Adaptive Resource Allocation for Faster Formation of 6TiSCH IoT Network

Thesis submitted to the
Indian Institute of Technology Guwahati
for the award of the degree

of

Doctor of Philosophy

in

Computer Science and Engineering

Submitted by

Alakesh Kalita

Under the guidance of

Dr. Manas Khatua

Department of Computer Science and Engineering

Indian Institute of Technology Guwahati

May, 2022

Dr. Manas Khatua
Assistant Professor
Computer Science and Engineering
Indian Institute of Technology Guwahati
North Guwahati, Assam, India – 781039
Email : manaskhatua@iitg.ac.in

Certificate

This is to certify that this thesis entitled, **"Adaptive Resource Allocation for Faster Formation of 6TiSCH IoT Network"**, being submitted by **Alakesh Kalita**, to the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, for partial fulfillment of the award of the degree of Doctor of Philosophy, is a bonafide work carried out by him under my supervision and guidance. The thesis, in my opinion, is worthy of consideration for award of the degree of Doctor of Philosophy in accordance with the regulation of the institute. To the best of my knowledge, it has not been submitted elsewhere for the award of the degree.

Date: 18.05.2022
Place: Guwahati

.....
Dr. Manas Khatua
(Thesis Supervisor)

Declaration

I certify that:

- The work contained in this thesis is original and has been done by myself and under the general supervision of my supervisor.
- The work reported herein has not been submitted to any other Institute for any degree or diploma.
- Whenever I have used materials (concepts, ideas, text, expressions, data, graphs, diagrams, theoretical analysis, results, etc.) from other sources, I have given due credit by citing them in the text of the thesis and giving their details in the references. Elaborate sentences used verbatim from published work have been clearly identified and quoted.
- I also affirm that no part of this thesis can be considered plagiarism to the best of my knowledge and understanding and take complete responsibility if any complaint arises.
- I am fully aware that my thesis supervisor is not in a position to check for any possible instance of plagiarism within this submitted work.

Date: 17/05/2022
Place: Guwahati

(Alakesh Kalita)

Dedicated to

Maa & Deuta

For their unconditional love, blessings and constant inspiration

Acknowledgements

With the patience, help, and knowledge of many people pursuing Ph.D. becomes very exciting adventure for me. Words are not enough to express my gratitude towards all the people who helps me to present my Ph.D. thesis in this form.

First and foremost, I would like to express my heartfelt gratitude to my supervisor *Dr. Manas Khatua* for his consistent support, encouragement, inexhaustible patience, and positive guidance in this adventure. He has given me the freedom to pursue research ideas and develop my research skills. I profusely thank him for correcting my mistakes repeatedly and keeping me engaged in my Ph.D. work. I fall short of words to express my gratitude to him.

I would also like to thank the members of my Doctoral Committee - *Prof. Diganta Goswami*, *Dr. Tamarapalli Venkatesh*, and *Dr. Sushanta Karmakar* for their insightful comments and suggestions, which made me improve the quality and clarity of my work. I am also thankful to the anonymous reviewers of my research work in various forums for their critical comments, which helped me to add quality to my work. I am also thankful to all the developers of the Contiki-NG Operating System and FIT IoT-LAB testbed for their amazing work and making them open source, by which I am able to perform my Ph.D. work experiments.

I want to thank the heads of the Department of Computer Science and Engineering during my Ph.D. at IITG - *Prof. S. V. Rao* and *Prof. Jatindra Kumar Deka* for allowing me to use the facilities and the available resources. I am deeply thankful to - *Ms. Gauri Khuttiya Deori* for efficiently handling the administrative work. I am obliged to all the faculty members, staff, and security personnel for their constant help and support. I would also like to thank the staff at the Academic Affairs office who were supportive in processing my applications and grant requests.

I would like to acknowledge *MHRD*, *Govt. of India* gratefully for the financial support rendered throughout my years of Ph.D., without which this research could not have taken shape. I would also like to acknowledge the Department of Computer Science and Engineering and the Welfare Board of IITG for the travel grants that helped me present my research work.

I want to mention a few friends who made my Ph.D. more memorable and exciting - Arabindu, Ato, Chaitanya, Gyanendra, Hemanta, Jennil, Kamal, Nayan, Naro, Omesh, Pankaj, and Pawan Da. Thank you all for the Cricket, Football, Badminton, Table Tennis, and Tennis matches. Thank you all for the countless and sleepless night parties, adventures trips, and many more. Thank you all for everything. All is just great - almost like a fairy tale. I am indeed thankful to my fellow Ph.D. colleagues - Anirban, Arijit, Atul, Dipojjwal Da, Karnish, Maithilee, Nilotpal, Moustafa, Parikshit Da, Soumiya, Surajit Da, Ujjwal Da for creating a wonderful experience at my workplace.

I want to thank Rimjhim for her encouragement, support, love, beautiful memories, and cherished moments during my Ph.D. days. I want to thank Nurzaman and Atis Elsts for their suggestions related to Contiki-NG. I would like to thank Mridul and Jitumani for the beautiful memories in IIIT Guwahati during my initial research career. I would also like to thank Rupam and Pankaj for the beautiful memories during our stay at Uzan Bazar, and their encouragement and support at every step of my life.

I want to thank the friends from my undergraduate and school days - Chandan, Dhiman, Diganta, Faruk, Gagan, Himadri, Sarma, Sibbir, Tony for the beautiful memories and cherished moments spent with them.

Finally, yet importantly, I would like to thank almighty God and my family - Ma, Deuta, Dada, Munu dada, for their unconditional love, support, caring, warmth, and profound encouragement all these years. My special regards to my parents for their intense trust and confidence over me which made my journey immensely happy, secure, and comfortable. I fall short of words to express my gratitude to them.

Abstract

Internet of Things (IoT) is a network of physical objects (also called "*things*") that are found in homes, healthcare systems, industries, to name a few. IoT helps the computational system to interact with the physical world by sensing, communicating, and actuating. The IPv6 over the Time Slotted Channel Hopping mode of IEEE 802.15.4e (6TiSCH) network is standardized to meet high reliability, end-to-end latency, and network lifetime requirements of various IoT applications. Basically, 6TiSCH leverages the traditional IPv6 based network protocols to build upon IEEE 802.15.4e TSCH Medium Access Control (MAC) protocol. The formation of 6TiSCH network should happen before establishing end-to-end data communication. So, the 6TiSCH Working Group (WG), formed by Internet Engineering Task Force (IETF), published 6TiSCH Minimal Configuration (6TiSCH-MC) standard for 6TiSCH network formation. 6TiSCH-MC allows only one cell per slotframe (known as *shared cell* or *minimal cell*) to transmit all types of control packets.

6TiSCH network formation process is started by the Join Registrar/Coordinator (JRC), and all the new nodes (aka pledges) join one-by-one following the joining procedure in the multi-hop 6TiSCH networks. Note that faster formation of 6TiSCH network is challenging because of the inherent channel hopping feature of TSCH as the pledges do not know in which channel and at what time control packets are transmitted by the already joined nodes and/or the JRC. Apart from this, the resource allocated by 6TiSCH-MC standard is static in nature, and it did not provide any mechanism to handle congestion in shared cell. Furthermore, 6TiSCH-MC did not provide any information regarding the transmission rates of different control packets used to form the network. Therefore, we set the objective of this thesis is to augment the 6TiSCH-MC standard by updated mechanisms for achieving faster formation of 6TiSCH IoT network.

It is observed that congestion in shared cell becomes an inevitable problem when the number of joined nodes increases, and it gradually degrades the performance of 6TiSCH network formation scheme. So, the first contribution of this thesis targets to reduce congestion in shared cell by dynamically varying the beacon generation interval. For this, we propose *channel condition based dynamic beacon interval* (C2DBI) scheme, which varies the beacon generation interval with the current congestion status in shared cell instead of using a fixed beacon generation interval. In the next work, we observe that due to fixed priority assignment to the control packets and insufficient transmission of routing control packet, formation of 6TiSCH network gets delayed. So, we propose *opportunistic transmission of control packets* (OTCP) scheme, which dynamically adjusts the priority of control packets and provides sufficient routing information during network formation. Furthermore, sometimes it may happen that a node needs to transmit its packets immediately. OTCP

further allows the nodes having such urgent packets to transmit their packets in less time. In brief, OTCP gives higher priority to the transmission of routing control packet whenever it is urgent. We further inspect the effect of routing control packet on 6TiSCH network formation by designing a Markov Chain-based analytical model. Analytical results show that improper selection of the Trickle Algorithm (TA)'s parameters, which are responsible for routing control packet generation and transmission, cause congestion in shared cell. Further, the default TA does not provide fair transmission opportunity to all the nodes in shared cell of 6TiSCH network. Therefore, to address these issues, we propose *adaptive control packet broadcasting* (ACB) scheme as our third contribution of this thesis. ACB reduces the congestion in shared cell and provides equal control packet transmission opportunities to the nodes. However, congestion in shared cell is not fully avoidable but can be controlled further. So, in the next contribution, we design a non-cooperative gaming model to determine the optimal control packet transmission probabilities of the nodes to counterbalance the congestion in minimal cell. After solving our gaming model, we obtain the optimal control packet transmission probabilities of the nodes. Subsequently, we propose a game theory based congestion control (GTCC) scheme by which nodes are restricted to transmit their packets frequently. This results in less congestion in shared cell which further improves the joining time and energy consumption of the pledges. Further, we realize that the number of shared cells per slotframe can be increased to achieve significant performance improvement during 6TiSCH network formation. The existing schemes do not use all the channels at a time to transmit control packet. So, in the fifth contribution of this thesis, we leverage all the available channels at a time in order to increase the number of shared cells per slotframe for quicker transmission of control packets. The *radio duty cycle* of the nodes is also considered while increasing the number of shared cells per slotframe. For this, at first, we propose *autonomous allocation and scheduling of minimal cell* (TACTILE) scheme, and later, we further enhance the performance of TACTILE by proposing *time-variant RGB* (TRGB) model for minimal/shared cell allocation and scheduling in 6TiSCH networks. It is noteworthy that both the TACTILE and TRGB do not hamper the data transmitting cell unlike the existing works. Both the TACTILE and TRGB significantly improve the performance of 6TiSCH formation compared to our previous works and existing benchmark schemes. Finally, it is worthwhile to mention that all the proposed schemes are evaluated by Markov Chain based theoretical analysis, simulation, and testbed experiments.

As a whole, this dissertation improves the performance of 6TiSCH network during its formation period in terms of joining time and energy consumption of the IoT nodes while maintaining stable network.

Keywords: IoT, IEEE 802.15.4e, TSCH, 6TiSCH, Network formation, RPL, Channel estimation, Opportunistic transmission, Autonomous allocation, Autonomous scheduling

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols

AJT Average Joining Time

ASF Average number of Slotframes

c Counter for received consistent DIOs

CBR Channel Busy Ratio

$E_{joinedNode}$ Average Energy consumption by a joined node

E_{pledge} Average Energy consumption by a pledge

E_{RxData} Energy require to receive a data packet

E_{TxData} Energy require to transmit a data packet

I_{dio} DIO generation interval

I_{dio}^{max} Maximum DIO generation interval

I_{dio}^{min} Minimum DIO generation interval

I_{eb} Beacon generation interval

I_{eb}^{max} Maximum beacon generation interval

I_{eb}^{min} Minimum beacon generation interval

k Trickle Algorithm's redundancy constant

L Slotframe length

N_c Number of available channels

N_D Maximum number of Trickle states

N_{nbr} Number of neighboring nodes

N_{s_cell} Number of shared cell in a DIO interval

P_{DIO_S} Successful transmission probability of DIO

P_{dio} Transmission probability of DIO in shared cell

P_{EB_S} Successful transmission probability of EB

P_{eb} Transmission probability of EB in shared cell

P_{empty} Probability of no transmission in shared cell

P_{jns} Probability of transmission in shared cell by a joined node

P_{JRQ_S} Successful transmission probability of JRQ

P_{jrq} Transmission probability of JRQ in shared cell

P_{JRS_S} Successful transmission probability of JRS

P_{jrs} Transmission probability of JRS in shared cell

P_{loss} Packet loss probability

P_{nns} Probability of transmission in shared cell by a new node

P_{ts} Probability of transmission in shared cell

PJT Parent's Joining Time

S Number of suppressed DIOs

T_i Time duration of slot i

T_{dio} Number of transmitted DIOs

W Time interval for CBR calculation

Acronyms

6LoWPAN IPv6 over Low power Wireless Personal Area Network.

6TiSCH IPv6 over the TSCH mode of IEEE 802.15.4e.

6TiSCH-MC 6TiSCH Minimal Configuration.

ACK Acknowledgment.

AJT Average Joining Time.

AMCA Asynchronous Multi-Channel Adaptation.

ASF Average Slotframe Number.

ASFC Absolute Slotframe Count.

ASN Absolute Slot Number.

BLINK Radio Frequency Identification Blink.

CCA Clear Channel Assessment.

CDF Cumulative Distribution Function.

CoAP Constrained Application Protocol.

CSMA/CA Carrier-Sense Multiple Access with Collision Avoidance.

CW Contention Window.

DIO DODAG Information Object.

DIS DODAG Information Solicitation.

DODAG Destination Oriented Directed Acyclic Graph.

DSME Deterministic and Synchronous Multi-channel Extension.

EB Enhanced Beacon.

IE Information Element.

IEEE-SA Institute of Electrical and Electronics Engineers Standards Association.

IETF Internet Engineering Task Force.

IoT Internet of Things.

IPv6 Internet Protocol Version 6.

IT Information Technology.

JRC Join Registrar/Coordinator.

JRQ Join Request.

JRS Join Response.

LLDN Low Latency Deterministic Network.

LLN Low power and Lossy Network.

MAC Medium Access Control.

RPL Routing Protocol for Low power and Lossy network.

SDN Software Defined Network.

SW Slotframe Window.

TA Trickle Algorithm.

TDMA Time Division Multiple Access.

TSCH Time Slotted Channel Hopping.

WG Working Group.

"You have to dream before your dreams can come true."

~A.P.J. Abdul Kalam (1931 - 2015)

1 Introduction

1.1 Research Context

In early 2000's, Kevin Ashton from the Massachusetts Institute of Technology coined the term "Internet of Things (IoT)", which attracts the governments and leading Information Technology (IT) companies and gets recognized as crucial part of sustainable economic growth of the nations. IoT is intended to connect almost all physical objects (refers to as "*things*" or "*nodes*" in IoT) that are found everywhere such as in homes, health-care systems, industries, various transportation systems [1–3].

According to [4], IoT is

"*World-wide network of interconnected uniquely addressable objects, based on standard communication protocols.*"

IoT allows the nodes to sense the environment and then exchange the sensed information among themselves by establishing communication among them, which helps the associated computational system to interact with the physical world. The information received from the nodes is used for analyzing the current environmental situation, and based on this analysis, IoT changes the behavior of the environment as per requirements [5], [6]. Various application domains such as industry [7], [8], health-care [9], [10], smart home [11], [12], smart city [13], [14], smart grid [15], [16], smart transportation [17], and smart precision agriculture [18] are the few applications of IoT. The nodes in IoT are called *smart* [1], [19] in three aspects: (i) they can *sense or monitor* their surroundings, (ii) they can *communicate* or *exchange* their information with other nodes, (iii) they can *change the behavior* of their surroundings. Nodes use different *sensors* for sensing or monitoring the environment, and most commonly, they usage *wireless communication* for exchanging information. The behaviors of the environment are changed (known as *actuation*) using some mechanical devices which work based on the input provided by the nodes [20], [21]. Therefore, IoT can improve the efficiency, safety, and security of its various applications in the physical world through sensing, communication, and performing actuation on the environment.

Most of the IoT devices[1] are resource-constrained in terms of processing capacity (which has only a few kilobytes (KBs) of RAM), memory (also in KBs), and energy (mainly battery operated) [22]. Nevertheless, IoT should provide highly reliable, delay-bounded, and energy-efficient communication to most of its applications. Therefore, it is very challenging to implement and fulfill various requirements of different IoT applications with resource-constrained devices.

In the recent few years, because of the increasing demand of wireless technology for

[1]Note that we use the terms device, mote, and node interchangeably in this thesis as all of them mean the same.

both sensing and actuating, many standards have been introduced for IoT networks as per the requirement of different applications [22–24]. Few of them are *ZigBee* [25], *IEEE 802.15.4* [26], *Bluetooth* [27], *WirelessHART* [28] and *ISA-100.11a* [29]. These standards use 2.4 Ghz frequency band, whereas the other standards such as Narrowband Internet of Things (NB-IoT) [30], IEEE 802.11.ah [31], LoRaWAN (Low Power and Long-range Wide Area Network) [32] use *Sub* $1GHz$ frequency band. Among these standards, *IEEE 802.15.4* is the most widely used standard for resource-constrained Low power and Lossy Network (LLN) [22]. The IEEE 802.15.4 defines both the Physical (PHY) layer and Medium Access Control (MAC) layer specifications and envisioned to be used in all IoT applications. However, it is observed that the IEEE 802.15.4 standard suffers from unbounded delay, low reliability, less protection against interference and multi-path fading [33], [34], and hence, not suitable for most of the IoT applications. One of the main reasons for these issues is the usage of a single channel in the $2.4GHz$ frequency band. On the other hand, ZigBee suffers from interoperability and scalability issues as it does not use the existing communication infrastructure and is a proprietary solution. Similarly, WirelessHART and ISA-100.11a suffer from scalability issues as both of them follow centralized approach for establishing/maintaining communication among the nodes. The *Sub* $1GHz$ based standards are designed for long-range but low data rate oriented applications. Note that LoRaWAN can be operated on $2.4GHz$ to achieve higher data rates but at the cost of range.

Therefore, to fulfill the increasing demands of sensor/actuator-based technologies and deal with the issues of existing standards, the Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA) published *IEEE 802.15.4e* [35] amendment in 2012. The IEEE 802.15.4e is an extension of the existing IEEE 802.15.4 [26]. It supports multiple transmissions at a time using multiple physical channels (maximum *16 channels*). It also allows the nodes to change their physical communication channel after each transmission, which is known as *channel hopping*. Channel hopping helps in getting rid of multi-path fading and interference on the transmission channels in LLNs. Thus, IEEE 802.15.4e improves the throughput and reliability in resource-constrained LLNs. Apart from the enhancement

in the Physical layer, it has added various functionalists in the MAC layer to support different application requirements. Specifically, IEEE 802.15.4e has mentioned five different MAC behavior modes depending on different types of applications. These modes are Time Slotted Channel Hopping (TSCH), Deterministic and Synchronous Multi-channel Extension (DSME), Low Latency Deterministic Network (LLDN), Asynchronous Multi-Channel Adaptation (AMCA), Radio Frequency Identification Blink (BLINK). Among these MAC behavior modes, both TSCH and DSME are included in the recent revised version of IEEE 802.15.4-2016 [36] because of their wide applicability in different IoT application domains.

TSCH gains attention among the researchers because it provides high reliability and guaranteed latency by utilizing its core features such as channel/frequency hopping and Time Division Multiple Access (TDMA) based transmission channel access. TSCH is intended to provide reliable and delay-bounded communication [37] in various application domains such as industrial automation, process monitoring, and oil and gas industry as a part of *Industry 4.0* revolution. The channel hopping feature greatly improves the network's reliability by effectively mitigating the effects of interference and multi-path fading at a considerable scale. On the other hand, TDMA based channel access, in which time is divided into small and fixed duration *timeslot*[1] provides delay-bounded and guaranteed packet delivery. Note that the collection of several timeslots is known as *slotframe*, which repeats over time. Slotframe is used for managing the schedule for both data and control packet transmission. It is further noteworthy that the IEEE 802.15.4e standard does not specify how the TSCH schedule is formed and managed.

As IEEE 802.15.4e defines only the PHY and MAC layers, therefore, to provide Internet connectivity to the resource-constrained devices, the Internet Engineering Task Force (IETF) created the IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) Working Group (WG). The main aim of this WG is to establish interoperability between TSCH and IPv6 [38]. It provides an *open communication network protocol stack* to connect IEEE

[1]A single timeslot duration is long enough, usually 10 *ms*, to transmit a packet (maximum 127 *Bytes* long), and receive its Acknowledgment (ACK), if it is needed. A timeslot repeats over time, and only one node is allowed to transmit its data packet using a particular channel at a given timeslot.

802.15.4e TSCH MAC based multi-hop and lossy networks to existing IPv6 networks. TSCH is limited in establishing global *synchronization* among the participated constrained devices. On the other hand, 6TiSCH layer is necessary to fill the gap between the IETF's IPv6 communication stack [22] and TSCH. Figure 1.1 shows a 6TiSCH based LLN connected with the traditional IPv6 network through a network gateway. Here, the user(s) can access the data in real-time, or data can be stored in the central server(s) for further processing using IPv6 connectivity.

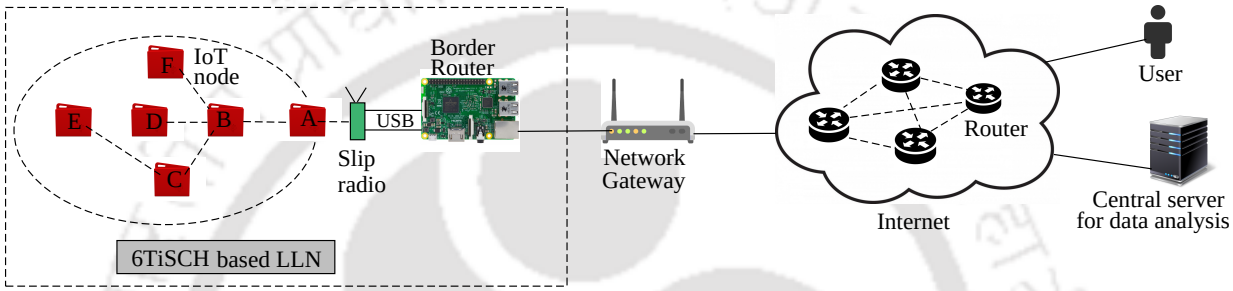


Figure 1.1: *A 6TiSCH network connected to the end user and central server over Internet*

Apart from the interoperability, the most important task is network bootstrapping/-formation by which pledges[1] can join with existing 6TiSCH networks. Several attributes are essential to consider during network formation, such as less joining time, low energy consumption, less resource allocation, and network stability. For this purpose, 6TiSCH-WG released the 6TiSCH Minimal Configuration (6TiSCH-MC) standard [40]. This standard mentioned the strategy for allocating timeslot to transmit control packets. Specifically, 6TiSCH-MC allocated only a single cell[2] in a slotframe (known as *shared cell* or *minimal cell*) for the transmission of all types of control packets for forming and maintaining the network throughout its lifetime. Whereas, manageable cells (also called dedicated cell) are used for data packet transmission. All the cells i.e., shared cell and manageable cells can be represented by a matrix of cells as shown in Figure 1.2. In this figure, the slot offsets

[1]IETF used the term "pledge" to designate a new joining node, which has not completed the joining process in a given secure network and is therefore not trusted by the network [38], [39]. So, we refer to a new node as a pledge in this thesis.

[2]A cell is represented by the touple [`slot offset` and `channel offset`], which denotes the time and channel together to transmit a packet. Slot offset denotes the timeslot (index number) within a slotframe, and channel offsets are mapped with unique physical channels.

and channel offsets are used as cell indices (showing in `X-axis` and `Y-axis`, respectively), and a slotframe repeats after 5 *timeslots*. Manageable cells are used for data packet transmission, and a particular manageable cell is assigned only for a pair of nodes in a slotframe for guaranteed data packet transmission. Whereas, in shared cell, any node can transmit its control packets. For example, only node Ⓔ transmits its data packet to node Ⓒ in the manageable cells (1,0) and (2,3) of slotframe "n". On the other hand, in the shared cell of same slotframe, node Ⓔ transmits to node Ⓒ and node Ⓐ transmits to node Ⓑ simultaneously.

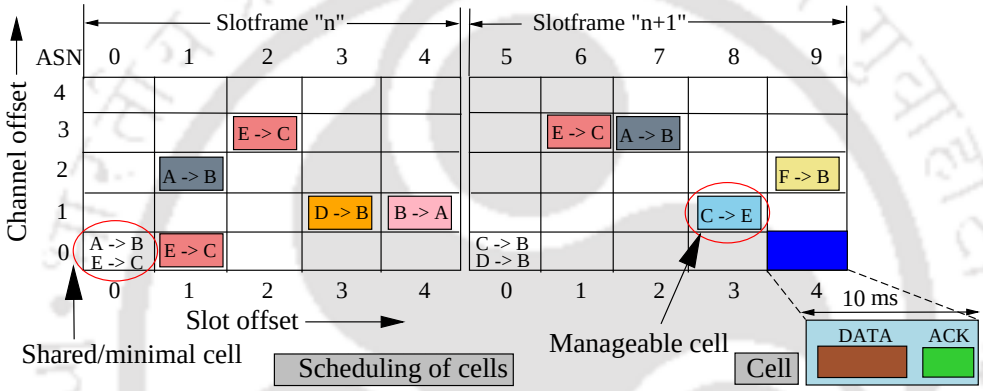


Figure 1.2: *Scheduling of shared cell and manageable cell in the repeated slotframes.*

Note that 6TiSCH-MC allocates the shared cell at the fixed position of each slotframe i.e., at `slot offset=0`, and `channel offset=0` for all the nodes as shown in Figure 1.2. Basically, to maintain synchronization and finding optimal routing route, this default allocation policy forces the nodes to use the same shared cell of a slotframe. The remaining cells starting from the second timeslot to the last timeslot of a slotframe i.e., *manageable cells* are scheduled using the scheduling function used in the network such as [41], [42] after/along with the formation of 6TiSCH network.

6TiSCH-MC allocates only one shared cell (or minimal cell) per slotframe for the transmission of all types of control packets.

1.2 Motivation of the Research Work

IEEE 802.15.4e standard has left several issues open to the researcher. Few of them are resource allocation during/after the formation of TSCH network, scheduling for exchanging data packet and control packet in time-frequency domains, and analysis of the protocols under different network settings. Among these issues, formation of TSCH based 6TiSCH network is the main focus of this dissertation. It is because the formation of 6TiSCH network becomes challenging as TSCH allows the nodes to change their physical communication channel after each timeslot. On the other hand, at the beginning of network formation, the pledges do not know about the channel where the joined nodes transmit their control packets. The transmission time of those control packets are also unknown to pledges. Note that control packets are not transmitted on a particular channel always as that channel might have interference and multi-path fading issues, which can severely degrade the performance. Further, the transmission of control packets on a fixed channel is not compliant with IEEE 802.15.4e. Therefore, pledges need to scan all the channels randomly one after another by keeping their radios active. This causes enormous energy consumption of the pledges. Furthermore, a pledge is allowed to transmit its control packets for further expansion of the multi-hop 6TiSCH network and can transmit/forward data packet only after joining the 6TiSCH network successfully. This leads to increased joining time of the pledges, which are located at the multiple hops away from the Join Registrar/Coordinator (JRC). The increasing joining time of the pledges also affects on the overall network throughput and end-to-end data packet latency. Therefore, faster formation of 6TiSCH network is essential to increase its lifetime and ensure reliability to its applications.

A pledge requires several types of control packets to join in a 6TiSCH network completely. Specifically, at the beginning, the pledge should receive Enhanced Beacon (EB) frame to get the basic network configuration information, followed by DODAG Information Object (DIO) packet transmitted by the Routing Protocol for Low power and Lossy network (RPL) [43] to join the routing tree, or in other words, Destination Oriented Directed Acyclic Graph (DODAG). The pledge further needs to exchange Join Request (JRQ) and Join Response

(JRS) frames with the JRC for its secure enrollment [39], [44]. An example of the step by step network formation process of a small network is presented in Figure 1.3. Note that

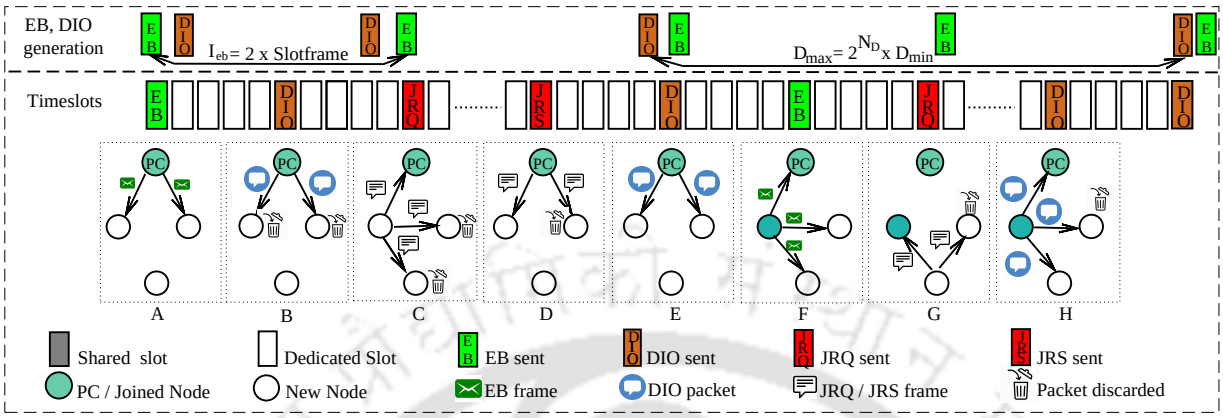


Figure 1.3: *An example of 6TiSCH network formation*

neither the IEEE 802.15.4e standard nor 6TiSCH-MC mentioned the transmission rate of these control packets. On the other hand, multiple joined nodes transmit their control packets using shared cell i.e., using the same channel simultaneously. So, the possibility of packet collision in wireless medium increases with the increasing number of nodes in the network. This increases the joining time of the pledges and the energy consumption of both joined nodes and pledges.

Although IETF has published 6TiSCH-MC standard for the formation of 6TiSCH network, Vallati *et.al* [45] proved that 6TiSCH-MC does not provide sufficient resource i.e., number of shared cells per slotframe during network formation. This insufficient resource allocation results in high congestion in shared cell. This further results in control packet collision and ultimately increases the formation time of 6TiSCH network as nodes take longer time to transmit their control packets. The work in [45] improved the performance of 6TiSCH network formation significantly by increasing the number of shared cells per slotframe but at the cost of high energy consumption. Furthermore, the work in [45] affects the performance of the networks in terms of throughput and end-to-end packet latency as some of the data transmitting cells are converted into shared cells. The solution provided by [46] may not be suitable for all types of networks as it fixed the EB broadcasting probability in a slotframe irrespective of the size and node density of the networks. On the other hand,

the works in [47–57] did not consider the formation of multi-hop 6TiSCH network. In brief, these works considered only the initial phase of 6TiSCH network formation i.e., one-hop TSCH synchronization. The key focus of these works was to transmit as many EB as possible for faster synchronization of one-hop TSCH network. This results in the allocation of more network bandwidth and the consumption of additional energy. Moreover, these works are not 6TiSCH-MC compliant and create synchronization issue among the nodes, which further results in unstable network. Apart from this, none of the existing works explore the effect of different control packets generation rate, priority of the control packets, uncertainty in accessing the shared cell, fair transmission by the available nodes, and usage of multiple channels at a time for control packet transmission during the formation of multi-hop 6TiSCH network.

Therefore, the existing issues and unexplored research directions of 6TiSCH network formation motivate us to develop improved techniques for it. Thus, the main aim of this dissertation is set to improve the performance of multi-hop 6TiSCH network formation.

Thesis goal

Improve the performance of multi-hop 6TiSCH network formation in terms of nodes' joining time, energy consumption, and network stability while following the resource allocation policy made by IETF 6TiSCH-MC standard.

1.3 Thesis Objectives

The faster formation of 6TiSCH network helps in efficient sensory data transmission and increases the network lifetime. However, efficient transmission of different control packets, fair transmission opportunity to the nodes, judicious transmission rate of different control packets, and utilization of all the available channels at a time for control packet transmissions have significant effect on 6TiSCH network formation and have not been explored. Therefore, we frame the objectives of this dissertation to improve the performance of multi-hop 6TiSCH network formation in terms of nodes' joining time, their energy consumption, and network

stability. In brief, we describe the objectives of this Thesis as follows:

- *Channel Condition based Dynamic Beacon Interval*: Proposing a scheme to reduce the congestion in shared cell by varying the beacon generation interval of the nodes dynamically instead of using fixed beacon generation interval all the time.
- *Opportunistic Transmission of Control Packet*: Assigning priority to the control packets depending on their requirement, providing sufficient routing information, and enabling quick access to shared cell.
- *Adaptive Control Packet Broadcasting*: Providing efficient and fair transmission of routing information carrying control packet and enabling fair control packet transmission opportunity to all the nodes.
- *Non-Cooperative Gaming based Control Packet Transmission*: Exploiting non-cooperative game theory to find optimal control packet transmission probability in a slotframe in order to alleviate congestion in shared cell.
- *Autonomous Allocation and Scheduling of Shared Cell*: Providing schemes to leverage all the available channels at a timeslot in order to increase the number of shared cells per slotframe for quick transmission of control packet.

1.4 Thesis Contributions

To improve the performance of multi-hop 6TiSCH network formation, we explore the effect of different control packets generation rate and their transmission probabilities, priorities of the control packets, uncertainty in accessing the shared cell, unfair transmission by the nodes, and misutilization of multiple channels at a single timeslot for control packet transmission and contribute several schemes to deal with these issues. In the next subsections, we briefly describe each of these proposed schemes one by one.

1.4.1 Channel Condition based Dynamic Beacon Interval (C2DBI)

At first, we design a Markov Chain model to analyze the formation of 6TiSCH-MC based multi-hop 6TiSCH network in terms of pledges joining time and their energy consumption. Analytical results show that the performance of 6TiSCH network formation degrades with the increasing number of joined nodes in the network. The reason for this performance degradation is the increasing congestion in shared cell due to fixed beacon generation intervals of the nodes. Hence, it proved that the allocation (i.e., one shared cell per slotframe) made by 6TiSCH-MC is not sufficient when the nodes use fixed beacon generation intervals in a large network. To overcome this problem, instead of using a fixed beacon generation interval, we propose to vary the EB generation interval of the joined nodes dynamically based on the present channel status. All the joined nodes measure the congestion in shared cell for a particular interval, and depending on that, the nodes decide their next EB generation interval. In brief, nodes increase their EB generation interval if the congestion is high, otherwise, set to default standardized value. Hence, the proposed scheme neither increases the waiting time for the pledges to receive EB frame nor congests the shared cell. Ultimately, it reduces the congestion in shared cell even though the number of nodes increases in the network. We provide a theoretical analysis of the proposed scheme in terms of both joining time and energy consumption. It shows how the performance of 6TiSCH network during its formation is improved with varied EB generation intervals. Furthermore, the proposed scheme is also evaluated using simulation and testbed experiments.

1.4.2 Opportunistic Transmission of Control Packet (OTCP)

We further analyze the formation of multi-hop 6TiSCH network using our previous Markov Chain model. Our probabilistic analysis further reveals that 6TiSCH-MC affects the formation of 6TiSCH network because it considers (i) EB frame has the highest priority over all other control packets, (ii) it does not provide sufficient and quick transmission of DIO packet during network formation, and (iii) it does not provide any provision to access the shared cell quickly. To deal with these problems, we propose two schemes i.e., *opportunistic*

priority alternation and rate control (OPR) and *opportunistic channel access* (OCA). In OPR, a joined node decides the priority of a control packet depending on the current network condition. It gives the highest priority to DIO packet over EB frame when a pledge or joined node needs the DIO packet immediately or urgently. We named this kind of packet as an urgent packet. OPR also allows quick generation of DIO packet during the formation of 6TiSCH network. On the other hand, OCA allows the nodes to access the shared cell quickly if they have urgent packet to transmit. We evaluate both OPR and OCA individually by designing separate Markov Chain based analytical models. We also perform extensive simulation experiments to evaluate both the proposed schemes individually and combine them. At last, we evaluate both OPR and OCA together using testbed experiments.

1.4.3 Adaptive Control Packet Broadcasting (ACB)

We design a Markov Chain model to study the behavior of various Trickle Algorithm (TA)'s parameters on DIO packet generation and transmission, and so 6TiSCH network formation. Our analysis reveals that improper selection of various Trickle parameters' values and default working principle of TA can degrade the performance of 6TiSCH network formation. During some network instances, TA enables burst transmission of DIO packet, whereas, sometimes, it does not allow the nodes to transmit their DIO packet for a long time. Furthermore, the default TA does not give a guarantee to the fair transmission of DIO packets among the nodes. To address these issues of TA, we propose *dynamic Trickle algorithm* scheme for efficient and fair transmission of DIO packets in 6TiSCH network. The proposed scheme skips some of the intermediate DIO packet generating Trickle states during network consistency to reduce congestion in shared cell. It also varies one Trickle parameter, i.e., *redundancy constant* dynamically depending on current network condition to alleviate the congestion in shared cell. Furthermore, we propose another scheme to provide fair DIO transmission opportunity to all the nodes. This scheme varies the Trickle *listen-only* period of the nodes depending on their previous DIO transmission history instead of using random listen-only time. Apart from this, to reduce the congestion and provide fair control packet transmission

opportunity to the nodes further, we propose *slotframe window (SW)-based adaptive control packet transmission* scheme. This scheme restricts the nodes from transmitting their control packets more frequently. This results in less congestion in shared cells and provides equal control packet transmission opportunities to all nodes. We perform testbed experiments to validate the proposed schemes together.

1.4.4 Game Theory based Congestion Control (GTCC)

In our previous analysis of C2DBI, GTCC and ACB, we find that the transmission rates of EB and DIO severely affect the formation of 6TiSCH network when the number of nodes increases in the network. Therefore, we design a non-cooperative game to find the optimal control packet transmission probability in a shared cell of the nodes. In this proposed game, joined nodes are considered as *players* who want to maximize their profit i.e., control packet transmission probability irrespective of the strategies of other neighboring nodes. In the *pay-off function* of the proposed game, control packet transmission probability is considered as a *utility function*, and the *minimal cell busy ratio* and *remaining energy* of the nodes are considered as *price functions*. We prove that the proposed game has a unique Nash equilibrium point and a unique solution. The obtained solution of the proposed game, i.e., optimal control packet transmission probability using Lagrange multiplier and Karush-Kuhn-Tucker (KKT) conditions is used in our proposed *game theory based congestion control* (GTCC) scheme. GTCC calculates *slotframe window* (SW) size for each node based on its calculated optimal transmission probability and restrains the node in transmitting its control packets for SW amount of time. Thus, GTCC reduces the congestion in shared cell and improves the performance of 6TiSCH network formation. Along with the theoretical analysis, we perform testbed experiments to validate the proposed scheme.

1.4.5 Autonomous Allocation and Scheduling of Shared Cell (AAS)

It is observed that the existing schemes do not use all the available physical channels at a time in each shared cell to transmit control packet. These works used only one channel

at a time. So, it reduces the EB reception probability of the pledges and increases the congestion in shared cell. This cumulatively degrades the performance of 6TiSCH network formation concerning joining time and energy consumption of the nodes. To deal with this under-utilization of channels, we propose *autonomous allocation and scheduling of minimal cell* (TACTILE) scheme. TACTILE autonomously allocates and schedules the shared cells of the nodes without any signaling overhead. For this, we propose *autonomous minimal cell allocation* (ALLOT) and *hierarchical odd-even minimal cell scheduling* (CHOICE) schemes to distribute the shared cells among all the available channels and for conflict-free scheduling of the allocated shared cells, respectively. We evaluate the proposed TACTILE using Markov Chain-based theoretical analysis and testbed experiments.

However, later, we observe that TACTILE suffers from network stability issues as it does not maintain a common cell for all the nodes for exchanging routing information. A common cell is necessary to maintain a stable network and construct an efficient routing tree for upward and downward routing. Furthermore, the allocation made by TACTILE (specifically, ALLOT) is static which remains unchanged throughout the lifetime. So, the repeated collision of control packet transmission is possible when two or more nodes add the same set of minimal cells. This is the motivation for further designing the *time-variant RGB* (TRGB) model for minimal cell allocation and scheduling in 6TiSCH network. For this, we propose *time-variant autonomous allocation of minimal cell* (INSTALL) scheme, which changes the position of autonomously allocated minimal cells after each slotframe to avoid repeated control packet collision. Subsequently, we further propose *Red-Green-Blue* (RGB) scheduling algorithm for conflict-free scheduling of the allocated shared cells by INSTALL and enabling close interaction with RPL. RGB separates the transmission of RPL control traffic from the transmission of other control packets, which significantly improves the joining time and energy consumption of the pledges by maintaining stable networks. We evaluate our TRGB model using Markov Chain based theoretical analysis and extensive testbed experiments.

1.4.6 Summary of Contributions

The major contributions of the thesis can be summarized as follows:

- A Markov Chain model is proposed to show the demerits of fixed beacon interval scheme during the formation of multi-hop 6TiSCH network.
- A *channel condition based dynamic beacon interval* (C2DBI) scheme is proposed to reduce congestion in shared cell, which, in turn, improves the performance of 6TiSCH network formation.
- An *opportunistic priority alternation and rate control* (OPR) scheme is proposed to deal with the demerits of EB's highest priority and to provide sufficient DIO packet quickly.
- An *opportunistic channel access* (OCA) scheme is proposed to transmit urgent control packets with minimum delay.
- A Markov Chain based analytical model is proposed for analyzing the effect of Trickle Algorithm (TA)'s parameters on 6TiSCH network formation.
- A *dynamic Trickle algorithm* is proposed to reduce congestion in minimal cell and to provide fair DIO transmission opportunity to all the nodes.
- A *slotframe window (SW) based adaptive control packet transmission* scheme is proposed to restrict the nodes from transmitting several control packets within a short period in order to reduce congestion in shared cell. The SW-based scheme also provides fair control packet transmission opportunities to the nodes.
- Design a non-cooperative game to find optimal control packet transmission probability in a shared cell to alleviate congestion.
- A *game theory based congestion control* (GTCC) scheme is proposed by which nodes can efficiently transmit their control packets.

- An *autonomous minimal cell allocation scheme* (ALLOT) is proposed to utilize all the available channels at a time, and so, to increase the number of shared cells per slotframe.
- A *hierarchical odd-even minimal cell scheduling scheme* (CHOICE) is proposed to schedule the shared cells allocated by ALLOT efficiently.
- A *time-variant autonomous allocation of minimal cell* (INSTALL) scheme is proposed for autonomous and time-dependent allocation of minimal cell.
- A *Red-Green-Blue* (RGB) scheduling scheme is proposed for conflict-free scheduling of the shared cells allocated by INSTALL and to enable close interaction with RPL.
- Finally, the proposed schemes are evaluated using one or more of the following methods: Markov Chain based theoretical analysis, Contiki-NG OS based Cooja simulator, and FIT IoT-LAB Testbed.

1.5 Organization of the Thesis

The Figure 1.4 shows the life cycle of a joined node's control packet transmission process following the 6TiSCH-MC standard. In this cycle, at the beginning, the joined node generates EB frame periodically and it follows the Trickle Algorithm (TA) to generate DIO packet. After that, the node chooses the highest priority frame/packet from its transmission buffer and then try to transmit that frame/packet by allocating and scheduling the shared cell followed by participating in the contention race for shared channel access. Our contributions i.e., C2DBI, OTCP, ACB, GTCC and AAS are shown in different events during the packet transmission process by the joined node. Finally, Figure 1.4 illustrates the organization of this thesis, which maps the different chapters onto the area of our work and the corresponding publications. In brief, the rest of the thesis is organized as follows:

Chapter 1: Introduction

Introduction, motivation, and contributions of the thesis are presented in this chapter.

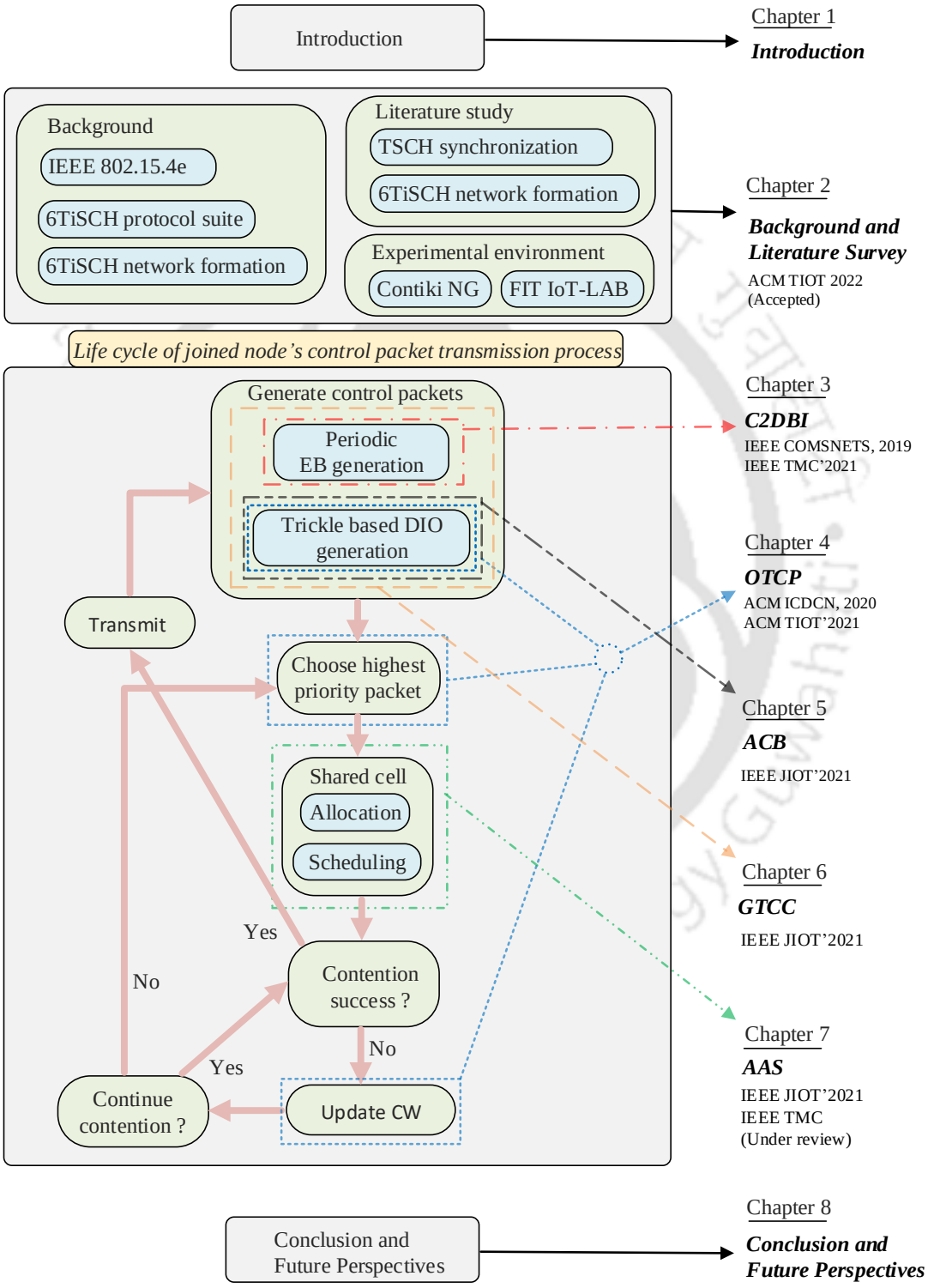


Figure 1.4: *Area of our work wise organization of the thesis*

Chapter 2: Background and Literature Survey

This chapter provides detailed background on 6TiSCH network followed by the discussion on state-of-the-art works related to 6TiSCH network formation. It also describes the experimental environment used for performing simulation and testbed experiments.

Chapter 3: Channel Condition based Dynamic Beacon Interval

This chapter proposes C2DBI scheme for generating EB frame dynamically to reduce congestion in 6TiSCH network. Theoretical analysis, simulation, and testbed experimental results are provided to validate the proposed C2DBI scheme.

Chapter 4: Opportunistic Control Packet Transmission

This chapter exploits the demerit of EB's highest priority, insufficient transmission of DIO packet, and uncertainty in accessing shared cell during the formation of 6TiSCH network. Subsequently, it proposes OTCP scheme to deal with these problems. Theoretical analysis, simulation, and testbed experimental results are provided to show the effectiveness of the proposed OTCP.

Chapter 5: Adaptive Control Packet Broadcasting

This chapter explores the effect of TA's parameters on 6TiSCH network formation and proposes ACB scheme to improve the formation of 6TiSCH network. Testbed experiments are carried out to validate the proposed ACB scheme.

Chapter 6: Non-cooperative Gaming based Control Packet Transmission

This chapter inspects the gaming based control packet transmission in 6TiSCH network and proposes GTCC scheme for efficient control packet transmission in 6TiSCH network. Theoretical analysis and testbed experimental results are provided to show the effectiveness of the proposed scheme.

Chapter 7: Autonomous Allocation and Scheduling of Shared Cell

This chapter proposes TACTILE and TRGB schemes for autonomous allocation and scheduling of shared cells for quick transmission of control packet. Theoretical analysis, testbed experimental results are provided to show the effectiveness of the proposed schemes.

Chapter 8: Conclusions and Future Perspectives

Finally, this chapter concludes the thesis with summarization of work done, and future direction for possible future work on 6TiSCH network formation.

"Education is not the learning of facts, but the training of the mind to think."

~Albert Einstein (1879 - 1955)

2

Background and Literature Survey

In this chapter, at the beginning, we briefly discuss the IEEE 802.15.4e TSCH MAC behavior followed by the brief discussion of 6TiSCH network protocols stack. As the major objectives of this dissertation concerns 6TiSCH network formation, we briefly discuss the procedure of how a 6TiSCH network is formed using an example scenario. Thereafter, a detailed survey on the works that considered 6TiSCH network formation is presented. Apart from this, we also provide detailed information about the experimental environments which are used to evaluate the proposed solutions in this dissertation.

2.1 Background

2.1.1 Overview of IEEE 802.15.4e TSCH

As stated before, IEEE 802.15.4e TSCH MAC behavior mode is designed to fulfill the basic requirements of LLNs such as reliability, guaranteed packet delivery, and energy efficiency. To achieve these, TSCH divides time into small and fixed duration *timeslot* that repeats over time. A timeslot is long enough, usually 10 *ms*, to transmit a packet and receive its ACK. And the collection of several timeslots is known as *slotframe*. Every slotframe repeats periodically for forming/maintaining a communication schedule for both data and control packet transmission. Figure 2.1 shows the timeslot architecture in TSCH. A node can be in any of the three different radio states in a given timeslot such as *receiving* (`Rx`), *transmitting* (`Tx`), or *idle*, and it can use only one physical channel at the given timeslot for either `Tx` or `Rx`.

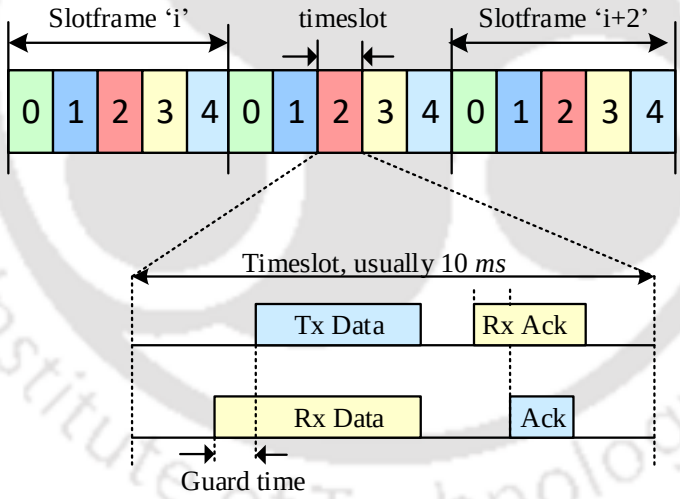


Figure 2.1: *Timeslot structure in TSCH*

Note that a pair of timeslot and channel is called *cell*, and a scheduling function is used to allocate unique cell to the nodes for their communication. So, a cell being used by a pair of nodes is not used by any other pairs present in their communication range. In this way, TSCH provides deterministic channel access to the nodes to exchange their information. This results in deterministic delay-bounded data packet delivery and also increases overall network throughput. It is noteworthy that there exist few timeslots in a slotframe which

are used by more than one node. This type of timeslot is called *shared timeslot*, and the cell associated with it is called *shared cell*. Nodes perform TSCH-CSMA/CA channel access mechanism to transmit their packet in shared cell in order to avoid repeated collision.

IEEE 802.15.4e TSCH takes advantage of the availability of multiple channels to deal with the problems of interference and multi-path fading on the transmission channels. It allows the nodes to change their physical communication channels after each transmission. This mechanism is called *channel hopping* and TSCH improves the reliability of the communication using channel hopping. Now, as the transmitting nodes change their physical communication channels in every timeslot, the receiving nodes should also change their receiving channel after each timeslot aligning/synchronizing with the transmitting nodes' channel hopping sequence. For this, the nodes compute their physical channels using the Equation (2.1) for maintaining this channel synchronization as follows:

$$f = F[(ASN + \texttt{channel offset})\ mod\ N_c] \tag{2.1}$$

where f denotes the calculated `channel index` associated with a particular frequency, F denotes the sequence of channel hopping i.e., `lookup table` for channels. Absolute Slot Number (ASN) denotes the number of total timeslots that have elapsed from the inception of the network. So, ASN is an integer number that starts from 0 and gets incremented by one after each timeslot and it is same for all the nodes in a network. The `channel offset` is the fixed and unique integer assigned to the pair of transmitter and receiver(s) nodes by the scheduling algorithm, and N_c denotes the number of channels used in the network. The output of this equation provides a pseudo-random channel hopping sequence.

Figure 2.2 depicts an example of channel hopping mechanism where in one case, a pair of nodes is assigned the `slot offset=2` and `channel offset=1` i.e., cell [2, 1] by the scheduling algorithm. Using the Equation (2.1), both the transmitter and receiver nodes change their physical communication channel in the assigned timeslot of a slotframe, e.g., from channel 5 to 11, 11 to 2, and so on. Apart from this channel hopping, TSCH also allows multiple transmissions at a time using multiple channels, which further improves the

network throughput. It is done by assigning different `channel offsets` to the nodes at the same `slot offset`.

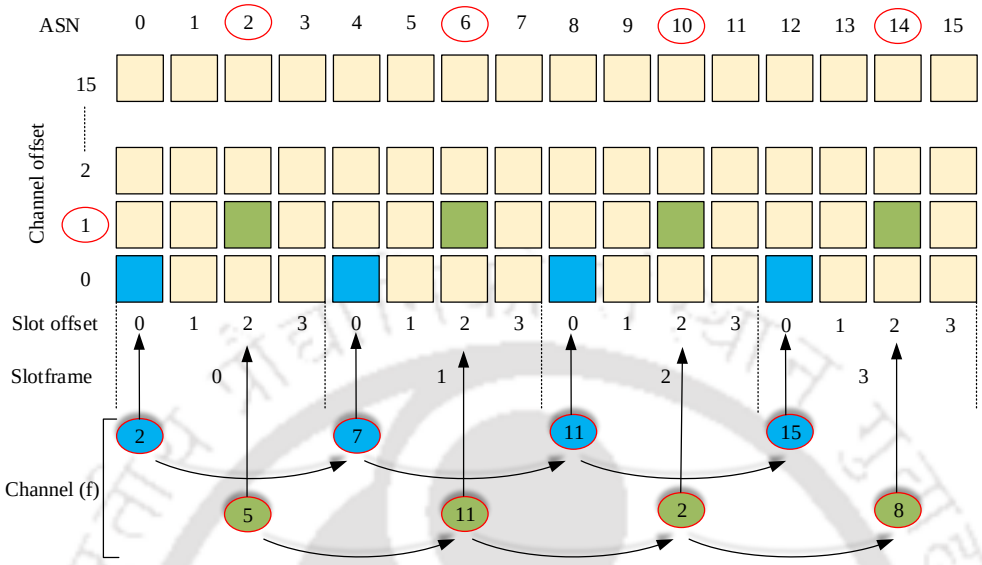


Figure 2.2: *An example of channel hopping in TSCH*

2.1.2 6TiSCH Protocol Suite

In 2007, IETF formed IPv6 over Low power Wireless Personal Area Network (6LoWPAN) WG to provide IPv6-based Internet connectivity to the IEEE 802.15.4 networks [58]. For this, the *6LoWPAN adaptation layer* is added on top of Layer-2 [59], whereas IETF's 6TiSCH WG is formed to provide Internet connectivity to the devices that run TSCH in their link-layer with the 6LoWPAN adaptation layer. So, currently, 6TiSCH WG is working on to provide interoperability between IEEE 802.15.4e TSCH and IETF's upper layer protocols stack. This WG deals with the scheduling of communication cells among the nodes based on the received signals from the TSCH link layer and other upper layers. Apart from this cell scheduling, 6TiSCH WG also deals with resource allocation (bandwidth or number of cells) for efficient 6TiSCH network bootstrapping. In brief, 6TiSCH bridges the IETF's upper layer protocols stack with the IEEE 802.15.4e TSCH to provide IPv6-based Internet connectivity to the LLN devices. The overall protocols stack is shown in Figure 2.3, where the 6TiSCH stack is rooted on the top of IEEE 802.15.4 physical layer [22], [60], and [38]. RFC8180 [40] is released by 6TiSCH WG for 6TiSCH network bootstrapping, which pro-

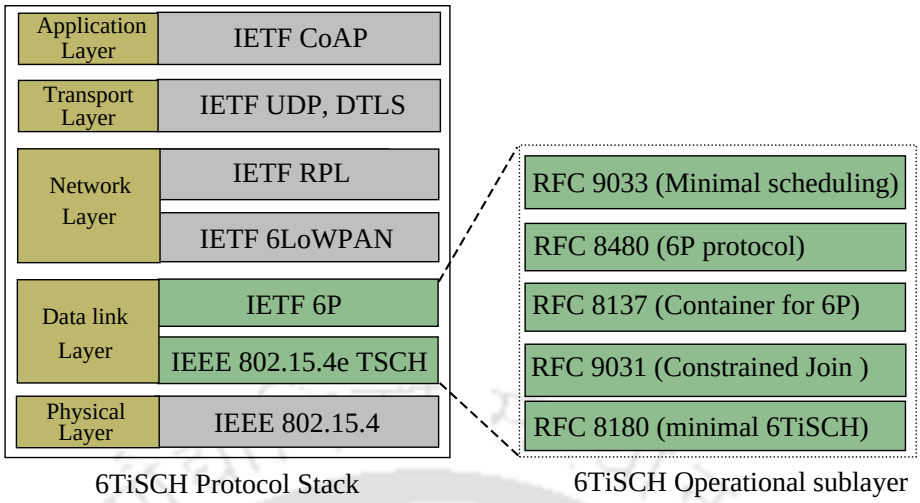


Figure 2.3: *The complete 6TiSCH network protocols stack*

vides the basic procedure to form a network and information about minimum resource in terms of cell used for forming the multi-hop 6TiSCH networks. On the other hand, works in [44], and RFC9031 [39] described the procedure for secure enrollment of nodes during their joining process. RFC8137 [61] and RFC8480 [42] described the distributed cell management protocols and [62–65] are few scheduling schemes used for scheduling data packet transmission cell. The 6LoWPAN adaptation layer includes stateless header compression and fragmentation technique as mentioned in RFC4944 [66] and contextual header compression techniques as mentioned in RFC6282 [67], RFC8025 [68] and RFC8138 [69] for providing the IPv6-based Internet connectivity. The RPL [43] is used in the network layer to organized the network in the form of DODAG topology following the RFC6552 [70]. On the other hand, Constrained Application Protocol (CoAP) [71] is used in the application layer for enabling low-overhead secure RESTful interaction.

