University of Windsor

Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2010

Feedback based real-time MAC (RT-MAC) protocol for data packet streaming in wireless sensor networks

Brajendra Kumar Singh
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation

Singh, Brajendra Kumar, "Feedback based real-time MAC (RT-MAC) protocol for data packet streaming in wireless sensor networks" (2010). *Electronic Theses and Dissertations*. 8085.
https://scholar.uwindsor.ca/etd/8085

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Feedback based Real-time MAC (RT-MAC) Protocol for Data Packet Streaming in Wireless Sensor Networks

By

Brajendra Kumar Singh

A Dissertation
Submitted to the Faculty of Graduate Studies
through the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada
2010

© 2010 Brajendra Kumar Singh

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-62764-8
Our file Notre référence
ISBN: 978-0-494-62764-8

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Declaration of Co-Authorship / Previous Publication

I. Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is result of joint research, as follows:

This thesis contains the outcome of a research undertaken by me under the supervision of Professor Dr. K. E. Tepe. The collaboration is covered in Chapters 3, 4, 5, and 6 of the thesis. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author, and the contribution of co-authors was primarily through the provision of valuable suggestions and helping in comprehensive analysis of the experimental results submitted for publication.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This thesis includes six original papers that have been previously published/submitted for publication in peer reviewed conference, as follows:

Thesis Chapter	Publication title/full citation	Publication status
Chapters 3, 4, and 7	B.K. Singh and K.E. Tepe, "Feedback based Real-time MAC (RT-MAC) protocol for wireless sensor networks", IEEE Global Communications Conference, GLOBECOM 2009.	Published
Chapters 3, 4, and 7	Brajendra Kumar Singh, and Kemal Ertugrul Tepe, "A Novel Real-time MAC Layer Protocol for Wireless Sensor Network Applications", IXth IEEE International Conference on Electro/Information Technology, EIT 2009.	Published
Chapters 3, 4, and 7	Brajendra Kumar Singh, and Kemal Ertugrul Tepe, "Feedback based MAC Protocol for Real-time Data Streaming in Wireless Sensor Networks", IEEE/ACM Transactions on Networking.	First phase reviews are received. The updated paper is to be submitted soon for the next review phase.
Chapters 3, 4, 6, and 7	Brajendra Kumar Singh, and Kemal Ertugrul Tepe, "Feedback based solution to false blocking problem at MAC layer in Wireless Sensor Networks", IEEE Communications Letters.	Submitted
Chapters 6, and 7	Brajendra Kumar Singh, and Kemal Ertugrul Tepe, "Multi-stream support in Real-Time MAC (RT-MAC) protocol for Wireless Sensor Networks ".	To be submitted
Chapters 1, 2, and 7	Brajendra Kumar Singh, and Kemal Ertugrul Tepe, "A survey on Real-time MAC protocols for Wireless Sensor Networks".	To be submitted

I certify that I am the copyright owner of the above published materials in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor does it violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Wireless sensor networks (WSNs) are generally used for event driven monitoring or periodic reporting. Once a triggering event happens, it needs to be reported in real-time as a continuous stream for some duration. In order to address such communication requirements, this thesis introduces a soft Real-Time MAC (RT-MAC) protocol for real-time data packet streaming in wireless sensor networks. RT-MAC eliminates contention for a wireless medium by introducing a feedback control packet, called Clear Channel (CC). As a result, RT-MAC has a consistent and predictable data transmission pattern that provides end-to-end delay guarantees. Additionally, RT-MAC has a lower end-to-end delay than other real-time WSN MAC protocols for two reasons: (1) it maximizes spatial channel reuse by avoiding the false blocking problem caused by request-to-send (RTS) and clear-to-send (CTS) exchanges in wireless MAC protocols (2) it reduces contention duration of control packets to facilitate faster data packet transfer. Thus, RT-MAC facilitates periodic data packet deliveries as well as alarming event reporting. RT-MAC operates both with and without duty cycle mode (sleep/wakeup schedule for sensor nodes). Duty cycle mode of RT-MAC is useful in situations where energy conservation is one of the goals along with real-time requirements. RT-MAC is well suited for multi-hop communication with a large number of hops. RT-MAC protocol supports single-stream communication between a randomly selected source and sink node pair as well as multi-stream communication among different source and sink node pairs. This thesis provides the lower and upper end-to-end delay bounds for data packets transfer in normal mode of operation of RT-MAC protocol. We used state diagram analysis to show the in-depth functioning of RT-MAC protocol. This thesis also presents Markov analysis of RT-MAC that shows the behavior of the protocol in fault scenarios. Extensive simulation results are also presented in this thesis. These results show significant improvement in delay, packet throughput performance, and uniformity in packet transmission pattern at a cost of a very

small increase in energy consumption as compared to other real-time MAC protocols such as VTS and general purpose MAC protocols such as S-MAC and T-MAC.

To my son Aditya, wife Ashita, and my parents.

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Kemal E. Tepe for his supervision, guidance, and moral and financial support during my research work. I greatly admire his wonderful personality, abundant patience, and innovative suggestions that helped me tremendously in my research. I consider myself very fortunate that I got help from my supervisor all along, sometimes even beyond my usual expectations. I would also like to thank my thesis committee members, Dr. Ganesan, Dr. Jaekel, Dr. Wu, and Dr. Khalid, for their valuable contributions and suggestions. I am thankful to my colleagues and seniors, P. Dhar, R.K. dave, P. L. Kulkarni, A.K. Lal, J.B. Bhatt, and S.C. Pandya, at Indian Space Research Organization, India, for giving me permission and study leave to pursue my doctoral studies at the University of Windsor. I would like to thank immensely Dr. Haresh Bhatt for teaching me ABC of research and generating a spark in me to pursue a PhD degree. I also would like to thank Dr. A.K. Aggarwal for encouraging me to come to the University of Windsor for pursuing doctoral research. I am extremely thankful to Nazrul Islam for being my friend, philosopher, and guide. Whenever I am in deep trouble, his name comes first in my mind for any kind of suggestion. I also would like to thank Khaja Mohammad Shazzad, and Kazi Atiqur Rahman for their useful discussion on Markov Analysis. I would like to thank all the staff members of the Department of Electrical and Computer Engineering, especially, Andria for her caring and helpful attitude throughout my study duration at the university. I would like to acknowledge the valuable support by the Ministry of Colleges and Universities, Ontario, Canada in my research. Especially, I am grateful to my loving parents for their encouragement, and moral support. I'll forever by indebted to them for my upbringing. I have no word to express my gratitude to my wife Ashita for her passionate, caring and loving approach, which has taken the load off my shoulders. Without her cooperative support, I would not be able to accomplish my work. I count myself blessed to have a caring and dedicated wife and a loving son, Aditya, in my life.

CONTENTS

List of Figures

List of Tables

Abbreviations

ACK	Acknowledgement
CA	Collision Avoidance
CC	Clear Channel
CSMA	Carrier Sense Multiple Access
CTS	Clear To Send
CR-SLF	Channel Reuse Smallest Latest-Start-Time First
DC	Duty Cycle
DTMC	Discrete Time Markov Chain
GDB	The GNU Debugger
GTS	Guaranteed Time Slot
GHz	Giga Hertz
GPS	Global Positioning System
I-EDF	Implicit Earliest Deadline First
IEEE	The Institute of Electrical and Electronic Engineers
LPRT	Low Power Real-time
MAC	Medium Access Control
MHz	Mega Hertz
PR-MAC	Path-oriented Real-time MAC
QoS	Quality of Service
RT	Real-time
RT-MAC	Real-time MAC
RTS	Request To Send
SYNC	Synchronization
SID	Stream ID
TDMA	Time Division Multiple Access

TDMA-EC	Time Division Multiple Access - Energy Conserving
VTS	Virtual TDMA for Sensors
WSN	Wireless Sensor Network

Chapter 1: Introduction

1.1 Motivation

Over the past decade, the wireless sensor networks (WSNs) [8], [17], [37] and [47] have become a rapidly developing research area. Akyildiz et al. [3], [4] and Yick et al. [73] highlight research challenges in general at various layers of the WSN protocol stack. However, most research in WSNs is devoted to energy conservation since batteries are generally not replaceable once a sensor network is deployed. Timing constraints received secondary importance. But with the advancement of sensor technologies, sensors are increasingly used for time critical (real-time) applications. To fulfill real-time requirements, time critical aspects need to be addressed both at the hardware as well as software levels in WSNs. References [1], [12], [13], [46], [55] and [70] focus on research issues related to real-time communication at various network layers of the WSNs. However, for a given sensor hardware, medium access control (MAC) protocol is crucial for guarantying real-time sensor data transmission. A real-time MAC protocol is essential for any time critical higher layer protocol development in WSNs. Thus, in this thesis, a real-time MAC protocol for WSNs is proposed and its operation is verified via computer simulations and Markov chain analysis.

A real-time MAC protocol should have the following properties:

1. It should provide a bounded end-to-end delay, which requires a consistent and predictable data packet transmission pattern.
2. It should have minimum scheduling, contention, and control packet transmission delays compared to the actual time taken for data packet transfer. Thus, the protocol should be fast enough to meet end-to-end delay deadlines of a real-time application.

3. It should be fault-tolerant (i.e., it should not go into any deadlock state).

Existing real-time MAC protocols for WSNs are designed either for some specific sensor network topologies, or provide large delay guaranties, or make assumptions which limits their practicality such as having special hardware requirements or fixed duty cycles (sleep/wakeup schedules for sensor nodes) during run time. Therefore, Real-Time MAC (RT-MAC) protocol for WSNs that removes some of the limitations of the existing protocols is proposed in this thesis. Verification of RT-MAC is done via computer simulations and Markov chain analysis.

1.2 Research Objectives

The objective of this research is to devise a real-time MAC protocol for WSNs that guaranties bounded and minimum end-to-end delay, supports MAC operation with and without duty cycle, supports event driven and periodic applications, has no hardware assumptions, is applicable for randomly deployed multi-hop WSN, is fault tolerant, and can vary duty cycle during run time.

1.3 Contributions and Applicability

RT-MAC, as presented in this thesis, uses a novel feedback mechanism to provide real-time capability to MAC protocol. This feedback mechanism controls the flow of data packet streams in multi-hop WSNs. RT-MAC introduces a novel feedback control packet, called Clear Channel (CC), to regulate the medium access for WSNs. Thus, unlike most of contention based MAC protocols, RT-MAC does not require large carrier sense duration prior to ready to send (RTS) control packets to resolve medium access problem. Major features and advantages of RT-MAC protocol are given as follows:

- RT-MAC works with any network topology.
- RT-MAC does not require any special hardware such as GPS systems, multi-frequency transceivers, router sensor nodes, high power cluster heads etc.

- RT-MAC has lower end-to-end delay guarantees as compared to time scheduling based real-time MAC protocols.
- RT-MAC can work with and without duty cycle and can operate at lower duty cycles than other contention based MAC protocols.
- RT-MAC can vary duty cycle during run time based on the application requirement or to accommodate load fluctuations. Varying duty cycle can also help in guarantying delay bounds in multi-stream scenarios.
- RT-MAC provides consistent and predictable data transmission pattern.
- RT-MAC has a very low number of collisions.
- RT-MAC is a distributed, scalable, and load balanced protocol.
- RT-MAC is a fault tolerant protocol.

In general, WSN is useful for a variety of real-life applications [2], [5], [6], [7], [14], [30] and [69]. Real-time communication applications can be classified as soft real-time and hard real-time applications. In soft real-time applications, a data packet is processed even if it misses the delay deadline but arrives at the destination within an acceptable delay threshold. In hard real-time applications, a data packet is dropped by the network if it misses a specified delay deadline. RT-MAC, presented in this thesis, is designed for soft real-time WSN applications. It is capable of meeting deadlines of both event driven and periodic applications. Event driven applications can include fault detection of power lines and security setups at important establishments. For example, if there is a short-circuit in power lines, then the alarm message can be sent to the controlling station within a guaranteed time limit. Periodic applications can include the monitoring of assembly lines in industries and natural disaster management applications. RT-MAC is also useful for the WSN applications that require variable periodicity. As RT-MAC supports change in duty cycle during run time, the frequency and end-to-end delay of data reporting can be controlled while sensors are in operation. For example, if controlling station gets information about any triggering event such as a rapidly spreading forest fire, then it can increase data collection rate to the maximum possible value. In such cases, timely reporting of event becomes crucial to the extent that one does not care even if network operate at 100% duty cycle (i.e., nodes are ON continuously) for the duration of the disaster then dies; thus, energy consumption generally assumes secondary importance in such scenarios.

1.4 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 provides literature review on existing MAC approaches in WSNs. A comparative study of existing real-time MAC protocols is also presented in Chapter 2. Chapter 3 presents an in depth description of the proposed protocol in the single-stream scenario. Chapter 4 presents analytical framework to calculate the lower as well as upper end-to-end delay bounds in normal operation of the protocol. Chapter 5 presents stochastic modeling of the proposed protocol in faulty scenarios. Chapter 6 presents the description of the proposed protocol in the multi-stream scenario. Chapter 7 gives information about the simulation environment, simulation results, and related discussions. Conclusion and future work are presented in Chapter 8.

Chapter 2: Background

This chapter presents a background study related to the research work presented in this thesis. The next section presents an overview of MAC protocols for WSNs. Section 2.2 presents a comparative study of the real-time MAC protocols for WSNs. An introduction of Markov analysis is presented in Section 2.3. Markov analysis is used to analyze the behavior of the protocol proposed in this thesis in realistic scenarios. Section 2.4 presents an overview of OMNeT++ simulator, which is used in this research work for comparative performance evaluation of the real-time MAC protocols for WSNs.

2.1 Overview of MAC protocols for WSNs

Czapski [18], and Demirkol et al. [19] present a comprehensive survey of MAC protocols for WSNs; though, most of MAC protocols presented in these papers give emphasis on energy saving.

MAC protocols for WSNs are broadly categorized as contention based (random access) protocols such as carrier sense multiple access (CSMA) based protocols and deterministic scheduling protocols such as time division multiple access (TDMA) based protocols [16], [24], [27], [28], [44], [54] and [74]. It is relatively easier to define a delay deadline within the TDMA mechanism. However, TDMA based protocols have some disadvantages with regard to real-time application requirements at MAC layer in WSNs. For example, TDMA based MAC protocols cannot adapt well to frequently changing load conditions due to higher synchronization requirements; thus, the TDMA based MAC protocols are good for periodic data packet delivery scenarios, but they are not good for event driven reporting. In addition to this, nodes can send data in turns, which makes TDMA unsuitable for reporting alarm events in real-time. The end-to-end delay bounds of TDMA based protocols are larger compared to those of contention based protocols. The TDMA based protocols exhibit large overhead to maintain synchronization, which increase the overall delay and energy consumption. Therefore, TDMA based protocols are not good for large multi-hop WSNs, particularly when data follows a long linear path. End-to-end delay in TDMA based protocols grows linearly with increasing number of sensors using slots in a TDMA frame. TDMA based MAC protocols give a fair chance of medium access to each node in a TDMA frame even if any node does not have data packet to send during a TDMA frame, which may add unnecessary delay and increase energy consumption. One of the widely referred TDMA based MAC protocol, called TDMA-EC, is proposed by Ren et al. in [52] for tree topology based WSNs. TDMA-EC has two types of staggered schedules to facilitate bidirectional message transfer. It takes a long time for TDMA-EC to come out with a new synchronized TDMA frame if the sink node changes. In contrast to TDMA based MAC approaches, RT-MAC presented in this thesis attempts to define delay deadlines with random access approach at MAC layer. RT-MAC can adapt quickly to fluctuating load conditions as it can vary the duty cycle during run time. Additionally, RT-MAC adapts to change in route very quickly.

In general, contention based MAC approaches are more suitable for real-time communication at MAC layer due to their scalability and flexibility in adapting changing application requirements or load scenarios, especially for event driven WSN applications. However, the general purpose contention based MAC protocols such as S-MAC [71], [72] and T-MAC [62], and D-MAC [41] and [42] are not suitable for real-time communication due to their inconsistent data transmission patterns, which cause unpredictable end-to-end delays. But these three general purpose MAC provide basis for the development of contention based real-time MAC protocols. That is why brief descriptions of these three protocols are provided in the following subsections.

2.1.1 Overview of S-MAC protocol

RT-MAC protocol, presented in this thesis, is based on S-MAC protocol [71] and [72]. S-MAC works on the basis of coordinated adaptive sleeping. S-MAC uses SYNC control packets to keep coordination of sleep and listen schedule locally for a group of sensor nodes in a neighborhood. A group of sensor nodes forms a virtual cluster. Thus, the whole network is divided into several virtual clusters. The boundary nodes follow the sleep and listen schedule of the virtual clusters that they are in. S-MAC works with or without adaptive listening. In case of adaptive listening in S-MAC, the ongoing communication between one hop neighboring node pair (e.g. N_1 as sender node and N_2 as receiver node in Figure 2.1) is overheard by their neighboring nodes in range. These neighboring nodes go into sleep mode for the duration of an active communication. The neighboring nodes (e.g. N_3 in Figure 2.1) of receiving node N_2 wake up as soon as the active communication finishes, and remain in listen mode for some time, irrespective of their sleep and listen schedule, with expectation that N_2 may try to forward data packet to them. Though adaptive listening helps to reduce sleep latency in S-MAC protocol, it has two limitations. First, effect of adaptive listening is limited to two hops with respect to N_1. It is because the nodes that are two or three hops away with respect of N_2 will not be able to overhear an active communication and will not be awake. Hence, one hop neighbor of the receiving node N_2, which was awake due to overheard communication will receive data packet from N_2, but it will not be able to forward this data packet further because its neighboring nodes will only come out of sleep mode at the beginning of the next frame. Throughout this thesis, a frame refers to one listen and sleep duration at the MAC layer. Second, if duty cycle is reduced to the extent that data packet travels only one hop per frame, then adaptive listening reduces end-to-end delay by half. However, if duty cycle is large, then data packet may travel a large number of hops per frame. In this case, the data packet transmission by an additional hop per frame due to adaptive listening is not going to have a significant impact on decreasing end-to-end delay.

S-MAC itself is not suitable for real-time applications for several reasons. First, S-MAC uses carrier sense, contention and back-off schemes for the wireless medium access. Thus, in general, it is not known a priori as to which wireless sensor node will win the contention in a neighborhood. Hence, it is not possible to predict data transmission pattern in the network. Thus, there is no guaranty of ordered delivery of data packets. It may happen that the first packet generated in a node may not be the first one to reach to the destination node. For example, Figure 2.1 shows one of such possible data packet

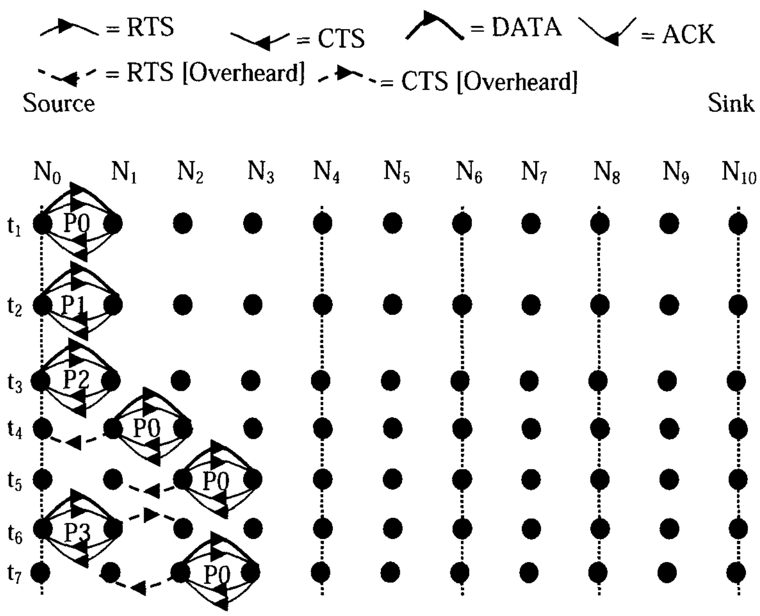

Figure 2.1: A possible data packet transfer pattern in S-MAC protocol

transfer pattern in S-MAC protocol. In this figure, let us assume that N_0 has four data packets to send to the sink node N_{10}. First, N_0 starts contending for the medium at the instant t_1 and wins contention. Thus, N_0 sends the first packet P0 to N_1. At t_2, both N_0 and N_1 contend for the medium. Now, if N_0 again wins the contention, then it sends the second packet P1 to N_1. However, from real-time point of view, it is desired that N_1 should have won the contention and should have forwarded P0 to N_2, instead of loosing contention to N_0 and receiving the next packet P1. As we see in this figure between time t_3 to t_7, it's totally unpredictable as to which node wins the contention. Second, irrespective of large carrier sense durations prior to transmission, collisions are still possible in S-MAC as two or more nodes in a range can initiate a transmission roughly at the same time after a carrier sense duration. Such unpredictable collisions cause unpredictable end-to-end delay for real-time WSN applications. Third, S-MAC protocol emphasizes primarily on minimizing energy consumption in WSNs, that is why the timing requirement assumes secondary importance.

S-MAC is used as a basis for the development of RT-MAC for several reasons. First, S-MAC is a widely referred general-purpose contention based energy-efficient MAC protocol for WSNs. In particular, Egea-lópez et al. in [20] and [21] present a soft real-time MAC layer protocol, called VTS, which is based on S-MAC protocol. Second, S-MAC supports data transmissions in a randomly deployed WSN between an arbitrary source and sink node pair. Third, S-MAC supports multi-streaming; thus, there could be several interfering or non-interfering data streams in the network at the same time. Fourth, as duty cycle doesn't change during run time in S-MAC, it is easier to define delay deadlines with some modification to this protocol.

The synchronization and periodic sleep/wake up strategies of RT-MAC are based on S-MAC. However, the major changes done in S-MAC protocol to develop RT-MAC protocol are given in Section 3.1 of the next chapter.

2.1.2 Overview of T-MAC protocol

Van dam, T., and Langend-oen, K. present T-MAC protocol for WSNs in [62]. The idea of adaptive duty cycle approach of T-MAC seems to be a good starting point to implement variable duty cycle during run time. In order to provide adaptive duty cycle, T-MAC uses a timeout duration prior to ending the listen mode. If any activation event such as overhearing RTS/CTS transmission or sensing collision etc. is detected by a node, then it will not go into sleep mode and will renew its timeout duration. T-MAC itself is not a good candidate for real-time communication for the following three main reasons. First, in T-MAC, every node tries to transmit all data packets from its transmit queue in burst at the start of a frame. In case of a collision, a node waits and listens for random amount of time within a fixed contention interval (no back-off scheme after a collision) before starting its transmission again. Thus, most of the time in unidirectional source to sink communication, a receiving node keeps on receiving data packets with fewer chances of winning the contention to forward packets which are already received. Consequently, transmit queue of the receiving node keeps on growing. As a result of this, all data packets travel together toward the sink in a bunch, which means that the first packet will reach to sink very late, but subsequent data packets will follow the first packet shortly. Therefore, most of the initial packets will miss their end-to-end delay deadlines particularly in a large multi-hop communication network. That makes T-MAC unsuitable for both event driven and periodic real-time applications. Second, T-MAC has an option of Futuristic RTS control packet (FRTS) to inform the third hop neighbor with respect to the sender node so that the third hop neighbor remains awake after two data transfer cycles, with an expectation that the second hop node might forward the packet to the third hop node. This reduces the sleep latency, consequently, increases the packet throughput. But, FRTS further decreases chances of receiving node to get medium access for forwarding, consequently, it will further help packets to arrive in bunches at the sink. Third, T-MAC has an optional full-buffer priority mechanism in which a node prefers sending a data packet than receiving when its transmit queue is almost full. Though full-buffer priority mechanism tries to reduce bunching process of data packets in transmission pattern, there are chances that a node might not receive/accept a new high priority data packet from other neighboring nodes. This is not acceptable in real-time applications. Thus, T-MAC is not suitable with or without FRTS and/or full-buffer priority mechanism for real-time applications in WSNs. Fourth, like S-MAC, T-MAC also gives emphasis on minimizing energy consumption rather than timing considerations. Halkes et al. in [26] gives a comprehensive comparison of S-MAC and T-MAC protocols.

2.1.3 Overview of D-MAC protocol

Lu et al. in [41] and [42] propose general purpose low sleep latency MAC protocol, called D-MAC, for unidirectional source to sink communication pattern in tree topology WSNs in which the sink node is placed at the top of a tree. D-MAC protocol is not suitable for real-time WSN applications for several reasons. First, a node at a depth from the sink node in the tree will have wake up time as offset multiplied by the depth. Thus, nodes have staggered sleep and wake up schedule, which means they wake up like a chain during a given frame. As a result of this, if a node dies, then it takes time to evolve to a new schedule as now the position of every node for wake up in a frame changes due to change of their depth with respect to sink. Second, as D-MAC is designed for the tree topology WSNs where all the nodes have sleep and listen schedule synchronized with the sink node. This makes synchronization difficult as the number of nodes increases in the system. Thus, D-MAC has scalability problem. Third, D-MAC is good for continuous reporting due to synchronized wake up schedule, but it's not good for event driven WSNs.

2.2 A comparative study of existing real-time MAC protocols for WSNs

In the literature, there are nine major real-time MAC protocols for WSNs. Table 2.1 at the end of this chapter presents a comparison of real-time and other related MAC layer protocols for wireless sensor networks. Here, we first explain real-time protocols based on deterministic scheduling, and then protocols based on contention.

2.2.1 Deterministic scheduling real-time MAC protocols

Egea-lópez et al. in [20] and [21] proposed virtual TDMA for sensors (VTS) protocol for soft real-time WSN applications. VTS is developed by introducing the TDMA concept in the S-MAC protocol. In VTS, a data packet can only travel one hop in a given TDMA slot (which is a frame in S-MAC and RT-MAC). Therefore, a data packet transmission cannot be accelerated by varying the duty cycle of a TDMA frame. Hence, it is more suitable for a periodic application, not necessarily for an event driven application. On the contrary, S-MAC and RT-MAC can adjust duty cycle to facilitate more than one hop transmission in a frame duration. In VTS, the number of slots in a TDMA frame equals to number of nodes in the transmission range. As any node gets a transmission slot again after m slots in a TDMA frame with m nodes, it limits the packet service rate in the source node. Precisely, the packets at the source are processed cyclically at the interval of mT_s, where T_s represents slot duration in a TDMA frame in VTS. Thus, it is evident that VTS cannot work for WSN applications which have higher packet generation rate. In general, though VTS gives timeliness guarantees, but these guarantees are too large, which makes it a slow protocol as compared to contention based protocols such as RT-MAC, S-MAC, and T-MAC with frame duration of these protocols equal to one TDMA slot of VTS. Additionally, if a source node is a border node of a virtual cluster of m number of nodes and it needs to forward data to its nearest neighbor in another virtual cluster, then in the worst case, the source may have to wait for *m* TDMA slots before its turn comes to initiate a packet transmission. This increases unpredictability of end-to-end delay in packet transmission. That makes VTS unsuitable for large multi-hop networks. In addition to this, being TDMA based protocol, VTS does not provide spatial channel reuse.

Chen et al. in [15] propose a path oriented real-time MAC layer protocol, called PR-MAC, for WSNs. This protocol is based on D-MAC [41] and [42]. Similar to D-MAC, PR-MAC is for tree based WSNs and has staggered sleep and wake up schedule with respect to the sink node. PR-MAC assumes multi-channel radio, while D-MAC, and RT-MAC are for single channel radio. PR-MAC has two normally ON (listen) durations in a frame (work cycle), while RT-MAC has only one listen duration in a frame. Consequently, PR-MAC facilitates bidirectional packet transfer, i.e., a data packet transmission from the source to the sink and a network control packet transmission from the sink to the source in the same frame. Thus, PR-MAC is more useful in scenario when there is one data packet to send from source to sink and one control packet to send from sink to source in every frame. Need of an end-to-end network control packet transmission per frame is rare in most of real-time WSN applications. Thus, the idle listening in nodes increases when there is no network control packet to send from sink to source in a frame. In contrast to PR-MAC, in RT-MAC, the end-to-end packet transfer can take one or more frames in a large multi-hop WSN. Similar to PR-MAC, RT-MAC also has bidirectional transmission support, which means that data packet transmission is possible from source to sink direction and control packet transmission is possible from sink to source direction. RT-MAC uses CC control packet that travels from sink to source direction to facilitate medium access for sensor nodes. This CC can carry additional control information, such as instruction for network wide change in duty cycle or sampling rate of events at the source, from the sink to the source without any extra overhead. Thus, CC takes numerically the same duration to reach the source as a data packet takes to reach to the sink node. Apart from this, data packet throughput of RT-MAC is much more than PR-MAC protocol with same frame duration. This is due to the fact that, for the same frame duration, there is only one end-to-end data transmission in PR-MAC, while there could be several data packets in network with each data transmission separated by at least four hops in RT-MAC. PR-MAC targets persistent applications, where communication path remains unchanged for quite some time; while RT-MAC does not have any such assumption. Theoretically, in RT-MAC, there could be just one data packet in a communication path between a source and sink pair, and the next packet can have another source and sink pair. PR-MAC is good only for periodic data collection, while RT-MAC is good for both periodic and event driven real-time WSN applications. In addition to this, the fault tolerance mechanism is needed in PR-MAC for some of potential deadlock scenarios. For example, a slight clock drift may cause collision between the source to sink data packet and the sink to source control packet. Such collisions are more likely at node that has both listen durations (i.e., source to sink listen schedule and sink to source listen schedule in a frame) side by side. If collision happens, then data transmission is interrupted till the next frame, and unless clock drift is corrected, collision will keep on happening. In another scenario, if a node goes out of sync, then it will not be able to listen to any packet from its neighbors. Thus, it will not be possible for the network to resolve such deadlocks unless out of sync node is put back in sync with its neighbors. RT-MAC does not have any of such deadlock scenarios as it is based on S-MAC protocol, which has adequate arrangement to bring any out of sync node again in sync with its neighbors.

Kim et al. [29] presented RRMAC, which is a TDMA based hard real-time MAC protocol for a multi-hop convergecast WSN. RRMAC's super frame is based on the IEEE 802.15.4 frame structure, which assigns time slots to sensor nodes in a hierarchical tree structure with the base station at the top of the tree. RRMAC needs special hardware to operate such as it needs two types of sensor nodes, one with a smaller and one with larger RF power levels. Being a tree based protocol, RRMAC needs global synchronization and has scalability issues due to constraint of superframe length.

Krohn et al. in [36] present a protocol called TOMAC that provides hard real-time message ordering at the MAC layer using nondestructive bit wise arbitration for one hop mesh network. This protocol is hard to generalize for multi-hop networks and other communication topologies.

Li et al. in [38] proposed a channel reuse-based smallest latest-start-time first (CR-SLF) algorithm for scheduling messages at the MAC layer to increase spatial channel reuse in soft real-time multi-hop WSNs. There are a few similarities between CR-SLF and RT-MAC protocol. For example, increased spatial channel reuse is a core feature of both protocols. Both protocols eliminate false blocking problem at MAC layer. Detail of false blocking avoidance in RT-MAC is given in Section 3.3 of Chapter 3. However, CR-SLF is developed for mobile wireless sensor networks such as a network of mobile robots with each robot having a wireless sensor device attached to it; while RT-MAC does not support mobility. CR-SLF uses centralized scheduling algorithm, in which a centralized scheduler decides as to when and who will transmit or receive messages, while RT-MAC presents a distributed real-time MAC layer protocol. Being a centralized algorithm, CR-SLF is not scalable, while RT-MAC is highly scalable protocol. CR-SLF needs up-to-date global position information of mobile wireless sensor nodes prior to scheduling or medium access decision, while RT-MAC needs relative location information for medium access decision.

Reference [22] presents a Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks (PEDAMACS). It is a hard real-time protocol based on TDMA scheme. It uses a high powered access point which accesses all the nodes in the network in one hop. It generates a centralized TDMA schedule for the sensor nodes to determine when a node should transmit and receive data. The requirement of a powerful access point makes it unsuitable for randomly deployed real-time WSNs.

2.2.2 Contention based real-time MAC protocols

Caccamo et al. in [10], and Caccamo and Zhang in [11] proposed a contention based implicit earliest deadline first (I-EDF) algorithm at the MAC layer for hard real-time WSN applications with periodic data transmission. It is a hard real-time MAC layer protocol for WSNs. It is not suitable for event driven WSNs. This protocol assumes cellular network structure. Nodes within a cell contend for a wireless medium by communicating among themselves, while cell to cell communications happen using more capable sensor nodes called routers. Cell to cell communications are done using frequency division multiplexing. Those specifications require special sensor hardware arrangements for I-EDF, while RT-MAC works with sensor nodes with same capabilities (i.e., no need of more capable router nodes). RT-MAC is designed for single channel radio, while I-EDF assumes multi-channel radio. In I-EDF, nodes in a cell have the same earliest deadline first schedule, while cell to cell communication is controlled by a global common schedule known to all router nodes. Thus, I-EDF needs synchronization on global basis, whereas RT-MAC needs relative position information of sensor nodes. I-EDF protocol assumes that each node in a cell knows about frequency, deadline, and duration of generated alarm messages from other nodes in the same cell, while RT-MAC doesn't assume such fore knowledge about data packets.

