ packets} \qquad (5.1)$$

- **Packet Collision Probability**: This is a number of packets lost due to packet collision or buffer overflow and packets are dropped. The assumption is that the channel is error-free. It is simply given as,

$$Packet\ collision\ probalility = 1 - PDR \qquad (5.2)$$

- **Average Delay**: It is the average delay from when the packet transmission is started at the source node to when the coordinator successfully received it.

$$Average\ delay = \frac{Total\ delay\ time}{Transmitted\ packets}\ (ms) \qquad (5.3)$$

- **Total Energy of Nodes**: To indicate energy consumption, consideration is given activities performed by active nodes in data transmission and reception of an ACK. A single slot consists of; $S_{TxDataRxACK.}$ Thus, energy consumed during active periods is more in the data packet communication from source nodes to coordinator node.

$$Total\ energy = \sum_{i=1}^{m} (P_t\ T_t(i) + P_r\ T_r(i)) \qquad (5.4)$$

Where m, P_t, T_t, P_r, and T_r are a number of packets, transmission power, transmission time, reception power and reception time respectively, TxDataRxACK implies data transmission and ACK reception.

5.2 Network Topology and Tested Scenarios

To elaborate on the performance analysis of the designed schemes, we present the network topology, parameters, and different tested scenarios.

5.2.1 Network topology and configurations

For performance evaluation, we concentrate on one cluster from the general topology shown initially in Figure 3.4. All nodes in the network connect to the cluster-head. Every node has a devoted connection to the cluster-head as shown in Figure 5.1 with a star topology which is suitable for static networks. The following but not limited reasons abound in star topology in IoT networks; network performance is consistent, predictable and fast as data travel only one hop from source nodes to the cluster-head. Additionally, there is an isolation of faulty individual node.

Figure 5.1: Scenarios topology (star) [11].

The default configurations considered in this thesis are given in *Table 5.1.* These configurations are the same in both reference model, RBS, and ABS schemes in all scenarios unless specifically mentioned where we varied them.

Table 5.1: Network Default Parameters

Parameter	**Value**	**Unit**
Relevance value range	10-20	Unit-less
Transmission Power	0	dBm
Number of packets	200	-
Packet size	20	Bytes
Timeslots (fixed)	8	-
EBs slot	1	-
TDMA slots	6	-
CSMA-CA slot	1	-
Number of nodes	2-20	-
Data Rate	250	Kbps
Topology	Star	-

5.2.2 Tested scenarios

This subsection is consisting of a set of test cases under different testing conditions. The list is shown in Table 5.2. From scenarios 2 to 5, one parameter is changed, and the results are reported in Section 5.4.

Table 5.2: Tested Scenarios

Scenarios	**Test Condition**
Scenario 1	Default configurations
Scenario 2	Change of timeslot
Scenario 3	Different packet lengths (sizes)
Scenario 4	Change in number of packets
Scenario 5	Different relevance ranges

In the forthcoming scenarios, we present a performance analysis of different parameters and compare the results.

5.3 Performance Evaluation Under Scenario 1

In this section, we present the effect of the set parameters on reference model, RBS, and ABS schemes by evaluating their performance metrics.

PDR: Figure 5.2 illustrates the average PDR under the set configurations. The result shows direct transmission of packets from source nodes to cluster-head. The achieved PDR performs a 100% delivery across schemes initially. This trend is expected to the configured 8 timeslots with 6 TDMA and single CSMA-CA slots. Thus, all active nodes assume dedicated stance until the population of nodes is 7 since cluster-head uses the first slot for EBs. Within this range, there is no collision expectancy for the packet. With the subsequent increase in node population more than slot number then, multiple nodes will compete for transmission time in the shared slot. So, there is a collision in the shared slot shown by the slight decrease in the result. The graph shows an apparent decline in more node population for all schemes because more packets are dropped due to packet collision.

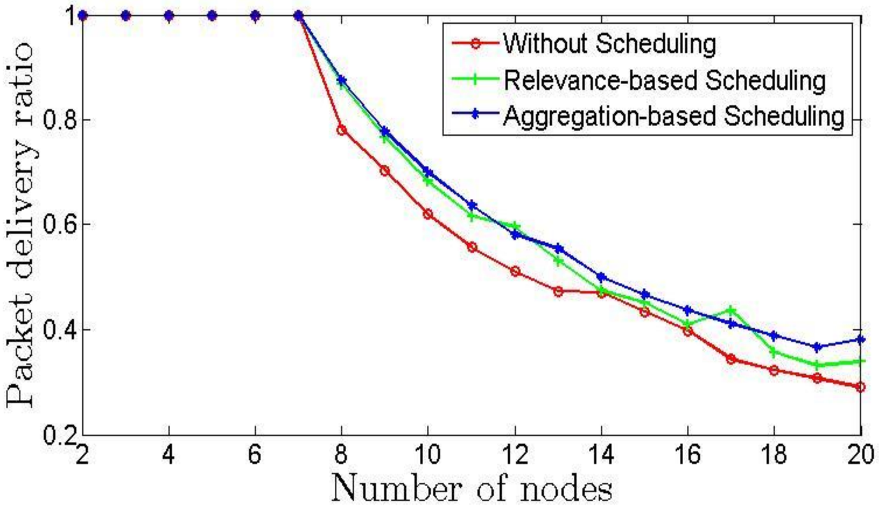


Figure 5.2: PDR for default set parameters.

Considering individual schemes, ABS and RBS outperform the reference model. As expected, ABS performs slightly better than RBS. It is following the definition of PDR. In ABS, more packets are delivered from aggregation procedure. More, the trend in RBS can be understood because of the randomness of the historical data value. The result also provides the insight about the observed decrease with increase in node population. The explanation is that node's ability to send a packet in the shared slot is affected by the amount of competition it experiences. Thus, this will be general behaviour in the forthcoming results.

Packet Collision Probability: Figure 5.3 illustrates packet collision probability as it is anonymous to PDR. Therefore, with the set parameters, and a number of active nodes equals to 7, no packet is lost meaning no collision. It is important to note that when collision happens in TSCH networks, retransmission of the packet is deferred to another cycle of the

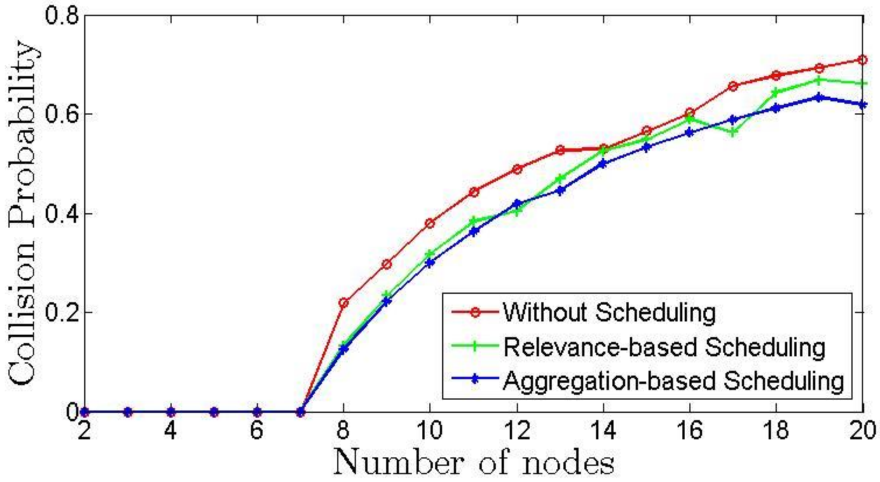


Figure 5.3: Packet collision probability for default set parameters.

transmitting/receiving devices within same assigned slot. It happens until maximum retransmission limit is attained. As noticed in the graph, collision starts building up as active nodes increase and resources are shared among competing nodes after a number of nodes > 7. The curves show the degree at which packets are collided and thus, dropped. Ideally, ABS and RBS perform better than the reference model. In ABS, aggregated packets account for less number of packet collision while in RBS, relevance values are considerations minimize redundant packets and thus, reduce packet collision.

Average Delay: Figure 5.4 shows variation the change of the average packet delay with node population = 20. TSCH MAC mode can tolerate negligible delay as it cares most for reliable and robust networks. Observe that the delay increases with increase in the number of nodes. It is because a network with small population experiences less traffic load, and hence, less transmission. On the other hand, with an increase in the number of nodes, more traffics are generated in the network with an increase in a number of packets to transmit to the coordinator. Buffer and mostly, transmission delays contribute significantly to the average delay.

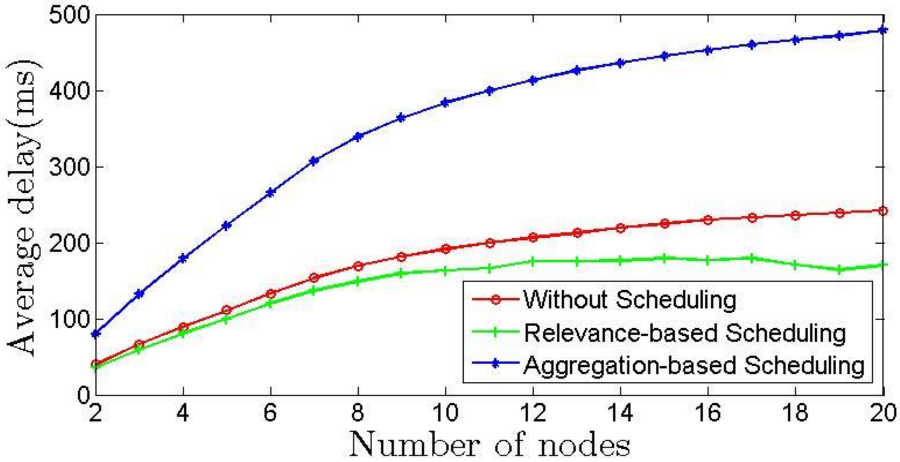


Figure 5.4: Packet average delay for default set parameters.