2.1.3 Formation of 6TiSCH Network

The formation process of a 6TiSCH network is initiated by its Join Registrar/Coordinator (JRC) (or the RPL root) by periodically broadcasting the EB frame. The EB frame contains the basic information of a network such as identification number of JRC, duration of a timeslot, number of timeslots in a slotframe, channel hopping sequence, and location of

shared cell. At the beginning, the pledges do not have such prior information (that an EB contains) to join a network, specifically, the time and channel where joined nodes transmit their control packets. Therefore, the pledges turn on their radios and randomly start scanning on all the available channels one by one to receive EB frame. The pledges switch their scanning channel from one channel to another channel after a specific amount of time and keep scanning until they receive valid EB. When a pledge receives a valid EB from an already joined node (it can be the JRC or any other joined nodes), the pledge becomes a TSCH synchronized node. If the pledge receives several EBs from different joined nodes, it selects the best joined node as its parent based on the value mentioned in the `Join Metric` field of the received EB frames. Now, the TSCH synchronized node knows the timeslot and the channel (i.e., `[slot offset, channel offset]`) where the joined node(s) transmit their control packets. Therefore, from now onward, the TSCH synchronized node keeps its radio active (i.e., either `Tx` or `Rx`) only in the shared timeslot to save energy. Now, in order to complete the joining process by the TSCH synchronized node, it needs to receive a valid DIO packet, which is *multicasted/unicasted* and the routing protocol i.e., RPL generates the DIO packet. The DIO packet contains all the necessary routing information required to join the DODAG routing tree constructed by the RPL for both upward and downward routing operations. However, before waiting for this DIO packet, the TSCH synchronized node exchanges JRQ and JRS control frames with the JRC via its parent node (if JRC is not the parent node) for secure enrollment RFC9031 [39], [44]. Once the secure enrollment is over, the TSCH synchronized node becomes a 6TiSCH joined node or RPL joined node after receiving a valid DIO packet. Now, the 6TiSCH joined node is eligible to transmit its control packets for further expansion of the network. The formation of the entire network is considered to be completed when all the pledges complete their joining process.

Journey of a pledge

New node/Pledge → *TSCH synchronized node* → *TSCH secured joined node* → *RPL/6TiSCH joined node.*

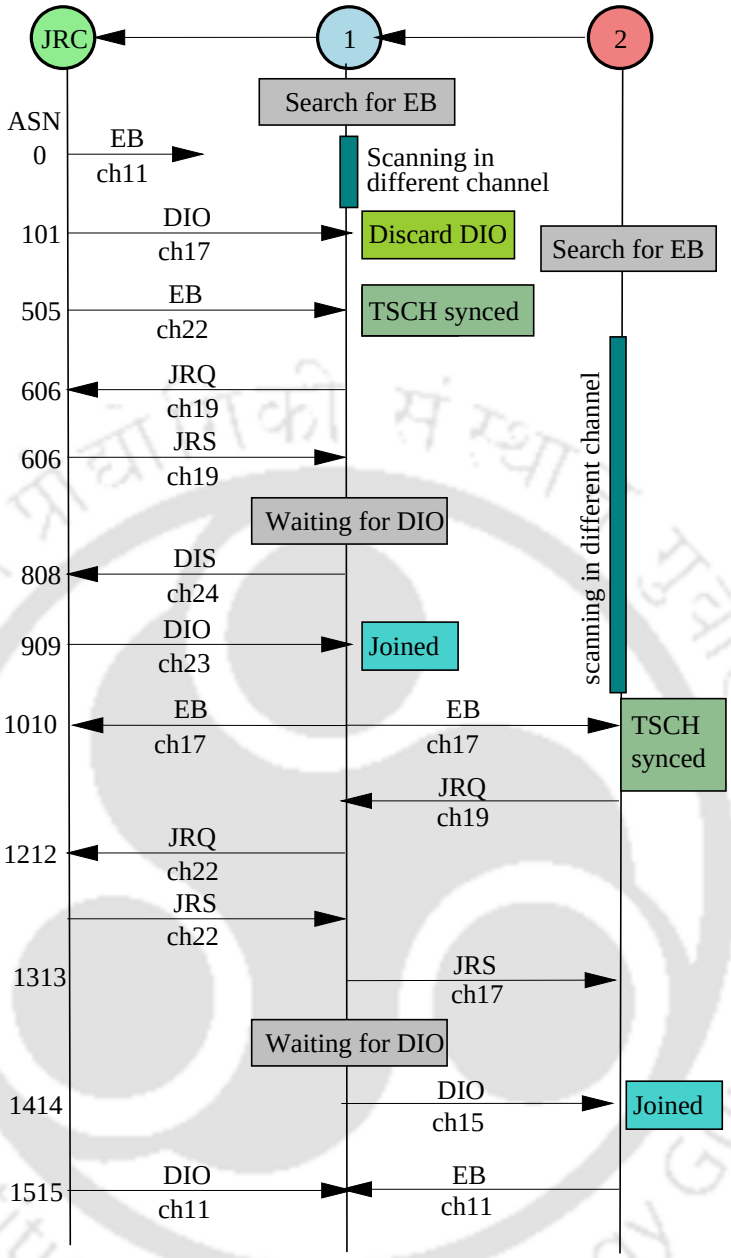


Figure 2.4: *6TiSCH network formation steps*

Figure 2.4 shows an example of the formation of the two-hop 6TiSCH network. At the beginning (i.e., at ASN 0), the JRC starts the formation process by broadcasting its first EB frame on channel 11. At that time, Node ① has also started scanning for EB frame to join the network. However, node ① is scanning on a different channel i.e., not on channel 11, and so it missed the EB frame broadcasted by the JRC. In the next slotframe, JRC broadcasts its first DIO packet. Note that the slotframe length is taken as 101 *timeslots*

in this example. Therefore, ASN becomes 101 in the next slotframe. Although node ① receives the DIO packet correctly, it has to discard the received DIO packet because it does not receive any valid EB yet. Node ① continues its searching for EB by performing random scanning on the channels, and it gets synchronized with the network when it receives a valid EB from the JRC on channel 22. It is important to note that from now onwards, node ① will keep its radio active only in the shared timeslot instead of keeping its radio all the time. In the meantime, node ② has also started scanning for EB by keeping its radio active all the time. Here, node ② is out of the communication range of JRC, so it needs to wait for node ① to complete its joining process. Node ① completes its joining process and becomes a 6TiSCH joined node once it has completed its secure enrollment process (ASN 606) and received a valid DIO packet (ASN 909) from JRC. Now, node ① is eligible to transmit its control packets for further expansion of the network. A node transmits DODAG Information Solicitation (DIS) packet to its parent node if it does not receive DIO packet for a long time i.e., to request for DIO packet. Similarly, node ② also joins the network after receiving both the EB (ASN 1010) and DIO packets (ASN 1414) from its parent node i.e., node ①. Node ② uses node ① as its join proxy to exchange its JRQ and JRS frame with the JRC. Subsequently, node ② has started broadcasting its control packet after joining the network successfully, and thus, the formation of the 6TiSCH network gets over.

2.2 Literature Survey

Researchers have separately considered the TSCH synchronization state and 6TiSCH joining state of the pledges during the formation of 6TiSCH networks. A pledge comes to know about the basic network configuration details (most importantly timeslot duration, slotframe length, channel hopping sequence) when it receives its first EB frame. After receiving its first EB, it knows the time and channel where next the control packets would be broadcasted. So, the pledge keeps its radio active only in the shared cell, which saves its energy. Otherwise, the pledge needs to keep its radio active all the time to receive an EB frame. Therefore, at the beginning, researchers tried to reduce this scanning time so that the energy consumption

of the pledges during their channel scanning could be reduced. This channel scanning time can be called as *TSCH synchronization time* because a pledge becomes a TSCH synchronized node after receiving an EB. Subsequently, to form a multi-hop 6TiSCH network, the TSCH synchronized nodes should get valid DIO packet from their respective parents to join the DODAG constructed by the RPL routing protocol. And after joining the DODAG, they become 6TiSCH joined nodes. Only the 6TiSCH joined nodes are allowed to transmit their control packet for further expansion of the multi-hop network. Furthermore, quick construction of DODAG also helps in efficient sensory data packet transmission [72] [73]. Therefore, it is important that the TSCH synchronized node should quickly join the DODAG constructed by RPL. Accordingly, researchers tried to reduce this 6TiSCH joining time of the TSCH synchronized nodes, so that multi-hop 6TiSCH network can be constructed in less time. Note that the construction of DODAG and the formation of 6TiSCH network happen together because a TSCH synchronized node becomes a 6TiSCH joined node after receiving DIO packet while the same DIO packet is used to form the DODAG.

Therefore, we divide the literature study on 6TiSCH network formation into two parts based on the prior research works on TSCH synchronization and 6TiSCH network formation. The works that considered only the TSCH network synchronization are included in one section. These works focused solely on the transmission of EB frames in order to create single-hop TSCH network. The works that considered the formation of a multi-hop 6TiSCH network, on the other hand, are described separately in another section. During the formation process, these works looked at the transmission of both EB frame and DIO packet. Hence, 6TiSCH network formation may be described as an extension of TSCH network formation in which nodes wait for DIO packets to join the DODAG created by the underlying RPL routing protocol. In the following sections, existing works on TSCH network synchronization and 6TiSCH network formation are described separately.

2.2.1 TSCH Network Synchronization

This section discusses the works that considered TSCH network synchronization, i.e., considered only the broadcasting of EB frame during the synchronization process. When a pledge receives its first EB frame, it becomes a TSCH synchronized node, and the time between when it starts scanning for EB frame and when it receives the first EB frame is referred to as the pledge's TSCH synchronization time.

Researchers considered different design characteristics while designing their proposed approaches for faster TSCH synchronization. Figure 2.5 shows the taxonomy of the existing TSCH network synchronization schemes, where researchers mainly considered different

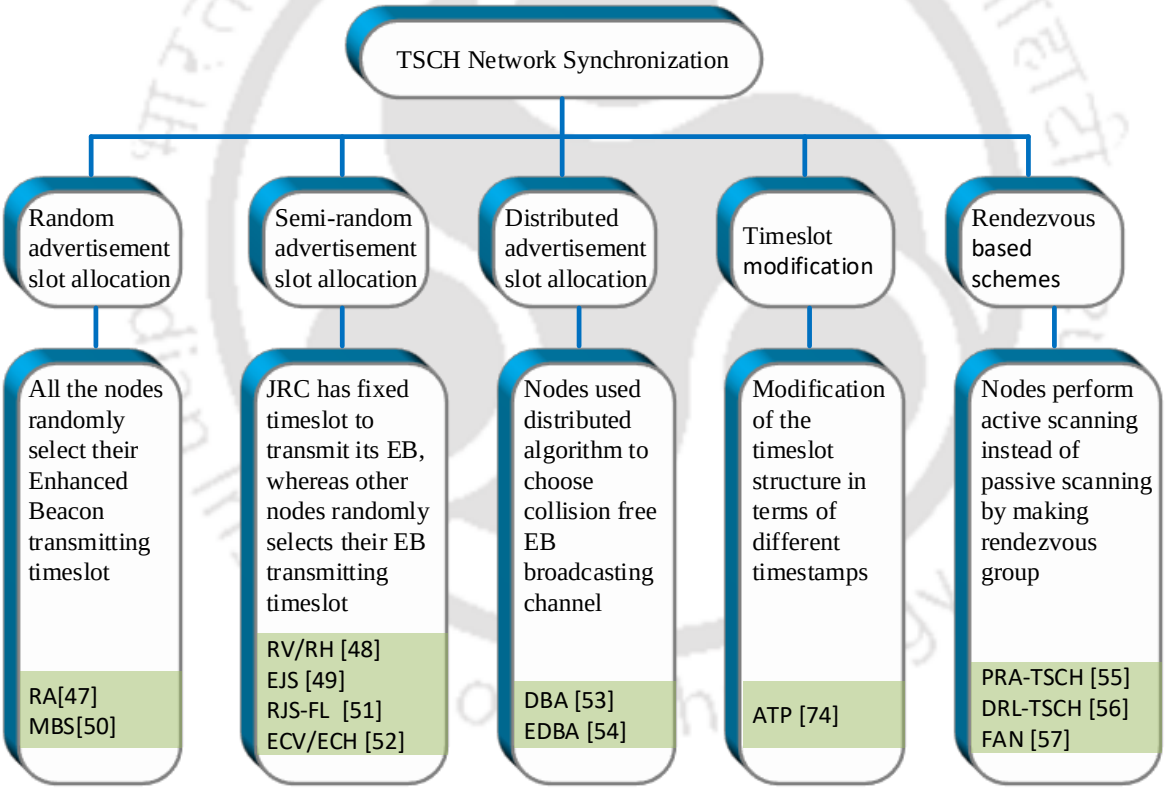


Figure 2.5: *Taxonomy of TSCH network synchronization schemes*

advertisement slot allocation policies such as random [47], [50], semi-random [48, 49, 51], and [52], deterministic [53], [54]. In random advertisement slot allocation, already synchronized nodes transmit their EB frame by randomly choosing any slot as their advertising slot. On the other hand, in semi-random allocation, fixed advertisement slot(s) are assigned

to the JRC, whereas, other synchronized nodes chose their advertisement slots randomly. In deterministic advertisement slot allocation, the synchronized nodes used to fix their advertisement slot. Apart from advertisement slot allocation, few works modified the advertisement timeslot format [74] and few other works proposed (e.g., [55–57]) rendezvous based active joining schemes for faster TSCH synchronization. All these existing works related to the TSCH network synchronization are briefly discussed one by one as follows.

Guglielmo *et. al* did the performance analysis of the formation process of TSCH network in [47]. The authors designed a Markov Chain-based analytical model to find the relation between the average synchronization time of the pledges and beacon transmission probability of the already synchronized nodes. Following that, the authors used a *random advertisement algorithm* (RA) based on their analytical model. The authors observed from their analytical analysis that the synchronization time of the pledges is inversely affected by the number of channels used for beacon transmission. Subsequently, the authors suggested using more number of channels to broadcast EB frame when more number of nodes are available in the network. The authors also mentioned that the overall synchronization time decreases when nodes increase in the network. Although the authors suggested various important points for improving the TSCH synchronization time, their work is limited to analytical analysis of the network synchronization process. Furthermore, their analytical model used only one channel for EB transmission. However, in the TSCH network, nodes use multiple channels to transmit their EB frame to get benefited from the availability of multiple channels, such as to get rid of interference, multi-path fading on a single channel, and improve the overall network throughput.

Vogli *et. al* proposed *random vertical* (RV) filling and *random horizontal* (RH) filling schemes for faster synchronization of TSCH networks in [48]. The authors used the term *multi-slotframe* (multi-SF), which is nothing but a collection of several slotframes together. Just like slotframe, multi-SF also repeats one after another. In RV, the coordinator (or JRC) transmits at the first slot of a multi-SF using the channel offset 0. On the other hand, other (new) synchronized nodes transmit using any random channel offset except the

channel offset 0. Therefore, the EB transmitted by the synchronized node(s) will never collide with the EB transmitted by the JRC. Note that using RV, all the nodes transmit their EBs simultaneously, i.e., at the first timeslot of a multi-SF. Therefore, as EBs are broadcasted using different channel offsets by different nodes simultaneously, the EB reception probabilities of the pledges also increase. However, as the nodes transmit their EBs in different channel offsets, this can lead to de-synchronization between parent-child pairs. On the other hand, in RH, all the nodes, including the JRC, use the same channel offset 0 but use different slotframes of a multi-SF to transmit their EBs. Here also, de-synchronization between the parent-child pairs is possible. Furthermore, collision is possible in both RV and RH schemes when two or more nodes choose the same channel offset and slotframe, respectively, to transmit their EBs. The authors designed probabilistic-based theoretical models for both the RV and RH schemes to estimate the synchronization time of the pledges. The designed analytical models were validated using simulation experiments by the authors. The results from both the theoretical and simulation experiments showed that the performance obtained by both RV and RH is the same. However, no comparison-based evaluation with any benchmark scheme was done for both schemes. The authors only showed the relation between the TSCH synchronization time and the number of synchronized nodes. Note that all the nodes in a network can either use the RV scheme or RH scheme.

The authors in [49] proposed a lightweight joining/synchronization scheme for TSCH networks to speed up the synchronization process. One slotframe is divided into two parts in their proposed approach, namely the advertisement plane and the communication plane. The advertisement plane contains the advertisement slot(s) where only the EB frame is broadcasted. On the other hand, the communication plane is reserved for transmitting the sensory data packet. The authors pre-configured the number of advertisement slot(s) at the beginning of their experiments and compared their scheme with the existing RV and RH schemes. The obtained results showed improvement over the RV and RH schemes. However, the authors did not provide any information about how many advertisement slots are needed for faster synchronization i.e., no scheme/method is provided to find out the

optimal value of advertisement slots for a particular network configuration. Rather, they pre-configured this value at the beginning of each experiment, which is static. Furthermore, the energy consumption of the nodes, throughput, and end-to-end packet delivery latency of the network get affected when more advertisement slots are used per slotframe. It is because the nodes need to keep their radios active in all the advertisement slots, and the communication slots are converted into advertisement slots. Furthermore, the authors did not consider the collision of packets in the advertisement slot while evaluating their scheme.

Guglielmo *et. al* [50] proposed a beacon scheduling algorithm based on their proposed discrete-time Markov Chain model for TSCH network synchronization. The authors described the behavior of a pledge during its network joining process using the proposed Markov Chain model, and accordingly, formulated an optimization problem from the proposed model. The optimization problem is used to minimize the average synchronization time by efficiently scheduling the EB transmission slot. At the end, the solution of the optimization problem is used in the *model-based beacon scheduling* (MBS) algorithm, which determines the unique beacon transmission link for every node. The authors analyzed their proposed MBS algorithm analytically, and results showed that MBS outperforms the existing schemes such as RD, RV, and RH in terms of average synchronization time. It is noteworthy that MBS behaves like RV when both MBS and RV use equal number of channels. On the other hand, MBS behaves like RH when both MBS and RH schemes use equal slotframe lengths. Hence, in some cases, all the schemes such as MBS, RV, and RH showed the same performance in terms of average synchronization time. However, increasing the number of EB transmission slots per slotframe in the MBS scheme can significantly affect the nodes' energy consumption. Furthermore, a large number of EB transmission slots per slotframe also affects the data transmission schedule. Additionally, the de-synchronization between the parent-child pairs is possible in MBS as the nodes use different channels to transmit their EB frames.

P. Duy *et. al* proposed a fuzzy logic-based adaptive beacon transmission scheme in [51] by extending their work EJS [49]. This scheme determines the EB rate of the TSCH syn-

chronized nodes adaptively depending on the number of available advertising nodes in their surroundings. The authors also took care of the energy consumption of the nodes as the EB rate is directly related to the energy consumption of the nodes. Like the EJS scheme, a slotframe is also divided into two parts, i.e., advertisement plane and communication plane, where advertisement slots are kept together in the advertisement plane. A synchronized node randomly chooses a channel for broadcasting its EB frame using a slot from the added advertisement slots. The purpose of fuzzy logic is to determine the required number of advertisement slots per slotframe depending on the node density. The fuzzy logic takes the number of synchronized nodes as input and provides the number of advertisement slots per slotframe as output. The theoretical and real hardware-based experimental results showed that RJS-FL improves the network synchronization time compared to the RV scheme. However, the idea of a random selection of advertisement slots can lead to collision.

Vogli *et. al* extended their previous work [48] in [52], where authors proposed enhanced version of their previous approaches i.e., random vertical (RV) and random horizontal (RH), and named them as *enhanced random vertical* (ECV) and *enhanced random horizontal* (ECH). As mentioned in RV, the JRC uses the channel offset 0, whereas other synchronized nodes use different channel offsets, and all the nodes transmit at the same timeslot i.e., at the first timeslot of the first slotframe of a multi-SF. However, when the number of synchronized nodes exceeds the total number of channel offsets (at max. 16 channel offsets are allowed), there will be collision in EB transmission as more than one node needs to choose the same channel offset. The authors proposed the ECV scheme to mitigate this problem, where a node can choose any channel offset other than channel offset 0 from the next slotframe of a multi-SF when the previous slotframe gets filled. Furthermore, the JRC can transmit its EB frame in every slotframe of a multi-SF using the channel offset 0, considering it has enough power source. On the other hand, in ECH, the nodes use the next channel offset when there is no free slotframe available in the multi-SF using the previous channel offset. Thus, both the ECV and ECH reduced the probability of EB collision by using different channel offsets and slotframes in dense networks. The authors designed analytical models

for the ECV and ECH and compared their analytical results with the experimental results performed using real hardware devices. The results showed the TSCH synchronization time of the nodes as a function of number of synchronized nodes. Although both ECV and ECH improve the performance of TSCH network formation, these two schemes also suffer from de-synchronization issues between the parent-child pairs, and high energy consumption of the nodes. It is not always true for every IoT application that JRC has enough energy source. Furthermore, the network's throughput and latency are also affected as many advertisement slots are used in both the ECV and ECH schemes.

Karalis *et. al* proposed a technique i.e., *advertisement timeslot partitioning* (ATP) to transmit more number of EB frames without allocating extra advertisement slots per slotframe in [74]. The EB control frame is generally broadcasted in the advertisement slot, where the receiver does not transmit an acknowledgment (ACK) packet after correctly receiving an EB frame, nor the transmitter waits for it. As the TSCH timeslot length is long enough to transmit a packet and receive its ACK, half of the total timeslot duration gets wasted when EB frame is transmitted. Therefore, ATP divides a timeslot into two parts and utilizes both the parts for advertising EB frame instead of allocating extra advertisement slots per slotframe. Hence, ATP can broadcast two EB frames in a single advertisement slot, unlike the other schemes. Thus, ATP improves the EB transmission rate in the network without increasing the number of advertisement slots per slotframe. The authors performed simulation experiments on a python-based simulator, and the obtained results showed that ATP can improve the synchronization time of the nodes. However, in real-life scenarios, some control packets require ACK such as unicast DIS request and keep-alive. Therefore, ATP becomes tough to implement in the presence of those control packets as they also need to be transmitted in the advertisement slot for constructing multi-hop 6TiSCH networks.

The authors Khoufi *et. al* proposed *deterministic beacon advertisement* (DBA) algorithm for TSCH networks in [53]. This work reduces the collision of the beacons where the transmitting nodes are not neighbor to each other. DBA distributes all the available channels among the nodes so that deterministic channel access is possible for the nodes to

transmit their beacons. DBA finds the unique channels for all the nodes and allows the nodes to transmit their EBs in the regularly spaced advertisement slots using their assigned channel. Therefore, the advertisement nodes calculate their unique channel offset during their association using the DBA scheme to achieve collision-free beacon broadcasting. The simulation results showed that the DBA outperforms the RV and RH in terms of joining time. The results considered the TSCH synchronization time of different network topologies as a function of number of beacon transmitting nodes and beacon transmitting interval.

Khoufi *et. al* further improved their previous work DBA [53] in [54]. Authors proposed *enhanced deterministic beacon advertising scheme* (EDBA) for collision-free EB broadcasting. EDBA transmits beacons on all the available channels without any collision. The authors developed a Markov Chain based analytical model for comparing EDBA with MBS [50] and also performed simulation experiments. Results showed that EDBA outperforms MBS in terms of the joining time of the nodes. However, no clear distinction is visible between the DBA and EDBA schemes. Both DBA and EDBA used the same method to add collision-free advertisement slots per slotframe.

Algora *et. al* proposed *parallel rendezvous-based association for TSCH networks* (PRA-TSCH) scheme in [55] to reduce the random channel scanning time of the pledges during their TSCH association. Unlike the previously mentioned works, PRA-TSCH allows the pledges to perform active scanning for EB frame instead of always waiting (i.e., scanning) for EB frame on random channels. It allows the non-associated nodes to turn up for their rendezvous during their channel scanning process. The non-associated (i.e., pledge) nodes form a network among themselves by exchanging *rendezvous beacon* (RB) frames. When a pledge receives an EB frame from an associated node (i.e., already synchronized), the pledge broadcasts the basic information carried by the EB frame among the other non-associated nodes, which helps the other pledges to join the network quickly. As the pledges in the rendezvous group scan on different channels and share the basic network configuration information among the nodes when they receive EB frame, it reduces the scanning time of the pledges in the network. Both theoretical and simulation experimental results showed that

PRA-TSCH scheme improves the TSCH association (or synchronization) time compared to the 6TiSCH-MC standard [40].

However, PRA-TSCH scheme may affect the existing TSCH schedule performed by the already associated/synchronized nodes for control and data packet transmission. Furthermore, this scheme could lead to the higher energy consumption of the nodes by unnecessarily broadcasting RB frames in sparse networks. On the other hand, collisions among the packets transmitted by the associated and non-associated nodes are possible in dense networks. Apart from this, there is a possibility that the non-associated nodes miss the EB frame broadcasted by the already associated nodes while gossiping among themselves in the rendezvous group.

The authors Byeong-Hwan Bae and Sang-Hwa Chung proposed the *distributed radio listening* (DRL-TSCH) scheme in [56] to form the TSCH networks quickly. The authors used a similar concept of exchanging rendezvous beacon (RB) among the non-associated nodes as mentioned in PRA-TSCH [55]. However, in the DRL–TSCH scheme, nodes also responded to RB frames by mentioning their own channel scanning sequences (i.e., their channel index, channel array, and ID), so that a node does not follow the same configuration used by the other nodes. As a result, all the channels are sub-divided among the pledges, which creates the rendezvous chain. Hence, the EB receiving probability of the pledges increases in DRL-TSCH compared to the PRA-TSCH [55], which allows random scanning. In PRA-TSCH, scanning on random channels may result in two or more nodes scanning on a single channel at the same time, which reduces the EB reception probability. The simulation results showed that DRL-TSCH outperforms the PRA-TSCH and 6TiSCH-MC in terms of TSCH association/synchronization time. However, the mentioned problems related to PRA-TSCH may also be possible in DRL–TSCH.

Recently, Mohamadi *et. al* proposed *fast and active network formation* (FAN) scheme for faster TSCH synchronization in [57]. Following the works in [55] and [56], FAN also allows the pledges to transmit *enhanced beacon requests* (EBRs) to accelerate the (re)association process. However, unlike PRA-TSCH and DRL-TSCH, FAN took care of the collision-free

EBRs broadcasting. This collision-free method avoids collisions between EBRs transmitted by pledges and underlying TSCH schedule running by the already associated nodes for both the EB and data packet transmission. Apart from this, FAN also proposed to use trickle-based EB transmission instead of using periodic EB transmission. This trickle-based EB transmission strategy allows the nodes to change their EB transmission rate depending on the current network condition. However, EBRs may cause a large number of EB transmission in the network because of this trickle-based EB transmission rate. The authors performed simulation-based experiments to evaluate FAN performance in terms of joining time as a function of number of EB transmitting nodes, number of used channels, and channel packet delivery rate. Results showed that FAN outperforms the RV, RH schemes.

The authors Algora *et. al* designed a theoretical model for the nodes' joining process in TSCH network in [75]. The model estimates the time required by a pledge to get associated with a synchronized node. The authors also considered possible link failures among the nodes during network formation in their theoretical model. Basically, their proposed theoretical model finds out the upper bound time to form a network with an arbitrary topology. From the analytical analysis and simulation experiments, the authors concluded that the formation of a TSCH network greatly depends on the number of channels used for communication and the used EB transmission interval by nodes in the network. In the future work plan, the authors mentioned that they would try to design a perfect channel selection mechanism for collision-free EB transmission and try to find out the optimal EB transmission interval depending on the total running time and/or the stability of the network topologies.

Finally, Table 2.1 summarizes all the above mentioned works related to TSCH network synchronization considering different parameters used in designing efficient TSCH network synchronization schemes. None of these existing works considered the formation of multi-hop 6TiSCH network (i.e., considered the transmission of EB frame only) and synchronization issue between the parent-child pair. Furthermore, most of the works used more number of advertisement slots per slotframe to transmit EB frame which is not 6TiSCH-MC compliant and consumes huge energy.

Table 2.1: *Comparison among the existing works on TSCH network synchronization*

Schemes	A	B	C	D	E	F	G	H	I
RA [47]	1/EB cycle	NIA	NIA	NIA	All	No	No	Passive	NIA
RV & RH [48]	1/Multi-SF	RV: 1 RH: >1	RV: 1^{st} slot of multi-SF RH: 1^{st} slot of each SF	Random	RV: Random RH: 0	Yes	No	Passive	NIA
EJS [49]	Implementation specific	>1	Beginning of a SF	Pre-configured	All	Yes	No	Passive	RV
MBS [50]	Implementation specific	>1	Anywhere	Random	Any	No	No	Passive	RV & RH
RJS-FL [51]	1/Multi-SF	>1	Beginning of a SF	Random	All	Yes	No	Passive	RV
ECV & ECH [52]	1/Multi-SF	ECV: >1 ECH: >1	ECV: 1^{st} slot of each SF ECH: 1^{st} slot of each SF	Random	ECV: All ECH: All	Yes	No	Passive	RV & RH
ATP [74]	Implementation specific	NIA	Implementation specific	Pre-configured	NIA	No	Yes	Passive	NIA
DBA [53]	NIA	>1	Anywhere	Distributed	All	No	No	Passive	RV & RH
EDBA [54]	NIA	>1	Anywhere	Distributed	All	No	No	Passive	MBS
PRA-TSCH [55]	1/SF	1	slot offset 0 channel offset 0	Centralized (6TiSCH-MC)	0	No	No	Active	6TiSCH-MC
DRL-TSCH [56]	1/SF	1	slot offset 0 channel offset 0	Centralized (6TiSCH-MC)	0	No	No	Active	6TiSCH-MC PRA-TSCH
FAN [57]	1/SF	1	slot offset 0 channel offset 0	Centralized (6TiSCH-MC)	0	No	No	Active	6TiSCH-MC RV & RH

*A: EB transmission rate, B: Number of advertisement slot, C: Advertisement slot position, D:Resource Allocation Type, E: Channel offset used for advertisement, F: Modify slotframe (SF), G: Modify timeslot, H: EB scanning type, I: Improved Over; ◇NIA: No information available; *SF: Slotframe length*

2.2.2 6TiSCH Network Formation

The works that considered multi-hop 6TiSCH network formation are briefly described in this section. These works considered that the pledges should receive both EB frame and DIO packet to join in multi-hop 6TiSCH network. Furthermore, the pledges require these two control packets in order to select their best parents and maintain a stable 6TiSCH network. After receiving its first EB, a pledge becomes a TSCH synchronized node, and when the TSCH synchronized node receives a valid DIO packet from its parent, it becomes a 6TiSCH joined node. Subsequently, a network is said to be completely formed when all the pledges present in the network become 6TiSCH joined nodes, which means after receiving both the control packets i.e., EB and DIO and exchanging JRQ and JRS frames with the JRC.

The 6TiSCH WG published the 6TiSCH minimal configuration (6TiSCH-MC) standard [40] for providing the information about minimum resource allocation during multi-hop 6TiSCH network formation. This standard mentioned three important points which need to be followed by the pledges and already joined nodes during the formation of 6TiSCH networks. These are: (i) all types of control packets (such as EB, DIO, DIS, keep-alive)

should be broadcasted/unicasted in shared cell. Control packets are not transmitted in the manageable cell. Only data packets are transmitted in the manageable cell. As the transmission of control packets happens only in shared cells, nodes need to access the shared cell first before transmitting their control packets. Therefore, nodes perform TSCH CSMA/CA channel access mechanism to access the channel associated with the shared cell. The main difference between the traditional CSMA/CA and TSCH CSMA/CA is in setting the *backoff* timer by a node when it finds that the shared channel is busy using Clear Channel Assessment (CCA). The backoff timer is set in terms of number of shared cells instead of continuous time period in TSCH CSMA/CA. (ii) This is related to the number of shared cell per slotframe for transmitting all the generated control packets by all the nodes. The standard mentioned to use only one shared cell per slotframe for transmitting the control packets. Note that this static allocation is made by the standard irrespective of the slotframe length. The default location of the shared cell is at `slot offset=0, channel offset=0` for all the nodes present in multi-hop 6TiSCH networks as shown in Figure 2.6. Therefore, this common shared cell helps the nodes get tightly synchronized within the

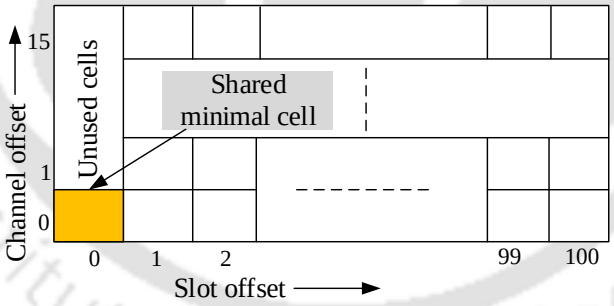


Figure 2.6: *Resource allocation by 6TiSCH-MC*

networks. (iii) The last but not the least important point mentioned by the standard for network formation is that a pledge should receive both the EB and DIO control packet to complete its joining process, and a pledge is allowed to transmit its control packets for further expansion of the network only after becoming a 6TiSCH joined node. Otherwise, according to 6TiSCH-MC, the network becomes unstable, and the overall formation process of the network gets delayed, which further affects the energy consumption of the nodes. After 6TiSCH-MC, two schemes have been proposed by different researchers considering

different design characteristics. The works in [45] increased the number of shared cells per slotframe to transmit the generated control packets quickly, whereas, the work in [46] considered fixed EB transmission probability per slotframe. The summary of these existing works are described below,

Vallati *et. al* [45] improved the 6TiSCH formation process by allocating more number of shared cells per slotframe unlike the 6TiSCH-MC standard, which considered only one shared cell per slotframe. At the beginning, the authors designed an analytical model for analyzing the formation of 6TiSCH network considering the similar resource allocation as 6TiSCH-MC. Using the analytical results, the authors proved that 6TiSCH-MC does not provide enough resources i.e., number of shared cells, to transmit all the generated control packets during the formation of 6TiSCH networks. The authors also performed simulation experiments on a 7×7 grid topology to validate their statement on 6TiSCH-MC's resource allocation. From both the analytical and simulation, the authors further concluded that because of the resource scarcity, pledges take more time to join the network and consume huge energy. Therefore, the authors proposed the *dynamic resource allocation* (DRA) scheme to deal with this insufficient resource allocation problem. Basically, DRA increases and allocates more than one shared cells per slotframe as shown in Figure 2.7, and this allocation is done depending on the number of control packets generated in the network.

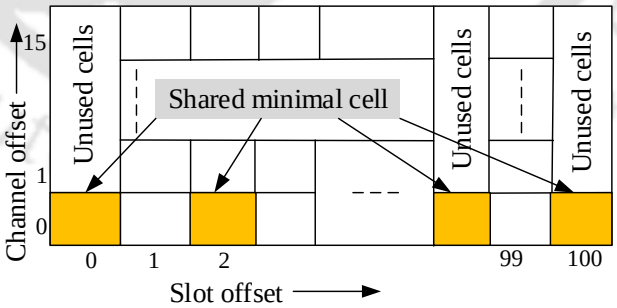


Figure 2.7: *Resource allocation by DRA*

So, unlike 6TiSCH-MC, DRA allocates the more than one shared cells dynamically when the number of control packet generated in the network is more. To add as many shared cells as the number of control packets generated in their surrounding, all the nodes exchange the information about their buffered control packets in their EBs. Accordingly, every node

calculates the required number of shared cells to transmit all the generated control packets in their surroundings. Finally, the nodes set the location of the added shared cells without any external signaling overhead by calculating the distance between two shared cells by taking the logarithmic value of the total number of added shared cells. Note that the number of allocated shared cells is taken as a power of 2 to ease the dynamic allocation/deallocation. The authors mentioned that their proposed scheme significantly improves the formation time compared to 6TiSCH-MC by performing real testbed experiments. Also, the authors mentioned that their proposed scheme creates stable networks compared to the networks created by 6TiSCH-MC.

However, this scheme has several drawbacks. This scheme converts the dedicated cells (or manageable cells) into shared cells, making more unusable cells per slotframe. It is because the nodes need to keep their radios active in the shared cells. Hence, the rest of the channel offsets become unusable apart from the channel offsets associated with the allocated shared cells. This decreases the network efficiency in terms of throughput and end-to-end latency. Furthermore, dedicated cells need to be rescheduled again because there is a possibility that some of the dedicated cells and shared cells are scheduled at the same timeslot. Apart from this, using this scheme, the *radio duty cycle* of the nodes increases as they need to keep their radios active in all the allocated shared cells, which causes more energy consumption. Note that the radio duty cycle denotes the amount of time a node keeps its radio active per unit. Again, nodes add shared cells based on the information received from their neighbors' EB. Therefore, if a node does not receive an EB frame from its neighbor due to collision or corruption during the transmission of their EB, the node might not receive important information transmitted by the neighbor nodes. Thus, this can result in an unstable network topology.

Vucinic *et.al* [46] performed simulation experiments on the formation of multi-hop 6TiSCH networks considering the same resource allocation policy stated by 6TiSCH-MC i.e., only one shared cell per slotframe. From the simulation experiments, the authors mentioned that the EB transmission probability per slotframe should be 0.1 in order to reduce the

Table 2.2: *Existing works in 6TiSCH network formation*

Schemes	EB transmission rate	Number of shared cell	Shared cell location	Resource Allocation Type	Dynamic congestion control	Provide fair transmission	Provide sufficient DIO	Don't disturb manageable cell	Synchronized utilization of all channels
RA [47]	EB period	NIA	NIA	NIA	No	No	No	No	No
RV & RH [48]	1/Multi-SF	RV: 1 RH: >1	RV: 1^{st} slot of multi-SF RH: 1^{st} slot of each SF	Random	No	No	No	No	No
EJS [49]	Implementation specific	>1	Beginning of a SF	Pre-configured	No	No	No	No	No
MBS [50]	Implementation specific	>1	Anywhere	Random	No	No	No	No	No
RJS-FL [51]	1/Multi-SF	>1	Beginning of a SF	Random	No	No	No	No	No
ECV & ECH [52]	1/Multi-SF	ECV: >1 ECH: >1	ECV: 1^{st} slot of each SF ECH: 1^{st} slot of each SF	Random	No	No	No	No	No
ATP [74]	Implementation specific	NIA	Implementation specific	Pre-configured	No	No	No	Yes	No
DBA [53]	NIA	>1	Anywhere	Distributed	No	No	No	No	No
EDBA [54]	NIA	>1	Anywhere	Distributed	No	No	No	No	No
PRA-TSCH [55]	1/SF	1	slot offset 0 channel offset 0	Centralized (6TiSCH-MC)	No	No	No	No	No
DRL-TSCH [56]	1/SF	1	slot offset 0 channel offset 0	Centralized (6TiSCH-MC)	No	No	No	No	No
FAN [57]	1/SF	1	slot offset 0 channel offset 0	Centralized (6TiSCH-MC)	No	No	No	No	No
DRA [45]	EB period	Dynamic	slot offset Dynamic channel offset 0	Distributed	Yes	No	No	No	No
BS [46]	1/10*EB period	1	slot offset 0 channel offset 0	Centralized (6TiSCH-MC)	No	No	No	Yes	No
This thesis	Dynamic, EB period	1	slot offset 0 channel offset 0, All	Centralized, Autonomous	Yes	Yes	Yes	Yes	Yes

°NIA: No information available; *SF: Slotframe length

congestion in shared cell. This EB broadcasting strategy is set irrespective of the number of nodes and node density in the networks. Furthermore, the authors considered the DIO transmission probability per slotframe is 0.3 in their simulation experiments. Nevertheless, the DIO transmission rate is governed by the Trickle Algorithm [76], which increases the transmission rate of DIO packet when there is an inconsistency in the networks. Note that during the formation of the networks, the nodes frequently transmit DIS request packet, which forces the Trickle algorithm to transmit more DIO packets in less time. This results in congestion in the shared cell. Therefore, in RPL-based 6TiSCH network, inaccurate formation time estimation is possible when a fixed DIO transmission rate is considered to form the networks. Moreover, the pledges need to wait for more time to receive EB frame when the EB transmission interval increases. This waiting time is more when only few nodes are deployed i.e., in sparse network topology. As a result, this increases the TSCH synchronization time of the nodes, and so their energy consumption. Subsequently, it increases the overall formation time of multi-hop networks. Therefore, the idea of using a fixed and low EB rate all the time in order to reduce congestion in shared cell is not always fruitful for all kinds of networks.

Finally, Table 2.2 briefly summarizes the existing works related to one-hop TSCH synchronization and multi-hop 6TiSCH network formation and compared them with the works done in this dissertation. As stated above, the existing works related to TSCH synchronization did not consider the formation of multi-hop 6TiSCH network and synchronization issue, and used more number of advertisement slots per slotframe. On the other hand, the existing works related to 6TiSCH network formation either used more number of shared cells per slotframe or did not consider the congestion in shared cell due to higher transmission rate of different control packets. Furthermore, none of these existing works looked into the fair transmission of control packets as well as fair transmission by the nodes, usage of all the physical channels at a time for control packet transmission. Therefore, these existing issues and unexplored research directions motivate us to design efficient scheme for faster formation of 6TiSCH network.

2.3 Experimental Environments

There are several major embedded operating systems (OS) available such as TinyOS[1] [77], Contiki [2] [78], RIOT[3] [79], FreeRTOS[4] for resource-constrained IoT devices, which run on limited amounts of memory and processing power. These OSs provide efficient resource allocation, preemptive or cooperative scheduling, and follow the modularity-based implementation. Their distinct modules are implemented for different functions, which helps in providing interoperability with heterogeneous hardware, enables less effort in reprogramming or making changes in the individual modules.

In this dissertation, we use the current version of Contiki i.e., Contiki Next Generation (Contiki-NG), to implement and evaluate our contributions. We use the Cooja simulator for simulation-based evaluation of few of our contributions. The Cooja Simulator is a network simulator, designed explicitly for IoT networks and freely distributed with Contiki/Contiki-

[1] http://www.tinyos.net/
[2] https://www.contiki-ng.org/
[3] https://www.riot-os.org/
[4] https://www.freertos.org/

NG OS. In order to perform testbed experiments, we use the devices (IoT nodes) that are deployed in the FIT IoT-LAB testbed. We briefly discuss the Contiki-NG and FIT IoT-LAB in the below subsections.

2.3.1 Contiki-NG Operating System

Contiki-NG is an open-source, cross-platform, memory-efficient OS for Next-Generation IoT devices. Its source code[1] is available on GitHub under the terms of the 3-clause BSD license, with a vast community of contributors and forks. Adam Dunkels first started the Contiki in 2002 [78], and many developers from research institutions and major industries are further developing it. Note that Contiki-NG started as a fork of the Contiki and now retains some of its own features. Contiki-NG focuses on low-power communication and standard protocols such as IPv6/ 6LoWPAN, RPL, and CoAP, including 6TiSCH. In brief, it supports the protocols standardized by IETF for low-power IPv6 connectivity in the IoT.

Contiki-NG is designed to run on resource-constrained IoT hardware devices limited in memory, power, and processing capacity. To fulfill the requirements of these resource-constrained devices, Contiki-NG requires approximately 38 kB of ROM and 11 kB of RAM for running a full IPv6 stack with 6LoWPAN and RPL. Contiki-NG also provides mechanisms to run the devices in low-power mode to reduce the power consumption of devices on which it runs. In brief, it consumes power in milliwatts.

Contiki-NG programming model uses protothreads to execute effectively on low-memory devices. A protothread is a memory-efficient programming concept that combines multi-threading and event-driven programming to execute a program with low memory. The concept of protothreads was developed within the context of Contiki in response to internal (i.e., time-based event) or external events (i.e., sensor-based event). Protothreads are scheduled cooperatively, such as a Contiki process must explicitly give control back to the kernel at regular intervals. In brief, protothreading is a way to code that allows the system to run other activities when the code is waiting for something to happen.

[1]https://github.com/contiki-ng/contiki-ng

Contiki-NG offers two main communication stacks in terms of networking: i) the μIP stack, which is a IETF-compliant TCP/IP stack for both IPv4 and IPv6 connectivity, and (ii) the Rime stack, which can be used instead of IPv4 or IPv6 stacks to reduce overhead. Contiki-NG supports a variety of channel access mechanisms at the MAC layer such as CSMA/CA, Contiki-MAC, TSCH, and Bluetooth Low Energy (BLE).

Contiki-NG also freely distributes a java-based IoT network simulator named Cooja, which simulates Contiki nodes. Cooja simulator helps the developers to test applications or protocols before testing them on real hardware platforms. In brief, with the Cooja simulator, Contiki networks can be emulated before being burned into hardware. It also provides a graphical user interface by which users can set network topology as per his/her requirements, set the radio propagation model and radio interference and communication range, visualize the serial line output of each node, nodes' communication activities in different network protocol layers, and their power consumption.

In Chapters 3 and 4, we use the Cooja simulator to validate our contributions.

2.3.2 FIT IoT-LAB testbed

FIT IoT-LAB[1] [80] is an open-source IoT lab that provides a very large-scale infrastructure suitable for testing small wireless sensor devices and heterogeneous communicating objects. IoT-LAB allows researchers to monitor nodes' energy usage and network-related metrics such as end-to-end delay, throughput, and overhead by giving them complete control over the nodes and direct access to the gateways to which nodes are connected. It facilitates rapid experiment placement and simple evaluation, data collection, and analysis. In brief, IoT-LAB provides a testing environment of multi-platform, multi-radio, multi-topology, and multi-OS based configurable IoT networks.

IoT-LAB testbeds are located at six different sites across France, giving access to 1786 wireless sensors nodes when this thesis is written. The nodes include `WSN430`, `M3`, `A8`, to name a few. All the testbeds are equipped with various node types, topologies, and settings

[1] https://www.iot-lab.info/

that can be configured to support a wide range of real-world use-cases of IoT networks. The testbeds are accessed through a web portal or using provided command-line tools. In addition to it, it also allows access to devices' serial ports via SSH front-ends.

Although due to licensing compatibility issues on hardware drivers, Contiki-NG official release does not support IoT-LAB boards, IoT-LAB releases the repository[1] to bring support for the IoT-LAB platform. Basically, this repository is a port of the official Contiki-NG repository and supports two boards i.e., `M3` and `A8-M3` that are available in IoT-LAB testbeds. The changes (e.g., modification in TSCH behavior, RPL, etc.,) should be done inside the respective Contiki-NG directories as per the user requirements. We provide more information about our experimental setup in Appendix A of this thesis.

All of our contributions are evaluated using the `M3` nodes with different network topologies such as grid, linear, tree, and with various network configurable parameters in different FIT IoT-LAB testbeds.

2.4 Summary

It can be seen that most of the TSCH network synchronization schemes reduced the synchronization time by scheduling the EB frame, increasing the number of advertisement slots per slotframe, and increasing the EB transmission rate. However, each of these three methods has significant drawbacks. Firstly, as each node transmit its EB frame using different channel at the same time to make collision-free broadcasting, the parent-child pair nodes might lose the synchronization to each other while transmitting their EB frames. No prior works mentioned this issue, and so did not overcome it. De-synchronization is an inevitable problem when nodes do not maintain communication for a long time. Secondly, adding more advertisement slots per slotframe affects the data communication cell, hindering network performance in terms of throughput and end-to-end delay. However, TSCH is designed to provide deterministic delay-bounded reliable communication to the applications using it. Furthermore, collision is possible in the advertisement slots when the nodes choose their

[1] https://github.com/iot-lab/iot-lab-contiki-ng

advertisement slots randomly. Thirdly, the ill effect of transmitting more EB frames by a node directly fall upon node's energy consumption, which further affects the network lifetime. Although the rendezvous based association schemes (i.e., [55–57]) look promising, the synchronization between the parent-child pairs needs to be addressed. Furthermore, the energy consumption of the nodes needs to be taken care of while the nodes transmit their EB request frame (or probing for EB frame) during their association process.

On the other hand, only a few works have been published related to 6TiSCH network formation. The work in [45] significantly improved the performance of 6TiSCH network during its formation, but it severely affects the data transmission cells. In total ((*number of channels used in a network -1)* $\times$ *number of allocated shared cells*) number of cells remain unused in each slotframe, which significantly impacts the network performance in terms of throughput and end-to-end packet latency. Although the work in [46] did not hamper the data transmission schedule, so, compliant to 6TiSCH-MC, the fixed EB transmitting probability might not be a good setting for all type network topologies. Indeed, the existing works did not explore the effect of EB and DIO generation/transmission rate, priority of control packets, uncertainty in accessing the shared cell, fair transmission of control packet by the nodes, usages of multiple channels at a time for control packet transmission during the formation of multi-hop 6TiSCH network. Accordingly, this thesis proposes a set of amendments over the 6TiSCH-MC standard to achieve the faster formation of 6TiSCH network. To begin with, in the next chapter, a dynamic EB generation scheme is provided to reduce congestion in shared cell, and so, improve the performance of 6TiSCH network formation following the similar resource allocation made by 6TiSCH-MC.

"Don't stop chasing your dreams, because dreams do come true."

~Sachin Tendulkar (1973)

3

Channel Condition based Dynamic Beacon Interval

In this chapter, we discuss the first contribution to improve the performance of 6TiSCH network during its formation by reducing congestion in shared cell. For this, we propose a *channel condition based dynamic beacon interval* (C2DBI) scheme to alleviate congestion in shared cells. In brief, C2DBI varies the joined nodes' beacon generation interval depending on the shared cell congestion status instead of using a fixed beacon generation interval all the time. The proposed work C2DBI is evaluated with Markov Chain based theoretical analysis, simulation, and testbed experiments.

3.1 Introduction

As discuss in Chapter 2, Vallati *et al.* [45] proposed an efficient multi-hop 6TiSCH network formation scheme considering the transmission of both EB frame and DIO control packet during network formation. The authors proposed a dynamic resource allocation scheme to provide enough resources i.e., shared cells per slotframe to the nodes for transmitting their control packets. On the other hand, to reduce the congestion in shared cells, Vucinic *et al.* [46] proposed to fix the EB transmission probability to 0.1 irrespective of the number of nodes and node density in the network for faster network formation. However, it is observed that allocation of extra shared cells per slotframe like the work in [45] costs in higher energy consumption, and severely affects the data transmission schedule. On the other hand, fixed EB transmission probability is not suitable for all type of network topologies. Apart from these two works, the works in [47–57] considered only the initial TSCH synchronization of 6TiSCH network. They did not consider the formation of multi-hop 6TiSCH network. The key focus of these works was to transmit as many EB as possible by either allocating more shared cells per slotframe or increasing the EB transmission rate for the faster synchronization of single-hop TSCH network. However, both these approaches either consume more bandwidth, and so affects the data transmission schedule or increases the energy consumption of the nodes.

Basically, both [45] and [46] considered the shared cell congestion issue which occurs due to limited resource allocation by 6TiSCH-MC and provided different schemes to deal with it. To understand the effect of resource allocation by 6TiSCH-MC on 6TiSCH network formation, at the outset, we design a Markov Chain based analytical model for estimating nodes' joining time and energy consumption in multi-hop 6TiSCH networks. Our analytical results show that the performance of network formation protocol (i.e., 6TiSCH-MC) degrades in terms of nodes' joining time and energy consumption due to fixed beacon generation intervals of the nodes. When the number of joined nodes increases in the network, the shared cells get congested due to their fixed beacon generation interval. And finally, due to the increasing congestion in shared cell, the pledges need to wait for longer to receive different

control packets, resulting in nodes' longer joining time and higher energy consumption. On the other hand, beacon transmission is essential to expand or reorganize the present network topology. To overcome this performance trade-off, we propose *channel condition based dynamic beacon interval (C2DBI)* scheme for improving the formation of 6TiSCH network. In the proposed scheme, the beacon generation interval of the joined nodes is varied based on the shared cell congestion status. In brief, in this work, the main target is to achieve an energy-efficient scheme for faster association of the pledges into the existing 6TiSCH network without effecting the existing data transmission schedule and increasing nodes' energy consumption. At first, the theoretical analysis of C2DBI shows the advantages of dynamic beacon generation interval over fixed beacon generation interval in terms of nodes' joining time and energy consumption. Finally, we implement the proposed scheme C2DBI on Contiki-NG and validate it by performing simulation and testbed experiments on Cooja simulator and FIT IoT-LAB, respectively.

We summarize the contributions of this chapter as follows,

- A Markov chain model is provided for analyzing the node joining process in a multihop 6TiSCH network.
- Analytical results show the demerits of fixed beacon generation interval scheme during network formation with respect to node joining time and energy consumption.
- A *channel condition based dynamic beacon interval (C2DBI)* scheme is proposed for efficient joining of nodes in 6TiSCH network based on shared cell congestion status.
- Theoretical analysis of the proposed scheme is provided to compute nodes' joining time and energy consumption.
- Comparison based simulations on Cooja simulator and testbed experiments on FIT IoT-LAB are conducted to validate the proposed scheme.

The rest of the chapter is organized as follows. In Section 3.2, we provide an analytical model of 6TiSCH-MC considering multi-hop 6TiSCH network. In Section 3.3, we present

our proposed channel condition based dynamic beacon interval scheme. Subsequently, in Section 3.4, we provide the theoretical analysis of C2DBI, followed by the performance evaluation of C2DBI using simulation and testbed experiments in Sections 3.5 and 3.6, respectively. Finally, we conclude this chapter in Section 3.7.

3.2 Theoretical Analysis of 6TiSCH-MC

In this section, we provide the analysis of 6TiSCH-MC during the formation of 6TiSCH network. In this analysis, we consider the secure enrollment of nodes in the network during their joining process. Hence, transmissions of all bootstrapping control traffic such as EB, DIO, JRQ, and JRS happen in shared cells during network formation. We model the node's joining time and energy consumption during network formation. Finally, it is shown that the formation of 6TiSCH network degrades in terms nodes' joining time and energy consumption due to limited/insufficient resource allocation (i.e., shared cell) by 6TiSCH-MC.

3.2.1 Analytical Model for Joining Time

In Section 2.1.3, we briefly discuss the four states of a pledge during its joining in 6TiSCH networks. The pledge must follow all the states one after another to complete its joining process. Please note that the next state of the pledge depends only on the current state, and not any state in the past. In brief, occurrence of the next state does not depend on the whole state transition history i.e., the future is independent of the past given the present state. Furthermore, the pledges' joining process is a stochastic process as moving from one state to another state depends on various control packets transmission probabilities of the joined nodes and different network conditions such as varied packet loss probability, link failure. Additionally, the control packet transmission probability by each node is independent of other nodes and happens in shared cell. Hence, packet collision can be expected. In brief, at a certain time, the outcome is a random variable. So, as the next state of the pledge's joining process depends only on the current state, and not any state in the past and the

whole joining process is stochastic process in nature, therefore, we can model the behavior of the pledge during its network joining process using the Discrete Time Markov Chain model. Following this, in Figure 3.1, we show the four states of a pledge in the form of

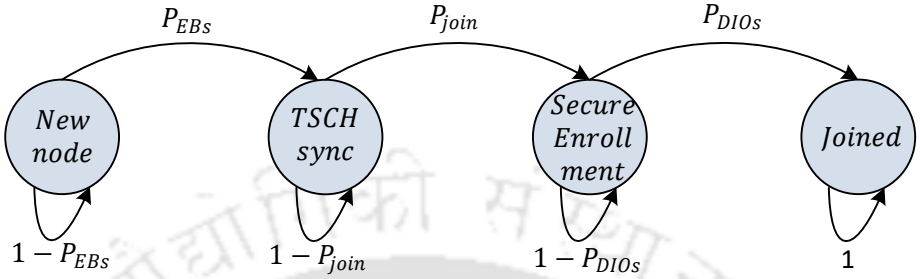


Figure 3.1: *Markov Chain model of node joining process*

Markov states with different transition probabilities. The last state of the model i.e., the *absorbing state* indicates that a pledge has joined with the existing network successfully. That means a pledge has become a 6TiSCH joined node, and now, it can transmit its own control packets for further expansion of the network.