Watteyne et al. in [66], and Watteyne and Augé-blum in [67] presented a dual-mode real-time MAC protocol for hard real-time WSN applications. This protocol has two modes. In the protected mode, the message travels slowly but reliably, while in unprotected mode, the message travels with full speed but unreliably. Dual-mode real-time MAC protocol is based on I-EDF protocol, but it has fewer assumptions than I-EDF protocol. For example, unlike I-EDF, Dual-mode real-time MAC protocol does not need a router node to coordinate cell to cell communication as well as nodes have single channel radio with a predefined constant power. It is observed that RT-MAC and dual-mode real-time protocol are closely related in certain aspects. For example, both dual-mode real-time MAC protocol and RT-MAC are contention based MAC protocols for randomly deployed WSNs, and both protocols avoid collisions by stopping neighboring nodes from initiating a transmission for a certain duration. However, there are many differences in both the protocols. For example, Dual-mode real-time MAC protocol relies heavily on global synchronization mechanism. It requires that all nodes need to know their absolute position information and accurate distance between them and their closest neighbors. This calls for special hardware requirements such as a global positioning system (GPS), which is a demanding assumption for a randomly deployed sensor network. In contrast to this, RT-MAC regulates medium access based on relative position of sensor nodes. Dual-mode real-time MAC protocol deals with multi-source to one sink scenario, whereas RT-MAC deals with multi-source and multi-sink scenario. If packet generation rate increases in Dual-mode real-time MAC protocol, then the protocol increases the chances of missing delay deadline and packet drop rate in the network. In this case, providing hard real-time guaranties even in protected mode of Dual-mode real-time MAC protocol is not possible. In contrast to this, RT-MAC protocol controls the packet transmission rate using feedback approach, which prevent network from being overwhelmed by too many packets in a neighborhood that can cause collisions. However, dual-mode real-time MAC protocol uses a reservation mechanism to avoid collisions. It reserves five cells (towards sink) ahead of a sending node to stop them from generating any new packet during reservation duration and to make them ready for forwarding packet sent by earlier nodes. In general, reservation based real-time MAC approaches are more unpredictable because the control packet sent for reservation may not find the next few hop neighbors free, which will make that particular reservation attempt unsuccessful. In contrast to this, RT-MAC uses a predictable feedback based approach to avoid collisions. RT-MAC starts sending the first data packet adventurously. Transmission of subsequent data packets depends upon successful progression of the previous data packets towards the destination. Thus, RT-MAC follows the philosophy of transmission first then clearance to other, while Dual-mode real-time MAC protocol follows the philosophy of reservation first then transmission. Additionally, in Dual-mode real-time MAC protocol, there are more than one node per cell that can reach any node in the next cell. Thus, essentially its one hop communication from one cell to another. Therefore, due to five cells reservation mechanism, $5n$ number of nodes will not be able to generate and transmit packets during whole reservation duration, though they can forward a packet sent by earlier nodes. Here, n is the average number of nodes per cell. This causes poor spatial channel reuse in Dual-mode real-time MAC protocol. In contrast to this, RT-MAC stops transmission for four hops only due to feedback mechanism, which means that only four nodes are stopped from transmitting a packet. This facilitates better spatial channel reuse in RT-MAC as compared to Dual-mode real-time MAC protocol. Dual-mode real-time MAC protocol relies heavily on a global synchronization mechanism as well as the nodes' absolute position information, whereas RT-MAC regulates medium access based on the relative position of sensor nodes. Dual-mode real-time MAC protocol is yet to implement the fault tolerance mechanism due to message loss or node loss that makes harder to fulfill end-to-end delay deadlines in hard real-time applications. Dual-mode real-time MAC protocol does not support operation with duty cycle, whereas RT-MAC supports both duty cycle and non duty cycle operations.

The IEEE 802.15.4 [77] standard provides medium access control specification for small devices that consume low power and require lower data rates. This standard provides bitrates of 20kbps, 40kbps, and 250kbps in the 868MHz, 915MHz, and 2.45GHz frequency bands, respectively. This standard facilitates communication in star topology as well as peer-to-peer topology. In IEEE 802.15.4 personal area network (PAN), a single device acts as a PAN coordinator that controls device association within the network. In the star topology, the PAN coordinator is directly responsible for all communication and resource reservations. However, in the peer-to-peer topology, devices operate independently, and need not communicate through the PAN coordinator, but all devices must associate with the PAN coordinator prior to participating in the network. The IEEE 802.15.4 standard focuses primarily on the star topology, and does not clearly define functionality of the peer-to-peer networks. In IEEE 802.15.4 network, nodes operate in beacon-enabled mode or in an unsynchronized mode without beacons. A beacon enabled PAN coordinator utilizes the synchronization provided by the beacon to perform slotted channel access, whereas a PAN coordinator without beacons uses unslotted access. IEEE 802.15.4 uses a slightly modified CSMA/CA algorithm to access the wireless channel.

Though IEEE 802.15.4 focuses on applications that are similar to WSN applications, there are several disadvantages of using IEEE 802.15.4 for WSN applications [35]. For example, IEEE 8.2.15.4 does not clearly define the operation of nodes in a peer-to-peer topology. In addition to this, as sensor nodes in a WSN are generally spread in a large geographical area, the use of a PAN coordinator or multiple PAN coordinators becomes an issue as it is not clearly defined in this standard. In this context, the Zigbee Alliance [80] attempts to define the upper layer protocols (on top of IEEE 802.15.4) that outline standards for some of these operations.

Although IEEE 802.15.4 protocol has provided guaranteed time slot (GTS) mechanism for time critical data, the details of how to use it to support explicit QoS guarantees in real-time WSNs are still developing [31] and [40]. Francomme et al. in [23] show that a PAN coordinator can distribute GTSs corresponding to deadline and bandwidth requirements of transmissions to support hard real-time guarantees. However, Koubâa et al. in [33] and [34] show that an enhanced IEEE 802.15.4 slotted CSMA/CA mechanisms can offer delay guarantees in soft real-time WSN applications. In this paper, the traffics are categorized into high and low priority queues that employ different CSMA/CA settings. It presents a heuristic solution to different quality of service (QoS) priorities of messages.

2.3 Overview of Markov Analysis

A Markov analysis looks at a sequence of random events, and analyzes the tendency of one event to be followed by another. It is useful for analyzing dependent random events whose likelihood depends on what happened last [80]. In this context, a Markov process is a random process where the value of a random variable at instant n depends only on its immediate past value at instant n-1. Here, the random variable represents the state of the system at a given instant n. The following are examples of Markov processes [25] and [43]:

1. The state of the daily weather.
2. Telecommunication protocols and hardware systems.
3. Arrival of cars at an intersection.
4. Machine breakdown and repair during use.
5. Customer arrivals and departures.

2.3.1 Markov Chains

A Markov process is called a Markov chain if the state space of the Markov process is discrete. In this case, the states can be denoted by the integers 0, 1, 2, etc.. Discrete time Markov chains (DTMC) arise naturally in many communication systems. A Markov Chain can be homogeneous or non-homogeneous. A homogeneous Markov Chain has constant transition rates between the states, whereas a non-homogeneous Markov Chain has the transition rates between the states that vary with time, e.g., due to the change in battery status of sensor nodes.

A Markov chain stays in a particular state for a certain amount of time, called the hold time and it moves randomly to another state at the end of the hold time. Markov chains can be broadly classified in two categories based on the criterion used to measure the hold time.

a) Discrete-time Markov chain: In a discrete-time Markov chain the hold time assumes discrete values. Therefore, the changes in the states occur at discrete time values such as $t = T0, T1, T2$, etc. In general, the spacing between the time steps need not be equal. However, often the discrete time values are equally spaced, therefore, it can be written as $t = nT$, where $n = 0, 1, 2$, etc.. The time step value T depends on the type of system.

b) Continuous-time Markov chain: The hold time assumes continuous values in a continuous-time Markov chain. Therefore, the changes in the states occur at any time, which makes time values continuous over a finite or infinite interval.

In a DTMC, the value of a random variable $S(n)$ represents the state of the system at time n. $S(n)$ is a function of its immediate past value. For example, $S(n)$ depends on $S(n\text{-}1)$. This is referred to as the memoryless property of the Markov chain, where the present state of the system depends only on its immediate past state [65] and [68]. Alternatively, the memoryless property of the Markov chain implies that the future state of the system depends only on the present state and not on its past states [60].

A Markov chain is said to be irreducible if it is possible to get to any state from any other state in the system.

2.3.2 Markov Chain Transition Matrix

$p_{ij}(n)$ represent the probability of being a system in the state i at time n given that the past state was the state j at time n-1. $p_{ij}(n)$ can be equated to the conditional probability and can be expressed mathematically as

$$p_{ij}(n) = P[S(n) = i \mid S(n-1) = j]. \tag{1}$$

If the transition probabilities are independent of the time, then we have a homogeneous Markov chain. In this case, Equation (1) can be written as

$$p_{ij} = P[S(n) = i \mid S(n-1) = j]. \tag{2}$$

Now, the probability of finding the system in state i at time n can be defined as

$$s_i(n) = P[X(n) = i]. \tag{3}$$

Further, using the Equation (1), the state probability s_i can be written as

$$s_i(n) = \sum_{j=1}^{m} p_{ij} s_j(n-1), \tag{4}$$

where, m is the number of possible states in the system, and the indices i and j are in the range $1 \le i \le m$ and $1 \le j \le m$.

Equation (4) can be expressed in the matrix form as

$$s(n) = Ps(n-1), \tag{5}$$

where, P is the state transition matrix of dimension m x m

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1m} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mm} \end{pmatrix}, \tag{6}$$

and $s(n)$ is the state vector, which is defined as the probability of the system being in each state at time step n. It is represented as

$$s(n) = \begin{bmatrix} s_1(n) & s_2(n) & \ldots & s_m(n) \end{bmatrix}^T. \tag{7}$$

The component $s_i(n)$ of the state vector $s(n)$ at time n indicates the probability of finding the system in the state s_i at time n. As state probability vector $s(n)$ in Equation (7) includes state probabilities for all possible m states in the system, we have the following equation:

$$\sum_{i=1}^{m} s_i(n) = 1 \qquad \text{for } n = 0, 1, 2 \ldots \tag{8}$$

State transition matrix P in Equation (6) gives a great insight about the behavior of a Markov chain. As all the elements of P represent state transition probabilities, it is a real and nonnegative matrix. Since, the columns of P matrix represent transitions out of a given state, the sum of each column is one as it includes all the possible transitions out of the state. Thus, for all values of j, we have

$$\sum_{i=1}^{m} p_{ij} = 1, \tag{9}$$

where, p_{ij} represents the transition probability from state j to state i. The columns of P matrix represent the present state, whereas the rows represent the next state of the system. By solving linear Equations (5) and (8), we can find steady state probabilities of the system. An alternative approach of solving Equations (5) and (8) by using Eigenvectors and Eigen values is given in [53] and [61].

2.4 Overview of OMNeT++ simulator

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework [63], [64], [79] and [81]. It stands for Objective Modular Network Testbed in C++. It is an object-oriented discrete event network simulation tool. It is useful for protocol modeling, computer networks and traffic modeling, modeling queuing systems, validating hardware architectures, and modeling multi-processors and other distributed systems. In general, any system in which the discrete event approach is suitable, can be modeled using OMNeT++. OMNeT++ works well on Linux and Windows platforms.

Component of OMNeT++

1. Simulation kernel library.
2. Compiler for the NED topology description language (nedc).
3. Graphical network editor for NED files (GNED).
4. GUI for simulation execution (Tkenv).
5. Command-line user interface for simulation execution (Cmdenv).
6. Graphical output vector plotting tool (Plove).
7. Utilities such as random number seed generation tool, makefile creation tools.
8. Documentation, sample simulations, contributed material, etc..

The basic entity in OMNeT++ is a module, which can be composed of submodules or atomic. An OMNeT++ model contains hierarchically nested modules. Modules communicate by passing messages to each other through gates. Gates are linked to each other using connections. Propagation delay, error rate and data rate can be assigned to a connection. Gates in OMNeT++ support only one-to-one communication. A model network in OMNeT++ consists of nodes connected by links. The nodes represent blocks, entities, modules, etc., whereas the links represent channels, connections, etc.. The structure of nodes in a network interconnected together is called topology.

OMNeT++ uses NED language, which is quite user friendly. It has a human-readable textual topology. NED files can be created using the GNED graphical editor or any other text editor.

Modular description of a network is given in NED language. A network description can contain import statements, channel definitions, simple and compound module declarations, and network definitions. The simple and compound modules of one network description can be reused in another network description.

Messages are used to convey information from a module in a simulation to itself in the future, or to other modules. Messages can represent packets or frames in a computer network, and jobs or customers in a queuing network etc. in an actual simulation. Various message types can be defined in OMNeT++. A message definition can contain complex data structures. Message definitions are translated into C++ classes during compilation process.

User interfaces

OMNeT++ 's design allows access to the internal workings of the model. It also allows the user to initiate and to terminate simulations, as well as to change variables inside a simulation model. These features increase flexibility during the development and debugging phase of the modules. The User interface of OMNeT++ is used with the simulation execution. The interaction of the user interface and the simulation kernel is done through well defined interfaces. It is possible to implement several types of user interfaces without changing the simulation kernel. Additionally, a simulation model can run under different interfaces without modifying the source code of the simulation. In general, the user can test and debug the simulation with a graphical user interface, and finally run the simulation with a fast command-line user interface that supports batch execution.

OMNeT++ has two user interfaces.

a) Tkenv: Tk-based graphical user interface.
b) Cmdenv: command-line user interface for batch execution.

Tkenv is a portable graphical user interface, which is used for tracing, debugging, and simulation execution. It provides a detailed view of the state of the simulation at any point during the execution. Thus, Tkenv is very useful in the development stage of a simulation or for presentations. Tkenv can be used with the GNU debugger (gdb) to provide a good environment for experimenting with the model, and the verification of the correct operation during the execuation of the program. It is possible to display simulation results during execution.

Tkenv has the following features.

1. Event-by-event execution.
2. Breakpoint insertion.
3. Animated message flow.
4. Animated execution.
5. Inspector windows to examine the objects and variables.
6. A separate window for each module's text output.
7. Graphical display of simulation results during execution.
8. Restarting a simulation.

Cmdenv is a fast and portable command line interface. It is designed primarily for batch execution. It compiles and runs on all platforms. Cmdenv executes simulation runs described in a configuration file. If one run stops with an error message, then subsequent ones will still be executed. The simulation runs to be executed can be passed to Cmdenv via command-line argument or through an "ini" file.

Cmdenv can be executed in the following two modes.

a) Normal mode: It is useful for debugging in textual format. In this mode, detailed information such as event banners and module's output etc. is written to the standard output or a file.

b) Express mode: This mode is useful for long simulation runs. In this mode, only periodical status update is displayed about the progress of the simulation.

2.5 Summary of the chapter

This chapter presents a study of existing real-time MAC protocols for WSNs. Table 2.1 summaries the comparative study of real-time and other related MAC layer protocols for WSNs. This chapter also provides a brief overview of Markov analysis that is pertinent to RT-MAC protocol presented in this thesis. An overview of OMNeT++ network simulator is also presented in this chapter. It is used for the simulation study of the protocol presented in this thesis.

The next chapter presents RT-MAC protocol in single-stream scenario.

Table 2.1: Comparison of real-time and other related MAC layer protocols for wireless sensor networks.

Protocol	Citations	Nature of protocol	Medium access strategy	Collision avoidance strategy	Type of comm. at MAC	Need of special hardware	Fault tolerance	Location information needed	Network topology	Support for sleep - listen schedule / Varying duty cycle at run time	Mobility support	Number of packets handled at MAC of a node at a time	Remark
RT-MAC	[55], [56], [57], [58]	Soft Real-time	Contention based	Feedback mechanism	Single channel	None	Adequate, Nodes will never go to deadlock states	Relative location information	Random multi-hop network, Multi source & multi sink, Single-strm & multi-streams	Yes/Yes	No	One	Based on S-MAC
VTS	[20], [21]	Soft Real-time	Deterministic, TDMA based	TDMA scheduling	Single channel	None	Adequate	Relative location information	Random network, Multi source & multi sink	Yes/Yes	No	One	Based on S-MAC
RRMAC	[28]	Hard Real-time	Deterministic, TDMA based	Hierarchical TDMA scheduling	Single channel	Sensor nodes with two power levels	Adequate	Relative location information	Hierarchical tree structure	Yes/Yes	No	One	Based on IEEE 802.15.4
Dual-mode real-time MAC protocol	[64], [67]	Hard Real-time	Contention based	Reservation mechanism	Single channel	GPS or some other mechanism for global positioning information	Not good, Message loss or node loss cases not addressed	Absolute location information	Random network, Comm. along a line, Multi source & one sink	No/No	No	One	Based on I-EDF
I-EDF	[4], [11]	Hard Real-time	Contention based	Same EDF schedule for intra-cell comm. & FDMA for inter-cell comm.	Multi channel	Special router nodes & multiple frequency transceivers	Not good, failure of router node will collapse network	Relative location information	Hexagonal cell based network. Multi source & one sink	No/No	No	One	TDMA based global sync schedule for routers
TOMAC	[36]	Hard Real-time	Scheduling based	message ordering using non destructive bit wise arbitration	Single channel	None	Adequate	Relative location information	Single hop mesh network.	Yes/No	No	One	-

PR-MAC	[15]	Soft Real-time	Scheduling based	Staggered sleep and wake up schedule	Multi-channel	None	Not good, Packet loss & node loss not addressed	Relative location information	Tree topology network, Multi source & one sink.	Yes/No	No	Many	Based on D-MAC
CR-SLF	[38]	Soft Real-time	Scheduling based	centralized scheduling algorithm	Single channel	Mobile nodes, Central scheduler	Not good, Validity of schedule is velocity dependent.	Absolute location information	Mobile sensor network	No/No	Yes	One	-
PEDAMACS	[22]	Hard Real-time	Scheduling based	centralized scheduling algorithm	Single channel	Powerful access point	Adequate	Relative location information	Any topology, but managed by a access point	No/No	No	One	TDMA based global sync schedule
S-MAC	[71], [72]	General purpose	Contention based	Large carrier sense	Single channel	None	Adequate	Relative location information	Random multi-hop network, Multi source & multi sink.	Yes/No	No	One	-
T-MAC	[62]	General purpose	Contention based	Fix contention interval for RTS (tuned for maximum load)	Single channel	None	Adequate	Relative location information	Random multi-hop network, Multi source & multi sink	Yes/Yes	No	One	-
D-MAC	[41], [42]	General purpose	Scheduling based	Staggered sleep and wake up schedule	Single channel	None	Not good, Packet loss & node loss not addressed	Relative location information	Tree topology network, Multi source & one sink.	Yes/No	No	One	-

Chapter 3: RT-MAC Protocol in Single-Stream Scenario

This chapter presents the RT-MAC protocol in a single-stream scenario. Section 3.1 presents the analysis of challenges involved in the design of RT-MAC protocol. Section 3.2 presents assumptions and description of RT-MAC protocol design. Section 3.3 presents the detailed operation of the protocol. Collision avoidance strategies in the RT-MAC are presented in Section 3.4.

3.1 Problem Analysis

S-MAC is selected for development of RT-MAC primarily because it supports data transmissions between an arbitrary source and a sink node pair in a randomly deployed WSN. As mentioned earlier in Subsection 2.1.1, S-MAC is not suitable for real-time WSN applications since carrier sense, contention, and back-off process prior to medium access make it unpredictable as to which node would finally get medium access first. The removal of this unpredictability of S-MAC is important in providing the end-to-end delay guarantees in real-time communication. Thus, the first major change made in S-MAC to enable real-time operation in RT-MAC is the introduction of a feedback mechanism in the medium access strategy with a control signal, called Clear Channel (CC). In addition to CC, three more control signals are added to S-MAC to provide fault tolerance. These are CC Acknowledgment (CCACK), CC Query (CCQ), and CCQ Reply (CCQR). However, CC is the most frequently used control signal in RT-MAC. Figure 3.1 shows the frame structure of S-MAC protocol, which needs large contention duration to resolve the medium access problem. However, the feedback mechanism used in RT-MAC does not need a large carrier sense duration for the contention resolution. Hence, the second

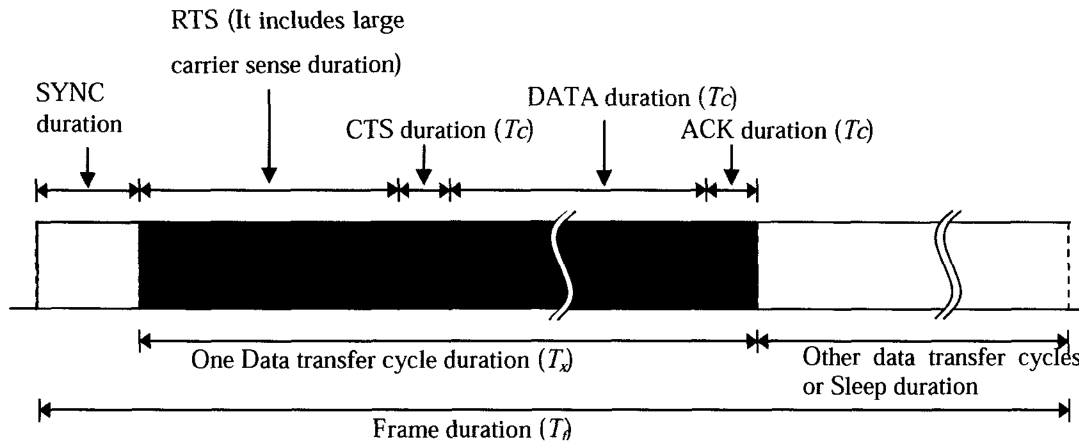

Figure 3.1: Frame structure in S-MAC

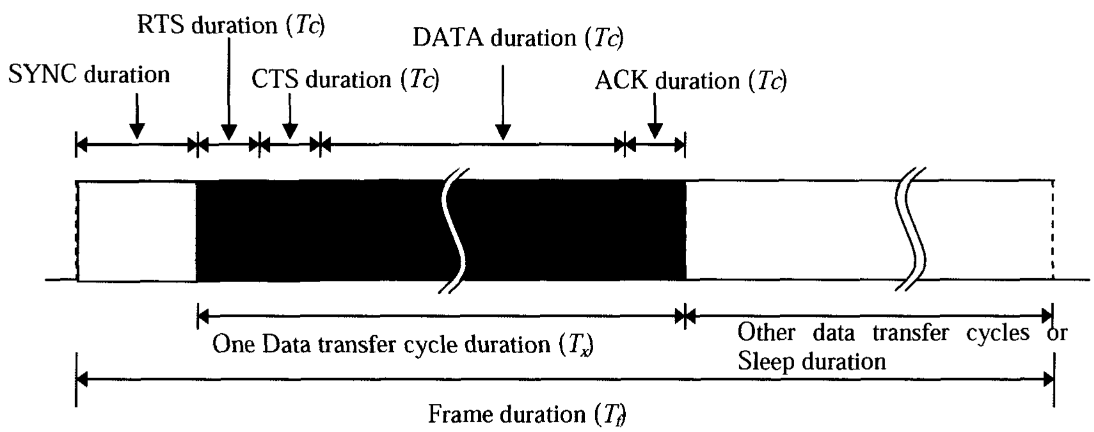

Figure 3.2: Frame structure in RT-MAC protocols

major change made in S-MAC is the removal of large carrier sense duration (which is typically 3 to 5 times larger than the duration of a typical control packet) prior to request to send (RTS) transmission. This modification reduces the end-to-end delay significantly since carrier sense in S-MAC needs the carrier sense duration prior to every data transmission regardless of whether the transmission is successful or not. All control packets in RT-MAC (including RTS) have roughly the same duration.

Figure 3.2 shows the frame structure in RT-MAC protocol. The third change made in S-MAC is about duty cycle, which is made variable during run time in RT-MAC. The variable duty cycle ensures to accommodate variations in load in WSNs. Only adaptive

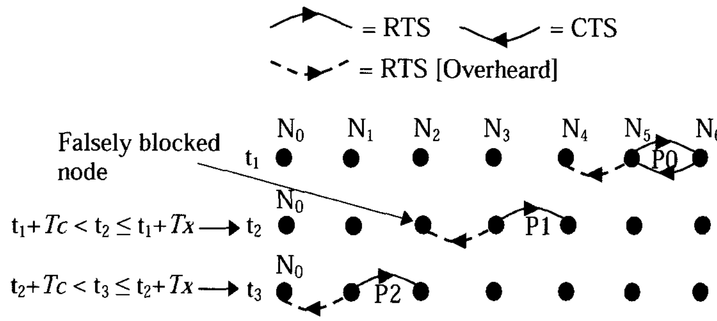

Figure 3.3: False blocking in single-stream scenario in RTS/CTS handshake based wireless MAC protocols.

listening mode of S-MAC is used for development of RT-MAC as it reduces end-to-end delay in data transmission particularly at low duty cycle mode of protocol.

Contention based MAC protocols for WSNs use RTS/CTS handshake mechanism to avoid hidden node problem [32] and [50]. However, Ray et al. in [49] and Ray and Starobinski in [51] suggested that RTS/CTS exchange mechanism potentially and falsely blocks the nodes which otherwise may have simultaneous collision free transmissions. The false blocking problem occurs due to unsuccessful attempts of RTS transmission by the neighboring nodes. This problem does not only reduce the network throughput, but also results in inconsistent and unpredictable data packet transmission pattern, which in turn makes impossible to define delay deadlines for real-time WSN applications.

Figure 3.3 explains the false blocking problem in a single-stream communication in WSNs. At the instant t_1, RTS control packet is sent from N_5 to N_6 to initiate transmission of a data packet P0. This RTS is overheard by N_4. Consequently, N_4 goes into sleep mode for duration of one data transfer cycle. Meantime, N_3 sends RTS (for data packet P1) to N_4 at the instant t_2 with $t_1+Tc < t_2 \leq t_1+Tx$, where Tx and Tc represent durations for data and control packet transmissions by one hop, respectively. As N_4 is in sleep mode, therefore, it will not respond to RTS request from N_3. However, this RTS request from N_3 is overheard by N_2 that goes to sleep mode for one data transfer duration, irrespective of the fact that there will not be any data transfer between N_3 and N_4 as N_4 is in sleep mode. Thus, in this scenario, N_2 is the falsely blocked node. Meantime, if N_1 sends RTS (for data packet P2) to N_2 at the instant t_3 with $t_2+Tc < t_3 \leq t_2+Tx$, then N_1 will not get any reply as N_2 is sleeping in this duration.

References [49] and [51] propose an RTS validation approach to deal with the false blocking problem. The RTS validation approach minimizes false blocking, thus it increases throughput of the network. However, the data transmission pattern still remains inconsistent and unpredictable in this approach. Li et al. in [38] and [39] propose centralized scheduling algorithms at MAC layer to avoid false blocking problem in soft real-time mobile multi-hop WSNs. The proposed centralized scheduler needs up-to-date global positioning information of sensor nodes. However, RT-MAC, presented in this thesis, removes false blocking problem completely. Thus, not only RT-MAC improves network throughput performance, but also it provides consistent and predictable data transmission pattern, which make the protocol suitable for real-time WSN applications. The RT-MAC deals with false blocking problem in a distributed manner. Additionally, by removing false blocking problem, the RT-MAC ensures maximum possible spatial channel reutilization by guarantying simultaneous collision free data packet transmissions if they are separated by four hops. This is another reason for lower end-to-end delay in RT-MAC. Exact mechanism behind the false blocking avoidance in RT-MAC is discussed in Section 3.3.

3.2 Protocol description

RT-MAC is designed for single channel radio sensor nodes with similar transmission power. It is designed for static sensor nodes. Any node can be a source or sink node in the WSN. RT-MAC does not need any global knowledge of positioning of wireless sensor nodes. Medium access decisions are made on relative hop distances. RT-MAC is a distributed and scalable real-time MAC protocol. Scalablity is provided because medium access decisions in RT-MAC only requires a maximum of six hops. The feedback based medium access strategy of RT-MAC facilitates controlled flow of data packets. Thus, the data packet transmission pattern is consistent and ordered delivery of data packets to the destination is guaranteed. Those increase the predictability of the end-to-end delay. Additionally, maximum spatial channel reuse and reduced carrier sense duration help to minimize the end-to-end delay in RT-MAC. There are no collisions between the data packets in RT-MAC due to its feedback mechanism. However, there are some rare

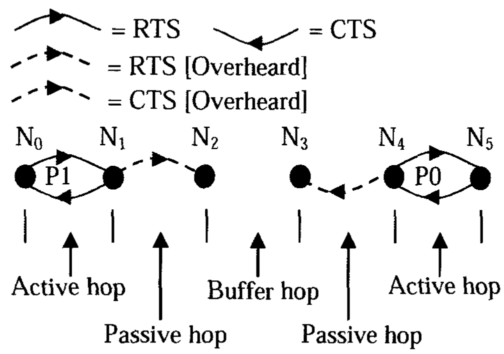

Figure 3.4: Communication pattern in RT-MAC

chances of collision between a data packet and a CC control packet. RT-MAC implements some strategies to avoid such collisions, as discussed in Section 3.4. RT-MAC is a load balanced protocol wherein the number of transmissions remains stable throughout a data packet stream.

3.3 Functioning of Protocol

Figure 3.4 shows the communication pattern of a data packet stream in RT-MAC. N_0 node initiates a data packet (P1 in this case) transmission with its one hop neighbor N_1 by sending the RTS control packet. N_1 responds by sending CTS to N_0; however, this CTS is also overheard by N_2. Thus, there is an active exchange of RTS, CTS, DATA and ACK between N_0 and N_1. Therefore, it is called an active hop. However, the next hop between N_1 and N_2 overhears communication between N_0 and N_1 passively; hence, it is called a passive hop. Now, if we leave one hop between N_2 and N_3 as a buffer hop, then there is a possibility of another data packet (P0 in this case) transmission between the N_4 and N_5 nodes that will not interfere with P1's transmission. Therefore, the buffer hop ensures simultaneous collision free transmission of P0 and P1. This active-passive-buffer-passive-active hop pattern is repeated throughout a communication stream that facilitates several simultaneous data packet transmissions from the nodes separated from each other by at least four hops (i.e., N_0 and N_4 in Figure 3.4). This communication pattern also prevents false blocking of any node, which is primarily caused by unsuccessful transmissions that are not four hops apart. Thus, RT-MAC ensures maximum possible spatial channel reutilization.

There are three variables that control functioning of RT-MAC protocol.

a) Clear Channel Flag (CCF): Each node has a CCF Boolean variable. The idea behind CCF is to let a node know whether it is in receive or transmit, or in receive only modes. If CCF is 1, then the node can transmit or receive a data packet, whereas if CCF is 0, then the node can only receive data packets. A node can transmit or receive a CC control packet irrespective of its CCF value. Initially, all nodes have a CCF value of 1. The CC control packet is used to toggle an appropriate CCF value of the nodes.

b) Clear Channel Counter (CCC): Every CC control packet has a CCC integer variable whose value ranges between 0 and 3. The value of CCC is 3 at the originating node of CC and is decreased by one with one hop transmission of CC. CC is always transmitted from sink to source direction. Thus, CCC is used to set the appropriate CCF value of a node. If the node has a CC control packet with its CCC as 2 or 3, then the CCF value of that node will be "0". However, if a node receives a CC with CCC as 0 or 1, then the CCF value of that node will be switched to "1", which enables node to initiate a data packet transmission.

c) Hop Counter (HC): Each data packet has an HC integer variable. As the data packet travels, its HC value varies from 4 to 0 for the first four hops of a communication stream and from 2 to 0 for all subsequent two hop segments of the communication stream. If a node receives a data packet with an HC value of 1, then it can initiate a CC transmission. Once a node receives a data packet with an HC value of 0, then it resets the value of HC to 2 again and waits for $2Tc$ duration prior to forwarding the data packet, where Tc represents control packet duration.

The operation of RT-MAC is illustrated by Figures 3.5 to 3.8. Figure 3.5 shows the data and CC packet transmission pattern in a continuously ON mode of operation (i.e., without duty cycle). Figures 3.6 and 3.7 show timing diagram of packet transfer in RT-MAC without duty cycle mode of operation, and Figure 3.8 shows the same with duty cycle mode of operation. In the duty cycle mode of operation, RT-MAC follows the frame synchronization mechanism of S-MAC protocol. Thus, it keeps some duration reserved at the beginning of each frame for synchronization as shown in Figure 3.8. In Figure 3.6, N_0

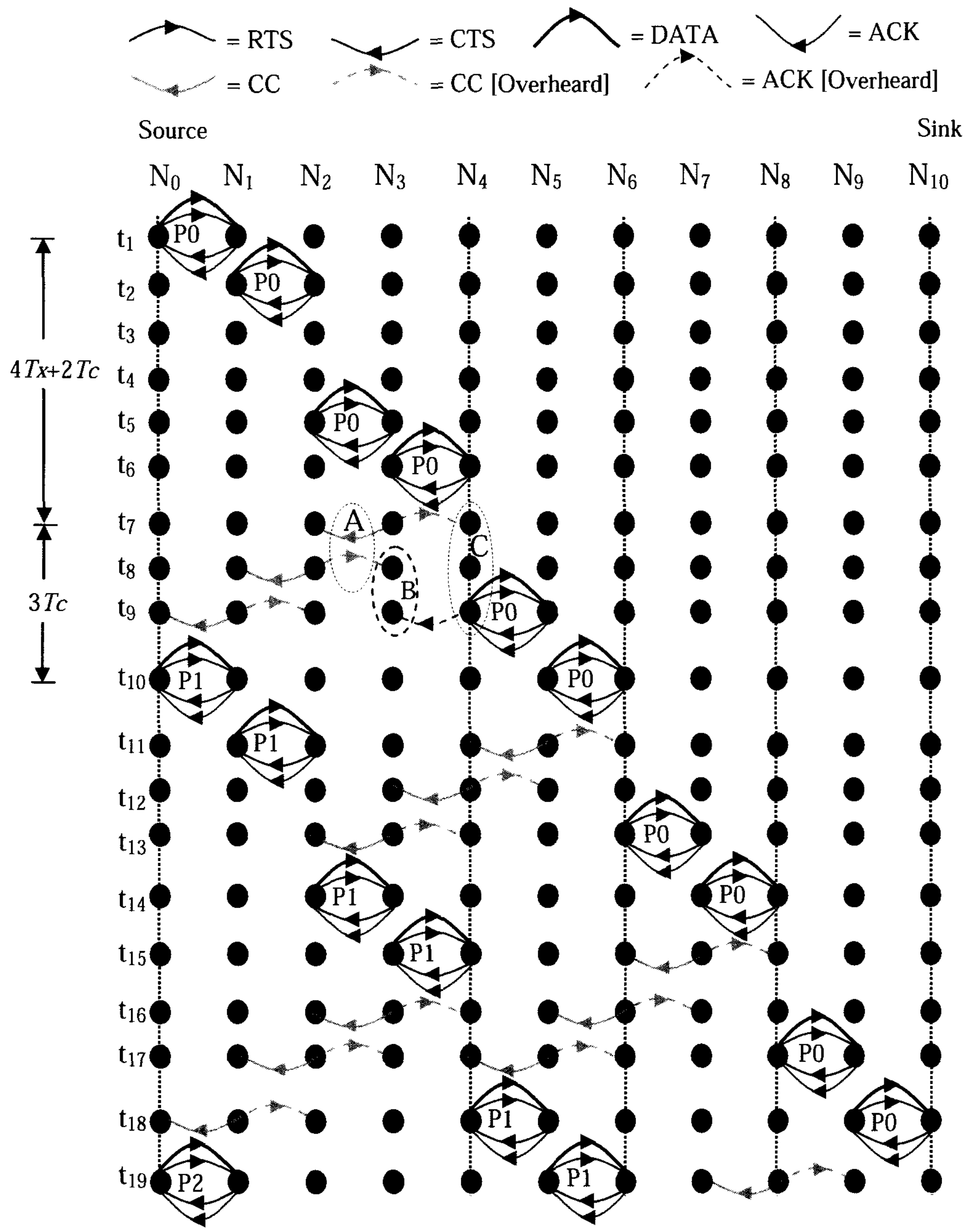

Figure 3.5: Data packet and Clear Channel (CC) transmission pattern with feedback approach in single-stream scenario in the continuous ON mode of sensor nodes (time scale is not linear).

is a source node and N_{10} is a sink node.