As anticipated, the performance of RBS is better than reference and ABS. The reason is that in RBS, redundant packets are dropped. As nodes in the cluster sensed the same data value, to reduce traffic congestion, these packets with the same value are dropped and in total ease packet transmission which is dominant and improve network communication. While in ABS, the high delay is expected because improved PDR is traded for the reasonable delay. There is an introduction of holding time as packet waits for an additional packet before aggregation function is implemented. So, buffer delay is significantly high in ABS as shown in the result.

Total Energy of Nodes: Figure 5.5 indicates average energy consumption of reference model, RBS, and ABS. In TSCH networks, a device can either transmit, receive, or keep it radio off signifying the activities of a slot. Coordinator maintains communication on topology and traffic requirements. These summed up the energy consumption of a node and in extension, the entire network. It explains the trend of energy consumption as shown in the graph. Observed that all schemes' energy consumption increase as the number of nodes increase. The reason is that more network activities take place in the slots and thus more devices are active either transmitting generated data packet. Additionally, when packets collide, significant energy is spent. The high number of collisions represent a waste of energy, and packets are dropped which explains the curves shown in the result.

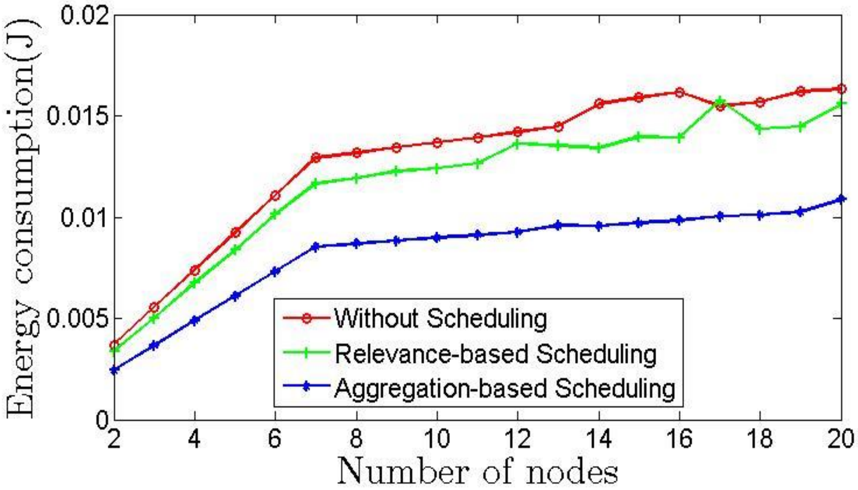


Figure 5.5: The total energy of nodes for default parameters.

Observed that both ABS and RBS schemes perform better than the reference model. Also, ABS performance is better than RBS. It is as expected because in ABS more data are integrated before transmission. In WSNs, transmission depletes more energy than other activities. Thu, considering the formulated expression of energy in 2.1.1 and the total number of packets (200), m = 100 in ABS due to aggregation function. Therefore, fewer packets are transmitted to the coordinator. Furthermore, in RBS because of historical value, redundant packets are dropped, and fewer data packets traffic occur. In this case, depending on the frequency of relevance value, m < 200, (e.g., 180).

5.4 Numerical Results for Additional Scenarios

In this section, we present the performance evaluation of reference model, RBS, and ABS schemes by altering some parameters while leaving others constant. Theses parameters are timeslots, packet length, the number of packets, and relevance range. All configurable parameters are used to evaluate performance metrics. In all scenarios, a number of nodes are fixed to 10 except if mentioned.

5.4.1 Performance evaluation under scenario 2

In this subsection, we vary a number of timeslots from 7-11 and evaluate its impact.

PDR: Figure 5.6 illustrates the effect of changing timeslots on the performance metrics. Observe that PDR gradually increase with a corresponding increase in timeslots for all schemes. It is expected considering the number of nodes (10). There is competition for access to the shared slot and packets are dropped regarding our timeslot configuration. So, with an increase in slots, more nodes get fair access until timeslot is increased to 11. At this point, all nodes have dedicated slot for both TDMA and CSMA-CA (e.g., the first slot for EBs, Last slot for CSMA-CA and 9 dedicated slots). So, the achieved PDR at this instance is 100% with no collision of packets. ABS and RBS perform better than reference model as observed. The reason follows as explained earlier.

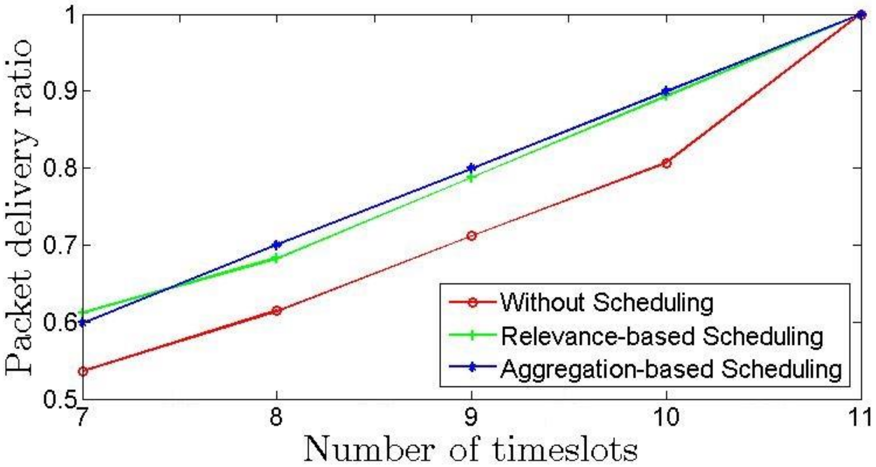


Figure 5.6: Effect of change of timeslot on PDR.

Average Delay: From Figure 5.7, observe the trend of all the schemes. There is a general increase in delay as timeslot increases. As explained earlier, the activities of the nodes constitute the average delay. More, in TSCH, consideration is given to the number of slots in a slot frame which repeats accordingly. So, a higher number of slots introduce additional delay as seen from the figure. ABS has the highest delay because of aggregation function while RBS performance is as expected and performs better than the reference model.

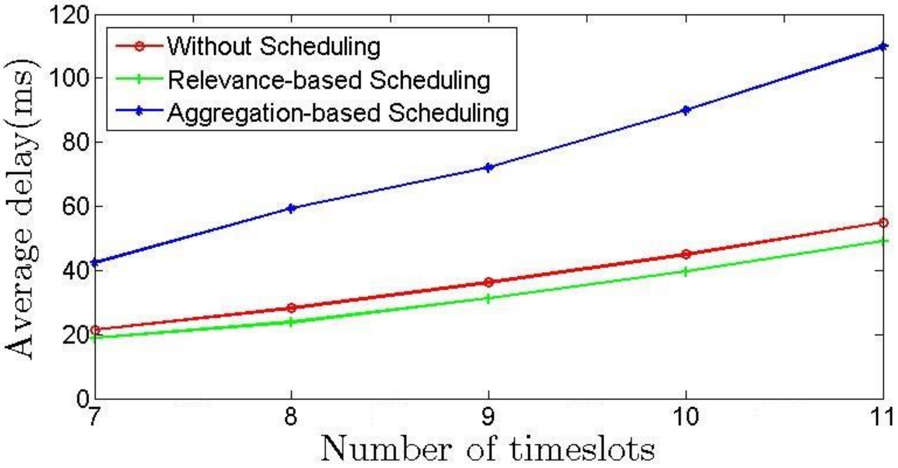


Figure 5.7: Effect of change of timeslot on average packet delay.

Total Energy of Nodes: In Figure 5.8, the impact of the change of timeslot is observed on energy consumption of nodes indicating activities performed in each slot. As explained initially, ABS and RBS consume less energy since a total number of transmitted packets are reduced by aggregation function/historical values respectively, thus minimizing the energy consumed during packet transmission than the reference model. As seen from the result, it increases with corresponding slot increase due to more activities per slot.

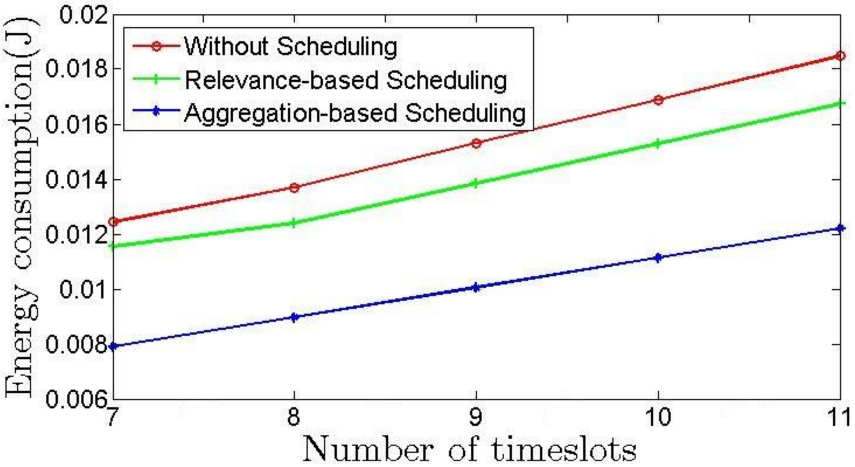


Figure 5.8: Effect of change of timeslot on the total energy of nodes.

5.4.2 Performance evaluation under scenario 3

In this subsection, we alter the size of the packet from 20-60 bytes and evaluate the impact on performance metrics of all the schemes.