In brief, at the initial state, a pledge waits for an EB from the JRC or any already joined nodes. Once a pledge receives a valid EB frame with probability P_{EB_S}, it moves to the second state. The second state represents that a pledge gets synchronized with the TSCH network and waiting to complete its secure enrollment to the network. For secure enrollment, a pledge transmits JRQ frame and waits for the corresponding JRS frame. Exchanging of these JRQ and JRS frames happens in shared cell. After finishing the secure enrollment process with probability P_{join}, a pledge moves to the third state. In the third state, a pledge waits for routing protocol information. Once it receives at least one fresh/recent DIO packet from its preferred parent, the pledge moves to its final absorbing state. In the absorbing state, a pledge successfully joins the 6TiSCH network and becomes a 6TiSCH joined node. Again, if we consider that the probability of receiving a DIO packet successfully is P_{DIO_S},

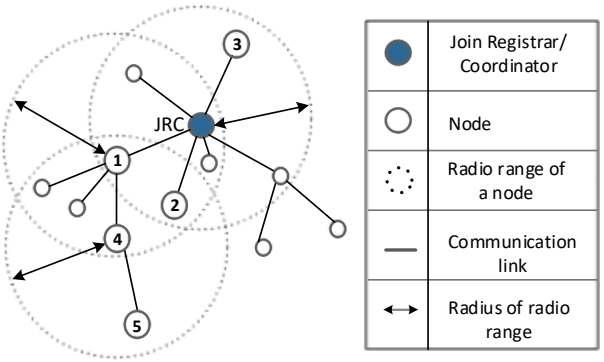


Figure 3.2: *An example of multi-hop network topology used in this work*

then the transition probability matrix of the Markov model can be written as follows,

$$M = \begin{bmatrix} 1 - P_{EB_S} & P_{EB_S} & 0 & 0 \\ 0 & 1 - P_{join} & P_{join} & 0 \\ 0 & 0 & 1 - P_{DIO_S} & P_{DIO_S} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using the Markov model, the *average number of slotframes* (ASF) require to reach the final absorbing state can be computed as follows,

$$\begin{aligned} ASF &= \frac{1}{P_{EB_S}} + \frac{1}{P_{join}} + \frac{1}{P_{DIO_S}} \\ &= \frac{1}{P_{EB_S}} + \left(\frac{1}{P_{JRQ_S}} + \frac{1}{P_{JRS_S}} \right) + \frac{1}{P_{DIO_S}} \end{aligned} \tag{3.1}$$

Now, to get the actual value of ASF, we need to calculate the values of P_{EB_S}, P_{DIO_S}, P_{JRQ_S}, and P_{JRS_S}, where P_{JRQ_S}, and P_{JRS_S} denotes the successful transmission probability of JRQ and JRS frame in a slotframe/shared cell, respectively. Note that the probability P_{join} is written in terms of P_{JRQ_S} and P_{JRS_S}. Let us consider a multi-hop network model as shown in Figure 3.2, where each node has a different number of neighbors. Let P_{eb}, P_{dio}, P_{jrq}, P_{jrs} are the transmission probabilities of EB, DIO, JRQ and JRS frame/packets in a shared cell, respectively. P_{loss} is the packet loss probability, and N_c is the total number of channels used in the network. The probability of receiving an EB frame successfully by a pledge in a shared cell is computed as,

$$P_{EB_S} = \frac{\sum_{n}^{M-n} nP_{eb}(1-P_{eb})^{n-1}(1-P_{dio})^{n-1}\times (1-P_{jrs})^{n-1}(1-P_{jrq})^{M-n}(1-P_{loss})P(N=n)}{N_c} \tag{3.2}$$

where n is the number of joined neighbor nodes, and M is the total neighbors of a pledge. The values of n and M can be different for every node present in a network. The above equation follows that the $(M-n)$ pledges join a network one by one, and the $P(N=n)$ denotes the probability that, at an instant, total number of joined neighbors is n. Again, when a joined node transmits its own EB frame, the remaining $(n-1)$ joined nodes should not transmit their EB frames, DIO packets, and JRS frames. The probability of this condition equals to $(P_{eb}(1-P_{eb})^{n-1}(1-P_{dio})^{n-1}(1-P_{jrs})^{n-1})$. Additionally, the neighbors but pledges also should not transmit any JRQ frames, which is defined by the probability equals to $((1-P_{jrq})^{M-n})$. Further, the transmitted frame should not be lost in the channel, which is computed by probability $(1-P_{loss})$. Finally, a pledge is not synchronized with the channel hopping sequence of the network at the beginning. Therefore, it searches in all the available N_c number of channels one by one, which reduces the EB success probability by N_c times. Further, the EB frames can be transmitted by any of the n joined neighbor nodes, which result in multiplying the computed probability by n. Note that a pledge can transmit JRQ frame only after receiving a valid EB frame, and a joined node transmits JRS frame only after receiving a valid JRQ frame. So, $P_{jrq} = P_{EB_S}$ and $P_{jrs} = P_{JRQ_S}$.

Considering that EB frame has higher priority over DIO packet, the probability of transmitting a DIO packet successfully in a shared cell is computed as,

$$P_{DIO_S} = \sum_{n}^{M-n} nP_{dio}(1-P_{dio})^{n-1}(1-P_{jrs})^{n-1}\times (1-P_{eb})^{n-1}(1-P_{jrq})^{M-n}(1-P_{loss})P(N=n) \tag{3.3}$$

Note that in the above equation, N_c is not used because a TSCH synchronized node knows the channel hopping sequence of the network after getting an EB frame. Likewise, the probabilities of transmitting a JRQ frame and receiving a JRS frame successfully in a shared cell are also computed as,

$$P_{JRQ_S} = \sum_{n}^{M-n}(M-n)P_{jrq}(1-P_{jrq})^{M-n-1}(1-P_{eb})^{n}\times (1-P_{dio})^{n}(1-P_{jrs})^{n}(1-P_{loss})P(N=n) \tag{3.4}$$

$$P_{JRS_S} = \begin{array}{l} \sum_{n}^{M-n} nP_{jrs}(1-P_{jrs})^{n-1}(1-P_{eb})^{n} \times \\ (1-P_{dio})^{n}(1-P_{jrq})^{M-n}(1-P_{loss})P(N=n) \end{array} \tag{3.5}$$

Now, to compute the Equations (3.2), (3.3), (3.4), and (3.5), we should know the transmission probabilities of EB and DIO in a shared cell. The probability P_{eb} can be calculated as,

$$P_{eb} = \frac{L}{I_{eb}} \tag{3.6}$$

where L is the slotframe length and I_{eb} is the beacon generation interval. We compute the probability P_{dio} following the procedure described in [45] as follows,

$$P_{dio} = (1-P_{eb})\frac{\begin{array}{c} 2^{N_D}(1-P_r)^{N_D} min(\frac{L}{2^{N_D} I_{dio}^{min}}, 1) + \\ \sum_{i=0}^{N_D-1} P_r 2^i (1-P_r)^i min(\frac{L}{2^i I_{dio}^{min}}, 1) \end{array}}{P_r + 2^{N_D}(1-P_r)^{N_D} + \sum_{j=1}^{N_D-1} P_r 2^j (1-P_r)^j} \tag{3.7}$$

where I_{dio}^{min} is the initial minimum DIO interval, P_r is the Trickle Algorithm (TA) [76] reset probability and N_D represents the total number of Trickle Algorithm states. Finally, we compute the *average joining time* (AJT) of a pledge as follows,

$$AJT = PJT + ASF \times L \tag{3.8}$$

where PJT is the*average joining time of the parent node* of a pledge in multi-hop network. For example, in Figure 3.2, joining time of node ④ is the summation of its own joining process time and its parent node's joining time (joining time of ① or ② whichever ④ selects as its parent).

Likewise, the joining time of each pledge in a network can be calculated, which is dependent on its parent's joining time and the number of neighbors. *Finally, the formation time of the entire network is the joining time of the pledge added last in the network.*

3.2.2 Analytical Model for Energy Consumption

During network formation process, two different types of nodes are present in the network. One is joined nodes, i.e., the nodes which have already joined the network, and the other type is pledge who wants to join the network. During network formation, both joined nodes

and pledges consume different amounts of energy because of their various radio activities. For example, a joined node needs to broadcast or receive a control packet once in each slotframe, whereas a pledge needs to keep its radio active all the time until it receives a valid EB frame. Therefore, we calculate the energy consumption of joined node and pledge separately.

In this analytical model, we do not consider the micro-controller states of a node to keep the analytical model as simple as possible. Only the radio state of a node during network formation is considered. The following subsections briefly described the computation method of calculating the energy consumption of pledge E_{pledge} and joined node $E_{joinedNode}$.

3.2.2.1 Energy Consumption by a Pledge

Referring to the Markov Chain model shown in Figure 3.1, the *average number of slotframes* (ASF) required to receive an EB frame successfully can be calculated as, $ASF_{EB} = \frac{1}{P_{EB_S}}$ During these $(\frac{1}{P_{EB_S}})$ slotframes, a pledge needs to keep active its radio to get a valid EB frame. Therefore, the average time a pledge needs to keep active its radio is,

$$Average\ Time_{EB} = PJT + \frac{L}{P_{EB_S}} \tag{3.9}$$

Note that in the above equation, parent node joining time is also added. It is because the parent node can transmit EB frame only after finishing its joining process. Converting this time duration into the number of timeslots, we get,

$$Average\ Slot_{EB} = \frac{Average\ Time_{EB}}{timeslot\ duration} \tag{3.10}$$

Therefore, average energy consumption by a pledge for successfully getting an EB frame is computed as,

$$Average\ Energy_{EB} = \frac{PJT \times P_{EB_S} + L}{T \times P_{EB_S}}\ (E_{RxData}) \tag{3.11}$$

where E_{RxData} is the average energy consumption by a node for receiving a frame/packet in a timeslot, and T is the duration of each timeslot. Similarly, average energy consumption

for successfully receiving a DIO packet is,

$$Average\ Energy_{DIO} = \frac{1}{P_{DIO_S}}(E_{RxData}) \tag{3.12}$$

Note that the variables L and T are not used in Equation (3.12) because a pledge is already synchronized with the network after getting a valid EB. Likewise, the average energy consumption by a pledge to transmit a JRQ and to receive its corresponding JRS frame successfully can be calculated as,

$$Average\ Energy_{JRQ} = \frac{1}{P_{JRQ_S}}(E_{TxData}) \tag{3.13}$$

$$Average\ Energy_{JRS} = \frac{1}{P_{JRS_S}}(E_{RxData}) \tag{3.14}$$

where E_{TxData} is the amount of average energy consumed to transmit a packet. Finally, the average energy consumption by a pledge during its network admission process is,

$$\begin{aligned} E_{pledge} &= Average\ Energy_{EB} + Average\ Energy_{DIO} \\ &+ Average\ Energy_{JRQ} + Average\ Energy_{JRS} \end{aligned} \tag{3.15}$$

3.2.2.2 Energy Consumption by a Joined Node

Average energy consumption by a joined node is the summation of energy consumption when it was a pledge and after becoming a joined node. However, in case of JRC, energy consumption as a pledge will be zero. A joined node needs to transmit different control packets for forming the rest of the network. For example, in Figure 3.2, $JN2$ needs to broadcast different control packets (e.g., EB, DIO, JRQ, JRS) so that $JN3$ can join the network. The number of such control packets transmitted by any joined node can be estimated as follows,

$$C_{EB} = \frac{ASF_{multihop}}{EB\ transmitting\ interval} = ASF_{multihop} \times P_{eb} \tag{3.16}$$

where $ASF_{multihop}$ denotes the average number of slotframes required to form the rest of the network, which can be calculated by summing all hop-by-hop ASF values and EB transmitting interval is written in terms EB transmission probability as we consider the

number of occurring slotframes (and so, shared cell) in the numerator. Similarly, the total number of DIO and JRQ packets received by a joined node can be calculated as,

$$C_{DIO} = ASF_{multihop} \times P_{dio} \tag{3.17}$$

$$\begin{aligned} C_{JRQ} &= \frac{ASF_{multihop}}{JRQ\ transmitting\ interval} \times No\ of\ new\ nodes \\ &= ASF_{multihop} \times P_{JRQ_S} \times K \end{aligned} \tag{3.18}$$

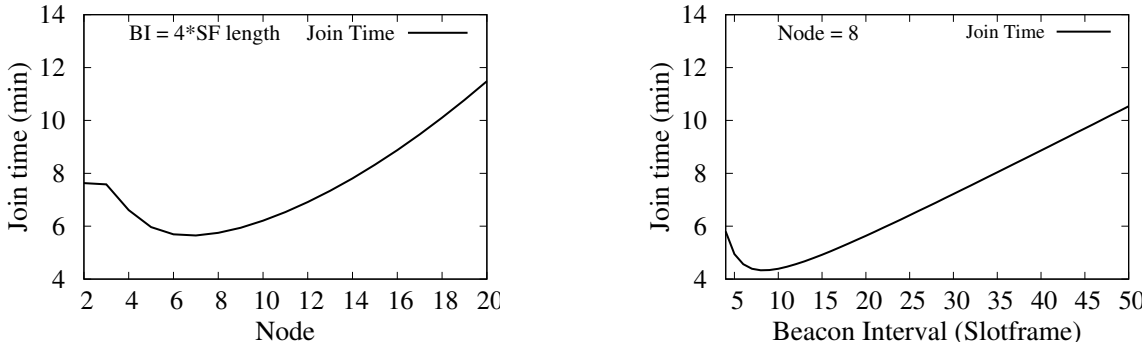
where K is the number of pledges to be admitted in the network. As a response to JRQ frames, same number of JRS frames also need to be transmitted. Therefore, $C_{JRS} = C_{JRQ}$, where C_{JRS} is the total number of JRS frames. Now, we compute the total average energy consumption by a joined node or JRC as follows,

$$\begin{aligned} E_{joinedNode} = E_{pledge} &+ (C_{EB})E_{TxData} + (C_{DIO})E_{TxData} \\ &+ (C_{JRQ})E_{RxData} + (C_{JRS})E_{TxData} \end{aligned} \tag{3.19}$$

where E_{TxData} and E_{RxData} are the average amount of energy consumed by a joined node during transmission and reception of a frame/packet in a slotframe, respectively. For the JRC, the value of E_{pledge} is 0, and for the other joined nodes, E_{pledge} depends on their respective joining times in the network.

3.2.3 Analytical Results

In this section, we graphically present the above analysis and show the limitations of the 6TiSCH-MC standard during network formation procedure. Let us consider a network with the following given values: $N_C = 16$, $I_{eb} = 4{*}L$, $P_{loss} = 0.2$, $I_{dio}^{min} = 8\ ms$, $L = 101\ timeslots$, $T_i = 10\ ms$, $N_D = 16$ and $P_r = 0.2$. We compute the *average joining time* (AJT) and energy consumption of nodes during node joining process using the above analytical models. Figure 3.3a depicts that, for a fixed beacon interval, the joining time of a pledge increases gradually with the increasing number of nodes after a certain threshold. In other words, the performance of the network degrades once it allows to join a pledge with it. It can be seen



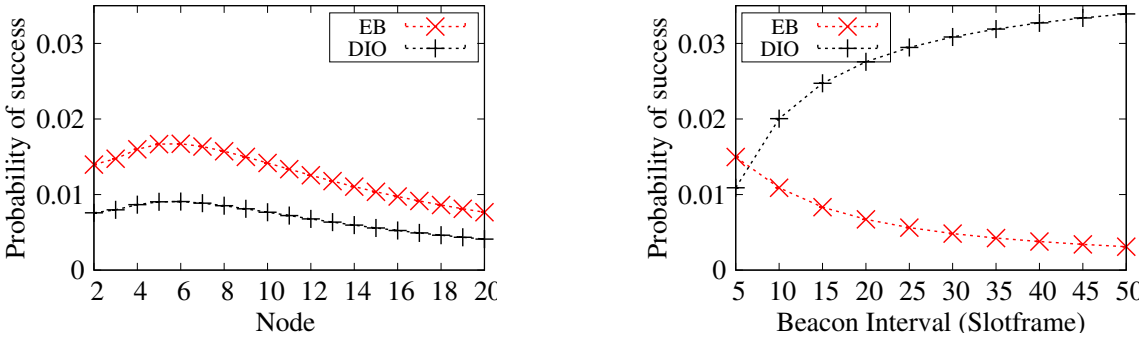
(a) *Joining time against varying number of nodes*

(b) *Joining time of a pledge with varying beacon interval*

Figure 3.3: *Numerical results on average 6TiSCH joining time during with increasing values of joined node and beacon interval*

that, initially, an increasing number of nodes helps the pledges to join in less time because more number of EB frames are transmitted by several joined nodes. But when the number of nodes increases further, contention among the participating nodes also increases due to the limited number of shared cells, i.e., one shared cell per slotframe. This, in turn, results in higher joining time.

Figure 3.3b shows the results of variable beacon interval with a fixed number of nodes, i.e., 8 nodes. The figure shows that joining time is high in case of low beacon intervals and high beacon intervals. In low beacon intervals, contention among the joined nodes increases as they frequently transmit their beacons. This, in turn, increases the time period to get a valid EB by a pledge. In high beacon intervals, a pledge has to wait for a long time to get a beacon. Therefore, once again, the joining time increases with the increase in beacon interval. Figure 3.4a shows the change in success probabilities of EB, DIO packets with the varied number of nodes. Because of the increasing number of joined nodes, congestion also increases, which results in low success probabilities. On the other hand, Figure 3.4a shows the change in success probabilities of EB, DIO packets with varied beacon generation intervals. The success probability of EB frame decreases with the increasing beacon generation interval as nodes transmit their packet after along interval. However, DIO packet gets more opportunity to get transmitted with increasing beacon generation interval, and so its success probability increases.

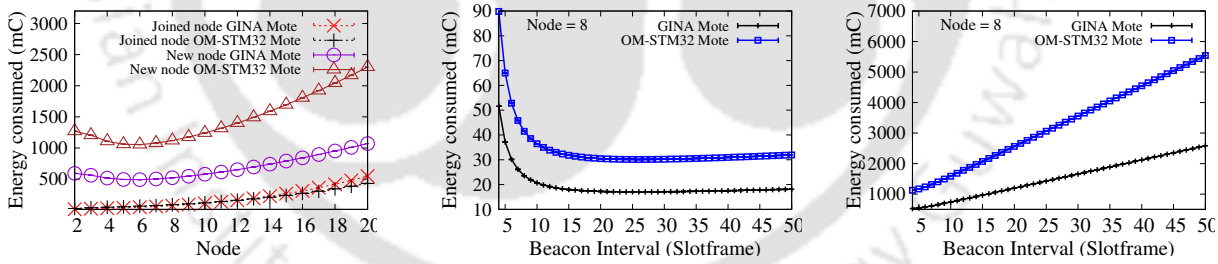


(a) *Success probability of different control packets with varied number of nodes*

(b) *Success probability of different control packets with varied beacon interval*

Figure 3.4: *Numerical results on success probabilities of different control packets with increasing values of joined node and beacon interval*

For calculating the energy consumption of the joined node and pledge, two different types of wireless motes, `GINA` and `OM-STM32` [81], are considered. These two types of nodes are considered to show the effect of fixed beacon interval scheme in energy consumption. Both the `GINA` and `OM-STM32` motes consume 69.6 and 119.2 μC (*micro coulombs*), respectively, for transmitting a packets. Again, the same motes consume 72.1 and 154.8 μC for receiving a packet, respectively.



(a) *Energy consumption of both joined and new nodes under varying number of nodes*

(b) *Energy consumption by a joined node under varying beacon interval*

(c) *Energy consumption by a pledge under varying beacon interval*

Figure 3.5: *Numerical results on energy consumption of nodes during 6TiSCH network formation with increasing values of joined node and beacon interval*

Considering this energy consumption data, we plot the average energy consumption graph of a joined node and a pledge by varying the number of nodes and beacon intervals in Figures 3.5a, 3.5b, and 3.5c, respectively. It can be observed in Figure 3.5a that the energy consumption by a joined node almost remains unchanged with respect to varying number of nodes. This is because the transmitting rate of control packets by a joined node is not

changed when a pledge joins into the network in fixed beacon interval scheme. On the other hand, energy consumption by a pledge increases with the increased number of nodes. It is because the contention in shared cells increases with the increased number of joined nodes. A pledge needs to wait for a long time to get a valid EB. Again, it can be seen that a joined node consumes less energy than a pledge. This is because a pledge has to keep its radio active until it gets a valid EB, whereas a joined node transmits/receives only in a single timeslot (i.e., shared timeslot) in a slotframe. Figure 3.5b shows the energy consumption of a joined node with varying beacon generation intervals. It can be observed that energy consumption by a joined node almost remains the same with the increasing beacon interval. Whereas a different scenario is observed for the pledges as shown in Figure 3.5c. Here, energy consumption increases with the increase of beacon interval. During high beacon interval, though the congestion in shared cell is reduced but a pledge has to wait for a long time to get a valid EB. During this entire time period the pledge needs to keep active its radio all the time which consumes more energy, and thus, energy consumption increases again.

From these analytical results, it can be established that the performance of a network decreases with the increased number of joined nodes, i.e. when a pledge joins in the network. Further, it is also important to keep generation beacon interval in the optimal range to maintain the joining time and energy consumption as low as possible.

3.3 Channel Condition based Dynamic Beacon Interval (C2DBI)

The previous analysis (in Section 3.2.3) shows that joining time of a pledge increases with the increasing number of nodes in a network that follows a fixed beacon generation interval scheme. The number of nodes has a significant impact on the energy consumption of a pledge. We propose a dynamic beacon generation interval scheme to reduce the joining time and energy consumption during network formation.

As the congestion in shared cell increases with the increasing number of joined nodes, it is crucial to reduce congestion in shared cells when the number of joined nodes increases. For this purpose, we propose a scheme to dynamically adjust the beacon generation interval depending upon the congestion status in shared cells. When congestion is increased in the shared cells, the proposed method increases the beacon interval to decrease the number of beacons generated by joined nodes. This decreasing rate of beacon generation reduces the congestion and thus gives opportunities for other control packets to be transmitted in the network. On the other hand, when the congestion is low in the shared cells, the beacon generation interval is reduced so that a pledge can quickly get a valid beacon. This, in turn, reduces the energy consumption of a pledge as it needs to keep its radio active until it gets synchronized with the TSCH network. The congestion estimation can be done using different methods considering many parameters such as the number of neighboring nodes, packet loss, queue length, and channel load.

In this work, shared cell or channel busy ratio (CBR) [82] is used for congestion estimation as it gives better results in both the static and dynamic networks. Static network means when the nodes are in fixed position, but the nodes are mobile in dynamic network. The CBR is measured for a fixed interval, and based on the measured CBR value, the beacon interval of a joined node is calculated and assigned. The assigned beacon interval will be used in the next time period. The CBR for a particular period of time is calculated as,

$$CBR = \frac{Busy\ shared\ cells}{Busy\ shared\ cells + Empty\ shared\ cells} \tag{3.20}$$

A shared cell is considered as a busy shared cell by a node if the measured signal strength in that cell is higher than the *clear channel assessment* (CCA), or either the node receives a packet or transmits a packet. Otherwise, a shared cell is considered as an empty or idle cell. After computing the CBR for a particular time interval, a joined node computes its

Algorithm 1 Channel Condition based Dynamic Beacon Interval

Input: N_i : Node i; W : CBR period; T_t : Current time; T_{cbr} : Time instant of last CBR calculation
Output: I_{eb} for the next period
At each timeslot T_t
if the current $Link_{type}$ is *Shared* **then**
    increment *totalSharedSlot* variable by unity
    if current CCA_{status} is *busy* or received packet/transmit a packet **then**
        increment *busySlot* variable by unity
    end if
end if
if the difference between T_t and T_{cbr} is greater than or equal to W **then**
    $CBR_{(N_i,T_{cbr},W)} = busySlot/totalSharedSlot$
    if the CBR is not equal to 0 **then**
        $I_{eb} = I_{eb}^{min} + (I_{eb}^{max} - I_{eb}^{min})^{CBR}$
    else
        $I_{eb} = I_{eb}^{min}$
    end if
    Update T_{cbr} by T_t
    Reset *busySlot* and *totalSharedSlot* to 0
end if

own beacon interval for the next time as follows,

$$I_{eb} = \begin{cases} I_{eb}^{min} & \text{if CBR} = 0 \\ I_{eb}^{min} + (I_{eb}^{max} - I_{eb}^{min})^{CBR} & \text{otherwise} \end{cases} \tag{3.21}$$

Using the above equation, a joined node always choose its beacon generation interval in between the minimum beacon generation interval I_{eb}^{min} and maximum beacon generation interval I_{eb}^{max} as the values of CBR varies between 0 to 1. Algorithm 1 describes the method of adjusting beacon generation interval dynamically based on present channel congestion status.

3.4 Theoretical Analysis of C2DBI

Let us consider that P_{jns} is the probability of an already joined node transmits a packet in a shared cell and P_{nns} is the probability that a pledge transmits a packet in a shared cell.

If P_{ts} denotes that there is a transmission in a shared cell, then P_{ts} can be written as,

$$P_{ts} = P_{jns} + P_{nns} \tag{3.22}$$

Now, an already joined node transmits all the EB, DIO, JRS control packets. So, we can write P_{jns} as follows,

$$P_{jns} = P_{eb} + P_{dio} + P_{jrs} \tag{3.23}$$

where P_{eb} and P_{dio} are the transmission probabilities of transmitting an EB frame and a DIO packet in a slotframe, respectively. And P_{jrs} is the probability of transmitting JRS frame after getting valid JRQ. On the other hand, a pledge only transmits JRQ frames during network formation. Hence, P_{nns} can be computed as,

$$P_{nns} = P_{jrq} \tag{3.24}$$

where P_{jrq} is the transmission probability of a JRQ frame in a slotframe. Now, considering both the P_{jns} and P_{nns}, the Equation (3.22) can be rewritten as,

$$P_{ts} = P_{eb} + P_{dio} + P_{JRQ_S} + P_{EB_S} \tag{3.25}$$

where P_{jrq}=P_{EB_S} as a pledge transmits JRQ frame only after receiving a valid EB frame, and P_{jrs}=P_{JRQ_S} as a joined node transmits JRS frame only after getting a valid JRQ frame from a pledge.

The same multi-hop network model is considered, as shown in Figure 3.2, for evaluating the proposed scheme. Different joined nodes may have different numbers of neighbor nodes, so the calculated CBR values of the joined nodes would also be different. Let's assume that joined node k has nn number of pledges and jn number of joined nodes in its communication range. These nodes use the same shared cell for transmitting their control packets. Sometimes, it is also possible that there is no transmission in a shared cell. These types of cells are known as empty cells. The probability of such an empty shared cell is,

$$P_{Empty} = (1 - P_{ts})^{nn+jn} \tag{3.26}$$

And the probability of having at least one transmission in a shared cell is,

$$P_T = 1 - (1 - P_{ts})^{nn+jn} \tag{3.27}$$

Now, the CBR at a particular time interval for node k can be calculated as follows,

$$\begin{aligned} CBR_k &= \frac{Busy\ shared\ cells}{Busy\ shared\ cells + Empty\ shared\ cells} \\ &= \frac{P_T}{P_T + P_{Empty}} \end{aligned} \tag{3.28}$$

Note that P_T and P_{Empty} are computed with respect to node k. This is because a CBR measuring time interval consists of several slotframes. In each shared timeslot of slotframe, either a transmission occurs, or there is no transmission. Now, if we consider that the CBR measuring time interval consists of W shared cells, then the Equation (3.28) can be written as follows,

$$CBR_k = \frac{\sum_{i=0}^{W}(1 - (1 - P_{ts})^{nn+jn}) * T_i}{\sum_{i=0}^{W}\left(1 - (1 - P_{ts})^{nn+jn} + (1 - P_{ts})^{nn+jn}\right) * T_i} \tag{3.29}$$

where T_i denotes the shared cell duration in i^{th} slotframe. As all the shared cell durations are same, we can simplify the above equation as follows,

$$CBR_k = 1 - (1 - P_{ts})^{nn+jn} = P_T \tag{3.30}$$

Now, the calculated CBR value can be directly used in the dynamic beacon interval formulation as shown in Equation (3.21) to calculate next beacon interval of node k i.e., I_{eb}^k.

When different joined nodes have different number of neighbors, then their calculated CBRs will also be different. This results in varied beacon interval of each joined node. Now, with the proposed dynamic beacon interval approach, the successful EB frame receiving probability can be calculated as follows,

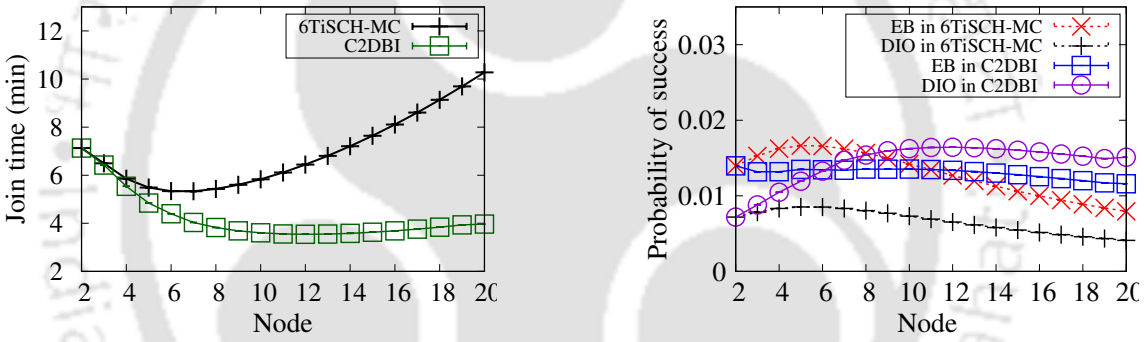
$$P_{EB_S} = \frac{\sum_{n}^{M-n} nP_{eb}^k \prod_{i=1,i\neq k}^{n-1}(1 - P_{eb}^i)(1 - P_{dio})^{n-1}}{(1 - P_{jrs})^{n-1}(1 - P_{jrq})^{M-n}(1 - P_{loss})P(N = n)} \tag{3.31}$$

where P_{eb}^i denotes the EB frame transmitting probability in a shared cell by the joined node i, which is calculated by $\frac{L}{I_{eb}^i}$. Similarly, we can calculate the successful packet transmission

probabilities in a slotframe for other control packets such as P_{DIO_S}, P_{JRQ_S}, P_{JRS_S} using varied beacon interval, i.e., I^i_{eb} of each joined node i.

3.4.1 Analytical Results of C2DBI

Assigning similar values to the variables as mentioned in Section 3.2.3, few graphs are plotted for the proposed method and compared them with the fixed beacon interval based 6TiSCH-MC scheme. Figure 3.6a depicts the join time to the number of nodes for the proposed method C2DBI, and also compares it with the 6TiSCH-MC method. Here, we take $I^{min}_{eb} = 4040$ *milliseconds*, which is the same value used in 6TiSCH-MC scheme, and $I^{max}_{eb} = 10100$ *milliseconds*. These values are taken randomly. However, any other value



(a) *Joining time against varying number of nodes*

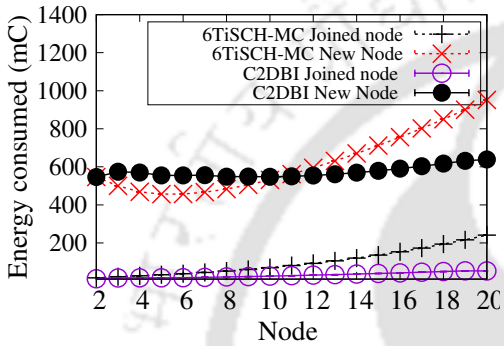
(b) *Probability of success of different control packets (EB and DIO)*

Figure 3.6: *Comparing numerical results on joining time and different control packets success probabilities of the C2DBI with 6TiSCH-MC*

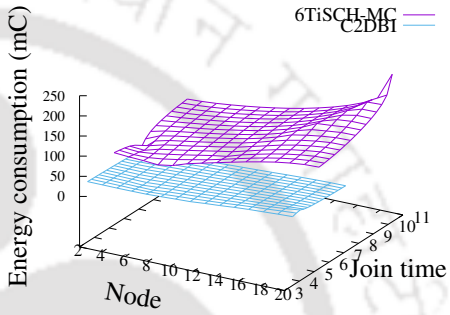
can also be considered. It is observed in Figure 3.6a that the proposed model outperforms the 6TiSCH-MC model in terms of joining time of the nodes. Figure 3.6b depicts the reason behind it. It shows the probabilities of successfully transmitting different control packets during network formation in both schemes. When the number of joined nodes increases, congestion in the shared cell also increases. This increasing congestion reduces the success probabilities of different control packets in 6TiSCH-MC. Whereas, in C2DBI, congestion is reduced by increasing the beacon generation interval, which improves the success probabilities of different control packets as shown in Figure 3.6b. This increasing successful control packets probabilities of EB and DIO frame/packets improve the overall

joining time of the nodes in the network.

The energy consumption graph of GINA motes for both the 6TiSCH-MC and the proposed C2DBI schemes is shown in Figure 3.7. The plotted graph shows how C2DBI maintains almost stable energy consumption when the number of nodes increases in the network. It can be observed in the graph that the joined node consumes almost same energy in both the approaches. This is because, in both the approaches, the joined nodes use similar radio duty cycles for transmitting their control packets. A small difference is observed because



(a) *Energy consumption of both type of nodes*



(b) *Join time and energy consumption of a pledge*

Figure 3.7: *Comparing numerical results of pledge's energy consumption the C2DBI with 6TiSCH-MC*

less number of EB frames are transmitted when the congestion is high in C2DBI. But a significant difference in energy consumption can be observed for the pledges when the number of nodes varies. This can be explained by the higher congestion in 6TiSCH-MC when the number of joined nodes increases. Higher congestion forces the pledges to wait for more amount of time to get a valid EB. This increasing amount of channel scanning time to get EB frame causes more energy consumption as the pledges need to keep their radio active. Whereas in our proposed scheme C2DBI, congestion is reduced by increasing the beacon intervals of the already joined nodes. Therefore, almost equal energy consumption can be seen in C2DBI even in the presence of more number of nodes. Figure 3.7b shows the energy consumption and joining time of a pledge together with respect to varying number of nodes for both 6TiSCH-MC and C2DBI schemes considering GINA motes.

3.5 Simulation based Evaluation of C2DBI

3.5.1 Simulation Settings

In this section, the performance of the proposed scheme is analyzed using the Cooja simulator on Contiki-NG OS [78]. In our evaluation, the nodes are deployed in a fixed square (6×6) size grid area where the JRC (or RPL root) is placed at the top left corner of the considered multi-hop grid topology. The simulation parameters and their corresponding values are mentioned in Table 3.1. In 6TiSCH-MC, the considered beacon interval is equal to 4 *seconds*. In the proposed scheme, we take the same value for minimum beacon interval I_{eb}^{min}, and the value of the maximum beacon interval I_{eb}^{max} equals to 12 *seconds*. The CBR calculation interval is considered to be 8 *seconds*. For DRA [45], the limit of maximum shared cells is set to 8 within a slotframe. The slotframe length is varied from 33 to 101 *timeslots* in order to obtain different results with different network configurations. We run each simulation for 120 *minutes* using a realistic *Multipath Ray-Tracer Medium* (MRM) channel model. This model provides various propagation effects such as multi-path, refraction, and diffraction. During simulation, the keep-alive packet is used to keep the communication between a node and its parent active. When a node does not hear from its parent for $30s$, it initiates a keep-alive packet.

Table 3.1: *Simulation parameters*

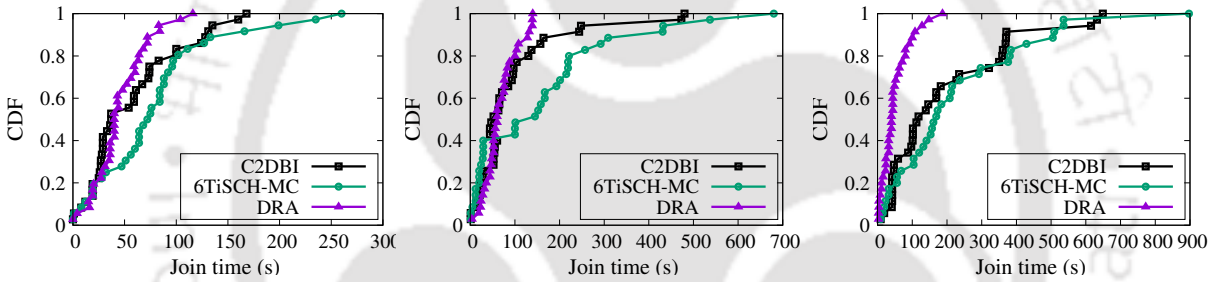
Parameter	Value
Operating System	Contiki-NG
MAC protocol	802.15.4e TSCH
Number of channels	16
Timeslot length	10 ms
Slotframe length	33,67,101 *timeslots*
RPL version	RPL Lite
RPL DIO interval	Trickle Algorithm
Keep-alive timeout	30 s
Mote type	Cooja Mote
Propagation model	MRM
Simulation duration	120 min

3.5.2 Simulation Results

We consider two metrics i.e., *TSCH network formation time* and *6TiSCH network formation time* to measure the performance of C2DBI compared to DRA [45] and 6TiSCH-MC [40], which are describes one by one as follows:

3.5.2.1 TSCH Network Formation Time

The performance of TSCH network formation is evaluated by means of total time required by the nodes to join the TSCH network. A pledge scans for EB frame to join the network. So, TSCH network formation time for a pledge is the time required by the pledge to get a valid EB. In brief, it is the time taken by a pledge to be a TSCH synchronized node.



(a) *TSCH formation, SF 33* **(b)** *TSCH formation, SF 67* **(c)** *TSCH formation, SF 101*

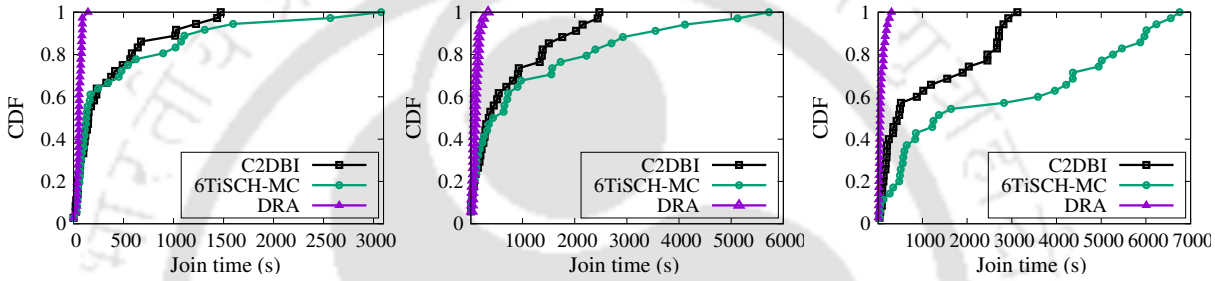
Figure 3.8: *Simulation results of TSCH formation time using different schemes for different slotframe (SF) size (= 33, 67, 101 timeslots)*

Figures 3.8a, 3.8b and 3.8c show the Cumulative Distribution Function (CDF) of TSCH synchronization/formation time with different slotframe lengths such as 33, 67, and 101 *timeslots*. It can be observed in all the three figures that DRA takes less amount of time than our proposed method (C2DBI) and 6TiSCH-MC. The reason for this is the use of multiple shared cells per slotframe by the DRA. DRA dynamically increases the number of shared cells per slotframe when more control packets are available in the network. However, in the C2DBI and 6TiSCH-MC, only one shared cell is used, which delays the transmission of already generated control packets. Although both the C2DBI and 6TiSCH-MC use a single shared cell in each slotframe, the C2DBI takes less amount of time to form a TSCH network compared to that in 6TiSCH-MC because C2DBI reduces congestion in the shared

cells by dynamically varying the beacon generation interval of the nodes when the shared cells are already congested.

3.5.2.2 6TiSCH Network Formation Time

To expand the network further, it is necessary for a pledge to join the network fully. That means the pledge should also get the routing information of the network along with EB frame to transmit its own beacon frame. The Figures 3.9a, 3.9b and 3.9c show the time required by a newly joined node to transmit its first EB after completely joining the network. Once



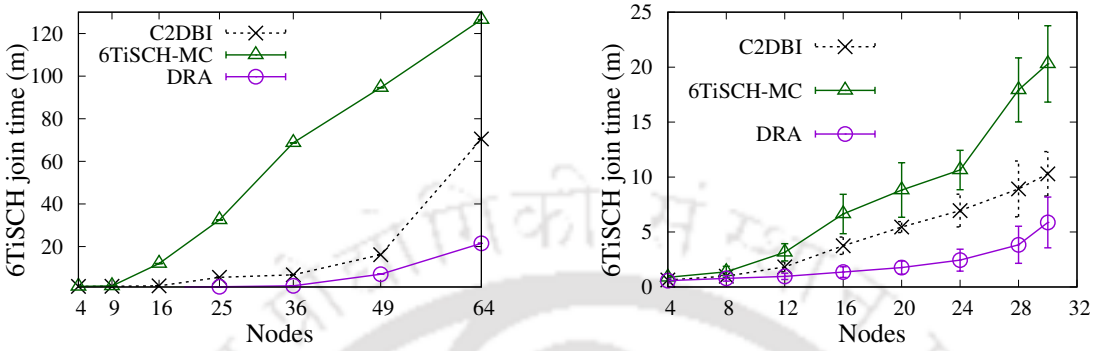
(a) *6TiSCH formation, SF 33* **(b)** *6TiSCH formation, SF 67* **(c)** *6TiSCH formation, SF 101*

Figure 3.9: *Simulation results of 6TiSCH network formation time using different schemes for different slotframe (SF) size (= 33, 67, 101 timeslots)*

again, it is observed that DRA performs better than C2DBI and 6TiSCH-MC. The reason is again usage of multiple shared cells per slotframe by the nodes in DRA. On the other hand, dynamic beacon interval reduces congestion in shared cells, and, thus, it performs better than 6TiSCH-MC. Note that C2DBI and 6TiSCH-MC use a single shared cell in a slotframe.

Again, it can be observed that, though the pledges join in the TSCH network quickly, they take a longer time to transmit their beacons. This is because of the delay in getting DIO packets when congestion in shared cells increases due to the limited available resources. Further, the joining time increases with the increasing slotframe lengths. This is because the frequency of the occurring shared cells decreases with the increasing slotframe length within a time period. Hence, even though, the control packets are generated at the same rate, less number of control packets are transmitted because of high congestion in less number of shared cells. This ultimately results in increasing joining time of the nodes. However,

in DRA, because of the usages of multiple shared cells per slotframe, nodes can transmit their packets without contending much. In brief, DRA significantly reduces the congestion in shared cell. Hence, DRA gives better results with respect to joining time.



(a) *Using grid topology* **(b)** *Using random topology*

Figure 3.10: *Simulation results on 6TiSCH network formation time under varying number of nodes*

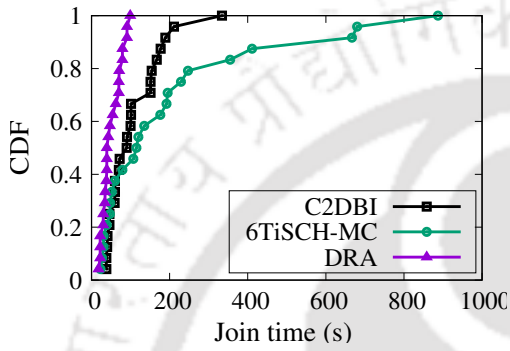
Apart from the above results, in Figures 3.10a and 3.10b, we show simulation results on 6TiSCH network formation time under varying number of nodes considering two types of topologies i.e., grid and random topology. In random topology, nodes are randomly deployed in a fixed area, whereas in grid topology, nodes are deployed in fixed square $x \times x$ size grid area, where x is an integer from 2 to 8. Please note that we show the results of random topologies using 95% confidence interval. In these simulation experiments also we observe similar behavior as mentioned before on pledges joining time using all the schemes. So, from these results, we can further conclude that our proposed scheme improves the joining time compared to 6TiSCH-MC in all kind of network settings, whereas, DRA always performs better compared to our proposed scheme in terms of joining time.

3.6 Testbed Evaluation of C2DBI

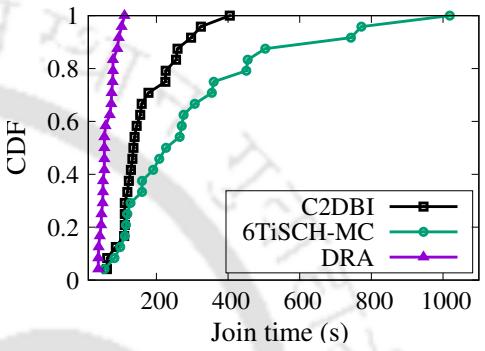
The implementation of C2DBI on Contiki-NG has been flashed into the `M3` nodes of FIT IoT-LAB [80]. We consider two topologies i.e., (5×5) grid topology and (2×12) linear topology in Lille and Grenoble locations, respectively, for testing our proposed scheme in FIT IoT-LAB. The used `M3` node is an STM32 (ARM Cortex M3) micro-controller based node

that supports FreeRTOS, Contiki, and RIOT operating systems. We consider a similar configuration used in our simulation experiments in terms of beacon interval for all the schemes except the slotframe length. Here, the slotframe length is set to 101 *timeslots*. The obtained results in our testbed experiments can be explained as follows:

Figures 3.11a and 3.11b show the time required by a pledge to be a TSCH synchronized node and 6TiSCH joined node in (5×5) topology, respectively. Similarly, Figures 3.12a and



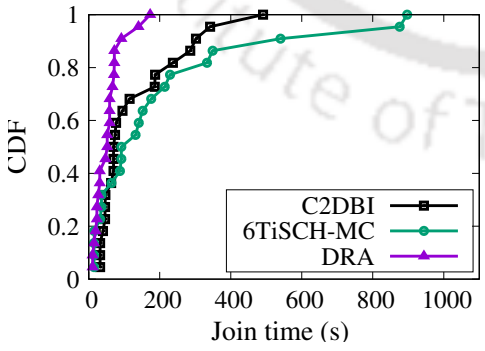
(a) *TSCH formation/synchronization time in* 5×5 *topology*



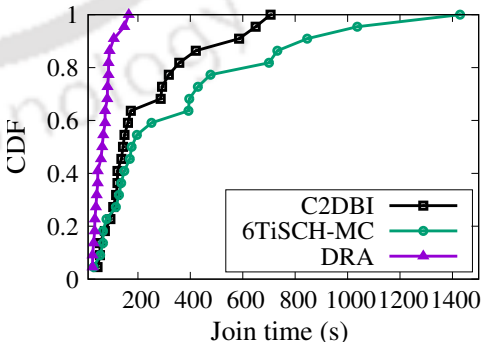
(b) *6TiSCH formation time in* 5×5 *topology*

Figure 3.11: *Testbed results corresponding to* 5×5 *topology*

3.12b show the time required time required by a pledge to be a TSCH synchronized node and 6TiSCH joined node in (2×12) topology, respectively. The plotted results show that the DRA performs better than the proposed scheme C2DBI as well as 6TiSCH-MC in terms of both TSCH joining time and 6TiSCH joining time. This is because of the allocation of



(a) *TSCH formation/synchronization time in* 2×12 *topology*



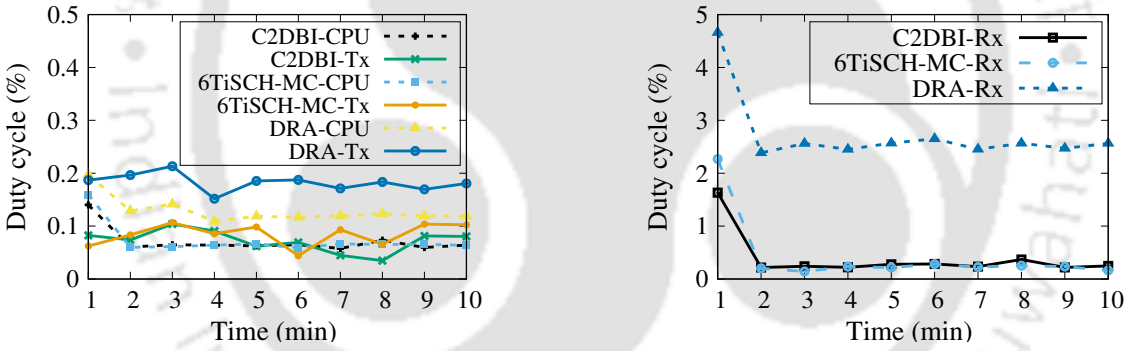
(b) *6TiSCH formation time in* 2×12 *topology*

Figure 3.12: *Testbed results corresponding to* 2×12 *topology*

more bandwidth (i.e., shared cell) per slotframe for the transmission of generated control

packets by DRA. On the other hand, the proposed scheme and 6TiSCH-MC use a single shared cell in a slotframe. In this regard, the proposed scheme performs better than the 6TiSCH-MC because of the reduction of congestion in the shared cell using dynamic beacon generation interval.

In our testbed experiments, we also collect the results on nodes' *radio duty cycle* during the formation of 5×5 grid network. Here, the radio duty cycle of a node corresponding to a micro-controller state is the ratio of time spent in that micro-controller state (e.g., Rx) in every 60 *seconds* by a node. We observe that even though DRA performs better with respect to joining time, it has very high radio duty cycle as shown in Figures 3.13a and 3.13b. Results in the Figures 3.13a and 3.13b show the average radio duty cycles of all nodes present in the network in their three different states (CPU_{active}, Tx, Rx) during the initial 10 *minutes* of network formation. In this initial 10 *minutes* of the network formation,



(a) *CPU active and Tx duty cycle* **(b)** *Rx duty cycle*

Figure 3.13: *Testbed results on radio duty cycle using* 5×5 *topology*

most nodes join the network. After this time interval, there is no such significant change in average radio duty cycles. As DRA uses more number of shared cells, the nodes need to keep their radio active for more amount of time than the single shared cell based schemes such as C2DBI and 6TiSCH-MC. Therefore, it results in a higher duty cycle. Again, higher duty cycle increases the energy consumption of a node. So, the nodes in DRA consume more energy than that in C2DBI and 6TiSCH-MC. Thus channel condition based dynamic beacon interval significantly improves the network formation performance with respect to joining time and energy consumption.

3.7 Summary

In this chapter, we proposed a *channel condition based dynamic beacon interval* (C2DBI) for faster association of nodes in 6TiSCH networks. At first, using a Markov Chain based analytical model, we showed that the performance of a network degrades with the increased number of joined nodes due to fixed beacon interval in 6TiSCH-MC. To overcome this problem, we proposed C2DBI, which dynamically adjusts the beacon generation interval of a joined node depending on its present channel congestion status measured by the CBR parameter. Theoretical analysis of the proposed scheme showed how network formation performance is improved with varied beacon generation intervals compared to the fixed beacon generation interval scheme. Furthermore, we also evaluated the proposed scheme using Cooja simulator and performing testbed experiments in FIT IoT-LAB. The received results were compared with the existing benchmark schemes i.e., DRA and 6TiSCH-MC standard. Though C2DBI does not perform better than the DRA regarding network formation time, from the energy consumption perspective, C2DBI is better than all the existing works. Further, to note that DRA uses multiple shared cells per slotframe, whereas C2DBI uses only one. It was observed in our simulation that around 8% of the total slotframe of size 101 is used as shared cells in DRA, whereas our proposed method uses only 0.99% during network formation. This observation signifies that sensory data packet transmissions are less affected by the proposed method C2DBI compared to that in DRA. From the obtained results, we can claim that the C2DBI scheme could be more suitable for networks where energy consumption is an important constraint along with network formation time. Please note that every node keeps its radio active in shared cell and also performs CCA before transmitting control packets. Hence, shared cell busy ratio can be calculated by each individual node without any signaling overhead in the network. Hence, no addition communication overhead there in the proposed scheme. Furthermore, we used a simple equation to vary the EB generation interval based on the shared cell congestion status. So, it does not incur much computational cost also. Therefore, in multi-user scenario, each node computes the ratio independently and varies its own EB generation interval accordingly. The transmission of

generated EB packet in shared cell is performed following the default contention mechanism as mentioned in the 6TiSCH-MC, and so C2DBI does not modify the contention procedure and thus the channel resource allocation process.

Apart from the problem of fixed beacon generation interval, our probabilistic analysis reveals that the performance of 6TiSCH-MC standard degrades the formation of 6TiSCH network because of the following reasons: (i) 6TiSCH-MC considered that beacon frame has the highest priority over all other control packets, (ii) 6TiSCH-MC provides minimal routing information during network formation, and (iii) sometimes, joined node can not transmit control packets due to high congestion in shared cell using 6TiSCH-MC. In Chapter 4, we briefly discuss these three problems and proposed solutions to address them.

"Intelligence plus character – that is the goal of true education."

~Martin Luther King Jr. (1929-1968)

4

Opportunistic Control Packet Transmission

This chapter proposes two schemes: *opportunistic priority alternation and rate control* (OPR) and *opportunistic channel access* (OCA) for efficient transmission of control packets in 6TiSCH network in order to reduce its formation time and enhance network lifetime. The first scheme changes the priority of control packets and enables sufficient transmission of DIO packet during network bootstrapping, whereas, the second scheme provides mechanism to access the shared cell quickly by the nodes which have urgent packet to transmit. The proposed schemes are evaluated using theoretical analysis, simulation, and testbed experiments.

4.1 Introduction

In the previous chapter (i.e., Chapter 3), we have shown that due to limited bandwidth allocation by 6TiSCH-MC standard, the formation of 6TiSCH network degrades when more pledges join the network. Apart from this, we observe few more problems in the allocation by 6TiSCH-MC standard as follows. According to 6TiSCH-MC, the network advertisement frame, i.e., Enhanced Beacon (EB), has the highest priority among all the control packets. Because of this fixed priority and scarcity of bandwidth, the other control packets starve to get transmitted by a joined node. Again, the requirement of routing information for a newly joining node is unseen to the already joined nodes. Hence, the joined nodes do not transmit routing information packets immediately when it is required for pledge. So, in worst case, the pledges suffer from long waiting times to get routing information i.e., DIO packet after getting valid EB frame. Furthermore, uncertainty in accessing the shared cells also contributes to longer network formation time in dense 6TiSCH network, which is co-located with other networks using the same frequency band. It is observed that none of the existing works on 6TiSCH network address the above mentioned problems. As mentioned before the state-of-the-art approaches (e.g. [47–57]) focused on either the number of transmissions or scheduling of EB frames during network formation. However, both the EB frame and DIO packet are necessary for multi-hop 6TiSCH network formation. Although Vallati *et al.* [45] and Vucinic *et.al.* [46] considered the transmission of both EB frame and DIO packet during network formation, their solutions have not considered the above mentioned three problems in 6TiSCH-MC, and these problems will persist even after applying their solutions.

To deal with EB's highest priority, incoherency between network information requirement and different control packet transmission rate, and uncertainty in accessing shared cell (or shared transmission channel), we propose two schemes. The first one is an intra-node scheme, namely, *opportunistic priority alternation and rate control* (OPR), which dynamically adjusts the priority of different control packets based on their requirement in the network. It also opportunistically increases the rate of transmission of DIO packet during network formation. The other scheme, namely, *opportunistic channel access* (OCA), pro-

vides a global solution to the nodes having urgent packets to transmit by allowing them to access shared transmission channels opportunistically. This is basically an inter-node scheme. The proposed schemes together reduce the overall network formation time.

We summarize the major contributions of this chapter as follows:

- Analytically, we show the demerits of EB's highest priority, absence of routing information, and uncertainty of shared channel access during bootstrapping in 6TiSCH network.
- An *opportunistic priority alternation and rate adjustment* (OPR) scheme is proposed to deal with the demerits of EB's highest priority and absence of routing information.
- An *opportunistic channel access* (OCA) scheme is proposed to transmit urgent packets with minimum delay.
- Theoretical analysis of the proposed schemes are provided using Markov Chain models.
- Comparison based performance evaluation of the proposed solutions are performed using Cooja simulator and FIT IoT-LAB testbed.

The rest of the chapter is organized as follows. In Section 4.2, we describe the shortcomings of 6TiSCH-MC based 6TiSCH network formation procedure along with the validation of the problem statements using theoretical analysis. Section 4.3 describes the proposed methodologies, and Section 4.4 provides analytical analysis of the proposed schemes. By presenting the performance evaluation of the proposed schemes and comparing their results with existing benchmark schemes in Section 4.5 and Section 4.6, we summarized this chapter in Section 4.7.

4.2 Problem Statement and Validation

We observe few shortcomings in the 6TiSCH-MC standard, which motivate us to propose a better scheme for the faster formation of 6TiSCH network. In this section, we describe

the our observations one by one, followed by their validations using numerical analysis as follows:

4.2.1 Starvation due to EB priority

Any node having packets to transmit, at first, participates in channel contention to get access in shared cells. The node performs random backoff once it encounters busy channel. The problem occurs when an EB frame is generated during the node's backoff period. Due to the EB's highest priority over all other control packets, that node transmits its newly generated EB frame, although other control packets are waiting in its transmission buffer for a long time. In a dense network, winning a contention by a node is very low. Hence, the node has higher chance to go to backoff. As a result, other control packets like DIO, DIS, keep-alive, etc., suffer from starvation because of EB's highest priority. When a pledge waits for a DIO packet to complete its joining process, the pledge needs to wait for longer time duration due to above mentioned problem. Hence, the total joining time of the pledge increases, which in turn affects the overall network formation time. Figure 4.1 pictorially describes the packet level starvation problem that happens within a node. The figure shows that a DIO packet is replaced by a newly generated EB frame during the backoff time of an already joined node.