At first, N_0 has several data packets to send to N_{10}. Now, N_0 sends RTS to N_1. N_1 responds with CTS. Then, N_0 sends data packet P0 to N_1. After receiving P0, N_1 sends ACK to N_0. This completes one data transfer cycle by one hop. As shown in Figure 3.6, the duration of one data transfer cycle is denoted as Tx and the duration of one control

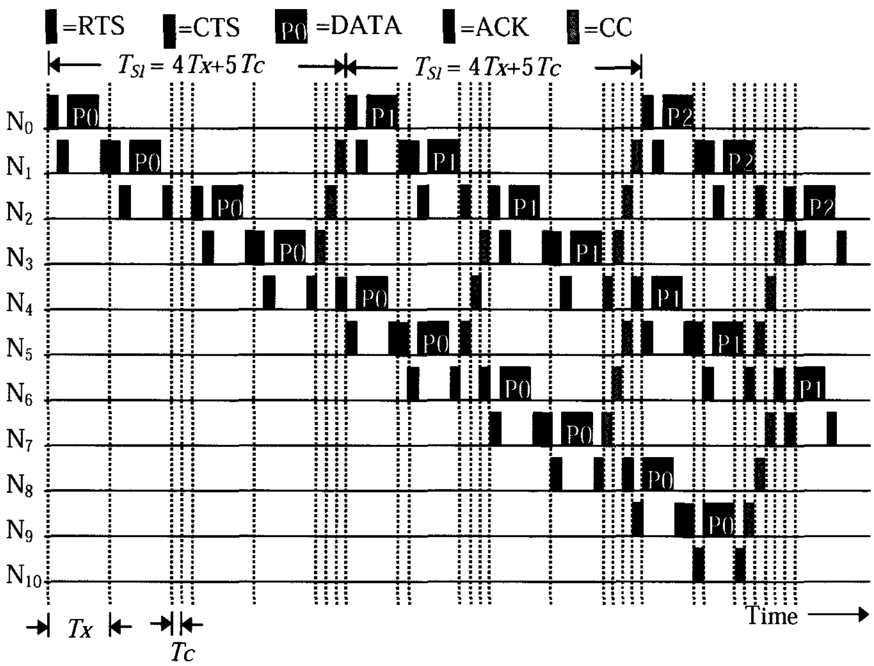

Figure 3.6: Timing diagram of packet transfer in RT-MAC protocol without duty cycle for $T_{AI} \leq 4Tx+5Tc$

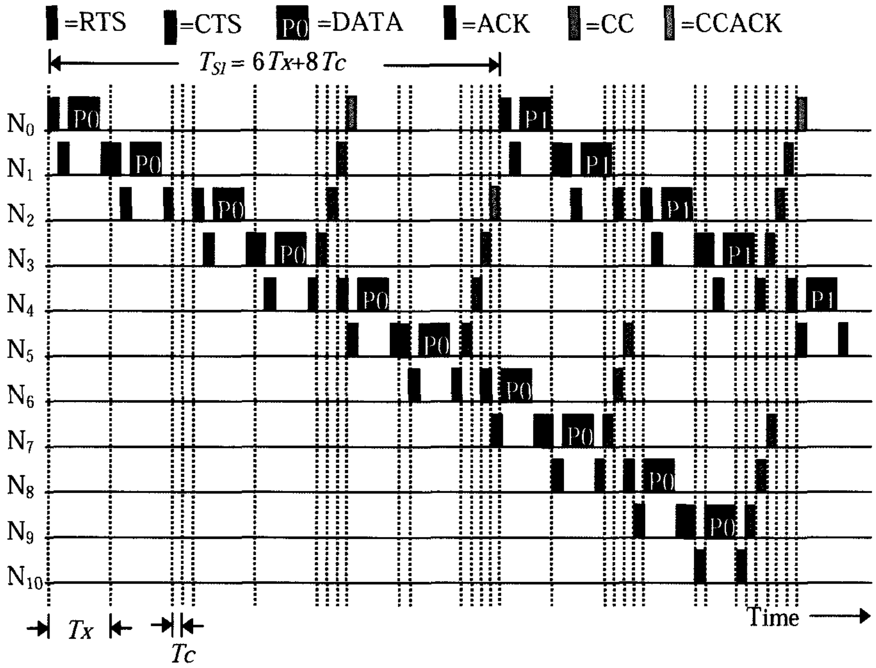

Figure 3.7: Timing diagram of packet transfer in RT-MAC protocol without duty cycle for $T_{AI} = 6Tx+8Tc$

packet is denoted as *Tc*. After getting ACK, N_0 sets its CCF value to 0; hence, it cannot transmit the next packet unless its CCF value is set to 1 again. Similarly, in the next three

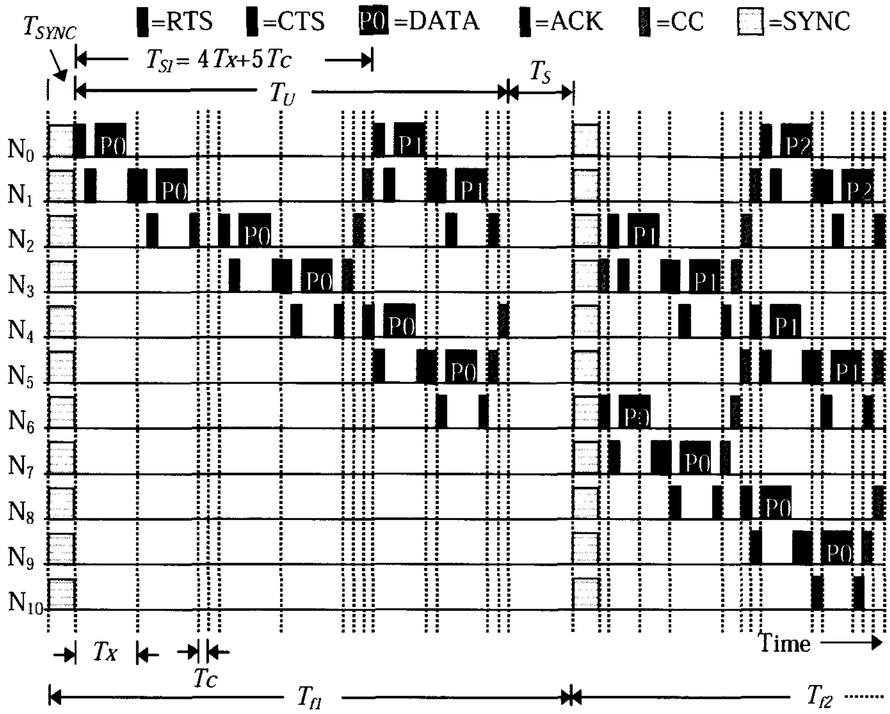

Figure 3.8: Timing diagram of packet transfer in RT-MAC with duty cycle for $T_{AI} \leq 4Tx{+}5Tc$

Tx durations, P0 is transmitted up to N_4 and CCF of N_1, N_2 and N_3 becomes 0 as shown in Table 3.1. At N_0, the value of HC of P0 was 4 and it is decreased by one each time P0 is transmitted successfully by one hop. Once P0 reaches N_4, its HC becomes 0. Then, N_4 sets HC of P0 to 2, which will become 0 again after the next 2 hops transmission. It is observed that HC of a data packet becomes 0 only at the even numbered nodes with respect to the source node. Therefore, their immediate previous hop nodes, which are odd numbered nodes with respect to the source, will initiate CC transmissions. From this point onward, even and odd numbered nodes with respect to the source node will be referred to as even numbered and odd numbered nodes, respectively.

Now, after successful transmission of ACK to N_3, N_4 waits for $2Tc$ duration prior to forwarding P0 to N_5. This waiting period prevents collision between CC originating from N_3 and data packet transmission originating from N_4. For example, if N_4 initiates an RTS transmission during the first *Tc* after sending ACK to N_3, then this RTS will collide with the CC transmission from N_3 to N_2. Additionally, even if N_4 tries to transmit RTS in the second *Tc* duration after sending ACK to N_3, then N_3 will experience a collision with

Table 3.1: RT-MAC Protocol variable values

Time	N_0	N_1	N_2	N_3	N_4	N_5	N_6
0	CCF=1 HC=4	CCF=1	CCF=1	CCF=1	CCF=1	CCF=1	CCF=1
Tx	CCF=0	CCF=1 HC=3	CCF=1	CCF=1	CCF=1	CCF=1	CCF=1
2*Tx*+2*Tc*	CCF=0	CCF=0	CCF=1 HC=2	CCF=1	CCF=1	CCF=1	CCF=1
3*Tx*+2*Tc*	CCF=0	CCF=0	CCF=0	CCF=1 HC=1	CCF=1	CCF=1	CCF=1
4*Tx*+2*Tc*	CCF=0	CCF=0	CCF=0	CCF=0 CCC=3	CCF=1 HC=2	CCF=1	CCF=1
4*Tx*+3*Tc*	CCF=0	CCF=0	CCF=0 CCC=2	CCF=0	CCF=1 HC=2	CCF=1	CCF=1
4*Tx*+4*Tc*	CCF=0	CCF=1 CCC=1	CCF=0	CCF=0	CCF=1 HC=2	CCF=1	CCF=1
4*Tx*+5*Tc*	CCF=1 CCC=0 HC=4	CCF=1	CCF=0	CCF=0	CCF=1 HC=2	CCF=1	CCF=1
5*Tx*+4*Tc*	CCF=1 HC=4	CCF=1	CCF=0	CCF=0	CCF=0	CCF=1 HC=1	CCF=1
5*Tx*+5*Tc*	CCF=0	CCF=1 HC=3	CCF=0	CCF=0	CCF=0	CCF=1 HC=1	CCF=1
6*Tx*+4*Tc*	CCF=0	CCF=1 HC=3	CCF=0	CCF=0	CCF=0	CCF=0 CCC=3	CCF=1 HC=2

RTS coming from N_4 and overheard CC from N_2. Therefore, N_3 will not have an implicit acknowledgement of CC sent to N_2 in the first *Tc* duration after receiving ACK from N_4.

Hence, it will try to retransmit CC to N_2 node, which will cause a collision at N_4 with incoming CTS from N_5. Thus, the waiting period for *2Tc* ensures collision free transmission of P0 to N_5. The same is true for all later nodes where HC of a data packet becomes 0. In the meantime, after receiving ACK from N_4, N_3 sends CC with its CCC as 3 to N_2 in the first *Tc* duration. N_2 decrements value of CCC by one upon successful reception of CC. In the second *Tc* duration, N_2 forwards CC with its CCC as 2 to N_1. This CC is overheard by N_3, which acts as implicit acknowledgement for N_3 that N_2 has received CC successfully. Therefore, in general, explicit acknowledgement of CC transmission is not required. In the third *Tc* duration, N_1 forwards CC with its CCC as 1 to N_0 and sets its own CCF to 1. Thus, after getting CC from N_1, N_0 decrements the value of CCC of CC to 0 and also sets its own CCF to 1. Then, N_0 sends RTS for new data packet P_1 to N_1. This RTS also serves as implicit acknowledgement to N_1 that its earlier CC transmission to N_0 was successful. Hence, N_0 can transmit P1 to N_1 in one *Tx* duration and after that, N_1 can also forward P1 to N_2 in subsequent *Tx* duration. However, N_2 cannot forward P1 further as its CCF value is 0 since it has to wait for the next CC packet originating from N_5. Here, N_5 will generate a new CC only after it receives ACK from N_6, which N_6 will send to N_5 after successful reception of P_0. Thus, with the exception of the initial 4 hops, the whole communication stream is divided into segments of two hops with respect to the source node. As the last segment may have one or two hops, the node just before the sink will initiate CC transmission.

In addition to CC, RT-MAC has three more control packets. These are CCACK, CCQ and CCQR. CCACK is needed for the case where implicit acknowledgement does not work. It usually happens when the data packet arrival interval is large. For example, referring to Figure 3.7, if N_1 forwards CC to N_0 and if N_0 does not have any data packet to transfer to N_1, then implicit acknowledgement of CC by RTS for a new data packet from N_0 is not possible. Thus, N_0 must send CCACK explicitly to N_1, or else it will keep on transmitting duplicate copies of CC to N_1. CCQ is used for querying status of the CC control packet and CCQR is used to respond to this query. CCQ is needed when the expected CC control packet does not arrive on time to a node. The next section presents delay bounds in RT-MAC protocol.

3.4 Collision avoidance on RT-MAC

In RT-MAC, there is no collision between data packet of a stream because two consecutive data packets are separated by 4 hops due to feedback mechanism. However, there could be a collision between backward traveling CC control packet and forward traveling data packet. In order to avoid such collision, RT-MAC gives preference to data packet transmission over the CC transmission. This preference is needed to avoid a collision between a data packet traveling towards the sink and a CC traveling towards the source. This is ensured by the following two rules of the protocol.

1) RT-MAC gives preference to data packet transmission over the CC transmission.

2) There has to be some optimum wait duration prior to CC retransmission after a collision to avoid any further collision as well as to avoid any undesired delay in CC retransmission.

The first rule is insured by introducing the following two functionalities in the protocol. First, CC has relatively large carrier sense duration than other control packets. Second, the first hop neighbor (towards the sink) of the node under consideration waits for one *Tc* duration prior to initiating a CC transmission at the beginning of a frame. The reason behind these functionalities is as follows. Suppose the node under consideration is the n^{th} even numbered node. Now, in the first *Tc* duration at the beginning of a frame, if the n-1^{th} node initiates a packet transmission by sending RTS to the n^{th} node, and the n+1^{th} node also sends CC to the n^{th} node, then there will be collision in the n^{th} node. However, if the n+1^{th} waits for one *Tc* duration prior to initiating a transmission, then RTS from the n-1^{th} node will reach successfully to the n^{th} node. Thus, in the second *Tc* duration, the n^{th} node wants to send CTS and the n+1^{th} node wants to send CC. Now, if carrier sense duration of CC is large as compared to CTS's carrier sense duration, then the n^{th} node will win the contention and will start transmission of CTS. On hearing CTS from the n^{th} node, the n+1^{th} node suspends transmission of CC and goes to sleep mode till the end of ongoing data packet transmission. Thus, wait period prior to initiating CC transmission for the first time avoids collision. However, if a collision still happens between a data packet traveling towards the sink and a CC traveling towards the source due to some fault, then a waiting period is needed prior to CC retransmission to avoid further collisions. For

example, in the first Tc duration at the beginning of a frame, if the n-2^{th} node sends RTS to the n-1^{th} node and the n+1^{th} node waits (as per the first functionality discussed above) in this Tc duration even though it has CC to transmit to the n^{th} node. In the second Tc duration, there could be a harmful collision at the n^{th} node due to overheard CTS originating from the n+1^{th} node (intended for the n-2^{th} node) and CC originating from the n+1^{th} node (intended for the n^{th} node). As collision always happens at the receiving node only that hears two of more incoming transmissions, neighboring nodes are not aware of collision at n^{th} node. Hence, in the third Tc duration, the n-1^{th} and the n-2^{th} nodes will continue as usual in their data transfer cycle, whereas the n+1^{th} node will expect acknowledgement from the n^{th} node for its CC transmission. Now, if the n+1^{th} node does not wait and retransmits CC to the n^{th} node in the fourth Tc duration, then the n^{th} node receives this CC and sends an explicit CCACK to the n+1^{th} node in the fifth Tc duration. This CCACK will further cause collision at the n-1^{th} node, which causes disruption of data transfer cycle between the n-1^{th} and the n-2^{th} nodes. To avoid such a situation, if the n+1^{th} node waits for $Td+Tc(ACK)+Tc(W) = Td+2Tc$ prior to retransmitting CC, then it will allow completion of ongoing data transfer cycle between the n-1^{th} and n-2^{th} even numbered node. Here, Td is data packet duration in a data transfer cycle of duration Tx. This situation is represented by the first row entry of Table 3.2. However, if n+1^{th} node waits longer than $Td+2Tc$, then it will unnecessarily increase the end-to-end delay. Thus, the second rule for collision avoidance in RT-MAC is useful to address such requirements as it recommends optimum wait duration prior to CC retransmission after a collision to avoid any further collisions as well as to avoid any undesired delay in CC retransmission. The optimum wait period prior to CC transmission is given in Table 3.2.
The first column of this table shows hop distance between originating node of CC and the node under consideration (the n^{th} even numbered node). RTS_n, CTS_n, $DATA_n$, ACK_n and CC_n represent that these control and data packets are originated from the n^{th} node. $\{(ACK_n \ \& \ CC_{n+2}) \rightarrow X\}$ represents a collision at the n+1^{th} odd numbered node due to ACK originating from the n^{th} node and CC originating from the n+2^{th} node. $Tc(RTS)$, $Tc(CTS)$, $Tc(ACK)$, $Tc(W)$ represent that the duration of RTS, CTS, ACK and wait period by CC is Tc. $CC_{n+1(\text{lange sense})}$ represents that there is a large carrier sense duration prior to CC transmission from the n-1^{th} node.

Table 3.2: Optimum wait period prior to CC transmission.

Hop distance	Description	Wait duration of at the $n+1^{th}$ node	Wait duration of at the $n+2^{th}$ node
1	$(RTS_{n-2}$ & $Tc_{n+1}(W))$ + $\{(CTS_{n-1}$ & $CC_{n+1}) \rightarrow X\}$	$[Td+Tc(ACK)+Tc(W)]$ $= Td+2Tc$ {Large carrier sense before the second attempt of CC transmission is needed. It's optional in the first attempt.}	-
	$(RTS_{n-1}$ & $Tc_{n+1}(W))$ + $\{CTS_n$ & $CC_{n+1(\text{lange sense})}\}$	$[Tc(CTS)+Td+Tc(ACK)+Tc(W)] = Td+3Tc$ {Large carrier sense before the first attempt of CC transmission is needed. It is optional in the second attempt.}	-
2	$(RTS_{n-2}$ & $CC_{n+2})$ + $\{(CTS_{n-1}$ & $CC_{n+1}) \rightarrow X\}$	$[Td+Tc(ACK)+Tc(W)]$ $= Td+2Tc$	-
	$(RTS_{n-1}$ & $CC_{n+2})$ + $\{CTS_n$ & $CC_{n+1(\text{lange sense})}\}$	$[Tc(CTS)+Td+Tc(ACK)+Tc(W)]$ $= Td+3Tc$ {Large carrier sense before the first attempt of CC transmission is needed. No large carrier sense carrier before the second attempt.}	$[Tc(CTS)+Td+Tc(ACK)+Tc_1(W)]$ $= Td+2Tc$ {No large carrier sense is needed before the first attempt of CC transmission. It is needed before the second attempt.}
3	$(RTS_{n-2}$ & $CC_{n+3})$ + $(CTS_{n-1}$ & $CC_{n+2})$ + $CC_{n+1(\text{lange sense})}$	$[\{Td+Tc(ACK)+Tc(W)\} - Tc] = Td+Tc$	-

	$(RTS_{n-1}$ & $CC_{n+3})$ + $\{(CTS_n$ & $CC_{n+2}) \rightarrow X\}$ or $(DATA_{n-1}$ & $CC_{n+3})$ + $\{(ACK_n$ & $CC_{n+2}) \rightarrow X\}$	-	*[Td+Tc(ACK)] = Td+Tc* {This is applicable if CC transmission starts in the beginning of a frame } or *Tc(W) = Tc* {This is applicable if CC transmission starts in the middle of a frame }

3.5 Summary of the chapter

This chapter presents RT-MAC protocol in the single-stream scenario. It explains the functioning of RT-MAC protocol using timing diagrams. There is no collision between data packets of a stream in RT-MAC. However, there is some possibility of collision between backward traveling CC control packet and forward traveling data packet. In order to avoid such collision, this chapter also presents the optimum wait periods prior to CC transmission for various scenarios.

The next chapter presents delay bound analysis in normal mode of operations of RT-MAC protocol in the single-stream scenario.

Chapter 4: Delay Bound Analysis in normal mode of operation of RT-MAC protocol

This chapter presents the delay bound analysis in normal mode of operation of RT-MAC protocol in single-stream scenario. In the normal mode, no fault condition is considered in the network. As this chapter analyzes delay bounds for single-stream scenario, we will refer here the timing diagrams of packet transfer as shown in Figures 3.6 to 3.8 of Chapter 3. In these figures, T_{AI} represents the data packet arrival interval at the MAC layer of the source node. It depends upon the sampling interval of a sensor node. T_{SI} represents the duration between the start of two consecutive data packet transmissions at the source node. It depends upon the medium access strategy of the protocol. As shown in Figure 3.6, for $T_{AI} \leq 4Tx+5Tc$, two consecutive data packet transmissions that are separated by 4 hops (i.e., at least by $T_{SI} = 4Tx+5Tc$ duration) ensure collision free transmission. In fact, if $4Tx+5Tc < T_{AI} < 6Tx+8Tc$, then there is still some possibility of a collision between CC originating from N_5 (and traveling to N_2) with data packet transmission between N_1 and N_2. Hence, in the case of $T_{AI} < 6Tx + 8Tc$, the communication stream is considered as not being in a settled state. This case shows the behavior of RT-MAC under high load conditions. However, referring to Figure 3.7, $T_{AI} \geq 6Tx + 8Tc$ ensures a greater than 6 hop separation between two consecutive data packet transmissions. In this case, the next data packet transmission can immediately be initiated by a node because CC from later hops has already reached this node and sets up the node in transmission mode. Thus, the communication stream is considered as being in a settled state in the sense that a newly arrived data packet at the source will travel towards the sink without being affected by the previous data packets. This chapter presents delay bound for the following six conditions.

A) Delay bound without duty cycle (i.e., continuously ON) mode of operation of RT-MAC with hop separation < 6.

B) Delay bound without duty cycle mode of operation of RT-MAC with hop separation ≥ 6.

C) Upper delay bound in duty cycle (periodic sleep/listen schedule) mode of operation with hop separation < 6.

D) Upper delay bound in duty cycle mode of operation of RT-MAC with hop separation ≥ 6.

E) Lower delay bound in duty cycle mode of operation of RT-MAC with hop separation < 6.

F) Lower delay bound in duty cycle mode of operation of RT-MAC with hop separation ≥ 6.

$T_D(m,n)$ denotes end-to-end delay (also referred to as packet transfer delay) for the m^{th} data packet over n hops. $T_D(m,n)$ consists of two parts: (1) offset delay and (2) packet transmission delay. The offset delay, denoted by α, is the time that m^{th} packet waits for transmission of the previous $m\text{-}1^{th}$ data packets. The second part is the time taken by the m^{th} data packet to travel over n hops. Once transmission of the m^{th} data packet is started, it essentially takes the same time as that of the first data packet (i.e., $T_D(1,n)$) in the communication stream. Thus, $T_D(1,n)$ calculation is useful for event driven applications, whereas $T_D(m,n)$ is more relevant to periodic WSN applications.

4.1 End-to-end delay bound without duty cycle mode of operation of RT-MAC with hop separation < 6

Time taken by the first data packet to reach from the source to the destination, $T_D(1,n)$, can be calculated by using the transmission pattern shown in Figure 3.6. From this figure, it is evident that there is a waiting period of $2Tc$ after every two hops packet transmission time. As there is no offset delay for the first data packet, its end-to-end delay is the same as its packet transmission delay. Thus, $T_D(1,n)$ is given as

$$T_D(1,n) = 2Tx + \sum_{i=1}^{\left(\frac{n-2}{2}\right)} (2Tc + 2Tx) \qquad \text{for even } n \text{, i.e.,}$$

$$T_D(1,n) = nTx + (n\text{-}2)Tc \qquad \text{for even } n, \qquad (1)$$

and

$$T_D(1,n) = \sum_{i=1}^{\left(\frac{n-1}{2}\right)} (2Tx + 2Tc) + Tx \qquad \text{for odd } n \text{, i.e.,}$$

$$T_D(1,n) = nTx + (n\text{-}1)Tc \qquad \text{for odd } n. \qquad (2)$$

As shown in Figure 3.6, for $T_{AI} \leq 4Tx+5Tc$, the start of the second data packet is delayed by an offset duration of $T_{SI} = 4Tx+5Tc$. Therefore, the end-to-end delay for the second data packet is given as

$T_D(2,n) = T_{SI} + T_D(1,n)$, i.e.,

$T_D(2,n) = [4Tx+5Tc] + T_D(1,n)$.

Similarly, the end-to-end delay for the m^{th} data packet is given as

$T_D(m,n) = \alpha + T_D(1,n)$,

$T_D(m,n) = (m\text{-}1)[\ T_{SI}] + T_D(1,n)$,

$T_D(m,n) = (m\text{-}1)[4Tx+5Tc] + T_D(1,n)$, i.e.,

$$T_D(m,n) = [4(m\text{-}1)+n]Tx + [5(m\text{-}1)+(n\text{-}2)]Tc \qquad \text{for even } n \qquad (3)$$

and

$$T_D(m,n) = [4(m\text{-}1)+n]Tx + [5(m\text{-}1)+(n\text{-}1)]Tc \qquad \text{for odd } n. \qquad (4)$$

4.2 End-to-end delay bound without duty cycle mode of operation of RT-MAC with hop separation ≥ 6

The method of calculating packet transmission delay for the hop separation ≥ 6 is similar to the one used in Section 4.1. However, in the case of offset delay calculation, T_{SI} is equal to the actual data packet arrival interval T_{AI} at the source node for $T_{AI} \geq 6Tx + 8Tc$, as shown in Figure 3.7. Thus, $T_D(m,n)$ is given as

$T_D(m,n) = \alpha + T_D(1,n)$, i.e.,

$T_D(m,n) = (m\text{-}1)[\ T_{AI}] + T_D(1,n)$.

Referring to $T_D(1,n)$ from (1) and (2), $T_D(m,n)$ is given as

$$T_D(m,n) = (m\text{-}1)[\ T_{AI}] + nTx + (n\text{-}2)Tc \qquad \text{for even } n \qquad (5)$$

and

$$T_D(m,n) = (m\text{-}1)[\ T_{AI}] + nTx + (n\text{-}1)Tc \qquad \text{for odd } n. \qquad (6)$$

4.3 Upper limit of end-to-end delay in duty cycle mode of operation of RT-MAC for hop separation < 6

As shown in Figure 3.8, a data packet can travel one or more hops in a given frame duration T_f depending upon the ON duration, T_U, of sensor nodes. Thus, in order to find

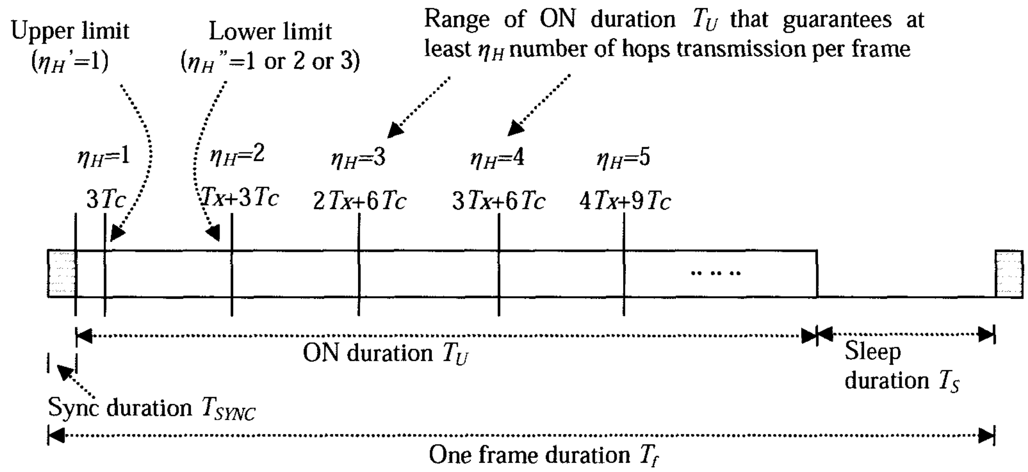

Figure 4.1: Dividing ON duration into Range of durations that ensure at least a given number of hops transmissions per frame (η_H) (Time scale is not linear).

the end-to-end delay bound, we need to find the number of hops, denoted by η_H, that a data packet can travel during the frame time with a given ON duration (i.e., for a given duty cycle). That is why η_H is an important parameter as it gives a range of T_U, which guarantees η_H number of hops transmission per frame. Now, in view of the communication pattern of RT-MAC, we will divide T_U into various contiguous ranges that ensure a given η_H as shown in Figure 4.1. The methodology of dividing T_U into various contiguous ranges is explained here.

In RT-MAC, one complete data transfer cycle is possible if two neighboring nodes, say N_1 and N_2, are ON at least for one Tc duration as shown in the first frame T_{f1} in Figure 4.2. In this case, N_0 sends RTS to N_1 and waits for CTS from N_1 for one more Tc duration. Then, N_1 replies with CTS and thus, both N_0 and N_1 will remain ON till completion of one data cycle. This figure also justifies that RT-MAC is capable of working at a much lower duty cycle ($T_U = Tc$) than other contention based protocols such as S-MAC and T-MAC protocols, which have large carrier sense duration prior to RTS transmission.

As explained in Section 3.2, the even numbered nodes wait for $2Tc$ duration prior to forwarding a data packet further. Now, if two neighboring nodes, for example N_2 and N_3 in Figure 4.3 are ON for just $2Tc$ duration at the beginning of a frame, then no data transfer is possible as both nodes will go into sleep mode after $2Tc$ duration.

However, if both N_0 and N_1 are ON for $3Tc$ duration as shown in the fourth frame T_{f4} in Figure 4.4, then N_1 can send CC to N_0 in the third Tc duration and waits for acknowledgement of this CC for one more Tc duration. Then, N_0 responds to CC by sending RTS for a data packet to N_1 in the next Tc duration, after which N_5 responds with CTS; and thus both N_0 and N_1 do not go into sleep mode until the end of this data packet transfer. Therefore, the minimum ON duration that guarantees at least one data transfer cycle per frame (i.e., η_H = 1 hop/frame) is $3Tc$. This is shown by the first row entry of Table 4.1 for η_H =1. Here, as shown the first and third frame in Figure 4.4, the data transfer by more than one hop is possible in $3Tc$ ON duration due to adaptive listening [71] and [72]. However, as odd numbered nodes do not need to wait for $2Tc$ duration prior to data transmission, therefore, one Tc duration will ensure one data transfer cycle. This is presented by the second row entry of Table 4.1 for η_H =1.

Table 4.1: Analysis of ON duration requirement to ensure a given number of hops data packet transfer per frame (η_H).

η_H	Starting Node	Packet transmission pattern for minimum ON duration that guarantees at least η_H number of data transfer cycle per frame for a given starting node	Duration needed
1	N_{even}	$2Tc+\{Tc \rightarrow Tx(CC\text{-}R)\}$	<u>$3Tc$</u>
	N_{odd}	$[Tc \rightarrow Tx(R)]$	Tc
2	N_{even}	$3Tc+[Tc \rightarrow Tx(R)]+[Tc \rightarrow Tx(COH)]$	$5Tc$
	N_{odd}	$Tx+[2Tc+\{Tc \rightarrow Tx(CC\text{-}R)\} = 3Tc]$	<u>$Tx+3Tc$</u>
3	N_{even}	$3Tc+2Tx+[2Tc+\{Tc \rightarrow Tx(CC\text{-}R)\} = 3Tc]$	<u>$2Tx+6Tc$</u>
	N_{odd}	$Tx+3Tc+[Tc \rightarrow Tx(R)]+[Tc \rightarrow Tx(COH)]$	$Tx+5Tc$

4	N_{even}	$3Tc+2Tx+3Tc+[Tc \rightarrow Tx(R)]+[Tc \rightarrow Tx(COH)]$	$2Tx+8Tc$
	N_{odd}	$Tx+3Tc+2Tx+[2Tc+\{Tc \rightarrow Tx(CC\text{-}R)\} = 3Tc]$	$3Tx+6Tc$
5	N_{even}	$3Tc+2Tx+3Tc+2Tx+[2Tc+\{Tc \rightarrow Tx(CC\text{-}R)\} = 3Tc]$	$4Tx+9Tc$
	N_{odd}	$Tx+3Tc+2Tx+3Tc+[Tc \rightarrow Tx(R)]+[Tc \rightarrow Tx(COH)]$	$3Tx+8Tc$
6	N_{even}	$3Tc+2Tx+3Tc+2Tx+3Tc+[Tc \rightarrow Tx(R)]+[Tc \rightarrow Tx(COH)]$	$4Tx+11Tc$
	N_{odd}	$Tx+3Tc+2Tx+3Tc+2Tx+[2Tc+\{Tc \rightarrow Tx(CC\text{-}R)\} = 3Tc]$	$5Tx+9Tc$

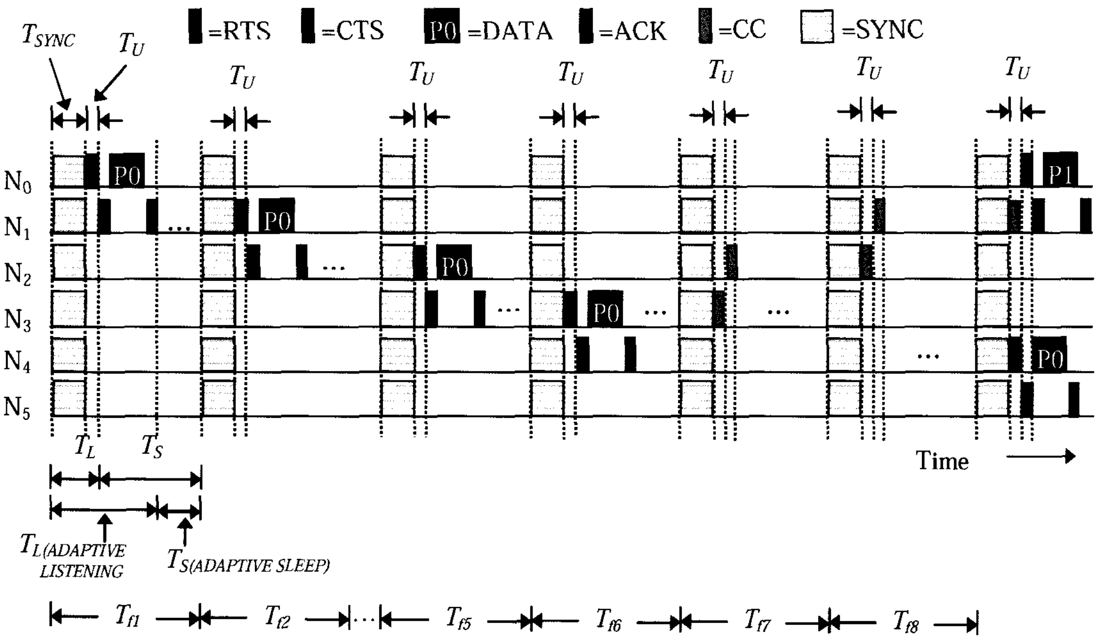

Figure 4.2: Timing diagram with $T_U = Tc$ (more precisely $Tc \leq T_U < 2Tc$) in RT-MAC protocol with periodic sleeping ($2Tc$ wait duration needed at N_2 will take the third and fourth frame. It is not shown here to fit the figure horizontally in the available space).