PDR: To explain the effect of different packet sizes on PDR, we use both generated result values as shown in Table 5.3 and graph. As shown from Figure 5.9, ABS and RBS perform better than the reference model. The reason is that PDR is proportional to packet size, so more packets are aggregated in ABS resulting in high PDR. Also, relevance function accounts for an increase in more packets delivered to the cluster-head. As seen from the table, no value is recorded in ABS when packet size is 60 bytes, and the reason is that in WSNs, the maximum transfer unit (MTU) is 128 bytes for packet frame including header part. Therefore, aggregating multiple data sizes of 60 bytes in a single payload has exceeded payload size.

Table 5.3: Result Values for the Effect of Different Packet Sizes on PDR

Packet size	Without Scheduling	RBS	ABS
20 B	0.6145	0.682496	0.7
30 B	0.614	0.682496	0.7
40 B	0.614	0.682496	0.7
50 B	0.614	0.682496	0.698
60 B	0.614	0.682496	-

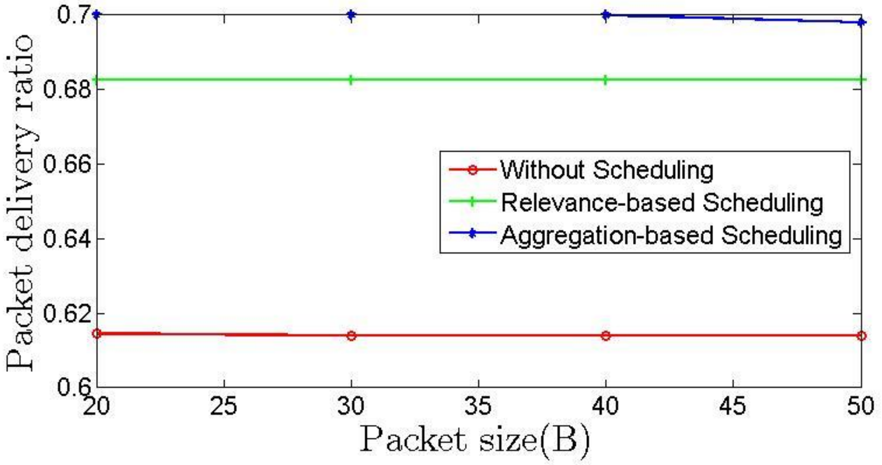


Figure 5.9: Effect of different packet sizes on PDR.

Average Delay: In explaining the effect of packet sizes on average packet delay, we also use both result value as shown in Table 5.4 and graph. As expected, ABS has more average packet delay, while RBS performs better than the reference model. As highlighted initially, the time it takes a packet to be transmitted is based on its size. It implies that smaller packet will get to the cluster-head faster. As shown in Figure 5.10, the delay experienced by all schemes increases as packet size increases. Observe from Table 5.4, the gradual increase in delay from 20-50 bytes. For example, consider packet size of 20 bytes and data rate of 250 kbps, then it takes approximately 0.64 *ms* to transmit, while it takes a packet of 40 bytes approximately 1.28 *ms* to transmit. No value is recorded in ABS when packet size is 60 bytes as explained. Schemes comparisons follow earlier explanations.

Table 5.4: Result Values for the Effect of Different Packet Sizes on Average Delay

Packet size	**Without Scheduling**	**RBS**	**ABS**
20 B	28.1904	23.8151	56.2914
30 B	28.5104	24.1351	56.6114
40 B	28.8604	24.4551	56.9614
50 B	29.2504	24.8451	57.3414
60 B	29.6104	25.1951	-

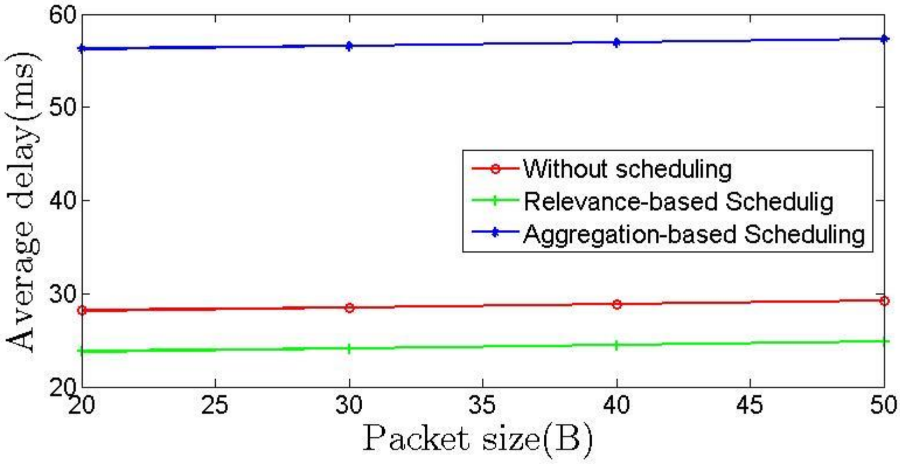


Figure 5.10: Effect of different packet sizes on average packet delay

Total Energy of Nodes: Figure 5.11 shows the effect of various packet sizes on energy consumption of nodes. As observed, energy consumption increases with continuous increase in packet sizes. It takes more time for a packet of larger size to be successfully received at the coordinator and subsequent ACK for the sending node hence, more energy is consumed. As seen, ABS and RBS perform better than the reference model.

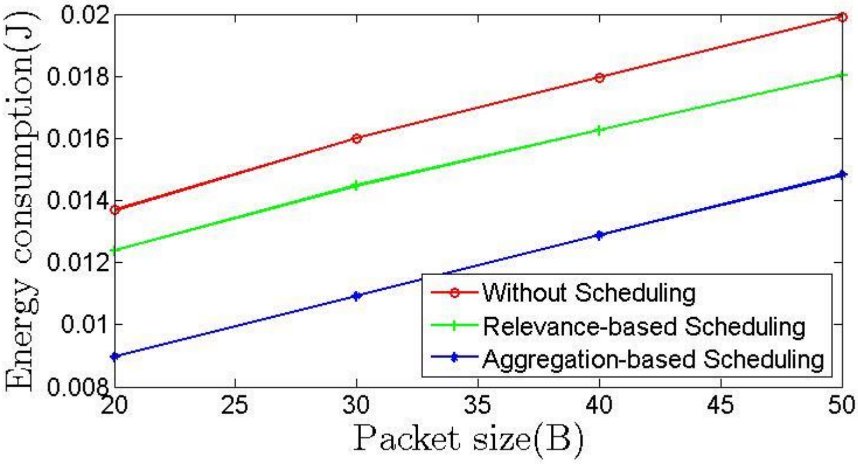


Figure 5.11: Effect of different packet sizes on the total energy of nodes.

5.4.3 Performance evaluation under scenario 4

In this subsection, we change a number of packets from 100-500 and evaluate the impact on performance metrics of all the schemes.

PDR: Figure 5.12 illustrates the effect of packet sizes on PDR. Observe that ABS and RBS perform higher than the reference model. Considering packet generation and transmission in an unsaturated network with uniform arrival rates. The total number of packets successfully transmitted to the coordinator in ABS per packet is counted while multiple packets are integrated and transmitted per slot both in TDMA and CSMA-CA. It reduces the possibility of packet collision. As the number of packet increases, an average of successfully received packets tends to be stable in all cases. More, the trend in RBS is due to the randomness of the historical packet values which also minimizes packet collision by allowing nodes with different packet value to send.

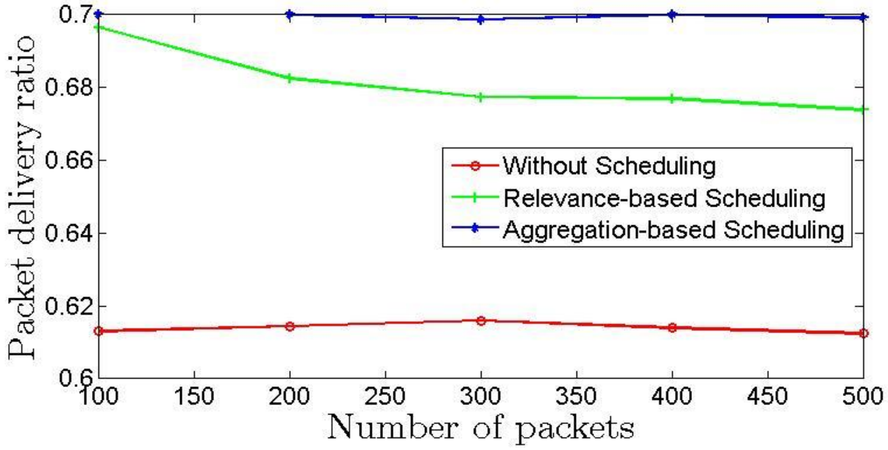


Figure 5.12: Effect of a number of packets on PDR.

Average Delay: The impact of a number of packets on average packet delay is illustrated in Figure 5.13. It is observed that average packet delay increases with subsequent increase in the number of packets. As explained initially, it is due to activities performed per slot. It is most noticed in ABS due to buffer and transmission delays regarding aggregation role. RBS has a least average delay since redundant packets are dropped removing some packet transmission delay which has more influence now.