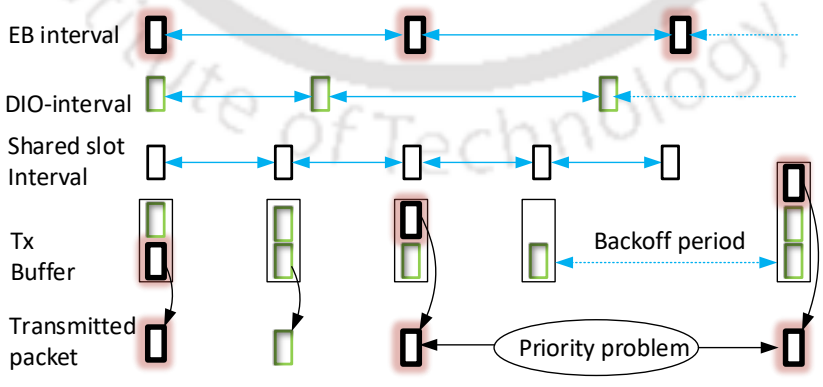


Figure 4.1: *Starving DIO packet due to EB's highest priority*

4.2.2 Negative Effect of Trickle Algorithm

In 6TiSCH network, the rate of transmitting DIO packet is decided by the Trickle Algorithm (TA) [76] in network layer. The main use of TA is to save energy consumption by transmitting minimum routing information packets. Initially, DIO transmission interval starts with a minimum duration of I_{dio}^{min}. Until the interval reaches its maximum value of I_{dio}^{max}, the interval duration becomes double at the end of each interval. If any inconsistency occurs, such as the parent node changing its rank, then the node resets its Trickle interval to I_{dio}^{min}. Other than that, the interval is not changed. Hence, the interval of DIO in TA is unaffected by a pledge's JRQ frame. Even though a node transmit DIS request for DIO packet, in the worst case (i.e., highly congested network), a pledge needs to wait for at least I_{dio}^{max} time to get a DIO packet from its parent. Moreover, after generating a DIO packet, a joined node may encounter packet level starvation as explained in Section 4.2.1. This scenario leads to higher joining time to a pledge, which is waiting for a DIO packet from its parent node. Figure 4.2 describes the effect of the Trickle Algorithm. Here, a pledge needs to wait for the completion of the maximum Trickle interval of its parent to receive a DIO packet.

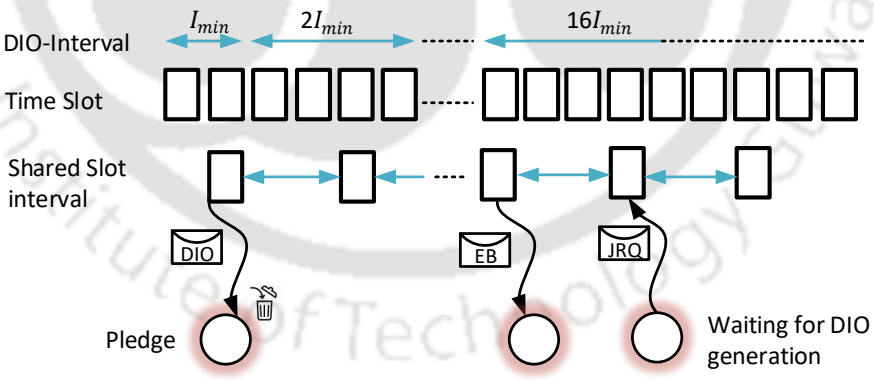


Figure 4.2: *Negative effect of Trickle Algorithm*

The works such as [73, 83–85] have been published to improve the performance of RPL. For this, the authors tried to transmit the DIO packet immediately for faster formation of DODAG. These works mainly tune various parameters such as node wise varied Trickle *listen-only* time and *redundancy constant* and perform encapsulation of DIO packet into the

EB frame. However, all these works did not consider the quick generation of a DIO packet when it required during network formation. These works follow the same DIO generation interval set by the TA. Hence, following these approaches, a pledge might need to wait for maximum Trickle interval time (I_{dio}^{max}) to get a DIO packet. The problem of long waiting for DIO packet during its requirement persists in these existing works.

4.2.3 Delayed Channel Access for Control Frames

We observe that the packet level starvation and negative effect of TA problem occur inside a node, and thus they can be solved locally within a node. On the other hand, the 6TiSCH-MC uses random channel access protocol (i.e., CSMA/CA) to transmit control packets in shared cells. Although random channel access provides fairness to all the contending nodes in the long run or steady-state conditions, it does not guarantee that a node can access the channel within a minimum time period. Again, with the increasing number of joined nodes, uncertainty in accessing shared channel also increases. In a dense network, a joined node may need to wait for several shared cells to transmit its control packet, which is already present in its buffer. Furthermore, nodes perform random backoff once they encounter busy channel. When the congestion increases, the possibility of getting into the backoff stage also increases. This results in a delay in the transmission of control packets, and sometimes, packets are discarded after the maximum number of re-tries. Because of this reason, a pledge has to wait for long time to get the required control packets. This increases joining time for pledges, which increases the overall network formation time. None of the existing works considered this problem during network formation. As this problem occurs within a network, therefore, an inter-node solution helps in reducing the overall network formation time of the 6TiSCH network.

4.2.4 Analytical Validation

For validation of our problem statements, we use the analytical model designed for the analysis of 6TiSCH-MC standard in our previous chapter i.e., Chapter 3 (Section 3.2) to

validate our problem statements one by one. However, for completeness of this chapter, we include the brief summary of the model here also.

During network formation, each pledge proceeds through four states. Figure 4.3 shows the different states of the Markov model, including an absorbing state. The *absorbing state* indicates that the pledge has joined with the existing network successfully.

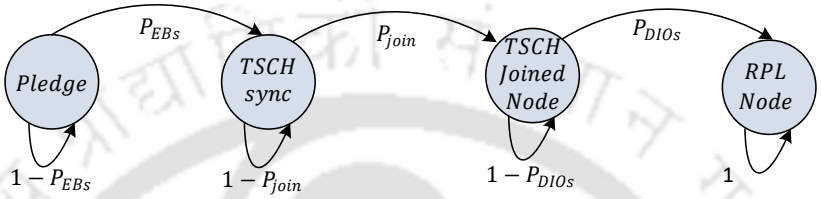


Figure 4.3: *Markov Chain model of node joining process*

Using the Markov model, the *average number of slotframes* (ASF) requires to reach the final absorbing state can be computed as follows,

$$\begin{aligned} ASF &= \frac{1}{P_{EB_S}} + \frac{1}{P_{join}} + \frac{1}{P_{DIO_S}} \\ &= \frac{1}{P_{EB_S}} + \left(\frac{1}{P_{JRS_S}} + \frac{1}{P_{JRQ_S}} \right) + \frac{1}{P_{DIO_S}} \end{aligned} \tag{4.1}$$

Now, to get the actual value of ASF, we need to calculate the values of P_{EB_S}, P_{DIO_S}, P_{JRS_S}, and P_{JRQ_S}. Let P_{eb}, P_{dio}, P_{jrq}, P_{jrs} are the transmission probabilities of EB, DIO, JRQ and JRS frames/packets in a slotframe respectively. P_{loss} is the packet loss probability and N_c is the total number of channels used in the network. The probability of receiving an EB frame successfully in a shared cell by a pledge is computed as,

$$P_{EB_S} = \frac{\sum_{n}^{M-n} nP_{eb}(1-P_{eb})^{n-1}(1-P_{dio})^{n-1}}{(1-P_{jrs})^{n-1}(1-P_{jrq})^{M-n}(1-P_{loss})P(N=n)} \tag{4.2}$$

Considering that EB frame has higher priority over DIO packet, the probability of transmitting a DIO packet successfully in a shared cell is computed as,

$$P_{DIO_S} = \sum_{n}^{M-n} nP_{dio}(1-P_{dio})^{n-1}(1-P_{jrs})^{n-1} (1-P_{eb})^{n-1}(1-P_{jrq})^{M-n}(1-P_{loss})P(N=n) \tag{4.3}$$

Likewise, the probability of transmitting a JRQ frame and receive a JRS frames successfully in a shared cell is also computed as,

$$P_{JRQ_S} = \sum_{n}^{M-n}(M-n)P_{jrq}(1-P_{jrq})^{M-n-1}(1-P_{eb})^{n} (1-P_{dio})^{n}(1-P_{jrs})^{n}(1-P_{loss})P(N=n) \tag{4.4}$$

$$P_{JRS_S} = \sum_{n}^{M-n} nP_{jrs}(1-P_{jrs})^{n-1}(1-P_{eb})^{n} (1-P_{dio})^{n}(1-P_{jrq})^{M-n}(1-P_{loss})P(N=n) \tag{4.5}$$

Now, to compute Equations (4.2), (4.3), (4.4), and (4.5) we should know the transmission probabilities of EB and DIO in a shared cell. The probability P_{eb} can be calculated as,

$$P_{eb} = \frac{L}{I_{eb}} \tag{4.6}$$

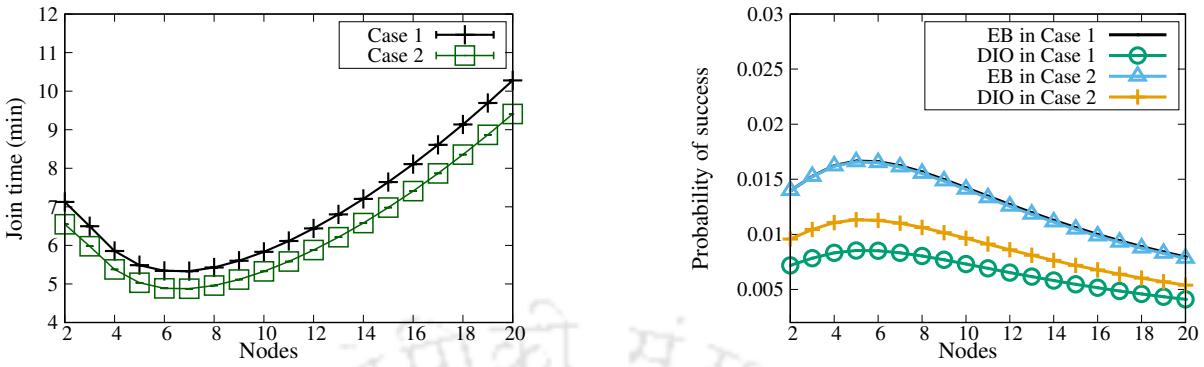
where L is the slotframe length and I_{eb} is the beacon interval. Note that the value of I_{eb} is always greater than L. Otherwise, EB frame will always replace the other control packets in the transmission buffer because of its highest priority [45]. We compute the probability P_{dio} following the procedure described in [45] as follows,

$$P_{dio} = (1-P_{eb})\frac{\begin{array}{c}2^{N_D}(1-P_r)^{N_D}min(\frac{L}{2^{N_D}I_{dio}^{min}},1)+ \\ \sum_{i=0}^{N_D-1}P_r2^i(1-P_r)^i min(\frac{L}{2^i I_{dio}^{min}},1)\end{array}}{P_r+2^{N_D}(1-P_r)^{N_D}+\sum_{j=1}^{N_D-1}P_r2^j(1-P_r)^j} \tag{4.7}$$

$$AJT = PJT + ASF \times L \tag{4.8}$$

where PJT is the average joining time of the parent node of a pledge in multi-hop network. Now, to validate packet level starvation, we compare the results of network formation method considering EB frame has the highest priority in one case (say, Case 1), and in the other case (say, Case 2), we consider all the control packets have equal probabilities. The probability of transmitting a DIO packet does not depend on the existence of an EB frame in node's buffer for Case 2. So, the probability of transmitting a DIO packet in a shared cell in Case 2 can be derived from Equation (4.7) as follows,

$$P_{dio} = \frac{2^{N_D}(1-P_r)^{N_D}min(\frac{L}{2^{N_D}I_{dio}^{min}},1)+\sum_{i=0}^{N_D-1}P_r2^i(1-P_r)^i min(\frac{L}{2^i I_{dio}^{min}},1)}{P_r+2^{N_D}(1-P_r)^{N_D}+\sum_{j=1}^{N_D-1}P_r2^j(1-P_r)^j} \tag{4.9}$$

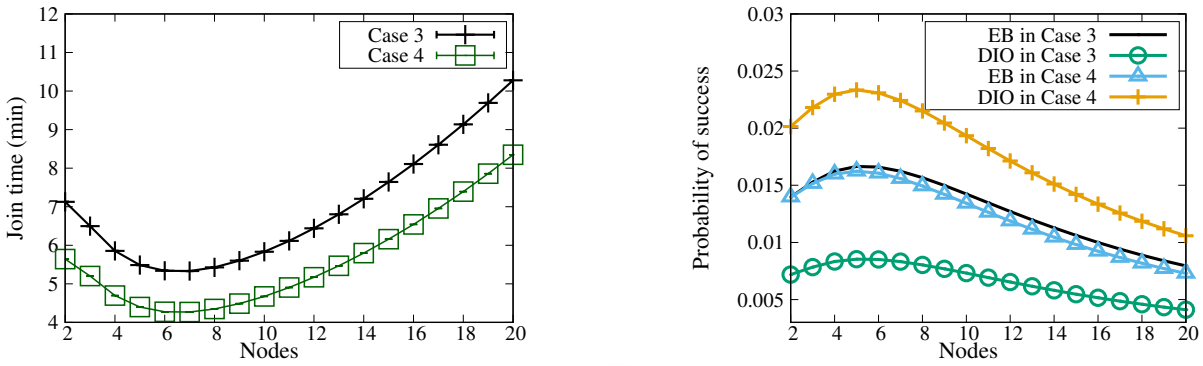


(a) *Node join time* **(b)** *Probability of success*

Figure 4.4: *Comparison on node joining time and the success probabilities of different control packets. (Case 1: EB has highest probability, Case 2: EB has equal probability with others)*

Now, let us consider a network with the following given values: $N_C = 16$, $P_{loss} = 0.2$, $I_{eb} = 4 * L$, $T_i = 10\ ms$, $I_{dio}^{min} = 8\ ms$, $L = 101\ timeslots$, $N_D = 16$ and $P_r = 0.2$. We plot a graph in Figure 4.4 considering both Case 1 and Case 2. The graph shows that the overall joining time of the network is less in Case 2. The reason is shown in Figure 4.4b. It shows how the probability of successfully transmitting a DIO packet in a shared cell increases when EB has equal priority with other control packets. The increase in success probability of DIO packet decreases the average network joining time of the pledges in the network. On the other hand, considering EB frame has the highest priority (Case 1), less successful transmission probability of DIO packet can be seen as EB frame suppresses the transmission of DIO packet because of its highest priority. This results in increasing joining time of the nodes.

To validate the problem exist in TA, we use fixed DIO-interval, i.e., $P_{dio} = .03$ instead of using TA based interval, and plot the results in Figure 4.5 along with TA based DIO-interval. In the figure, results using TA based DIO-interval and fixed DIO-interval is shown as Case 3 and Case 4, respectively. The figure shows that there is significant improvement in node joining time when we do not allow TA to update the DIO packet transmission interval. Figure 4.5b shows that the probability of successfully transmitted DIO control packet in a shared cell can be improved without using TA. TA decreases the number of transmitted DIO

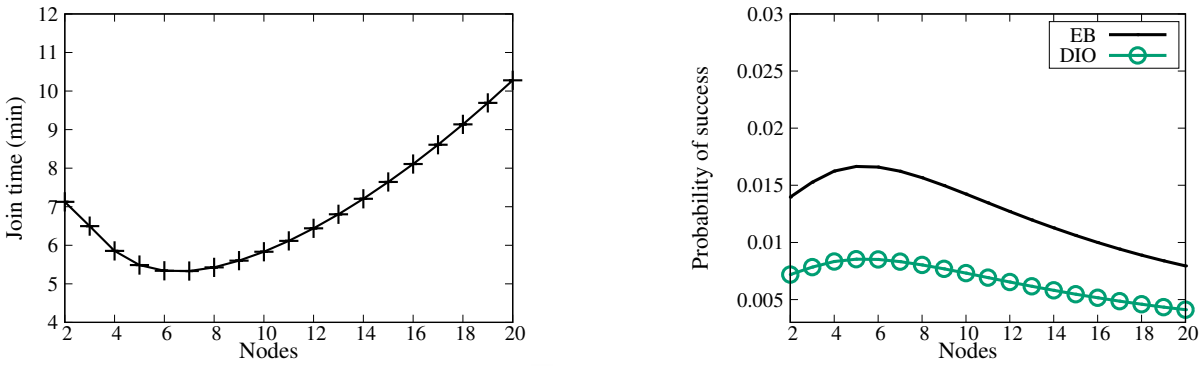


(a) *Node join time* **(b)** *Probability of success*

Figure 4.5: *Comparison on node joining time and the success probabilities of different control packets. (Case 3: Trickle Algorithm based DIO-interval, Case 4: fixed DIO-interval)*

packets in a stable network by increasing DIO generation interval to save energy. However, the decreasing number of transmitted DIO packets harm the reception probability of DIO packet in a shared cell. Hence, pledges have to wait for longer time for DIO packets, which in turn increases the overall network joining time. Figure 4.5 shows how TA impacts on the network joining time (Case 3). Note that a stable network means there are no changes in the network topology at the current state.

To validate the third problem statement, we plot the node joining time with respect to the increasing number of joined nodes in Figure 4.6. This figure shows the impact of increasing uncertainty of shared channel access due to the increasing number of nodes on network joining time, considering TA based DIO packet generation. Initially, when the number of nodes is small (e.g., 2-4 nodes), the pledges need to wait for more amount time because of infrequent transmission of control packets in the network. Therefore, adding a few more nodes (e.g., 6-8 nodes) helps the pledges join the network in less time because of more number of transmission of control packets by more nodes without congesting the shared channel. However, further increment of nodes (e.g., 9-20 nodes) increases the congestion in the shared channel, which in turn increases the uncertainty in accessing the shared channel. This reduces channel access probability, which increases the joining time of the pledges as pledges need to wait more time for control packets. Further, Figure 4.6 shows that the successful transmission probability for both the EB frame and DIO packet in a slotframe



(a) *Join time comparison* **(b)** *Probability of success*

Figure 4.6: *Impact of number of joined node on average network formation time and success probabilities of control packets*

decreases with the increasing number of nodes.

4.3 Opportunistic Transmission of Control Packet (OTCP)

In brief, none of the existing works considered the above mentioned three problems in the context of 6TiSCH network formation. Therefore, in this chapter, we try to overcome all these shortcomings with the aim of faster formation of 6TiSCH network. In 6TiSCH network, the EB frame is used to broadcast network existence information. Furthermore, EB frame helps in maintaining synchronization in a network. For these reasons, the highest priority is given to EB frame among all the control packets used in 6TiSCH network. Therefore, it would not be suitable for a 6TiSCH network to consider all types of control packets with equal priority. Again, Trickle Algorithm (TA) based DIO-interval is used to save node energy in RPL based LLN. TA suppresses the unnecessary transmission of frequent DIO packet to save energy. Hence, we cannot use fixed rate DIO interval in an LLN because of its energy constraint. Therefore, to deal with the above mentioned first two problems, we need to dynamically assign priority to control packets based on their network requirement and maintain the rate of transmitting routing information based on the pledge's requirement. Again, to deal with the uncertainty in accessing the transmission channel, there should be some mechanism by which the node with an important packet(s) in its transmission buffer can access the shared channel before the nodes that do not have important control packets

to transmit immediately.

We propose two schemes to solve all the above three problems. The first one, the *opportunistic priority alternation and rate control* (OPR) scheme, solves the problem of packet level starvation and negative effect of TA. The second one, the *opportunistic channel access* (OCA) scheme is used to provide quick channel access to the nodes with urgent packets. Both the schemes together we called *opportunistic transmission of control packet* (OTCP). The proposed schemes are described one by one in the below subsections.

4.3.1 Opportunistic Priority Alternation and Rate Control

The OPR scheme changes the priorities of control packets depending upon network requirements and also opportunistically increases the rate of transmission of DIO packet when it is required. The main purpose of EB frame is to broadcast network availability information and maintain global synchronization within a network. Therefore, in stable networks, we always consider EB must have the highest priority. However, in few cases (e.g., when joined node receives JRQ frame), transmission of DIO packet is more critical than EB frame. In such cases, the proposed scheme gives the highest priority to DIO packet for reducing overall network formation time. Again, the TA-based DIO-interval affects the performance of a node in terms of energy consumption. So, the existing TA is slightly modified such that it resets its interval when a joined node receives JRQ frame from a pledge. It enables quick generation of DIO packet during network joining process instead of waiting for DIS request. Algorithm 2 describes the OPR scheme.

4.3.2 Opportunistic Channel Access Scheme

The OCA scheme provides a solution to any node having an urgent packet in its transmission buffer. The DIO packet generated by a joined node after receiving a JRQ frame or DIS packet from its child node is considered as an urgent packet. It is because the child node(s) require DIO packet immediately to complete their joining process in less time. The node which contains such an urgent (DIO) packet is considered as an urgent node. As all the nodes

Algorithm 2 Opportunistic Priority Alternation and Rate Control

```
1: if the current Link_type is Shared then
2:     if got the channel access to transmit then                ▷ Tx mode
3:         Transmit highest priority control packet (EB/DIO) from the buffer
4:         Set highest priority to EB (default value)
5:         Set flagOCA to FALSE                                  ▷ for Algorithm 3
6:     end if
7:     if received a packet then                                 ▷ Rx mode
8:         if received packet is JRQ frame then
9:             transmit Join Response (JRS) frame
10:            Set highest priority to DIO
11:            if no DIO packet is available in transmission buffer then
12:                Reset the Trickle interval for generation of DIO
13:                Set flagOCA to TRUE                           ▷ for Algorithm 3
14:            end if
15:        end if
16:        if received packet is DODAG Information Solicitation (DIS) request for DIO
   then
17:            Set highest priority to DIO
18:            Set flagOCA to TRUE                               ▷ for Algorithm 3
19:        end if
20:    end if
21: end if
```

present in a network try to access the shared channel simultaneously, there is a requirement for an inter-node solution for opportunistic channel access associated with random channel access. A node does not know whether the neighbor nodes have urgent packets to transmit or not. Therefore, we need to modify the default channel access mechanism (i.e., CSMA/CA) for providing opportunistic access to nodes. In this modified scheme, we propose that the nodes, which have an urgent packet to transmit, use fixed Contention Window (CW) even if they sense busy channel during Clear Channel Assessment (CCA). In other words, the CW size does not increase after an unsuccessful attempt in case of urgent packet transmission. On the other hand, the normal nodes, which do not have any urgent packet to transmit, use variable CW when they encounter busy channel. In other words, normal nodes calculate their CW size using the default *backoff algorithm*. We noted that in 6TiSCH network, backoff is calculated in terms of number of shared cell. Algorithm 3 describes the procedure of opportunistic channel access.

Algorithm 3 Opportunistic Channel Access

1: Initialization: NB (number of backoff stages) =0; BE (backoff exponent) =$macMinBE$; ω (random backoff counter) =0
2: **if** the current $Link_{type}$ is $Shared$ AND ω is not 0 **then**
3: ω is decremented by 1
4: **else if** the current $Link_{type}$ is $Shared$ AND ω *is* 0 **then**
5: Node perform Clear Channel Assesment (CCA)
6: **if** channel is free **then**
7: Transmit the frame;
8: BE=$macMinBE$;
9: NB=0;
10: Exit
11: **else if** (NB+1) greater than $macMaxFrameRetries$ **then**
12: Drop the frame and Exit
13: **else if** flagOCA is FALSE **then** ▷ i.e., packet is not urgent and flagOCA is obtained from Algorithm 2
14: BE=min(BE+1, $macMaxBE$)
15: **end if**
16: Increment of NB by unity;
17: Select a random backoff counter $\omega \in [0, 2^{BE}\text{-}1]$
18: **end if**
19: Goto step 2

4.4 Theoretical Analysis of OPR and OCR

In this section, we perform theoretical analysis of the proposed schemes. As OPR solves the intra-node problem and OCA solves the inter-node problem within a network, therefore, separate theoretical modelings of both the schemes are performed and shown in the below subsections.

4.4.1 Opportunistic Priority Alternation and Rate Control

OPR assigns higher priority to a DIO packet than an EB frame when there is a need for DIO in the network. However, in a stable network, EB frame always has the highest priority over all other control packets. Therefore, the probability of transmitting an EB frame in a shared cell by our proposed OPR scheme i.e., P_{ebN} also depends on the requirement and availability of DIO packet in the network. Hence, P_{ebN} can be calculated as follows,

$$P_{ebN} = (1 - P_{JRQ_S})P_{eb} + P_{JRQ_S}(1 - P_{dioG})P_{eb} \tag{4.10}$$

where the term $(1 - P_{JRQ_s})P_{eb}$ denotes that JRQ frame is not received, which indicates no urgent requirement of DIO packet. So, P_{ebN} depends on the probability of generation of EB frame in a slotframe i.e., P_{eb}. Furthermore, $P_{JRQ_s}(1 - P_{dioG})P_{eb}$ denotes that JRQ frame is received (i.e., urgent requirement of DIO packet) but DIO packet is not available in node's buffer. Here, P_{dioG} denotes the probability of generating a DIO packet in a slotframe and it is calculated using the Equation (4.15). Hence, EB transmitting probability depends on the generation of an EB frame in a slotframe. Note that the value of P_{ebN} always will be in between $[0, 1]$. This is because, the values of P_{ebN} depend on the values of P_{JRQ_S}, P_{dioG} and P_{eb}. The values of these variables always remain in between $[0, 1]$ as these are the transmitting probabilities of different control packets in a shared cell.

Again, the generation of DIO packet is controlled by our modified TA. Hence, the new probability of transmitting a DIO packet in a shared cell (i.e., P_{dioN}) depends on the reception of JRQ frame from a pledge and the network status. We follow the similar *Semi-Markov Process* (SMP) approach mentioned in [45] to calculate the value of P_{dioN}. The main difference between the approach in [45] and our approach is that the consideration of the non-deterministic JRQ frame along with the network status. The modified TA of a joined node resets its current interval to minimum value I_{dio}^{min} during network inconsistency as well as upon receiving a JRQ frame. Figure 4.7 shows the states of Markov process based on modified Trickle algorithm. In the figure, the probability P_r denotes the inconsistency in

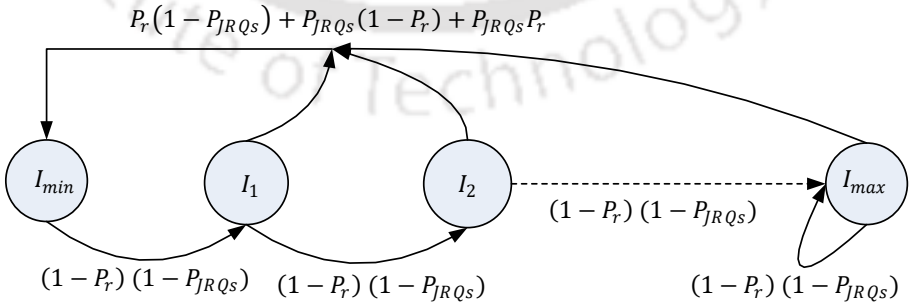


Figure 4.7: *State transition diagram of DIO generation interval in modified Trickle Algorithm*

networks. P_{JRQ_S} denotes the probability of successfully receiving a JRQ frame in a slot-

frame. Therefore, when neither any network inconsistency nor any JRQ frame is received, Trickle state moves from its I_i state to I_{i+1} state. When Trickle state reaches its final state I_{dio}^{max}, it remains in the same state until there is an event to reset it. In brief, the modified Trickle algorithm resets its DIO-interval to minimum in three conditions:

- When there is inconsistency in the network but does not receive any JRQ frame, i.e., $P_r(1 - P_{JRQ_S})$
- When there is no inconsistency in the network but receives JRQ frame, i.e., $(1 - P_r)P_{JRQ_S}$
- When there is inconsistency in the network and also receives JRQ frame, i.e., $P_r P_{JRQ_S}$

Therefore, the TA resetting probability can be written as, $P_r(1-P_{JRQ_S})+(1-P_r)P_{JRQ_S}+P_rP_{JRQ_S} = P_r + P_{JRQ_S} - P_rP_{JRQ_S}$. Considering all these conditions, the probability matrix of the Markov process can be written as follows,

$$M = \begin{bmatrix} x & y & 0 & ... & 0 \\ x & 0 & y & ... & 0 \\ ... & ... & ... & ... & ... \\ x & 0 & 0 & ... & y \end{bmatrix}$$

where $x = P_r + P_{JRQ_S} - P_rP_{JRQ_S}$ and $y = (1 - P_r)(1 - P_{JRQ_S})$. Now, having the sojourn time as T_i in state i, the stationary distribution of the state in the embedded Markov Chain can be computed as follows,

$$\pi_i = \frac{\mu_i T_i}{\sum_{j=0}^{N} \mu_j T_j}$$

where $\mu = P\mu$. Now, applying this in our Markov process, we get,

$$\pi_0 = \frac{x}{x + 2^{N_D} y^{N_D} + \sum_{j=1}^{N_D-1} x 2^j y^j} \tag{4.11}$$

$$\pi_i = \frac{x 2^i y^i}{x + 2^{N_D} y^{N_D} + \sum_{j=1}^{N_D-1} x 2^j y^j}, i \in [1, N_D - 1] \tag{4.12}$$

$$\pi_{N_D} = \frac{2^{N_D} y^{N_D}}{x + 2^{N_D} y^{N_D} + \sum_{j=1}^{N_D - 1} x 2^j y^j} \tag{4.13}$$

Again, the average probability of generating a DIO packet in a state i can be calculated as follows,

$$P(P_{dioG} \mid I_i) = \begin{cases} 1 & \text{if L} \geq I_i \\ \frac{L}{I_i} & \text{otherwise} \end{cases} \tag{4.14}$$

where I_i is the duration of state i. Hence, the average probability of generating a DIO packet in a slotframe i.e., P_{dioG} can be calculated as follows,

$$\begin{aligned} P_{dioG} &= \sum_{i=0}^{N_D} \pi_i P(P_{dioG} \mid I_i) \\ &= \frac{2^{N_D} y^{N_D} min(\frac{L}{2^{N_D} I_{dio}^{min}}, 1) + \sum_{i=0}^{N_D - 1} x 2^i y^i min(\frac{L}{2^i I_{dio}^{min}}, 1)}{x + 2^{N_D} y^{N_D} + \sum_{j=1}^{N_D - 1} x 2^j y^j} \end{aligned} \tag{4.15}$$

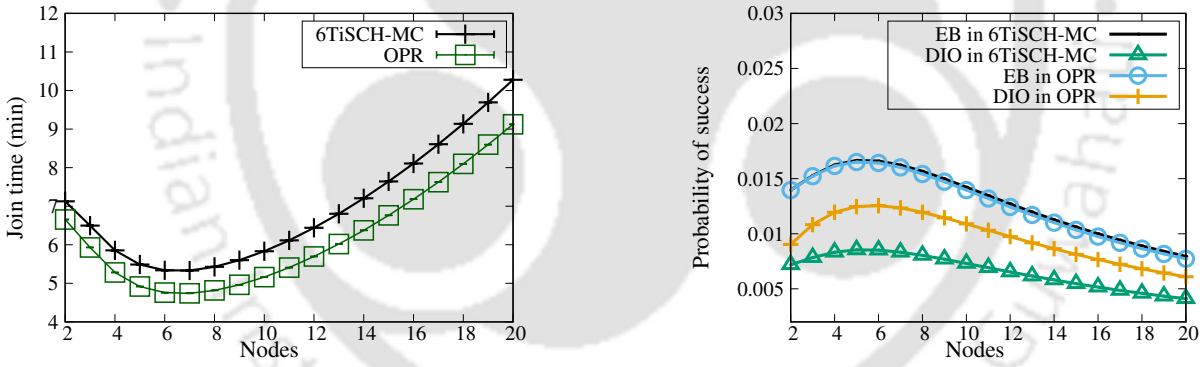
Hence, probability of transmitting DIO packet in a slotframe (i.e., P_{dioN}) can be computed as follows,

$$P_{dioN} = (1 - P_{JRQ_S})(1 - P_{eb}) P_{dioG} + P_{JRQ_S} P_{dioG} \tag{4.16}$$

Here, the term $(1 - P_{JRQ_S})(1 - P_{eb}) P_{dioG}$ denotes that a joined node does not receive JRQ frame and there is no EB frame available in its buffer, and hence, node can transmit DIO packet. $P_{JRQ_S} P_{dioG}$ denotes that node transmit its available DIO packet when it receives JRQ frame without conditioning availability of EB frame.

Now, by using the value of P_{ebN} and P_{dioN} in Equations (4.1 - 4.8), we can calculate the *average joining time* (AJT) of pledge. However, we notice that there exists a circular dependency among the Equations (4.8), (4.10), and (4.16). Therefore, an iterative approach is used to compute the steady state AJT of a pledge. At the beginning, only EB and DIO packets are transmitted, therefore, the initial values of P_{EB_s} and P_{JRQ_s} are calculated using the Equations (4.6) and (4.7) and taking the initial values of P_{jrq} and P_{jrs} as 0 to break the circular dependency. Now, assigning $P_{jrq} = P_{EB_S}$ and $P_{jrs} = P_{JRQ_S}$ and using Equations (4.4), (4.6), and (4.7), we calculate new value of P_{JRQ_S}. Using this calculated P_{JRQ_S} value, we calculate the transmission probability of EB frame P_{ebN} and DIO packet P_{dioN} in a

shared cell based on our proposed method. In the next step, we calculate the AJT by using the new values of P_{ebN} and P_{dioN} in the Equation (4.8). Simultaneously, we also obtain the value of P_{JRQ_S} using the Equation (4.4). Using the new value of P_{JRQ_S}, again, we calculate new values of P_{ebN} and P_{dioN}. Again, this newly calculated P_{ebN} and P_{dioN} are used to calculate the AJT and P_{JRQ_S}. The processes is repeated until a steady average joining time is obtained. The final obtained results are plotted in Figure 4.8. Note that in this calculation process, the values of other network parameters like I_{eb}, P_r, etc., are taken same as mentioned in Section 4.2.4. The Figure 4.8a shows the improvement in average joining time of a pledge of the proposed scheme over 6TiSCH-MC. And the Figure 4.8b shows the reason for this improvement. In the proposed scheme, the negative effect of EB's highest priority and Trickle algorithm are eliminated, and thus, there are improvements in success probabilities of different control packets. As a result of this, average joining time of a network is reduced.



(a) *Node join time* **(b)** *Probability of success*

Figure 4.8: *Comparison on node joining time and the success probabilities of different control packets*

4.4.2 Opportunistic Channel Access

OCA provides higher chance to access the shared transmission channel to the nodes who have urgent packets to transmit by allowing them to use fixed CW even if they sense busy channel during CCA. The size of the fixed CW is smaller than the CW of other normal nodes that have experienced busy channel. For analyzing OCA scheme, we consider two

classes of nodes. One is the urgent priority class, which includes the nodes with an urgent packet to transmit. The other priority class contains the normal nodes which do not have urgent packet to transmit. Xiao *et.al.* [86] and Khatua *et.al.* [82] analyzed the random access MAC protocol with different priority classes of nodes. According to [86], the transmission probability τ_i of a node of priority class i in a generic slot can be computed as follows,

$$\tau_i = \frac{1 - {P_i}^{m+1}}{1 - P_i} \frac{1}{\sum_{j=0}^{m} \left[1 + \frac{1}{1-P_i} \sum_{k=1}^{W_{i,j}-1} \frac{W_{i,j}-k}{W_{i,j}}\right] {P_i}^j} \tag{4.17}$$

where m is the maximum retries, P_i denotes the probability of a node from priority class i senses busy channel, $W_{i,j}$ denotes the CW size of a node of priority class i in its j^{th} backoff stage. From the Equation (4.17), we can derive the transmission probabilities τ_u and τ_n of an urgent node and a normal node, respectively, as follows,

$$\tau_u = \frac{1 - {P_u}^{m+1}}{1 - P_u} \frac{1}{\sum_{j=0}^{m} \left[1 + \frac{1}{1-P_u} \sum_{k=1}^{W_{u,j}-1} \frac{W_{u,j}-k}{W_{u,j}}\right] {P_u}^j} \tag{4.18}$$

$$\tau_n = \frac{1 - {P_n}^{m+1}}{1 - P_n} \frac{1}{\sum_{j=0}^{m} \left[1 + \frac{1}{1-P_n} \sum_{k=1}^{W_{n,j}-1} \frac{W_{n,j}-k}{W_{n,j}}\right] {P_n}^j} \tag{4.19}$$

Note that in Equation (4.18) the value of $W_{u,j}$ remains same as we use fixed CW, whereas, in Equation (4.19), $W_{n,j}$ varies as $W_{n,j+1} = \alpha W_{n,j}$, where α is a CW incremental factor.

The value of P_u and P_n, i.e., collision probabilities of an urgent node and a normal node, can be computed as follows,

$$P_u = 1 - (1 - \tau_u)^{N_u - 1} (1 - \tau_n)^{N_n} \tag{4.20}$$

$$P_n = 1 - (1 - \tau_n)^{N_n - 1} (1 - \tau_u)^{N_u} \tag{4.21}$$

where N_u and N_n are the number of urgent nodes and normal nodes, respectively. Again, the Equations (4.18), (4.19), (4.20), and (4.21) have circular dependency among themselves. Therefore, once again an iterative approach is used to solve this dependency as mentioned in the previous section. For this purpose, the initial values of τ_u and τ_n are assumed randomly

as 0.5. The other initialization parameters $W_{i,0} = 4$, $m = 5$ are taken same for both the classes of nodes. For normal node, we take $\alpha = 2$. In the first step, we calculate the P_u and P_n using τ_u and τ_n. This calculated P_u and P_n values are used to calculate new values of τ_u and τ_n. Again, this new values of τ_u and τ_n is used to further calculate the new values of P_u and P_n. We continue this process, until we reach to a steady state, i.e., there is no significant change in the values of τ_u and τ_n.

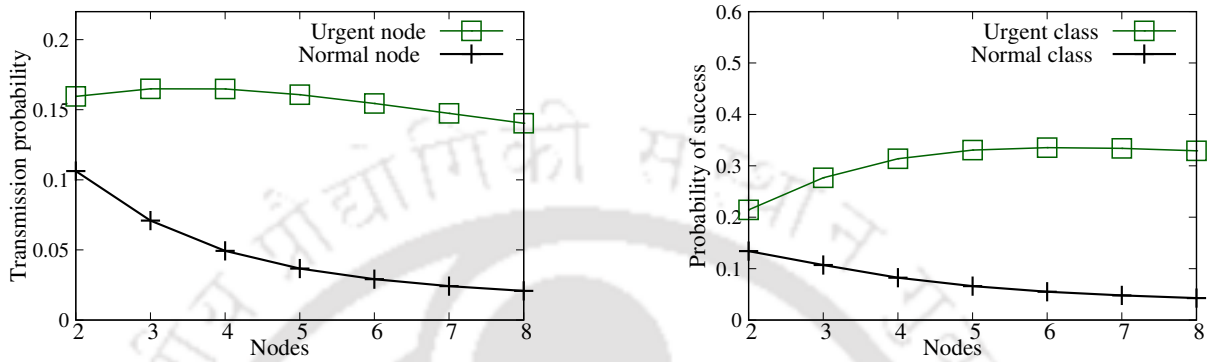
Let P^u_{succ} and P^n_{succ} denote the probabilities that a successful transmission occurs in a slot from the class of urgent node and normal node, respectively. The value of P^u_{succ} and P^n_{succ} can be computed as,

$$P^u_{succ} = N_u \tau_u (1-\tau_u)^{N_u - 1} (1-\tau_n)^{N_n} \tag{4.22}$$

$$P^n_{succ} = N_n \tau_n (1-\tau_n)^{N_n - 1} (1-\tau_u)^{N_u} \tag{4.23}$$

The ratio $\frac{P^u_{succ}}{P^n_{succ}}$ denotes the opportunistic channel access by the class of urgent nodes over normal nodes. The Propositions 4.4.1 and 4.4.2 prove that $\frac{P^u_{succ}}{P^n_{succ}} > 1$. This, in turn, confirms that the proposed scheme provides opportunistic channel access to urgent packet containing nodes. Figure 4.9a shows that an urgent node obtains higher transmission probability because of non-increasing small CW. And Figure 4.9b shows the opportunistic successful transmission probabilities of the urgent node class against the whole normal node class. In both the plots, the result obtained by the nodes belonging to normal class gets degraded with the increasing number of nodes. It is because when the number of nodes increases in the network, congestion also increases and it forces the nodes to increase their contention window length. It results in less transmission probability and probability of success. So, the lines are going downward in both the plots with the increasing number of nodes in the network. On the other hand, when we consider the nodes belonging to urgent class, these nodes do not increase their contention window even when they find the shared channel busy. Therefore, before achieving maximum throughput, the lines are slightly going upward until maximum throughput is reached by the network. However, after reaching the maximum throughput,

due to increasing number of nodes in the network, the nodes start contending heavily in a shared channel and it results in both less transmission probability and probability of success. Hence, the lines are going downward after reaching an apparent peak which results in concave nature of the plot.



(a) *Transmission probability* **(b)** *Probability of success*

Figure 4.9: *(a) Transmission probabilities of an urgent node and a normal node, and (b) successful transmission probabilities of whole urgent node class and normal node class*

Proposition 4.4.1. *After first backoff stage, OCA scheme provides* $\tau_u > \tau_n$ *always.*

Proof. From the Equation (4.17), by expanding the term $\sum_{k=1}^{W_{i,j}-1} \frac{W_{i,j}-k}{W_{i,j}}$ we get,

$$\begin{aligned} &= \frac{W_{i,j}-1}{W_{i,j}} + \frac{W_{i,j}-2}{W_{i,j}} + ... + \frac{W_{i,j}-(W_{i,j}-1)}{W_{i,j}} \\ &= \frac{W_{i,j}-1}{2} \end{aligned} \tag{4.24}$$

Hence, the value of τ is reciprocal to CW size. At the first stage, the values of $W_{u,1}$ and $W_{n,1}$ are same in Equations (4.18) and (4.19). Hence, we get same values for τ_u and τ_n, which, in turn gives same values for P_u and P_n. But in the later backoff stages say, stage 2, $i = 2$, $W_{n,2} > W_{u,2}$, $\tau_u = \tau_n$ and $P_u = P_n$ and all these values results in $\tau_u > \tau_n$ and $P_u < P_n$. This is because the term $\sum_{j=0}^{m} \left[1 + \frac{1}{1-P_i} \sum_{k=1}^{W_{i,j}-1} \frac{W_{i,j}-k}{W_{i,j}}\right] P_i^j$ in Equation (4.17) increases with larger values of CW. And the larger value of the term $\sum_{j=0}^{m} \left[1 + \frac{1}{1-P_i} \sum_{k=1}^{W_{i,j}-1} \frac{W_{i,j}-k}{W_{i,j}}\right] P_i^j$ reduces the value of τ_i and increases P_i. In the later backoff stages also $\tau_u > \tau_n$ for $W_{n,i} > W_{u,i}$. □

Proposition 4.4.2. *After first backoff stage, OCA scheme provides* $P^u_{succ} > P^n_{succ}$ *always.*

Proof. If we calculate the ration $\frac{P^u_{succ}}{P^n_{succ}}$ we get,

$$\begin{aligned}\frac{P^u_{succ}}{P^n_{succ}} &= \frac{N_u\tau_u(1-\tau_u)^{N_u-1}(1-\tau_n)^{N_n}}{N_n\tau_n(1-\tau_n)^{N_n-1}(1-\tau_u)^{N_u}} \\ &= \frac{N_u\tau_u(1-\tau_n)}{N_n\tau_n(1-\tau_u)} \end{aligned} \tag{4.25}$$

We already show that if $W_{n,i} > W_{u,i}$, then it results $\tau_u > \tau_n$. From the Equation (4.25), it is clear that $\frac{P^u_{succ}}{P^n_{succ}} > 1$ when $\tau_u > \tau_n$. So, $P^u_{succ} > P^n_{succ}$ □

4.5 Simulation based Evaluation of OTCP

4.5.1 Simulation Settings

The performance of the proposed schemes is investigated by using the Cooja simulator on Contiki-NG OS. The used parameters and their corresponding values are mentioned in Table 4.1. In our evaluation, the nodes are deployed in a fixed square size grid area where JRC i.e., RPL root is always placed at the top left corner of the grid. An increment in the number of nodes in a fixed area increases node density. As a result, congestion in the network also changes with varying numbers of nodes. The JRC initializes network formation by broadcasting network advertisement information carrying EB frame and routing information carrying DIO packet. The realistic *Multipath Ray-Tracer Medium* (MRM) channel model is used in simulations. This model provides various propagation effects such as multi-path, refraction, and diffraction in the physical channel. Along with the EB frame and various RPL control packets, few other control packets such as JRQ, JRS and keep-alive are also considered during our evaluation.

4.5.2 Benchmark Schemes and Performance Metrics

We compare our proposed scheme, i.e., *opportunistic transmission of control packets* (OTCP) with the existing benchmark schemes 6TiSCH-MC [40] and BS [46]. Note that the OTCP is

Table 4.1: *Simulation parameters*

Parameter	Value
Operating System	Contiki-NG
Mote type	Cooja Mote
Network size	16, 25, 36, 49 nodes
Timeslot length (L)	10 ms
Slotframe length (SF)	33, 101 *timeslots*
Number of channels	16
Propagation model	MRM
EB interval	4 s
RPL version	RPL Lite
RPL DIO-interval	Trickle Algorithm based
DIO interval (min-max)	$2^{10} - 2^{18}$, $2^{12} - 2^{20}$, $2^{13} - 2^{21}$, $2^{14} - 2^{22}$ ms
RPL Objective Function	MRHOF - ETX
RPL Redundancy constant	10
Keep-alive interval	30 s
Simulation duration	2 hrs

nothing but executing both the OPR and OCA schemes together. We also show how OPR and OCA schemes work individually and compare them with the benchmark schemes.

Mainly, three performance metrics such as *TSCH joining time*, *6TiSCH joining time*, and *average energy consumption* are considered in evaluation. The TSCH joining time denotes the maximum time required to receive at least one EB frame (i.e., TSCH synchronization time) by a pledge among all the pledges in the network. In other words, the maximum time required by the last pledge to reach to TSCH sync state of the Markov chain as shown in Figure 4.3. The 6TiSCH joining time denotes the maximum time required to receive at least one EB frame followed by exchanging of JRQ and JRS frames and receiving a valid DIO packet by a pledge among all the pledges present in the network. Similar to TSCH joining time, the 6TiSCH joining time is the maximum time required by the last pledge to reach to RPL node state of the Markov chain as shown in Figure 4.3. The energy consumption metric shows the average energy consumption by the last pledge admitted in the network. The procedure of energy consumption computation is described below. The energy consumption by a node is the summation of energy consumption during communication (E_{Comm}) and energy consumption during CPU active state (E_{CPU}).

Energy consumption of a node in communication (E_{Comm}) computed as,

$$E_{Comm} = \frac{Tx \times 18.8\ mA\ +\ Rx \times 17.4\ mA}{(RTIMER\_SECOND)} \times 3Volts \tag{4.26}$$

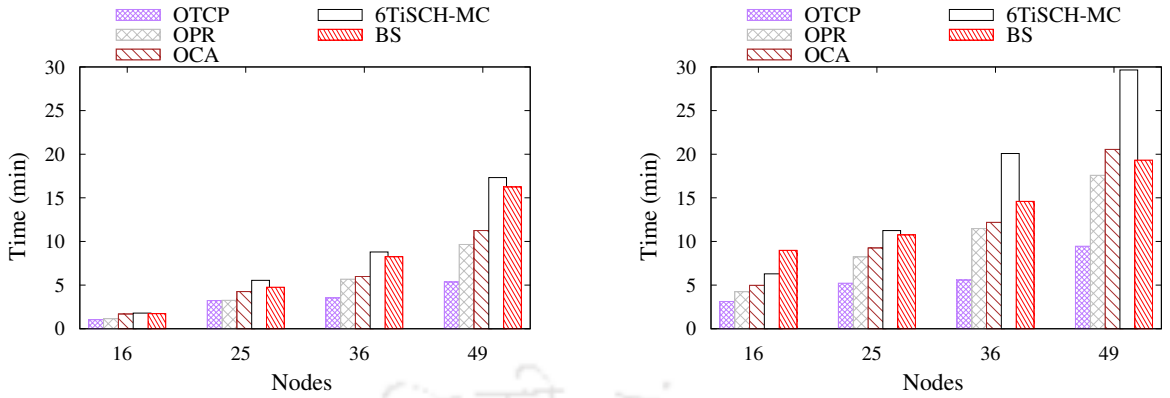
where `Tx` and `Rx` are the total number of ticks the radio has been transmitting and receiving mode, respectively, during network formation. The available Energest module obtains these values in Contiki-NG. The $RTIMER\_SECOND$ denotes the number of ticks per second, and its value is 32768. We consider the `CC2420` wireless chip, which provides IEEE 802.15.4 connectivity for estimating energy consumption. `CC2420` operates in low voltage of 2.1- 3.6 V, and it consumes 18.8 mA, 17.4 mA, and 1.8 mA during transmission (`Tx`), reception (`Rx`), and CPU active state, respectively. Similarly, the energy consumption of a node during CPU active period (E_{CPU}) is computed as,

$$E_{CPU} = \frac{T_{CPU} \times 1.8\ mA}{(RTIMER\_SECOND)} \times 3Volts \tag{4.27}$$

where T_{CPU} is the number of ticks the CPU has been in active mode during network formation.

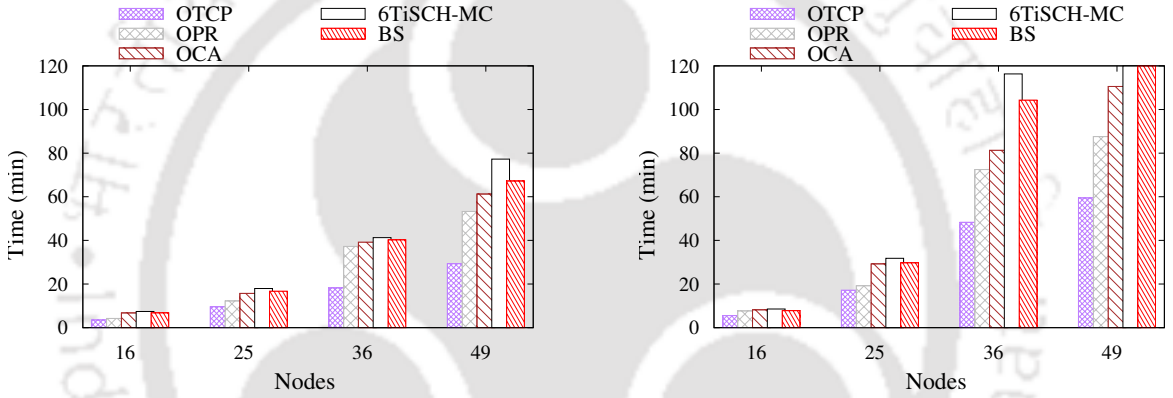
4.5.3 Simulation Results

Simulation results are shown in Figure 4.10 – 4.15. The Figure 4.10 – 4.12 show the results using different network sizes and Figure 4.13 – 4.15 show the results using different sets of DIO intervals. The evaluation is done using two different slotframe lengths (SF), such as 33 and 101 timeslots. The frequency of the occurrence of shared cells using $SF = 33$ is more than $SF = 101$ within a fixed time interval. Hence, nodes can transmit their control packets more frequently using $SF = 33$. Figure 4.10a and 4.10b show the TSCH joining time and Figure 4.11a and 4.11b show the 6TiSCH joining time, respectively, in varied size of grid topologies. From the result, we can see that the OTCP outperforms all the benchmark schemes. Furthermore, OPR and OCA also individually show better results than the 6TiSCH-MC and BS. This is because of priority alternation of control packets and providing sufficient routing information packet along with opportunistic channel access



(a) *TSCH joining time, SF=33* **(b)** *TSCH joining time, SF=101*

Figure 4.10: *Simulation results of TSCH joining time using different size of slotframe lengths (SF) and grid topologies (number of nodes)*

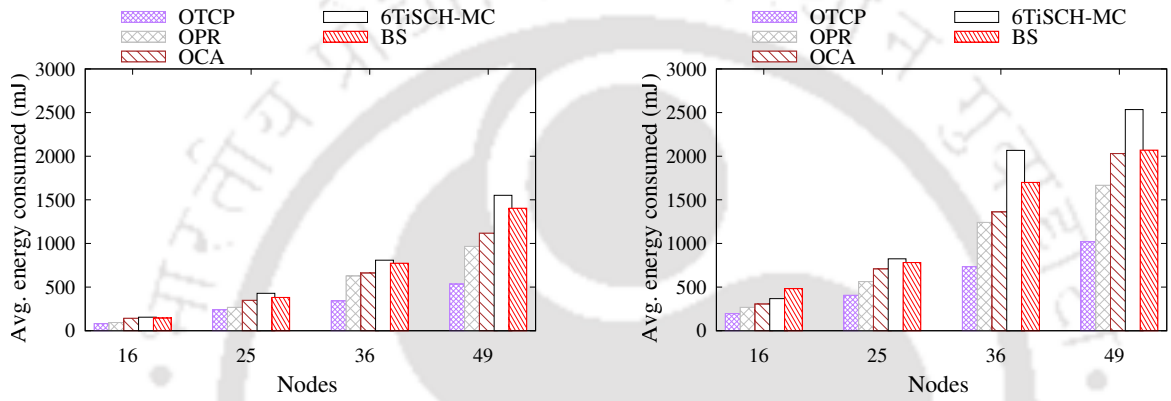


(a) *6TiSCH joining time, SF=33* **(b)** *6TiSCH joining time, SF=101*

Figure 4.11: *Simulation results of 6TiSCH joining time using different size of slotframe lengths (SF) and grid topologies (number of nodes)*

to the nodes which have generated DIO packet after receiving JRQ frame or DIS request. The proposed schemes allow the pledges to join the network in less time and transmit their control packets as soon as possible. This, in turn, helps the multi-hop distant pledges to get EB frame in less time. Again, using $SF = 33$, nodes perform better than using $SF = 101$. It is because of the frequent occurrence of shared cells using $SF = 33$. The joined nodes can transmit more control packets in less time, and hence, it improves the performance of the pledges in terms of joining time. Therefore, all the benchmark schemes and proposed schemes show similar performance when nodes are less (e.g., 16 and 25 nodes) using $SF = 33$. However, when the number of nodes increases, congestion in the shared cell also increases, resulting in longer joining time of the pledges. Significant effect of congestion can be seen

when more nodes are present in the network (e.g., 36 and 49 nodes) and slotframe length is 101 timeslots. It is because of the severe congestion in less occurring shared cells. All the nodes try to transmit their control packets in less number of available shared cells, which reduces the probability of getting the required control packets by the pledges. Hence, longer joining time can be observed due to high congestion in the shared cell. Note that the chances of occurring all the three problems, mentioned in Section 4.2, increase with the increasing congestion in the shared cell.

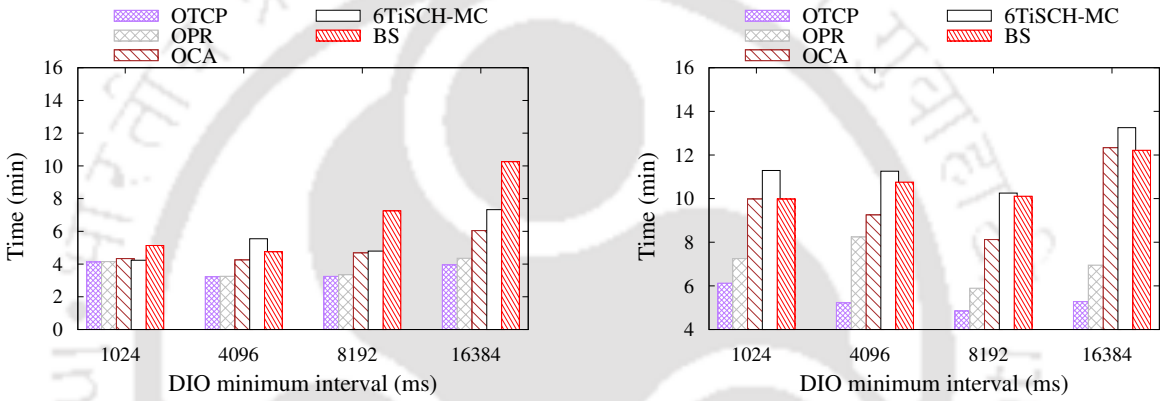


(a) *Average energy consumption, SF=33* **(b)** *Average energy consumption, SF=101*

Figure 4.12: *Simulation results of energy consumption using different size of slotframe lengths (SF) and grid topologies (number of nodes)*

Similarly, the proposed scheme OTCP performs better than the benchmark schemes in terms of energy consumption also. It is because of the less amount of channel scanning time as the pledges join TSCH network more quickly. The pledges need to keep their radio active before TSCH synchronization (during this period radio duty cycle of a pledge is 100%) to receive a valid EB frame, which in turn consumes more energy. Using the proposed scheme, the multi-hop nodes receive their EB frame in less time, which saves their energy. Figure 4.12a and 4.12b show comparison of the average energy consumption by a node using slotframe lengths $SF = 33$ and $SF = 101$. The nodes' energy consumption is almost the same in all the schemes when fewer nodes use more shared cells (in less slotframe length) to transmit their control packets. It is because of the less congestion in the frequently occurring shared cell. The pledges spend less amount of time to join both the TSCH and 6TiSCH networks. More energy consumption can be observed when there is severe

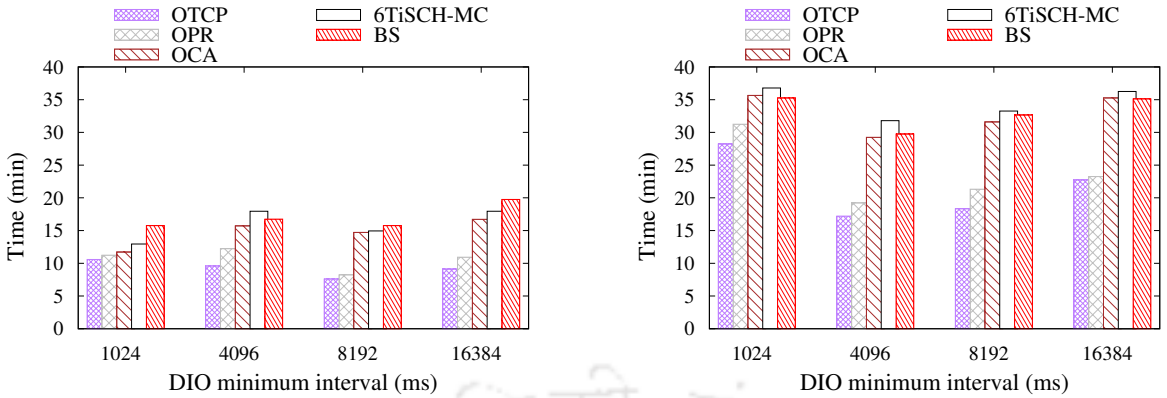
congestion in the shared cells, i.e. when the network size is large and slotframe length is more. OTCP shows better performance than the benchmark schemes because it allows the nodes to transmit their urgent packet even when the congestion in the network increases. Furthermore, OTCP provides sufficient routing information packets in a network during its formation and prioritizes the control packets based on their requirement. OTCP does not always allow the joined nodes to consider EB frame as the highest priority packet. Hence, the proposed scheme, OTCP performs significantly better than the benchmark schemes in terms of joining time and energy consumption even in a severely congested network.