In Table 4.1, $Tc \rightarrow Tx(R)$ represents that one complete data transfer cycle is possible if two neighboring nodes are ON at least one Tc duration so that the receiving node can receive RTS successfully from the sending node. $Tc \rightarrow Tx(CC\text{-}R)$ represents that though nodes are ON for one Tc duration, still one packet transfer cycle is completed by sending RTS in response to CC packet. $Tc \rightarrow Tx(COH)$ represents that one data transfer cycle is possible between one and two hop neighbors of a sending node if they are awake for $2Tc$ duration. In this case, two hop neighbors overhear the CTS in the second Tc duration (sent by one

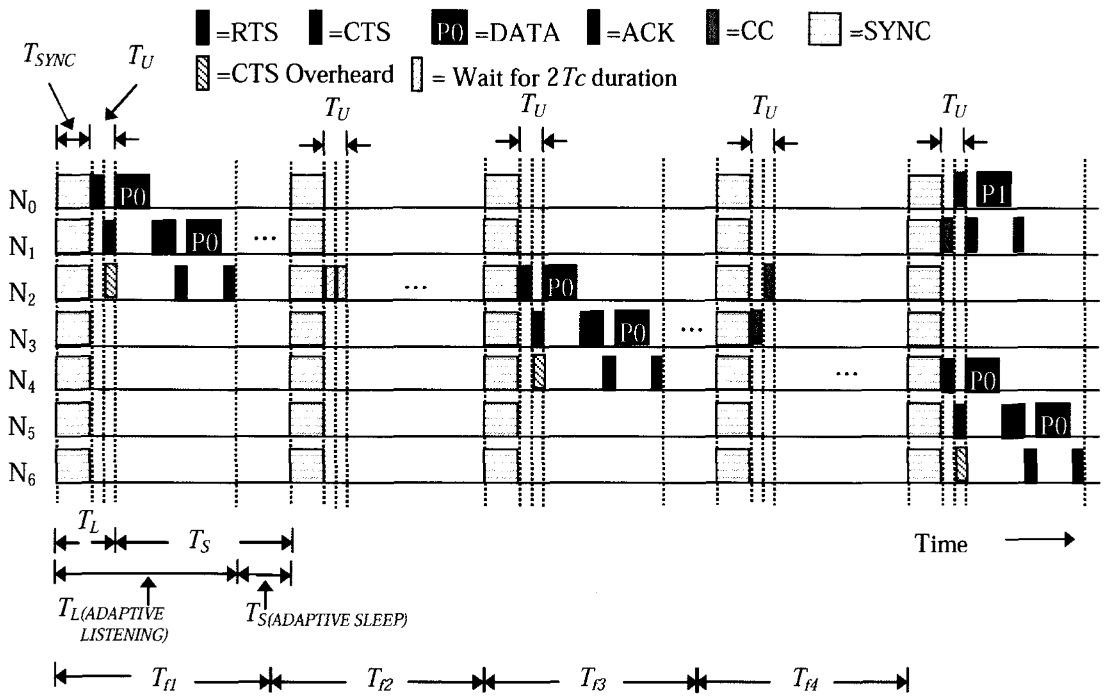

Figure 4.3: Timing diagram with $T_U = 2Tc$ in RT-MAC protocol with periodic sleeping

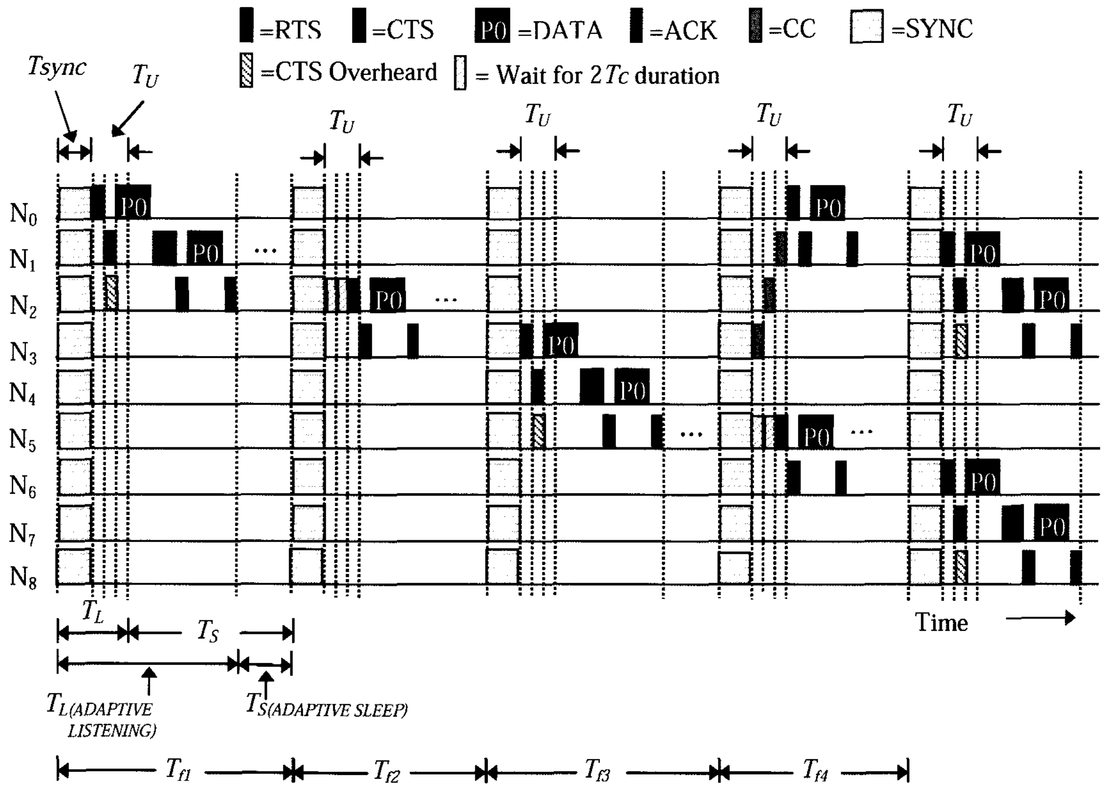

Figure 4.4: Timing diagram with $T_U = 3Tc$ in RT-MAC protocol with periodic sleeping

hop neighbor to the sending node); hence it wakes up at the end of data packet transmission between the sending node and the one hop neighbor. This phenomenon is called adaptive listening as discussed in S-MAC. Thus, $Tc \rightarrow Tx(COH)$ transmission pattern is always preceded by $Tc \rightarrow Tx(R)$ transmission pattern. Here, *COH* stands for the overheard CTS control packet.

Now, if we take the maximum of two possible values of T_U for any η_H from the fourth column of Table 4.1, then this will be the minimum required ON duration T_U, which will ensure, irrespective of the starting node, a data packet transfer in the beginning of a frame for a particular η_H. Thus, the minimum of $3Tc$ ON duration is needed to ensure at least for $\eta_H = 1$. The same explanation applies to other entries of Table 4.1 . Carrying out analysis similar to above, it is found that $T_U = Tx+3Tc$ ensures $\eta_H = 2$ hops/frame, $T_U = 2Tx+6Tc$ ensures $\eta_H = 3$ hops/frame, $T_U = 3Tx+6Tc$ ensures $\eta_H = 4$ hops/frame, and so on. Therefore, the range of T_U for a given η_H can be given as

$3Tc \leq T_U < Tx+3Tc$ for $\eta_H = 1$, (7a)

$2Tx+6Tc \leq T_U < 3Tx+6Tc$ for $\eta_H = 3$, (7b)

$4Tx+9Tc \leq T_U < 5Tx+9Tc$ for $\eta_H = 5$, (7c)

....

$(\eta_H-1)Tx+\{3(\eta_H+1)/2\}Tc \leq T_U < \eta_H Tx+\{3(\eta_H+1)/2\}Tc$

for $\eta_H = 1, 3, 5, 7,$ .. upto n). (7d)

Similarly, for even η_H, the range of T_U can be given as

$Tx+3Tc \leq T_U < 2Tx+6Tc$ for $\eta_H = 2$, (8a)

$3Tx+6Tc \leq T_U < 4Tx+9Tc$ for $\eta_H = 4$, (8b)

$5Tx+9Tc \leq T_U < 6Tx+12Tc$ for $\eta_H = 6$, (8c)

....

$(\eta_H-1)Tx+(3\eta_H/2)Tc \leq T_U < \eta_H Tx+\{(3\eta_H/2)+3\}Tc$

for $\eta_H = 2, 4, 6, 8,$ .. upto n. (8d)

Inequalities in (7a) to (8d) are valid with the following constraint for T_U for a given frame duration T_f.

$T_U < T_f - 2Tx + 2Tc$ (9)

The above constraint on T_U is due to adaptive listening. If this constraint is not met, then delay is not guaranteed as the ongoing transmission of a data packet at the end of ON duration in the current frame, and the transmission will be terminated due to the start of SYNC duration of the next frame. In order to ensure a given η_{H}, if we select ON duration, T_U, to the minimum possible value (i.e., the lower bound of inequalities in (7a) to (8d)), then we will obtain an upper limit of the end-to-end delay for that η_H. Additionally, for a given Tx, Tc and T_U, we can find an integer value of η_H using the lower bound of inequalities in (7d) or (8d) (though the usage of (7d) is recommended as it has the larger value of T_U that accommodates both possibilities of even and odd η_H). For example, if the total number of hops, n, is 50 and T_U is taken such that it ensures η_H=6 hops per frame, then $\lfloor n/\eta_H \rfloor = 8$ full frames are needed by a data packet to reach the 48th hop node with the remaining two hops being reached in the 9th frame. Here, $\lfloor\ \rfloor$ is a floor operator. The number of hops traveled by a data packet in the last frame is represented by n_L, which is 2 hops in the above example. Equations (1) and (2) can be used to calculate the packet transmission delay for n_L hops in the last frame because a node remains continuously ON to facilitate up to η_H hops packet transmission in a frame and n_L is always less than η_H.

Thus, the end-to-end delay for the first data packet is given as

$$T_D(1,n) = \left\lfloor \frac{n}{\eta_H} \right\rfloor T_f + [n_L\, Tx + (n_L - 2)Tc] \qquad \text{for even } n_L \qquad (10a)$$

and

$$T_D(1,n) = \left\lfloor \frac{n}{\eta_H} \right\rfloor T_f + [n_L\, Tx + (n_L - 1)Tc] \qquad \text{for odd } n_L, \qquad (10b)$$

where, n_L is mod(n, η_H) and mod represents a modulo operation.

Referring to Figure 3.8, the minimum time interval between two consecutive data packets is $T_{SI} = 4Tx+5Tc$. Thus, in the case of the second data packet, an offset delay of 4 hops transmission time is added (to the packet transmission time), which can be considered as if the second data packet needs to travel by 4+n hops, instead of n hops. Therefore, the end-to-end delay for the second data packet is given as

$T_D(2,n) = \alpha + T_D(1,n)$, i.e.,

$$T_D(2,n) = \left\lfloor \frac{(4+n)}{\eta_H} \right\rfloor T_f + [n_L\, Tx + (n_L - 2)Tc] \qquad \text{for even } n_L$$

and

$$T_D(2,n) = \left\lfloor \frac{(4+n)}{\eta_H} \right\rfloor T_f + [n_L\, Tx + (n_L - 1)Tc] \qquad \text{for odd } n_L,$$

where, n_L is mod($4+n$, η_H).

Similarly, for the m^{th} data packet, the upper limit of end-to-end delay is given as

$$T_D(m,n) = \left\lfloor \frac{(4(m-1)+n)}{\eta_H} \right\rfloor T_f + [n_L\, Tx + (n_L - 2)Tc] \qquad \text{for even } n_L \qquad (11a)$$

and

$$T_D(m,n) = \left\lfloor \frac{(4+n)}{\eta_H} \right\rfloor T_f + [n_L\, Tx + (n_L - 1)Tc] \qquad \text{for odd } n_L, \qquad (11b)$$

where, n_L is mod($4(m-1)+n$, η_H).

4.4 Upper limit for end-to-end delay in duty cycle mode of RT-MAC for hop separation ≥ 6

As mentioned in Subsection 4.2, T_{SI} is equal to T_{AI} for $T_{AI} \geq 6Tx + 8Tc$. Thus, the offset delay, α, depends upon the actual data packet arrival interval, T_{AI}, at the source node. T_{AI} is based on the sampling interval of a sensor node; thus, it is known apriori. In the following analysis, it is assumed that data sampling is done periodically only when a node is ON. Once transmission of a data packet starts from the source node, it will have the same packet transmission delay as the end-to-end delay of the first data packet, which is given by (10a) and (10b). Therefore, the next step is to find the offset delay, α, for the m^{th} data packet, which is equal to $(m-1)T_{AI}$. But, in order to combine α with $T_D(1,n)$, α needs to be represented in terms of T_f. In this regard, Figures 4.5 and 4.6 show two possible cases of data packet arrival patterns. In either case, the number of frames (denoted by a) needed for m-1 data packet transfers is given by

$$a = \left\lceil \frac{(m-1)T_{AI}}{T_U} \right\rceil, \qquad (12)$$

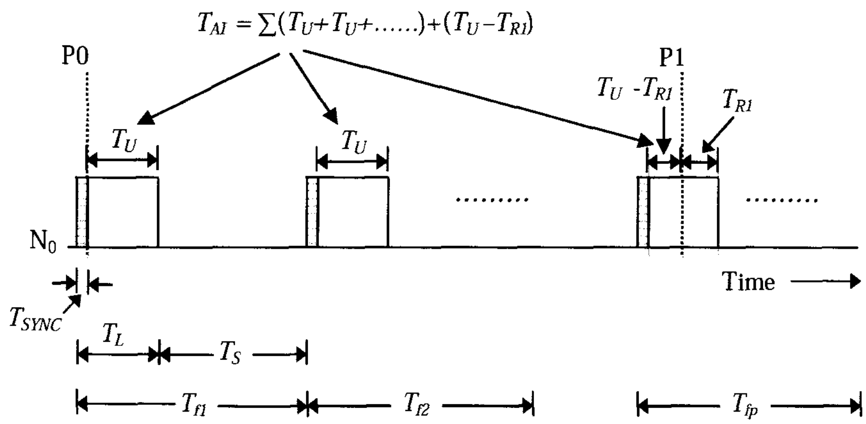

Figure 4.5: Timing diagram for RT-MAC with $T_U \leq 6Tx+8Tc < T_{AI}$

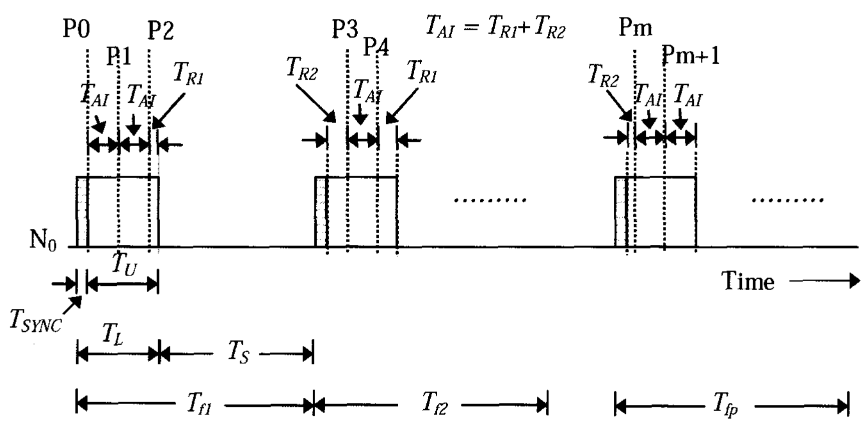

Figure 4.6: Timing diagram for RT-MAC with $6Tx+8Tc \leq T_{AI} < T_U$

where $\lceil \;\rceil$ is a ceiling operator. Figure 4.5 shows a case where a is greater than 1. Figure 4.6 shows a case where a is 1. If $(m-1)T_{AI}$ is not fully divisible by T_U, then some portion of the ON duration (denoted by T_{R1} in Figures 4.5 and 4.6) of the a^{th} frame will also be available for the transmission of the m^{th} data packet. It is given as

$$T_{R1} = aT_U - (m-1)T_{AI}. \tag{13}$$

Thus, α is equal to $aT_f - T_{R1}$. The upper limit of end-to-end delay for the m^{th} data packet over n hops is given by

$$T_D(m,n) = \alpha + T_D(1,n),$$

$$T_D(m,n) = aT_f - T_{R1} + T_D(1,n),$$

$$T_D(m,n) = aT_f + [T_D(1,n) - T_{R1}]. \tag{14}$$

Using the lower bound of inequality in (7d), we try to find out the possible number of hops transmission (denoted by n') of the m^{th} data packet in T_{R1} duration of the a^{th} frame.

It is given as

$(n'-1)Tx+\{3(n'+1)/2\}Tc \leq T_{R1}$,

$$n' = \left\lfloor \frac{T_{R1} + Tx - 1.5Tc}{Tx + 1.5Tc} \right\rfloor. \qquad (15)$$

Now, after the a^{th} frame, the m^{th} data packet still needs to travel $(n - n')$ hops. Therefore, the packet transmission delay for $(n - n')$ hops can be calculated using (10a) and (10b). Hence, the end-to-end delay for the m^{th} data packet is given as

$T_D(m,n) = aT_f + [T_D(1,n) - T_{R1}]$,

$T_D(m,n) = aT_f + [T_D(1, n - n')]$, i.e.,

$$T_D(m,n) = \left[a + \left\lfloor \frac{n-n'}{\eta_H} \right\rfloor \right] T_f + [n_L\, Tx + (n_L - 2)Tc]$$

for even n_L (16a)

and

$$T_D(m,n) = \left[a + \left\lfloor \frac{n-n'}{\eta_H} \right\rfloor \right] T_f + [n_L\, Tx + (n_L - 1)Tc]$$

for odd n_L, (16b)

where, n_L is mod($(n - n')$, η_H).

4.5 Lower limit of end-to-end delay in duty cycle mode of operation of RT-MAC for hop separation < 6

In general, if ON duration is longer, then end-to-end delay is smaller. Therefore, to find the lower bound for the end-to-end delay, we need to take the maximum permissible ON duration from a specified range of T_U for a given η_H. Thus, if T_U is taken as the upper limit of inequalities in (7a) to (8d), then it gives the lower limit of end-to-end delay. Using (7a), (8a), (7b) and (8b), the upper limits of T_U that guarantee at least $\eta_H = 1, 2, 3$ and 4 hops transmission per frame is given as

$T_U = Tx+2Tc$ for $\eta_H = 1$, (17a)

$T_U = 2Tx+5Tc$ for $\eta_H = 2$, (17b)

$T_U = 3Tx+5Tc$ for $\eta_H = 3$, (17c)

$T_U = 4Tx{+}8Tc$ for $\eta_H = 4$. (17d)

As Tc is the smallest ON duration that is needed for a successful transmission, the above equations are achieved by subtracting Tc from the lower limit of range for the next integer value of η_H. Calculating the lower bound of end-to-end delay of RT-MAC gives us a greater insight in operation of the protocol. For example, as per (7a) and (7b), T_U is equal to $3Tc$, which guarantees at least η_H=1; and T_U is equal to $Tx{+}3Tc$, which guarantees at least η_H=2. However, if T_U is taken as $Tx{+}2Tc$, then the data packet may travel more than one hop per frame in some scenarios. Thus, in this subsection, we try to analyze all possible scenarios in RT-MAC for a T_U as given by (17a) and (17d). As shown in Figures 3.6 and 3.8, the same packet transmission pattern is repeated between N_0 to N_4 at the interval of $4Tx{+}5Tc$ (i.e., with hop separation = 4). Because of this 4 hop symmetry, the lower limit of the end-to-end delay for η_H=1 can be easily generalized for η_H = 5, 9…etc. The same is true of η_H = 2, 6, 10… and so on. Therefore, the lower bound of end-to-end delay is calculated here separately for η_H = 1, 2, 3 and 4.

To calculate the lower limit of end-to-end delay *for* η_H = 1, we need to calculate the offset delay, α, and packet transmission delay for the current data packet for $T_U = Tx{+}2Tc$. The exact value of offset delay and packet transmission delay depends upon data packet transmission per frame for a given set of protocol variables (i.e., CCF, CCC and HC) and initial conditions (i.e., relative positions of previous data packet and CCs at the beginning of the current frame). A given set of values of protocol variables and initial conditions give rise to the concept of the state of a communication stream in a neighborhood that is involved in a medium access decision. One state represents a complete transmission pattern for the current frame. Thus, we need to find out all possible states for a valid combination of protocol variables and initial conditions. Thus, we construct an offset state table and a packet transmission state table for the current data packet transmission.

All the state tables are constructed with the following four conditions.

a) The current data packet transmission starts from an even numbered node.

b) The current data packet transmission starts from an odd numbered node.

c) The start of the current data packet transmission is delayed by $2Tc$ duration due to a wait duration for CC from one hop or two hop neighbor towards the sink, or to allow a successful transmission of CC towards the source from previous hop neighbors.

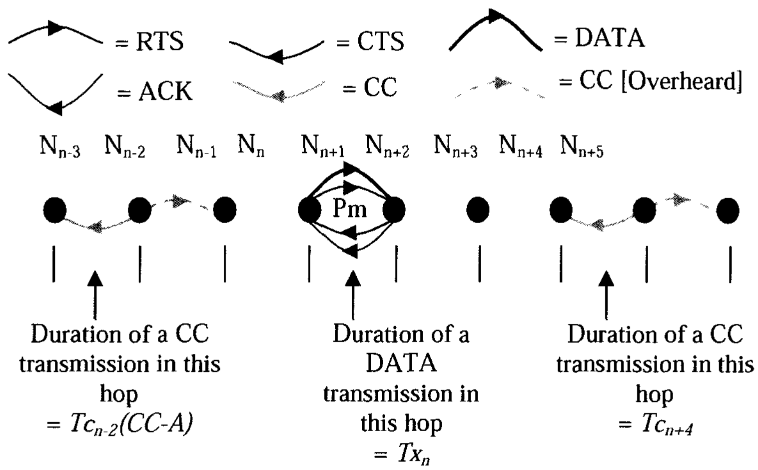

Figure 4.7: Description of notations used in offset delay and packet transmission state tables of RT-MAC

d) The start of the current data packet transmission is delayed by $3Tc$ duration due to a wait duration for CC from a three hop neighbor towards the sink.

As offset delay calculation depends upon the first four hop nodes in a stream, hence, an offset delay state table will show transmission among N_0 to N_4 nodes. However, the packet transmission state table shows nodes in terms of even and odd numbered nodes (N_{even} and N_{odd}). It is because of symmetry in packet transmission patterns started from even or odd numbered nodes anywhere in a stream. There are three types of states. General states occur frequently during normal operation of the protocol. Rare states occur occasionally during normal operation of the protocol such as the first state at the start of data transmission or the states that involve boundary nodes of virtual clusters of sensor nodes [71] and [72]. Fault states occur in a scenario such as a clock shift in sensor nodes, dying sensor nodes or collisions. A group of general states forms an equilibrium of states that keep on repeating until some fault occurs. If the protocol operates in equilibrium states, then the specified end-to-end delay deadline will not be missed. Tables 4.2 and 4.3 show all the possible states in the current frame as well as expected states in the next frame for η_H=1 with $T_U = Tx+2Tc$.

Notations used in the following analysis are shown in Figure 4.7. In this figure, the n^{th} node is referred to as "node under consideration". In general, the node under consideration is a node that is about to send or about to receive a data packet at the beginning of a frame. All other nodes and transmissions originating from them are represented with respect to the node under consideration. $\{Tc_{-1}(CC\text{-}A)\ \&\ /\ |\ Tc_2\}$ indicates that any one or both of the CC transmissions are possible simultaneously (i.e., without collision) in the same Tc duration as these CC transmissions are occurring between different pairs of nodes. $Tc_n(W)$ represents wait for Tc duration by the n^{th} node prior to initiating a transmission. $Tx_n(W)$ represents wait for Tx duration by the n^{th} node prior to initiating a transmission. If the ON duration available for a data packet transmission is $Tx+Tc$, then the transmission pattern is represented as $Tx+Tx(R)$ since the additional Tc duration can ensure one data packet transfer cycle if the concerned nodes are eligible to initiate a transmission. However, if the ON duration available for data packet transmission is just one Tx, then the transmission pattern is represented as $Tx+Tx(COH)$ since one extra hop transmission is possible due to an adaptive listening mechanism provided the concerned nodes are eligible to initiate a transmission. $Tx_n(CC\text{-}R)$ represents one data packet transmission done by the n^{th} node in one Tc ON duration in response to CC from its one hop neighbor towards the sink.

In Tables 4.2 and 4.3, GS, RS and FS represent general state, rare state and fault state respectively. Reported CC transmissions in the second column of Tables 4.2 and 4.3 do not overlap with the data packet transmissions in a neighborhood.

Table 4.2: Offset delay States detail for $T_U = Tx+2Tc$ (that ensures η_H=1) and $T_{AI} < 6Tx+8Tc$ in fault scenario

Current state	Transmission pattern description for the current frame	Starting Node	η_H'	Wait for $2Tc$ / $3Tc$ in the next state by the next starting node	Next state	(Type of current state): Initial conditions
O1	$Tx_0+Tx_1(R)$	N_0	2	$Tc_5 + Tc_4 + Tc_3 = 3Tc$	O19	(RS): Previous packet transmission starts from N_4 at the beginning of current frame.
O2	$Tx_0+Tx_1(R)$	N_0	2	$Tc_4 + Tc_3 = 2Tc$	O17	(FS): Previous packet transmission starts from N_5 at the beginning of current frame.

O3	$Tx_0+Tx_1(R)$	N_0	2	$Tc_3(W)+Tc_3=2Tc$	O17	(FS): If CC transmission starts from N_5 in the current frame.
O4	$Tx_0+Tx_1(R)$	N_0	2	$Tc_3(W)+Tc_3=2Tc$	O17	(FS): If CC transmission starts from N_4 in the current frame.
O5	$Tx_0+Tx_1(R)$	N_0	2	$Tc_3(W)+Tc_3=2Tc$	O17	(FS): If CC transmission starts from N_3 in the current frame.
O6	$Tx_0+Tx_1(R)+Tx_2(COH)$	N_0	3	-	O15	(FS): If CC=1 is already available at N_2.
O7	Tx_0	N_0	1	$Tc_3(W)+Tc_3=2Tc$	O13	(FS): If CC transmission starts from N_3 immediately after the first data transfer cycle in the current frame.
O8	$Tx_2+Tx_3(R)$	N_2	2	$Tc_3+Tc_2+Tc_1=3Tc$	O18	(RS): Previous packet transmission starts from N_6 in current frame.
O9	Tx_1	N_1	1	$Tc_5+Tc_4+Tc_3=3Tc$	O19	(FS): If previous packet transmission starts from N_4 in current frame.
O10	$Tx_1+Tc_5+Tc_4$	N_1	1	$Tc_3(W)+Tc_3=2Tc$	O17	(RS): Previous packet transmission starts from N_5 at the beginning of current frame followed by CC.
O11	$\{Tx_1$ & $Tc_5\}+Tc_4+Tc_3+Tx_2(CC\text{-}R)$	N_1	2	-	O15	(FS): If CC transmission starts from N_5 in the current frame.
O12	$Tx_1+Tc_3(W)+Tc_3+Tx_2(CC\text{-}R)$	N_1	2	-	O15	(FS): If CC transmission starts from N_4 in the current frame.

O13	$Tx_1+Tc_3(W)+Tc_3+$ $Tx_2(CC\text{-}R)$	N_1	2	-	O15	(FS): If CC transmission starts from N_3 in the current frame.
O14	$Tx_1+ Tc(W)+ Tc(W)$	N_1	1	-	O8	(FS): If CC=1 is already available at N_2.
O15	$Tx_3+Tc_3(CC\text{-}A)+$ $Tc_2(CC\text{-}A)$	N_3	1	$Tc_1(W) +Tc_1= 2Tc$	O16	(RS): Previous packet transmission starts from N_7 at the beginning of current frame.
O16	$\{Tc_2 \mid Tc_1(W)\} + Tc_1+Tx_0+Tx_1(COH)$	N_0	2	$Tc_5 +Tc_4 +Tc_3= 3Tc$	O19	(RS): Previous packet transmission starts from N_4 at the beginning of current frame.
O17	$\{Tc_4 \mid Tc_3(W)\}$ $+Tc_3+Tx_2+Tx_3(COH)$	N_2	2	$Tc_3 +Tc_2 +Tc_1= 3Tc$	O18	(RS): Previous packet transmission starts from N_6 at the beginning of current frame.
O18	$Tc_3+Tc_2+Tc_1+Tx_0+$ $Tx_1(COH)$	N_0	2	$Tc_5 +Tc_4 +Tc_3= 3Tc$	O19	(GS): Previous packet transmission starts from N_4 in current frame but its start is delayed by $3Tc$.
O19	$Tc_5 +Tc_4$ $+Tc_3+Tx_2+Tx_3(COH)$	N_2	2	$Tc_3 +Tc_2 +Tc_1= 3Tc$	O18	(GS): Previous packet transmission starts from N_6 in current frame but its start is delayed by $3Tc$.

Table 4.3: Packet Transmission delay States Detail for $T_U = Tx+2Tc$ (that ensures η_H=1) and $T_{AI} < 6Tx+8Tc$ in fault scenario

Current state	Transmission pattern description for the current frame	Starting Node	η_H"	Wait for $2Tc$ / $3Tc$ in the next state by the next starting node	Next state	(Type of current state): Initial conditions

T1	$Tx_n+Tx_{n+1}(R)$	N_{even}	2	$\{Tc_{n+1}(CC\text{-}A)$ & $Tc_{n+5}\}$ + $\{Tc_n(CC\text{-}A)$ & $Tc_{n+4}\}+Tc_{n+3}$ $= 3Tc$	T15	(RS): Previous packet transmission starts from the fourth hop neighbor towards the sink at the beginning of the current frame.
T2	$Tx_n+\{Tx_{n+1}(R)$ & $Tc_{n+5}\}$	N_{even}	2	$\{Tc_{n+1}(CC\text{-}A)$ & $Tc_{n+4}\}$ + $\{Tc_n(CC\text{-}A)$ & $Tc_{n+3}\}= 2Tc$	T14	(FS): Previous packet transmission starts from the fifth hop neighbor towards the sink at the beginning of the current frame.
T3	$\{Tx_n$ & Tc_{n+5} & $Tc_{n+4}\}+Tx_{n+1}(R)$	N_{even}	2	$\{Tc_{n+1}(CC\text{-}A)$ & $Tc_{n+3}(W)\}$ + $\{Tc_n(CC\text{-}A)$ & $Tc_{n+3}\}= 2Tc$	T14	(FS): If CC transmission starts in the current frame from its fifth hop neighbor towards the sink.
T4	$\{Tx_n$ & $Tc_{n+4}\}+Tx_{n+1}(R)$	N_{even}	2	$\{Tc_{n+1}(CC\text{-}A)$ & $Tc_{n+3}(W)\}$ + $\{Tc_n(CC\text{-}A)$ & $Tc_{n+3}\}= 2Tc$	T14	(FS): If CC transmission starts in the current frame from its fourth hop neighbor towards the sink.
T5	$Tx_n+Tx_{n+1}(R)$	N_{even}	2	$\{Tc_{n+1}(CC\text{-}A)$ & $Tc_{n+3}(W)\}$ + $\{Tc_n(CC\text{-}A)$ & $Tc_{n+3}\}= 2Tc$	T14	(FS): If CC transmission starts in the current frame from its third hop neighbor towards the sink.
T6	$Tx_n+Tx_{n+1}(R)+Tx_{n+2}(COH)$	N_{even}	3	-	T9	(FS): If CC=1 is already available at its two hop neighbor.
T7	Tx_n	N_{even}	1	$Tc_{n+3}(W)$ $+Tc_{n+3}= 2Tc$	T12	(FS): If CC transmission starts in the current frame from its third hop neighbor towards the sink immediately after the first data transfer cycle.