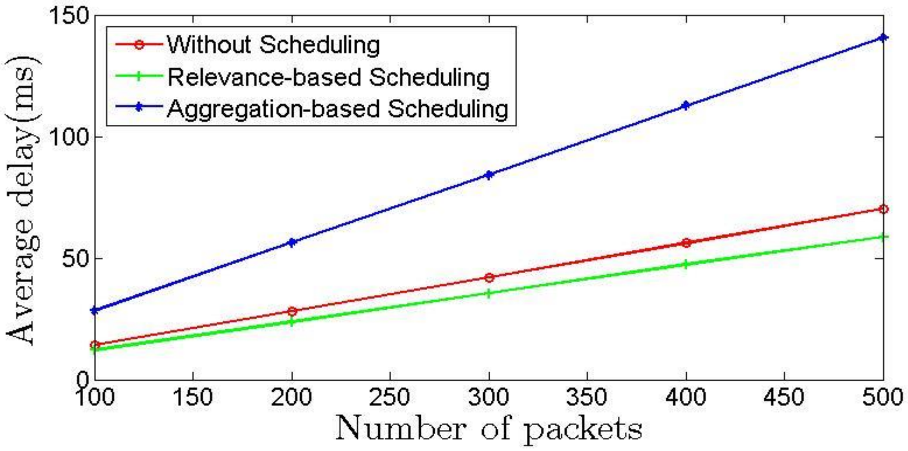


Figure 5.13: Effect of a number of packets on average packet delay.

Total Energy of Nodes: Figure 5.14 displays the variation of a number of packets on total energy consumption. The overall trend shows an increase in energy consumption for the tested schemes with more packets signalling various network activities. Observe that with 100 packets, and 10 nodes, there is less occurrence of packet collision and packet transmission to compare when there are 500 packets with the same number of nodes. As explained earlier, ABS and RBS perform better than the reference model.

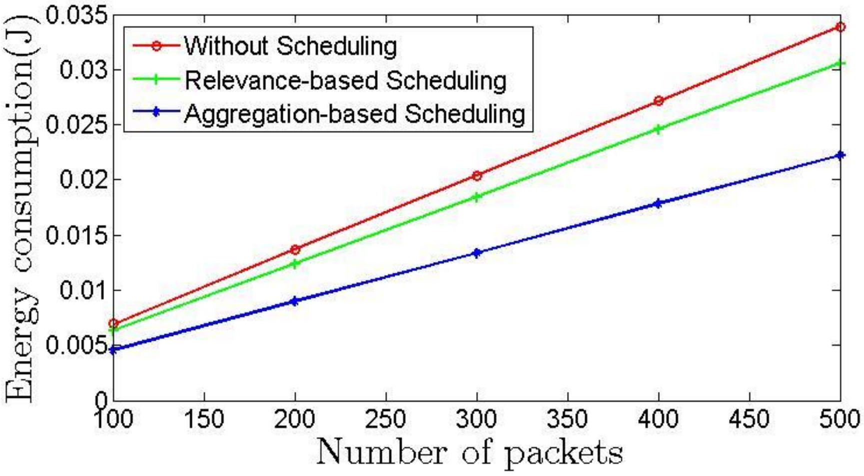


Figure 5.14: Effect of a number of packets on the total energy of nodes.

5.4.4 Performance evaluation under scenario 5

In this subsection, we evaluate the performance metrics of all the schemes based on relevance range (e.g., 10-15, 10-20, and 10-30). We use windows to represent different ranges (e.g., window1= 10-15, window2=10-20, window3=10-30), Other parameters are as default values.

PDR: The influence of different relevance ranges on PDR is shown in Figure 5.15 based on default configurations. Observe the variation of the windows after slot 7 as node population increases. As explained, consideration is given to TDMA and CSMA-CA. In this case, when the window size is small with a number of nodes = 20, there is less packet collision as there is few recurrence of similar values. It accounts for high PDR as seen in the result for window1. The trend applies to Windows 2 and 3 respectively. In this regard, window3 has least PDR in comparison.

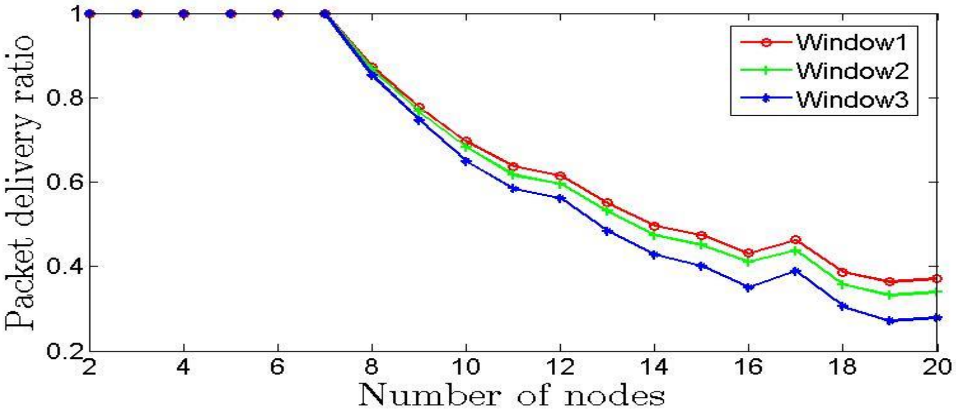


Figure 5.15: Effect of relevance ranges on PDR.

Average Packet Delay: Figure 5.16 illustrates the impact of different relevance ranges on average packet delay. Considering network activities, there is the general increase in average packet delay with nodes increases from 2-20 as shown. Specifically, window3 experiences higher delay as expected. The reason is that it contains more relevance values than other windows. Packets have high chances of picking different values and thus, more node activities regarding generation and transmission which constitute average delay.

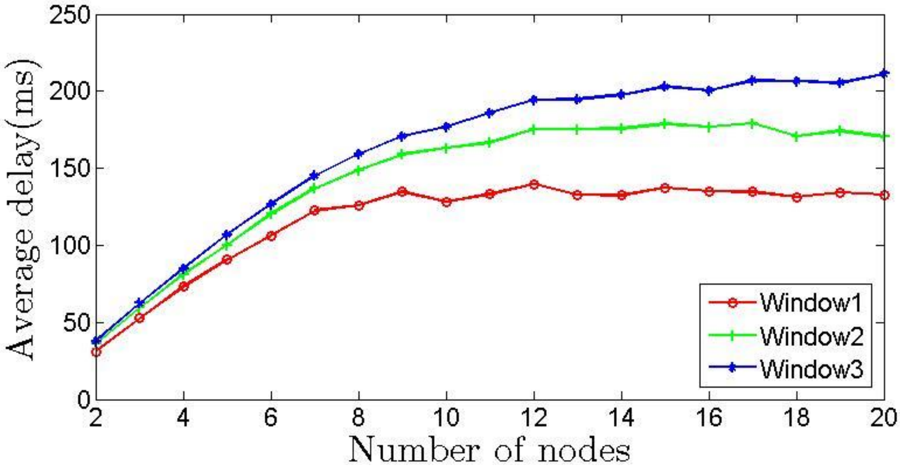


Figure 5.16: Effect of relevance ranges on average packet delay.

Total Energy of Nodes: The influence of relevance ranges on energy consumption of nodes is shown in Figure 5.17. Energy consumption of nodes is characterized by various activities performed in slots. As discussed in this thesis, it is observed that less energy is consumed in window1, followed by window2 while window3 has highest energy consumption as depicts by the result. With all windows having an equal number of nodes (20), windows3 has high chances of packet occurrence with most relevance range values. The same reason applies to Windows 1 and 2, respectively. The fluctuation observed is because of relevance value randomness.

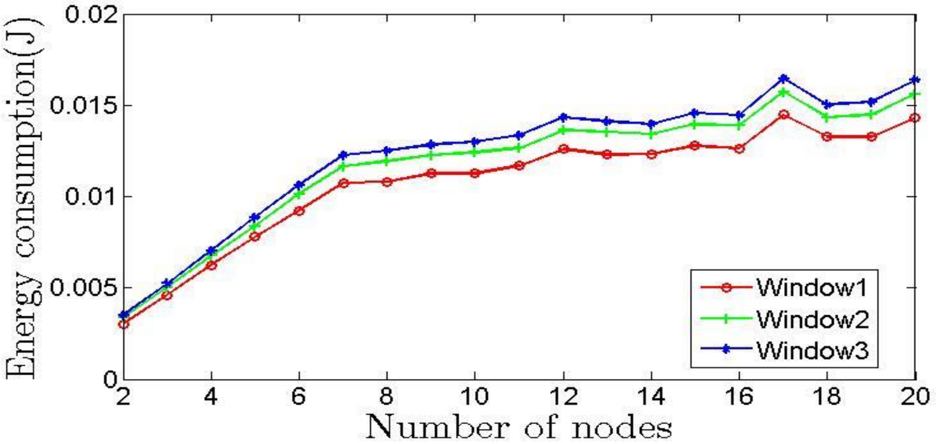


Figure 5.17: Effect of relevance ranges of energy consumption of nodes.

5.5 Chapter Summary

This chapter gives an outline of the performance evaluation methods with an adoption of computer simulation for our thesis work. Further, network topology, configuration, and different scenarios are presented. Correspondingly, performance evaluation of all the considered metrics under different test conditions are discussed with the numerical result. These numerical results under default and additional configurable parameters obtained from simulations show an improvement of the reference model and therefore match the theoretical design idea.

Chapter 6

Conclusions and Future work

In this chapter, we summarize the entire thesis work, present our contributions, and also identify potential directions to follow in future.

6.1 Conclusion

After surveying state-of-the-art on IEEE 802.15.4e MAC schemes, and design approaches to improve energy reliability, building/maintaining data communication, and robust networks needed for IoT applications in the literature, we observe that there is no standardized policy to build and maintain data transmission in IEEE 802.15.4e IoT networks. It is basically per application dependent. Additionally, activities perform per slot by network devices contribute to the exhaustion of its lifetime. To minimize energy consumption and at the same time increase data transmission, this thesis work has proposed IoT MAC schemes capable of employing hybrid slot assignment with insight into a historical data value, and temporal integration of data in time domain. Correspondingly, the proposed schemes can reduce energy consumption, increase PDR, and at the same time, adhere to the latency requirement of the network. More, discrete-event based simulation and MATLAB environment are used for implementation and graphical representation of results.