(a) *TSCH joining time, SF=33* **(b)** *TSCH joining time, SF=101*

Figure 4.13: *Simulation result of TSCH joining time using different sets of DIO intervals and slotframe (SF) lengths in a* (5×5) *grid topology*

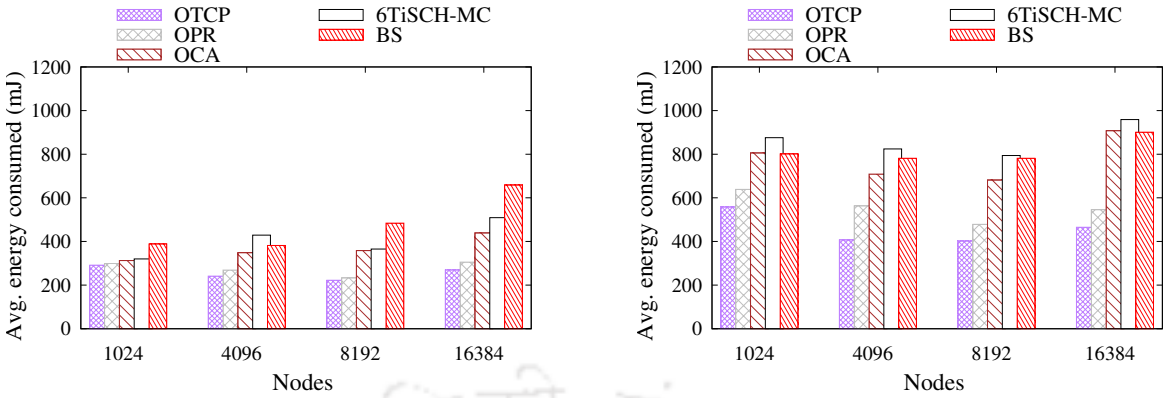
To provide sufficient routing information, the proposed scheme changes the behavior of the TA during network formation. Therefore, we further evaluate the performance of the proposed schemes using different sets of minimum and maximum DIO generation intervals. The results are shown using a 5×5 grid topology and different slotframe lengths such as $SF = 33$ and $SF = 101$ timeslots. The Figure 4.13a and 4.13b show the TSCH joining time and Figure 4.14a and 4.14b show the 6TiSCH joining time, respectively, using different sets of DIO intervals. In the graphs, it can be seen that the network formation time is high for very low as well as very high value of DIO interval. The frequent transmission of DIO packet congests the shared cells, which in turn results in longer joining time. And again, due to very less frequent transmission of DIO packet, the pledges need to wait for longer amount of time to receive a valid DIO packet, which again results in higher network



(a) *6TiSCH joining time, SF=33* **(b)** *6TiSCH joining time, SF=101*

Figure 4.14: *Simulation results of 6TiSCH joining time using different sets of DIO intervals and slotframe (SF) lengths in a* (5×5) *grid topology*

formation time. In case of BS scheme, due to less frequent transmission of both the EB frame and DIO packet, both the TSCH and 6TiSCH joining times increase even using $SF = 33$. It is because the pledges need to wait for more amount of time to get both the control packets. However, the same BS scheme performs better than 6TiSCH-MC when $SF = 101$ is used. It is due to more congestion in the shared cells in 6TiSCH-MC. All the joined nodes try to transmit their control packets in less number of shared cells. Whereas, in BS scheme, joined nodes transmit less number of EB frames than that in 6TiSCH-MC, and thus, less congestion in the shared cells. It can also be seen that the OPR performs almost similar to OTCP in terms of 6TiSCH joining time and energy consumption as their behaviors are almost the same except the opportunistic channel access in OTCP. At the same time, OCA performs similar to BS and 6TiSCH-MC in terms of 6TiSCH joining time and energy consumption when $SF = 33$ is used. It is because the OCA does not change the behavior of the Trickle Algorithm. Therefore, the number of routing packets generated in the network is similar to BS and 6TiSCH-MC. The longer joining time also affects in node's energy consumption. Figure 4.15a and 4.15b show the average energy consumption of a node during network formation using different DIO intervals. Due to higher joining time, nodes consume more energy when the value of minimum DIO interval equals 1024 ms and 16384 ms. Our proposed scheme OTCP outperforms all the existing benchmark protocols, as the pledges do not keep the radios active for more time in this scheme.



(a) *Average energy consumption, SF=33* **(b)** *Average energy consumption, SF=101*

Figure 4.15: *Simulation result of energy consumption using different sets of DIO intervals and slotframe (SF) lengths in a* (5×5) *grid topology*

It is worth mentioning that the proposed scheme transmits sufficient routing information packets by opportunistically altering the priority of control packets EB and DIO, resetting the Trickle interval and providing channel access by adjusting the backoff exponent. However, the Trickle algorithm has a DIO suppression mechanism that suppresses the transmission of generated DIO when a sufficient number of consistent DIOs are already transmitted by neighbors in the network. For this suppression mechanism, a *redundancy constant* k is used as a threshold on the number of overheard DIOs. If a joined node receives more than or equal number of k consistent DIOs from its neighbors in a Trickle interval, then the node suppresses its own DIO to save energy. It is pertinent to mention that the proposed scheme has added one more reason for Trickle reset. There are many other reasons for the same such as multicast DIS request, DODAG reset time reached, reception of inconsistent DIO, changed in DODAG version etc., as mentioned in the RPL [43]. The number of DIO transmission increases if a node performs Trickle reset frequently. The authors in [83] have already shown that the increasing number of DIO transmission leads to DIO suppression of neighbors, which in turn forces the neighbors to remain silent for longer amount of time, resulting in poor performance. However, the resetting of the Trickle interval is not a continuous event in the proposed scheme as it only occurs during network formation under the given condition as mentioned in Algorithm 2. Combining with the other modifications as mentioned in Algorithms 2 and 3, the proposed scheme OTCP suppresses less number of

DIO transmission. Table 4.2 shows a comparison of the number of DIO suppression and the number of Trickle reset observed in a network of 5×5 grid topology nodes. Simulation is performed for 60 min. It can be seen from Figure 4.14b that the 6TiSCH network is formed completely within 30 min and 20 min for $I_{dio}^{min} = 1024\ ms$ and 4096 ms, respectively, for the above mentioned simulation setup. Beyond this time period the network performs regular operations such as data transmission. Therefore, we have tabulated the DIO suppression results for the initial 30 min and next 30 min separately in the table.

Table 4.2: *Results of number of DIO suppression and Trickle reset count within* 60 *minutes of simulation using* 5×5 *grid topology and slotframe length=101. The results are presented as follows: <number of DIO suppression, Trickle reset count>*

Simulation Time	Scheme	$I_{dio}^{min} = 1024ms$			$I_{dio}^{min} = 4096ms$		
		k=1	k=3	k=10	k=1	k=3	k=10
First 30 min	6TiSCH-MC	<95, 916>	<14, 544>	<0, 1018>	<107, 870>	<14, 949>	<0, 907>
	BS	<67, 759>	<9, 413>	<0, 873>	<93, 771>	<8, 713>	<0, 857>
	OTCP (this work)	<45, 488>	<7, 280>	<0, 325>	<82, 549>	<6, 515>	<0, 325>
Next 30 min	6TiSCH-MC	<84, 707>	<14, 649>	<0, 760>	<63, 855>	<21, 773>	<0, 661>
	BS	<79, 471>	<11, 612>	<0, 756>	<58, 798>	<12, 551>	<0, 573>
	OTCP (this work)	<56, 277>	<9, 514>	<0, 378>	<29, 464>	<6, 352>	<0, 378>

The results are shown by varying DIO minimum interval (I_{dio}^{min}) and redundancy constant (k) as they also play role in DIO suppression [83]. The results show that the proposed scheme suppresses less number of DIOs as well as performs less number of Trickle reset compared to that in the benchmark schemes. Please note that if the benchmark schemes are used, the already joined nodes reset the Trickle interval more frequently due to more multicast DIS requests sent by other nodes. One of the reasons for transmitting multicast DIS request by a node is that when the node does not hear anything from its parent for a long period of time. It is already mentioned that, in the benchmark schemes, nodes need to wait for longer amount of time on average to get DIOs from their parents, so the number of multicast DIS transmission is more, which increases the Trickle reset count as well. Thus, it results in a higher value of suppression count as shown in the table. On the other hand, in the proposed scheme, the Trickle reset count is less due to less multicast DIS request. This is because of the opportunistic transmission of DIOs by which the waiting time to get a DIO packet for the nodes actually reduces. Furthermore, the number of DIO transmission is also less as

the number of Trickle reset count is less. It further results in less congestion in shared cell and also saves energy by transmitting less number of DIOs.

Finally, quick reception of DIO packet helps the pledges in finding the best route or new route immediately. This, in turn, reduces the possibility of network inconsistency. Hence, the proposed scheme helps in best route selection or multiple route selections during (or after) network formation by allowing opportunistic transmission of DIO packets. Additionally, the proposed schemes help the multi-hop distance pledges to join the network in less time by quickly providing sufficient routing information to the nodes.

4.6 Testbed Evaluation of OTCP

4.6.1 Testbed Experimental Setup

The implementation of the OTCP scheme in Contiki-NG has been flashed on the `M3` nodes in FIT IoT-LAB. We use two different topologies deployed in Grenoble locations for testing our proposed scheme in the testbed. The used topologies are shown in Figure 4.16. Total 31 `M3` nodes are distributed very densely in topology 1, whereas in topology 2, same number of nodes are distributed sparsely. Again, the maximum hop length is more in topology 2 than the topology 1. The used parameters during the testbed experiments are mentioned in Table 4.3.

4.6.2 Benchmark Schemes and Performance Metrics

In the testbed experiments, the same benchmark schemes and performance metrics are used, as mentioned in Section 4.5.2, to evaluate the proposed scheme. Unlike the plots of simulation results in which total joining time of all the pledges are plotted, the experimental results are plotted in different ways. In this section, how many pledges have become the TSCH and 6TiSCH joined nodes within a fixed time interval such as 0-3, 6-9, 12-15 minutes are shown in the plots. Note that a pledge is called a TSCH joined node when it receives its first valid EB frame. On the other hand, a TSCH joined node becomes a 6TiSCH

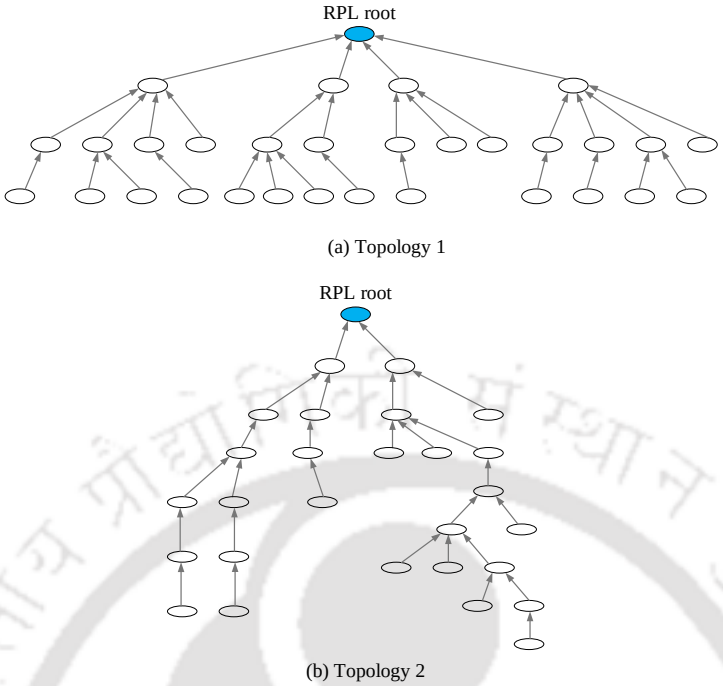


Figure 4.16: *Two different topologies considered for testbed experiments.*

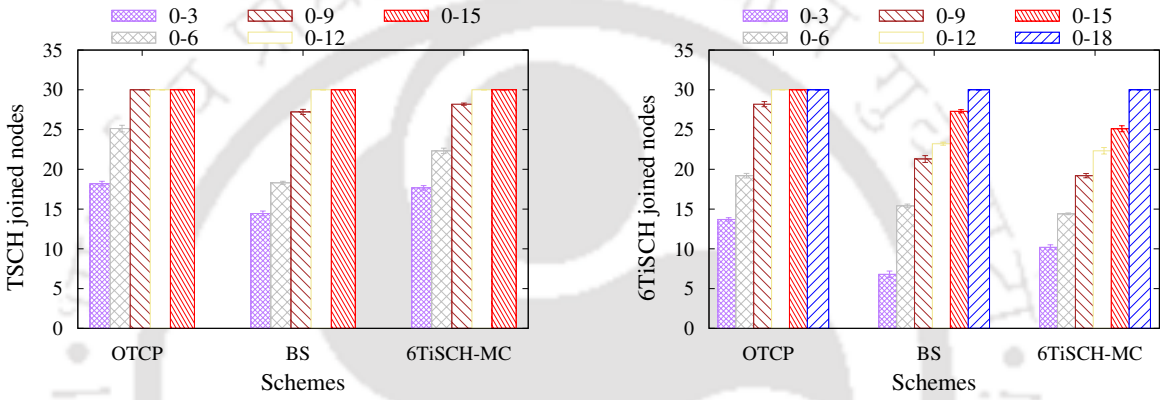
Table 4.3: *Expermental settings*

Parameter	Value
Operating System	Contiki-NG
Testbed	Grenoble, FIT IoT-LAB
Node type	M3
Network size	31
Number of channels	16
Timeslot length (L)	10 ms
Slotframe size	101 *timeslots*
EB interval	4 s
RPL version	RPL Lite
DIO interval (min-max)	$2^{12} - 2^{20}$ ms
Keep-alive interval	30 s
RPL DIO-interval	Trickle interval
RPL Redundancy constant	10
RPL Objective Function	MRHOF - ETX

joined node when it receives a valid DIO packet after its secure enrollment in the network. Similarly, the average energy consumption of the nodes are also shown in such intervals.

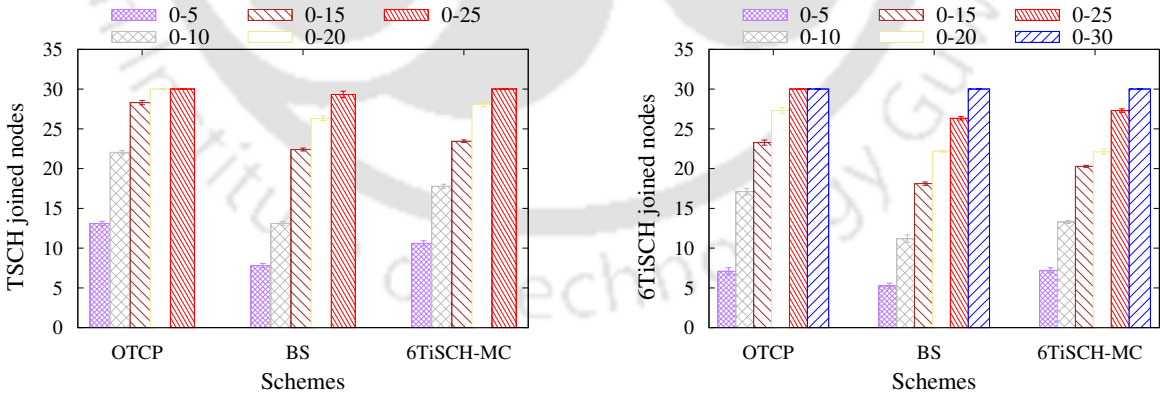
4.6.3 Testbed Results and Discussion

The received results are plotted in Figure 4.17, 4.18, and 4.19 using *slotframe* length SF=101 timeslots, DIO minimum interval=4096 *ms* with 95% confidence interval. Each bar in the Figure 4.17, 4.18 shows the number of joined nodes, and in Figure 4.19 shows average energy consumption under various time intervals (e.g., 0-3, 0-9, 6-9, 12-15 etc.). The measuring units of time intervals are *minute*. Figure 4.17a and 4.18a show the TSCH joining time and Figure 4.17b and 4.18b show the 6TiSCH joining time for both the topologies. The



(a) *TSCH join time* **(b)** *6TiSCH join time*

Figure 4.17: *Testbed results on different node joining time.*

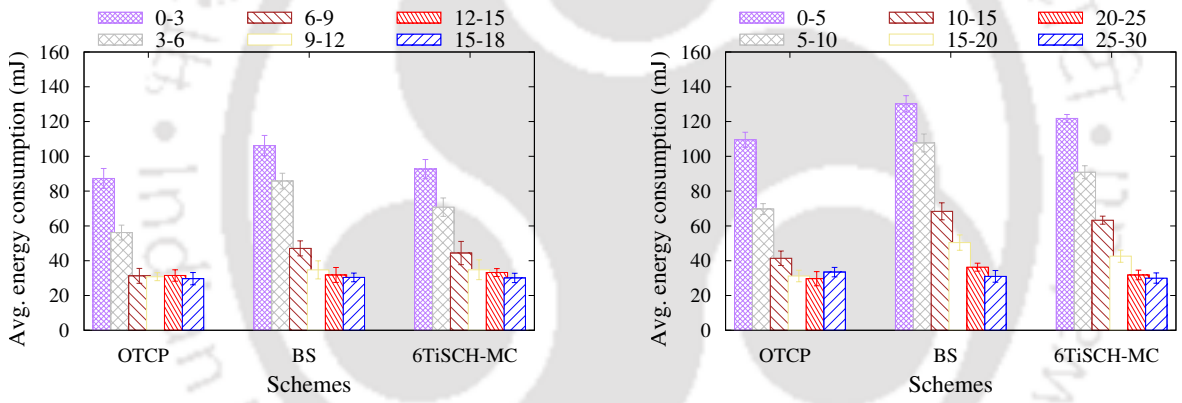


(a) *TSCH join time using Topology 2* **(b)** *6TiSCH join time using Topology 2*

Figure 4.18: *Testbed results on different node joining time.*

received results show that the proposed scheme OTCP outperforms the benchmark schemes in terms of joining time and energy consumption. The reason is due to providing sufficient routing information by alternating the priority of different control packets and resetting

the TA during the requirements of routing information packet. Furthermore, the proposed scheme also allows the urgent packet containing nodes to access the transmission channel opportunistically, reducing the channel access delay during network formation. The received results also signify that when the hop distance increases from the RPL root node, the joining times are also increased. This is because a node needs to join the network completely, i.e., a node should get both EB frame and DIO packet, before transmitting its control packets. So, the nodes located in more depth need to wait for more amount of time than the nodes which are in less depth from the RPL root node to get a valid EB frame and DIO packet. Hence, both the TSCH and 6TiSCH joining time of the pledges are more in topology 2 in all the (existing and proposed) schemes as it has nodes with more hop distance from the RPL root than the topology 1. Furthermore, the 6TiSCH-MC outperforms BS scheme



(a) *Avg. energy consumption for Topology 1* **(b)** *Avg. energy consumption for Topology 2*

Figure 4.19: *Testbed results on different node's average energy consumption.*

in topology 2 as the nodes are distributed sparsely in topology 2. Hence, nodes need to wait for more amount of time to receive their EB frames because less number of EB frames are transmitted by the less numbers of neighbors. It results in less EB reception probability, which in turn increases the overall formation time of the network using the BS scheme. As longer joining time contributes to more energy consumption; hence, our proposed scheme outperforms the benchmark scheme in terms of energy consumption as shown in Figure 4.19a and 4.19b. From the testbed experimental results, it can be concluded that the proposed scheme performs better than benchmark schemes in terms of joining time and energy consumption.

4.7 Summary

In this chapter, we proposed *opportunistic transmission of control packets* (OTCP), which is the combination of *opportunistic priority alternation and rate control* (OPR) scheme and *opportunistic channel access* (OCA) scheme for faster association of nodes in 6TiSCH networks. In 6TiSCH network, by default, the EB frame is given the highest priority over other control packets. Further, the rate of DIO transmission by joined nodes is independent of pledge's joining process. Therefore, at first, analytically we showed that the performance of 6TiSCH network formation degrades because of the EB's highest priority and insuffcient transmission of DIO packet. To overcome these two problems, we proposed the scheme OPR. In OPR, a joined node gives the highest priority to DIO packet over EB frame whenever it is required. OPR also allows a joined node to reset its DIO transmission interval in TA when it receives a JRQ frame from a pledge to transmit a DIO packet immediately. Receiving a DIO packet in less time also helps the pledges to choose their routing parents efficiently, which further reduces inconsistency in a network. Apart from these two problems, this chapter deals with the channel congestion problem of CSMA/CA. Analytically we showed that an increasing number of joined nodes delay the network formation process because of increasing congestion in shared cell. The joined nodes or pledges may not be able to transmit their urgent control packet within a minimum time when required, which, in turn, increases network formation time. Therefore, we proposed an OCA scheme for such an urgent packet containing nodes to access the shared cell immediately. Theoretical results were provided for the validation of the proposed schemes. Again, simulation and testbed results of the proposed schemes were compared with 6TiSCH-MC and BS. The obtained results showed significant improvement w.r.t network joining time of 6TiSCH networks.

In this chapter, we found that the routing information carrying DIO packet has a significant impact on the formation of 6TiSCH network. In the next chapter i.e. Chapter 5, we study the impact of DIO packet generating/transmitting Trickle Algorithm (TA) on 6TiSCH network formation in more details.

"Stay Hungry! Stay Foolish!"

~Steve Jobs (1976–2011)

5

Adaptive Control Packet Broadcasting

In the previous chapter, we find that the routing information carrying DIO packet has a significant effect on the construction of RPL-DODAG, and so on 6TiSCH network formation. Therefore, in this chapter, we explore the effect of DIO packet generating/transmitting Trickle Algorithm (TA) [76] on the formation of 6TiSCH network in more detail by designing a Markov Chain based analytical model. Subsequently, we propose two schemes to improve the formation of 6TiSCH network. The first proposed scheme changes the default behavior of TA dynamically depending on the current network condition for efficient transmission of DIO packet. The second proposed scheme restricts the nodes to transmit their control packets frequently, and thus, decreases the formation time of 6TiSCH network. We perform testbed experiments in FIT IoT-LAB to evaluate the proposed schemes together after

implementation them on Contiki-NG.

5.1 Introduction

According to 6TiSCH-MC, a pledge should get both the network information carrying EB frame and routing layer protocol i.e., RPL [43] information carrying DIO packet to fully join a 6TiSCH network. The generation and transmission of DIO packet is governed by the Trickle Algorithm (TA) [76]. TA tunes the DIO packet transmission rate depending on the network condition to save network bandwidth and nodes' energy. When a network is stable, i.e., there is no inconsistency, the TA reduces the DIO generation rate by increasing the DIO generation interval. Otherwise, when the network is not stable, the TA increases the DIO transmission rate by reducing the DIO generation interval, so that the (new) routing information reaches all neighbor nodes in less time. Furthermore, sometimes, TA suppresses the transmission of generated DIO packet when sufficient number of DIO packets containing similar information have already been transmitted by some other nodes in the network. Many parameters are used in TA to generate and transmit DIO packets. The complete behavior of TA and description of the used parameters along with different Trickle steps are described in Section 5.2 of this chapter.

Note that none of the existing works studied the impact of DIO transmission rate on 6TiSCH network formation. However, many works [72], [73], [83], [85], [87] mentioned that the default TA creates unfairness among the nodes during DIO transmission which results in longer construction of RPL's-DODAG. It happens due to the use of listen-only period for the suppression mechanism of TA. In every Trickle interval, a node randomly selects one time instant between half of the interval to the end of the interval. The node listens for consistent DIO packets from the starting of the interval to that selected random time. This listening time is called *listen-only* time. If the node receives consistent DIOs more than a threshold called *redundancy constant* k, then it suppresses its own DIO, otherwise, it transmits. Figure 5.1 shows few examples of Trickle operation considering $k = 1$, where node ②, ③ always suppress their DIOs because of the improper selection of the random

time. This results in longer DODAG construction time. This fairness issue also causes quick energy draining for the frequently transmitting node ①.

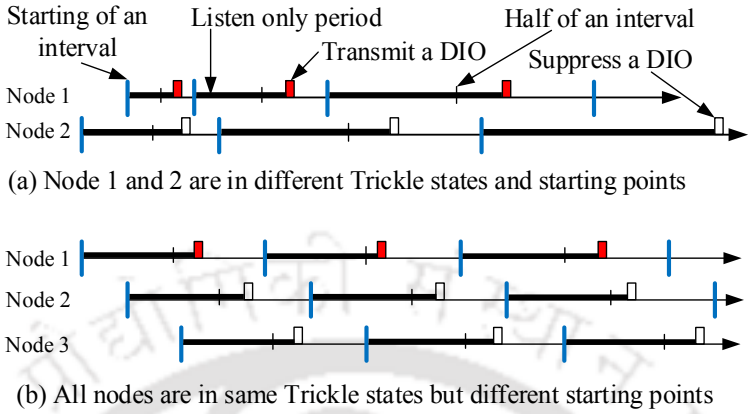


Figure 5.1: *Example of Trickle operation in different scenarios*

The works [72], [73], [83], [85], [87] attempted to solve the problem in various ways, such as by varying the listen-only period depending on previous suppression and transmission counts of DIO packets, neighbor nodes. However, their solutions consider the previous version of IEEE 802.15.4, where the transmission of DIO packets was not performed in shared cell. A node needs to wait for shared cell to transmit its DIO packet and it may need to wait for a longer time to transmit DIO packet due to the congestion in shared cell of 6TiSCH networks. Hence, the performance obtained by the above mentioned existing works may not be the same in 6TiSCH network as sufficient bandwidth was used in their solutions. Furthermore, we observe that the default TA creates burst transmission of DIO packets in some network instances, which severely congests the shared cells of 6TiSCH network. Hence, the TA should be modified so that it can provide fair DIO packet transmission opportunities among the nodes and does not create severe congestion in shared cells.

On the other hand, the work in [45] proved that joining time and energy consumption of a pledge increase due to the increasing congestion in shared cells. Although the existing work [46] studied the impact of fixed EB rate on congestion in shared cell, and thus on network formation, the impact of DIO transmission rate is still unexplored. However, our Markov Chain based analysis reveals that DIO transmission rate affects 6TiSCH network formation. Furthermore, the EB transmission rate is set by the network administrator at

the beginning and remains fixed in 6TiSCH network. This fixed rate creates congestion in shared cells when the number of nodes increases in the network (ref. Chapter 3). Hence, the EB rate needs to be modified adaptively in order to reduce congestion. Again, there is no limitation on the number of transmitted control packets by a node (within a period of time). This results in severe congestion in shared cells when all the nodes transmit their control packets more frequently. Hence, there should be some mechanism by which congestion in the shared cell can be reduced in such scenarios.

The main goal of this chapter is to reduce the joining time and energy consumption of the pledges during 6TiSCH network formation by efficiently transmitting the DIO packet and reducing congestion in shared minimal cell. The contribution of this chapter is twofold. First, a *dynamic Trickle algorithm* is proposed for efficient transmission of DIO packet, which also minimize the congestion in shared cells. It is done by skipping the intermediate Trickle states, which causes unnecessary DIO transmission during network consistency. To minimize the congestion further, the redundancy constant k is set dynamically instead of using a fixed value of it all the time. Furthermore, the proposed dynamic Trickle algorithm allows a node to adjust its Trickle listen-only period based on its previous transmission records to achieve fair DIO transmission opportunity, which helps in faster formation of RPL's DODAG, and so 6TiSCH network formation.

Secondly, we propose a *slotframe window* (SW) *based adaptive control packet transmission* scheme to restrict a node on generating and transmitting more control packets within a short period of time. This restriction reduces congestion in the shared cell and also provides fair control packet transmission opportunities among the nodes. Nodes calculate their *SWs* depending on the number of nodes and generated DIO packets in their surroundings. So, it does not require additional signaling overhead to compute the *SWs*. In brief, the contributions of this chapter are as follows:

- A Markov Chain based analytical model is proposed for analyzing the effect of Trickle parameters on 6TiSCH network formation.
- A *dynamic Trickle algorithm* is proposed to reduce congestion in shared cells and to

provide fair DIO transmission opportunities to all the nodes.

- A *slotframe window based adaptive control packet transmission* scheme is proposed to restrict the nodes from transmitting several control packets within a short period of time in order to reduce congestion in shared cell.

- The proposed schemes are implemented on Contiki-NG and evaluated using FIT IoT-LAB testbed.

The rest of the chapter is organized as follows. Section 5.2 presents an analytical model to show the impact of various Trickle parameters on 6TiSCH network formation. Both the proposed schemes are briefly described in Section 5.3. In Section 5.4, we provide the experimental evaluation of the proposed schemes together and finally, we summarized this chapter in Section 5.5.

5.2 Theoretical Analysis of Trickle Algorithm

In this section, the impact of Trickle parameters on 6TiSCH network formation is described briefly. Although the transmission rate of DIO packet is governed by TA [76] but it is worth mentioning that the DIO transmission rate depends on few Trickle parameters such as *minimum DIO generation interval*, I_{dio}^{min} and *redundancy constant*, k. These parameters are set by the network administrator at the beginning and remain unchanged. The smaller value of I_{dio}^{min} and bigger value of k allow more DIO transmission in a network, and vice versa [76]. Figure 5.2 shows the flowchart of TA and the used parameters along with different Trickle steps are described as follows:

- The Trickle algorithm starts with *minimum DIO generation interval* I_{dio}^{min}.

- The generation interval I is doubled after each interval, until the interval reaches the *maximum DIO generation interval* I_{dio}^{max} in a steady network.

- In every interval I, each node selects a random time $t \in [\frac{I}{2}, I]$.

- From the interval beginning to t, the node maintains a *counter* c to keep track of the number of consistent DIO packets transmitted by other neighbor nodes. This period is called the *listen-only* period of the node.
- During the listen-only period, if the counter c surpluses the value of predefined (and fixed for all nodes) *redundancy constant* k, then the node suppresses the generated DIO packet. Otherwise, it is transmitted. The value c is reset to 0 after each interval.
- In case of events such as inconsistency in the network, reception of multicast DIS packet, and DODAG version changed, I is reset to the minimum interval I_{dio}^{min}.

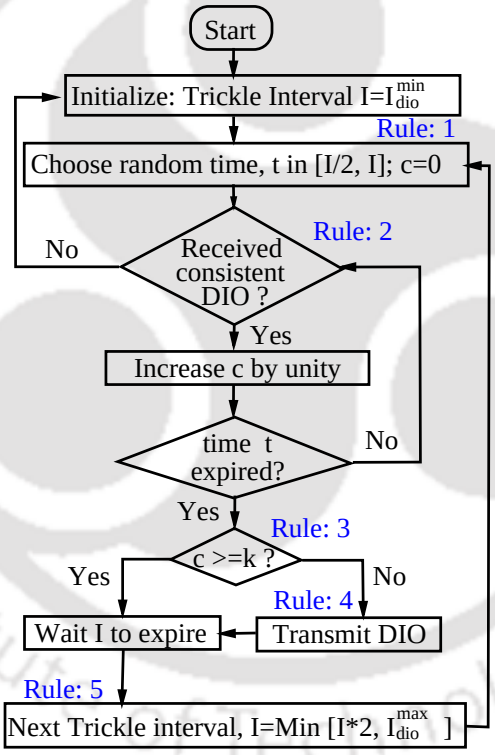
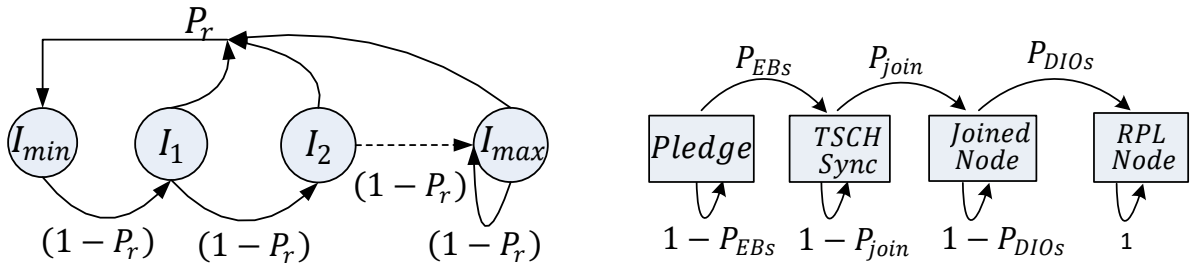


Figure 5.2: *Flowchart of Trickle Algorithm*

5.2.1 Modeling the Pledge Joining Process

Let us assume a two dimensional area of size A, in which N nodes are deployed randomly. Let the coverage radius of each node is r. All nodes run the RPL protocol and thus the Trickle algorithm in it. This setup is used to calculate the *average joining time* (AJT)



(a) *Semi-Markov model for Trickle states* **(b)** *Markov model of node joining time*

Figure 5.3: *Markov Chain models used for theoretical analysis*

of the deployed pledges. The AJT of a pledge denotes the time taken by the pledge to receive one valid EB frame followed by its secure enrollment (i.e., exchanging of JRQ and JRS frame) and reception of a valid DIO packet. Therefore, AJT mainly depends on the transmission probabilities of EB and DIO packet in a shared cell. Trickle algorithm doubles the DIO generation interval after each interval, until maximum Trickle interval I_{dio}^{max} is reached (Figure 5.2, Rule: 5). The value of the current interval I reset to I_{dio}^{min} when network inconsistency occurs (Figure 5.2, Rule: 2). Therefore, the next Trickle state depends on the current Trickle state and network condition. Hence, the average probability of being in a Trickle state can be calculated using a *Semi-Markov model* as shown in Figure 5.3a. Note that the DIOs are generated irregularly i.e., the DIO generation rate does not follow any constant average rate, and the next DIO generating state depends on how long the current state has lasted. Therefore, Semi-Markov model is used to find the total time spent by a node in each Trickle state i.e., sojourn time. Figure 5.3a shows how a node changes its Trickle state starting from the first state I_{dio}^{min} to the last state I_{dio}^{max}. Each Trickle state I_j represents $2^{j-1} \times I_{dio}^{min}$ duration of Trickle interval. There are maximum N_D Trickle states. The state transition probabilities P_r and $(1-P_r)$ are shown in the figure, where P_r is the Trickle reset probability.

Now, the probability of generating a DIO packet in a shared cell being in Trickle State I_j can be calculated as,

$$P_{dio}^{j} = \sum_{j=1}^{N_D} \binom{N_D}{j} x(1-x)^{(N_D-1-j)} min(1, \frac{L}{2^{j-2} I_{dio}^{min}}) \tag{5.1}$$

where, $x = \frac{P_r(1-P_r)^{j-1}2^{j-1}}{\sum_{j=1}^{N_D} P_r(1-P_r)^{j-1}2^{j-1}}$ implies the probability of being in the Trickle state I_j following the sojourn time of Semi-Markov process as mentioned in [45]. The term $min(1, \frac{L}{2^{j-2}I_{dio}^{min}})$ denotes the probability of generating a DIO within the interval $[2^{j-2}I_{dio}^{min}, 2^{j-1}I_{dio}^{min}]$ for the I_j Trickle state, where L is the slotframe length. This is because, a node generates a DIO at randomly selected time t in between $[\frac{I}{2}, I]$ when the interval length is I (Figure 5.2, Rule: 1). Now, the transmission of a generated DIO packet follows the suppression mechanism of Trickle algorithm. A node suppresses its generated DIO when it receives greater than or equal to k number of consistent DIO packets till the randomly selected time t of an interval (Figure 5.2, Rule: 3, c denotes the number of received consistent DIO). Otherwise, the node transmits its generated DIO packet (Figure 5.2, Rule: 4). Therefore, when all the neighbor joined nodes (say jn) of a node generate and transmit their DIOs, the node can surely transmit its own DIO if $jn < k$. It is because less than k number of DIOs would be transmitted by its surrounding nodes.

On the other hand, when $jn \geq k$, the node can be able to transmit only when it generates DIO within the first k transmissions of DIOs. Thus, DIO transmission probability in a shared cell P_{dioTx} depends on its generation probability P_{dio}^{j} and suppression conditions as follows,

$$P_{dioTx} = P_{dio}^{j} P(jn < k) + P_{dio}^{j} P(jn \geq k) P(f) \tag{5.2}$$

where, $P(jn < k)$ and $P(jn \geq k)$ denote the probabilities of satisfying the $jn < k$ and $jn \geq k$ conditions, respectively, and $P(f)$ denotes the probability that a node generates its DIO within the first k transmissions of DIOs. These probabilities can be calculated as,

$$\begin{aligned} P(jn < k) &= \frac{(k-1)}{M} \\ P(jn \geq k) &= \frac{(M-k+1)}{M} \\ P(f) &= \frac{k}{(jn+1)} \end{aligned} \tag{5.3}$$

where, $M = \left(\frac{N}{A}\pi r^2\right)$ is the total number of neighbors of a node. Both the pledges and joined nodes are included in those M neighbors. We assume that initially, all M nodes are pledges. They join the network one by one. So, at any instant, the probability of having any number

of joined nodes equals to $\frac{1}{M}$ considering uniform probability distribution. Therefore, $\frac{(k-1)}{M}$ denotes the probability that at any instant of time at most $(k-1)$ joined nodes are available. Similarly, we compute the other probabilities.

We use the Markov Chain model (shown in Figure 5.3b) describe in our Chapter 3 (Section 3.2) to compute the AJT of the pledges as follows,

$$AJT = PJT + ASF \times L \tag{5.4}$$

where PJT is the *parent's average joining time* of the pledge in multi-hop network, and ASF is the *average number of slotframes* that is required for a pledge to join the network, which is calculated as follows,

$$ASF = \frac{1}{P_{EB_S}} + \left(\frac{1}{P_{JRQ_S}} + \frac{1}{P_{JRS_S}}\right) + \frac{1}{P_{DIO_S}} \tag{5.5}$$

where, P_{EB_S}, P_{DIO_S}, P_{JRQ_S}, and P_{JRS_S} are the probabilities of successfully transmitting an EB, DIO, JRQ, and JRS frames/packets in a shared cell, respectively. The values of these probabilities can be calculated using the value of P_{dioTx} (from Equation 5.2) and following the method as described in Chapter 3 (Section 3.2).

5.2.2 Analytical Results and Discussion

For performing theoretical analysis, we consider that N nodes are deployed at an area of $A = 100 \times 10\ meters^2$, where radio range of each node is $r = 10\ meters$. The default values of other variables are as follows: number of channels $N_C = 16$, packet loss probability $P_{loss} = 0.2$, beacon interval $BI = 4sec$, timeslot duration $= 10ms$, $L = 101 timeslots$, $N_D = 8$, $I_{dio}^{max} = 2^{N_D} I_{dio}^{min}$ and $P_r = 0.2$. Figure 5.4 and 5.5 shows the *average joining time* (AJT) against the varied I_{dio}^{min}, k, BI and N.

Figure 5.4a shows the joining time for different values of I_{dio}^{min} with $N = 100$ and $k = 4$. When the values of I_{dio}^{min} is less, frequent transmission of DIO packet congests the shared cell, which results in higher joining time of the pledges. On the other hand, using higher value of I_{dio}^{min}, pledges need to wait longer time for receiving DIO packet. Hence, their joining times

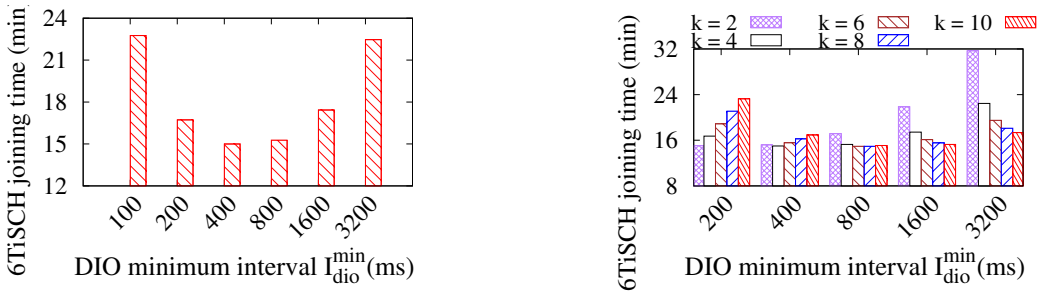


Figure 5.4: *Analytical results on joining time under varied (a) I_{dio}^{min}, (b) I_{dio}^{min}, k*

are high again. It indicates that neither higher nor lower values of I_{dio}^{min} are optimal with respect to the performance of network joining time. In Figure 5.4b, along with I_{dio}^{min}, k is also varied. Results show that some correlation exists between these two parameters I_{dio}^{min} and k. When the values of I_{dio}^{min} are either very low or very high, the values of k need to be very low and very high, respectively, to achieve less joining time.

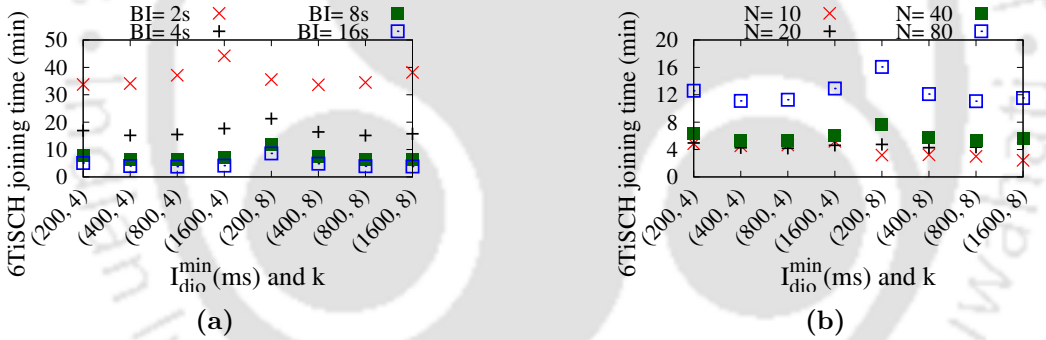


Figure 5.5: *Analytical results on joining time under varied (a) I_{dio}^{min}, k, BI and (b) I_{dio}^{min}, k, N*

In Figure 5.5a, the effect of EB generation rate combining with I_{dio}^{min} and k are shown. Frequent transmission of EB frame severely congests the shared cell. Again, because of the highest priority of EB frame, other control packets are replaced by the EB from the transmission buffer. Hence, the joining time of the pledges increase. So, longer EB generation interval, helps in reducing congestion in shared cell, and thus, yields lesser joining time irrespective of the values of I_{dio}^{min} and k as shown in the figure. It is further observed that congestion in the shared cell increases with the increasing node densities, and so the joining time. However, the value of increment varies with the values of I_{dio}^{min} and k as shown in Figure 5.5b.

Analytical results show that the same values of I_{dio}^{min} and k show different results under different network conditions such as network size and beacon generation interval. It is mainly due to the congestion in shared cells. So, it can be concluded that the TA should dynamically change its values depending on the network condition, rather than being fixed, to reduce congestion in shared cells.

5.3 Adaptive Control Packet Broadcasting (ACB)

In this section, the proposed *dynamic Trickle algorithm* and the *slotframe window* (*SW*) *based adaptive control packet transmission* schemes are described briefly.

5.3.1 Dynamic Trickle Algorithm

The above mentioned analytical results show that improper selection of Trickle parameters such as I_{dio}^{min} and k results in more DIO transmission in the network, which increases the congestion in shared cell, and so 6TiSCH formation time. When the TA is reset, it increases the transmission of DIO packets. In Figure 5.2, Rule: 2 shows one of such Trickle resetting conditions that is when a joined node receives an inconsistent DIO packet. Note that there are few more Trickle resetting conditions such as reception of multicast DIS, resetting of DODAG, changing the DODAG version etc. A single multicast DIS request forces all the recipient nodes to reset their Trickle Algorithms. This results in burst transmission of DIO packets in the network for a long time, which creates severe congestion in shared cells. So, in order to reduce such burst transmission, the proposed scheme allows the receiving node to generate only a single DIO packet in the consistent networks. This is done by allowing the receiving node to directly jump to the Trickle state where it has received the multicast DIS after generating one DIO packet. Hence, it reduces the number of transmitted DIOs by skipping the intermediate Trickle state(s). Thus, this modification reduces congestion in the shared cells and energy consumption of the transmitting nodes. This modification is done at Figure 5.2, Rule: 2 and the steps are mentioned in Algorithm 4.

Algorithm 4 Skipping Intermediate Trickle State

```
1: if Received consistent multicast DIS then
2:     if I is not I_dio^min then
3:         flag = j /*save the Trickle state*/
4:         Reset I, j, S and T_dio to their default values
5:     end if
6: else if flag then
7:     I = 2^(flag-1) I_dio^min /*jump to saved state*/
8:     Reset flag
9: end if
```

To reduce the transmission of unnecessary DIO packet further, the *redundancy constant* k is set based on the number of neighbor and current Trickle state instead of using a fixed value all the time. The propose scheme allows all the nodes to transmit their DIOs at the initial Trickle state as a node would be in that state only when there is a strong requirement of DIO in the network. So, in the initial Trickle state, a node sets k equal to one more than its neighbor node count. However, after this initial state to the half of the total Trickle states i.e., $\frac{N_D}{2}$, k is set to half of the neighbors so that frequent transmission of DIO can be reduced. Furthermore, when the nodes generate their DIOs less frequently i.e., their current Trickle state is greater than $\frac{N_D}{2}$, again k is set to one more than neighbor node count. However, the maximum value of k is set to 10 (specified by RPL standard). The k is calculated before the condition at Figure 5.2, Rule: 3 and Algorithm 5 describes the steps.

Algorithm 5 Dynamic Redundancy Constant k

```
1: if I=I_dio^min then
2:     k = min(N_nbr + 1, 10) /*all nodes are allowed*/
3: else if j > I_dio^min && j ≤ N_D/2 then
4:     k = min(⌈(N_nbr+1)/2⌉, 10)
5: else
6:     k = min(N_nbr + 1, 10)
7: end if
```

Now, to deal with the fairness issue of Trickle algorithm, the proposed dynamic Trickle algorithm allows a node to calculate its DIO transmitting shared cell based on its previous transmission record and number of neighbors. When a node suppresses its DIOs continuously in the previous intervals, it gets more opportunity to transmit in the next interval.

This is done by selecting the random time t within a short interval (started from the beginning of the interval) depending on the number of neighbors N_{nbr} and total suppression count S i.e., $t = \left[0, \left\lceil \frac{N_{s\_cell}}{2} \right\rceil - (\frac{N_{s\_cell}/2}{N_{nbr}+1} \times S)\right]$. Here, N_{s_cell} denotes the number of shared cells present in a Trickle state. When the listen-only period of a node is small, less number of consistent DIOs will be received by the node, and so it can transmit its generated DIO immediately and at the beginning of the interval. Furthermore, the node does not need to wait more than half of the interval like the default Trickle algorithm. On the other hand, a frequent DIO transmitting node calculates its listen-only period depending on the number of already transmitted DIOs T_{dio} i.e., $t = \left[\left\lceil \frac{N_{s\_cell}}{2} \right\rceil + (\frac{N_{nbr}+1}{N_{s\_cell}/2} \times T_{dio}), N_{s\_cell}\right]$. This allows the node to choose its transmitting cell from the end of the interval. Thus the proposed scheme maintains fairness among the nodes. This modification is done at Figure 5.2, Rule: 1 and steps are mentioned in Algorithm 6.

Algorithm 6 Providing Fair DIO Transmission Opportunity

1: Calculate: N_{s_cell}= $\left\lfloor \frac{2^{j-1} I_{dio}^{min}}{L \times T} \right\rfloor$ /*count the number of shared cells present in a Trickle state j*/
2: **if** I = I_{min} OR S != 0 **then**
3: $t = \left[0, \left\lceil \frac{N_{s\_cell}}{2} \right\rceil - (\frac{N_{s\_cell}/2}{N_{nbr}+1} \times S)\right]$
4: **else if** T_{dio} != 0 **then**
5: $t = \left[\left\lceil \frac{N_{s\_cell}}{2} \right\rceil + (\frac{N_{nbr}+1}{N_{s\_cell}/2} \times Tr), N_{s\_cell}\right]$
6: **end if**

5.3.2 Slotframe Window based Adaptive Transmission

The dynamic Trickle algorithm only deals with transmission of DIO packet in the 6TiSCH network. However, the transmission rate of EB frame also has a significant affect on 6TiSCH network formation. Therefore, apart from the dynamic Trickle algorithm, we propose the *slotframe window* (*SW*) *based adaptive control packet transmission* scheme to reduce congestion in shared cell. Basically, this scheme restricts the joined nodes from frequent control packets transmission when the number of neighbor nodes and generated control packets are more in the network. To understand the proposed scheme, let us assume I_{eb} is the fixed EB

generation rate for all the nodes. So, the probability of generating an EB per shared cell by a node is $P_{eb} = \frac{(L \times T)}{I_{eb}}$. Here, L is the slotframe length in timeslots and T is the timeslot duration. Now, if N_{nbr} denotes the number of neighbors, then $P_{eb} \times (N_{nbr} + 1)$ should be less than 1 i.e., $P_{eb} \times (N_{nbr} + 1) < 1$; otherwise, when $P_{eb} \times (N_{nbr} + 1) > 1$ then at least one node will not be able to transmit its EB in every slotframe by the *Pigeon Hole theory*. Therefore, the proposed scheme allows a node to generate its EB depending on its number of neighbors rather than using a fixed rate as follows,

$$I_{eb} \geq Min \left\{ I_{eb}^{max}, Max \left\{ I_{eb}^{min}, L \times T \times (N_{nbr} + 1) \right\} \right\} \tag{5.6}$$

where, I_{eb}^{min} and I_{eb}^{max} are the minimum and maximum beacon generation interval. Here, a node waits at least $(N_{nbr} + 1)$ shared cells before transmitting its next EB frame. However, the same node also has other *sporadic* control packets such as DIO, DIS, and keep-alive to transmit. For this, we introduce the term *slotframe window SW*, which is calculate as follows,

$$SW = L \times T \times (N_{nbr} + 1) + \left\lceil x \ \% \ of L \times T(N_{nbr} + 1) \right\rceil \tag{5.7}$$

where the term $x \ \% \ of L \times T(N_{nbr} + 1)$ defines the number of extra shared cells allocated for other control packets, which is mainly controlled by the variable x. We consider two different values of x. By default the value of x is set to 50 but when a multicast DIS packet is multicasted, all the receiving nodes will generate one new DIO. Therefore, in such cases, the value of x is set to 100, so that additional $(N_{nbr} + 1)$ number of shared cells can be used for the transmission of newly generated DIOs.

After calculating the value of *SW*, a node is restricted to broadcast more than two control packets (e.g., one EB frame and one DIO packet) within the *SW* amount of time. Therefore, when there is a situation like more number of control packets are generated in the network, a node is restricted to transmit its control packets more frequently. This reduces congestion in the shared cell and also provides equal opportunities to all the nodes to transmit their control packets.

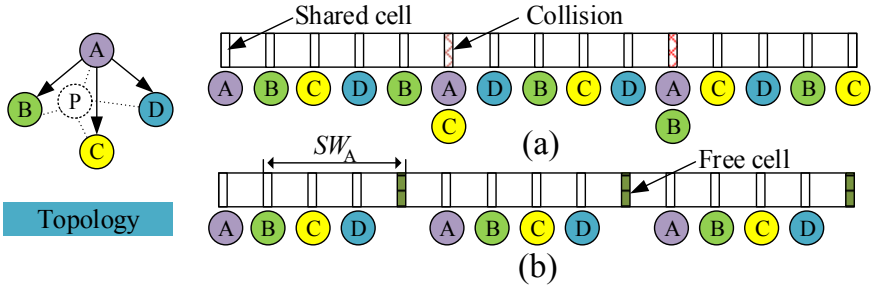


Figure 5.6: *Transmission of control packet by the nodes following (a) 6TiSCH-MC, and (b) Proposed SW-based scheme*

However, we observe that the nodes with more neighbors get less opportunity to transmit their packets than the nodes with fewer neighbors. The Figure 5.6(a) depicts one such type of scenario. When the nodes calculate their *SW* following the Equation (5.7) and $x = 0$, Node (A) gets less opportunity to transmit its packets compared to the nodes (B), (C), and (D). Furthermore, collision is possible, and the common neighbor node (P) does not get any free shared cell to transmit its control packets. To deal with this problem, the final value of *SW* is calculated as follows,

$$SW_f = Max \left\{ SW,\ SW_i^{max} \right\} \tag{5.8}$$

where *SW* is calculated by a node itself (Equation (5.7)) and SW_i^{max} is the maximum value of *SW* among its neighboring nodes. The improvement of this step is shown in Figure 5.6(b) where all nodes (including (P)) can transmit their packets without any collision. In the figure, SW_A denotes the calculated value of *SW* by node (A). The overall steps of this *SW*-based scheme are mentioned in Algorithm 7.

5.4 Testbed Evaluation of ACB

5.4.1 Experimental Setup

The proposed schemes are implemented on Contiki-NG and evaluated using testbed experiments in FIT IoT-LAB. Performance of both the proposed schemes together is compared with two benchmark schemes i.e., 6TiSCH-MC [40] and BS [46]. Every node running the

Algorithm 7 Slotframe Window based Adaptive Transmission

```
Initialize: SW_f = I_eb^min, Transmitted packet tp=0, x=50
if Received consistent multicast DIS then
    if SW_f time is over then
        Calculate: I_eb using Equation (5.6); tp = 0; x = 50;
        Calculate: SW using Equation (5.7) & (5.8)
    end if
end if
if the current Link_type is Shared then
    if tp < 2 && channel is free then
        Transmit a packet from broadcasting buffer; tp++;
    else if received multicast DIS then
        x = 100 /*to increase SW length*/
    end if
end if
```

proposed schemes includes its calculated *SW* in the Information Element (IE) of its EB frame. Thus, all the nodes exchange their calculated *SW* without any signaling overhead. The experimental testbed settings are mentioned in Table 5.1, and two different topologies are used for evaluation in which the first one is a Tree topology (Figure 5.7) and the other one is a 6×6 Grid topology. Both the topologies have different node densities and hop distances.

Table 5.1: *Experimental settings*

Parameter	Value
Operating System	Contiki-NG
Testbed	FIT IoT-LAB, Grenoble
Testbed `mote` type	M3
Number of channels	16
Timeslot, Slotframe length	10*ms*, 101 *timeslots*
RPL DIO interval	Trickle Algorithm
Experiment duration	60 min

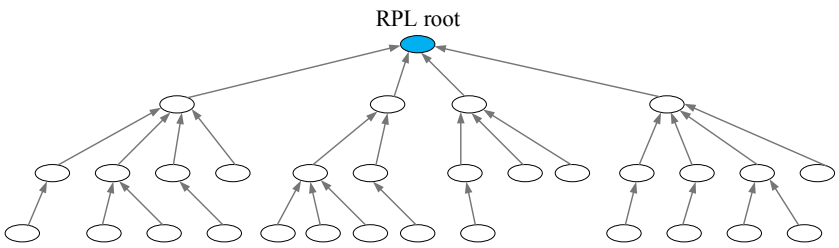


Figure 5.7: *The tree topology used in testbed experiments*

5.4.2 Performance Metrics

For evaluation, mainly four metrics are considered. The first metric is *TSCH joining time*. It denotes the time when a pledge receives its first valid EB from an already joined node. Before TSCH joining time, the pledges keep on scanning for EB frame by activating their radios all the time, which causes maximum energy consumption. However, after receiving an EB, the pledges would keep their radios active only in the shared cell, which saves energy. The second metric *6TiSCH joining time* denotes the time when a TSCH joined node receives its first valid DIO packet, i.e., TSCH joined node joins the topology created by routing protocol RPL. Now, the 6TiSCH joined node can transmit its control packets for further expansion of the 6TiSCH networks.

The `motes` used in IoT network constrained in terms of processing capacity, memory, and energy. As mentioned above, pledges consume huge energy before their TSCH joining. Therefore, *energy consumption* by pledge during its network joining is considered as our third metric for evaluation.

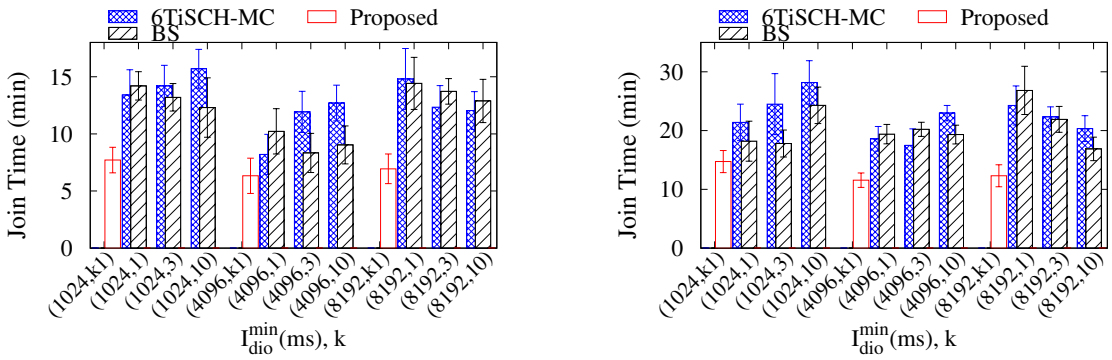
Apart from this, to prove that the proposed dynamic Trickle Algorithm gives equal opportunity to all the nodes to transmit their DIOs (i.e., provides fairness), *Jain's fairness index* [88] is used, which is defined as follows,

$$F(x) = \frac{\left(\sum_{i=1}^{z} x_i\right)^2}{z \sum_{i=1}^{z} {x_i}^2}, \quad x_i \geq 0 \tag{5.9}$$

where, z denotes the number of users (here, `motes`) contending for the resource (here, shared cell) and i^{th} user receive an allocation x_i. The value of $F(x)$ becomes 1 when the resource is equally distributed among the z users.