T8	Tx_n+Tc_n(CC-A)+Tc_{n-1}(CC-A)	N_{odd}	1	Tc_{n+4}+Tc_{n+3}+ Tc_{n+2}= 3Tc	T15	(FS): Previous packet transmission starts from the third hop neighbor towards the sink at the beginning of the current frame.
T9	Tx_n+{Tc_{n+4} & Tc_n(CC-A)} +{Tc_{n+3} & Tc_{n-1}(CC-A)}	N_{odd}	1	Tc_{n+2}(W) +Tc_{n+2}= 2Tc	T14	(RS): Previous packet transmission starts from the fourth hop neighbor towards the sink at the beginning of the current frame.
T10	{Tx_n & Tc_{n+4} }+{Tc_{n+3} & Tc_n(CC-A)} +{Tc_{n+2} & Tc_{n-1}(CC-A)}+Tx_{n+1}(CC-R)	N_{odd}	2	-	T9	(FS): If CC transmission starts in the current frame from its fourth hop neighbor towards the sink.
T11	{Tx_n &Tc_{n+3} }+{Tc_{n+2}(W) &Tc_n(CC-A)} +{Tc_{n+2} & Tc_{n-1}(CC-A)}+Tx_{n+1}(CC-R)	N_{odd}	2	-	T9	(FS): If CC transmission starts in the current frame from its third hop neighbor towards the sink.
T12	Tx_n+{Tc_{n+2}(W) & / | Tc_n(CC-A)} +{Tc_{n+2} & / | Tc_{n-1}(CC-A)}+Tx_{n+1}(CC-R)	N_{odd}	2	-	T9	(FS): If CC transmission starts in current the frame from its two hop neighbor towards the sink.
T13	Tx_n+Tc_n(CC-A)+Tc_{n-1}(CC-A) +Tx_{n+1}(COH)	N_{odd}	2	-	T9	(FS): If CC=1 is already available at its one hop neighbor.
T14	{Tc_{n-1}(CC-A) & / | (Tc_{n+2} | Tc_{n+1}(W))} + {Tc_{n-2}(CC-A) & / | Tc_{n+1}}+Tx_n+ Tx_{n+1}(COH)	N_{even}	2	Tc_{n+5} +Tc_{n+4} +Tc_{n+3} = 3Tc	T15	(RS): Previous packet transmission starts from the fourth hop neighbor towards the sink at the beginning of the current frame.
T15	{Tc_{n-1}(CC-A) & / | Tc_{n+3}} +	N_{even}	2	Tc_{n+5} +Tc_{n+4} +Tc_{n+3} = 3Tc	T15	(GS): Previous packet transmission starts from the

	$\{Tc_{n-2}(CC\text{-}A)$ & / | $Tc_{n+2}\}$ + $Tc_{n+1}+Tx_n+Tx_{n+1}(COH)$					fourth hop neighbor towards the sink in current frame but its start is delayed by $3Tc$.

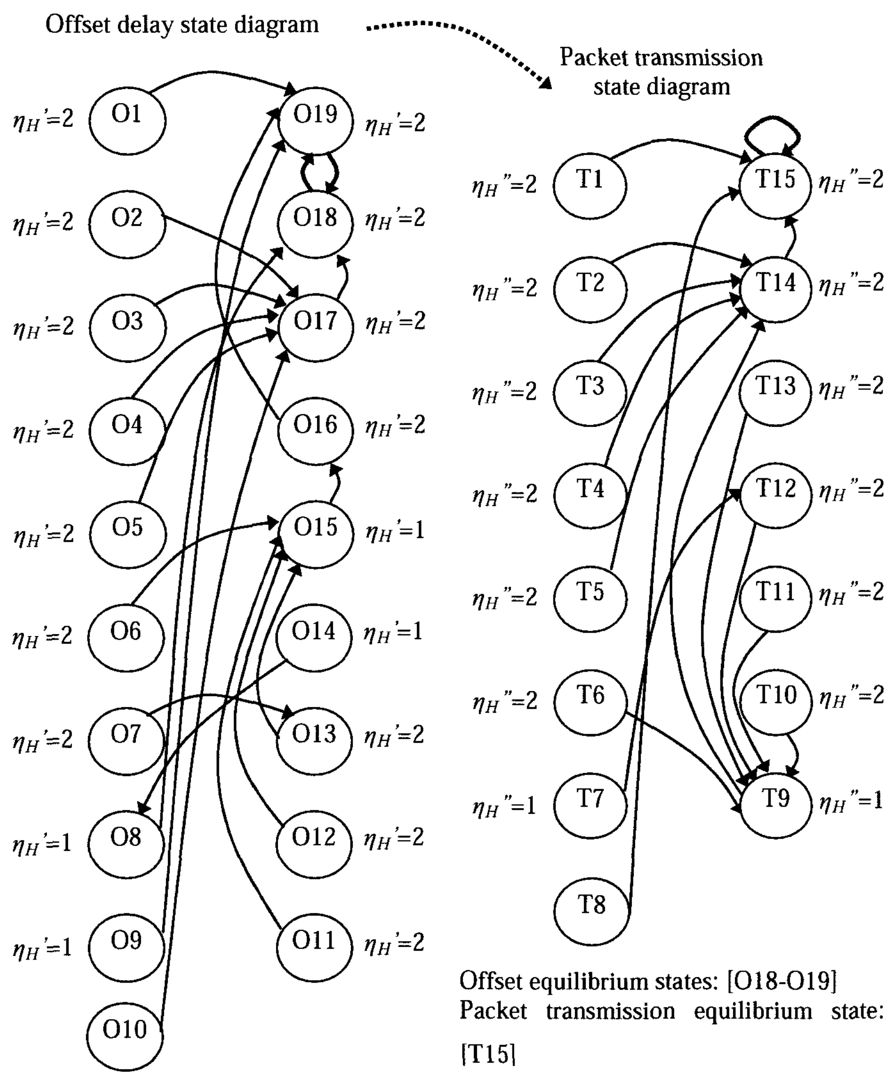

Figure 4.8: State diagrams for RT-MAC with fault states for $T_U = Tx+2Tc$ (that ensures η_H=1) and $T_{AI} < 6Tx+8Tc$.

Figure 4.8 presents the state diagrams with fault states for $T_U = Tx+2Tc$ and $T_{AI} < 6Tx+8Tc$, which are constructed using the 1st, 4th and 6th columns of Tables 4.2 and 4.3. This figure shows that RT-MAC does not remain locked into a fault state; instead, it recovers from a fault state to equilibrium state in the next few frames.

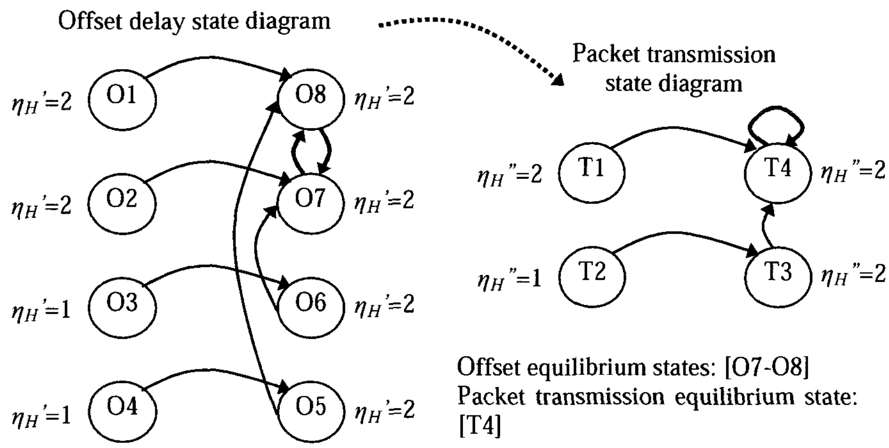

Figure 4.9: State diagrams for RT-MAC with $T_U = Tx+2Tc$ (that ensures η_H=1) and $T_{AI} < 6Tx+8Tc$.

Tables 4.2 and 4.3 include fault states that are useful in analyzing the fault tolerance of RT-MAC protocol. However, in this subsection, we try to find out the lower limit of end-to-end delay of RT-MAC in a normal mode of operation (without fault states). Delay bound analysis of RT-MAC with fault scenario is given in Chapter 5. Hence, the fault states O2, O3, O4, O5, O6, O7, O9, O11, O12, O13 and O14 from Table 4.2, and T2, T3, T4, T5, T6, T7, T8, T10, T11, T12 and T13 from Table 4.3 are removed. The remaining states (O1, O8, O10, O15, O16, O17, O18 and O19) of Table 4.2 are resequenced as O1, O2, O3, O4, O5, O6, O7 and O8. Similarly, the remaining states (T1, T9, T14 and T15) of Table 4.3 are resequenced as T1, T2, T3 and T4. Hence, we get Tables 4.4 and 4.5 for the normal mode of operation of RT-MAC. In Tables 4.4 and 4.5, *Tc(CC-A)* is not mentioned in the fifth column because if a node waits for $2Tc$ or $3Tc$ durations, then *Tc(CC-A)* is taken care of in that duration. Using Tables 4.4 and 4.5, we construct the state diagrams of Figure 4.9 for the normal mode of operation of RT-MAC for η_H=1 with $T_U = Tx+2Tc$. Similarly, Tables 4.6 and 4.7 present offset and packet transmission stats details respectively for $T_U = 2Tx+5Tc$ that ensures η_H=2. Tables 4.8 and 4.9 present offset and packet transmission status details respectively for $T_U = 3Tx+5Tc$ that ensures η_H=3; and Tables 4.10 and 4.11 present offset and packet transmission stats details respectively for $T_U = 4Tx+8Tc$ that ensures η_H=4.

Table 4.4: Offset delay States detail for $T_U = Tx+2Tc$ (that ensures η_H=1) and $T_{AI} < 6Tx+8Tc$ in normal mode of operation in RT-MAC

Current state	Transmission pattern description for the current frame	Starting Node	η_H'	Wait for $2Tc$ / $3Tc$ in the next state by the next starting node	Next state	(Type of current state): Initial conditions
O1	$Tx_0+Tx_1(R)$	N_0	2	$Tc_5+Tc_4+Tc_3 = 3Tc$	O8	(RS): Previous packet transmission starts from N_4 at the beginning of current frame.
O2	$Tx_2+Tx_3(R)$	N_2	2	$Tc_3+Tc_2+Tc_1= 3Tc$	O7	(RS): Previous packet transmission starts from N_6 in current frame.
O3	$Tx_1+Tc_5+Tc_4$	N_1	1	$Tc_3(W)+Tc_3 = 2Tc$	O6	(RS): Previous packet transmission starts from N_5 at the beginning of current frame followed by CC.
O4	$Tx_3+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$	N_3	1	$Tc_1(W)+Tc_1= 2Tc$	O5	(RS): Previous packet transmission starts from N_7 at the beginning of current frame.
O5	$\{Tc_2 \mid Tc_1(W)\} + Tc_1+Tx_0+Tx_1(COH)$	N_0	2	$Tc_5+Tc_4+Tc_3 = 3Tc$	O8	(RS): Previous packet transmission starts from N_4 at the beginning of current frame.
O6	$\{Tc_4 \mid Tc_3(W)\}+Tc_3+Tx_2+Tx_3(COH)$	N_2	2	$Tc_3+Tc_2+Tc_1= 3Tc$	O7	(RS): Previous packet transmission starts from N_6 at the beginning of current frame.
O7	$Tc_3+Tc_2+Tc_1+Tx_0+Tx_1(COH)$	N_0	2	$Tc_5+Tc_4+Tc_3 = 3Tc$	O8	(GS): Previous packet transmission starts from N_4 in current frame but its start is delayed by $3Tc$.
O8	$Tc_5+Tc_4+Tc_3+Tx_2+Tx_3(COH)$	N_2	2	$Tc_3+Tc_2+Tc_1 = 3Tc$	O7	(GS): Previous packet transmission starts from N_6 in current frame but its start is delayed by $3Tc$.

Table 4.5: Packet Transmission delay States Detail for $T_U = Tx+2Tc$ (that ensures η_H=1) and $T_{AI} < 6Tx+8Tc$ in normal mode of operation in RT-MAC

Current state	Transmission pattern description for the current frame	Starting Node	η_H"	Wait for $2Tc$ / $3Tc$ in the next state by the next starting node	Next state	(Type of current state): Initial conditions
T1	$Tx_n+Tx_{n+1}(R)$	N_{even}	2	$\{Tc_{n+1}(CC\text{-}A)$ & $Tc_{n+5}\}$ $+\{Tc_n(CC\text{-}A)$ & $Tc_{n+4}\}$ $+Tc_{n+3} = 3Tc$	T4	(RS): Previous packet transmission starts from the fourth hop neighbor towards sink at the beginning of the current frame.
T2	$Tx_n+\{Tc_{n+4}$ & $Tc_n(CC\text{-}A)\}$ $+\{Tc_{n+3}$ & $Tc_{n-1}(CC\text{-}A)\}$	N_{odd}	1	$Tc_{n+2}(W)$ $+Tc_{n+2} = 2Tc$	T3	(RS): Previous packet transmission starts from the fourth hop neighbor towards the sink at the beginning of the current frame.
T3	$\{Tc_{n+2} \mid Tc_{n+1}(W)\}$ $+Tc_{n+1}+Tx_n+$ $Tx_{n+1}(COH)$	N_{even}	2	$Tc_{n+5}+Tc_{n+4}$ $+Tc_{n+3}$ $= 3Tc$	T4	(RS): Previous packet transmission starts from the fourth hop neighbor towards the sink at the beginning of the current frame.
T4	$\{Tc_{n-1}(CC\text{-}A)$ & / $\mid Tc_{n+3}\}$ + $\{Tc_{n-2}(CC\text{-}A)$ / $\mid Tc_{n+2}\}$ $+Tc_{n+1}+Tx_n$ $+Tx_{n+1}(COH)$	N_{even}	2	$Tc_{n+5}+Tc_{n+4}$ $+Tc_{n+3}$ $= 3Tc$	T4	(GS): Previous packet transmission starts from the fourth hop neighbor towards the sink in current frame but its start is delayed by $3Tc$.

Table 4.6: Offset delay States detail for $T_U = 2Tx+5Tc$ (that ensures η_H=2) and $T_{AI} < 6Tx+8Tc$ in normal mode of operation in RT-MAC

Current state	Transmission pattern description for the current frame	Starting Node	η_H'	Wait for $2Tc$ / $3Tc$ in the next state by the next starting node	Next state	(Type of current state): Initial conditions

O1	$Tx_0+Tx_1+Tc_5+Tc_4$ $+Tc_3+Tx_2(R)+Tx_3(COH)$	N0	4	$Tc_3+Tc_2+Tc_1$ $= 3Tc$	O7	(RS): Previous packet transmission starts from N_4 at the beginning of the current frame.
O2	$Tx_2+Tx_3+Tc_3(CC\text{-}A)$ $+Tc_2(CC\text{-}A)$ $+Tc_1(CC\text{-}A)$ $+Tx_0(R)+Tx_1(COH)$	N2	4	$Tc_5+Tc_4+Tc_3$ $= 3Tc$	O8	(RS): Previous packet transmission starts from N_6 at the beginning of the current frame.
O3	$Tx_1+Tc_5+Tc_4+Tc_3+Tx_2$ $+Tx_3(R)$	N1	3	$Tc_3+Tc_2+Tc_1$ $= 3Tc$	O7	(GS): Previous packet transmission starts from N_5 at the beginning of the current frame.
O4	$Tx_3+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$ $+Tc_1(CC\text{-}A)+$ $Tx_0+Tx_1(R)$	N3	3	$Tc_5+Tc_4+Tc_3$ $= 3Tc$	O8	(GS): Previous packet transmission starts from N_7 at the beginning of the current frame.
O5	$\{Tc_2 \mid Tc_1(W)\}+Tc_1+Tx_0$ $+\{Tx_1$ & $Tc_5\}+Tc_4(W)+Tc_4$ $+Tc_3+Tx_2(CC\text{-}R)$	N0	3	-	O4	(GS): Previous packet transmission starts from N_4 at the beginning of current frame.
O6	$\{Tc_4 \mid Tc_3(W)\}$ + $Tc_3+Tx_2+Tx_3$ $+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$ $+Tc_1(CC\text{-}A)+Tx_0(CC\text{-}R)$	N2	3	-	O3	(GS): Previous packet transmission starts from N_6 at the beginning of the current frame.
O7	$Tc_3+Tc_2+Tc_1+Tx_0+Tx_1+$ Tc_5 $+Tc_4$	N0	2	$Tc_3(W)+Tc_3$ $= 2Tc$	O6	(GS): Previous packet transmission starts from N_4 in the current frame but its start is delayed by $3Tc$.
O8	$Tc_5+Tc_4+Tc_3+Tx_2+Tx_3$ $+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$	N2	2	$Tc_1(W)+Tc_1$ $=2Tc$	O5	(GS): Previous packet transmission starts from N_6 in the current frame but its start is delayed by $3Tc$.

Table 4.7: Packet Transmission delay States Detail for $T_U = 2Tx+5Tc$ (that ensures η_H=2) and $T_{AI} < 6Tx+8Tc$ in normal mode of operation in RT-MAC.

Current state	Transmission pattern description for the current frame	Starting Node	η_H"	Wait for $2Tc$ / $3Tc$ in the next state by the next starting node	Next state	(Type of current state): Initial conditions
T1	Tx_n+Tx_{n+1} $+\{Tc_{n+1}(CC\text{-}A)\ \&\ Tc_{n+5}\}$ $+\{Tc_n(CC\text{-}A)\ \&\ Tc_{n+4}\}$ $+Tc_{n+3}+Tx_{n+2}(R)$ $+Tx_{n+3}(COH)$	N_{even}	4	$Tc_{n+7}+Tc_{n+6}$ $+Tc_{n+5}$ $= 3Tc$	T4	(RS): Previous packet transmission starts from the fourth hop neighbor towards the sink at the beginning of the current frame.
T2	$Tx_n+\{Tc_{n-1}(CC\text{-}A)$ & $Tc_{n+4}\}$ $+\{Tc_{n-2}(CC\text{-}A)\ \&\ Tc_{n+3}\}$ $+Tc_{n+2}+Tx_{n+1}$ $+Tx_{n+2}(R)$	N_{odd}	3	$Tc_{n+5}+Tc_{n+4}$ $+Tc_{n+3}$ $= 3Tc$	T4	(GS): Previous packet transmission starts from the fourth hop neighbor towards the sink at the beginning of the current frame.
T3	$\{Tc_{n+2} \mid Tc_{n+1}(W)\}$ $+Tc_{n+1}$ $+Tx_n+\{Tx_{n+1}\ \&\ Tc_{n+5}\}$ $+\{Tc_{n+1}(CC\text{-}A)$ $\&\ Tc_{n+4}(W)\}$ $+\{Tc_n(CC\text{-}A)\ \&\ Tc_{n+4}\}$ $+\ Tc_{n+4}+Tx_{n+2}(CC\text{-}R)$	N_{even}	3	-	T2	(GS): Previous packet transmission starts from the fourth hop neighbor towards the sink at the beginning of the current frame.
T4	$\{Tc_{n-1}(CC\text{-}A)\ \&\ Tc_{n+3}\}\ +$ $\{Tc_{n-2}(CC\text{-}A)\ \&\ Tc_{n+2}\}$ $+Tc_{n+1}+Tx_n\ +Tx_{n+1}$ $+\ \{Tc_{n+1}(CC\text{-}A)\ \&\ Tc_{n+5}\}$ $+\ \{Tc_n(CC\text{-}A)\ \&\ Tc_{n+4}\}$	N_{even}	2	$Tc_{n+4}(W)$ $+Tc_{n+3}$ $= 2Tc$	T3	(GS): Previous packet transmission starts from the fourth hop neighbor towards the sink in the current frame but its start is delayed by $3Tc$.

Table 4.8: Offset delay States detail for $T_U = 3Tx+5Tc$ (that ensures η_H=3) and $T_{AI} < 6Tx+8Tc$ in normal mode of operation in RT-MAC

Current state	Transmission pattern description for the current frame	Starting Node	η_H'	Wait for $2Tc$ / $3Tc$ in the next state by the next starting node	Next state	(Type of current state): Initial conditions
O1	$Tx_0+Tx_1+Tc_5+Tc_4+Tc_3+Tx_2+Tx_3(R)$	N0	4	$Tc_3+Tc_2+Tc_1 = 3Tc$	O7	(RS): Previous packet transmission starts from N_4 at the beginning of the current frame.
O2	$Tx_2+Tx_3+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)+Tc_1(CC\text{-}A)+Tx_0+Tx_1(R)$	N2	4	$Tc_5+Tc_4+Tc_3 = 3Tc$	O8	(RS): Previous packet transmission starts from N_6 at the beginning of the current frame.
O3	$Tx_1+Tc_5+Tc_4+Tc_3+Tx_2+Tx_3+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$	N1	3	$Tc_1(W)+Tc_1 = 2Tc$	O5	(RS): Previous packet transmission starts from N_5 at the beginning of the current frame.
O4	$Tx_3+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)+Tc_1(CC\text{-}A)+Tx_0+Tx_1+Tc_5$	N3	3	$Tc_3(W)+Tc_3 = 2Tc$	O6	(RS): Previous packet transmission starts from N_7 at the beginning of the current frame.
O5	$\{Tc_2 \mid Tc_1(W)\}+Tc_1+Tx_0+\{Tx_1 \& Tc_5\}+Tc_4(W)+Tc_4+Tc_3+Tx_2+Tx_3(COH)$	N0	4	$Tc_3+Tc_2+Tc_1 = 3Tc$	O7	(RS): Previous packet transmission starts from N_4 at the beginning of the current frame.
O6	$\{Tc_4 \mid Tc_3(W)\}+Tc_3+Tx_2+Tx_3+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)+Tc_1(CC\text{-}A)+Tx_0+Tx_1(COH)$	N2	4	$Tc_5+Tc_4+Tc_3 = 3Tc$	O8	(RS): Previous packet transmission starts from N_6 at the beginning of the current frame.
O7	$Tc_3+Tc_2+Tc_1+Tx_0+Tx_1+Tc_5+Tc_4+Tc_3+Tx_2+Tx_3(COH)$	N0	4	$Tc_3+Tc_2+Tc_1 = 3Tc$	O7	(GS): Previous packet transmission starts from N_4 in the current frame but its

						start is delayed by $3Tc$.
O8	$Tc_5+Tc_4+Tc_3+Tx_2+Tx_3$ $+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$ $+Tc_1(CC\text{-}A)+$ $Tx_0+Tx_2(COH)$	N2	4	$Tc_5+Tc_4+Tc_3$ $= 3Tc$	O8	(GS): Previous packet transmission starts from N_6 in the current frame but its start is delayed by $3Tc$.

Table 4.9: Packet Transmission delay States Detail for $T_U = 3Tx+5Tc$ (that ensures η_H=3) and $T_{AI} < 6Tx+8Tc$ in normal mode of operation in RT-MAC

Current state	Transmission pattern description for the current frame	Starting Node	η_H"	Wait for $2Tc$ / $3Tc$ in the next state by the next starting node	Next state	(Type of current state): Initial conditions
T1	Tx_n+Tx_{n+1} $+\{Tc_{n+1}(CC\text{-}A)\ \&\ Tc_{n+5}\}$ $+\{Tc_n(CC\text{-}A)\ \&\ Tc_{n+4}\}$ $+Tc_{n+3}+Tx_{n+2}+Tx_{n+3}(R)$	N_{even}	4	$Tc_{n+7}+Tc_{n+6}$ $+Tc_{n+5}$ $= 3Tc$	T4	(RS): Previous packet transmission starts from fourth hop neighbor towards the sink at the beginning of the current frame.
T2	$Tx_n+\{Tc_n(CC\text{-}A)$ & $Tc_{n+4}\}$ $+\{Tc_{n-1}(CC\text{-}A)\ \&\ Tc_{n+3}\}$ $+Tc_{n+2}+Tx_{n+1}+Tx_{n+2}+Tc$ $_{n+6}$ $+Tc_{n+5}$	N_{odd}	3	$Tc_{n+4}(W)$ $+Tc_{n+4}$ $= 2Tc$	T3	(RS): Previous packet transmission starts from the fourth hop neighbor at the beginning of the current frame.
T3	$\{Tc_{n+2} \mid Tc_{n+1}(W))\}$ $+Tc_{n+1}+Tx_n+\{Tx_{n+1}$ & $Tc_{n+5}\}+\{Tc_{n+1}(CC\text{-}A)$ & $Tc_{n+4}(W)\}+\{Tc_n(CC\text{-}A)$ $\&\ Tc_{n+4}\}+Tc_{n+3}+Tx_{n+2}$ $+Tx_{n+3}(COH)$	N_{even}	4	$Tc_{n+7}+Tc_{n+6}$ $+Tc_{n+5}$ $= 3Tc$	T4	(RS): Previous packet transmission starts from the fourth hop neighbor towards the sink at the beginning of the current frame.
T4	$\{Tc_{n-1}(CC\text{-}A)\ \&\ Tc_{n+3}\}$ $+\{Tc_{n-2}(CC\text{-}A)\ \&\ Tc_{n+2}\}$	N_{even}	4	$Tc_{n+7}+Tc_{n+6}$ $+Tc_{n+5}$	T4	(GS): Previous packet transmission starts from the

	$+Tc_{n+1}+Tx_n+Tx_{n+1}$ $+\{Tc_{n+1}(CC\text{-}A)\ \&\ Tc_{n+5}\}$ $+\{Tc_n(CC\text{-}A)\ \&\ Tc_{n+4}\}$ $+Tc_{n+3}+Tx_{n+2}+Tx_{n+3}(COH)$			$= 3Tc$		fourth hop neighbor in the current frame but its start is delayed by $3Tc$.

Table 4.10: Offset delay States detail for $T_U = 4Tx+8Tc$ (that ensures η_H=4) and $T_{AI} < 6Tx+8Tc$ in normal mode of operation in RT-MAC

Current state	Transmission pattern description for the current frame	Starting Node	η_H'	Wait for $2Tc$ / $3Tc$ in the next state by the next starting node	Next state	(Type of current state): Initial conditions
O1	$Tx_0+Tx_1+Tc_5+Tc_4$ $+Tc_3+Tx_2+Tx_3+Tc_3(CC\text{-}A)$ $+Tc_2(CC\text{-}A)+Tc_1(CC\text{-}A)$ $+Tx_0(R)+Tx_1(COH)$	N0	6	$Tc_5+Tc_4+Tc_3 = 3Tc$	O8	(RS): Previous packet transmission starts from N_4 at the beginning of the current frame.
O2	$Tx_2+Tx_3+Tc_3(CC\text{-}A)$ $+Tc_2(CC\text{-}A)+Tc_1(CC\text{-}A)$ $+Tx_0+Tx_1+Tc_5+Tc_4+Tc_3$ $+Tx_2(R)+Tx_3(COH)$	N2	6	$Tc_3+Tc_2+Tc_1 = 3Tc$	O7	(RS): Previous packet transmission starts from N_6 at the beginning of the current frame.
O3	$Tx_1+Tc_5+Tc_4+Tc_3+Tx_2$ $+Tx_3+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$ $+Tc_1(CC\text{-}A)+Tx_0+Tx_1(R)$	N1	5	$Tc_5+Tc_4+Tc_3 = 3Tc$	O8	(GS): Previous packet transmission starts from N_5 at the beginning of the current frame.
O4	$Tx_3+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$ $+Tc_1(CC\text{-}A)+Tx_0+Tx_1+Tc_5$ $Tc_4+Tc_3+Tx_2+Tx_3(R)$	N3	5	$Tc_3+Tc_2+Tc_1 = 3Tc$	O7	(GS): Previous packet transmission starts from N_7 at the beginning of the current frame.

O5	$\{Tc_2 \mid Tc_1(W)\}+Tc_1+Tx_0+\{Tx_1$ & $Tc_5\}+Tc_4(W)+Tc_4+Tc_3+Tx_2$ $+Tx_3+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$ $+Tc_1(CC\text{-}A)+Tx_0(CC\text{-}R)$	N0	5	-	O3	(GS): Previous packet transmission starts from N_4 at the beginning of the current frame.
O6	$\{Tc_4 \mid Tc_3(W)\} + Tc_3+Tx_2+Tx_3$ $+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$ $+Tc_1(CC\text{-}A)+Tx_0+Tx_1+Tc_5$ $+Tc_4+Tc_3+Tx_2(CC\text{-}R)$	N2	5	-	O4	(GS): Previous packet transmission starts from N_6 at the beginning of the current frame.
O7	$Tc_3+Tc_2+Tc_1+Tx_0+Tx_1+Tc_5$ $+Tc_4+Tc_3+ Tx_2+Tx_3$ $+ Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$	N0	4	$Tc_1(W)+Tc_1$ $= 2Tc$	O5	(GS): Previous packet transmission starts from N_4 in the current frame but its start is delayed by $3Tc$.
O8	$Tc_5+Tc_4+Tc_3+Tx_2+Tx_3$ $+Tc_3(CC\text{-}A)+Tc_2(CC\text{-}A)$ $+Tc_1(CC\text{-}A)+$ $Tx_0+Tx_2+Tc_5$ $+Tc_4$	N2	4	$Tc_3(W)+Tc_3$ $= 2Tc$	O6	(GS): Previous packet transmission starts from N_6 in the current frame but its start is delayed by $3Tc$.

Table 4.11: Packet Transmission delay States Detail for $T_U = 4Tx+8Tc$ (that ensures η_H=4) and $T_{AI} < 6Tx+8Tc$ in normal mode of operation in RT-MAC

Current state	Transmission pattern description for the current frame	Starting Node	η_H"	Wait for $2Tc$ / $3Tc$ in the next state by the next starting node	Next state	(Type of current state): Initial conditions
T1	Tx_n+Tx_{n+1} $+\{Tc_{n+1}(CC\text{-}A)$ & $Tc_{n+5}\}$	N_{even}	6	$Tc_{n+9}+Tc_{n+8}$ $+Tc_{n+7}$	T4	(RS): Previous packet transmission starts from the

	$+\{Tc_n(CC\text{-}A)\ \&\ Tc_{n+4}\}$ $+Tc_{n+3}+Tx_{n+2}+Tx_{n+3}$ $+\{Tc_{n+3}(CC\text{-}A)\ \&\ Tc_{n+7}\}$ $+\{Tc_{n+2}(CC\text{-}A)\ \&\ Tc_{n+6}\}$ $+Tc_{n+5}+Tx_{n+4}(R)+Tx_{n+5}(COH)$			$= 3Tc$		fourth hop neighbor towards the sink at the beginning of the current frame.
T2	$Tx_n+\{Tc_n(CC\text{-}A)\ \&\ Tc_{n+4}\}$ $+\{Tc_{n-1}(CC\text{-}A)\ \&\ Tc_{n+3}\}$ $+Tc_{n+2}+Tx_{n+1}+Tx_{n+2}$ $+\{Tc_{n+2}(CC\text{-}A)\ \&\ Tc_{n+6}\}$ $+\{Tc_{n+1}(CC\text{-}A)\ \&\ Tc_{n+5}\}$ $+Tc_{n+4}+Tx_{n+3}+Tx_{n+4}(R)$	N_{odd}	5	$Tc_{n+8}+Tc_{n+7}$ $+Tc_{n+6}$ $= 3Tc$	T4	(GS): Previous packet transmission starts from the fourth hop fourth hop neighbor at the beginning of the current frame.
T3	$\{Tc_{n+2} \mid Tc_{n+1}(W))\}$ $+Tc_{n+1}+Tx_n+\{Tx_{n+1}\ \&\ Tc_{n+5}\}+\{Tc_{n+1}(CC\text{-}A)\ \&\ Tc_{n+4}(W)\}+\{Tc_n(CC\text{-}A)\ \&\ Tc_{n+4}\}+Tc_{n+3}+Tx_{n+2}$ $+\{Tx_{n+3}\ \&\ Tc_{n+7}\}$ $+\{Tc_{n+3}(CC\text{-}A)\ \&\ Tc_{n+6}(W)\}$ $+\{Tc_{n+2}(CC\text{-}A)\ \&\ Tc_{n+6}\}$ $+Tc_{n+5}+Tx_{n+4}(CC\text{-}R)$	N_{even}	5	-	T2	(GS): Previous packet transmission starts from the fourth hop neighbor towards the sink at the beginning of the current frame.
T4	$\{Tc_{n-1}(CC\text{-}A)\ \&\ Tc_{n+3}\}$ $+\{Tc_{n-2}(CC\text{-}A)\ \&\ Tc_{n+2}\}$ $+Tc_{n+1}+Tx_n+Tx_{n+1}$ $+\{Tc_{n+1}(CC\text{-}A)\ \&\ Tc_{n+5}\}$ $+\{Tc_n(CC\text{-}A)\ \&\ Tc_{n+4}\}$ $+Tc_{n+3}+Tx_{n+2}+Tx_{n+3}$ $+\{Tc_{n+3}(CC\text{-}A)\ \&\ Tc_{n+7}\}$ $+\{Tc_{n+2}(CC\text{-}A)\ \&\ Tc_{n+6}\}$	N_{even}	4	$Tc_{n+5}(W)$ $+Tc_{n+5}$ $= 2Tc$	T3	(GS): Previous packet transmission starts from the fourth hop neighbor in the current frame but its start is delayed by $3Tc$.

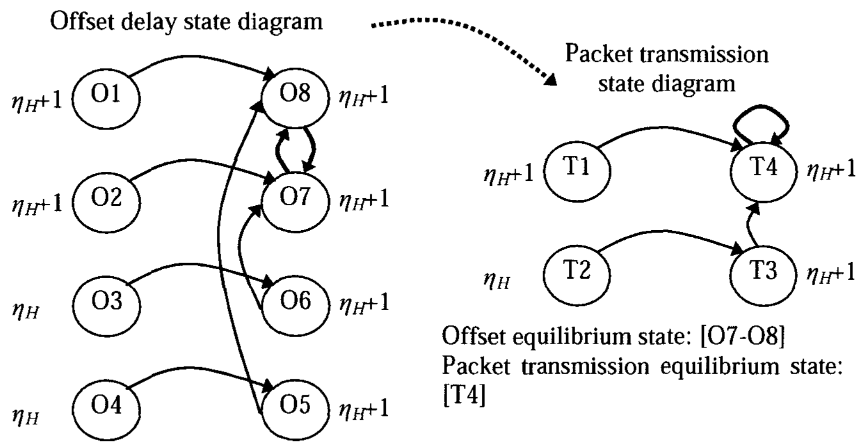

Figure 4.10: State diagrams for RT-MAC with $T_U = \eta_H Tx + \{(3\eta_H+1)/2\} Tc$ (that ensures η_H=1,5, 9, ...) and $T_{IA} < 6Tx+8Tc$.