6.2 Contributions

Through various research on the state-of-the-art of relevant literature, on building an energy reliable and efficient data transmission maintenance in IoT networks. This thesis has gone further to propose two schemes in this regard. In brief, the contributions are as follows.

- To minimize energy consumption, increase data communication and robust IoT networks, RBS has been proposed, and its performance is compared to IEEE 802.15.4e TSCH standard scheme. RBS can be applied to a small cluster of static networks with a star topology. It leverages the advantages of TDMA and CSMA-CA slot assignment with emphasis on historical data value to minimize network resource communication cost. Additionally, through single hop communication from the source node to cluster-head, a lifetime of the network is improved.

- Packet aggregation is also one of the ways to minimize energy consumption of networks. The ABS has been proposed and compared with the IEEE 802.15.4e TSCH standard scheme. ABS uses both dedicated and shared slot allocation to enhance data communication where data are merged at the source node in time domain before transmitting to the cluster-head. Thus, multiple sensed data are transmitted in a single packet payload and on average, maintains a lifetime of the network. In ABS, there is a trade-off between high packet delivery to negligible introduced delay.

6.3 Future Work

Throughout this thesis work, we concentrated on IoT MAC layer for data transmission in WSNs. We have proposed two schemes, and the performance metrics have been examined and showed an improvement in the IEEE 802.15.4e TSCH standard. Outside the research objectives, there are still open issues within IoT MAC layer that can be imagined. Consequently, some of the potential topics we are considering are in the following directions.

- The thesis work has proposed schemes that can improve the lifetime of networks as well as increasing PDR with the consideration of small cluster of networks and star topology. The schemes could be developed and applied in large scale networks that could extend to tree or mesh topologies.
- We have implemented and evaluated metrics of the proposed schemes through computer simulations. For practical verification and assess to the proposed schemes, real-life experimentation based on testbeds could be carried out.
- Temporal data aggregation in the time domain has shown to improve lifetime and increase PDR of IoT networks. Considering MTUs for packets in WSNs and IPv6 respectively, the thesis could further work to reduce packet loss, improve response time, and decrease infrastructure cost by employing IPv6 header compression strategy for packet transmission in WSNs. Thus, packet aggregation from different nodes in space domain where the idea of label switching approach would also be considered.

References

[1] O. Vermesan, P. Friess, G. Woysch, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, M. Eisenhauer, and K. Moessner, "Europe's IoT Strategic Research Agenda," in *The Internet of Things-European Research Cluster on the Internet of Things (IERC)*, 3rd ed., Halifax, New Horizons, 2012, pp. 22-107.

[2] Lopez Research, "An Introductory to Internet of Things," *Part 1. of The IoT Series*, November 2013.

[3] O. Vermesan, P. Friess, G. Woysch, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, M. Eisenhauer, and K. Moessner, Internet of Things-From Research and Innovation to Market Deployment, Denmark: River Publishers Series in Communication, 2014, pp. 3-47.

[4] Web Site Directory, "Internet World Stats: Usage and Population Statistics," 31 March 2017. [Online]. Available: http://www.internetworldstats.com/stats.htm. [Accessed 8 April 2017].

[5] Allied Business Intelligence (ABI) Research, " Over 5 Billion Wireless Connectivity Chips will Ship in 2013," 21 Dec. 2012. [Online]. Available: https://www.abiresearch.com/press/over-5-billion-wireless-connectivity-chips-will-sh/. [Accessed 8 March 2017].

[6] S. C. Mukhopadhyay and N. K. Suryadevara, "Internet of Things: Challenges and Opportunities," in *Internet of Things*, vol. 9, Switzerland, Springer International Publishing, 2014, pp. 1-17.

[7] S. C. Mukhopadhyay, "Wearable Sensors for Human Activity Monitoring: A Review," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1321-1329, March 2015.

[8] O. Vermesan, P. Friess, P. Guillemin, R. Giaffreda, H. Grindvoll, M. Eisenhauer, K. Moessner, M. Spirito, and E. Z. Tragos, "Internet of Things Beyond the Hype: Research, Innovation and Deployment," in *Internet of Thing*, European Research Cluster on Internet of Things (IERC) SRIA, 2015, pp. 15-65.

[9] IEEE draft std. 802.15.4™, Part 15.4: Wireless Access Medium Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal area Networks (LR-WPAN), New York: IEEE Standards for Information Technology, Inc., 2003.

[10] IEEE 802.15, "IEEE 802.15 WPAN™ Task Group 4 (TG4)," IEEE Inc., 2010.

[11] IEEE draft std. 802.15.4e™, Part 15.4: Low-Rate Wireless Area Network (LR-WPANs) Amendment 1: MAC sublayer, New York: IEEE Standard for Information Technology, 2012.

[12] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Commun. Surveys & Tuts.*, vol. 15, no. 3, pp. 1389 - 1406, 2013.

[13] R. Soua, P. Minet, and E. Livolant, "MODESA: An Optimised Multichannel Slot Assignment for Raw Data Convergecast in Wireless Sensor Networks," in *Proc. 31st International Performance Comput. and Communs. Conference (IPCCC)*, 2012.

[14] R. Soua, E. Livolant, and P. Minet, “MUSIKA: A Multichannel Multi-sink Data Gathering in Wireless Sensor Networks,” in *Proc. 9th International Wireless Communs. and Mobile Comput. Conference (IWCMC)*, 2013.

[15] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, “Traffic-Aware Scheduling Algorithm for Reliable Low-power Multi-hop IEEE 802.15.4e Networks,” in *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Commun. - (PIMRC)*, 2012.

[16] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, “A Centralized Scheduling Algorithm for IEEE 802.15.4e TSCH based Industrial Low-power Wireless Networks,” in *IEEE Wireless Communs. and Net. Conference*, 2016.

[17] A. Morell, X. Vilajosana, J. L. Vicario, and T. Watteyne, “Label Switching Over IEEE802.15.4e Networks,” *Trans. on Emerg. Telecommun. Technol.*, vol. 24, no. 5, pp. 458-475, 2013.

[18] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler, “Decentralized Traffic Aware Scheduling for multi-hop Low power Lossy Networks in the Internet of Things,” in *Proc. 14th International Symposium on WoWMoM*, 2013.

[19] R. Soua, P. Minet, and E. Livolant, “DiSCA: A Distributed Scheduling for Convergecast in Multichannel Wireless Sensor Networks,” in *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015.

[20] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, “Fast Join and Synchronization Schema in the IEEE 802.15.4e MAC,” in *Proc. IEEE Wireless Commun. and Net. Conference Workshops (WCNCW)*, 2015.

[21] D. D. Guglielmo, S. Brienza, and G. Anastasi, “A Model-based Beacon Scheduling algorithm for IEEE 802.15.4e TSCH networks,” in *Proc. IEEE 17th International Symposium on WoWMoM*, 2016.

[22] I. Khoufi, P. Minet, E. Livolant, and B. Rmili, “Building an IEEE 802.15.4e TSCH network,” in *Proc. International Performance Comput. and Commun. Conference (IPCCC)*, 2016.

[23] D. Stanislowski, X. Vilajosana, Q. Wang, T. Watteyne, and K. Pister, “Adaptive Synchronization in IEEE802.15.4e Networks,” *IEEE Trans. on Industrial Informatics*, vol. 10, no. 1, pp. 795 - 802, 2014.

[24] T. Chang, T. Watteyne, K. Pister, and Q. Wang, “Adaptive Synchronization in Multi-hop TSCH Networks,” *International Journal of Comput. and Telecommun. Netw.*, vol. 76, no. C, pp. 165-176, 2015.

[25] IETF Standard, “RFC 6550- RPL: IPv6 Routing Protocol for Low-power and Lossy Networks,” IETF Trust, 2012.

[26] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. Pister, “A Realistic Energy Consumption Model for TSCH Networks,” *IEEE Sensors Journal*, vol. 14, no. 2, pp. 482 - 489, 2014.

[27] T. Watteyne, J. Weiss, L. Doherty, and J. Simon, “Industrial IEEE802.15.4e Networks: Performance and Trade-offs,” in *Proc. IEEE International Conference on Commun. (ICC),* 2015.

[28] Cisco DevNet Creations, “Cisco DevIoT-Version Alpha,” 4 January 2015. [Online]. Available: https://developer.cisco.com/site/devnetcreations/detail.gsp?submitId=171067. [Accessed 20 April 2017].

[29] Cisco White Paper, “The Internet of Everything and the Connected Athlete: This Challenges... Everything,” Cisco Public Information, 2013.

[30] Cisco White Paper, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021,” Cisco Public Information, 2016.

[31] A. Osseiran, V. Braun, T. Hidekazu, P. Marsch, H. Schotten, H. Tullberg, M. Uusitalo, and M. Schellman, “The Foundation of the Mobile and Wireless Communications System for 2020 and Beyond: Challenges, Enablers, and Technology Solutions,” in *Proc. IEEE 77th Vehicular Technology Conference (VTC Spring),* 2013.

[32] ACATECH National Academy of Science and Engineering, “Securing the Future of German Manufacturing Industry- Recommendations for Implementing The Strategic Initiative INDUSTRIE 4.0,” Industry-Science Research Alliance, Germany, 2013.

[33] Ericsson, “APPIOT: The Application Platform for Internet of Things,” Ericsson, 2017. [Online]. Available: https://www.ericsson.com/industries/iot-for-industry/appiot. [Accessed 24 April 2017].

[34] Telenor Group, “Internet of Things with Telenor,” 2008. [Online]. Available: https://www.telenor.com/innovation/machines/internet-of-things/internet-of-things-with-telenor/. [Accessed 24 April 2017].