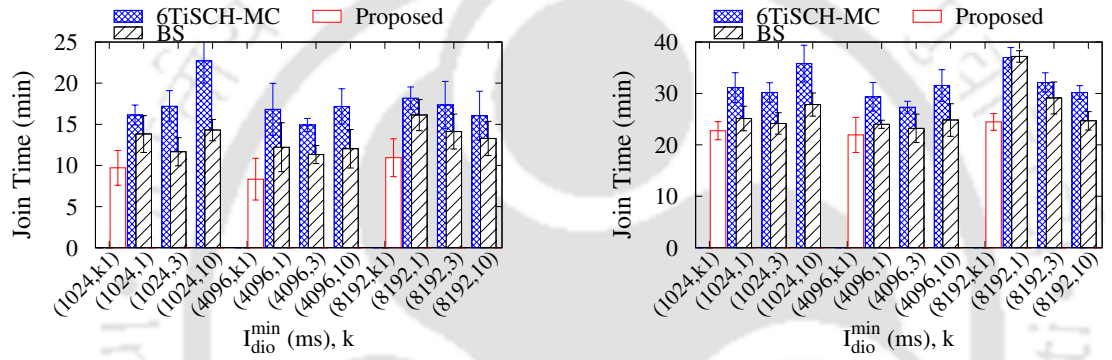
5.4.3 Results and Discussion

The received results from the testbed experiments are shown using 95% confidence interval in Figure 5.8, 5.9, and 5.10. In Figure 5.8a, 5.9a, we show the TSCH joining time and Figure 5.8b, 5.9b show the 6TiSCH joining time of the nodes in Tree and Grid topologies, respectively. The results are shown by varying the minimum DIO interval I_{dio}^{min} and redun-



(a) *TSCH join time* **(b)** *6TiSCH join time*

Figure 5.8: *Testbed experimental results of joining times using tree topology with respect to I^{min}_{dio}, and k or $k1$. $k1$ denotes redundancy constant is set dynamically*



(a) *TSCH join time in Grid* **(b)** *6TiSCH join time in Grid*

Figure 5.9: *Testbed experimental results of joining times using grid topology with respect to I^{min}_{dio}, and k or $k1$. $k1$ denotes redundancy constant is set dynamically*

dancy constant k in the 6TiSCH-MC and BS schemes. As the proposed scheme dynamically sets the value of k, so results are obtained only varying the value of I^{min}_{dio}. From the received result, it can be seen that the proposed scheme outperforms both 6TiSCH-MC and BS under all the configured settings. It is mainly because of the congestion reduction in shared cells and providing equal transmission opportunity to all the nodes by both the proposed schemes. The effect of the dynamic Trickle algorithm on the average number of transmitted and suppressed DIO packets is shown in Table 5.2 for both the topologies. By skipping the intermediate Trickle states and dynamically setting k, the number of transmitted DIOs has been reduced. Furthermore, *SW*-based scheme also helps in reducing this count. Similarly, Algorithm 6 helps in providing fairness in DIO transmission among the nodes i.e., reduces suppression count. On the other hand, 6TiSCH-MC suffers from heavy congestion in shared

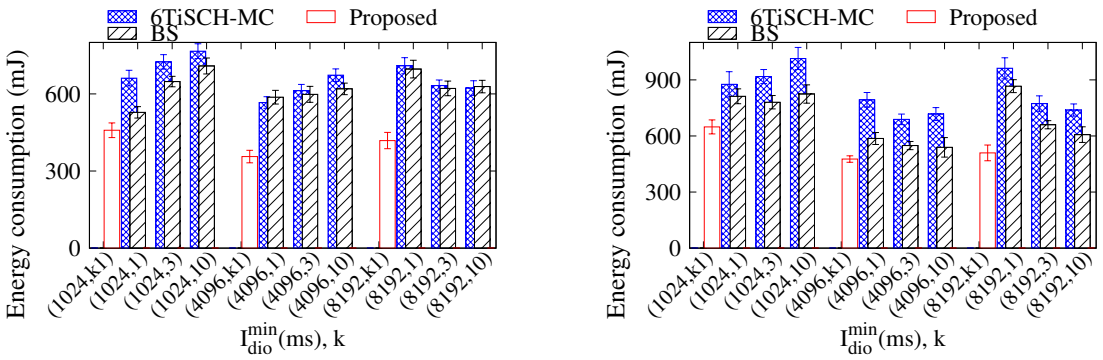
Table 5.2: *Results on average number of transmitted and suppressed DIOs shown as tuple (transmitted DIOs, suppressed DIOs) for both the topologies*

Topo -logy	I_{dio}^{min} (ms)	6TiSCH-MC			BS			Proposed
		$k=1$	$k=3$	$k=10$	$k=1$	$k=3$	$k=10$	Dynamic
Tree	1024	521, 82	568, 15	467, 0	469, 94	483, 18	599, 0	169, 1
	4096	397, 67	403, 9	490, 0	488, 123	483, 15	468, 0	152, 4
	8192	335, 57	379, 4	443, 0	568, 79	534, 2	423, 0	158, 3
Grid	1024	658, 102	692, 22	513, 0	602, 137	587, 94	667, 0	207, 3
	4096	479, 76	468, 17	459, 0	529, 54	561, 14	580, 0	187, 2
	8132	435, 67	462, 12	507, 0	524, 45	471, 9	453, 0	193, 3

cells in Grid topology as the nodes are densely deployed and so experience more joining time and energy consumption. Although the suppression mechanism is used to save node's energy and network bandwidth, the suppression of more number of DIOs affects on routing operation as shown by [83]. Although, the work in [83] does not consider the shared cell case, the suppression mechanism of Trickle algorithm is not affected whether transmission of DIO packet happens in shared cell or not. Hence, from the results obtained by the work in [83], we can infer that the proposed scheme does not affect routing operation as it suppresses less number of DIO transmission.

Nodes can save their energy by quickly joining the TSCH network and transmitting less number of control packets. As both the proposed schemes together reduce the joining time and the number of transmitted control packets, the energy consumption by the nodes also gets reduced using the proposed schemes together compared to the benchmark schemes. The energy consumption results are shown in Figure 5.10a and 5.10b for both the experimental topologies. From the results, it can be claimed that the proposed schemes together improve the network's lifetime.

Now, coming to the Jain's fairness index, when all the nodes transmit their DIOs without any suppression, $F(x)$ becomes 1. However, the transmission of more number of DIOs increases the load in the shared cell and so the congestion. Therefore, the results on fairness are shown along with *transmission load* in shared cell in Table 5.3. The transmission load in shared cells is calculated by taking the ratio of the number of transmitted DIOs to the number of occurring shared cells during the entire experimental time. The results clearly show that the proposed schemes improve the performance compared to the benchmark



(a) *Energy consumption in Tree* **(b)** *Energy consumption in Grid*

Figure 5.10: *Testbed experimental results of energy consumption with respect to I^{min}_{dio}, and k or k1. k1 denotes redundancy constant is set dynamically*

Table 5.3: *Results on fairness and transmission load shown as tuple (fairness, transmission load) for both the topologies*

Topo -logy	I^{min}_{dio} (ms)	6TiSCH-MC			BS			Proposed
		$k=1$	$k=3$	$k=10$	$k=1$	$k=3$	$k=10$	Dynamic
Tree	1024	.87, .14	.98, .15	1, .13	.83, .13	.96, .14	1, .17	.99, .05
	4096	.86, .11	.98, .11	1, .13	.80, .13	.97, .13	1, .13	.97, .04
	8192	.86, .09	.99, .1	1, .13	.88, .15	.98, .15	1, .12	.98, .04
Grid	1024	.87, .18	.97, .19	1, .14	.81, .17	.86, .16	1, .19	.99, .06
	4096	.86, .13	.96, .13	1, .13	.90, .15	.98, .16	1, .16	.99, .05
	8192	.87, .12	.97, .13	1, .14	.92, .15	.98, .13	1, .13	.98, .05

schemes. Though in some cases 6TiSCH-MC and BS give better fairness (specifically when, $k = 10$), their transmission load is also more. Nevertheless, the proposed schemes are able to maintain a balance between fairness and transmission load.

5.5 Summary

The existing works did not consider the effect of DIO transmission rate on 6TiSCH network formation. Initially, our Markov Chain based probabilistic analysis revealed that improper selection of various Trickle parameter values can increase the 6TiSCH network formation time due to the increased congestion in shared cells. Therefore, in this chapter, a *dynamic Trickle algorithm* was proposed to efficiently transmit the DIO packets in 6TiSCH network. The proposed scheme skips some intermediate DIO generation Trickle states during network consistency and varies one Trickle parameter i.e., *redundancy constant* dynamically to reduce

congestion in shared cells. Furthermore, it provides fair DIO transmission opportunities to all the nodes by varying the Trickle listen-only period of the nodes depending on their previous DIO transmission history. Apart from this, *slotframe window* (*SW*) *based adaptive control packet transmission* scheme was proposed to further reduce the congestion in shared cell. This scheme restricts the nodes from transmitting their control packets more frequently and also provides fair control packet transmission opportunities among the nodes. Both the proposed schemes were implemented on Contiki-NG and evaluated using FIT IoT-LAB testbed. Testbed results showed that both the proposed schemes together improve the joining time and energy consumption of the nodes significantly compared to the state-of-the-art schemes while providing fair DIO transmission opportunity to all the nodes.

In the previous chapters, we found that both the transmission rates of EB and DIO packets affect the performance of 6TiSCH network formation because of the congestion in shared cells. Therefore, in the next Chapter 6, we model a non-cooperative gaming approach to find out the optimal control packet transmission probability in shared cells to alleviate congestion in those cells of 6TiSCH network.

"Live as if you were to die tomorrow. Learn as if you were to live forever."

~Mahatma Gandhi (1869 - 1948)

6

Non-cooperative Gaming based Control Packet Transmission

In the previous chapters, we have shown that both the transmission rates of EB frame and DIO packet have significant impact on 6TiSCH network formation. The existing works in the literature did not study the effect of transmission rates of all control packets together during 6TiSCH network formation. Therefore, in this chapter, we inspect the optimal rate of control packet transmission to alleviate congestion in shared cell, and so, to improve the performance of 6TiSCH network during its formation. At first, we model a non-cooperative gaming approach to find the optimal control packet transmission probabilities in shared cell (or minimal cell) of the nodes. Later, the obtained optimal control packet transmission

probability is used in the proposed congestion control scheme – *game theory based congestion control* (GTCC), which restricts the nodes from frequent transmission of control packet whenever it is not required. This restriction results in minimum congestion in shared cell, and so, improves the joining time of the pledges and network lifetime. We implement the proposed GTCC on Contiki-NG and perform testbed experiments in FIT IoT-LAB to evaluate it.

6.1 Introduction

In 6TiSCH network, nodes selfishly transmit their control packets without considering the congestion in shared cell. This selfish behavior increases the joining time of the pledges by increasing the congestion in shared cells and thus reduces the lifetime and throughput of the network and increases end-to-end packet delivery latency. In fact, a joined node is not aware of the transmission rates of its neighboring nodes. The work [45] mentioned that 6TiSCH-MC does not provide enough resources i.e., number of shared cells per slotframe to transmit all the generated control packets during network bootstrapping. The shared/minimal cell severely gets congested when the number of nodes increases, which ultimately degrades the performance of 6TiSCH network during its formation in terms of pledges' joining time and their energy consumption [45]. To deal with this problem of 6TiSCH-MC, as mentioned before, Vallati *et.al.,* [45] increased the number of shared cells per slotframe. However, their proposed scheme consumes more energy and also hinders the throughput and end-to-end latency. The work in [46] suggested to use beacon transmission probability 0.1 per slotframe irrespective of the number of nodes present in a network to reduce congestion in shared cells. Similarly, we vary the beacon generation interval of the joined nodes depending on the congestion in shared cells in Chapter 3 of this thesis. On the other hand, in Chapter 4 and Chapter 5 of this thesis, we consider only efficient and sufficient transmission of DIO packet during network bootstrapping. In brief, none of the existing works (including the works done in the previous chapters of this thesis) considered the impact of the transmission rates of EB and DIO packet together on the congestion in shared cell, and so the 6TiSCH

network formation. With the increasing number of nodes, on an average transmission of EB and DIO also increases. This creates congestion in shared cell, and so increases the 6TiSCH network formation time. However, both of these control packets are required during network formation.

Few research works [89–91] used game theory-based solutions to solve the congestion problem in different networks such as 6LoWPAN and VANET (Vehicular Ad-Hoc Networks). The works [89] and [90] tried to obtain the optimal data packet transmission rates of different priority based applications in 6LoWPAN network. The authors considered the remaining buffer capacities of the intermediate forwarding nodes, congestion in transmission channel, and priority of an application as *price functions*, which were attempted to keep minimum. On the other hand, data packet transmission rate is considered as *utility function*, which was attempted to keep maximum. However, both the works have signaling overhead as DIO packet is used for sharing the information related to buffer occupancy. The work [91] tried to find out optimal beacon transmission rate for VANET, where congestion in transmission channel is considered as *price function* and beacon transmission rate as *utility function* as a higher beacon rate is desirable because it creates higher awareness within the VANET. However, in VANET, beacon is transmitted in a dedicated physical channel. On the other hand, in 6TiSCH network, including EB, other control packets are transmitted in shared cell. So, these existing game theory-based solutions will not be suitable to reduce congestion in shared cells.

However, the existing game-theory based solutions (i.e., [89–91]) look promissing in terms of controlling the congestion in different wireless networks. Therefore, in this chapter, we formulate a *non-cooperative game* for efficient control packet transmission in shared cell based 6TiSCH network. The proposed game considers all the joined nodes as *players*. The optimal solution of the non-cooperative gaming is to transmit control packets with maximum probability without congesting the shared cell. This improves the pledges' joining time and their energy consumption. Furthermore, the optimal solution is obtained without any signaling overhead in the network. The obtained optimal solution is used in the proposed

game theory based congestion control (GTCC) scheme, which reduces the congestion in shared cell while allowing the players to maximize their pay-offs irrespective of the strategies of the neighboring players. In brief, the contributions of this chapter are,

- Design a non-cooperative game to find out optimal control packet transmission probability to reduce congestion in shared cell, which improves the joining time and energy consumption of the pledges.
- It is proved that the proposed game has a unique Nash equilibrium point.
- A *game theory based congestion control* (GTCC) scheme is proposed using the obtained optimal solution by which nodes can efficiently transmit their control packets.
- At first, GTCC is validated using an existing analytical model, then it is implemented on Contiki-NG and evaluated using the FIT IoT-LAB testbed.

The rest of the chapter is organized as follows. In the next Section 6.2, we describe the modeling of the proposed game. We provide the detail description of the proposed game theory based congestion control scheme in Section 6.3. In Section 6.4, we discuss the selection of different parameter values used in the proposed game and analytical validation of the proposed scheme. Finally, in Section 6.5, testbed evaluation of the proposed scheme is provided, and in Section 6.6, we summarize this chapter.

6.2 Non-cooperative Game Formulation

Let us consider a 6TiSCH network with one JRC and a set of joined nodes defined by the set $JN = \{J_1, J_2, \ldots, J_n\}$, where J_i denotes the i^{th} joined node (i.e., $\forall J_i \in JN$, $i = \{1, 2, \ldots, n\}$). The joined nodes transmit their control packets selfishly, i.e., they do not bother about other nodes' control packet transmission rate. Hence, high congestion in shared cell becomes an obvious outcome when more nodes join the network. The joined nodes do not have any information about the transmission rates of their neighboring nodes. Hence, a non-cooperative game theoretic approach is proposed to find the optimal control

packet transmission probabilities in a slotframe of the joined nodes to reduce congestion in shared cells.

Let $G = \{JN, (\rho_i)_{J_i \in JN}, (\psi_i)_{J_i \in JN}\}$ is the proposed non-cooperative game to find out the optimal control packet transmission probabilities in a slotframe of the nodes, in which all the three tuples are as follows:

Players: J_i; J_i is the member of set JN i.e., $\forall J_i \in JN$, who wants to transmit its control packets in shared cell. Note that the JRC is also included in the set JN.

Strategies: S_i; S_i denotes the strategy for player J_i, $\forall J_i \in JN$. Each strategy defines the transmission probability of control packet in shared cell. The transmission rates of EB frame and DIO packet are different. Therefore, it is required to normalize them as follows. If the slotframe length is L *timeslots* with one *timeslot* duration is T and the minimum EB generation interval is I_{eb}^{min}, then the EB transmission probability per shared cell is $p_{eb} = \frac{(L \times T)}{I_{eb}^{min}}$. On the other hand, if the average DIO transmission probability per shared cell is p_{dio} (p_{dio} can be calculated following the method described in [45]), then the probability of transmitting both these two control packets in a shared cell by a joined node is $\rho = p_{eb} + p_{dio}$. However, a node can transmit either an EB frame or a DIO packet in a shared cell/slotframe. So, the minimum and maximum value of ρ_i can be in between 0 and 1, respectively. Thus, strategy space for player J_i is $S_i = [0, 1]$ and strategy space for all the players is $S = \prod_{i=1}^{n} \rho_i$, $\forall J_i \in JN$.

Pay-off function: $\psi_i; \rho_i \leftarrow \mathbb{R}$ denotes the pay-off function of player J_i, $\forall J_i \in JN$. Player J_i tries to optimize its profit by maximizing its pay-off function ψ_i considering the best value of its ρ_i over $[0, 1]$.

Frequent transmission of control packets helps the pledges to join the network quickly when the congestion in the shared cell is minimum. However, it consumes more energy of the transmitting node and also increases congestion in shared cell. Therefore, the proposed pay-off function is designed to keep balance among the control packet transmission probability, channel congestion, and energy consumption of the transmitting nodes. Thus, the proposed

pay-off function for a player J_i is described as follows:

$$\psi_i(\rho_i, \rho_{-i}) = U_i(\rho_i) - \frac{1}{1 - MBR_i(\rho_i, \rho_{-i})} - E_{TX_i}(\rho_i) \tag{6.1}$$

where $\psi_i(\rho_i, \rho_{-i})$ denotes the pay-off function of player J_i; ρ_i denotes the control packet transmission probability by J_i and ρ_{-i} denotes the control packet transmission probabilities of other players except J_i. The term $U_i(\rho_i)$ in the Equation (6.1) denotes the *utility function*, and the other two terms i.e., $\frac{1}{(1-MBR_i(\rho_i,\rho_{-i}))}$ and $E_{TX_i}(\rho_i)$ denote the *price functions* correspond to *shared/minimal cell busy ratio* and *energy consumption*, respectively. $MBR_i(\rho_i, \rho_{-i})$ is the observed minimal cell busy ratio by player J_i, and E_{TX_i} is the required amount of energy to transmit a control packet with respect to its current residual energy. The values of $U_i(\rho_i)$, $MBR_i(\rho_i, \rho_{-i})$ and E_{TX_i} are calculated as follows,

$$U_i(\rho_i) = log\ (\rho_i + 1) \tag{6.2}$$

$$\begin{aligned} MBR_i(\rho_i, \rho_{-i}) &= \frac{Busy\ minimal\ cell}{Busy\ minimal\ cell + Idle\ minimal\ cell} \\ &= \frac{\sum_{t=0}^{W} 1 - (1-\rho_i)^n}{\sum_{t=0}^{W} 1 - (1-\rho_i)^n + (1-\rho_i)^n} \\ &= 1 - (1-\rho_i)^n \end{aligned} \tag{6.3}$$

$$E_{TX_i} = \frac{(\rho_i \times E_{tx})}{RE_i} \tag{6.4}$$

We use logarithmic *utility function* in Equation (6.2) because it has strict concave property and its second derivative is always negative. The value of MBR is calculated using the packet transmission probabilities of the joined nodes and number of occurred minimal cells in an interval W as shown in Equation (6.3). A player considers a minimal cell busy if it receives any control packet transmitted by other joined nodes or the player transmits itself, otherwise, the minimal cell is considered as idle. Similarly, in Equation (6.4), E_{tx} is the required energy to transmit a packet and RE_i is the residual energy of player J_i. The overall pay-off function of player J_i can be derived from Equation (6.1) as follows,

$$\psi_i(\rho_i, \rho_{-i}) = \alpha_i log\ (\rho_i + 1) - \frac{\beta_i}{(1-\rho_i)^n} - \frac{\gamma_i \rho_i E_{tx}}{RE_i} \tag{6.5}$$

where α_i, β_i, and γ_i are the preference parameters for player J_i with positive values for the

utility function and *price functions*, respectively.

6.2.1 Solution of the Game to Find its Optimal Solution

The procedure for calculating the optimal solution (ρ_i^*) for the proposed game G is described in this section. At first, it is important to check the existence of Nash equilibrium point in the proposed game G because G can have a solution only when there exists Nash equilibrium point. Further, the solution of the proposed game should be unique, and unique solution is possible when there exists only one unique Nash equilibrium point. We mention these two conditions in Lemma 6.2.1 and Lemma 6.2.2 with their corresponding proofs as follows:

Lemma 6.2.1. *The proposed pay-off function $\psi_i(\rho_i, \rho_{-i})$; $\forall J_i \in JN$ is strictly concave in its strategy space S_i, $\forall J_i \in JN$ and the game G has at least one Nash equilibrium point.*

Proof. The Nash equilibrium of the proposed game G exists when a player can not improve its pay-off function by changing its strategy while the other players do not change their strategies. The strategy space for player J_i is closed and bounded i.e., $S_i = [0, 1]$, which implies that the set S_i is compact $\forall i \in \mathbb{N}$. Now Considering x, y to be two points in the strategy space S_i in a Euclidean space where $S = \prod_{i=1}^{n} \rho_i$, $\forall J_i \in JN$, the strategy set S_i convex if for any x, $y \in S_i$ and $\eta = [0, 1]$, $\eta x + (1 - \eta)y \in S_i$. Now, the Hessian matrix H of $\psi_i(\rho_i, \rho_{-i})$, $\forall J_i \in JN$, can be defined as follows,

$$H = \begin{bmatrix} H_{11} & H_{12} & .. & H_{1z} \\ H_{21} & H_{22} & .. & H_{2z} \\ .. & .. & .. & .. \\ H_{z1} & H_{z2} & .. & H_{zz} \end{bmatrix}$$

Or, we can write the Hessian matrix in short as follows,

$$H = [H_{rs}]_{r \times s} \tag{6.6}$$

where $H_{rs} = \frac{\delta^2 \psi_r}{\delta \rho_r \delta \rho_s}$, $\forall H_{rs} \in JN$. Now, using the Equation (6.5), we can calculate H_{rs} as

follows,

$$H_{rs} = \begin{cases} -\frac{\alpha_r}{(\rho_r+1)^2} < 0 & \text{if } r = s, \forall H_{rs} \in JN \\ 0 & \text{if } r \neq s, \forall H_{rs} \in JN \end{cases} \tag{6.7}$$

From the Equation (6.7), it can be seen that the leading principal minor of H is negative definite for all ρ_i, $\forall J_i \in JN$, which implies that it reaches a local maximum as shown in Equation 6.7 [92]. Thus, from the work in [92], we can say that the function $\psi_i(\rho_i, \rho_{-i})$, $\forall J_i \in JN$, is a strictly concave function. Now, using the Rosen's theorem [93] (Theorem 1), it can be further concluded that the proposed game G has at least one Nash equilibrium point as it is a n-person concave game. However, to proof the existence of a unique solution in game G, the weighted non-negative sum ($\delta(\rho_i, \rho_{i-1}; m)$) of ψ_i must be diagonally strictly concave. □

Lemma 6.2.2. *The solution of the proposed game G is unique as it has a unique Nash equilibrium point.*

Proof. To prove the existence of a unique solution in game G, the weighted non-negative sum $\delta(\rho_i, \rho_{-i}; m)$ of ψ_i must be diagonally strictly concave. For that, using the Equation (6.5), we can write $\forall J_i \in JN$,

$$\frac{\delta\psi_i}{\delta\rho_i} = \nabla \psi_i = \frac{\alpha_i}{\rho_i + 1} - \frac{n\beta_i}{(1-\rho_i)^{n+1}} - \frac{\gamma_i E_{tx}}{RE_i} \tag{6.8}$$

Based on the Rosen's Theorem [93], the weighted non-negative sum of the proposed pay-off function can be calculated as follows;

$$\delta(\rho_i, \rho_{-i}; m) = \sum_{i=1}^{n} m_i \psi_i(\rho_i, \rho_{-i}), m_i \geq 0 \tag{6.9}$$

The pseudogradient of $\delta(\rho_i, \rho_{-i}; m)$ can be written as,

$$g(\rho_i, \rho_{-i}; m) = [m_i \nabla \psi_i]_{n \times 1} \tag{6.10}$$

Similarly, the Jacobian matrix $M(\rho_i, \rho_{-i}; m)$ of $g(\rho_i, \rho_{-i}; m)$ can be written as follows,

$$M = \begin{bmatrix} M_{11} & M_{12} & .. & M_{1y} \\ M_{21} & M_{22} & .. & M_{2y} \\ .. & .. & .. & .. \\ M_{y1} & M_{y2} & .. & M_{yy} \end{bmatrix}$$

Or, we can write as follows,

$$M(\rho_i, \rho_{-i}; m) = [m_i H_{is}]_{n \times n};\ M_{is} \in JN;\ m_i > 0 \tag{6.11}$$

where $m_i H_{is}$ is calculated as follows,

$$m_i H_{is} = \begin{cases} -\frac{\alpha_i m_i}{(\rho_i+1)^2} < 0 & \text{if } i = s; \forall H_{is} \in JN \\ 0 & \text{if } i \neq s; \forall H_{is} \in JN \end{cases} \tag{6.12}$$

From the Equations (6.11) and (6.12), it can be written that $[M(\rho_i, \rho_{-i}; m) + M^T(\rho_i, \rho_{-i}; m)]$ is again negative definite in the entire strategy space. Hence, the Rosen's theorem [93] confirms that the function $\delta(\rho_i, \rho_{-i}; m)$ is diagonally strictly concave, and so the proposed game G has a unique Nash equilibrium point and it has a unique solution. □

It is already proven that the game G has a unique Nash equilibrium point and a solution. Each player choose a strategy to maximizing its control packet transmission probability (ρ_i) per slotframe i.e., its pay-off function. Therefore, the following *constrained non-linear optimization* problem is constructed to solve the proposed game G,

$$\begin{aligned} &\underset{\rho_i \in S_i}{maximize}\ \ \psi_i(\rho_i, \rho_{-i}) \\ &subject\ to\ 0 \leq \rho_i \leq \rho_i^{max}; \forall J_i \in JN \end{aligned} \tag{6.13}$$

To solve this optimization problem, *Lagrange function* $L_i(\rho_i, a_i, b_i)$ is used for player $J_i, \forall J_i \in JN$, as follows:

$$L_i(\rho_i, a_i, b_i) = \psi_i(\rho_i, \rho_{-i}) + a_i \rho_i + b_i(\rho_i^{max} - \rho_i) \tag{6.14}$$

where a_i and b_i are the Lagrange multipliers. The *Karush-Kuhn-Tucker (KKT)* conditions

for player J_i are as follows:

$$a_i, b_i \geq 0$$
$$\rho_i \geq 0$$
$$\rho_i^{max} - \rho_i \geq 0$$
$$\bigtriangledown_{\rho_i} \psi_i(\rho_i, \rho_{-i}) + a_i \bigtriangledown_{\rho_i}(\rho_i) + b_i \bigtriangledown_{\rho_i}(\rho_i^{max} - \rho_i) = 0$$
$$a_i\rho_i, b_i(\rho_i^{max} - \rho_i) = 0$$

The Equation (6.14) has three unknown variables i.e., ρ_i, a_i, and b_i. Using the KKT conditions, the optimal control packet transmission probability (ρ_i^*) of player J_i can be calculated by solving the Equation (6.14) as follows:

$$\rho_i^* = \begin{cases} 0 & \text{; if condition 1} \\ \rho_i^{max} & \text{; if condition 2} \\ \frac{\alpha_i}{\frac{n\beta_i}{\chi_i} + \frac{\gamma_i E_{tx}}{RE_i}} - 1 & \text{; otherwise} \end{cases} \tag{6.15}$$

where condition 1 is,

$$\frac{n\beta_i}{\alpha_i - \frac{\gamma_i E_{tx}}{RE_i}} \geq \chi_i \tag{6.16}$$

The term $\chi_i = (1-\rho_i)^{n+1}$ in Equation (6.16) denotes the *minimal cell idle ratio* observed by the player J_i; $\forall J_i \in JN$. The condition 1 is obtained considering $\rho_i = 0$ and $b_i = 0$, which implies

$$\begin{aligned} \Rightarrow & \frac{\alpha_i}{\rho_i + 1} - \frac{n\beta_i}{(1-\rho_i)^{n+1}} - \frac{\gamma_i E_{tx}}{RE_i} + a_i = 0 \\ \Rightarrow & a_i = -\frac{\alpha_i}{\rho_i + 1} + \frac{n\beta_i}{(1-\rho_i)^{n+1}} + \frac{\gamma_i E_{tx}}{RE_i} \end{aligned} \tag{6.17}$$

The solution $\rho_i = 0$ is feasible, if $a_i > 0$ holds, therefore,

$$\begin{aligned} -\frac{\alpha_i}{\rho_i + 1} + \frac{n\beta_i}{(1-\rho_i)^{n+1}} + \frac{\gamma_i E_{tx}}{RE_i} \geq 0 \\ \frac{n\beta_i}{\alpha_i - \frac{\gamma_i E_{tx}}{RE_i}} \geq \chi_i \end{aligned} \tag{6.18}$$

Similarly, the condition 2 is obtained by considering $\rho_i = {\rho_i}^{max}$, $a_i = 0$, and here also $\rho_i = {\rho_i}^{max}$ is feasible only when the condition, $b_i > 0$ holds. Therefore, the final obtained condition 2 is as follows:

$$\frac{n\beta_i}{\frac{\alpha_i}{\rho_i^{max}+1} - \frac{\gamma_i E_{tx}}{RE_i}} \leq \chi_i \tag{6.19}$$

Finally, the "*otherwise*" condition denotes all the conditions which do not satisfy condition 1 and condition 2 and is obtained considering the Lagrange multiplier constants equals to zero, i.e., $a_i = 0$, $b_i = 0$, and $(0 < \rho_i < \rho_{max})$.

Proposition 6.2.1. *The value of ρ_i^* lies between 0 and 1 in the "otherwise" condition of Equation (6.15) always.*

Proof. The value of ρ_i^{max}=1, because it denotes the maximum control packet transmission probability in minimal cell of player J_i. Now, let us assume the value of ρ_i^* in the "*otherwise*" condition as follows:

$$\begin{aligned}
&\rho_i^* \geq 1 \\
&\Rightarrow \frac{\alpha_i}{\frac{n\beta_i}{\chi_i} + \frac{\gamma_i E_{tx}}{RE_i}} - 1 \geq 1 \\
&\Rightarrow \alpha_i \geq 2 \times (\frac{n\beta_i}{\chi_i} + \frac{\gamma_i E_{tx}}{RE_i}) \\
&\Rightarrow \chi_i \geq \frac{n\beta_i}{\frac{\alpha_i}{\rho_i^{max}+1} - \frac{\gamma_i E_{tx}}{RE_i}}
\end{aligned}$$

But, after simplification, it can be seen that the value of ρ_i^* in "*otherwise*" condition is contradicting with the condition 2 of Equation (6.15). Therefore, ρ_i^* in "*otherwise*" condition can not be more than one. Similarly, let's assume

$$\begin{aligned}
&\rho_i^* \leq 0 \\
&\frac{\alpha_i}{\frac{n\beta_i}{\chi_i} + \frac{\gamma_i E_{tx}}{RE_i}} - 1 \leq 0 \\
&\Rightarrow \alpha_i \leq (\frac{n\beta_i}{\chi_i} + \frac{\gamma_i E_{tx}}{RE_i}) \\
&\Rightarrow \chi_i \leq \frac{n\beta_i}{\alpha_i - \frac{\gamma_i E_{tx}}{RE_i}}
\end{aligned}$$

But, here also, the value of ρ_i^* in "*otherwise*" condition contradicts with the condition 1 of

Equation (6.15). So, the value of ρ_i^* in "*otherwise*" condition can not be less than 0. □

6.3 Game Theory based Congestion Control (GTCC)

In this section, the solution of the proposed game G (i.e. Equation (6.15)) is used to propose a novel *game theory based congestion control* (GTCC) scheme. As all the nodes keep their radios active in the minimal cell either for transmission or reception, so a joined node can calculate the *minimal cell busy ratio* (MBR) for a time interval by itself. It can be done by dividing the total number of minimal cells in which the joined node has received control packets, or the node transmits itself by the total number of minimal cells present in the interval. It does not require any implicit signal to/from other neighbors. Now, by subtracting this MBR value from 1, it gives the *minimal cell idle ratio* i.e., χ_i, which is used in Equation (6.15). Now, using the value of χ_i and Equation (6.15), the joined node calculates its optimal control packet transmission probability (ρ_i^*) in a minimal cell.

As ρ_i^* is the control packet transmission probability in a minimal cell, therefore, the value of ρ_i^* is converted into number of minimal cells (referring as, *slotframe window*) SW_i as follows:

$$SW_i = \begin{cases} SW_{max} & ; \text{ if } \rho_i^* = 0 \\ min(\left\lceil \frac{1}{\rho_i^*} \right\rceil, SW_{max}) & ; \text{ otherwise} \end{cases} \tag{6.20}$$

where SW_i denotes the number of slotframes that node J_i has to wait to transmit its next control packet and SW_{max} denotes the maximum *slotframe window* length, which is same for all the nodes. Therefore, a node can not transmit its control packets in every consecutive minimal cell when there is high congestion, which ultimately reduces congestion. Even though, GTCC does not modify the control packets generation rate of the joined nodes, it restrains their transmissions by forcing them to wait for their respective *SW* periods, which results in less congestion in minimal cell. Note that GTCC is run only by the joined nodes to reduce congestion in minimal cell. Algorithm 8 describes the steps of the proposed GTCC.

Algorithm 8 Game Theory based Congestion Control

```
Set χ_i calculation interval I
if I ends then
    Calculate the value of χ_i = (1 − ρ_i)^{n+1}
    Calculate the value of ρ_i^* using Equation (6.15)
    Calculate the value of SW_i using Equation (6.20)
    Wait SW_i amount of time
    if SW_i number of slotframe ends then
        Attempt to transmit next buffered control packet
    end if
    Wait I amount of time
end if
```

6.4 Parameter Selection and Theoretical Analysis of GTCC

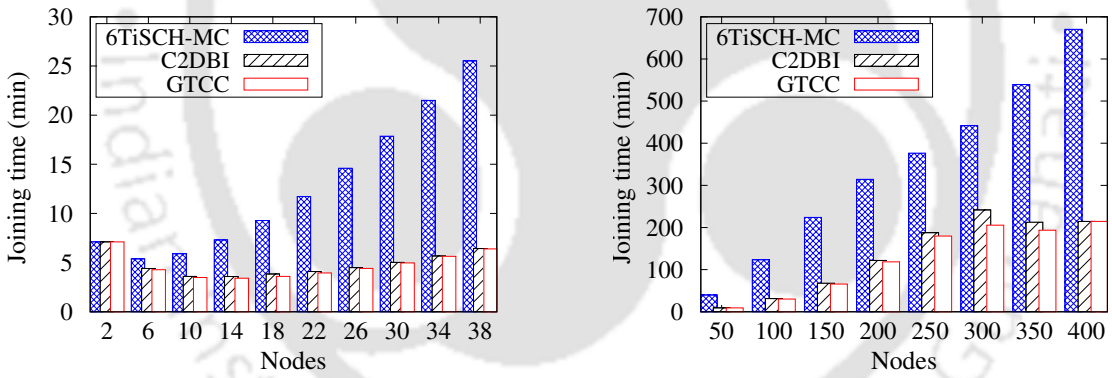
The analytical model described in the Chapter 3 (Section 3.2) of this thesis is used to evaluate how the parameters (i.e., α, β, γ) of the proposed game G affect the joining time and energy consumption of nodes in 6TiSCH networks. The same Markov Chain based analytical model is used to compare GTCC with the 6TiSCH-MC [40] standard, and our previous scheme C2DBI (ref. Chapter 3). For evaluation, the slotframe length (L) is taken as 101 *timeslots*, with each *timeslot* duration of $10ms$. We consider the minimum EB generation interval as $4 \times slotframe\ length$ for C2DBI, which is same as the EB generation interval of 6TiSCH-MC, and the maximum value of SW is taken as $10 \times slotframe\ length$, which is same as the maximum EB generation interval of C2DBI. The values of other variables are as follows: number of channels $N_c = 16$, packet loss probability $P_{loss} = 0.2$, number of Trickle states $N_D = 8$.

At the beginning, a single-hop network topology with 20 nodes is considered, where all the nodes are neighbors to each other. The results are obtained by varying any of the preference parameters while keeping the other two parameters constant. Observed results are tabulated in Table 6.1. When the *price function* for the minimal cell congestion i.e., β is less, all the nodes transmit their control packets more frequently. Hence, it congests the minimal cell, and so the joining time of the pledges increases. On the other hand,

Table 6.1: *Joining time under varied preference parameter*

Constant	$\alpha = 5, \gamma = 0.1$				$\beta = 0.5, \gamma = 0.1$		
Varied	β				α		
	0.05	0.1	0.5	1	1	5	10
Joining time (min)	3.91	3.77	3.75	3.86	3.78	3.758	3.82

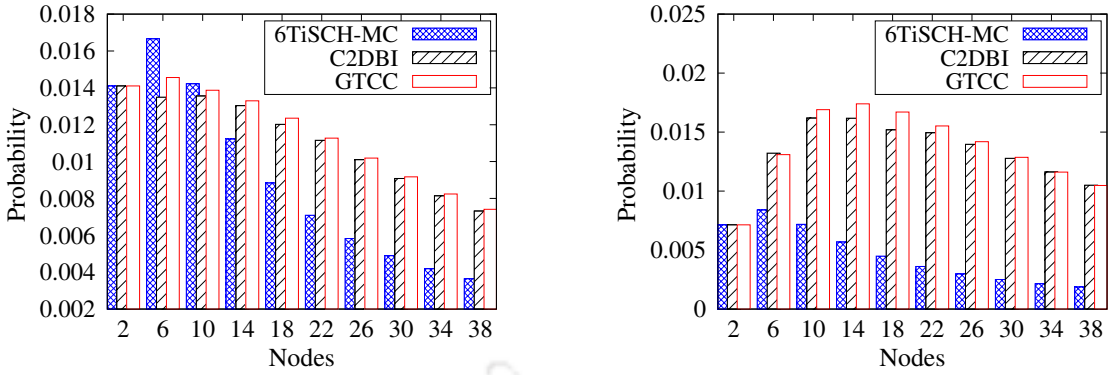
when the value of β is equal to 1, nodes do not transmit their control packets frequently because of the high price. Though it reduces congestion but increases the waiting time of the pledges to get the control packets, which again results in longer network formation. On the other hand, when the value of α is 1, the nodes transmit less frequently, resulting in longer formation time. However, when $\alpha = 10$, nodes transmit their control packets more frequently. This results in higher congestion in the minimal cell, and so the joining time of nodes increases. Note that the varied values of γ with the constant values of α and β have a similar effect like β on network formation time. Therefore, for analytical comparison of



(a) *6TiSCH formation time* **(b)** *Scalability experiment*

Figure 6.1: *Analytical results on 6TiSCH joining time*

GTCC with 6TiSCH-MC [40] and C2DBI, we consider the best values of α, β and γ as 5.0, 0.5, and 0.1, respectively. Figure 6.1a shows the 6TiSCH formation time, and Figure 6.1b shows the 6TiSCH formation time of few bigger networks. On the other hand, Figure 6.2a and 6.2b show the EB and DIO reception probabilities per slotframe of a pledge. In all the three schemes, the EB and DIO reception probabilities of the pledges are very less in small networks as only few nodes transmit them. This results in longer formation of the networks as shown in Figure 6.1a .

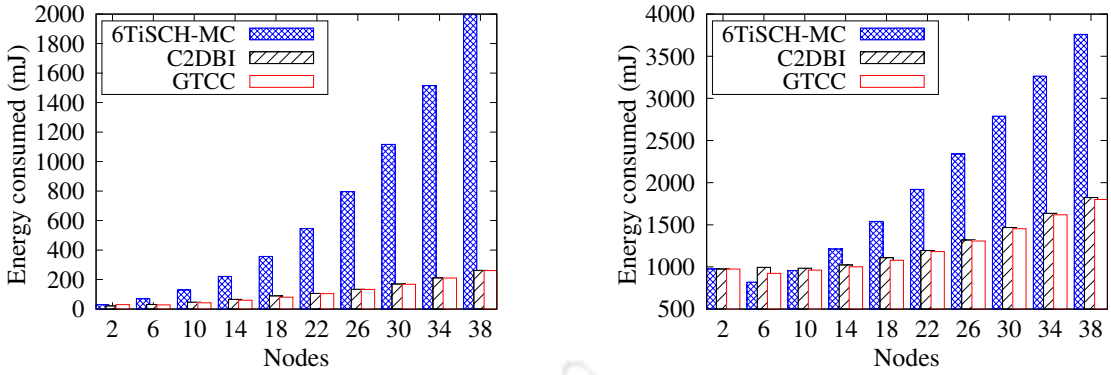


(a) *EB reception probability* **(b)** *DIO reception probability*

Figure 6.2: *Analytical results on EB and DIO reception probabilities of a pledge*

However, by increasing the number of nodes (up to 7 nodes), the joined nodes transmit a sufficient number of control packets without congesting the minimal cell, which helps in faster formation. Nevertheless, further increment of the nodes congests the minimal cell (worst in 6TiSCH-MC), which again results in low EB and DIO reception probabilities of the pledges, and so delays the formation of the networks. Both the C2DBI and GTCC reduced the congestion in minimal cells by limiting the control packet transmission and so achieved better results than 6TiSCH-MC. Although GTCC improves the joining time compared to C2DBI, it is not significant. This is because the complete behavior of DIO packet generation Trickle algorithm is not considered in the analytical model of the work in [45], and so the congestion due to DIO transmission is not so significant. Therefore, using the analytical model, GTCC only controls the EB transmission probability like C2DBI but does not control the DIO transmission probability. Hence, both C2DBI and GTCC show almost similar results.

As longer joining time of the pledges also increases the energy consumption of the nodes; therefore, from Figure 6.3a and 6.3b, it can be seen that the proposed scheme GTCC reduces the energy consumption of both the joined nodes and the pledges compared to that in existing benchmark schemes.



(a) *Pledge Energy consumption* **(b)** *Joined node energy consumption*

Figure 6.3: *Analytical results on energy consumption of the nodes*

6.5 Testbed Evaluation of GTCC

6.5.1 Experimental Setup

The proposed GTCC is implemented on Contiki-NG OS. Along with the 6TiSCH-MC standard [40], we compare our proposed GTCC with our previous scheme C2DBI (ref. Chapter 3) for showing comparative performance study. The values of different parameters of all the schemes are taken the same as the values mentioned in Section 6.4. We use the FIT IoT-LAB for running our testbed experiments. The various testbed experimental settings are mentioned in Table 6.2, and we use two different topologies for our testbed experiments. The first topology is a 5×5 grid topology, and the other is a 2×12 linear topology.

Table 6.2: *Experimental settings*

Parameter	Value
Operating System	Contiki-NG
Network Topologies	5×5 and 2×12
Testbed	FIT IoT-LAB, Grenoble
Testbed `mote` type	M3
Number of channels	16
Timeslot	10 ms
Slotframe length	101 *timeslots*
RPL version	RPL Lite
DIO interval	Trickle Algorithm
RPL Objective Function	MRHOF-ETX
Experiment duration	60 min

6.5.2 Performance Metrics

We consider three metrics for the evaluation of the proposed scheme GTCC. The first metric is *TSCH synchronization time*. It is the time when a pledge receives its first valid EB frame. Before receiving an EB, a pledge needs to keep its radio active all the time, which consumes a significant amount of energy. A pledge knows the location of minimal cell after receiving a valid EB. So, it keeps radio active only in the minimal cell to save energy. So, the TSCH synchronization time is an important metric to consider during network formation.

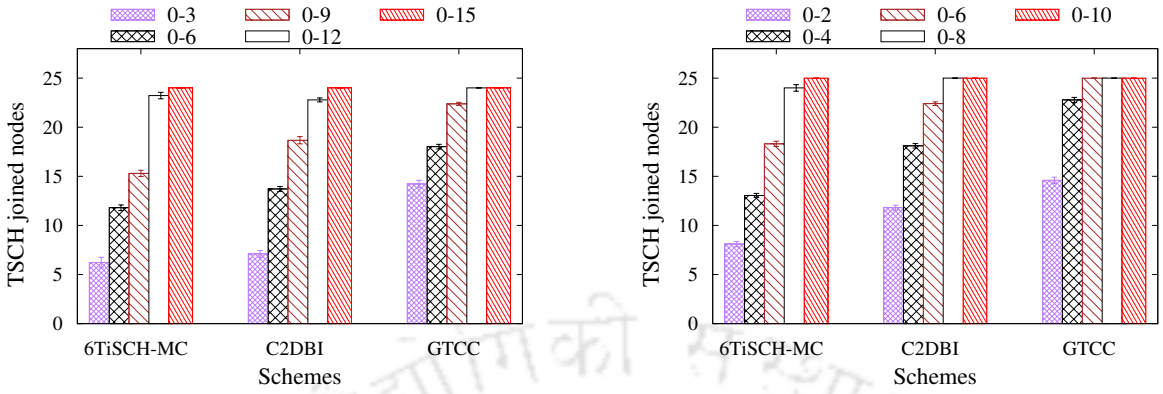
6TiSCH joining time is considered as the second metric, which denotes the time when the TSCH synchronized node receives the DIO packet from its parent. Note that the 6TiSCH joining time is the same time for the node to join the DODAG. A node is allowed to transmit its control packet for further expansion of the network after joining the 6TiSCH network. Hence, this 6TiSCH or DODAG joining time of the nodes is essential to consider during network formation.

As it is already mentioned that a pledge consumes more energy before it gets synchronized with the TSCH network. Therefore, *energy consumption* is taken into consideration as the third metric. Although, the *radio duty cycle* (RDC) of a node reduces from 100% to 1% after getting an EB frame, its child node(s) still have the RDC of 100%. It is because neither the node has started transmitting its control packets nor the child receive EB frame. Therefore, quick and sufficient transmissions of both EB and DIO packet are necessary to increase the lifetime of a network.

6.5.3 Results and Discussion

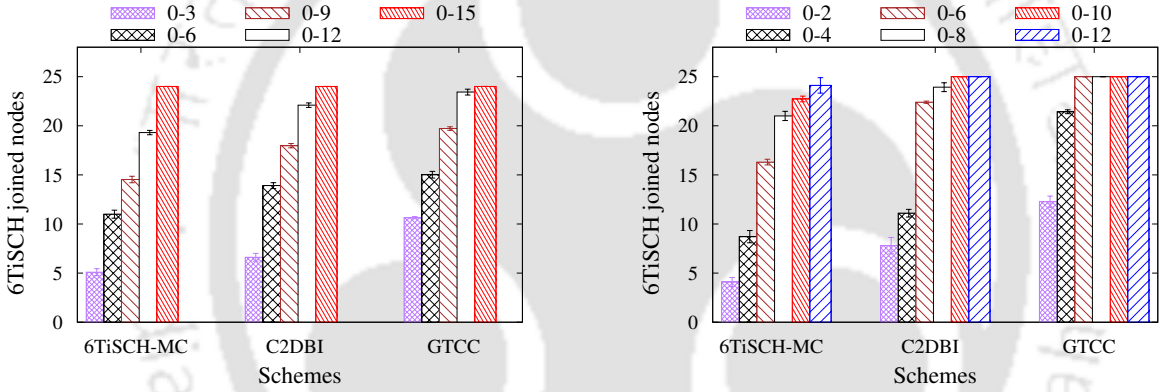
The received results from the testbed experiments are shown using 95% confidence interval. Each bar in Figure 6.4 shows the number of *TSCH synchronized nodes*, in Figure 6.5 shows the number of *6TiSCH joined nodes*, and in Figure 6.6 shows the *average energy consumption* under various time intervals (e.g., 0-3, 0-2, 2-4, 9-12). The measuring units of time intervals are minutes. Figure 6.4a and 6.4b show the number of TSCH synchronized nodes for both the 2×12 linear and 5×5 grid topologies, respectively. Similarly, Figure 6.5a and 6.5b

show the number of 6TiSCH joined nodes in different time intervals. It can be seen from



(a) *In* 2×12 *topology* **(b)** *In* 5×5 *topology*

Figure 6.4: *Testbed results on TSCH joining time.*



(a) *In* 2×12 *topology* **(b)** *In* 5×5 *topology*

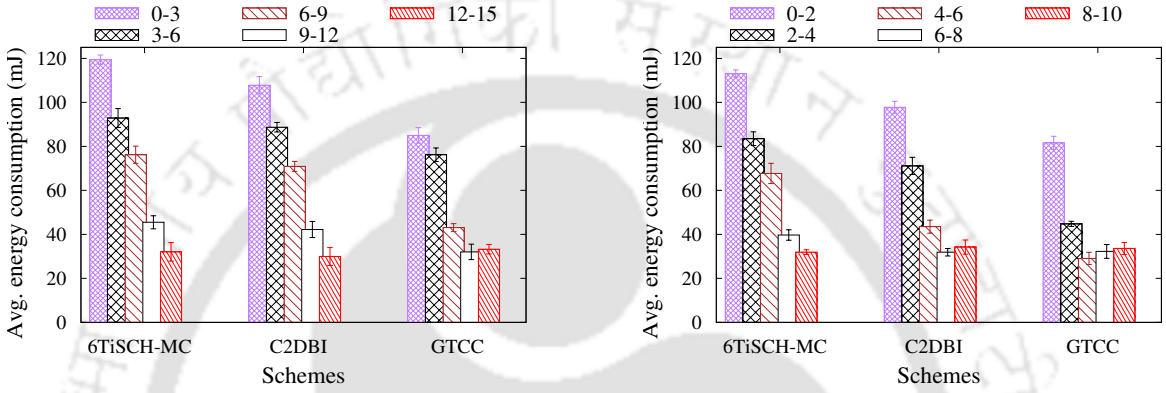
Figure 6.5: *Testbed results on 6TiSCH joining time.*

the results that GTCC outperforms both 6TiSCH-MC and C2DBI in terms of number of joined TSCH synchronized nodes and 6TiSCH joined nodes in different time intervals. It is because GTCC is used to maintain balance between the congestion in minimal cells and sufficient transmission of the EB and DIO packet in the networks during their formation.

Although C2DBI reduces the congestion in minimal cell by dynamically changing the EB generation rate, that is not sufficient to fully reduce the congestion in minimal cells. It is because C2DBI does not control the DIO transmission rate, which is necessary. When a node transmits DIS request for DIO packet, nearby joined node(s) reset their Trickle algorithm, which turns outburst transmission of DIOs and congests the minimal cell. Therefore, to deal with this kind of scenario, GTCC restrains the transmission of joined nodes by forcing

them to wait for their respective *SW* periods so that congestion gets reduced. Hence, it significantly achieves better performance compared to C2DBI. 6TiSCH-MC suffers the most because it does not provide any mechanism to deal with the congestion in minimal cell.

The average energy consumption by a pledge in different time intervals is shown in Figure 6.6a and 6.6b for the 2×12 and 5×5 topologies, respectively. At the beginning



(a) *In* 2×12 *topology* **(b)** *In* 5×5 *topology*

Figure 6.6: *Testbed results on energy consumption.*

of the formation process, all the schemes consume more energy. It is because, initially, most of the nodes do not get synchronized with the TSCH networks. They randomly scan on different channels by keeping their radios active for EBs, which causes more energy consumption. Therefore, in 6TiSCH-MC, the average energy consumption of the nodes is more because only few nodes immediately get synchronized with the TSCH networks due to congestion in minimal cell. On the other hand, in GTCC, nodes do not need to wait for a longer time to receive their first EBs because of the less congestion in minimal cell, which significantly saves nodes' energy. All three schemes consume almost same energy when their formation process is over as nodes keep their radios active only in the minimal cell.

6.6 Summary

In this chapter, we proposed a game theoretic approach to reduce congestion in minimal cell (shared cell) so that performance of 6TiSCH network can be improved during its forma-

tion. For this, we designed a non-cooperative game, where the joined nodes act as *players*. The players want to maximize their profit irrespective of the strategies of other neighboring players. In the pay-off function, control packet transmission probability is considered as a *utility function*, and the minimal cell busy ratio and energy of a node are considered as *price functions*. The obtained solution of the proposed game using Lagrange multiplier and Karush-Kuhn-Tucker (KKT) conditions were further used in a newly proposed *game theory based congestion control* (GTCC) scheme. GTCC calculates *Slotframe Window (SW)* size of the nodes to restrain them in transmitting control packets frequently so that congestion in the minimal cell can be reduced without any signaling overhead. We used the analytical model mentioned in our Chapter 3 of this thesis to find out the best combination of preference parameters used in the proposed game. We used the same analytical model to compare the proposed GTCC with the benchmark schemes such as 6TiSCH-MC [40] and our previous scheme C2DBI (ref. Chapter 3). Furthermore, the GTCC was implemented on Contiki-NG and evaluated using FIT IoT-LAB testbed. The received results from both the theoretical analysis and the testbed experiments showed that GTCC significantly improves the joining time and energy consumption of the pledges.

In Chapters 3, 4, 5, and 6, we considered the same resource allocation (i.e., only one shared/minimal cell per slotframe) policy mentioned by 6TiSCH-MC. However, to improve the performance of 6TiSCH network formation significantly, the number of shared cells per slotframe can be increased as mentioned in [45]. Therefore, in the next chapter, we leverage the available channels at a single timeslot of each slotframe up to their full extent to increase the number of shared cells per slotframe, which significantly improves the performance of 6TiSCH network formation.

"The cure for boredom is curiosity. There is no cure for curiosity"

~Dorothy Parker (1893–1967)

7

Autonomous Allocation and Scheduling of Shared Cell

We observe that, including 6TiSCH-MC, the existing schemes did not use all the available cells vertically, and so, all the available physical channels at the timeslot where the minimal/shared cell resides i.e., at timeslot zero. It results in underutilization of channel resources, and thus, higher network formation time. Therefore, this chapter proposes two schemes: *Autonomous allocation and scheduling of minimal cell* (*TACTILE*) and Time-Variant RGB (*TRGB*) to leverage all the available cells (so, channel) at timeslot zero of each slotframe, and so to improve the performance of 6TiSCH network. Basically, both the schemes increase the number of shared cells (or minimal cells) per slotframe in 6TiSCH

network without affecting the data transmission cells and radio duty cycles of the nodes. We evaluate both *TACTILE* and *TRGB* using Markov Chain based analytical model and testbed experiments in FIT IoT-LAB.

7.1 Introduction

As mentioned before, IETF released the 6TiSCH-MC [40] standard for describing minimal resource allocation during 6TiSCH network bootstrapping. 6TiSCH-MC mentioned that all the control packets (i.e., for bootstrapping and maintaining the network) should be transmitted in the *minimal cell* of a *slotframe*. The shared/minimal cell of a slotframe is shown in Figure 7.1. To maintain synchronization, the default allocation policy forces all nodes to use minimal cell at the same location `slot offset=0`, `channel offset=0` of a slotframe. Therefore, the position of minimal cell for all the nodes of a network remains same i.e., at the location ($[0, 0]$). The remaining cells starting from timeslot one of a slotframe are called *managed cells*, and they are used for data transmission using a communication schedule assigned by a scheduling function such as [41] and [42].

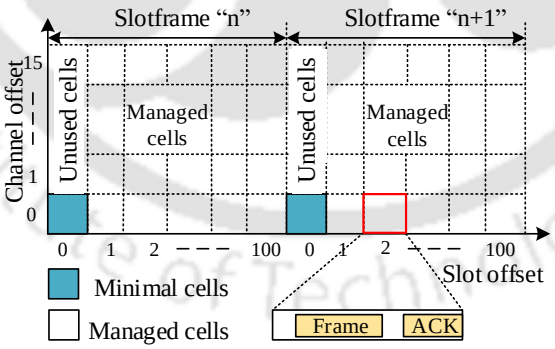


Figure 7.1: *Position of shared slot and minimal cell in slotframe*

Vallati *et al.,* [45] revealed that the performance of 6TiSCH network formation degrades when more pledges join in the network due to the allocation of only one shared cell per slotframe for control packet transmissions. To address this problem, the authors in [45] increased the number of shared cells per slotframe dynamically i.e., depending on the number

of control packets generated in a network. However, the additional shared cells are added in different timeslots of a slotframe, and thus, the default data transmission schedule and cells are affected by their dynamic allocation (ref. Figure 2.7 in Chapter 2). Moreover, as mentioned before, this allocation scheme is inefficient in energy consumption as the nodes need to keep their radios active in the allocated shared cells of a slotframe.