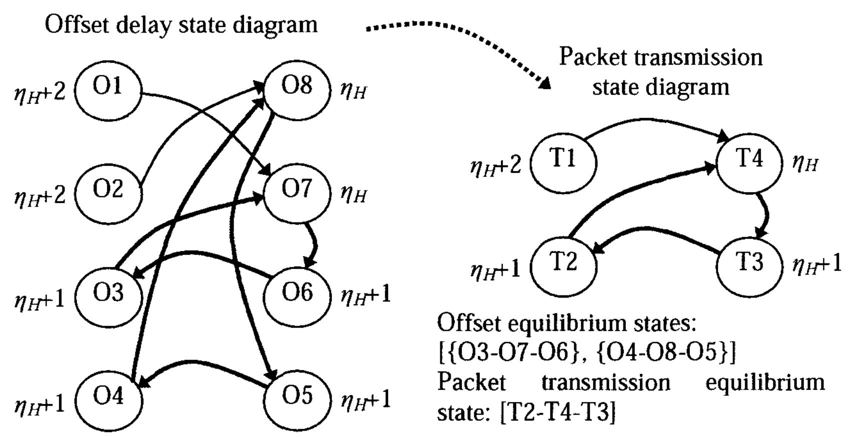

Figure 4.11: State diagrams for RT-MAC with $T_U = \eta_H Tx + \{(3\eta_H/2)+2\} Tc$ (that ensures η_H=2,6,10 ....) and $T_{AI} < 6Tx+8Tc$.

As discussed earlier, the state diagrams for η_H = 1 can be generalized for η_H = 5, 9, ... and so on due to the 4 hop symmetry of the transmission pattern in RT-MAC. Therefore, Figure 4.10 presents state diagrams for η_H = 1, 5, 9, ... and so on. Following a similar procedure, we can find state diagrams for η_H = 2, 6, 10, ..., η_H = 3, 7, 11, ... and η_H = 4, 8, 12, ... as shown in Figures 4.11 to 4.13 respectively using Tables 4.6 to 4.11.

Now, the lower bound of end-to-end delay for the ON duration $T_U = \eta_H Tx + \{(3\eta_H+1)/2\} Tc$ (that ensures η_H=1, 5, 9, ...) can be calculated using state diagrams

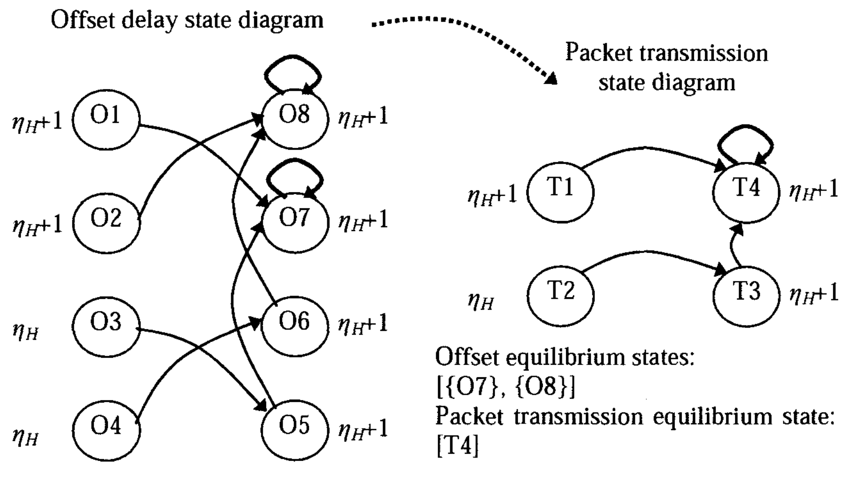

Figure 4.12: State diagrams for RT-MAC with $T_U = \eta_H Tx + \{(3\eta_H+1)/2\} Tc$ (that ensures η_H=3,7,11 ...) and $T_{AI} < 6Tx+8Tc$.

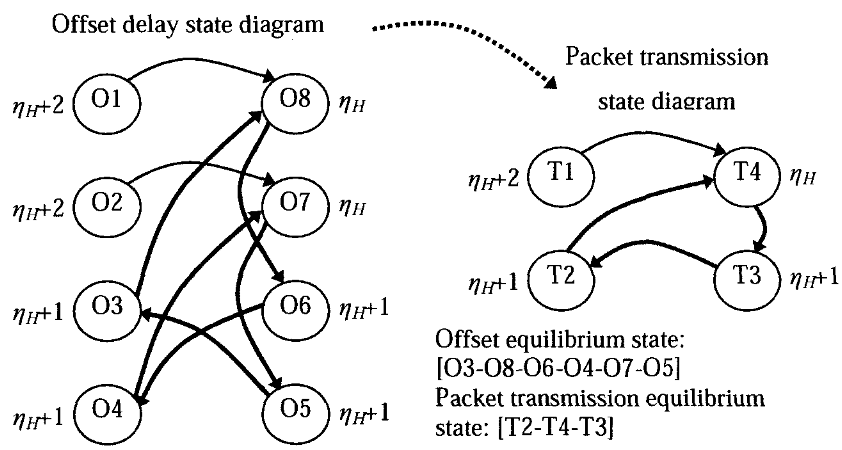

Figure 4.13: State diagrams for RT-MAC with $T_U = \eta_H Tx + \{(3\eta_H/2)+2\} Tc$ (that ensures η_H=4,8,12...) and $T_{AI} < 6Tx+8Tc$.

as shown in Figure 4.9. The first data packet will start with the T1 packet transmission state in the first frame of Figure 4.9. From the second frame onward, it remains in a T4 equilibrium state until some fault occurs. Thus, the first data packet transmission follows the pattern of $\{\eta_H+1, (\eta_H+1), (\eta_H+1), ...\}$ hops per frame until it reaches the destination. Therefore, packet transmission delay for the first data packet can be calculated using (10a) and (10b) with η_H in the denominator replaced by η_H+1.

$$T_D(1,n) = \left\lfloor \frac{n}{\eta_H+1} \right\rfloor T_f + [n_L\, Tx + (n_L -2)\, Tc]$$

for η_H= 1,5, 9, ... and even n_L (18a)

$$T_D(1,n) = \left\lfloor \frac{n}{\eta_H + 1} \right\rfloor T_f + [n_L\, Tx + (n_L - 1)Tc]$$

for $\eta_H = 1, 5, 9, \ldots$ and odd n_L , (18b)

where, n_L is mod(n, (η_H +1)).

However, in the case of the second data packet, the offset delay state diagram of Figure 4.10 is used to calculate the offset delay due to the first data packet. The offset delay calculation starts from the O1 offset delay state and follows the pattern of {η_H+1, (η_H+1, η_H+1), (η_H+1, η_H+1), ...} hops per frame till the start of the current data packet. Referring to Figure 3.8, an offset delay of 4 hops transmission time is added to the packet transmission time of the second data packet and so on.

Thus, for the m^{th} data packet, the lower limit of end-to-end delay with $T_U = \eta_H Tx + \{(3\eta_H+1)/2\}Tc$ is given by

$$T_D(m,n) = \left\lfloor \frac{(4(m-1)+n)}{\eta_H + 1} \right\rfloor T_f + [n_L\, Tx + (n_L - 2)Tc]$$

for $\eta_H = 1, 5, 9, \ldots$ and even n_L, (19a)

and

$$T_D(m,n) = \left\lfloor \frac{(4(m-1)+n)}{\eta_H + 1} \right\rfloor T_f + [n_L\, Tx + (n_L - 1)Tc]$$

for $\eta_H = 1, 5, 9, \ldots$ and odd n_L , (19b)

where, n_L is mod(4(m-1)+n, (η_H +1)).

Similarly, the lower limit of the end-to-end delay for the m^{th} data packet for other values of ON durations that ensure $\eta_H = 2, 6, 10, \ldots$, $\eta_H = 3, 7, 11, \ldots$ and $\eta_H = 4, 8, 12, \ldots$ can be derived from the state diagrams in Figures 4.11 to 4.13 respectively.

4.6 Lower limit of end-to-end delay in duty cycle mode of operation of RT-MAC for hop separation ≥ 6

As mentioned earlier in Section 4.4, for $T_{AI} \geq 6Tx + 8Tc$, the offset delay depends upon the actual data packet arrival interval at the MAC layer. Hence, the offset delay calculation for the lower limit will be the same as in Section 4.4. It will not depend on the offset state diagrams of Figures 4.10 to 4.13. However, once the transmission of the current data packet starts, then the packet transmission state diagrams of Figures 4.10 to 4.13 need to be referred in order to find out the exact packet transmission pattern. In the case of ON duration $T_U = \eta_H Tx + \{(3\eta_H+1)/2\}Tc$ (that ensures η_H=1,5, 9, ...), the packet transmission delay is given by (18a) and (18b). Thus, following a procedure similar to that of Section 4.4, the lower limit of the end-to-end delay for the m^{th} data packet with $T_U = \eta_H Tx + \{(3\eta_H+1)/2\}Tc$ is given by

$$T_D(m,n) = \left[a + \left\lfloor \frac{n-n'}{\eta_H+1} \right\rfloor \right] T_f + [n_L\, Tx + (n_L - 2)Tc]$$

for η_H= 1,5, 9, ... and even n_L, (20a)

and

$$T_D(m,n) = \left[a + \left\lfloor \frac{n-n'}{\eta_H+1} \right\rfloor \right] T_f + [n_L\, Tx + (n_L - 1)Tc]$$

for η_H= 1,5, 9, ... and odd n_L, (20b)

where, n_L is mod($(n - n')$, $(\eta_H + 1)$) and n' is given by (15).

Similarly, the lower limit of the end-to-end delay for the m^{th} data packet for η_H = 2, 6, 10, ..., η_H = 3, 7, 11, ... and η_H = 4, 8, 12, ... can be derived by following similar procedures as explained above. In general, the lower and upper delay bounds of the end-to-end delay for a given η_H differ significantly for lower duty cycles.

4.7 Summary of the chapter

This chapter presents the end-to-end delay bounds with and without duty cycle operation of RT-MAC. Specifically, the delay bounds are presented for 6 cases, which consist of lower and upper delay bounds for dependent (hop separation < 6) and independent (hop separation ≥ 6) transmission of data packets in a stream.

This chapter presents delay analysis of RT-MAC when no fault condition is considered in the network. However, the next chapter presents RT-MAC protocol in more realistic conditions, where fault conditions are also considered.

Chapter 5: Delay Bound Analysis in Fault Scenario in RT-MAC protocol

In Sections 4.5 and 4.6 of the last chapter, the lower delay bounds of RT-MAC are calculated using the state diagrams for normal mode of operation RT-MAC protocol. State diagrams for the normal mode do not include fault states. However, to analyze the behavior of RT-MAC in realistic scenario, the fault states need to be considered using stochastic modeling of RT-MAC protocol. Thus, this chapter presents delay bound analysis of RT-MAC protocol under fault conditions in the single-stream scenario. It is observed that Markov modeling is suitable to analyze the behavior of RT-MAC protocol for the following reasons.

a) RT-MAC is a memory less MAC protocol. The immediate future state depends on the current state in RT-MAC.

b) RT-MAC has well defined states, i.e., general states, rare states and fault states. A packet transmission pattern in a network neighborhood in one frame duration falls on one of these states.

c) Time step needed to change one state into another is also well defined. It is equal to the frame duration in RT-MAC protocol.

Thus, the objective of Markov analysis of RT-MAC is to predict mean end-to-end delay and mean throughput when there are faults in WSN.

5.1 Markov analysis of RT-MAC

We will start our analysis with Figure 4.8 of the last chapter. This figure shows the state diagram with fault state with ON duration $T_U = Tx+2Tc$ that ensures at least one hops transmission per frame (i.e., η_H=1). However, the same procedure can be followed for η_H=2, 3 and 4. Further, due to 4 hops symmetry in transmission patter of RT-MAC protocol, the results of η_H=1 can be generalized for η_H=1, 5, 9... etc. The same is true for η_H = 2, 6, 10, ..., η_H = 3, 7, 11, ... and η_H = 4, 8, 12, ...etc.

As our objective in this Markov analysis is to find mean delay and mean throughput, we need to represent the states of Figure 4.8 differently such that it gives information about these performance parameters. Specifically, to calculate mean delay, we need to know the number of hops traversal by a data packet in any state. This will eventually give information about the number of state changes (i.e. number of frames) needed to reach a data packet to the destination. In view of this, we need to do the following changes into Tables 4.2 and 4.3.

1) We group the states of Tables 4.2 and 4.3 as per the starting node of a packet transmission at the beginning of a frame. Here, the fourth column of Tables 4.2 and 4.3 give the number of hops traversal for any state.

2) We remove the third, fifth and seventh column of Tables 4.2 and 4.3 and reorder the states as per the starting node. The sequence number of offset and packet transmission state starts with O0 and T0 (instead of O1 and T1) respectively to keep the notations consistent with general Markov transition matrix notations.

3) We add one extra column at the end of the modified tables (given below) which shows the number of current states (for a given starting node) that goes to the same state in the next frame. It is calculated by counting all the rows for any current state with a particular starting node that have the same number of hop traversal (i.e., η_H' in Table 4.2 and η_H'' in Table 4.3); thus, it includes all the general, rare and faulty current states, which cause the transition into the same future state. These number of current states give us typical transition probabilities, though actual transition probabilities depend upon the real conditions such as battery life distribution of sensor nodes [9], [45], and [48], characteristics of the protocol, and surrounding environment that affects the communication of sensor nodes. In general, it requires long run time data to fine tune the state transition probabilities.

4) We added an extra state transition condition for failed states. The state that does not change after one time step is called failed state. Thus, the number of hop traversed in this case will be 0. A failed state may recover to some valid state (general, rare or other fault

state), for example in cases when involved nodes are operating near the battery threshold. However, it may not recover from the failed state if the involved nodes die permanently.

5) In case of packet transmission state of Table 4.3, there seems to be only two valid states as there are only two possibilities of the starting node, i.e., N_{even} and N_{odd} at the beginning of a frame. However, a closer look into the states reveals that there are more states needed to represent all scenarios. The reason for this is that the two hop traversal will result in the same state, and a failed state also results in the same state. Hence there is a need to separately represent the failed state. Hence, in total, four states are needed to represent packet transmission states of Table 4.3 to carryout Markov analysis.

Thus, Tables 5.1 and 5.2 show the modified offset and packet transmission state tables for η_H=1.

Table 5.1: Modified Offset delay States detail for $T_U = Tx+2Tc$ (that ensures η_H=1) and $T_{AI} < 6Tx+8Tc$ in fault scenario

Current state	Starting Node	Number of hop traversal in the current state (η_H')	Next state	Number of current states that reach to the same next state
O0	N_0	0	O0	1
		1	O1	7
		2	O2	1
		3	O3	1
O1	N_1	0	O1	1
		1	O2	3
		2	O3	3
O2	N_2	0	O2	1

		2	O0	2
O3	N_3	1	O3	1
		2	O0	1

Table 5.2: Modified Packet Transmission delay States Detail for $T_U = Tx+2Tc$ (that ensures η_H=1) and $T_{AI} < 6Tx+8Tc$ in fault scenario

Current state	Starting Node	Number of hop traversal in the current state (η_H")	Next state	Number of current states that reach to the same next state
T0	N_{even}	0	T5	1
		1	T1	1
		2	T0	7
		3	T2	1
T1	N_{odd}	0	T4	1
		1	T0	2
		2	T1	4
T2	N_{even}	0	T3	1
		1	T0	1
T3	N_{even}	0	T3	1
		1	T0	1
T4	N_{odd}	0	T4	1
		1	T0	2

		2	T1	4
T5	N_{even}	0	T5	1
		1	T1	1
		2	T0	7
		3	T2	1

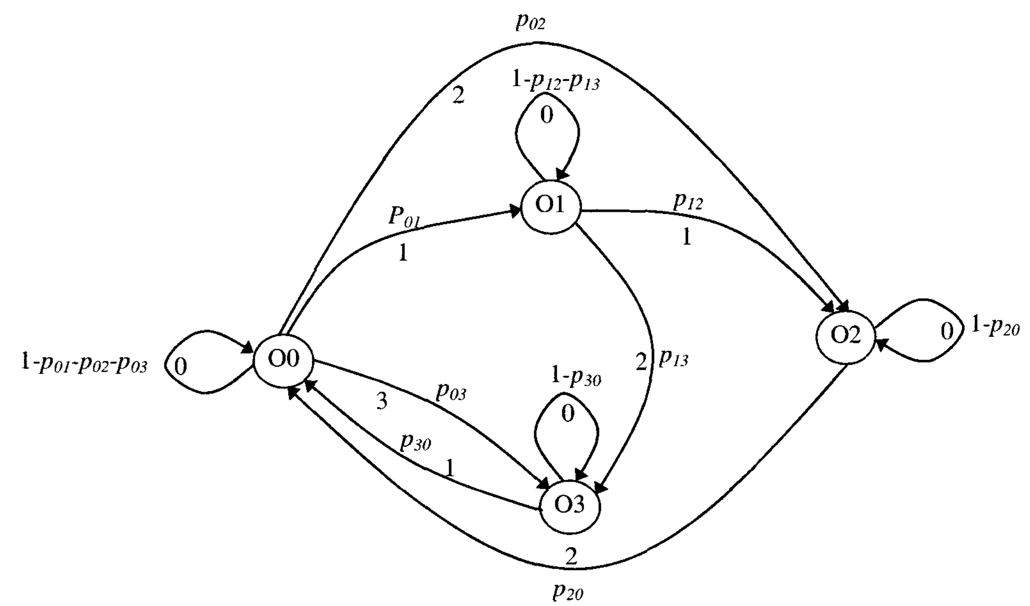

Figure 5.1: Offset state transition diagram

Using Tables 5.1 and 5.2, the offset state and packet transmission state transition diagrams can be constructed. These are shown in Figures 5.1 and 5.2 respectively.

In Figure 5.1, p_{01} represents the offset state transition probability of state O0 to go to the state O1. The same explanation applies to p_{02}, p_{03}, p_{04}, etc. where subscript shows the transition from a particular state to another state. Similarly, in Figure 5.2, q_{01} represents the packet transmission state transition probability of state T0 to go to the state T1. The number associated with each branch shows the number of hop traversal in each state transition. In Figure 5.2, state T5 represents the failed state, which is a mirror reflection

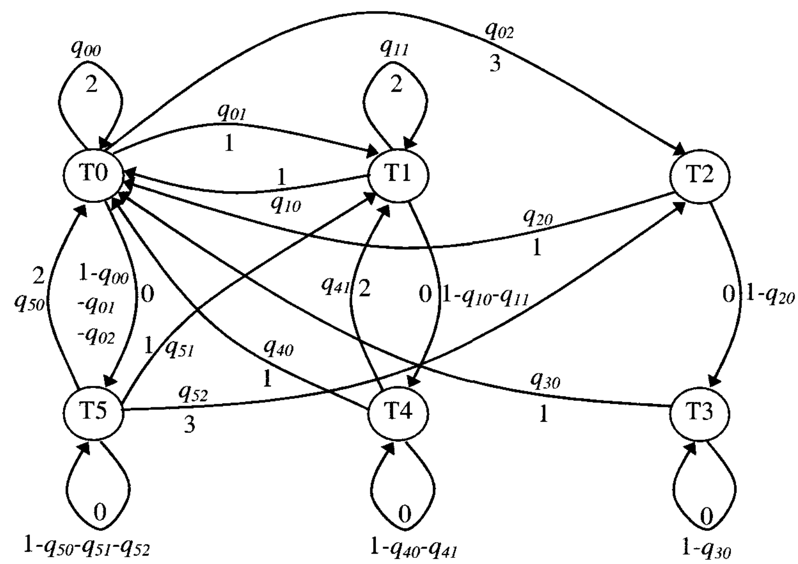

Figure 5.2: Packet transmission state transition diagram

of T0 general state. It signifies that T0 and T5 are the same in the sense that if T0 remains in the same state after one time step, and thus it is represented by a separate failed state, which is T5 in Figure 5.2. Here, T0 and T5 are needed separately because T0 provides 2 hops transmission in one time step, and T5 does not provide any transmission at all after one time step. Similarly, in Figure 5.2, T4 represents the failed state, which is a mirror reflection of T1 state.

Using Figure 5.1, the following offset state discrete time Markov chain (DTMC) transition matrix can be constructed.

$$\begin{bmatrix} 1-p_{01}-p_{02}-p_{03} & p_{01} & p_{02} & p_{03} \\ 0 & 1-p_{12}-p_{13} & p_{12} & p_{13} \\ p_{20} & 0 & 1-p_{20} & 0 \\ p_{30} & 0 & 0 & 1-p_{30} \end{bmatrix}$$

Similarly, using Figure 5.2, we get the following packet transmission DTMC state transition matrix.

$$\begin{bmatrix} q_{00} & q_{01} & q_{02} & 0 & 0 & 1-q_{00}-q_{01}-q_{02} \\ q_{10} & q_{11} & 0 & 0 & 1-q_{10}-q_{11} & 0 \\ q_{20} & 0 & 0 & 1-q_{20} & 0 & 0 \\ q_{30} & 0 & 0 & 1-q_{30} & 0 & 0 \\ q_{40} & q_{41} & 0 & 0 & 1-q_{40}-q_{41} & 0 \\ q_{50} & q_{51} & q_{52} & 0 & 0 & 1-q_{50}-q_{51}-q_{52} \end{bmatrix}$$

Here, we present the steady state analysis of DTMC. The steady state probability vectors of offset state DTMC and packet transmission state DTMC are $\pi = [\pi_0\ \pi_1\ \pi_2\ \pi_3]$ and $v = [v_0\ v_1\ v_2\ v_3\ v_4\ v_5]$ respectively.

Now, the state probabilities of offset states can be obtained using

$$[\pi_0\ \pi_1\ \pi_2\ \pi_3] \begin{bmatrix} 1-p_{01}-p_{02}-p_{03} & p_{01} & p_{02} & p_{03} \\ 0 & 1-p_{12}-p_{13} & p_{12} & p_{13} \\ p_{20} & 0 & 1-p_{20} & 0 \\ p_{30} & 0 & 0 & 1-p_{30} \end{bmatrix} = [\pi_0\ \pi_1\ \pi_2\ \pi_3]. \tag{20}$$

Additionally, we also have the following equation for the steady state condition,

$$\pi_0 + \pi_1 + \pi_2 + \pi_3 = 1 \tag{21}$$

From Equation (20), following equations can be written as

$$\pi_0 (1-p_{01}-p_{02}-p_{03}) + \pi_2 p_{20} + \pi_3 p_{30} = \pi_0, \tag{22}$$

$$\pi_0 p_{01} + \pi_1 (1-p_{12}-p_{13}) = \pi_1,$$

$$\pi_0 = \frac{p_{12}+p_{13}}{p_{01}} \pi_1, \tag{23}$$

$$\pi_0 p_{02} + \pi_1 p_{12} + \pi_2 (1-p_{20}) = \pi_2,$$

$$\pi_0 p_{02} + \pi_1 p_{12} - \pi_2 p_{20} = 0, \tag{24}$$

$$\pi_0 p_{03} + \pi_1 p_{13} + \pi_3 (1-p_{30}) = \pi_3,$$

and

$$\pi_0 p_{03} + \pi_1 p_{13} - \pi_3 p_{30} = 0. \tag{25}$$

Adding Equations (23) and (24), we get

$$\pi_1 = \frac{p_{01}p_{20}}{p_{01}p_{12} + p_{12}p_{02} + p_{03}p_{13}}\pi_2. \quad (26)$$

From Equations (21), (23) and (26), we get

$$\pi_2\left[1+\frac{(p_{01}+p_{12}+p_{13})\,p_{30}}{p_{02}p_{12}+p_{02}p_{13}+p_{12}p_{01}}\right]+\pi_3=1. \quad (27)$$

From Equations 23 and 25, we get

$$\pi_3 = \frac{p_{12}p_{03} + p_{13}p_{03} + p_{01}p_{13}}{p_{03}p_{30}}\pi_1. \quad (28)$$

From Equations (26), (27) and (28), we get π_2. Subsequently, π_0, π_1, and π_3 can also be calculated. These are given as

$$\pi_0 = (p_{12}+p_{13})p_{20}D, \quad (29a)$$

$$\pi_1 = p_{01}p_{20}D, \quad (29b)$$

$$\pi_2 = AD, \quad (29c)$$

$$\pi_3 = \left(\frac{p_{20}p_{01}}{p_{03}p_{30}}\right)CD, \quad (29d)$$

where, A, B, C and D are

$$A = p_{01}p_{12} + p_{02}p_{12} + p_{02}p_{13},$$

$$B = p_{12}p_{20} + p_{13}p_{20} + p_{01}p_{20},$$

$$C = p_{12}p_{03} + p_{13}p_{03} + p_{01}p_{13},$$

and

$$D = \frac{1}{(A+B)+\left(\frac{p_{20}p_{01}}{p_{03}p_{30}}\right)C}.$$

Similarly, the state probabilities of packet transmission states can be obtained using

$$[v_0\, v_1\, v_2\, v_3\, v_4\, v_5]\begin{bmatrix} q_{00} & q_{01} & q_{02} & 0 & 0 & 1-q_{00}-q_{01}-q_{02} \\ q_{10} & q_{11} & 0 & 0 & 1-q_{10}-q_{11} & 0 \\ q_{20} & 0 & 0 & 1-q_{20} & 0 & 0 \\ q_{30} & 0 & 0 & 1-q_{30} & 0 & 0 \\ q_{40} & q_{41} & 0 & 0 & 1-q_{40}-q_{41} & 0 \\ q_{50} & q_{51} & q_{52} & 0 & 0 & 1-q_{50}-q_{51}-q_{52} \end{bmatrix}$$

$= [v_0 \, v_1 \, v_2 \, v_3 \, v_4 \, v_5]$. (30)

In addition to this, we have the following equation also for the steady state condition:

$v_0 + v_1 + v_2 + v_3 + v_4 + v_5 = 1$. (31)

Following the procedure similar to the one used for calculating offset steady state probabilities, the Equations (30) and (31) give packet transmission steady state probabilities as

$v_0 = L,$ (32a)

$v_1 = IL,$ (32b)

$v_2 = FL,$ (32c)

$v_3 = JL,$ (32d)

$v_4 = CKL,$ (32e)

$v_5 = HL,$ (32f)

where, E, F, G, H, I, J, K, and L are given by

$$E = p_{00} + p_{01} + p_{02},$$

$$F = p_{50} + p_{51} + p_{52},$$

$$G = \frac{1 - p_{10} - p_{11}}{p_{40} + p_{41}},$$

$$H = \frac{1 - E}{F},$$

$$I = p_{02} + Hp_{52},$$

$$J = \frac{I(1 - p_{20})}{P_{30}},$$

and

$$K = \frac{p_{01} + Hp_{51}}{1 - Cp_{41} - p_{11}}.$$

Figures 5.3 and 5.4 show the offset steady state probabilities as a function of transition probabilities. In Figure 5.3, transition probability p_{00} and p_{02} are varied, and other transition probabilities are kept constant. In this figure, π_0 represents the failed state. As transition probability p_{00} increases, the probability of being in the steady state π_0 increases. However, if probability p_{02} increases, the probability of being in the steady state π_2 increases. Referring to Figure 5.1, π_2 is the desired next offset state provided the present state of the system is the π_0 state. Thus, a higher value of transition p_{02} is desired. As RT-MAC does not go into any deadlock state, there is no sudden discontinuity in the offset steady state transition probabilities in Figures 5.3 and 5.4.

Figures 5.5 and 5.6 show the packet transmission steady state probabilities as a function of transition probabilities. Referring to Figure 5.2, the desired next steady state is v_0 and the present steady state is also v_0. It is because v_0 to v_0 transition amounts to 2 hops packet transmission. Therefore, in this case, a higher value of transition probability q_{00} and lower value of q_{05} is desired. Figures 5.5 and 5.6 also do not show sudden discontinuity in the packet transmission steady state transition probabilities, which signifies RT-MAC does not go into any deadlock state suddenly due to its functionality. In real scenarios, a system goes to failed state gradually generally with the decay of the battery of nodes in a WSN.

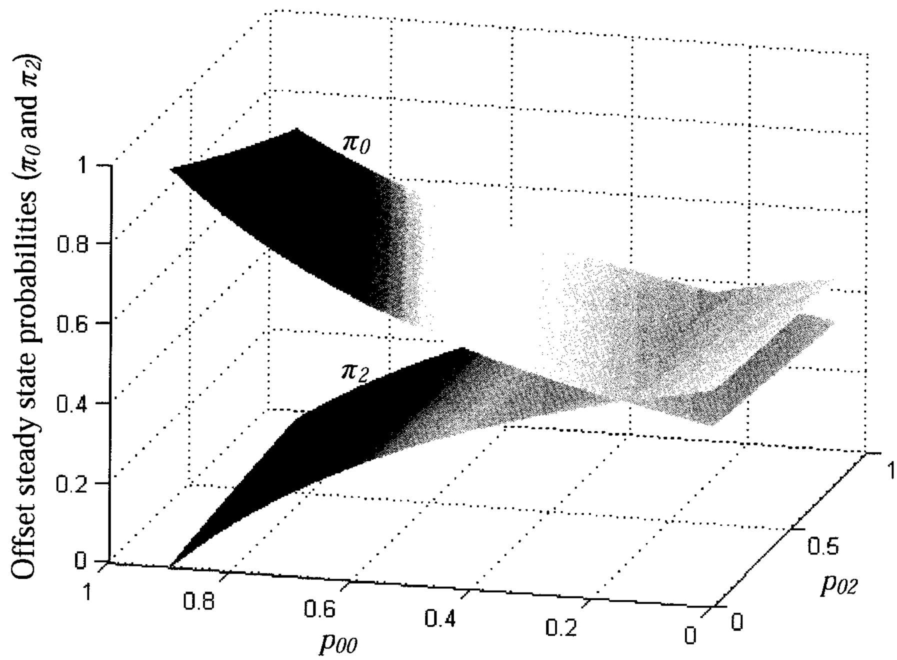

Figure 5.3: Offset steady state probabilities (π_0 and π_2) as a function of transition probabilities (p_{00} and p_{02})

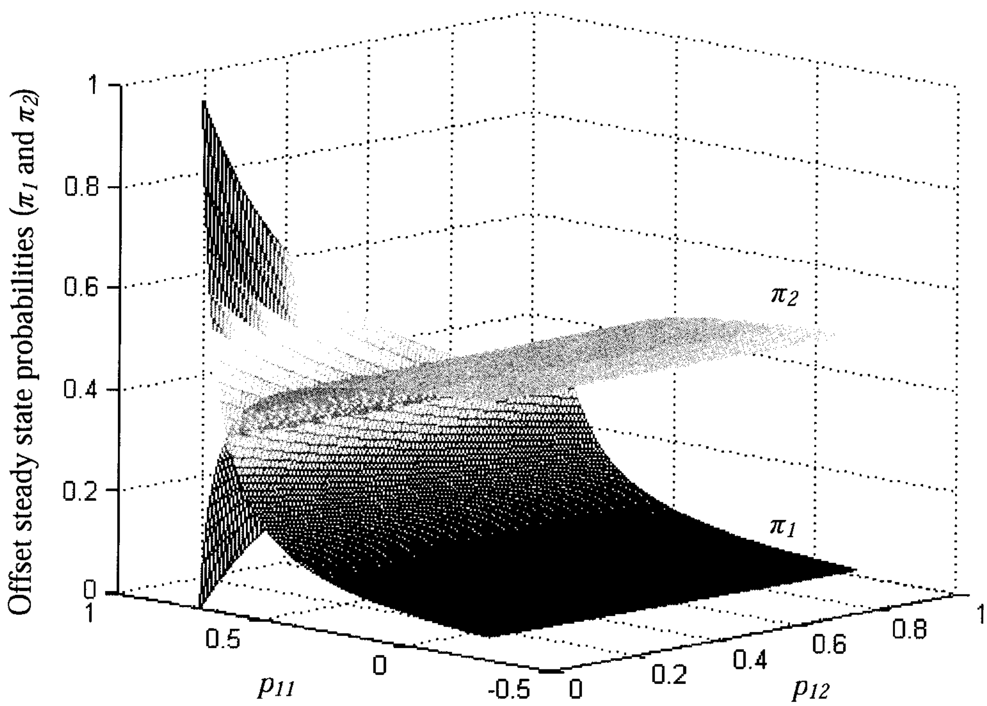

Figure 5.4: Offset steady state probabilities (π_1 and π_2) as a function of transition probabilities (p_{11} and p_{12})

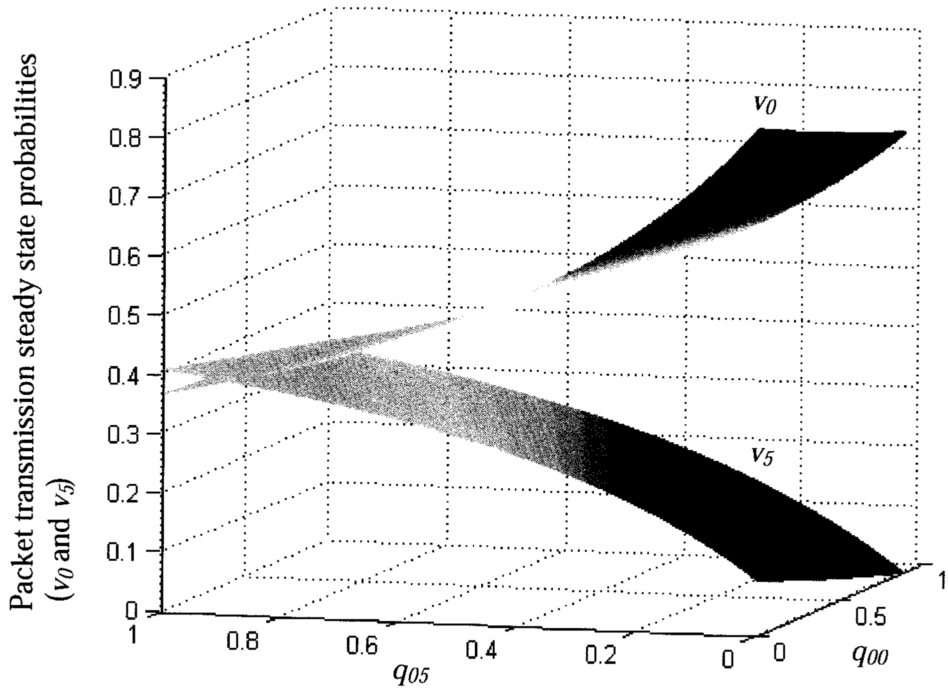

Figure 5.5: Packet transmission steady state probabilities (v_0 and v_5) as a function of transition probabilities (q_{00} and q_{05})

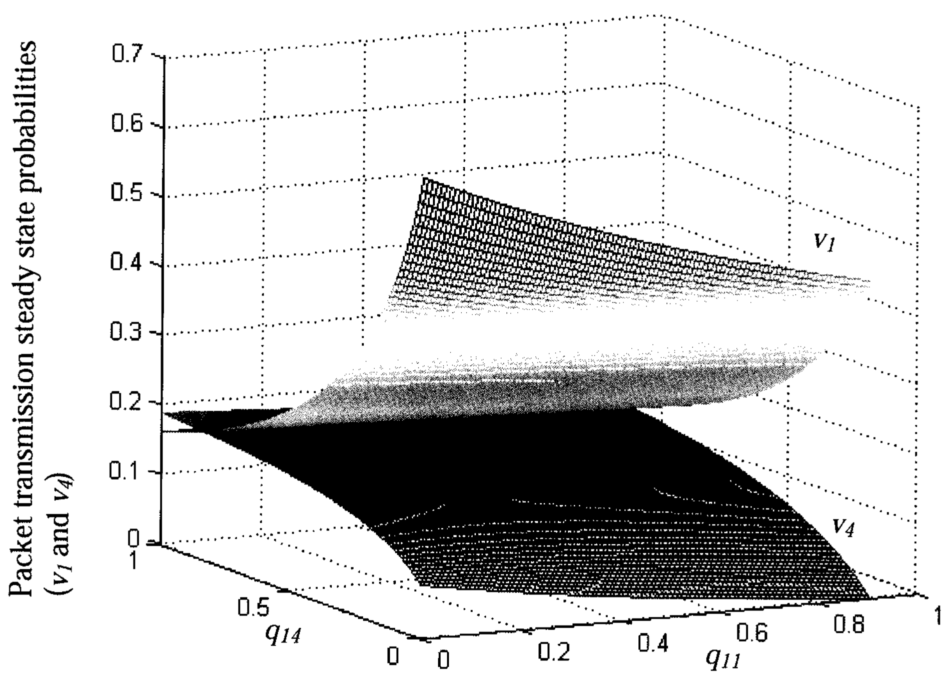

Figure 5.6: Packet transmission steady state probabilities (v_1 and v_4) as a function of transition probabilities (q_{11} and q_{14})

5.2 Mean end-to-end delay calculation

As mentioned in Section 4.5, end-to-end delay bound for the m^{th} packet has two parts. The first part is the offset delay α that occurs due to the previous m-1 packets, and the second part is the packet transmission delay that is the transmission time of the m^{th} packet. In the context of stochastic analysis, we refer these two parts of mean end-to-end delay $\overline{T_D(m,n)}$ as the mean offset delay $\overline{\alpha}$ and the mean packet transmission time $\overline{T_D(1,n)}$. Here, n is the total number of hops between the source and the destination.