[35] ELTEL, “Eltel's Power Distribution Wins New Close to 300,000 Smart Meter Rollout Contract with The Soria Consortium in Norway,” 22 December 2015. [Online]. Available: http://www.eltelgroup.com/en/eltels-power-distribution-wins-new-close-to-300-000-smart-meter-rollout-contracts-with-the-soria-consortium-in-norway/. [Accessed 24 April 2017].

[36] E. Savitz, “Forbes: Gartner- Top 10 Strategic Technology Trends For 2013.,” 23 Oct. 2012. [Online]. Available: https://www.forbes.com/sites/ericsavitz/2012/10/23/gartner-top-10-strategic-technology-trends-for-2013/#d3630fab7615. [Accessed 15 March 2017].

[37] J. R. Gartner, “Gartner Newsroom: Internet of Things Installed Base Will Grow to 26 Billion Units By 2020,” 12 Dec. 2013. [Online]. Available: http://www.gartner.com/newsroom/id/2636073. [Accessed 15 March 2017].

[38] D. Evans, “Internet of Things- How the Next Evolution of Internet is Changing Everything,” Cisco Internet Business Solution Group (IBSG), 2011.

[39] HART Communication Foundation STd., “WirelessHART Specification 75: TDMA Data-LINK, Rev. 1.1, HCF_SPEC-75,” Field Communication Group, Austin, 2008.

[40] ISA 100.11a Standards, “Wireless Systems for Industrial Automation: Process Control and Related Applications,” International Society of Automation, 2009.

[41] IEEE Standards Association, “GET 802® Standards- IEEE 802.15™: Wireless Personal Area Networks (PANs),” IEEE-SA, 2005.

[42] ZigBee Alliance, “The ZigBee Specification,” ZigBee Alliance Inc., San Ramon, 2012.

[43] M. Conti, C. Boldrini, S. S. Kanhere, E. Mingozzi, E. Pagani, P. M. Ruiz, and M. Younis, “From MANET to People-centric Networking: Milestones and Open Research Challenges,” *Comput. Commun.*, vol. 71, no. C, pp. 1-21, 2015.

[44] IETF Propose Standard, “RFC 4944: Transmission of IPv6 Packets Over IEEE 802.15.4 Networks,” IETF Trust, 2007.

[45] IETF Standard, “RFC 7252: The Constrained Application Protocol (CoAP),” IETF Trust, 2014.

[46] P. Thubert, T. Watteyne, M. R. Palattella, X. Vilajosana, and Q. Wang, “IETF 6TSCH: Combining IPv6 Connectivity with Industrial Performance,” in *Proc. 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2013.

[47] R. Daidone, G. Dini, and G. Anastasi, “On Evaluating the Performance Impact of the IEEE 802.15.4 Security Sub-layer,” *Comput. Commun.*, vol. 47, no. 1, p. 65–76, 2014.

[48] P. D. Marco, C. Fischione, F. Santucci, and K. H. Johansson, “Modeling IEEE 802.15.4 Networks Over Fading Channels,” *IEEE Trans. on Wireless Commun.*, vol. 13, no. 10, pp. 5366 - 5381, 2014.

[49] D. D. Guglielmo, F. Restuccia, G. Anastasi, M. Conti, and S. K. Das, “Accurate and Efficient Modeling of 802.15.4 Unslotted CSMA/CA through Event Chains Computation,” *IEEE Trans. on Mobile Comput.*, vol. 15, no. 12, pp. 2954 - 2968, 2016.

[50] S. Brienza, D. D. Guglielmo, G. Anastasi, M. Conti, and V. Neri, “Strategies for Optimal MAC Parameter Setting in IEEE 802.15.4 Wireless Sensor Networks: A Performance Comparison,” in *Proc. IEEE Symposium on Comput. and Commun. (ISCC)*, 2013.

[51] K. Pister and L. Doherty, “TSMP: Time Synchronized Mesh Protocol,” in *Proc. IASTED International Symposium on Distributed Sensor Networks (DSN08)*, Florida, 2008.

[52] IETF Standard, “RFC 7554: Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in The Internet of Things (IoT): Problem Statement,” IETF Trust, 2015.

[53] T. Watteyne, L. Doherty, J. Simon, and K. Pister, “Technical Overview of SmartMesh IP,” in *Proc. 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2013.

[54] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, “On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH,” *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3655 - 3666, 2013.

[55] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Networks*, vol. 7, no. 3, p. 537–568, 2009.

[56] Z. Li, Y. Peng, D. Qiao, and W. Zhang, “LBA: Lifetime Balanced Data Aggregation in Low Duty Cycle Sensor Networks,” in *Proc. IEEE INFOCOM*, 2012.

[57] L. Guntupalli, “Energy Efficient Data Transmission- Chapter 3,” in *Energy Efficient In Wireless Sensor Networks: Transmission Protocols and Performance Evaluation. Ph.D. Dissertation,* 2016, pp. 24-28.

[58] L. Guntupalli, J. Martinez-Bauset, F. Y. Li, and M. A. Weitnauer, “Aggregated Packet Transmission in Duty-Cycled WSNs: Modeling and Performance Evaluation,” *IEEE Trans. on Vehicular Techn.,* vol. 66, no. 1, pp. 563 - 579, 2017.

[59] M. C. Albrecht and P. AZ, Introduction to Discrete Event Simulation, 2010.

[60] T. Issariyakul and E. Hossain, Introduction to Network Simulator NS2, 2nd ed., Springer US, 2012.

[61] G. J. Carneiro, NS-3: Network Simulator 3- UTM Lab Meeting, Porto: INESC Porto, 2010.

[62] A. Varga and R. Hornig, “An Overview of the OMNeT++ Simulation Environment,” in *Proc. 1st International Conference on Simulation Tools and Techni. for Commun. Nets. and Systems & Workshops,* Marseille, France, 2008.

[63] P. Levis, N. Lee, M. Welsh, and D. Culler, “TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications,” in *Proc. 1st International Conference on Embedded Networked Sensor Systems,* Los Angeles, 2003.

[64] G. Z. Papadopoulos, K. Kritsis, A. Gallais, P. Chatzimisios, and T. Noel, “Performance Evaluation Methods in Ad-hoc and Wireless Sensor Networks: A Literature Study,” *IEEE Commun. Magazine,* vol. 54, no. 1, pp. 122 - 128, 2016.

[65] J. Horneber and A. Hergenröder, “A Survey on Testbeds and Experimentation Environments for Wireless Sensor Networks,” *IEEE Commun. Surveys & Tut.,* vol. 16, no. 4, pp. 1820 - 1838, 2014.

[66] G. Werner-Allen, P. Swieskowski, and M. Welsh, “MoteLab: A Wireless Sensor Network Testbed,” in *Proc. Fourth International Symposium on Information Processing in Sensor Networks, 2005.,* 2005.

[67] V. Handziski, A. Köpke, A. Willig, and A. Wolisz, “TWIST: A Scalable and Reconfigurable Testbed for Wireless Indoor Experiments with Sensor Networks,” in *Proc. 2nd International Workshop on Multihop Ad hoc Networks: From Theory to Reality,* Florence, Italy, 2006.

[68] S. Fischer, D. Pfisterer, S. P. Fekete, and A. Kröller, “WISEBED – Pan-European Wireless Sensor Network Testbeds,” Fachgespraech der GI/ITG-Fachgruppe Kommunikation und Verteilte System, 2008.

[69] Berkeley Webs, “Berkeley Webs: Wireless Embedded Systems,” 2007. [Online]. Available: http://smote.cs.berkeley.edu/motescope/. [Accessed 4 April 2017].

[70] M. Amjad, M. Sharif, M. K. Afzal, and S. W. Kim, “TinyOS-New Trends, Comparative Views, and Supported Sensing Applications: A Review,” *IEEE Sensors Journal,* vol. 16, no. 9, pp. 2865 - 2889, 2016.

[71] K. Tan, D. Wu, A. Chan, and P. Mohapatra, “Comparing Simulation Tools and Experimental Testbeds for Wireless Mesh Networks,” in *Proc. IEEE International Symposium on WoWMoM,* 2010.

[72] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, “A Centralized Scheduling Algorithm for IEEE 802.15.4e TSCH based Industrial Low-power Wireless Networks,” in *Proc. IEEE Wireless Commun. and Net. Conference*, 2016.

Appendix A

Implementation Structure

NS-3 is built on modules which have dependencies on different libraries. Specifically, examples and test are used to build the necessary module of importance. The module of interest to us is lr-wpan. Subsequently, we present the structure.