Because of the fixed position of the minimal cell at $[0, 0]$, only a single channel offset is used corresponding to that shared timeslot. The usage of a single channel offset has several disadvantages while multiple channels are available. (1) The other channel offsets remain unused in `slot offset 0` (i.e., shared slot). These channel offsets, so the corresponding physical channels are not used even for data packet transmission. Figure 7.1 shows the unused cells in `slot offset` 0. In brief, none of the existing schemes used all the channel offsets at `slot offset 0`. Therefore, all the channel offsets remain unused except one, as shown in Figure 7.1. (2) All joined nodes participate in contention at the same physical channel for transmitting their control packets. This scenario reduces the probability of getting EB by the pledges even when more joined nodes are available because the pledges might continue scanning in different physical channels. Furthermore, it increases congestion in one single channel.

Therefore, in this chapter, we attempt to leverage the available channels at the timeslot zero of each slotframe up to their full extent to improve the performance of network formation. In brief, this chapter proposes two techniques to improve the pledges' joining process by efficiently utilizing the unused cells (shown in Figure 7.1) as minimal cells. However, the main challenge is to utilize the available cells at timeslot zero as there is a minimal cell at [0,0] already scheduled for network bootstrapping. For this, at first, we propose *autonomous allocation and scheduling of minimal cell* (*TACTILE*). *TACTILE* modifies the static allocation policy of the minimal cell. Instead of allocating minimal cell at fixed position [0,0] for all the nodes in the network, *TACTILE* autonomously allocates in any cell at timeslot zero based on the given allocation policy. This modification increases the number of shared cells used in a network at `slot offset 0`. In such allocation, the position of min-

imal cells varies from node to node. Therefore, there is a need for synchronization among them. A hierarchical scheduling technique is proposed to maintain synchronization among the nodes. *TACTILE* improves the EB reception probabilities of the pledges. Further, it reduces congestion in shared cell and does not hamper data packet transmission as the modified allocations are done in unused cells.

Later on, we find that *TACTILE* creates unstable networks and allows repeated control packet collision. Therefore, we propose Time-Variant RGB (*TRGB*) model as an enhanced version of *TACTILE*. Although *TRGB* autonomously allocates the minimal cell like *TACTILE*, allocation made by *TRGB* is time-variant so that nodes change their allocated minimal cells after each slotframe to alleviate repeated control packet collision. Again, TRGB allocates one common cell for all the nodes to exchange RPL control traffics. This common cell separates the RPL traffics from other control traffics such as EB, keep-alive, which helps in maintaining a stable network. Therefore, for scheduling the added time dependable minimal cells and providing tight interaction with the RPL for both upward and downward routing, we propose an autonomous scheduling algorithm. This scheduling algorithm manages conflict-free scheduling of the allocated minimal cells of the nodes as well as schedules different radio states i.e., transmission (`Tx`), reception (`Rx`), or `idle` of the parent-child pairs efficiently. Both the *TACTILE* and *TRGB* do not have signaling overhead for allocating and scheduling the minimal cells per slotframe. In brief, the contributions of this chapter are summarized as follows:

- An *autonomous minimal cell allocation* (*ALLOT*) scheme is proposed for faster transmission of control packets.
- A *hierarchical odd-even minimal cell scheduling* (*CHOICE*) scheme is proposed for maintaining synchronization between a transmitter and a receiver.
- Markov Chain based theoretical analysis, and testbed experiments are performed for evaluating the proposed *TACTILE* scheme, which combines both *ALLOT* and *CHOICE* schemes.

- A *time-variant autonomous allocation of minimal cell* (*INSTALL*) scheme is proposed for changing the location of the autonomously allocated minimal cells periodically to alleviate repeated collision.
- A *Red-Green-Blue* (*RGB*) scheme is proposed by which nodes can autonomously manage conflict free scheduling of the added minimal cells and provide tight intereaction with RPL.
- Markov Chain based theoretical analysis and testbed experiments are performed for evaluating the proposed (TRGB) model, which combines both the *INSTALL* and *RGB* schemes.

The rest of the chapter is organized as follows. In Section 7.2, we briefly describe our proposed *TACTILE* scheme. In Sections 7.3, and 7.4, we provide the theoretical analysis and testbed evaluation of *TACTILE*, respectively. In Section 7.5, we briefly describe our proposed *TRGB* scheme, followed by in Sections 7.6, and 7.7, we provide the theoretical analysis and testbed evaluation of *TRGB*, respectively. Finally, this chapter is summarized in Section 7.8.

7.2 Autonomous Allocation and Scheduling of Minimal Cell (TACTILE)

The work in [45] mentioned that the number of shared cells per slotframe allocated by 6TiSCH-MC is not sufficient to transmit all the generated control packets during 6TiSCH network bootstrapping. To validate this claim, we perform simulation experiments on the Cooja simulator. Two grid topologies i.e., 5×5 and 6×6 are used for simulation experiments, considering slotframe length equals to 101 *timeslots* and 10 *milliseconds* long timeslot. Following the 6TiSCH-MC, we consider only one shared cell per slotframe i.e., at `slot offset 0` to simulate in one scenario, and for second scenario, we consider two shared cells per slotframe at `slot offset 0` and `1`. Results show that the pledges join the

6TiSCH networks in less time in the second scenario compared to the first scenario. The formation time of both the network topologies i.e., 5×5 and 6×6 are improved by 37% and 23%, respectively, using two shared cells per slotframe. Hence, it can be claimed that the formation time of 6TiSCH networks can be improved by increasing the number of shared cells per slotframe. A similar type of improvement is also proved by the scheme DRA [45]. However, as mentioned in our previous chapters, DRA has several disadvantages, such as throughput, network lifetime.

Based on the above understanding, at the beginning, this chapter proposes an *autonomous allocation and scheduling of minimal cell* (*TACTILE*) for allocating multiple shared cells by leveraging two attributes – the location of node links in the DODAG topology and the availability of multiple channels at a fixed timeslot. *TACTILE* contains two sub-schemes. At first, *autonomous minimal cell allocation* (*ALLOT*) sub-scheme is proposed for distributing the allocation of minimal cell vertically at `slot offset 0` based on the nodes' position in the DODAG topology. It autonomously allocates at most three shared/minimal cells at `slot offset 0` to a joined node for transmitting different types of control packets. As all the shared/minimal cells are allocated at the same timeslot, only one cell can be used at a time by the joined node, and thus, it follows 6TiSCH-MC standard. However, unlike DRA, the number of shared cells per slotframe is increased by allocating the shared cells in one of those unused cells in `slot offset 0` intelligently. This utilization of multiple channels simultaneously gives better performance during network formation as it reduces frame collision.

Now the important question is *which minimal cell will be used and when by which node*? Further, *how does the receiver know which channel the transmitter is using currently*? In short, the *ALLOT* scheme creates synchronization issue. For solving it, we propose a *hierarchical odd-even minimal cell scheduling* (*CHOICE*) sub-scheme. So, the joined nodes should follow both the sub-schemes together to transmit their control packets. How the proposed sub-schemes are followed together is mentioned in the *TACTILE* framework. In the below subsections, both the subschemes and the framework are discussed in detail.

7.2.1 Autonomous Minimal Cell Allocation

Algorithm 9 describes the procedure for autonomous minimal cell allocation by the nodes of a 6TiSCH network. At first, only the Join Registrar/Coordinator (JRC) is there in the network, and thus, it assigns one minimal cell at `[0,hash(EUI64(JRC),`N_c`)]`. The function `hash(EUI64(XY),`N_c`)` returns a value between $[0, N_c - 1]$ based on the `EUI64`[1] address of the node XY. Here, N_c is the total number of channels available in the network. Detailed procedure of the hash function is described in [94]. However, any hash function can be used to allocate the minimal cell(s) to the nodes. If a pledge receives an EB frame from the JRC, then the pledge allocates a minimal cell at `[0,hash(EUI64(JRC),`N_c`)]` for both transmission (`Tx`) and reception (`Rx`). Otherwise, it allocates two minimal cells at `[0,hash(EUI64(parent),`N_c`)]` and `[0,hash(EUI64(grandparent),`N_c`)]` considering that it has received the EB frame from another joined node (not JRC) who is selected as its parent node. Here, `EUI64(parent)` and `EUI64(grandparent)` are the `EUI64` addresses of parent

Algorithm 9 Autonomous Minimal Cell Allocation

```
1: if node is JRC then
2:     Rx, Tx cell = [0,hash(EUI64(JRC),Nc)]
3: else if node is TSCH sync but not join DODAG then
4:     if parent of the node is JRC then
5:         Rx, Tx cell = [0,hash(EUI64(JRC),Nc)]
6:     else
7:         Rx cell=[0,hash(EUI64(parent),Nc)]
8:         Tx cell=[0,hash(EUI64(grandparent),Nc)]
9:     end if
10: else if  node is TSCH sync and join DODAG then
11:     Tx cell = [0,hash(EUI64(own),Nc)]
12: else                                  ▷ node is not TSCH sync or disconnected
13:     Scan channel for EB frame
14: end if
```

and grandparent of the pledge, respectively. The minimal cell at `[0,hash(EUI64(parent),`N_c`)]` is used for listening any control frame from both its parent and child nodes. Whereas, the minimal cell at `[0,hash(EUI64(grandparent),`N_c`)]` is used for transmitting control

[1]EUI64 (Extended Unique Identifier) is a method by which IPv6 addresses are automatically configured to the hosts. An IPv6 device will use the MAC address of its interface to generate a unique 64-bit interface ID (i.e., EUI64) by adding extra 16-bit.

frames such as JRQ and DIS to its parent. When a TSCH synchronized node gets a valid DIO packet, i.e., the node joins to DODAG of the 6TiSCH network, then it allocates one more minimal cell at `[0,hash(EUI64(own),`N_c`)]` to transmit its own control packets. Here, `EUI64(own)` is the `EUI64` address of the node itself. It is noteworthy that all the allocations are done at `slot offset 0`, and so, in the unused cells.

In this process, hash collision may occur when the number of joined nodes exceeds the available channels. The hash collision results in same minimal cell allocation to the colliding nodes. Note that the congestion in shared minimal cells occurs only when the hash colliding nodes simultaneously generate their control packets. The channel offsets are calculated based on nodes `EUI64` addresses, making the allocation distinct. Information about parent `EUI64` address is implicitly mentioned in EB frame, whereas, grandparent `EUI64` address can be added in the *information elements* (IE) of an EB frame. Hence, no other control packet needs to be transmitted to provide information about added minimal cells' location. Thus, *ALLOT* allocates extra shared cells without affecting the data transmission schedule and increasing the radio duty cycle of the nodes.

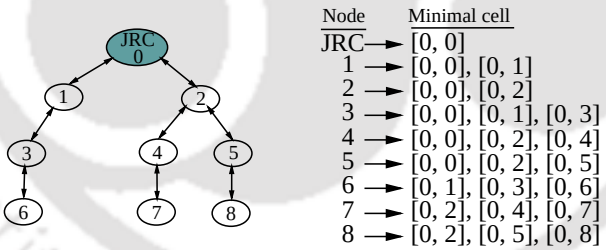


Figure 7.2: *Autonomous minimal cell allocation by ALLOT*

Figure 7.2 depicts an example of cell allocations by *ALLOT*. In the figure, node ⑥ allocates three shared minimal cells in `slot offset 0` which are $[0, 1]$, $[0, 3]$, and $[0, 6]$. Channel offsets 1, 3, and 6 are calculated using the `EUI64` addresses of the grandparent, parent, and the node itself, receptively. On the other hand, node ① allocates two shared cells using channel offsets 1 and 0 as its parent and grandparent is the same.

7.2.2 Hierarchical Odd-Even Minimal Cell Scheduling

A synchronization problem will occur as the *ALLOT* scheme allocates minimal cells at different locations for the different nodes. So, when a parent node transmits its control packet, its immediate child node may be busy transmitting its own control packet using a different channel offset. In this case, the child node might miss the transmitted control packet. This event can lead to inconsistency in a network. Therefore, proper scheduling of the allocated cells is necessary for maintaining synchronization among the pairs of parent-child nodes in a network. Algorithm 10 describes the procedure of the proposed scheduling scheme.

Algorithm 10 Hierarchical Odd-Even Minimal Cell Scheduling

1: **if** node is JRC and odd-even selection is not done **then**
2: Randomly decide odd-even shared cell for listen/transmit ▷ Do it once at the beginning
3: **else if** node's parent transmits in this shared slot **then**
4: Schedule Rx cell `[0,hash(EUI64(parent),`N_c`)]`
5: **else if** node has packet to transmit **then** ▷ Transmit
6: **if** packet is broadcast or unicast to child **then**
7: Schedule Tx cell `[0,hash(EUI64(own),`N_c`)]`
8: **else if** node's parent is JRC **then**
9: Schedule Tx cell `[0,hash(EUI64(JRC),`N_c`)]`
10: **else**
11: Schedule Tx cell `[0,hash(EUI64 (grandparent),`N_c`)]`
12: **end if**
13: **else** ▷ Listen as node does not have packet to transmit
14: Schedule Rx cell `[0,random no %` N_c`)]`
15: **end if**

The Absolute Slot Number (ASN), which denotes the total number of timeslots that have elapsed since the network started, is used in the proposed scheduling scheme. Note that if the ASN value of the minimal cell in i^{th} slotframe is odd, then that in $(i+1)^{th}$ slotframe it would be even, and vice versa when a slotframe length is an odd number. At the beginning, the JRC randomly decides whether it is going to transmit its control packets in the minimal cell associated with either odd or even ASN value. The other nodes decide their `Rx` and `Tx` cells based on their parent nodes' `Tx` and `Rx` strategies. A joined node listens from both its parent and child nodes during its parent node's transmission

slot i.e. at `[0,hash(EUI64(parent),`N_c`)]`. On the other hand, a joined node transmits its own control packet or unicast response/request to its child node during its parent node's listening slot i.e., at `[0,hash(EUI64(own),`N_c`)]`. At the same listening slot of its parent, unicast request/response to its parent is also transmitted but using the shared cell at `[0,hash(EUI64(JRC),`N_c`)]` or `[0,hash(EUI64(grandParent),`N_c`)]` according to node's position in the DODAG. A node uses grandparent's `EUI64` address to transmit control packets to its parent because at that time, the parent node listens using the grandparent's (i.e., parent of the parent) `EUI64` address. Note that when a node does not have a grandparent, its parent node is treated as its grandparent node for allocating and scheduling `Tx` and `Rx` minimal cells. In this way, hierarchical scheduling is performed to transmit control packets in a network from top to bottom of a DODAG to maintain synchronization.

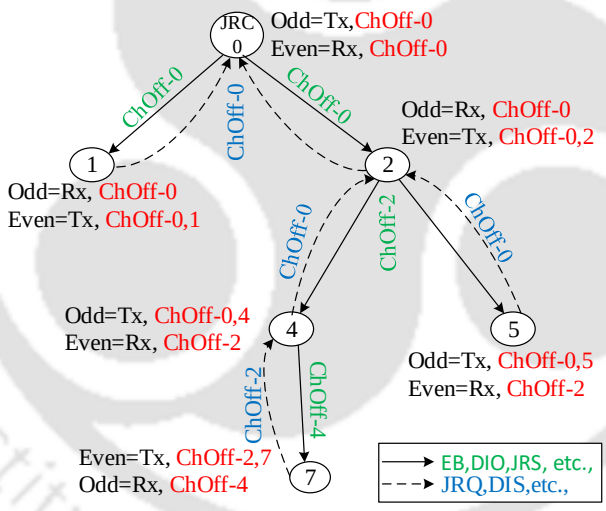


Figure 7.3: *Autonomous minimal cell scheduling by CHOICE*

This scheduling technique reduces congestion in a network as a node can transmit only when its parent is listening and vice versa. Figure 7.3 shows how the scheduling of minimal cells is done by the proposed *CHOICE* scheme, where node ⑦ transmits its request control packets (such as JRQ, DIS) to its parent node ④ using the `EUI64` address of node ②. This is because, at that time, node ④ listens for control packets using its parent `EUI64` address. Furthermore, when node ⑦ and node ② transmit (`Tx`) at the even ASN timeslot, node

④ used to listen (`Rx`) in that timeslot. Hence, there is no scheduling conflict between a pair of parent-child nodes. On the other hand, child node (say, node ④) transmits its own control packets to its own child node (i.e., node ⑦) in the tree using its own `EUI64` address. And, if parent node transmits, its child node should listen in the same transmitting shared cell using parent's `EUI64` based shared cell. For example, node ④ listens using the `EUI64` address of node ②.

7.2.3 TACTILE Framework

It is already mentioned that both the *ALLOT* and *CHOICE* schemes are running together in *TACTILE*. Figure 7.4 shows a complete framework of a new node's joining process for the construction of 6TiSCH network. A node goes through four states – Pledge, TSCH synced

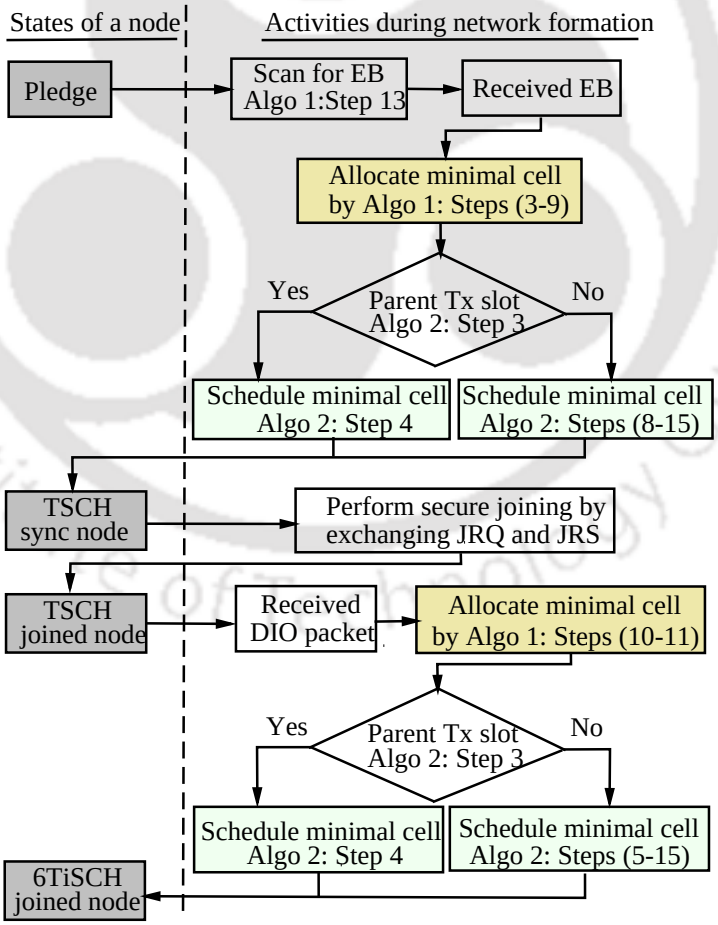


Figure 7.4: *TACTILE framework describing various states and activities of a pledge during its joining process*

node, TSCH joined node, and 6TiSCH joined node/RPL node for joining to a 6TiSCH network. Initially, a pledge waits for a valid EB on a randomly selected channel. Once it receives an EB, the node moves to TSCH synced state. At this state, the node is TSCH synchronized but has not yet completed the secured enrollment steps by exchanging JRQ and JRS frames. Once it is done, the node moves to TSCH joined node state. At this state, the node waits for RPL routing information required for multi-hop network formation. Once the node gets the RPL routing information through DIO control packet, it moves to the last state, which is RPL node or 6TiSCH joined node. At this state, the joining procedure to 6TiSCH network is completed for that node. A node allocates minimal cell for transmitting or listening control packets during bootstrapping steps. The framework in Figure 7.4 shows when the node will allocate minimal cell using the Algorithm 9 and when the cell will be scheduled using Algorithm 10. Note that if a node gets disconnected from the network at any time, the node needs to rejoin the network as a fresh pledge node. After rejoining, the previous allocation and scheduling status change based on the newly selected parent. Note that the steps 1-2 in both the algorithms are not mentioned in the framework, as they are associated with the JRC but the framework is for new node.

7.3 Theoretical Analysis of TACTILE

In this section, the *TACTILE* scheme is evaluated using Markov Chain based theoretical model. We use the same Markov Chain mode described in our Chapter 3 to estimate the *Average Joining Time (AJT)* of a pledge. Following the computation methodology as described in Chapter 3, we can calculate the *Average Slotframe Number (ASF)* required to reach the absorbing state of the Markov model can be computed as follows,

$$ASF = \frac{1}{P_{EB_S}} + \left(\frac{1}{P_{JRQ_S}} + \frac{1}{P_{JRS_S}} \right) + \frac{1}{P_{DIO_S}} \tag{7.1}$$

However, as *TACTILE* allocates more shared cells per slotframe, so the values of P_{EB_S}, P_{DIO_S}, P_{JRS_S}, and P_{JRQ_S} to compute the value of ASF are different compared to the values mentioned in our Chapter 3. These values can be caluculated as follows.

According to the *TACTILE*, every node transmits its control packet using its own Tx cell `[0,hash(EUI64(own),`N_c`)]`. Furthermore, if a node transmits its control packet in an even shared slot, it listens in an odd shared slot, and vice versa. Let us assume a node has n joined neighbor nodes and M total neighbor nodes in its surrounding. In the best case, there is at least one joined neighbor, and all the remaining $(M-1)$ neighbors are pledge. In this case, only one node transmits its control packet, so no collision occurs. However, in the worst case, all the n joined nodes transmit their control packets at the same shared slot. Hence, there is no transmission by these n joined nodes in the immediate next shared slot. So, for simplicity, we can assume that on average $\frac{n}{2}$ nodes transmit their control packets in each shared slot. Following this assumption, the probability of receiving an EB frame successfully by a pledge in a shared minimal cell can be computed as follows,

$$P_{EB_S} = \sum_{n}^{M-n} (\frac{\frac{n}{2}}{N_c}) \frac{n}{2} P_{eb}(1-P_{eb})^{\frac{n-2}{2N_c}} (1-P_{dio})^{\frac{n-2}{2N_c}} \tag{7.2}$$
$$(1-P_{jrs})^{\frac{n-2}{2N_c}} (1-P_{jrq})^{\frac{M-n}{N_c}} (1-P_{loss}) P(N=n)$$

where, the value of n and M can be different for every node of a network. Here, $(M-n)$ pledges join the network one by one, and the $P(N=n)$ denotes the probability of being total n joined neighbour node at any instant of time which equals to $\frac{1}{M}$ considering uniform probability distribution. When a RPL node transmits its EB frame, on an average the remaining $(\frac{n}{2}-1)$ RPL nodes should not transmit their control packets (EB/DIO/JRS). The probability of this condition equals to $(P_{eb}(1-P_{eb})^{\frac{n-2}{2N_c}}(1-P_{dio})^{\frac{n-2}{2N_c}}(1-P_{jrs})^{\frac{n-2}{2N_c}})$ in Equation (7.2). Additionally, the neighbor pledges also should not transmit their JRQs, which is represented by the probability $((1-P_{jrq})^{\frac{M-n}{N_c}})$. The term $\frac{n-2}{2N_c}$ denotes that, in *TACTILE*, collision occurs in worst case only when $\frac{n}{2}-1 > N_c$ in a single hop scenario. Otherwise, all the joined nodes transmit their control packets using different channel offsets. $(1-P_{loss})$ is the probability that a transmitted frame should not be lost in the channel. Finally, successful transmission probability of EB is reduced by $\frac{n/2}{N_c}$ as a pledge is not synchronized initially. Further, the EB frame can be transmitted by any of the $\frac{n}{2}$ neighbor joined nodes which result in multiplying the computed probability by $\frac{n}{2}$.

Similarly, the probability of transmitting a DIO packet successfully in a minimal cell is computed as,

$$P_{DIO_S} = \sum_{n}^{M-n} \frac{n}{2} P_{dio}(1-P_{dio})^{\frac{n-2}{2N_c}}(1-P_{eb})^{\frac{n-2}{2N_c}} \tag{7.3}$$
$$(1-P_{jrs})^{\frac{n-2}{2N_c}}(1-P_{jrq})^{\frac{M-n}{N_c}}(1-P_{loss})P(N=n)$$

Likewise, the probability of transmitting a JRQ frame and receive a JRS frames successfully in a minimal cell is also computed as,

$$P_{JRQ_S} = \sum_{n}^{M-n} (M-n)(1-P_{eb})^{\frac{n}{2N_c}}(1-P_{dio})^{\frac{n}{2N_c}} P_{jrq} \tag{7.4}$$
$$(1-P_{jrq})^{\frac{M-n+1}{N_c}}(1-P_{jrs})^{\frac{n}{2N_c}}(1-P_{loss})P(N=n)$$

$$P_{JRS_S} = \sum_{n}^{M-n} \frac{n}{2} P_{jrs}(1-P_{jrs})^{\frac{n-2}{2N_c}}(1-P_{eb})^{\frac{n-2}{2N_c}} \tag{7.5}$$
$$(1-P_{dio})^{\frac{n-2}{N_c}}(1-P_{jrq})^{\frac{M-n+1}{N_c}}(1-P_{loss})P(N=n)$$

Now, to compute the above mentioned probabilities, the transmission probabilities of EB, P_{eb} and DIO, P_{dio} in a minimal cell are calculated as follows,

$$P_{eb} = \frac{L}{I_{eb}} \tag{7.6}$$

where, L is the slotframe length and I_{eb} is the beacon interval. We compute the probability P_{dio} following the procedure described in [45] as follows,

$$P_{dio} = (1-P_{eb}) \frac{\begin{array}{c} 2^{N_D}(1-P_r)^{N_D} min(\frac{L}{2^{N_D} I_{dio}^{min}}, 1) + \\ \sum_{i=0}^{N_D-1} P_r 2^i (1-P_r)^i min(\frac{L}{2^i I_{dio}^{min}}, 1) \end{array}}{P_r + 2^{N_D}(1-P_r)^{N_D} + \sum_{j=1}^{N_D-1} P_r 2^j (1-P_r)^j} \tag{7.7}$$

where I_{dio}^{min} is the initial DIO interval, P_r is the Trickle algorithm reset probability and N_D represents the total number of trickle algorithm states. Finally, the *average joining time* (AJT) of a new node is calculated as follows,

$$AJT = PJT + ASF \times L \tag{7.8}$$

where PJT is the *average joining time of parent node* of a new node in multi-hop network.

7.3.1 Analytical Results and Discussion

We consider a random network topology with the values of few parameters as follows for theoretical evaluation of the proposed scheme: $N_C = 16$, $I_{eb} = 4$ sec, $P_{loss} = 0.2$, $I_{dio}^{min} = 8$ ms, $L = 101$ $timeslots$, $T_i = 10$ ms, $N_D = 16$ and $P_r = 0.2$. Figure 7.5 pictorially depicts the

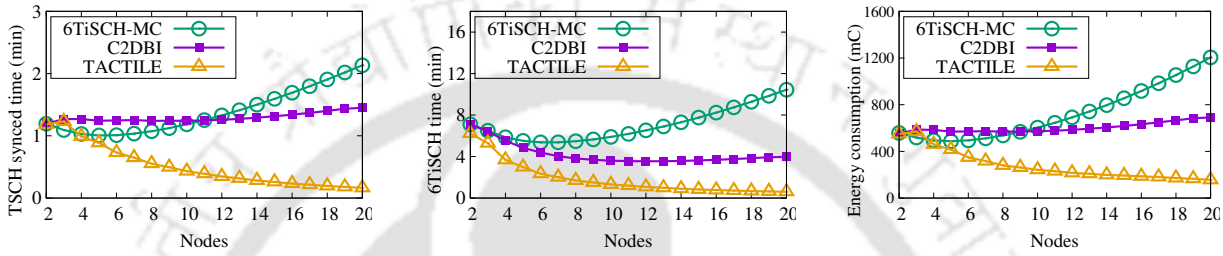


Figure 7.5: *Analytical results on (a) TSCH synchronization; (b) 6TiSCH formation time; and (c) energy consumption by a node*

theoretical results along with a comparative study with 6TiSCH-MC [40] and and our previous scheme C2DBI (ref. Chapter 3). In our Chapter 3, we provide a complete theoretical analysis of 6TiSCH-MC and C2DBI. For evaluating C2DBI, we consider minimum beacon interval $I_{eb}^{min} = 4 \times L$, and maximum beacon interval $I_{eb}^{max} = 10 \times L$. In this chapter, we have used the same analysis to draw the plots.

Figure 7.5a and 7.5b show the TSCH synchronization time and 6TiSCH joining time, respectively. Results show that the pledges' joining time increases with the increasing number of joined nodes using the 6TiSCH-MC and our previous scheme C2DBI. It is because both the schemes use only a single physical channel in the shared timeslots to transmit control packets even though IEEE 802.15.4e allows to use multiple physical channels at a time. So, the congestion in the single transmission channel increases with the increasing number of nodes. When the number of nodes increases, congestion in the transmission channel also increases. Because of this, nodes fail to transmit their control packet in minimal time, and sometimes, they drop packets due to congestion in shared cells. This ultimately increases the formation time of 6TiSCH network. Although the congestion in shared cell is reduced in

C2DBI by varying the beacon generation interval of the joined nodes, that is not sufficient compared to that in *TACTILE*.

TACTILE reduces the congestion using the distribution of minimal cell position of different nodes along the channel offset axis per slotframe. Therefore, instead of using a single channel for all the nodes, *TACTILE* uses all the available physical channels in the shared timeslots when the number of nodes increases in the network topology. Hence, even when the number of nodes increases, congestion in a particular transmission channel does not increase abruptly. So, less delay in transmission of control packets, unlike the 6TiSCH-MC and C2DBI. This helps in improving the formation time. Furthermore, as the pledges scan random channel for EB, in *TACTILE*, the transmission of more EBs by multiple joined nodes simultaneously using different physical channels increases the EB reception probabilities of the pledges.

Figure 7.5c shows the energy consumption of a node during network formation. Again, for calculating the energy consumption of the nodes, we follow our previous model designed in Chapter 3. It is observed that when the number of nodes increases, the congestion in minimal cell also increases. Because of this increasing congestion, pledges need to keep their radios active for more amount of time to get an EB frame. In other words, nodes are having 100% radio duty cycle for a longer duration, and it causes quick energy drain. Hence, 6TiSCH-MC shows worst result than C2DBI and *TACTILE*. On the other hand, leveraging all the physical channels for transmitting control packets, *TACTILE* reduces the TSCH synchronization time of the pledges, which in turn reduces their energy consumption.

7.4 Testbed Evaluation of TACTILE

7.4.1 Experimental Setup

To perform the testbed experiments, at the beginning, the proposed *TACTILE* scheme is implemented in Contiki-NG. As the grandparent `EUI64` address is also used in the scheme, therefore, a node include its parent `EUI64` address in the Information Element (IE) of its EB

Table 7.1: *Experimental settings*

Parameter	Value
Operating System	Contiki NG
Testbed	FIT IoT-LAB, Grenoble, Lille
Mote	IoT-LAB M3
Number of channels	16
Timeslot length	10 ms
Slotframe length	101 *timeslots*
RPL version	RPL Lite
RPL DIO interval	Trickle Algorithm
Experiment duration	60 min

frame. So, the child nodes can use that `EUI64` address as their grandparent `EUI64` address. Subsequently, the implementation of *TACTILE* has been flashed on the `M3` nodes in FIT IoT-LAB for performance evaluation. The fixed beacon interval I_{eb} is set to 4 *seconds* for 6TiSCH-MC, DRA [45] and *TACTILE*, whereas, for C2DBI, I_{eb}^{min} is set to 4 *seconds* and I_{eb}^{max} is set to 12 *seconds*. The *channel busy ratio* (CBR) is calculated in the interval of 8 *seconds*. Note that in the testbed experiments, the *TACTILE* is also compared with one more benchmark scheme called DRA along with the 6TiSCH-MC and C2DBI. Here, maximum 8 *shared slots* are allocated to DRA in a slotframe. In the testbed experiments, `M3` nodes are deployed using a (2×12) linear topology and a (5×5) grid topology in Grenoble and Lille locations, respectively. Other configuration settings used in the real testbed experiments are tabulated in Table 7.1.

7.4.2 Performance Metrics

As both the joining time and energy consumption of the nodes are important for efficient data transmission and longer lifetime of 6TiSCH networks, therefore, we mainly considered three metrics to evaluate our proposed scheme *TACTILE*. Before getting synchronized with the TSCH network, pledges need to scan for EB frame by keeping their radios active all the time. So, pledges consume their maximum energy during channel scanning. Hence, the time required by a pledge to be a TSCH synchronized node is very much crucial. Therefore, we consider ***TSCH synchronization time*** of a pledge as our first metric to evaluate

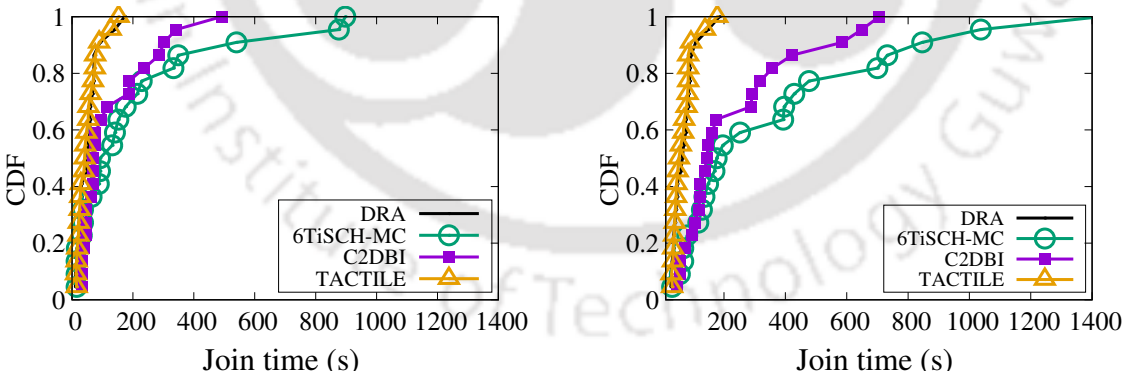
TACTILE.

A pledge becomes a complete 6TiSCH joined node when it receives both the EB and DIO control packets and also successfully exchanges the JRQ and JRS frames with the JRC. Only a 6TiSCH joined node is allowed to transmit its control packets and data packets. Hence, the time required to become a 6TiSCH joined node i.e., ***6TiSCH joining time*** should be minimum, which is considered as the second metric for evaluation.

As energy consumption of the nodes is essential for longer network lifetime, therefore, we consider the ***energy consumption*** of the nodes during the formation of 6TiSCH network as the third metric for evaluation.

7.4.3 Results and Discussion

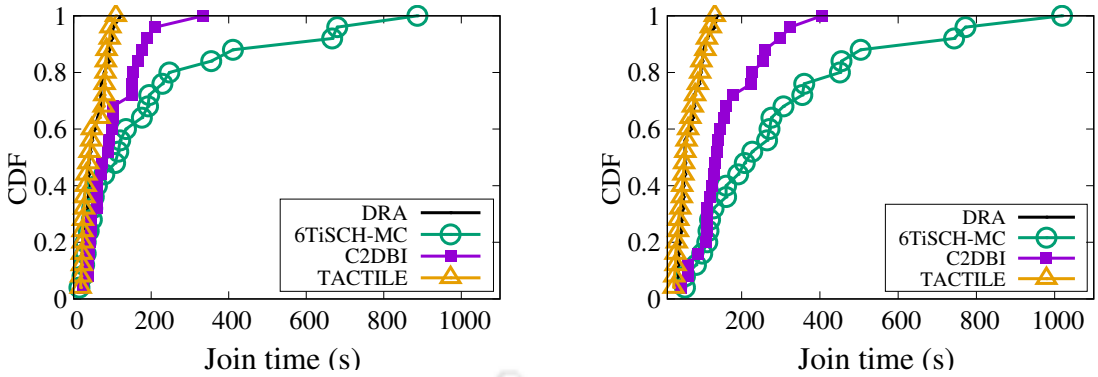
The received results are plotted in Figure 7.6 and 7.7 using CDF of TSCH and 6TiSCH formation time. Figure 7.6a and 7.7a show the time required by a pledge to receive its first EB frame corresponding to two different topologies. On the other hand, the Figure 7.6b and 7.7b shows the time required by a pledge to receives its first DIO packet after the completion of its TSCH joining process.



(a) *TSCH formation/synchronization in* 2×12 *network topology*

(b) *6TiSCH formation in* 2×12 *network topology*

Figure 7.6: *Testbed results on TSCH and 6TiSCH formation time under the* (2×12) *network topology.*

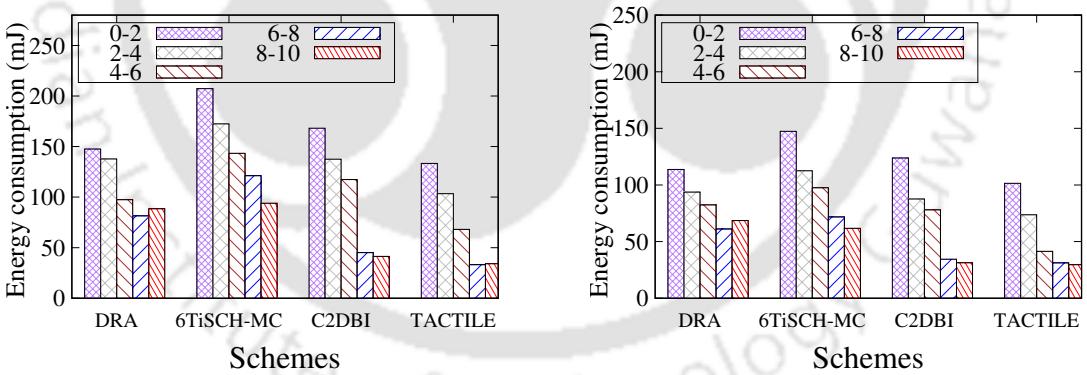


(a) *TSCH formation/synchronization in* 5×5 *network topology*

(b) *6TiSCH formation in* 5×5 *network topology*

Figure 7.7: *Testbed results on TSCH and 6TiSCH formation time under the* (5×5) *network topology.*

Received results show that the *TACTILE* performs better than 6TiSCH-MC and C2DBI, whereas it shows almost similar results with DRA in terms of both TSCH and 6TiSCH joining times. The reason for comparative performance by the DRA and *TACTILE* is the usage of more shared cells per slotframe. So, DRA and *TACTILE* mitigate the congestion in shared cells significantly, which improves the performance of 6TiSCH network.



(a) *Energy consumption in* 2×12 *network topology*

(b) *Energy consumption in* 5×5 *network topology*

Figure 7.8: *Testbed results on average energy consumption corresponding to various experimental time intervals (e.g. 0-2 min, 2-4 min, etc.,) under the* (2×12) *and* (5×5) *network topologies.*

Figure 7.8a and 7.8b show the average energy consumption of a pledge during initial 10 *minutes* of network formation. In the initial 10 *minutes* most of nodes are synchronized with TSCH, and so, the energy consumption of the network almost remains constant after this period. In DRA, all the nodes need to keep their radios active in all the allocated

shared slots as the node does not know in which shared slot its expected control packet will be transmitted. This increases the radio duty cycle of the nodes, which in turn consumes more energy consumption. On the other hand, in *TACTILE*, a receiver knows the exact location of a transmitter's cell. Hence, the nodes save their energy by not keeping their radios active in more than one cell in a slotframe. It can be seen that DRA consumes more energy, even more than C2DBI.

In brief, the obtained results show that the proposed *TACTILE* scheme achieves 87%, 67%, and 3% improvements in terms of 6TiSCH joining time, and 42%, 23%, and 29% improvements in terms of average energy consumption over 6TiSCH-MC, C2DBI, and DRA, respectively, in 5×5 grid topology. Similarly, results show significant improvements in joining time and energy consumption in 2×12 grid topology.

7.5 Time-Variant RGB Model for Minimal Cell Allocation and Scheduling (TRGB)

In *TACTILE*, we increase the number of minimal cells per slotframe in the whole network but allow a node to use only one minimal cell in a given slotframe. Hence, *TACTILE* have shown better performance compared to the existing schemes using similar low radio duty cycle as 6TiSCH-MC. However, in *TACTILE*, we do not maintain a common cell for all the nodes for exchanging routing information control packets. This common cell is necessary for maintaining stable network and constructing efficient routing tree for both upward and downward routing. Furthermore, in *TACTILE*, the nodes autonomously allocate their minimal cells only once and then remain unchanged throughout the lifetime. Therefore, the repeated collision of control packet transmission is possible when two or more nodes add the same set of minimal cells in every slotframe.

Therefore, for maintaining stable network and minimize the repeated collision, we propose Time-Variant RGB (TRGB) model as an enhanced version of *TACTILE* in this section. *TRGB* has two sub-schemes and both the sub-schemes must be executed by all the joined

and synchronized nodes. The first sub-scheme i.e., *time-variant autonomous allocation of minimal cell* (*INSTALL*) autonomously allocates minimal cells to the nodes whereas the other sub-scheme i.e., *Red-Green-Blue* (*RGB*) autonomously schedules the allocated minimal cells. In brief, *INSTALL* suffers from the synchronization issue without the *RGB* scheme. Both these sub-schemes are described separately in the below subsections.

7.5.1 Time-variant Autonomous Allocation of Minimal Cell

INSTALL usages the unique `EUI64` addresses of the nodes to allocate minimal cells at slot offset 0. For the efficient construction of the routing tree i.e., RPL's DODAG, *INSTALL* adds a common cell at the `slot offset 0` and `channel offset 0` i.e., at cell [0,0] for all the nodes for exchanging RPL routing traffic. Apart from this common minimal cell, *INSTALL* allows the nodes to add at most three more minimal cells at the same `slot offset 0` but with different `channel offsets` depending on the position of the nodes in the DODAG tree. Let the different channel offsets x, y and z of three minimal cells are defined as follows:

$$x = mod(hash(\texttt{EUI64(Own)} + ASFC), N_c - 1) + 1 \tag{7.9}$$

$$y = mod(hash(\texttt{EUI64(Parent)} + ASFC), N_c - 1) + 1 \tag{7.10}$$

$$z = mod(hash(\texttt{EUI64(GrandPa)} + ASFC), N_c - 1) + 1 \tag{7.11}$$

where `hash`[1] function is used to find a unique integer number from the `EUI64` addresses of the nodes and *ASFC*. The `EUI64(own)`, `EUI64(Parent)`, and `EUI64(GrandPa)` denote the `EUI64` addresses of a node, its parent and grandparent, respectively, and *Absolute Slotframe Count (ASFC)* is calculated as $\lfloor \frac{ASN}{L} \rfloor$. Here, *ASN* and L denote the *absolute slot number* and slotframe length, respectively. The function $\lfloor f \rfloor$ denotes the floor function and N_c denotes the number of used channels in the network. We add 1 in the equations as `channel offset 0` is reserved for common cell. As the JRC does not have parent and grandparent, so JRC adds only one minimal cell corresponds to its own address. Similarly, the nodes that do not have grandparent i.e., one hop children of JRC, add two minimal cells considering

[1]Network administrator can choose/design his/her own hash function which is implementation specific.

Algorithm 11 Time-variant Autonomous Allocation of Minimal Cell

```
1: if node is JRC then
2:     Add two minimal cells as follows:
3:     [0, 0]                                          ▷ Common cell
4:     [0, Own]                                        ▷ By Eq. (7.9)
5: else if node is only TSCH synchronized then
6:     if parent is JRC then
7:         Add two minimal cells as follows:
8:         [0,0]                                       ▷ Common cell
9:         [0, Parent]                                 ▷ By Eq. (7.10)
10:    else                                            ▷ i.e. parent is not JRC
11:        Add three minimal cells as follows:
12:        [0, 0]                                      ▷ Common cell
13:        [0, Parent]                                 ▷ By Eq. (7.10)
14:        [0, GrandPa]                                ▷ By Eq. (7.11)
15:    end if
16: else if 6TiSCH joined node then
17:    Add one more minimal cell as follows:
18:    [0, Own]                                        ▷ By Eq. (7.9)
19: else                                               ▷ It's a pledge
20:    Scan random channel for EB frame
21: end if
```

the parent i.e., JRC's `EUI64` address and their own `EUI64` addresses. Otherwise, the nodes add three minimal cells at the slot offset 0 and the channel offsets x, y and z defined by the above equations. Algorithm 11 describes the steps in *INSTALL*. Few important points about *INSTALL* are mentioned in the following.

Alleviate repeated control packet collision: To reduce repeated usage of a set of minimal cells by more than one nodes, $ASFC$ is used in the above equations. After each slotframe, the value of $AFSC$ will be increamented by one i.e., $AFSC$ will vary over time. Therefore, $AFSC$ is used in the hashing equations, so that the `hash` function returns different output after each slotframe. This prevents specific cells are being overlapped forever and alleviates repeated usage of same set of minimal cells by more than one nodes. Ultimately, this reduces the repeated control packet collision. Therefore, $AFSC$ is included in the equations instead of allocating fixed channel offsets directly. Now, as x, y and z are used as channel offset, so TSCH's channel hopping mechanism adds different transmission channels to the nodes over time. Otherwise, the nodes would always get the same physical

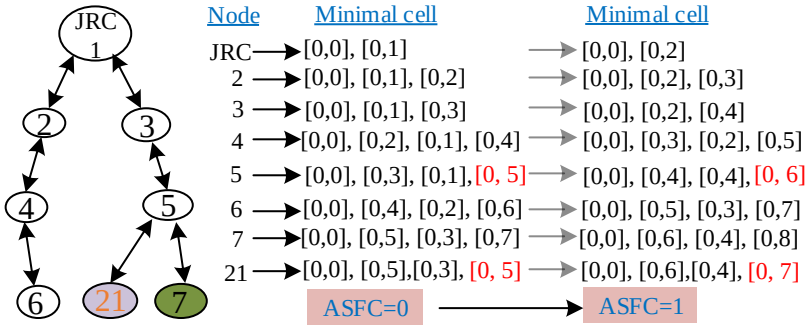


Figure 7.9: *Minimal cells allocation by INSTALL when absolute slotframe count (ASFC) equals to 0 and 1, respectively.*

channels if their allocated channel offsets are same.

Autonomous allocation of minimal cell without signaling overhead: A node can easily extract its parent's `EUI64` address from the received EB frame, whereas, `EUI64` address of the grandparent can be added as Information Element (IE) in the EB frame. Hence, nodes can autonomously add their minimal cells at slot offset 0 without any signaling overhead. Furthermore, the `EUI64` addresses of the nodes are unique, so, *INSTALL* allocates distinct minimal cells to the nodes. Note that there will be `hash` collision when the number of nodes within a single hop network exceeds the total number of physical channels used in the network. `Hash` collision results in assignment of same minimal cell to the colliding nodes. However, packet collision is only possible when more than one nodes use to transmit their packets at the same time using same minimal cell.

Leveraging unused cells: All the minimal cells of a node are added at the same slot offset, but using different channel offsets, which varies with time. So, a node can use only one minimal cell per slotframe. Hence, *INSTALL* neither affects the existing data transmission schedule and manageable cells nor increases the radio duty cycles of the nodes. However, *INSTALL* increases the number of unique minimal cells per slotframe in the whole network without any signaling overhead.

Figure 7.9 shows an example of allocation made by *INSTALL* in first two consecutive slotframes i.e., $ASFC = 0$ and $ASFC = 1$. For example, node ⑤ allocates four minimal cells based on the `EUI64` addresses of his own `[0,5]`, parent `[0,3]`, and grandparent `[0,1]` and including the common minimal cell `[0,0]` in $ASFC = 0$. However, in the next slotframe

i.e., $ASFC = 1$, the positions of the added minimal cells of node ⑤ have changed to other location except the common cell due to the used `hash` function and value of *ASFC*. All nodes change the location of autonomously allocated minimal cells after each slotframe as shown in Figure 7.9. It helps in eliminating cell overlapping by multiple nodes. For example, node ⑤ and node ㉑ use different minimal cells based on own their `EUI64` addresses in $ASFC = 0$ and $ASFC = 1$. Thus, *INSTALL* reduces repeated minimal cell overlapping, and so control packet collision

7.5.2 Red-Green-Blue Scheduling

RGB is proposed to schedule the allocated minimal cells by *INSTALL* to avoid de-synchronization for the parent-child pairs. Furthermore, a node can be in any of the three different radio states i.e., `Tx`, `Rx` and `Idle` in a given minimal cell of a slotframe. *RGB* also schedules these radio states of a node effectively. For example, when the parent node is in `Tx` state, the child needs to be scheduled in `Rx` state, and vice versa. Algorithm 12 describes the steps of *RGB*. Few important points about *RGB* scheduling are present in the following.

What these *R, G, and B* indicate? We consider a slotframe as *Red* (R) when the modulus of the *ASN* number of the starting timeslot of the slotframe with respect to 3 is 0 i.e. $(starting\ ASN)\ \%\ 3 = 0$. Similarly, we consider a slotframe *Green* (G) and *Blue* (B) when the modulus of the *ASN* number of the starting timeslot of the slotframe with respect to 3 is 1 and 2, receptively. As, we consider the modulus of 3, so these, R, G and B slotframes are repeated after 3 slotframes. Note that when the slotframe length is small, R, G, and B slotframe appear more frequently and vice versa.

Usage of *R, G and B* slotframes: All the nodes reserve R slotframe for exchanging routing information carrying packets such as multicast DIS, multicast DIO, whereas, G and B slotframe are used either for `Tx` or `Rx` of EB, unicast DIO, keep-alive, unicast DIS, and DAO (Destination Advertisement Object).

Conflict free radio states scheduling: All the nodes perform communication in these R, G, and B slotframes as follows: **In *R* slotframe** all the nodes either transmit (`Tx`) or

Algorithm 12 Red-Green-Blue Scheduling

```
JRC decides its Tx slotframe either at G or B
if a pledge receives EB in G then
    set flag = G
else
    set flag = B
end if
if current slotframe is R then
    Nodes set Tx/Rx at [0, 0]                          ▷ Common cell
else if node is JRC then
    if current slotframe is decided Tx slotframe then
        Set Tx at [0,Own]                              ▷ By Eq. (7.9)
    else
        Set Rx at [0,Own]                              ▷ By Eq. (7.9)
    end if
else if current slotframe is flag then
    Set Rx at [0,Parent]                               ▷ By Eq. (7.10)
else if current packet is for parent then
    if JRC is parent then
        Set Tx at [0,Parent]                           ▷ By Eq. (7.10)
    else
        Set Tx at [0,GrandPa]                          ▷ By Eq. (7.11)
    end if
else if packet is available then                       ▷ Broadcast packet
    Set Tx at [0,Own]                                  ▷ By Eq. (7.9)
else                                                   ▷ Own turn but no packet available
    Turn off the radio
end if
```

receive (Rx) routing information carrying packet using the common minimal cell i.e., [0, 0] depending on the availability of the routing packet in their transmission buffer. All the nodes are allowed to transmit their routing information using the common cell only so that the nodes can change different parents as the routing protocol discovers new neighbors over time via the reception of DIO packets. This process of receiving DIO packets from new neighbors helps in finding new routes towards the root node with a better link quality or a lower cost. Thus the nodes would not stick with the first parent, and so not lead to suboptimal routes and poor performance for data transmission at the steady state. Again, assigning the ***R* slotframe** particularly for RPL traffic (excluding other traffic such as EB, keep-alive, unicast RPL traffic) helps in reducing congestion in minimal cell even when the

number of nodes increases as shown by [45]. Therefore, nodes can easily discover other neighbors and consequently other routes that the first one discovered at bootstrap.

On the other hand, regarding ***G* and *B* slotframes**, the JRC randomly chooses its `Tx` slotframe either G or B at the beginning; and always uses that slotframe as uni-directional `Tx` slotframe and the other slotframe as `Rx`. The other nodes stored their EB receiving slotframe, and accordingly decide their control packets transmitting/receiving slotframe. For example, if a node receives the first EB from its parent at G, then it (until the node performs re-association or changes its parent) considers G as `Rx` slotframe and B as `Tx` slotframe, otherwise, vice-versa. In brief, when a node is in `Tx` state, both the parent and child(s) are in `Rx` state, and vice versa. Thus, nodes eliminates the conflict between the radio states of their own and parents.

Which minimal cell should be used at what time? We allow the nodes (except JRC) to **listen** using their parent `EUI64` addresses only in either G or B slotframe i.e., in the slotframe where their parent nodes transmit. As JRC does not have parent, so it listens on the minimal cell based on its own `EUI64` address in its pre-decided `Rx` slotframe. On the other hand, every node decides its control packet **transmitting** minimal cell depending on the type of the current control packet. It works as follows, if the current control packet type is broadcast, such as EB, then the node uses the minimal cell based on the channel offset calculated by own `EUI64` address. Whereas, if the control packet type is 'unicast' request packet to the parent node such as unicast DIS or keep-alive, then the node uses the minimal cell based on the channel offset calculated using the grandparent `EUI64` address. It is because, at that moment, the parent of the node is listening using the channel offset calculated using its own parent `EUI64` address (i.e., grandparent address of the node). The nodes which do not have a grandparent (i.e., their parent node is JRC), transmit their request control packets using the `EUI64` address of the JRC. Thus, all nodes can autonomously schedule their minimal cells without any conflict with their parent and child node(s).

In brief, the common minimal cell is used for exchanging routing control packet in case of both `Tx` and `Rx`; the minimal cell calculated by utilizing own `EUI64` address is used for

broadcasting own control packets i.e., only for `Tx`; the minimal cell calculated by utilizing parent `EUI64` address is used for listening control packets from parent i.e., only for `Rx`; and the minimal cell calculated by utilizing grandparent `EUI64` address is used for transmitting unicast control packet to parent i.e., only for `Tx`.

RGB makes the scheduling more energy efficient: *INSTALL* already forces the nodes to use low radio duty cycle as like 6TiSCH-MC. However, *RGB* further allows the nodes to turn off their radios when the nodes do not have any control packet to transmit in their `Tx` slotframes. It is because, at their `Tx` slotframe, the parent and child(s) of a node are in `Rx` state. So, there is no chance to receive control packets either from parent or child(s). Thus, *RGB* further reduces the radio duty cycles of the nodes, and so makes the scheduling more energy efficient.

Furthermore, the unused cells are converted into minimal cells without affecting the RDC and data communication cells, and both allocation and scheduling are done autonomously without any signaling overhead. So, the proposed *TRGB* follows the 6TiSCH-MC standard without incurring any overhead in the network.

7.6 Theoretical Analysis of TRGB

We follow the same Markov model and procedure described in Chapter 3 to calculate the *Average Joining Time (AJT)* of a pledge using the proposed *TRGB* model. Therefore, we follow the Equation (7.1) to calculate the value of *Average Slotframe Number (ASF)*. Now, the values of P_{EB_S}, P_{JRS_S}, P_{JRQ_S}, and P_{DIO_S} are calculated following the *TRGB* model as follows; The probability of successful EB transmission is computed as follows:

$$P_{EB_S} = \sum_{n}^{M-n} \frac{n}{N_c} \times n \times P_{eb}(1-P_{eb})^{\left\lfloor \frac{n}{N_c} \right\rfloor} \times (1-P_{jrs})^{\left\lfloor \frac{n}{N_c} \right\rfloor}(1-P_{loss})P(N=n) \qquad (7.12)$$

where n and M denote the number of already joined nodes (including JRC) and total number of nodes in the given multi-hop network, so $1 \leq n \leq M$. Different pledges can have different values of n in a given network. Therefore, the rest of the $(M-n)$ pledges need to join the network in presence of n joined nodes at any instant. The number of joined node at

any instant is described by $P(N = n)$ and considering the uniform probability distribution, $P(N = n)$ can be written as $\frac{1}{M}$. Now, in a worst case network condition such as all the joined nodes are connected to JRC, when a 6TiSCH joined node transmits its EB in a slotframe with the probability P_{eb}, the other joined nodes $(n-1)$ should not transmit either EB or JRS frames i.e., denoted by $(1-P_{eb})^{\lfloor \frac{n}{N_c} \rfloor}(1-P_{jrs})^{\lfloor \frac{n}{N_c} \rfloor}$. Note that here, other joined nodes means the nodes that are using the same minimal cell as the 6TiSCH joined node and it happens when the number of joined surpluses the total number of channels used, N_c. The term $\left\lfloor \frac{n}{N_c} \right\rfloor$ describes this condition. Now, before getting synchronized with the TSCH network, the pledges are not aware about the location of minimal cell in a slotframe, so the entire probability value of P_{EB_S} is divided by the N_c. However, the EBs are broadcasted by the n joined nodes using different channels, and so n is multiplied with the total probability. Note that *TRGB* allows the nodes to transmit DIO packet in R slotframe and JRQ frames are being transmitted by the pledges in parent `Rx` slotframe, so probabilities of transmitting these two control packets in a minimal cell i.e., P_{dio} and P_{jrs} are not considered in the above equation i.e., Equation (7.12). In the equation, P_{loss} denotes the packet loss probability in the wireless medium.

Similarly, the successful transmission probabilities of different control packets such as P_{DIO_S}, P_{JRQ_S}, and P_{JRS_S} are calculated as follows:

$$P_{DIO_S} = \sum_{n}^{M-n} nP_{dio}(1-P_{dio})^{n-1}(1-P_{loss})P(N=n) \tag{7.13}$$

$$P_{JRQ_S} = \sum_{n}^{M-n} (M-n)P_{jrq}(1-P_{jrq})^{\lfloor \frac{M-n}{N_c} \rfloor}(1-P_{loss})P(N=n) \tag{7.14}$$

$$P_{JRS_S} = \sum_{n}^{M-n} n \times P_{jrs}(1-P_{eb})^{\lfloor \frac{n}{N_c} \rfloor}(1-P_{jrs})^{\lfloor \frac{n}{N_c} \rfloor}(1-P_{loss})P(N=n) \tag{7.15}$$

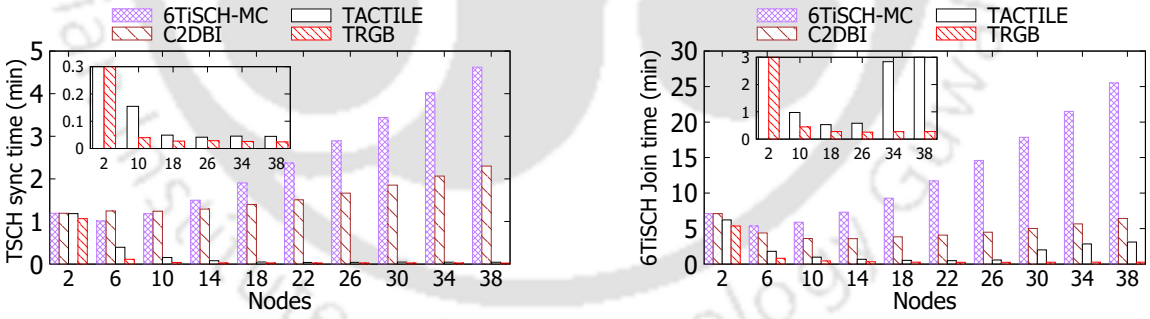
Pledges know the location of minimal cell after getting synchronized with the network, so, N_c is not used in the denominators of the Equations (7.13), (7.14), and (7.15). Again, using *TRGB* joined nodes transmit their DIO packet in R slotframe, so, the transmission of DIO packet never collides with the transmission of other control packets such as EB, JRQ, JRS.