In order to calculate mean offset delay, we need to find out the mean number of hops traversal in a frame during the offset state transition. It is denoted by $\overline{\eta_H}'$. It is calculated using offset steady state probabilities, offset state transition probabilities, and the number of hops traversal in a frame in the offset state transition. Here, matrix for number of hop traversal per frame (η_H') for the offset state transitions can be constructed using the fifth column of Table 5.1. It is given as follows for $\eta_H = 1$:

$$[\eta_H'] = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \qquad (33)$$

Thus, $\overline{\eta_H}'$ is calculated as

$$\overline{\eta_H}' = \sum_{i=0}^{3} \pi_i \left(\sum_{j=0}^{3} p_{ij} (\eta_H')_{ij} \right). \qquad (34)$$

Now, $\dfrac{4(m-1)}{\overline{\eta_H}'}$ number of frames needed for previous m-1 data packets since each previous packet adds an offset delay equivalent to 4 hop transmission time. Thus, the mean offset delay is given as

$$\overline{\alpha} = \left[\frac{4(m-1)}{\overline{\eta_H}'} \right] T_f, \qquad (35)$$

where, T_f represents a frame duration, which is a one time step in this DTMC analysis of RT-MAC.

Similarly, to calculate mean packet transmission delay, we need to find out the mean number of hops traversal in a frame during the packet transmission state transition. It is denoted by $\overline{\eta_H''}$. It is calculated using packet transmission steady state probabilities, packet transmission state transition probabilities, and the number of hops traversal in a frame in the packet transmission state. The matrix for number of hop traversal per frame (η_H'') for the packet transmission state transitions can be constructed using the fifth column of Table 5.2. It is given as

$$[\eta_H''] = \begin{bmatrix} 2 & 1 & 3 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 \\ 2 & 1 & 3 & 0 & 0 & 0 \end{bmatrix}. \tag{36}$$

Thus, $\overline{\eta_H''}$ is calculated as follows.

$$\overline{\eta_H''} = \sum_{i=0}^{3} v_i \left(\sum_{j=0}^{3} q_{ij} \left(\eta_H''\right)_{ij} \right) \tag{37}$$

Now, mean number of frames needed for the m^{th} data packets over n hop is $\frac{n}{\overline{\eta_H''}}$. Thus, the mean packet transmission delay is given as

$$\overline{T_D(1,n)} = \left\lceil \frac{n}{\overline{\eta_H''}} \right\rceil T_f, \tag{38}$$

where, T_f represents frame duration, which is a one time step in this DTMC analysis of RT-MAC.

Thus, mean end-to-end delay is given by

$$\overline{T_D(m,n)} = \overline{\alpha} + \overline{T_D(1,n)},$$

$$\overline{T_D(m,n)} = \left\lceil \frac{4(m-1)}{\overline{\eta_H'}} \right\rceil T_f + \left\lceil \frac{n}{\overline{\eta_H''}} \right\rceil T_f,$$

$$\overline{T_D(m,n)} = \left\lceil \frac{4(m-1)}{\overline{\eta_H'}} + \frac{n}{\overline{\eta_H''}} \right\rceil T_f, \tag{39}$$

where, $\overline{\eta_H'}$, $\overline{\eta_H''}$ are given by Equations (31) and (34).

5.3 Mean throughput calculation

Mean throughput $\bar{\gamma}$ is defined as mean number of packet that reaches the destination per second. It can be calculated using mean end-to-end delay as given by Equation 39. It is given by

$$\bar{\gamma} = \frac{m}{\overline{T_D(m,n)}}$$

$$\bar{\gamma} = \frac{m\overline{\eta_H'}\,\overline{\eta_H''}}{\left[4(m-1)\overline{\eta_H''} + n\overline{\eta_H'}\right]T_f} \quad (40)$$

Though, Equation (39) and (40) are derived the case of $\eta_H = 1$, the same equation remains valid for $\eta_H = 5, 9, 13\ldots$etc. due to 4 hops symmetry in transmission pattern of RT-MAC protocol. However, generalized case of η_H=1, 5, 9..., the $\overline{\eta_H'}$ and $\overline{\eta_H''}$ matrix will change as

$$[\eta_H'] = \begin{bmatrix} 0 & \eta_H & \eta_H+1 & \eta_H+2 \\ 0 & 0 & \eta_H & \eta_H+1 \\ \eta_H+1 & 0 & 0 & 0 \\ \eta_H & 0 & 0 & 0 \end{bmatrix}, \quad (41)$$

and

$$[\eta_H''] = \begin{bmatrix} \eta_H+1 & \eta_H & \eta_H+3 & 0 & 0 & 0 \\ \eta_H & \eta_H+1 & 0 & 0 & 0 & 0 \\ \eta_H & 0 & 0 & 0 & 0 & 0 \\ \eta_H & 0 & 0 & 0 & 0 & 0 \\ \eta_H & \eta_H+1 & 0 & 0 & 0 & 0 \\ \eta_H+1 & \eta_H & \eta_H+3 & 0 & 0 & 0 \end{bmatrix}.$$

(42)

As mentioned earlier in this subsection, a similar procedure can be followed to find mean end-to-end delay for η_H = 2, 6, 10, ..., η_H = 3, 7, 11, ... and η_H = 4, 8, 12, ...etc.

5.4 Example scenario

We take a scenario in which we need to calculate the mean-end-delay for 10^{th} packet over 30 hops with frame duration of 610 milliseconds. Here, ON duration T_U is taken such that it facilitates at least one hop traversal per frame. We also like to calculate the mean throughput of the WSN in a scenario where m = 10, n = 30, T_f = 610, and η_H = 1.

Now, using the fifth column of Tables 5.1 and 5.2, we can write a typical offset state transition matrix and packet state transition matrix as

$$[p]=\begin{bmatrix} \frac{1}{10} & \frac{1}{10} & \frac{7}{10} & \frac{1}{10} \\ 0 & \frac{1}{7} & \frac{3}{7} & \frac{3}{7} \\ \frac{2}{3} & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix},$$

and

$$[q]=\begin{bmatrix} \frac{7}{10} & \frac{1}{10} & \frac{1}{10} & 0 & 0 & \frac{1}{10} \\ \frac{2}{7} & \frac{4}{7} & 0 & 0 & \frac{1}{7} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\ \frac{2}{7} & \frac{4}{7} & 0 & 0 & \frac{1}{7} & 0 \\ \frac{7}{10} & \frac{1}{10} & \frac{1}{10} & 0 & 0 & \frac{1}{10} \end{bmatrix}.$$

Using Equations (29a) to (29d) and (32a) to (32d), we can get offset steady state probabilities and packet transmission steady state probabilities. These are given by

$[\pi]$ = [0.3934 0.0459 0.4426 0.1180], and

$[v]$ = [0.5806 0.1935 0.0645 0.0645 0.0323 0.0645].

Using Equations (34) and (37), mean number of hops traversal per frame for offset and packet transmission states are given as

$\overline{\eta_H}' = 1.4163$, and

$\overline{\eta_H}'' = 1.5483$.

Using Equation (39), the mean end-to-end delay for the 10^{th} packet is calculated as

$\overline{T_D(m,n)} = 27.3253$ Seconds.

Similarly using Equation 40, the mean throughput of the network is calculated as

$\bar{\gamma} = 0.3660$ packets per seconds.

5.5 Summary of the chapter

This chapter presents analysis of RT-MAC protocol under realistic scenarios where fault condition is also considered in the network in single-stream scenario. A Markov modeling of RT-MAC protocol is done in this chapter. Using this model, mean end-to-end delay and mean throughput can be estimated.

Chapters 5 and 6 present analysis of RT-MAC in single-stream scenario, whereas the next chapter presents RT-MAC operations in multi-stream scenarios.

Chapter 6: RT-MAC Protocol in Multi-Stream Scenario

In the case of a multi-stream communication, there are multiple source and sink node pairs simultaneously active in a network. There are two types of multi-stream communication possible in the network, namely non-interfering and interfering multi-stream communication. In the context of RT-MAC protocol, in non-interfering multi-stream communication, the streams are separated from each other by at least 3 hops. Thus, they don't affect each other in any way as shown in Figure 6.1. Hence, this scenario is considered as a collection of several independent single-stream communications in the network. In this case, all the single-stream analysis of RT-MAC done in Chapter 3 is valid. However, in interfering multi-stream communication case, there are multiple source, and sink node pairs simultaneously active in a network such that some of the nodes may be shared among various communication streams. This occurs when streams are separated from each other by 2 or less hops. It is discussed

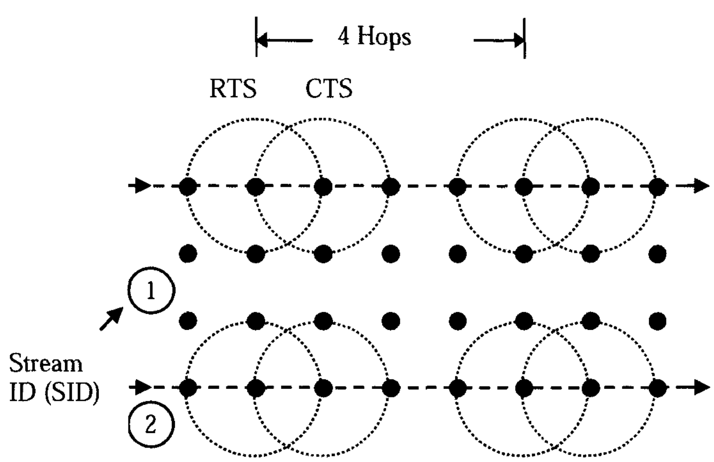

Figure 6.1: Communication pattern with 3 hop stream separation. Two parallel bi-directional streams are possible. It is a non-interfering multi-stream scenario.

further in Section 6.2.

This chapter presents RT-MAC protocol in interfering multi-stream scenario. In Section 6.1, we discuss the issues involved in this RT-MAC protocol design in multi-stream scenario. Section 6.2 illustrates operation of RT-MAC protocol in multi-stream scenario. In Section 6.3, we present an example scenario to show the effect of varying the duty cycle on ensuring delay bounds.

6.1 Problem analysis

With regard to timing considerations, there are two key challenges involved in the interfering multi-stream communication at MAC layer in WSNs.

a) It is difficult to ensure end-to-end delay deadline for any data packet stream when some of the nodes are shared among various streams. There are two possible solutions to this problem. The first approach is to provide service differentiation at MAC layer [33], and [34]. With service differentiation, a prioritized packet transmission is possible. Thus, in multi-stream scenario, it can provide a privileged access to a stream with high priority data. However, RT-MAC is designed to be a general purpose real-time MAC protocol, thus, currently it does not support prioritized packet transmission. Rather, it tries to provide a fair access of the medium to all the involved streams. The second approach is to vary duty cycle during run time of the shared sensor nodes. Multi-stream version of RT-MAC is based on this approach. The limitation in this approach is that ON duration of any sensor node cannot be increased more than the frame duration. Hence, there is a limit to the number of streams that can be accommodated by varying duty cycle. Hence, if nodes are operating at a low duty cycle, then more number of streams can be accommodated and vice versa. Apparently, if nodes are continuously ON, then there is no scope of adjusting ON duration; thus in this case, it's not possible to guaranty delay deadlines of two or more interfering streams that came into contact during run time.

b) The problem of false blocking of nodes becomes far more dangerous in multi-stream scenario. It is because the localized problem of *false blocking of nodes* in single-stream communication becomes the globalized problem of *false blocking of streams* in multi-stream scenario. Figure 6.2 shows the false blocking problem in multi-stream scenario.

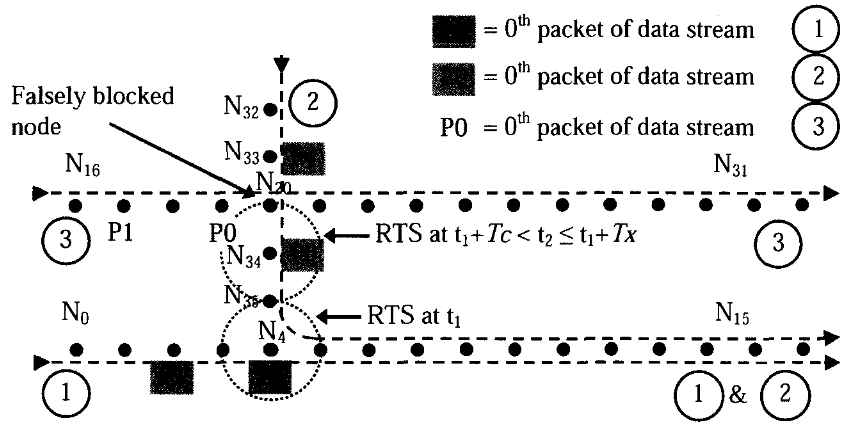

Figure 6.2: False blocking in multi-stream scenario in RTS/CTS handshake based wireless MAC protocols

In this figure, there are three data packet streams. Stream 1 and 3 are separated by three hops. Thus, in the absence of Stream 2, Stream 1 and 3 are able to travel independently, without any interaction between them. However, the situation changes when Stream 2 is considered. For example, N_4 initiates transmission of P0 of stream 1 at t_1, which causes N_{35} to go into sleep mode. Now, if N_{34} initiates transmission of P0 of Stream 2 at t_2 with $t_1+Tc < t_2 \leq t_1+Tx$, then N_{20} will be falsely blocked after overhearing RTS because transmission of P0 of Stream 2 cannot be completed while N_{35} is sleeping. Consequently, P0 of data Stream 3 cannot move forward to N_{20} at t_3 with $t_2+Tc < t_3 \leq t_2+Tx$. However, if N_{20} is not falsely blocked, then data Stream 1 and 3 can travel simultaneously without any collision as N_4 and N_{20} are separated by three hops. The next section presents a solution to such false blocking scenarios.

6.2 Protocol description

The RT-MAC, in general, avoids false blocking problem and collisions due to feedback based MAC approach as discussed in Chapter 3 for the single-stream scenario. However, this feedback based approach is modified here as per various situations (discussed later in this section in connection with Figures 6.4 to 6.7) in interfering multi-stream communica-

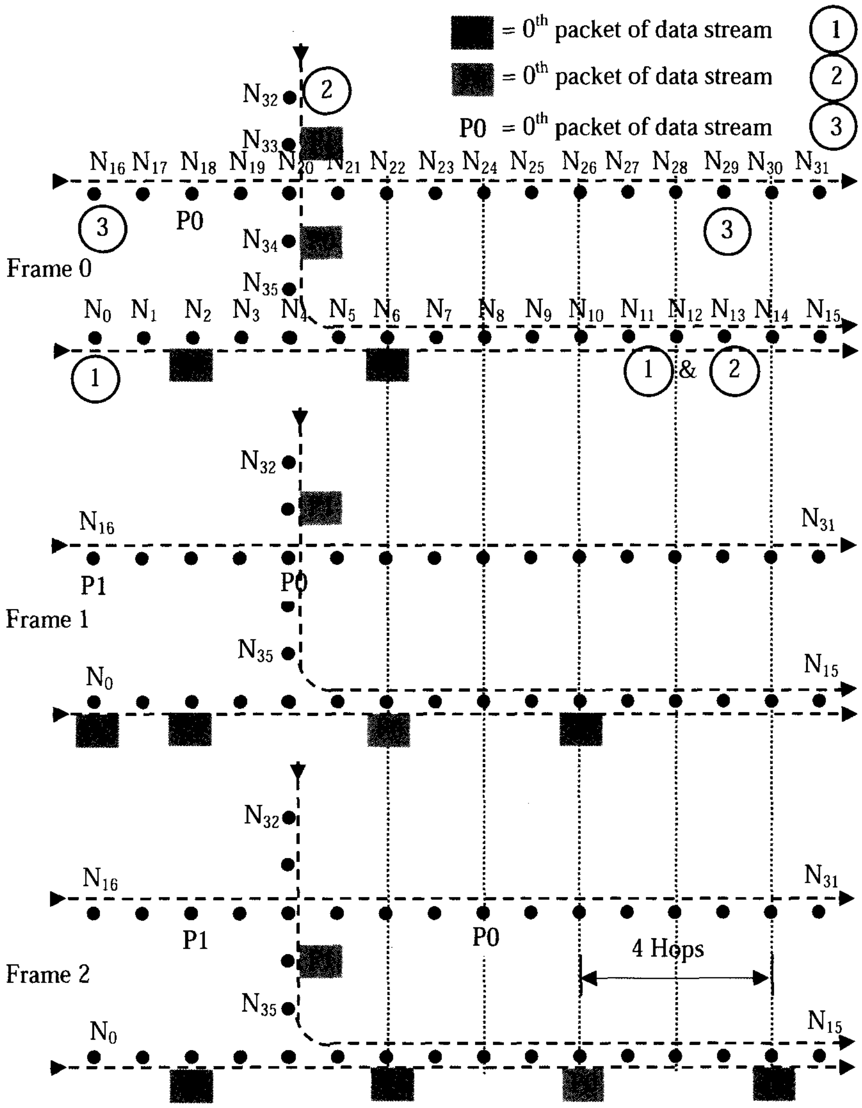

Figure 6.3: An example of data packet transmission pattern in multi-stream scenario with feedback approach in periodic sleep/listen mode of sensor nodes (CC transmission is not shown).

tion. In general, the major modification made in the feedback approach is the introduction of the concept of a junction node, which has the responsibility of forwarding CC control packets alternately among various streams. This concept is explained with the help of Figure 6.3, which provides a solution of false blocking scenario of Figure 6.2. Figure 6.3 shows three data packet transmission patterns at the end of three frames (i.e., three listen/sleep schedules at MAC layer). This figure is drawn for the case when sensor nodes have ON duration such that it facilitates 4 hops data packet transmission per frame (η_H=

4). In this figure, N_4 and N_{20} act as the junction nodes. A junction node allows the flow of data packets among involved data streams alternately by diverting CC transmission to the appropriate stream, i.e., by giving clearance to one stream at a time. For example, at the end of Frame 0, P0 of stream 1 is at N_6. Hence, N_5 tries to send CC to N_2 in the beginning of Frame 1. However, N_4 (junction node) grabs this CC, and forwards it towards N_{34} to give clearance to P0 of Stream 2. Similarly, in the middle of Frame 1, P0 of Stream 2 reaches to N_4. Then, N_{35} tries to send CC to N_{33} to give clearance to the next packet of Stream 2. However, again N_{20} grabs this CC and forwards it to N_{18} to give clearance to P0 of stream 3. In the beginning of the second frame, as P0 for Stream 2 had reached to N_6 node, N_5 tries to sends CC to N_{34}; however, N_4 grabs this CC and forwards it to stream 1 to give clearance to P1 of Stream 1. Similarly, in the middle of the second frame, P0 of Stream 3 reaches to N_{22}; hence, N_{21} tries to send CC to N_{18} to give clearance to the next packet of Stream 3. However, this CC is again captured by N_{20} and forwarded to N_{33} to give clearance to P1 of Stream 2. Thus, at the end of the second frame, P1 of Stream 1 is at the same node (i.e., N_6), where P0 of Stream 1 was there at the beginning of frame 0. This mechanism allows each stream to get a fair access of the medium, and thus, it prevents false blocking of streams effectively. A similar methodology is applicable for avoiding false blocking problem in overlapping streams, and for streams with one and two hop separation.

RT-MAC is based on S-MAC protocol. S-MAC is a fix duty cycle protocol. However, the RT-MAC can vary duty cycle during run time. Thus, it can maintain delay guarantees for the case when multiple streams share some of the sensor nodes. This could be accomplished by changing duty cycle of these shared nodes during run time as soon as any stream joins or leaves the shared sensor nodes. RT-MAC can change duty cycle during run-time in both single-stream and multi-stream communication. However, there are many differences in the objective and methodology of changing duty cycles in both single and multi-stream version of RT-MAC. For example, in the single-stream version of RT-MAC, the change in duty cycle is initiated by the sink node using CC control packet. Thus, it takes at least one end-to-end data transfer time to change duty cycle of all nodes in a stream. In addition to this, it would be a system wide duty cycle change; thus, all the nodes in a stream will have same duty cycle. The goal of increasing duty cycle in

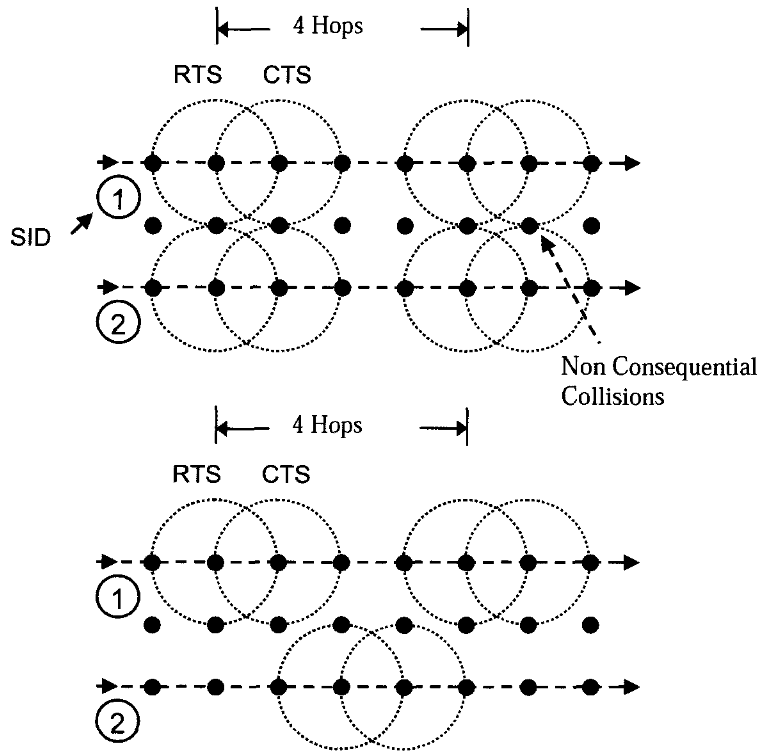

Figure 6.4: Communication pattern with Stream Separation = 2 Hops

the single-stream mode is to reduce the end-to-end delay. However, in the multi-stream version of RT-MAC, the change in duty cycle is initiated by a junction node. It is done using RTS control packets. Additionally, the goal of varying duty cycle in multi-stream scenario is to meet delay guarantees set at the beginning of the transmissions.

Figures 6.4 to 6.7 show the multi-stream communication and their impact on providing delay guarantees with overlapping streams, and for streams with one and two hop separation. For example, Figure 6.4 shows multi-stream communication with two hop separation between streams. In this case, Streams 1 and 2 will cause collision at one hop neighbors. However, as these collisions are happening at the nodes that are not involved in the communication, hence, it will not affect the delay deadlines of the streams. This case can be considered as non-interfering multi-stream communication; thus, there is no need to change duty cycle of nodes to guaranty delay deadlines. Figure 6.5 shows the communication pattern of bi-directional streams with one hop separation. In this case, the feedback mechanism of RT- MAC ensures 4 hops separation between two consecutive

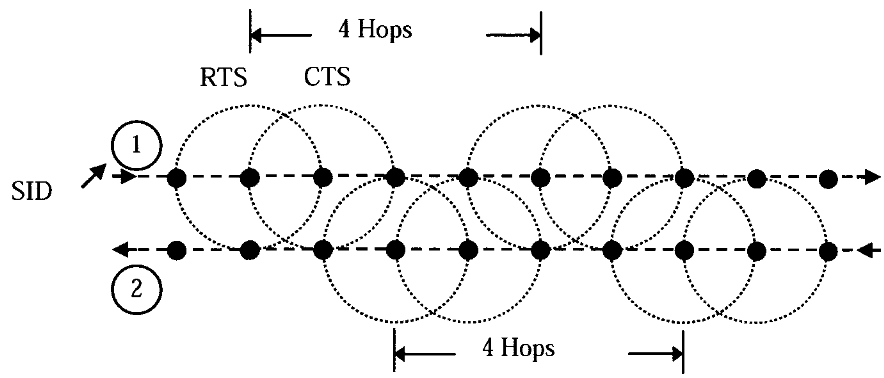

Figure 6.5: Communication pattern with bi-directional streams and one hop stream separation

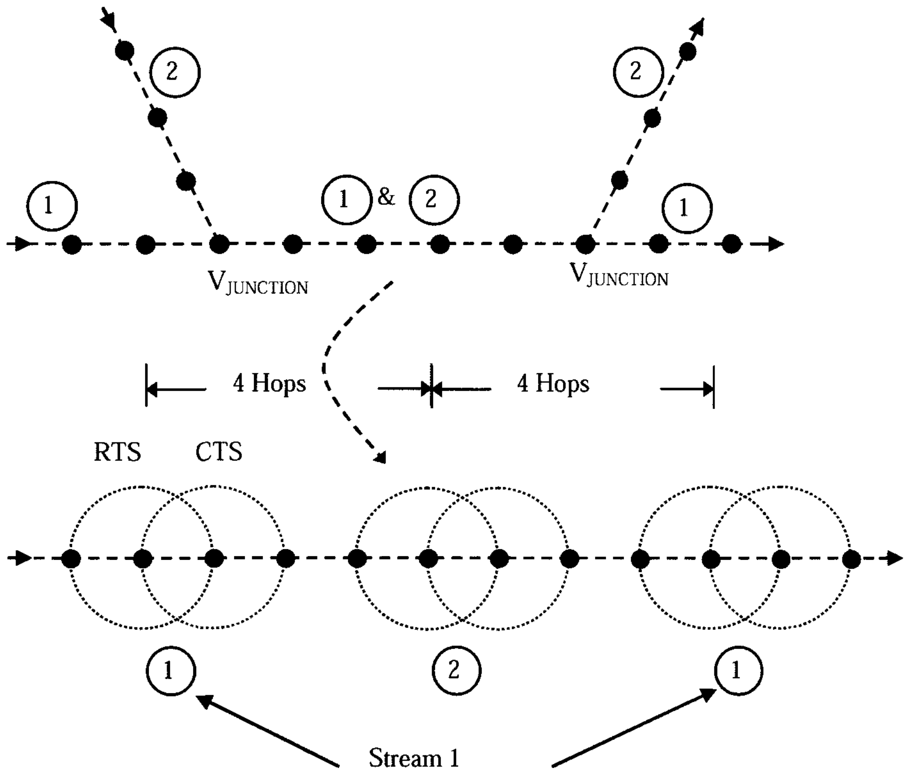

Figure 6.6: Communication pattern of unidirectional streams and no stream separation

data packet transmissions of a stream, and it also maintains two hop separation between the nearest data transmissions of two streams. It appears that there are no junction nodes in this scenario, thus who will facilitate the forwarding of CC to appropriate stream is a question. The answer to this question is that at the beginning, when any node of a stream

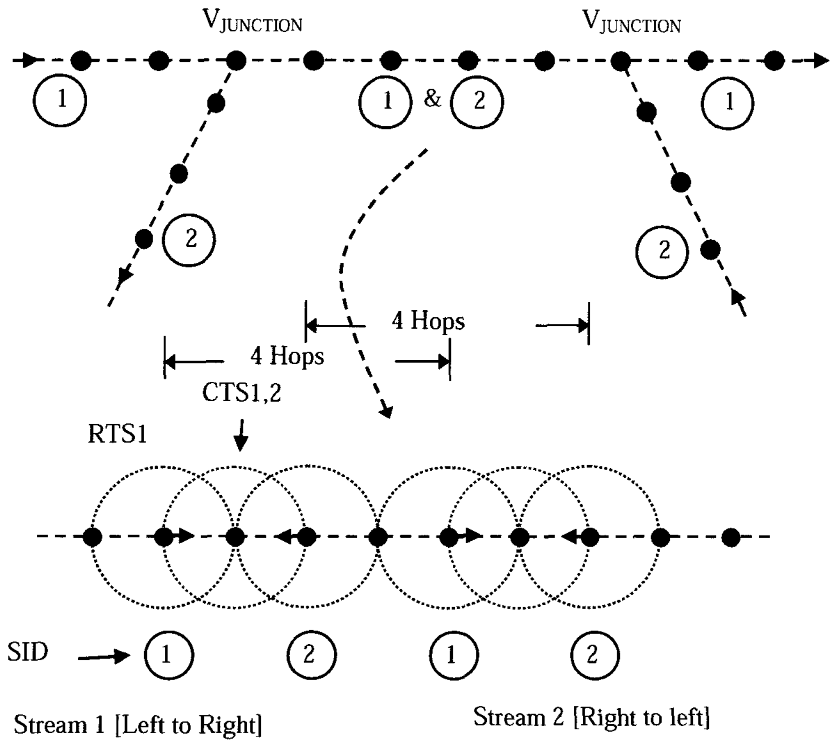

Figure 6.7: Communication pattern with bi-directional streams and no Stream Separation

becomes aware of some communication in the neighborhood, then it assumes the responsibility of the junction node. Here, one hop neighbors of an existing stream are always aware of communication status of this stream; hence, when they receive RTS from any newly approaching stream for the first time, they responds with NACK to inform newly approaching stream to wait for CC. This NACK is also overheard by its one hop neighbor of the existing stream, which will serve as an indication to it that a new stream is waiting two hops away to get the medium access in the same area. Thus, this particular node of the existing stream assumes the responsibility of a junction node.

Figure 6.6 shows the communication pattern of unidirectional overlapping streams. From feedback approach point of view, this case is the same as single-stream case. Thus, four hop separation is maintained between the two nearest data packet transmissions of two streams. However, as the medium is shared among two streams in this case, hence the duty cycle needs to be increased at the shared nodes to ensure delay deadlines of the involved streams. Figure 6.7 shows the communication pattern of bi-directional

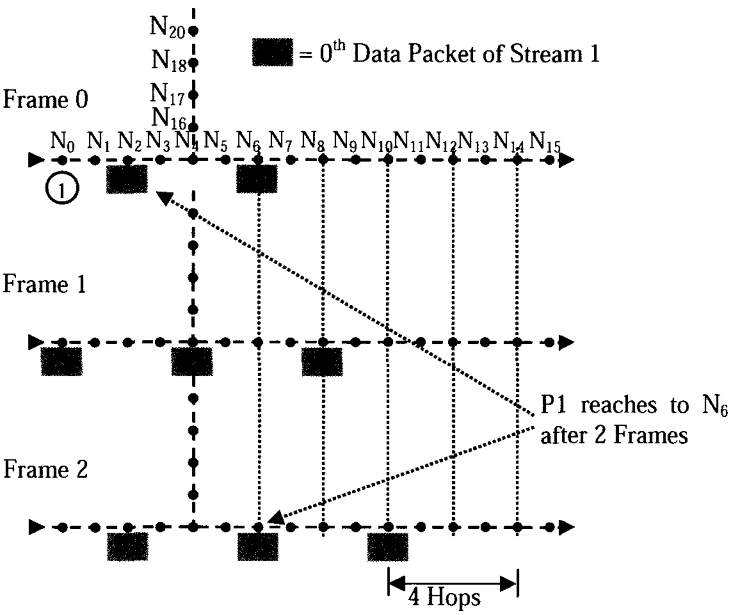

Figure 6.8: Example scenario of communication pattern with single-stream for η_H=2 hops per frame

overlapping streams. In this case, two hop separation is maintained between the nearest two data packet transmissions of two streams, whereas there is a 4 hop separation between two consecutive data packet of the same streams. In this scenario, a packet of a stream goes forward by two hops and a CC is released for the previous hop neighbors of the same streams; however, this CC is captured by its two hop neighboring node, who has a data packet from the other stream to be sent in the opposite direction. The same process is repeated when this node forwards the data by two hops in the opposite direction, and a CC is released. The duty cycle of the shared nodes needs to be increased in this scenario to maintain delay deadlines of the involved streams.