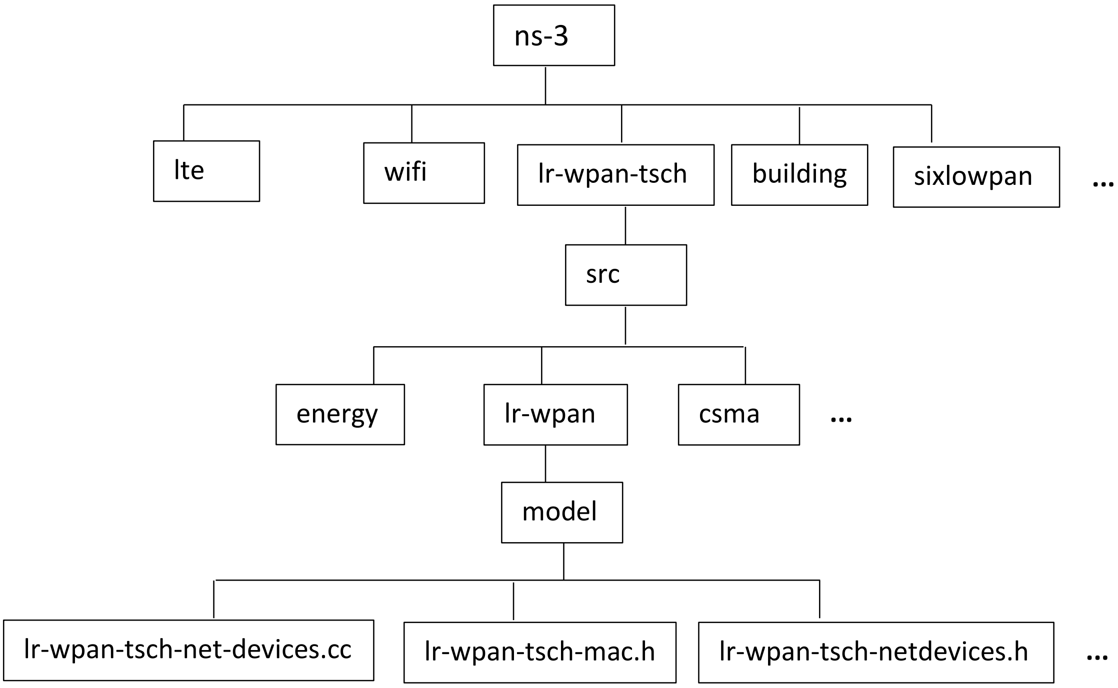


Figure A- 1: Illustration of ns-3 implementation structure.

The modules highlighted in the structure are not limited to only these but depends on application scenario. Typically, running of specific scripts is under the control of waf to ensure that the system builds all shared libraries, set their paths correctly and make them available at runtime. To run a program, for example, RBS, the following is used; ./waf -- run scratch/RBS, if RBS is inside the scratch directory. The below figure shows scenario structure.

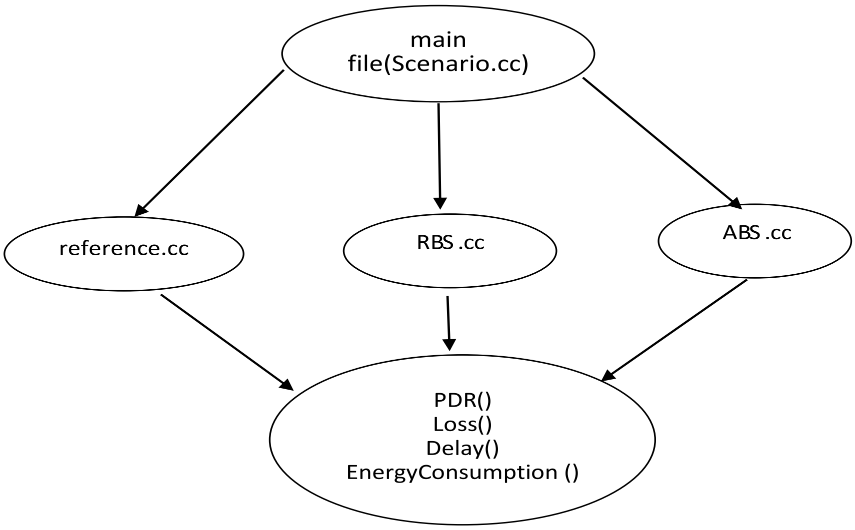


Figure A- 2: Illustration of scenario structure

The structure shows how scenarios are connected. The main file contains all schemes, thus running the main file will call all other sub-files in different file locations. All schemes are evaluated based on the same performance metrics. In the forthcoming appendix, we present RBS implementation codes.

Appendix B

RBS Implementation Codes

We present RBS implementation code as part of the structure in Figure A-2. To run the RBS script, we need to specify the complete path to the script file after using ./waf command.