Similarly, JRQ frame is transmitted in the parent `Rx` slotframe using the minimal cell based on parent `EUI64` address, whereas, the joined nodes transmit their EBs using their own `EUI64` addresses, so collision of these packets is not possible. To get the final value of *AJT*, the EB and DIO transmission probabilities in a minimal cell by the joined nodes (P_{eb} and P_{dio} are calculated following the Equation (7.6) and (7.7). Similarly, AJT of a pledge to join a multi-hop 6TiSCH network is calculated using the Equation (7.8).

7.6.1 Analytical Results and Discussion

The values of few parameters to evaluate the *TRGB* are taken as follows: $I_{eb} = 4sec$, $N_C = 16$, $I_{dio}^{min} = 8ms$, $P_{loss} = 0.2$, $T_i = 10ms$, $L = 101timeslots$, $P_r = 0.2$, and $N_D = 16$. We use the same metrics as mentioned in Section 7.4.2 i.e., TSCH Joining time, 6TiSCH joining time and energy consumption of the nodes to evaluate our *TRGB* model.

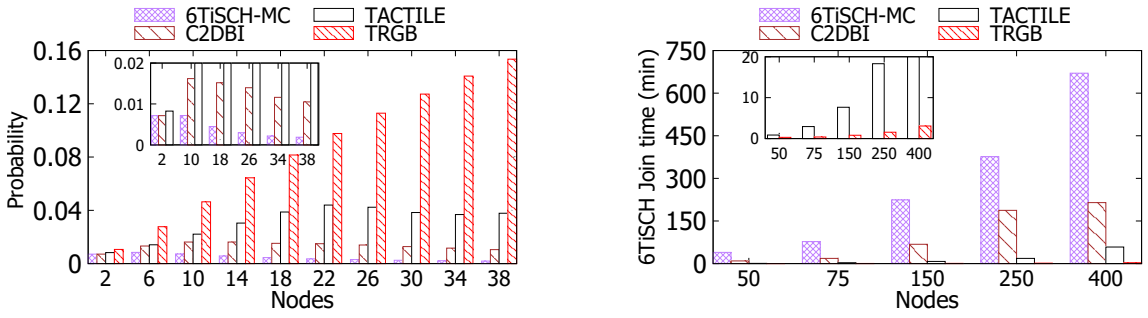
Figure 7.10, 7.11, and 7.12 show the comparison based theoretical results on the above mentioned three metrics with the state-of-the-art scheme such as 6TiSCH-MC [40], and our proposed scheme C2DBI (ref. Chapter 3) and *TACTILE*.



(a) *TSCH Joining time* **(b)** *6TiSCH joining time*

Figure 7.10: *Analytical results on TSCH and 6TiSCH joining time*

Figure 7.10a and 7.10b show the TSCH synchronization time and 6TiSCH joining time of the pledges in different size networks, respectively. Both the joining times increase with the increasing number of nodes in the networks using the 6TiSCH-MC and C2DBI schemes. It is because both schemes transmit control packets using just one minimum cell per slotframe. As a result, as the number of nodes increases, so does the minimum cell congestion, hence



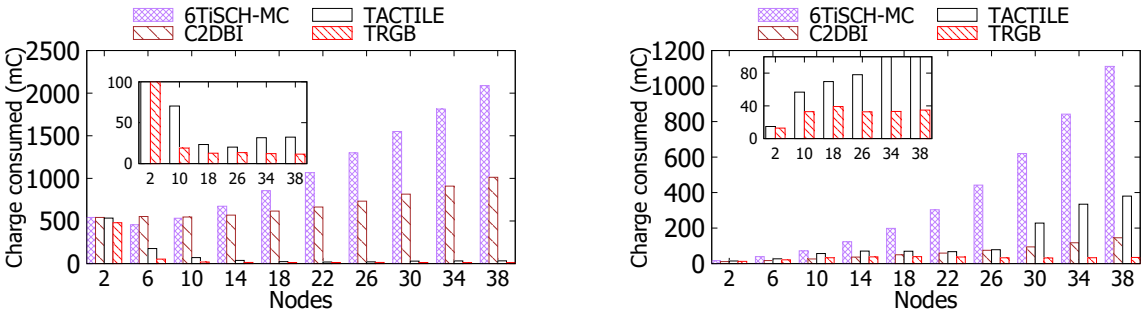
(a) *DIO reception probability* **(b)** *Join time in large networks*

Figure 7.11: *Analytical results on DIO reception probabilities, and 6TiSCH joining time in large network*

the joining times. Although C2DBI reduces the congestion by dynamically varying the beacon generation interval, this modification is not sufficient for quick network formation. However, *TACTILE* reduces the congestion in minimal cell fully by allocating more minimal cells per slotframe using the unused cells. Hence, *TACTILE* improved both the joining times of the pledges significantly. However, using *TRGB*, the joining times are further improved compared to *TACTILE*. It is because *TRGB* adds more minimal cells per slotframe and also separates the transmission of DIO packet from the other control packets. Nevertheless, using *TACTILE* and other schemes, DIO packet gets less opportunity to get transmitted within a node because of the highest priority of EB frame. Therefore, the pledges need to wait for more amount time to get the DIO packet compared to *TRGB*.

To validate this claim, the values of successful DIO transmission probabilities P_{DIO_S} is shown in Figure 7.11a, where *TRGB* dominates all the other schemes. Furthermore, as the pledges do scan on random channels for EB frame, both the *TACTILE* and *TRGB* increase the EB reception probability of the pledges by transmitting more EB frames using different physical channels at the same time and it results in quick synchronization of the pledges with the TSCH network.

As the charge consumption of the pledges during their network joining process depends on the channel scanning time, therefore, using the *TRGB*, pledges consume less charge compared to the other schemes as shown in Figure 7.12a. It is because, using *TRGB*, pledges do not need to active their radios continuously for more amount of time. Hence, it shows



(a) *Pledge energy usage* **(b)** *Joined energy consumption*

Figure 7.12: *Analytical results on charge consumption*

better results compared to the other schemes. Similarly, using *TRGB*, joined nodes do not need to transmit control packets (specially, DIO packet) more frequently as the pledges join the network quickly and start transmitting their control packets immediately. This improves the charge consumption of the joined nodes as shown in Figure 7.12b. Finally, Figure 7.11b shows the performance of *TRGB* in large size networks. 6TiSCH joining time significantly increases with the increasing size of networks using 6TiSCH-MC because of the congestion in minimal cell. Although C2DBI reduces the congestion, *TACTILE* significantly reduces the congestion by employing more minimal cells per slotframe. However, *TRGB* outperforms all the schemes by using more number of minimal cells per slotframe and separating the transmission of DIO packet from the transmission of other control packets.

7.7 Testbed Evaluation of TRGB

7.7.1 Experimental Setup

For the testbed experimental evaluation of *TRGB*, both the sub-schemes *INSTALL* and *RGB* have been implemented on Contiki-NG OS. The grandparents' `EUI64` addresses are added in the IE of the EB frames and 32-bit integer mix function[1] is used for hashing. The binary file generated on Contiki-NG has been flashed in a static topology consisting of 32-bit ARM Cortex M3 micro-controller based 60 `M3` nodes on the FIT IoT-LAB in Strasbourg, France. The nodes are deployed in a 3D space as shown in Figure 7.13. Nodes used all the

[1]https://gist.github.com/badboy/6267743

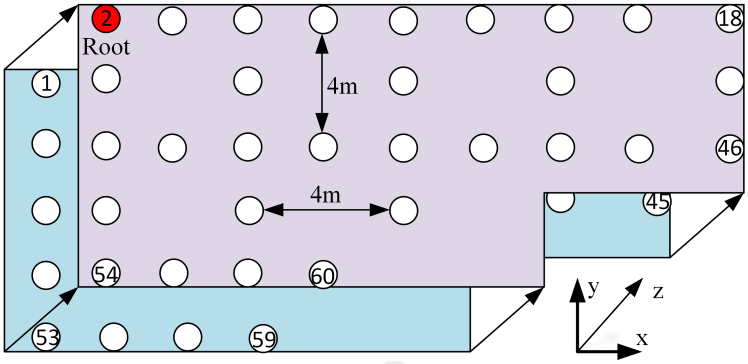


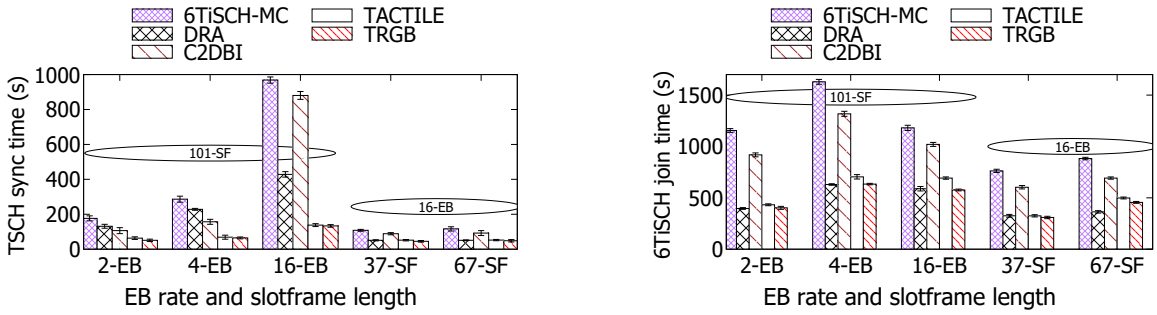
Figure 7.13: *Deployed 60 M3 nodes in Strasbourg FIT IoT-LAB*

16 communication channels with $-17dBm$ packet transmission (`Tx`) power. The timeslot duration is taken as *10 ms* and RPL-Lite (storing mode) is used as network layer protocol.

7.7.2 Results and Discussion

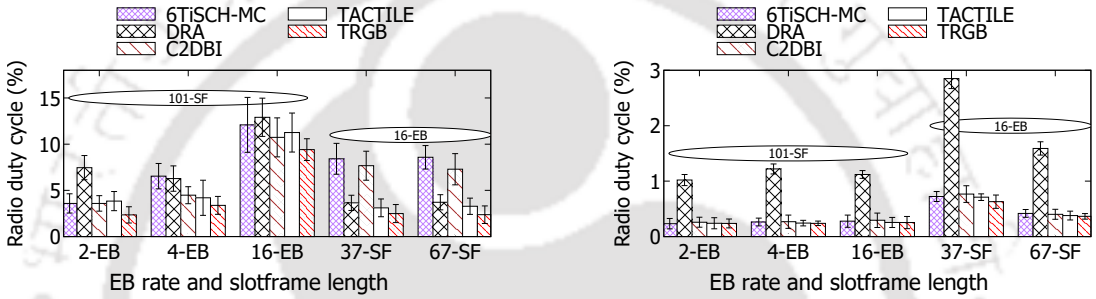
The experimental testbed results are plotted with 95% confidence interval error bars by running each experiment 4-10 times for 60 minutes. The metrics described in Section 7.3 are used again to show the comparison based testbed experimental results. Figure 7.14a, 7.14b, and 7.16a show the TSCH synchronization time, 6TiSCH joining time, and charge consumption of the nodes, respectively. Apart from these metrics, the ***radio duty cycle*** (RDC) of the pledges and joined nodes, and ***number of parent switches*** also shown in Figure 7.15a, 7.15b, 7.16b, receptively. The testbed results are obtained in different network conditions such as varied EB generation rate of the joined nodes and slotframe length, specifically, with varied congestion in minimal cell for better understanding of the schemes. Note that maximum 4 *minimal cells* per slotframe is taken for DRA and *8 seconds* is taken to measure the channel busy ration in C2DBI. *TRGB* is compared with the existing benchmark schemes such as 6TiSCH-MC [40], DRA [45], C2DBI (ref. Chapter 3), and *TACTILE*.

The testbed experimental results show that *TRGB* outperforms all the schemes (except DRA, in few cases) in terms of TSCH synchronization time and 6TiSCH joining time (Figure 7.14a, 7.14b). The reason for this performance improvement is the usage of more number of minimal cells per slotframe which decreases the congestion in minimal cell during net-



(a) *TSCH synchronization time* **(b)** *6TiSCH joining time*

Figure 7.14: *Testbed results on joining time obtained by varying the EB generation rate (represented by x-EB where EB is generated after x sec) and slotframe length (SF) (represented by y-SF, where SF has y timeslots).*

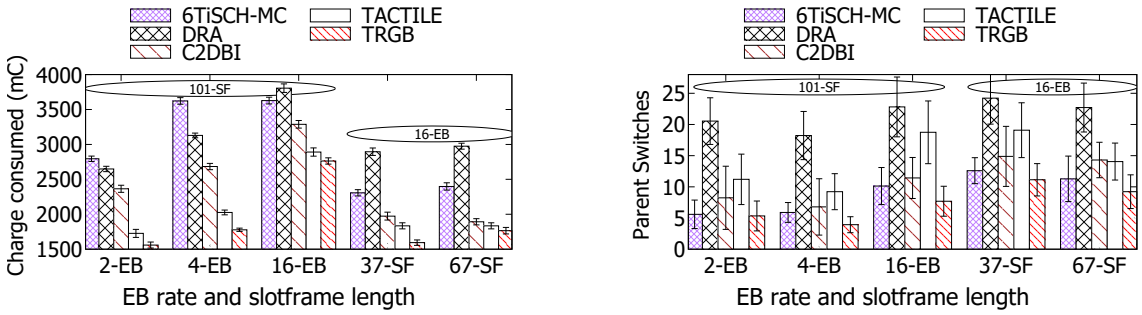


(a) *RDC of the pledges* **(b)** *RDC of the joined nodes*

Figure 7.15: *Different testbed results obtained by varying the EB generation rate (represented by x-EB where EB is generated after x sec) and slotframe length (SF) (represented by y-SF, where SF has y timeslots).*

work formation. Although *TACTILE* increases the number of minimal cells per slotframe, it allows the nodes to transmit their DIO packet in the same minimal cell where they transmit their other control packets such as EB, unicast/multicast DIS, keep-alive, and DAO. This results in increasing waiting time of the pledges to receive their DIO packets. On the other hand, *TRGB* separates the transmission of DIO packet from the transmission of other control packets, which results in faster joining of the pledges in both TSCH and 6TiSCH networks.

Although DRA shows improvement over *TRGB* in few cases in terms of joining time, it consumes highest energy compared to all other schemes as the nodes need to keep their radios active in all the dynamically allocated minimal cells. Therefore, the RDCs of both the joined nodes and pledges increase, and so, their charge consumption (Figure 7.15a, 7.15b). Further, DRA increases the unused number of cells, significantly affecting the manageable



(a) *Charge consumption* **(b)** *Parent switching count*

Figure 7.16: *Energy consumption and Stability results obtained by varying the EB generation rate (represented by x-EB where EB is generated after x sec) and slotframe length (SF) (represented by y-SF, where SF has y timeslots).*

cells, data transmission schedule, and the throughput and end-to-end latency of heavy traffic networks. Note that results on throughput and end-to-end latency are not provided as *TRGB* is designed for only control packet transmission, which can be integrated with any data transmission schedule. *TRGB* shows significant performance improvement compared to all the benchmark schemes in terms of charge consumption. It is because, using *TRGB*, joined nodes listen/transmit only in the minimal cell of a slotframe, and pledges quickly receive both the EB and DIO because of the less congestion in the minimal cells.

Furthermore, *TRGB* constructs stable network compared to the benchmark schemes. RPL tries to change the preferred parent when it does not get responses from the parent quickly. RPL assumes responses are not received because of link failure and so tries to change the parent, which creates unstable network. So, to get recovered from it, more control packets are transmitted by RPL, which again increases the congestion in the minimal cell, and also the energy consumption of the nodes. Using 6TiSCH-MC, nodes change their parent more frequently as they do not get response from their parents because of the heavy congestion in minimal cell (Figure 7.16b). On the other hand, one possible reason for more parent switching in DRA is EB's highest priority. Although nodes have sufficient resource to transmit their control packets, the other control packets need to wait longer time to get transmitted due to EB's highest priority, which increases the waiting time of the nodes which has transmitted request control packets, and so more parent switching. However, By providing more number of minimal cells per slotframe and separating the transmission of

DIO packet from other control packets, *TRGB* is able to construct stable networks.

7.8 Summary

We proposed two techniques to deal with the under-utilization of the communication channels while transmitting the control packets in this chapter. For this, at the beginning, we proposed *autonomous allocation and scheduling of minimal cell* (*TACTILE*) scheme. *TACTILE* autonomously allocates and schedules the shared cells of the nodes without any signaling overhead. For allocating the minimal cell autonomously, we proposed *autonomous minimal cell allocation* (*ALLOT*), and for scheduling the allocated minimal cell, we proposed *hierarchical odd-even minimal cell scheduling* (CHOICE) schemes. *ALLOT* distributes the shared/minimal cells among all the available channels, and CHOICE schedules the allocated minimal cell without any conflict among the nodes. Along with the Markov Chain based theoretical analysis, testbed experiments were also performed to evaluate the proposed scheme and compare it with the existing benchmark schemes. The received results showed that *TACTILE* can achieve 87% and 42% improvements in terms of joining time and average energy consumption, respectively, compared to 6TiSCH-MC.

However, later, we observed that *TACTILE* suffers from network stability issues as it does not maintain a common cell for all the nodes for exchanging routing information. The common cell is necessary for maintaining stable networks and constructing efficient routing tree for both upward and downward routing. Furthermore, the allocation made by *TACTILE* (specifically, *ALLOT*) is static which remains unchanged throughout the lifetime. So, repeated collision of control packet transmission is possible when two or more nodes add the same minimal cell. To address these problems and improve the performance of 6TiSCH network formation further, Time-Variant RGB (*TRGB*) model had been proposed. Like, *TACTILE*, *TRGB* has *time-variant autonomous allocation of minimal cell* (*INSTALL*) scheme for minimal cell allocation and *Red-Green-Black* (*RGB*) for scheduling the allocated minimal cell. *TRGB* efficiently leveraged the unused cells of 6TiSCH-MC to use them as minimal cells and scheduled them effectively for both collision free packet transmission

and conflict-free radio state management of the nodes. *TRGB* also allows the nodes to change their allocated minimal cells after each slotframe to alleviate repeated control packet collision, which occurs due to overlapping minimal cells. Apart from these, *TRGB* maintains a common cell for all the nodes to transmit only DIO packet so that it can tightly work with RPL routing protocol. Both the Markov Chain based theoretical analysis and testbed experiment on FIT IoT-LAB showed that *TRGB* significantly improve the performance of 6TiSCH network formation compared to all the existing schemes, including *TACTILE*. *TRGB* neither affects manageable cells, so the data transmission schedule, nor increases the radio duty cycles of the nodes, but maintains stable networks. In brief, compared to the benchmark schemes – 6TiSCH-MC, DRA, C2DBI, and *TACTILE*, using the EB rate equals to 16-EB and slotframe length equals to 101-SF, *TRGB* achieved 51%, 2%, 43%, and 16% improvements in 6TiSCH formation time and 23%, 27%, 15%, and 4% improvements in charge consumption of the nodes, respectively.

"You got a dream. You gotta protect it. People can't do somethin' themselves, they wanna tell you you can't do it. If you want somethin', go get it. Period."

~The Pursuit Of Happyness (An English movie)

8

Conclusions and Future Perspectives

This research work is motivated towards improving the performance of 6TiSCH network formation for efficient data transmission and increasing network lifetime. The channel hopping feature of IEEE 802.15.4 TSCH MAC, limited bandwidth offered by 6TiSCH Minimal Configuration (6TiSCH-MC) standard, and under-utilization of all the available physical channels at a time significantly affect on 6TiSCH network formation. Therefore, to improve the performance of 6TiSCH network formation, we worked in the following directions: (i) reduce the congestion in shared minimal cell, (ii) provide fair transmission opportunities to the nodes (within a network) as well as to the control packets (within a node), and (iii) increase the number of shared cells per slotframe without affecting the data transmission cells and radio duty cycle of the nodes. Towards the congestion alleviation in shared cell, we

provided two techniques which are dynamic beacon generation interval (C2DBI) and non-cooperative gaming based optimal control packet transmission (GTCC). We provided two techniques towards our second research direction which are opportunistic and adaptive control packet transmission (OTCP and ACB). Whereas, towards our last research direction, we proposed two more techniques for utilizing all the available channels at a time, and thus to increase the number of shared cells per slotframe (AAS: TACTILE and TRGB). All the works were extensively validated theoretically, and experimentally and we observed that all validation methods assert the same observations. For example, when the number of joined nodes increases in the network, the joining time of the pledges also increases and so their energy consumption. This chapter sums up all the proposed solutions of this dissertation along with the future direction of research on 6TiSCH network formation.

8.1 Summary of Contributions

The minimal resource allocation (i.e., only one shared cell per slotfram) policy of 6TiSCH-MC standard creates congestion in the shared cell when the number of nodes increases in the network. One of the major reasons for this increasing congestion is fixed EB generation interval of the joined nodes. Our proposed Markov Chain based analytical model of 6TiSCH-MC standard shows the effect of fixed beacon generation interval on the performance degradation of 6TiSCH network in terms of nodes' joining time and energy consumption. Hence, to reduce the congestion in shared cell, we proposed to vary the beacon generation interval. The nodes measure the congestion in shared cell for a particular interval, and depending on the current congestion status, they decide their next EB generation intervals instead of using fixed beacon generation interval all the time. In brief, nodes increase their EB generation interval if the congestion is high and vice versa. The effectiveness of the proposed scheme was verified using theoretical analysis, simulation, and testbed experiments.

Our further analysis of 6TiSCH-MC standard during the formation of multi-hop 6TiSCH network revealed that 6TiSCH-MC affects the formation of 6TiSCH network because (i) it considers EB frame has the highest priority over all other control packets, (ii) it does

not provide sufficient and quick transmission of DIO packet during network formation, and (iii) it does not provide any provision to access the shared cell quickly. To address these problems, we proposed two schemes. These schemes assign highest priority to the control packets depending on their requirement in the network, provide sufficient and quick transmission of DIO packet, and allow the nodes to access the shared cell quickly if they have an urgent packet to transmit. We evaluated our proposed schemes with Markov Chain based theoretical analysis, simulation, and testbed experiments.

Further, we noticed that routing information carrying DIO packet has a significant impact on the formation of 6TiSCH network. To confirm it, we designed another Markov Chain based analytical model for analyzing the effect of Trickle Algorithm (TA) on 6TiSCH network formation. Our analysis revealed that improper selection of the values of various Trickle parameters and the default working principle of TA degrade the performance of 6TiSCH network formation. Sometimes, TA creates burst transmission of DIO packet, which severely congests the shared cell. On the other hand, in some cases, TA does not allow the nodes to transmit their DIO packet fairly. To address these issues of TA, we proposed a scheme which enables efficient and fair transmission of DIO packets in 6TiSCH network. To alleviate the congestion further, we proposed another scheme. This scheme also provides fair control packets transmission opportunities to the nodes without any signaling overhead in the network. We performed testbed experiments to validate both the proposed schemes together.

Nevertheless, we also designed a non-cooperative gaming model to find optimal control packet transmission probabilities of the nodes. We considered the joined nodes as *players* in our proposed gaming model where a player attempts to maximize its control packet transmission probability (i.e., *profit*) considering the congestion in minimal cell and nodes' remaining energy. Therefore, in the *pay-off function* of the proposed game, we considered the control packet transmission probability as a *utility function*, and the *minimal cell busy ratio* and *energy consumption* of a node w.r.t. its remaining energy as *price functions*. This pay-off function maintains a balance among the control packet transmission probability,

increasing congestion in the minimal cell, and nodes' energy consumption. The obtained solution of the proposed game was used in our proposed game theory based congestion control scheme, which improves the performance of 6TiSCH network formation. Along with the theoretical analysis, we also performed testbed experiments to validate our proposed scheme.

Finally, we proposed a scheme to significantly improve the performance of 6TiSCH network during its formation. This scheme increases the number of shared cells per slotframe without affecting the data transmission cell and increasing the radio duty cycle of the nodes. The existing schemes did not use all the available physical channels at the shared timeslot to transmit control packet. Instead, they used only one channel. To deal with this under-utilization of channels, the proposed scheme autonomously allocates and schedules the minimal cells in such a way which enables the utilization of all the unused channels at the shared timeslot, resulting in increasing number of shared cells per slotframe. To improve the performance of the proposed scheme further, we proposed another improved scheme. This improved scheme changes the position of autonomously allocated minimal cells after each slotframe to avoid repeated control packet collision, which is not done in our previous scheme. The improved scheme also provides close interaction with RPL by maintaining a common cell for all the nodes for exchanging routing information. Both the proposed schemes significantly improve the formation of 6TiSCH network in terms of joining time and energy consumption. The improved scheme also able to maintain stable network compared to all the existing schemes. We evaluated both the proposed schemes with Markov Chain based theoretical analysis and extensive testbed experiments.

8.2 Scope for Future Work

While we have made significant improvement in the formation of 6TiSCH network by working on several research directions, some of the future research directions still can be considered towards the faster and efficient formation of 6TiSCH network. We describe these research scopes as follows:

8.2.1 Impact of Clock Drift on Nodes' Re-association

As the nodes use timeslot to communicate with their neighboring nodes for both control and data packet transmission in IEEE 802.15.4e TSCH based networks, therefore, both the transmitter and receiver's clocks should be aligned to each other [95, 96]. In brief, both the transmitter and receiver clocks should be synchronized within $1ms$. However, due to various environmental factors such as fluctuation in temperature as well as supplied voltage, the difference in mote fabrication by different vendors, both the transmitter-receiver clocks used to drift from each other. Although this drifting happens in terms of microseconds each time, both the transmitter and receiver gets de-synchronized when they are drifted by more than $1ms$. And because of this de-synchronization, the child joined node(s) might need to rejoin the network as a fresh pledge. Although various schemes have already been proposed to deal with this clock drifting issue, researchers still can consider clock drifting in various environmental conditions and propose solution for it. Furthermore, only few research works had been published, which studied the rate of drifting in the presence of multiple time-source.Therefore, clock drifting is still an open and important research problem because the timeslot shared by a parent-child (transmitter-receiver) nodes needs to be closely aligned to receive the transmitted data correctly in 6TiSCH network. As de-synchronization forces an already joined node to join the network as a pledge node again, so it may affect the entire formation of multi-hop 6TiSCH networks if the rejoining of the node does not happen in minimal time. Therefore, clock drifting is a vital research issue that can be explored in more detail from a network formation perspective.

Note that during the formation of 6TiSCH network, data transmission is not started yet. Therefore, ACK-based synchronization can not be used for clock drift correction. Only Frame-based synchronization mechanism is used by the nodes to align their clocks with their parents' clocks after getting EB frame. So, de-synchronization among the nodes due to clock drifting can be expected when there is a high congestion in shared cell. Hence, de-synchronization due to clock drift can be considered as an open research problem during the formation of 6TiSCH network.

8.2.2 Formation of 6TiSCH Networks in presence of Malicious Node

The work in [97] studied the 6TiSCH network formation process in the presence of malicious nodes. It is the first work that considered the formation of 6TiSCH network under misbehaved nodes. The authors mentioned from their simulation experiments that the formation time of the entire network drastically increases in presence of non-cooperative malicious nodes. However, the authors did their experiments considering only the 6P transaction [42], which is used for scheduling the data packet transmission cell in a distributed manner. However, the formation of 6TiSCH network can be studied considering other security threats which a malicious node can cause, such as jamming the common shared cell by cracking the channel hopping sequence, DIS attack or DIS Sybil attack [98], DIO suppression attack, DIO black hole attack, RPL rank attack, etc. As both the congestion in shared cell and correct/valid transmission of all kinds of routing information carrying control packets play a critical role in forming the of multi-hop 6TiSCH networks, therefore, it is very much necessary to investigate the performance of 6TiSCH network formation in the presence of malicious node. Furthermore, the other security threads to IoT as mentioned in [99] need to be investigated during the formation of 6TiSCH networks.

8.2.3 Formation of 6TiSCH Networks in presence of Mobile Node

The IEEE 802.15.4e TSCH based 6TiSCH network is mainly designed for industrial application. Although some industrial applications do not have mobile nodes, some industries, such as the petroleum and oil industry, actuation-based industrial monitoring, and waste management, mobile nodes are used. It is noteworthy that none of the published works considered the formation of 6TiSCH network in the presents of mobile nodes. It is essential because when a joined node moves out of the communication range of the child node(s), all the child node(s) need to again rejoin the network as a pledge. This can severely impact on the overall formation of the network. Therefore, it is another research scope where

researchers can think about the formation of 6TiSCH network in the presence of mobile node.

8.2.4 Formation of SDN based 6TiSCH Network

Nowadays, researchers are trying to incorporate Software Defined Network (SDN) concept in LLNs. However, only a few works had been published, integrating SDN in 6TiSCH networks to schedule the communication cell [100, 101]. Implementing a centralized SDN architecture in LLNs will face considerable challenges such as control traffic generated by SDN subject to jitters due to unreliable wireless links and contention in the communication cell, which can further hinder the performance of SDN-based network. Furthermore, the SDN control traffic also needs to be transmitted in the communication cell. Therefore, how the control traffic overhead generated by SDN can be transmitted along with the existing 6TiSCH control traffic during the formation of SDN-based 6TiSCH networks needs to be adequately investigated. The resource allocation by 6TiSCH-MC is not sufficient for 6TiSCH control traffic as per the existing works. In this case, the transmission of SDN control traffic along with the existing 6TiSCH network traffic will be some extra burden on the shared cell. Hence, this will surely degrade the performance of the networks during their formations. Therefore, it is crucial to investigate how to transmit SDN control traffic along with the 6TiSCH control traffic during or after the formation of 6TiSCH network.

References

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

[2] L. D. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[3] J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[4] "Internet of Things in 2020, Roadmap for the Future, Version 1.1," *INFSO D.4 Networked Enterprise, RFID INFSO G.2 Micro and Nanosystems*, p. 1, May 2008.

[5] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 44–51, 2010.

[6] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the Internet of Things: A survey," in *Proc. of International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2011, pp. 1–6.

[7] T. Watteyne, V. Handziski, X. Vilajosana, S. Duquennoy, O. Hahm, E. Baccelli, and A. Wolisz, "Industrial Wireless IP-Based Cyber –Physical Systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1025–1038, 2016.

[8] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green Industrial Internet of Things Architecture: An Energy-Efficient Perspective," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 48–54, 2016.

[9] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities," *IEEE Access*, vol. 5, pp. 26 521–26 544, 2017.

[10] A. B. Noel, A. Abdaoui, T. Elfouly, M. H. Ahmed, A. Badawy, and M. S. Shehata, "Structural Health Monitoring Using Wireless Sensor Networks: A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1403–1423, 2017.

[11] M. Alaa, A. Zaidan, B. Zaidan, M. Talal, and M. Kiah, "A review of smart home applications based on Internet of Things," *Journal of Network and Computer Applications*, vol. 97, pp. 48–65, 2017.

[12] E. Park, Y. Cho, J. Han, and S. J. Kwon, "Comprehensive Approaches to User Acceptance of Internet of Things in a Smart Home Environment," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2342–2350, 2017.

[13] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The Internet of things is the backbone," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, 2016.

[14] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[15] Y. Li, X. Cheng, Y. Cao, D. Wang, and L. Yang, "Smart Choice for the Smart Grid: Narrowband Internet of Things (NB-IoT)," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1505–1515, 2018.

[16] M. Yun and B. Yuxin, "Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid," in *Proc. of International Conference on Advances in Energy Engineering*, 2010, pp. 69–72.

[17] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 349–359, 2014.

[18] N. Ahmed, D. De, and I. Hussain, "Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4890–4899, 2018.

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

[20] M. Sha, D. Gunatilaka, C. Wu, and C. Lu, "Empirical Study and Enhancements of Industrial Wireless Sensor–Actuator Network Protocols," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 696–704, 2017.

[21] I. Stojmenovic, "Machine-to-Machine Communications With In-Network Data Aggregation, Processing, and Actuation for Large-Scale Cyber-Physical Systems," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 122–128, 2014.

[22] 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 Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.

[23] A. Aijaz and A. H. Aghvami, "Cognitive Machine-to-Machine Communications for Internet-of-Things: A Protocol Stack Perspective," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, 2015.

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

[25] S. Safaric and K. Malaric, "ZigBee wireless standard," in *Proc. of International Symposium on Electronics in Marine (ELMAR)*, 2006, pp. 259–262.

[26] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–320, 2006.

[27] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks (WPAN)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 1–700, 2005.

[28] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control," in *Proc. of IEEE Real-Time and Embedded Technology and Applications Symposium*, 2008, pp. 377–386.

[29] "Wireless Systems for Industry Automation: Process Control and Related Applications," *ISA-100.11a-2011 Standard*, pp. 1–793, 2011.

[30] A. D. Zayas and P. Merino, "The 3GPP NB-IoT system architecture for the Internet of Things," in *Proc. of IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017, pp. 277–282.

[31] "IEEE Standard for Information technology–Telecommunications and information exchange between systems - Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer

(PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation," *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)*, pp. 1–594, 2017.

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

[33] H. Kurunathan, R. Severino, A. Koubaa, and E. Tovar, "IEEE 802.15.4e in a Nutshell: Survey and Performance Evaluation," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 1989–2010, 2018.

[34] S. Pollin, M. Ergen, S. C. Ergen, B. Bougard, L. V. Der Perre, I. Moerman, A. Bahai, P. Varaiya, and F. Catthoor, "Performance Analysis of Slotted Carrier Sense IEEE 802.15.4 Medium Access Layer," *IEEE Transactions on Wireless Communications*, vol. 7, no. 9, pp. 3359–3371, 2008.

[35] "IEEE Standard for Local and metropolitan area networks-Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1–225, April 2012.

[36] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1–709, April 2016.

[37] T. Watteyne, M. R. Palattella, and L. A. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement," RFC 7554, Internet Engineering Task Force, Tech. Rep. 7554, May 2015.

[38] X. Vilajosana, T. Watteyne, T. Chang, M. Vučinić, S. Duquennoy, and P. Thubert, "IETF 6TiSCH: A Tutorial," *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 595–615, 2020.

[39] M. Vučinić, J. Simon, K. Pister, and M. Richardson, "Constrained Join Protocol (CoJP) for 6TiSCH," RFC 9031, Internet Engineering Task Force, Tech. Rep. 9031, May 2021.

[40] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration," Internet Engineering Task Force, RFC 8180, May 2017.

[41] N. Accettura, E. Vogli, M. R. Palattella, L. A. Grieco, G. Boggia, and M. Dohler, "Decentralized Traffic Aware Scheduling in 6TiSCH Networks: Design and Experimental Evaluation," *IEEE Internet of Things Journal*, vol. 2, no. 6, pp. 455–470, 2015.

[42] Q. Wang, X. Vilajosana, and T. Watteyne, "6TiSCH Operation Sublayer (6top) Protocol (6P)," Internet Engineering Task Force, RFC 8480, November 2018.

[43] T. Winter *et al.*, "RPL: IP 6 Routing Protocol for Low-Power and Lossy Networks," Internet Engineering Task Force, RFC 6550, March 2012.

[44] M. Richardson, "6TiSCH Zero-Touch Secure Join protocol," Internet Engineering Task Force, draft-ietf-6tisch-dtsecurity-zerotouch-join-04 [work-in-progress], July 2019.

[45] C. Vallati, S. Brienza, G. Anastasi, and S. K. Das, "Improving Network Formation in 6TiSCH Networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 98–110, 2019.

[46] M. Vucinic, T. Watteyne, and X. Vilajosana, "Broadcasting Strategies in 6TiSCH Networks," *Internet Technology Letters*, December 2017. [Online]. Available: https://doi.org/10.1002/itl2.15

[47] D. D. Guglielmo, A. Seghetti, G. Anastasi, and M. Conti, "A performance analysis of the network formation process in IEEE 802.15.4e TSCH wireless sensor/actuator networks," in *Proc. of IEEE Symposium on Computer and Communication*, June 2014, pp. 1–6.

[48] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, "Fast join and synchronization scheme in the IEEE 802.15.4e MAC," in *Proc. of IEEE Wireless Communication Network Conference Workshops*, March 2015, pp. 85–90.

[49] T. P. Duy and Y. Kim, "An efficient joining scheme in IEEE 802.15.4e," in *Proc. of International Conference on Information and Communication Technology Convergence (ICTC)*, 2015, pp. 226–229.

[50] D. D. Guglielmo, S. Brienza, and G. Anastasi, "A Model-based Beacon Scheduling algorithm for IEEE 802.15.4e TSCH networks," in *Proc. of IEEE International Symposium: A World Wireless, Mobile Multimedia Network*, June 2016, pp. 1–9.

[51] T. P. Duy, T. Dinh, and Y. Kim, "A rapid joining scheme based on fuzzy logic for highly dynamic IEEE 802.15.4e time-slotted channel hopping networks," *International Journal of Distributed Sensor Networks*, vol. 12, no. 8, pp. 1–10, 2016.

[52] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, "Fast network joining algorithms in industrial IEEE 802.15.4 deployments," *Ad Hoc Netw.*, vol. 69, pp. 65–75, 2018.

[53] I. Khoufi, P. Minet, and B. Rmili, "Beacon Advertising in an IEEE 802.15.4e TSCH Network for Space Launch Vehicles," in *Proc. of 7th European Conference for Aeronautics and Aerospace Sciences*, July 2017, pp. 1–15.

[54] I. Khoufi and P. Minet, "An Enhanced Deterministic Beacon Advertising Algorithm for Building TSCH Networks," *Annals of Telecommunications*, May 2018.

[55] C. M. García Algora, V. Alfonso Reguera, E. M. García Fernández, and K. Steenhaut, "Parallel Rendezvous-Based Association for IEEE 802.15.4 TSCH Networks," *IEEE Sensors Journal*, vol. 18, no. 21, pp. 9005–9020, 2018.

[56] B.-H. Bae and S.-H. Chung, "Fast Synchronization Scheme Using 2-Way Parallel Rendezvous in IEEE 802.15.4 TSCH," *Sensors*, vol. 20, no. 5, 2020.

[57] M. Mohamadi, B. Djamaa, M. R. Senouci, and A. Mellouk, "FAN: Fast and Active Network Formation in IEEE 802.15.4 TSCH Networks," *Journal of Network and Computer Applications*, vol. 183-184, p. 103026, 2021.

[58] G. Mulligan, "The 6LoWPAN Architecture," in *Proc. of Workshop on Embedded Networked Sensors (EmNets)*, 2007, p. 78–82.

[59] C. Gomez, J. Paradells, C. Bormann, and J. Crowcroft, "From 6LoWPAN to 6Lo: Expanding the Universe of IPv6-Supported Technologies for the Internet of Things," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 148–155, 2017.

[60] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6TiSCH: deterministic IP-enabled industrial internet (of things)," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, 2014.

[61] T. Kivinen and P. Kinney, "IEEE 802.15.4 Information Element for the IETF," RFC 8137, Internet Engineering Task Force, Tech. Rep. 8137, May 2017.

[62] T. Chang, M. Vučinić, X. Vilajosana, S. Duquennoy, and D. R. Dujovne, "6TiSCH Minimal Scheduling Function (MSF)," RFC 9033, Internet Engineering Task Force, Tech. Rep. 9033, May 2021.

[63] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH," in *Proc. of ACM Conference on Embedded Networked Sensor Systems*, 2015, pp. 337–350.

[64] S. Jeong, H.-S. Kim, J. Paek, and S. Bahk, "OST: On-Demand TSCH Scheduling with Traffic-Awareness," in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, 2020, pp. 69–78.

[65] S. Kim, H.-S. Kim, and C. Kim, "ALICE: Autonomous Link-Based Cell Scheduling for TSCH," in *Proc. of International Conference on Information Processing in Sensor Networks (IPSN)*, ser. IPSN '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 121–132.

[66] G. Montenegro, J. Hui, D. Culler, and N. Kushalnagar, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, Internet Engineering Task Force, Tech. Rep. 4944, Sep. 2007.

[67] P. Thubert and J. Hui, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks," RFC 6282, Internet Engineering Task Force, Tech. Rep. 6282, Sep. 2011.

[68] P. Thubert and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch," RFC 8025, Internet Engineering Task Force, Tech. Rep. 8025, Nov. 2016.

[69] P. Thubert, C. Bormann, L. Toutain, and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header," RFC 8138, Internet Engineering Task Force, Tech. Rep. 8138, Apr. 2017.

[70] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)," RFC 6552, Internet Engineering Task Force, Tech. Rep. 6552, March 2012.

[71] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," RFC 7252, Internet Engineering Task Force, Tech. Rep. 7252, 2014.

[72] M. Vučinić, M. Król, B. Jonglez, T. Coladon, and B. Tourancheau, "Trickle-D: High Fairness and Low Transmission Load With Dynamic Redundancy," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1477–1488, 2017.

[73] B. Ghaleb, A. Y. Al-Dubai, E. Ekonomou, I. Romdhani, Y. Nasser, and A. Boukerche, "A Novel Adaptive and Efficient Routing Update Scheme for Low-Power Lossy Networks in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5177–5189, 2018.

[74] A. Karalis, "ATP: A Fast Joining Technique for IEEE802.15. 4-TSCH Networks," in *Proc. of International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2018, pp. 588–599.

[75] C. M. G. Algora, E. O. Guerra, S. Montejo-Sánchez, E. M. G. Fernández, and K. Steenhaut, "A Theoretical Association Time Model for IEEE 802.15.4 TSCH Networks," *IEEE Communications Letters*, vol. 25, no. 2, pp. 656–659, 2021.

[76] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm," Internet Engineering Task Force, RFC 6206, March 2011.

[77] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, *TinyOS: An Operating System for Sensor Networks*. Springer Berlin Heidelberg, 2005, pp. 115–148.

[78] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors," in *Proc. of Annual IEEE International Conference on Local Computer Network*, 2004, pp. 455–462.

[79] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, and T. C. Schmidt, "RIOT OS: Towards an OS for the Internet of Things," in *Proc. of Computer Communications Workshops (INFOCOM WORKSHOPS)*, 2013, pp. 79–80.

[80] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne, "FIT IoT-LAB: A large scale open experimental IoT testbed," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 459–464.

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

[82] M. Khatua and S. Misra, "D2D: Delay-Aware Distributed Dynamic Adaptation of Contention Window in Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 2, pp. 322–335, 2016.

[83] C. Vallati and E. Mingozzi, "Trickle-F: Fair broadcast suppression to improve energy-efficient route formation with the RPL routing protocol," in *Proc. of Sustainable Internet and ICT for Sustainability (SustainIT)*, 2013, pp. 1–9.

[84] M. Vučinić, G. Romaniello, L. Guelorget, B. Tourancheau, F. Rousseau, O. Alphand, A. Duda, and L. Damon, "Topology construction in RPL networks over beacon-enabled 802.15.4," in *Proc. of IEEE Symposium on Computers and Communications (ISCC)*, 2014, pp. 1–7.

[85] M. B. Yassein *et al.*, "A new elastic trickle timer algorithm for Internet of Things," *Journal of Network and Computer Applications*, vol. 89, pp. 38 – 47, 2017.

[86] Y. Xiao, "Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e wireless LANs," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1506–1515, 2005.

[87] T. M. M. Meyfroyt, M. Stolikj, and J. J. Lukkien, "Adaptive broadcast suppression for Trickle-based protocols," in *Proc. of IEEE International Symposium: A World Wireless, Mobile Multimedia Network*, 2015, pp. 1–9.

[88] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems," *Eastern Research. Lab, Digital Equipment Coporation*, vol. 38, 1984.

[89] H. A. A. Al-Kashoash, M. Hafeez, and A. H. Kemp, "Congestion Control for 6LoWPAN Networks: A Game Theoretic Framework," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 760–771, 2017.

[90] S. Chowdhury, A. Benslimane, and C. Giri, "Noncooperative Gaming for Energy-Efficient Congestion Control in 6LoWPAN," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4777–4788, 2020.

[91] F. Goudarzi and H. Asgari, "Non-Cooperative Beacon Rate and Awareness Control for VANETs," *IEEE Access*, vol. 5, pp. 16 858–16 870, 2017.

[92] H. Nikaidô and K. Isoda, "Note on non-cooperative convex games," *Pacific Journal of Mathematics*, vol. 5, no. S1, pp. 807 – 815, 1955.

[93] J. B. Rosen, "Existence and Uniqueness of Equilibrium Points for Concave N-Person Games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.

[94] M. V. Ramakrishna and J. Zobel, *Performance in Practice of String Hashing Functions*, 1997, pp. 215–223.

[95] D. Stanislowski, X. Vilajosana, Q. Wang, T. Watteyne, and K. S. J. Pister, "Adaptive Synchronization in IEEE802.15.4e Networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 795–802, 2014.

[96] T. Chang, T. Watteyne, K. Pister, and Q. Wang, "Adaptive synchronization in multi-hop TSCH networks," *Computer Networks*, vol. 76, pp. 165–176, 2015.

[97] Y. Boufenneche, R. Zitouni, L. George, and N. Gharbi, "Network Formation in 6TiSCH Industrial Internet of Things under Misbehaved Nodes," in *Proc. of International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 2020, pp. 1–6.

[98] C. Pu, "Sybil Attack in RPL-Based Internet of Things: Analysis and Defenses," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4937–4949, 2020.

[99] J. Granjal, E. Monteiro, and J. Sá Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[100] M. Baddeley, R. Nejabati, G. Oikonomou, S. Gormus, M. Sooriyabandara, and D. Simeonidou, "Isolating SDN control traffic with layer-2 slicing in 6TiSCH industrial IoT networks," in *Proc. of IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2017, pp. 247–251.

[101] M. Baddeley, U. Raza, A. Stanoev, G. Oikonomou, R. Nejabati, M. Sooriyabandara, and D. Simeonidou, "Atomic-SDN: Is Synchronous Flooding the Solution to Software-Defined Networking in IoT?" *IEEE Access*, vol. 7, pp. 96 019–96 034, 2019.

Publications Related to Thesis

Journals

1. **Alakesh Kalita** and Manas Khatua, "6TiSCH – IPv6 Enabled Open Stack IoT Network Formation: A Review", ***ACM Transactions on Internet of Things***, 2022, (Accepted) [Chapter 2]

2. **Alakesh Kalita** and Manas Khatua, "Channel Condition Based Dynamic Beacon Interval for Faster Formation of 6TiSCH Network", ***IEEE Transactions on Mobile Computing***, vol. 20, no. 7, pp. 2326-2337, August, 2021, [Chapter 3]

3. **Alakesh Kalita** and Manas Khatua, "Opportunistic Transmission of Control Packets for Faster Formation of 6TiSCH Network", ***ACM Transactions on Internet of Things***, vol. 2, no. 1, pp. 1-29, February, 2021, [Chapter 4]

4. **Alakesh Kalita** and Manas Khatua, "Adaptive Control Packet Broadcasting Scheme for Faster 6TiSCH Network Bootstrapping", ***IEEE Internet of Things Journal***, vol. 8, no. 24, pp. 17395–17402, 2021, [Chapter 5]

5. **Alakesh Kalita** and Manas Khatua, "A Non-cooperative Gaming Approach for Control Packet Transmission in 6TiSCH Network", ***IEEE Internet of Things Journal***, vol. 9, no. 5, pp. 3954-3961, 2022, [Chapter 6]

6. **Alakesh Kalita** and Manas Khatua, "Autonomous Allocation and Scheduling of Minimal Cell in 6TiSCH Network", ***IEEE Internet of Things Journal***, vol. 8, no. 15, pp. 12242-12250, August, 2021, [Chapter 7]

Conferences

1. **Alakesh Kalita** and Manas Khatua, "Faster Joining in 6TiSCH Network using Dynamic Beacon Interval", ***Proceedings of the*** $11th$ ***International Conference on Communication Systems Networks***, January 7-11, 2019, Bangalore, India, pp. 454-457, [Chapter 3]

2. **Alakesh Kalita** and Manas Khatua, "Opportunistic Priority Alternation Scheme for Faster Formation of 6TiSCH Network", ***Proceedings of the*** $21st$ ***International Conference on Distributed Computing and Networking***, January 4-7, 2020, Kolkata, India, pp. 1-5, [Chapter 4]

Under Review

Journals

1. **Alakesh Kalita** and Manas Khatua, "Time-Variant RGB Model for Minimal Cell Allocation and Scheduling in 6TiSCH Networks", ***IEEE Transactions on Mobile Computing***, [Chapter 7]

Publications Outside Thesis

Journal

1. **Alakesh Kalita**, A. Brighente, M. Khatua, and M. Conti, "Effect of DIS attack on 6TiSCH Network Formation", ***IEEE Communications Letters***, vol. 26, no. 5, pp. 1190-1193, May, 2022

Conferences

1. **Alakesh Kalita**, N. Ahmed, H. Rahman, and M. I. Hussain, "A QoS-aware MAC protocol for large-scale networks in Internet of Things," ***Proceedings of the International Conference on Advanced Networks and Telecommunications Systems***, December 2017, pp. 1–6.

2. **Alakesh Kalita**, K. Ray, A. Biswas, and M. A. Hussain, "A topology for network-on-chip," ***Proceedings of the International Conference on Information Communication and Embedded Systems***, February 2016, pp. 1–7.

3. K. Ray, **Alakesh Kalita**, A. Biswas, and M. A. Hussain, "A multipath networkon-chip topology," ***Proceedings of the International Conference on Information Communication and Embedded Systems***, February 2016, pp. 1–7.

4. A. Biswas, M. A. Hussain, and **Alakesh Kalita**, "An improved congestion free modified fat tree network," ***Proceedings of the International Conference on Signal Processing, Communication, Power and Embedded System***, October 2016, pp. 759-763.

Appendices

A

Simulation and Testbed Setup

This appendix focuses on the simulation and testbed experimental set up used in our thesis. We used Contiki-NG, which is an open-source, cross-platform operating system for next generation IoT devices. It focuses on dependable (secure and reliable) low-power communication and standard protocols, such as IPv6/6LoWPAN, 6TiSCH, RPL, and CoAP. Contiki-NG allows the user to modify standard protocols for research and developments. We used FIT IoT-LAB to test our contributions on real IoT network. FIT IoT-LAB is an open-source IoT lab that provides a very large-scale infrastructure suitable for testing small wireless sensor devices and heterogeneous communicating objects. In brief, IoT-LAB provides a testing environment of multi-platform, multi-radio, multi-topology, and multi-OS based configurable IoT networks. Here, we provide detailed description of the steps to modify standard protocols stack and to create binary executable file for resource constrained IoT nodes. We also provide the steps required to flash the Contik-NG OS on the devices available at FIT IoT-LAB and information about how to obtain output from those devices through `SSH` connection.

A.1 Modification on Contiki-NG

We modified existing IoT protocol layers as per our requirement. The related files are available on different directories of Contiki-NG. For example, we modified the `MAC` layer files, which are available at below directory.

`/home/alakesh/contiki-ng/os/net`

Therefore, at the begining, user needs to make changes in different files of Contiki-NG as per the requirement. Specifically, we modified the files related to `TSCH MAC` layer and `RPL-Lite` routing layer.

A.2 Creating and Uploading Contiki-NG OS

The user needs to select the application first which he/she wants to run on hardware devices. After selecting the application, user needs to open the terminal inside the selected

application directory and execute the following commands one by one. We selected the simple-node application to test all of our contributions which is available at the `example` directory i.e.,

`/home/alakesh/contiki-ng/examples/ 6tisch/simple-node`

- At first, user needs to specify the types of the CPU and board (model) details of the device where he/she wants to run the application. It was done by using the following command,

  `make TARGET=cc26x0-cc13x0 BOARD=launchpad/cc2650 savetarget`

  Please note that the above command is for Texax Instruments' CC2650 Launchpad device. TARGET and BOARD types both vary from device to device. For example, for FIT IoT-LAB's M3 device, following command is used

  make TARGET=iotlab BOARD=m3 savetarget

  So, the user needs to correctly specify the TARGET and BOARD types as per the device used in the experiments.

- Now, if the user has made any changes in any of the Contiki-NG files then the below command needs to be executed, otherwise, changes will not be reflected during compilation.

  `make distclean`

- Finally, the binary file of the chosen application can be generated using the following command,

  `sudo make BOARD=launchpad/cc2650`

  *After successful compilation, one **"build"** directory will be created inside the application directory, and the generated binary files (with the file extension such as .elf, .hex, and .bin) will available in that new directory.*

- User can also directly upload/flash the binary file using a single command as follows,

```
sudo make PORT=/dev/ttyACM0 BOARD=launchpad/cc2650
```

Here, `PORT` specifies the port number (here, it is ttyACM0) where the device is connected via `USB` cable.

- Otherwise, user can upload the binary file using the below command,

```
make TARGET=cc26x0-cc13x0 BOARD=launchpad/cc2650
node.upload PORT=/dev/ttyACM0
```

Here, `node.upload` is the generated binary file.

- User can also use the open source **UNIFLASH** (a stand-alone flashing tool) software to upload the binary file on the devices.

A.3 FIT IoT-LAB: SSH connection

To upload binary file on the available devices in FIT IoT-LAB and receive aggregated output from more than one devices into a single file, user should follow the below steps.

- At first, the user needs to establish "`SSH`" connection with the FIT IoT-LAB server. Note that FIT IoT-LAB deploys the nodes in different cities across the France. But, a user can establish only one `SSH` connection at a time. For this, the user needs to do the following steps:
  - Create an account in FIT IoT-LAB,
  - Generate `SSH` key for our system,
  - Update the generated `SSH` key of our system in FIT IoT-LAB account,
- After that, user needs to build the Contiki-NG binary file at his/her own system (.elf/.bin/.hex) using the following commands,
  - ARCH_PATH=../../../../arch make TARGET=iotlab BOARD=m3 savetarget
  - ARCH_PATH=../../../../arch make clean

– ARCH_PATH=../../../../arch make

Note that before creating binary files, user needs the architecture of `iotlab-m3` board inside his/her Contiki-NG directory. For this, user can clone the files from the below git link,

(git clone) https://github.com/iot-lab/iot-lab.git

- After building the binary file, user needs to connect a server using the following `SSH` command,

  `ssh kalita@grenoble.iot-lab.info`

  Here, `kalita` is the user name of FIT IoT-LAB and `grenoble` is the chosen server.

- Note that when the user tries to establish the connection for the first time, he/she needs to perform the authentication using the following command,

  `Iotlab auth -u kalita`

  After, executing this command, user needs to enter the FIT IoT-LAB his/her password in the `SSH` terminal.

- Next, user needs to upload the binary file(s) on the server using the below command,

  `scp node.iotlab kalita@grenoble.iot-lab.info`

  Here, node.iotlab is the uploaded binary file.

- Next, user needs to specify the time duration of an experiment as follows,

  `export d=61`

- User also needs to specify the device(s)' id

  `export nodes='1-60'+63`

  Here, using this command, total 61 nodes will be selected for the experiment. User can choose any random nodes from the deployed sites. However, the chosen nodes should support the binary file that the user wants to flash.

- Next, user needs to mention the file name in which he/she wants the aggregated output from all selected the devices,

  ```
  export logfile=testbed.log
  ```

- Then, user can submit his/her experiment using the following command,

  ```
  iotlab-experiment submit -n
  myexperiment -d $d -l grenoble,m3,$nodes;
  ```

  Sometimes, the devices selected by a user may not be available (used by some other user as FIT IoT-LAB is open source), therefore, in this case, the user needs to run the following the command before uploading the binary file on the devices,

  ```
  iotlab-experiment wait;
  ```

- Once all the selected devices are available, user can upload the binary file to the selected nodes using the following command,

  ```
  iotlab-node --update node.iotlab
  ```

- When the experiment time gets expired, user needs to execute the following command to aggregate the output of the nodes into a single file,

  ```
  serial_aggregator > $logfile
  ```

- Finally, the output file can be downloaded using the following command,

  ```
  scp kalita@grenoble.iot-lab.info:testbed.log
  ```

- The collected output file needs to be processed further to get the desired metrics.

A.4 Processing Motes' Output

The output file contains all the log information of all the layers of each nodes. To get the desired output, we used an already available python script. This script is available at Contiki-NG example directory. However, we modified the script as per our requirement.

`contiki-ng/examples/benchmarks`

Please note that user needs to make changes in the Contiki-NG source files so that his/her required output gets printed on the logfile generated by Contiki-NG. For example, we processed the DIO receiving time and first EB generation time of the nodes as follows:

```
if "Received first DIO" in line:
    nodes[node].SixTiSCH_time=int(fields[8])
    print(nodes[node].SixTiSCH_time)

#1626088126.725579;m3-50;[INFO: TSCH] First EB is
generated : 56 seconds
if "First EB is generated" in line:
    nodes[node].SixTiSCH_time=int(fields[10])
    #print(nodes[node].SixTiSCH_time)

print(result)
plot(ids, [r[1] for r in results], "tsch", "TSCH Joining
time (seconds)")
plot(ids, [r[2] for r in results], "sixtisch", "6TiSCH
Joining time (seconds)")
```

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati 781039, India