****

****

```
/* -*-  Mode: C++; c-file-style: "gnu"; indent-tabs-mode:nil; -*- */
/*
 * Copyright (c) 2011 The Boeing Company
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307
USA
 *
 * Authors:
 *  Gary Pei <guangyu.pei@boeing.com>
 *  Tom Henderson <thomas.r.henderson@boeing.com>
 *  Luis Pacheco <luisbelem@gmail.com>
 *  Peishuo Li <pressthunder@gmail.com>
 *  Peter Kourzanov <peter.kourzanov@gmail.com>
 */
#include <cassert>
#include "lr-wpan-scenario2b.h"
#include <ns3/energy-module.h>
#include <ns3/lr-wpan-error-model.h>
#include <ns3/lr-wpan-tsch-net-device.h>
#include <ns3/mobility-model.h>
#include <ns3/single-model-spectrum-channel.h>
#include <ns3/friis-spectrum-propagation-loss.h>
#include <ns3/log.h>
#include <iostream>
#include "ns3/core-module.h"
#include "ns3/netanim-module.h"
#include <ns3/lr-wpan-module.h>
#include <ns3/propagation-loss-model.h>
#include <ns3/propagation-delay-model.h>
#include <ns3/simulator.h>
#include <ns3/single-model-spectrum-channel.h>
#include <ns3/mobility-module.h>
#include <ns3/constant-position-mobility-model.h>
#include <ns3/packet.h>
```

```
#include <ns3/node-container.h>
#include <string>
#include <ns3/uinteger.h>
#include <ns3/nstime.h>
#include <ns3/abort.h>
#include <ns3/command-line.h>
#include <fstream>
#include <vector>
#include <ns3/test.h>
#include <ns3/log.h>
#include <ns3/callback.h>
#include <ns3/lr-wpan-error-model.h>
#include <ns3/lr-wpan-net-device.h>
#include <ns3/spectrum-value.h>
#include <ns3/lr-wpan-spectrum-value-helper.h>
#include <ns3/lr-wpan-mac.h>
#include <ns3/node.h>
#include <ns3/net-device.h>
#include <ns3/mac16-address.h>
#include "lr-wpan-tsch-helper.h"
NS_LOG_COMPONENT_DEFINE ("LrWpanTschScenario2B");
using namespace ns3;
using namespace std;

namespace ns3{
      uint32_t g_received_b = 0; // throughput result
      int nrnodes_b = 3;             //number of nodes, not including
the coordinator
      int pktCounter_b = 0;
      int historic_b = 0 ;
      double rate_delay_b = 10 ;
      double duration_b = 0 ;
      struct BasicPacketData
      {
      public:
            BasicPacketData (Ptr<const Packet>& p) : p(p)
            {
                  //get packets 802.15.4-Header
                  LrWpanMacHeader h;
                  p->PeekHeader(h);
                  seq_nr = h.GetKeyIdSrc32();
            }
            uint32_t seq_nr;
            double packet_time;
      private:
            Ptr<const Packet> p;
      };

      std::vector<BasicPacketData> enqueue_packetList_b;
      std::vector<BasicPacketData> dequeue_packetList_b;
void StopSimulation_B()
{
      Simulator::Stop();
}
```

```
static void PacketEnqued (Ptr<LrWpanTschNetDevice> dev, Ptr<const
Packet> p)
{
            pktCounter_b++;


}
static void PacketDequed (Ptr<LrWpanTschNetDevice> dev, Ptr<const
Packet> p)
{
    double simtime = Simulator::Now().GetSeconds() ;
      BasicPacketData pkt_info(p) ;
      pkt_info.packet_time = simtime ;
      dequeue_packetList_b.push_back(pkt_info) ;

}
static void PacketSend(Ptr<LrWpanTschNetDevice> dev, Ptr<const Packet>
p)
{
      g_received_b++;
      double simtime = Simulator::Now().GetSeconds() ;
      BasicPacketData pkt_info(p) ;
      pkt_info.packet_time = simtime ;
      enqueue_packetList_b.push_back(pkt_info) ;
}
void EnableTschB(LrWpanTschHelper* lrWpanHelper,NetDeviceContainer&
netdev)
{
      lrWpanHelper->EnableTsch(netdev,0,duration_b);
}

static void PacketSendInfo (Ptr<LrWpanTschNetDevice> dev, Ptr<const
Packet> p, uint8_t retries )
{

}
static void MakeCallbacks(std::vector<Ptr <LrWpanTschNetDevice> > devs)
{
    for (std::vector<Ptr <LrWpanTschNetDevice> >::const_iterator i =
devs.begin(); i!=devs.end(); ++i)
    {
        Ptr<LrWpanTschNetDevice> dev = *i;

        dev->GetNMac()->TraceConnectWithoutContext ("MacTxEnqueue",
MakeBoundCallback(&PacketEnqued,dev));
            dev->GetNMac()->TraceConnectWithoutContext ("MacTx",
MakeBoundCallback(&PacketSend,dev));
        dev->GetNMac()->TraceConnectWithoutContext ("MacRx",
MakeBoundCallback(&PacketDequed,dev));
            dev->GetNMac()->TraceConnectWithoutContext ("MacSentPkt",
MakeBoundCallback(&PacketSendInfo,dev));
    }

}
```

```
LrWpanTschScenario2B::LrWpanTschScenario2B(int nodecount , double txp ,
uint32_t chn,uint32_t mindegree,uint32_t maxdegree,int
packetCount,uint32_t slot_num)
{
      pktsize = 20;
      nrnodes_b = nodecount;
      txPower = txp;
      channelNumber = chn;
      min_PacketDegree = mindegree;
      max_PacketDegree = maxdegree;
      pkt_count = packetCount ;
      duration_b=8;
      timeslotnum = slot_num ;
}
LrWpanTschScenario2B::~LrWpanTschScenario2B()
{

}
void LrWpanTschScenario2B::CreateNodes()
{
      panCoord.Create (1);
      sensors.Create(nrnodes_b);
      lrwpanNodes.Add(panCoord);
      lrwpanNodes.Add(sensors);

      MobilityHelper mobility;
      mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
      mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
            "GridWidth", UintegerValue(4),
            "MinX", DoubleValue (0.0),
            "MinY", DoubleValue (0.0),
            "DeltaX", DoubleValue (5),
            "DeltaY", DoubleValue (5),
            "LayoutType", StringValue ("RowFirst"));
      mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
      mobility.Install (lrwpanNodes);
}
double LrWpanTschScenario2B::delayTime()
{
      double TotalDelayTime = 0.0 ;
      for(uint32_t i = 0 ; i < dequeue_packetList_b.size() ; i++)
      {
            BasicPacketData pktde_info = dequeue_packetList_b[i];
            uint32_t seq_num = pktde_info.seq_nr;
            double   dequeue_time = pktde_info.packet_time;
            for(uint32_t j = 0 ; j < enqueue_packetList_b.size() ; j++)
            {
                  BasicPacketData pkten_info = enqueue_packetList_b[j];
                  if (pkten_info.seq_nr == seq_num)
                  {
                        double perDelayTime= dequeue_time -
pkten_info.packet_time ;
                        TotalDelayTime += perDelayTime ;
                        break;
```

```
}
            }
      }
      double avg_delay_time = (TotalDelayTime * rate_delay_b) /
(double)(pktCounter_b) ;
      avg_delay_time = (1 - (double)historic_b / (double)g_received_b)
* avg_delay_time;
      return avg_delay_time ;
}
std::vector<double> LrWpanTschScenario2B::DoRun()
{
      NS_LOG_UNCOND("================================Scenario2B
Simulation Result ======================================") ;
    std::vector<double> sim_result ;
      std::vector<uint32_t> orglist_b ;
      CreateNodes() ;
      //Enable PCAP and Ascii Tracing
      AsciiTraceHelper ascii;
      /////////////////////////////////
      // Configure lrwpan nodes
      /////////////////////////////////
      // Each device must be attached to the same channel
      Ptr<SingleModelSpectrumChannel> channel =
CreateObject<SingleModelSpectrumChannel> ();
      Ptr<LogDistancePropagationLossModel> propModel =
CreateObject<LogDistancePropagationLossModel> ();
      Ptr<ConstantSpeedPropagationDelayModel> delayModel =
CreateObject<ConstantSpeedPropagationDelayModel> ();
      propModel->SetAttribute ("Exponent", DoubleValue (3.3));
      propModel->SetAttribute ("ReferenceLoss", DoubleValue (58.5));
      // chain models
      channel->AddPropagationLossModel (propModel);
      channel->SetPropagationDelayModel (delayModel);
      LrWpanTschHelper lrWpanHelper(channel,nrnodes_b+1,false,true);

      Ptr<OutputStreamWrapper> stream = ascii.CreateFileStream ("lr-
wpan-tsch.tr");
      NetDeviceContainer netdev = lrWpanHelper.Install (lrwpanNodes);

//    lrWpanHelper.EnablePcapAll (string ("lr-wpan-tsch"), true);
//    lrWpanHelper.EnableAsciiAll (stream);

      //--------------------------AssociateToPan------------------------
--------------
      lrWpanHelper.AssociateToPan(netdev,123);
      // -------------------------AddSlotFrame-------------------------
--------------
      MlmeSetSlotframeRequestParams slotframeRequest;
      slotframeRequest.slotframeHandle = 0;
      slotframeRequest.Operation = MlmeSlotframeOperation_ADD;
      slotframeRequest.size = timeslotnum;
      for (uint32_t i = 0 ; i < netdev.GetN() ; i++)
      {
```

```
DynamicCast<LrWpanTschNetDevice>(netdev.Get(i))->GetNMac()-
>MlmeSetSlotframeRequest (slotframeRequest);
	}

	//--------------------------AddAdvLink for beacon----------------
--------------------------
	AddLinkParams alparams;
	alparams.slotframeHandle = 0;
	alparams.channelOffset = 0;
	alparams.linkHandle = 0;
	alparams.timeslot = 0;
	lrWpanHelper.AddAdvLink(netdev,0, alparams) ;
	//--------------------------AddLink------------------------------
-------------

	uint32_t ASN_TimeSlot = 0 ;
	for (uint32_t i = 1 ; i < netdev.GetN() ; i++)
	{
		ASN_TimeSlot++ ;
		if (ASN_TimeSlot%timeslotnum == 0)
		{
			ASN_TimeSlot++ ;
			alparams.linkHandle = ASN_TimeSlot ;
			alparams.timeslot = ASN_TimeSlot% timeslotnum ;
			lrWpanHelper.AddLink(netdev,i,0,alparams,true);
		}
		else if((ASN_TimeSlot+1) % timeslotnum == 0)
		{
			alparams.linkHandle = ASN_TimeSlot;
			alparams.timeslot = ASN_TimeSlot% timeslotnum ;
			lrWpanHelper.AddLink(netdev,i,0,alparams,true);
			ASN_TimeSlot++;
		}
		else
		{
			alparams.linkHandle = ASN_TimeSlot;
			alparams.timeslot = ASN_TimeSlot% timeslotnum ;
			lrWpanHelper.AddLink(netdev,i,0,alparams,true) ;
		}
		DynamicCast<LrWpanTschNetDevice>(netdev.Get(i))-
>GetCsmaCa()->SetMacMinBE(0);
		DynamicCast<LrWpanTschNetDevice>(netdev.Get(i))-
>GetCsmaCa()->SetSlottedCsmaCa();
	}

/*	}*/
	/////////////////////////////////
	// Configure Trace
	/////////////////////////////////

	std::vector<Ptr <LrWpanTschNetDevice> > devs;
	LrWpanSpectrumValueHelper svh;
	Ptr<SpectrumValue> psd = svh.CreateTxPowerSpectralDensity
(txPower, channelNumber);
```

```
for(unsigned int i = 0 ; i < netdev.GetN(); i++)
      {

      devs.push_back(DynamicCast<LrWpanTschNetDevice>(netdev.Get(i))) ;
            DynamicCast<LrWpanTschNetDevice>(netdev.Get(i))->GetPhy ()-
>SetTxPowerSpectralDensity (psd);
      }
      MakeCallbacks(devs) ;
      //////////////////////////////////
      // Configure Energy
      //////////////////////////////////
      EnergySourceContainer sources =
lrWpanHelper.InstallEnergySource(lrwpanNodes) ;
      DeviceEnergyModelContainer deviceModels =
lrWpanHelper.InstallEnergyDevice(netdev,sources) ;
      //////////////////////////////////
      // Start Simulation
      //////////////////////////////////
      Simulator::Schedule(Seconds(0),&EnableTschB,&lrWpanHelper,netdev)
;
      uint32_t rndbuf[nrnodes_b] ;
      for(uint32_t pktcount = 0 ; pktcount < (uint32_t)pkt_count ;
pktcount++)
      {
            for(int i = 0 ; i < nrnodes_b ; i++)
            {
                  uint8_t buffer[pktsize];
                  memset(buffer,0,sizeof(buffer)) ;
                  Ptr<UniformRandomVariable> x =
CreateObject<UniformRandomVariable> ();
                  uint32_t rn = x-
>GetInteger(min_PacketDegree,max_PacketDegree);
                  for (uint32_t index = 0 ; index < (uint32_t)pktsize ;
index++)
                  {
                        if (index < rn)
                              buffer[index] = 1 ;
                        else
                              buffer[index] = 0 ;
                  }
                  Ptr<Packet> packet= Create<Packet> (buffer,pktsize);
                  DynamicCast<LrWpanTschNetDevice>(netdev.Get(i+1))-
>SendPacket(packet,netdev.Get(0)-
>GetAddress(),true,0x86DD,false,0,false) ;
                  if (pktcount != 0 &&rndbuf[i] == rn)
                  {
                        historic_b++;
                        g_received_b--;
                  }
                  rndbuf[i] = rn ;
            }
      }
      Simulator::Run ();
      NS_LOG_UNCOND("                 ===========node count :" <<
nrnodes_b << "==============                      ");
```

```
      double successRate ,lossrate;
      successRate = (double)( g_received_b)/ (double)(pktCounter_b -
historic_b)  ;
      lossrate = 1-successRate ;
      double energyConsumption = 0 ;
      for (int i = 0 ; i < nrnodes_b+1 ; i++)
      {
            energyConsumption += deviceModels.Get (i)-
>GetTotalEnergyConsumption () * (1 - (double)historic_b /
(double)pktCounter_b);
      }
      double delay_time = delayTime() ;
      NS_LOG_UNCOND ("delay per packet (ms) " << delay_time);
      NS_LOG_UNCOND ("success per packet " << successRate );
      NS_LOG_UNCOND ("loss per packet " << lossrate );
      NS_LOG_UNCOND ("Total Energy Consumption" << energyConsumption) ;
      NS_LOG_UNCOND ("Co-ordinator energy consumption :" <<
deviceModels.Get (0)->GetTotalEnergyConsumption () * (1 -
(double)historic_b / (double)pktCounter_b));
      for (int i = 1 ; i <= nrnodes_b ; i++)
      {
            double node_energy = deviceModels.Get (i)-
>GetTotalEnergyConsumption () * (1 - (double)historic_b /
(double)pktCounter_b);
            NS_LOG_UNCOND ("node" << i << "  energy consumption :" <<
node_energy);
      }
      NS_LOG_UNCOND ("total " << pktCounter_b <<" history " <<
historic_b) ;
      sim_result.push_back(successRate);
      sim_result.push_back(lossrate) ;
      sim_result.push_back(energyConsumption);
      sim_result.push_back(delay_time) ;
      pktCounter_b = 0 ;
      g_received_b = 0 ;
      historic_b = 0 ;
      duration_b = 0 ;
      Simulator::Destroy ();
      return sim_result ;
}
}
```