6.3 Example scenario of varying duty cycle to ensure end-to-end delay guarantees

In this section, consider an example scenario to show that the delay deadlines can be insured by increasing the duty cycle of the shared nodes. Figure 6.8 shows a scenario of a communication pattern with single-stream for the ON duration that ensures η_H=2 hops

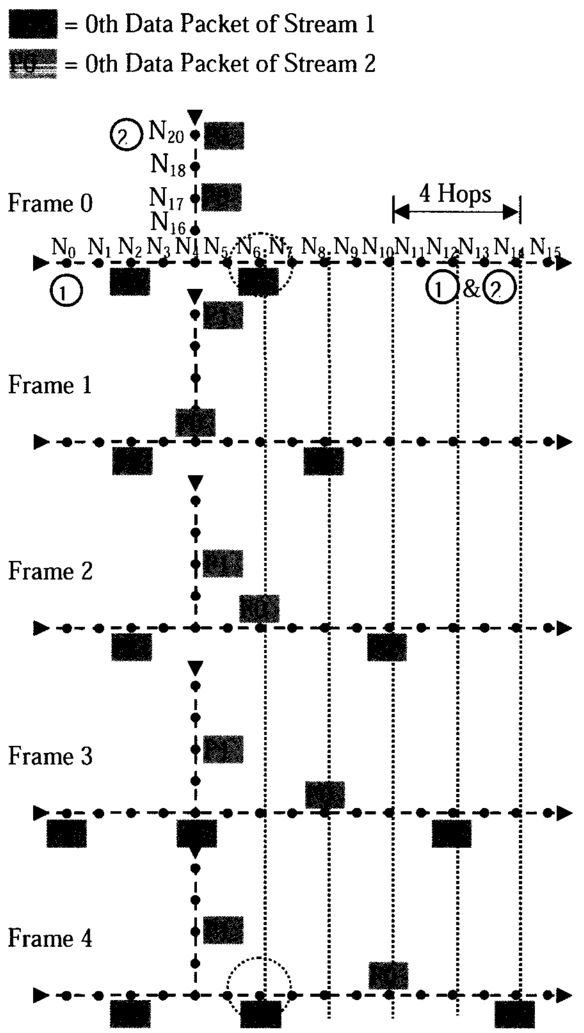

Figure 6.9: Example scenario communication pattern with two approaching streams for η_H=2 hops per frame (i.e., without change in duty cycle)

per frame. In this figure, P0 of Stream 1 is at N_6 at the end of frame 0. After 2 frames, P1 of Stream 1 reaches to the same node (N_6). In Figure 6.9, Stream 2 is introduced in the scenario at junction node N_4, but the duty cycle is kept the same. Following the feedback based MAC approach as mentioned in connection with Figure 6.3, it is found that P1 of Stream 1 reaches to N_6 after four frames. Now, similar to Figure 6.3, there are two streams in Figure 6.10; however, in this case, duty cycle of the shared node is doubled. Thus, it is found that P1 of Stream 1 can reach to N_6 in two frames, which is the same duration that was found in the single-steam case of Figure 6.8.

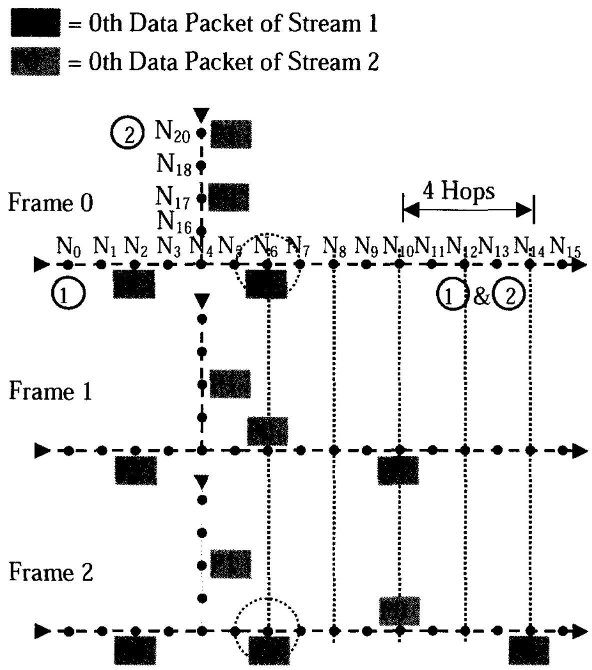

Figure 6.10: Example scenario of communication pattern with two approaching streams with i.e., $\eta H=4$ hops per frame (i.e., with increased (doubled) duty cycle)

Similar to the example mentioned above, various other multi-stream test scenarios have been generated using OMNeT++ network simulator. Subsection 7.2.2 of the next chapter presents extensive simulation results and related discussion.

6.4 Summary of the chapter

This chapter presents RT-MAC protocol in multi-stream scenario. It is found that the parallel streams separated by two or more hops travels independently. Hence, this case can be treated as single-stream scenarios. However, if parallel streams are overlapping or separated by one hop, then duty cycle of shared nodes needs to be increased during run time to meet delay deadlines of individual streams.

The next chapter presents simulation results of RT-MAC protocol for both single-stream as well multi-stream scenarios.

Chapter 7: Simulation results and discussions

7.1 Simulation environment

A simulation study of RT-MAC is carried out using OMNeT++ simulator [63], [64], and [78]. Propagation delay is neglected as the distance involved in the sensor communication is very small, typically a few meters. It is assumed that data packet size do not change during run time in the network. Thus, data aggregation during run time in the WSN is not considered. Major simulation parameters are the size of data and control packets, contention durations, duty cycle, frame duration, packet arrival interval, transmission range, and node placement (grid). Values of the simulation parameters, as given in Table 7.1, are based on [20], [21], [62] and [72]. These parameters are largely based on MICA [75] and EYES [76] wireless sensor nodes. In Table 7.1, one Tic of a crystal oscillator of a sensor node is taken as 30.518 μsec. For a given number of hops, we took the average of 10 simulation runs, where each run was taken with different source and sink node pairs.

Table 7.1: Parameter Values in Simulation

Parameter	Value
Power consumption in receive mode	14.4 mW
Power consumption in transmit mode	36 mW
Power consumption in sleep mode	15 μW
RTS contention duration (in RT-MAC)	5 Tics
Radio bandwidth	40 Kbps

Maximum value of RTS contention duration without backoff (in S-MAC)	300 Tics
Typical value of RTS contention duration without backoff (in S-MAC)	180 Tics
Maximum RTS contention time (in T-MAC)	300 Tics
CTS, DATA, ACK, CCACK, CCQ, and CCQR contention duration (in RT-MAC)	5 Tics
CC contention duration (in RT-MAC)	15 Tics
CTS, DATA, ACK contention duration (in S-MAC, T-MAC and VTS)	5 Tics
Frame duration in RT-MAC, S-MAC and T-MAC	20000 Tics
Typical value of RTS,CTS,ACK,NACK, CC,CCACK, CCQ, and CCQR duration in RT-MAC	47 Tics
Typical value of RTS duration without backoff (including contention duration) in S-MAC	225 Tics
Typical value of RTS duration (including contention duration) in T-MAC	225 Tics
Typical value of RTS in VTS	47 Tics
Typical value of CTS, and ACK duration in S-MAC, T-MAC and VTS	47 Tics
Maximum SYNC duration	300 Tics
Data packet size	20 Byte
VTS super frame length	20 Nodes
VTS slot duration	20000 Tics
Session duration for RT-MAC, S-MAC and T-MAC	25 Seconds
Session duration for VTS	Till the end of data transfer

7.2 Results and Discussions

This section presents simulation results in single and multi-stream scenarios. The minimum hop routing protocol is used for all protocols in the simulation study in single-stream scenario, whereas a user defined routing path is used to generate various test cases for the performance evaluation in multi-stream scenario.

7.2.1 Single-stream scenario simulation results

Figures 7.1 and 7.2 show packet transfer delay for the first data packet as a function of number of nodes for low and high duty cycle respectively. It is evident from these figures that the packet transfer delay of the first data packet is the lowest in RT-MAC at both low and high duty cycles. This is due to the fact that the first data packet travels independently towards the sink as it does not need to wait for a CC control signal. Additionally, the reduced carrier sense duration also contributes to low delays in RT-

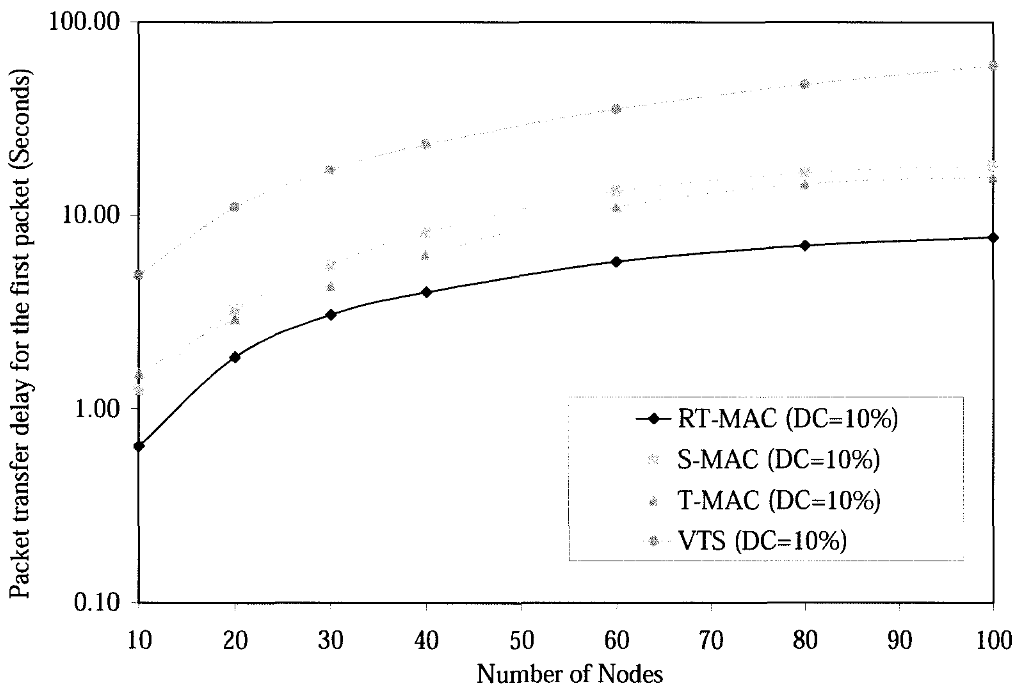

Figure 7.1: Packet transfer delay pattern for the first packet at 10 percent duty cycle

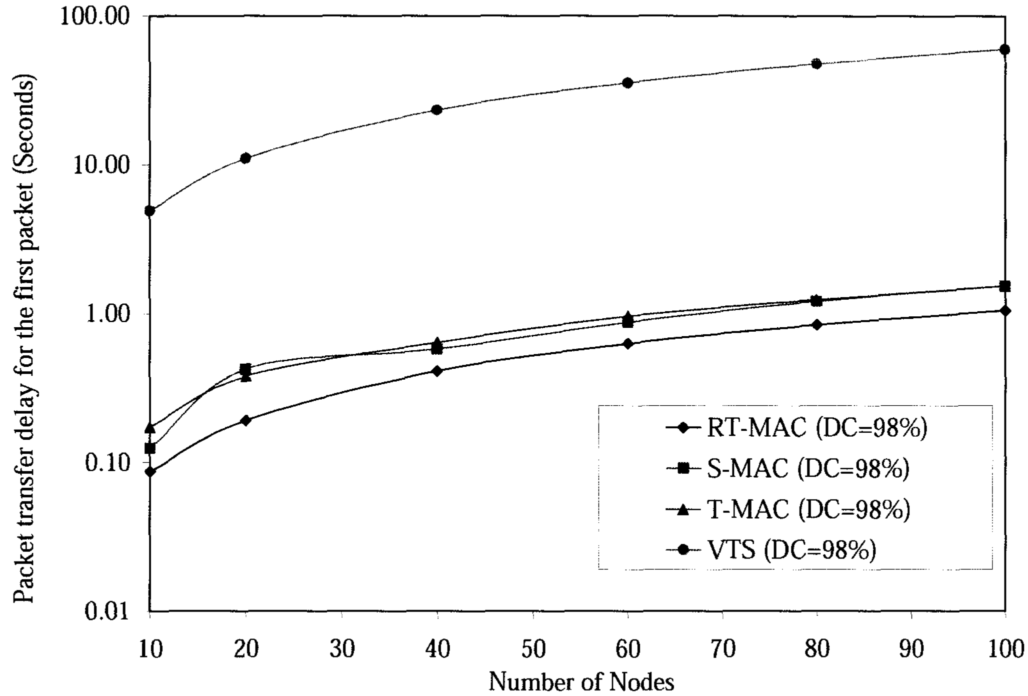

Figure 7.2: Packet transfer delay pattern for the first packet at 98 percent duty cycle

MAC. VTS takes the maximum time to deliver the first data packet to the sink at lower duty cycles because the data packet travels only one hop for a given TDMA slot; thus, the remaining ON duration is not used.

Simulation session duration is 25 seconds for RT-MAC, S-MAC and T-MAC, whereas in the case of VTS, it is taken until the end of all data packets reception at the sink because VTS could not complete the data transfer up to 100 nodes in a 25 second session limit. As shown in Figure 7.3, at a low duty cycle, all 25 data packets can be transferred up to 40 nodes in S-MAC and up to 60 nodes in RT-MAC within a 25 second session limit. However, T-MAC is able to transfer all data packets up to 100 nodes even at lower duty cycles because T-MAC has an adaptive duty cycle mechanism that prevents nodes from going into sleep mode when there are data packets in the neighborhood for transmission. However, T-MAC itself is not a good candidate for real-time communication due to its inconsistent data transmission pattern as all data packets keep on traveling together like a bunch toward the sink. Figure 7.4 shows that at higher duty cycles, RT-MAC provides the lowest end-to-end delay due to the combined effect of maximum spatial channel

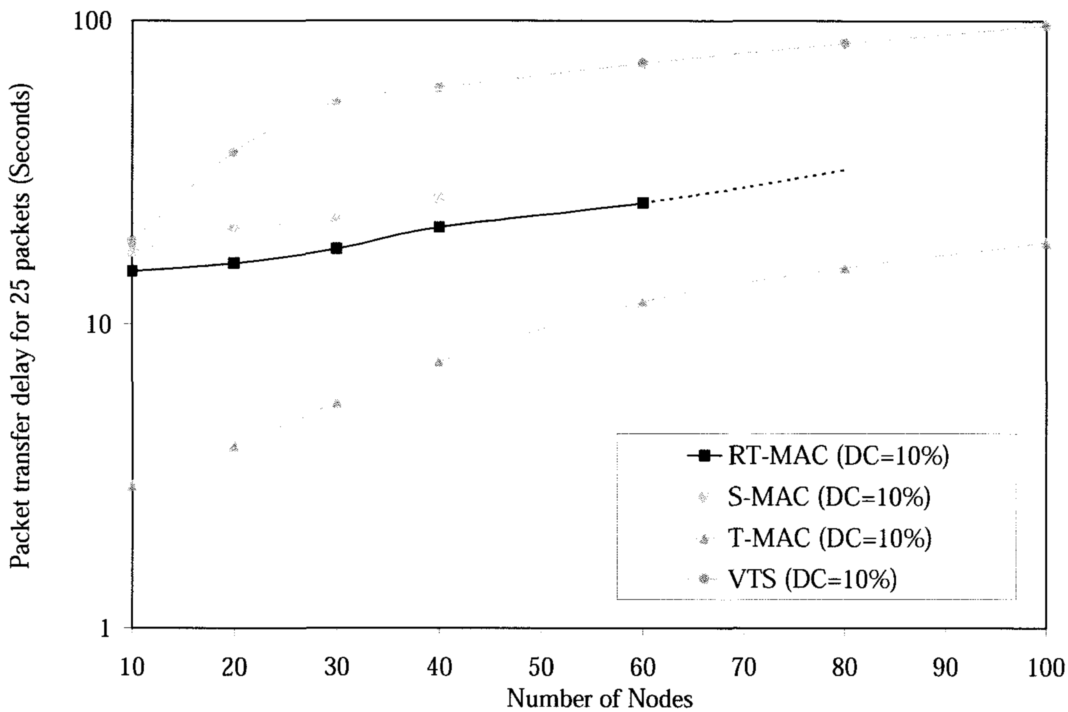

Figure 7.3: Packet transfer delay pattern for 25 packets at 10 percent duty cycle

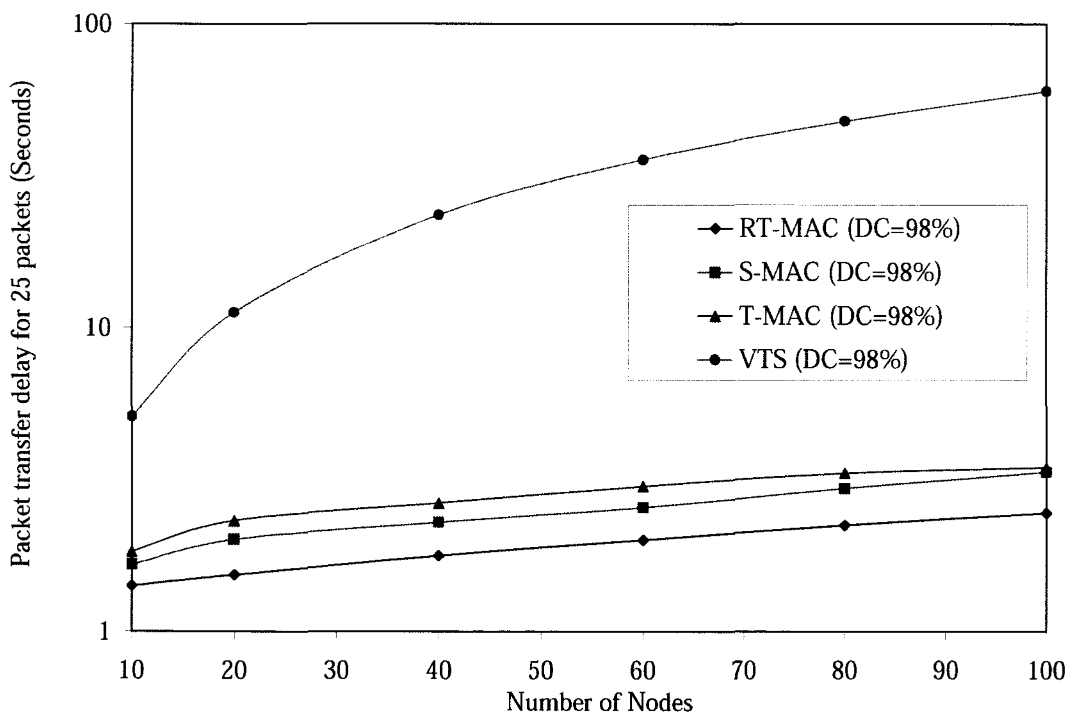

Figure 7.4: Packet transfer delay pattern for 25 packets at 98 percent duty cycle

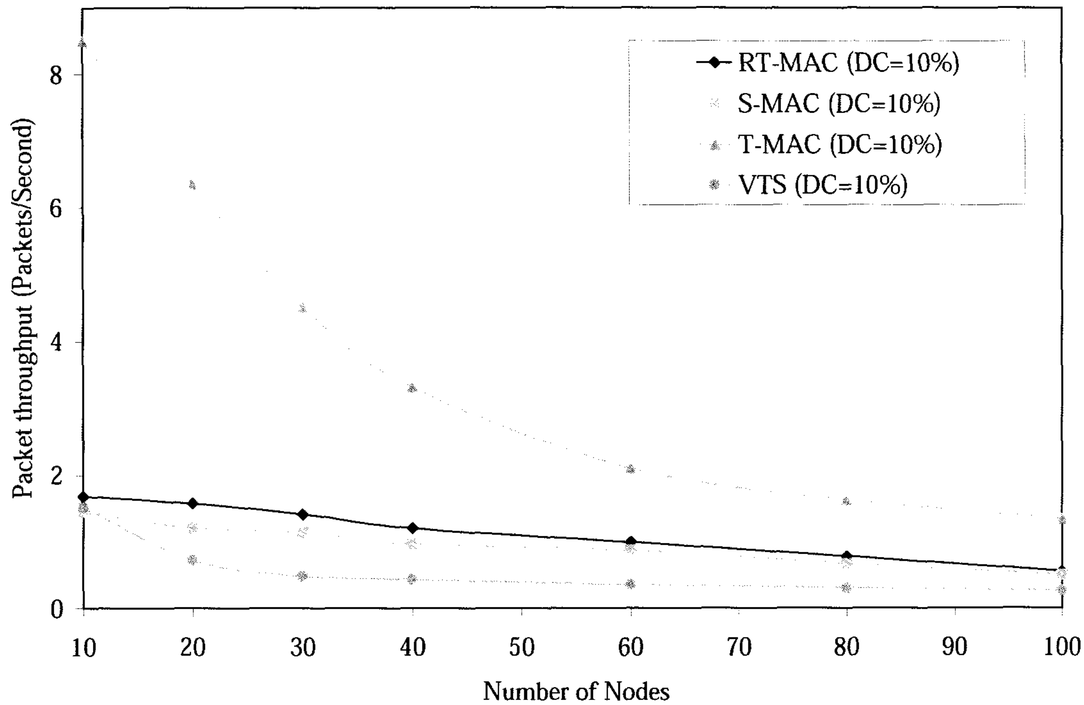

Figure 7.5: Packet throughput pattern at 10 percent duty cycle

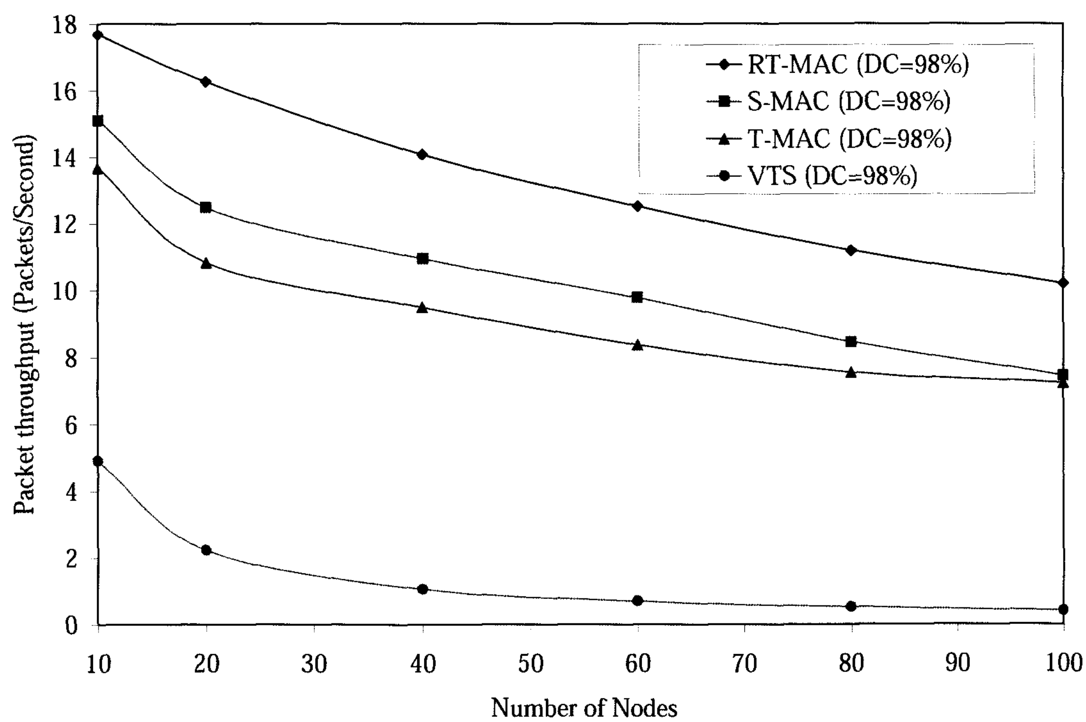

Figure 7.6: Packet throughput pattern at 98 percent duty cycle

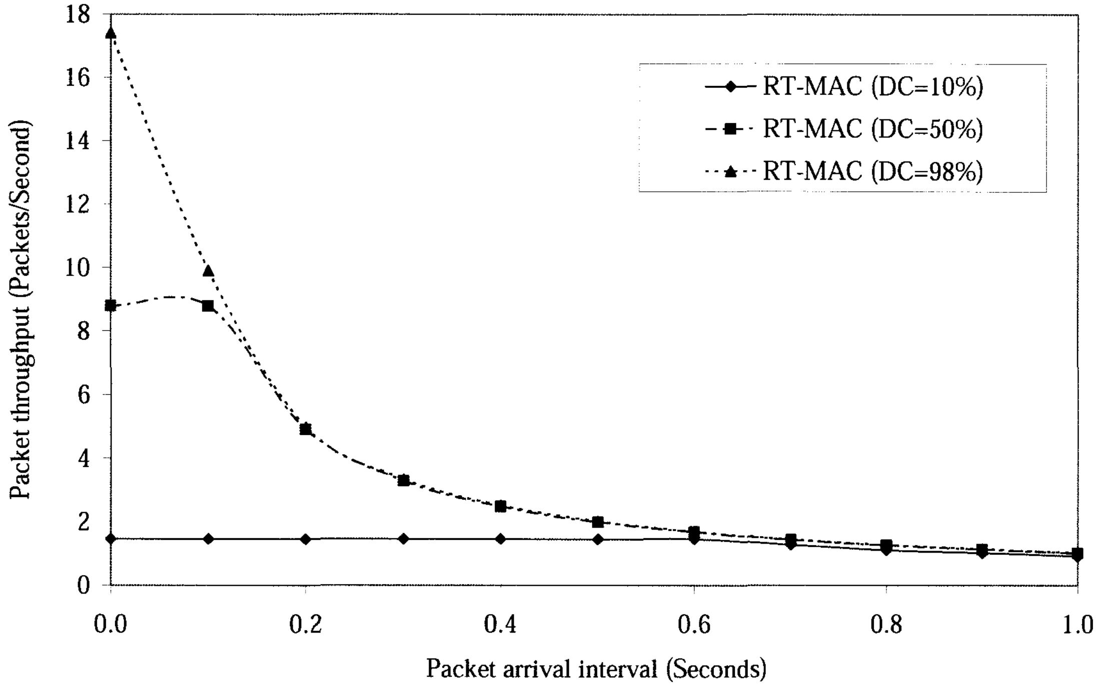

Figure 7.7: Packet throughput pattern in RT-MAC for 30 nodes

reutilization and reduced carrier sense duration. In general, delay performance of RT-MAC decreases with decreasing duty cycle because the packet transmission pattern is disrupted frequently.

When the 25 second simulation duration limit was removed, we were able to find the exact duration when all 25 data packets reached the destination even at lower duty cycles. Thus, the packet throughput graphs of Figures 7.5 and 7.6, and packet overhead graphs of Figures 7.9 and 7.10 show the results for the case when the simulation is allowed to run until the end of all data packet transfers.

As shown in Figure 7.6, packet throughput is the highest and most consistent (decreases linearly as number of nodes increases) in the case of the RT-MAC protocol at higher duty cycles. As mentioned in the case of Figure 7.4, the reason for better throughput performance at higher duty cycles is that a larger ON duration maintains the same transmission pattern. Thus, the delay and throughput performance of S-MAC and RT-MAC are more or less the same at lower duty cycles, whereas the advantage of RT-MAC over S-MAC begins to show as the number of nodes or duty cycle increases.

Figure 7.7 shows packet throughput as a function of data packet arrival interval. At a low

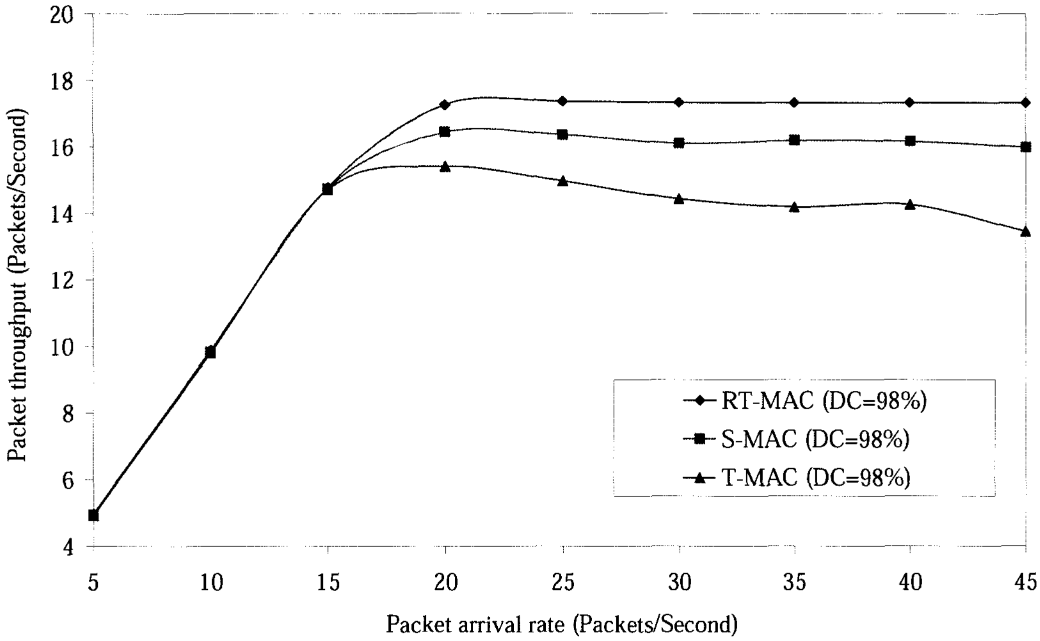

Figure 7.8: Packet throughphput pattern for 30 nodes at 98 percent duty cycle

packet arrival interval, packet throughput in the RT-MAC reaches the maximum possible value for a given duty cycle. This confirms the stability of RT-MAC protocol. It is because of two reasons. First, RT-MAC avoids collisions between closely arriving consecutive data packets by ensuring at least 4 hop separation between them. Second, RT-MAC prevents the nodes from being falsely blocked due to feedback based MAC strategy. Hence, it prevents dropping of packet throughput at the maximum load condition, i.e., when data packets arrive simultaneously. Figure 7.8 shows the packet throughput as a function of packet arrival rate in RT-MAC, S-MAC and T-MAC protocol for 30 node case. This figure further shows that RT-MAC provides a stable throughput performance at higher loads as compared to S-MAC and T-MAC protocol.

Packet overhead is calculated on the basis of average number of control packets needed for one data packet transfer by one hop. Figures 7.9 and 7.10 show that the packet overhead is the highest in RT-MAC protocol. This is due to extra control packets such as CC, CCACK, CCQ and CCQR, which are used in RT-MAC. However, this increased packet overhead does not increase delay or energy consumption of RT-MAC significantly compared to other protocols. The feedback based MAC strategy of

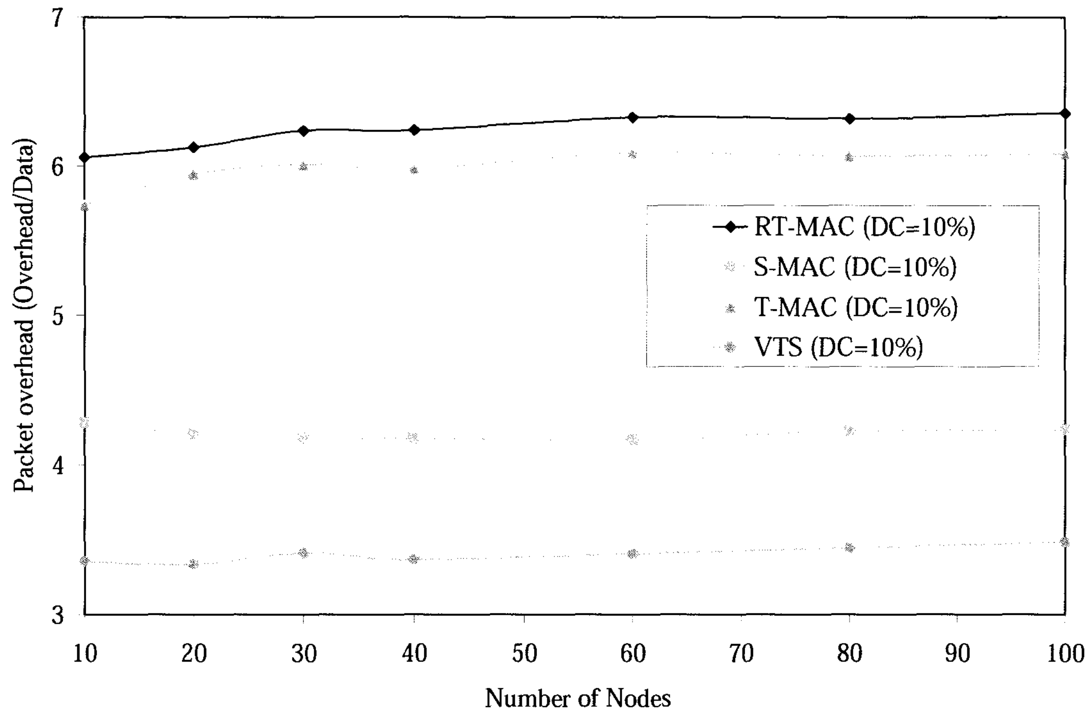

Figure 7.9: Packet overhead pattern at 10% duty cycle

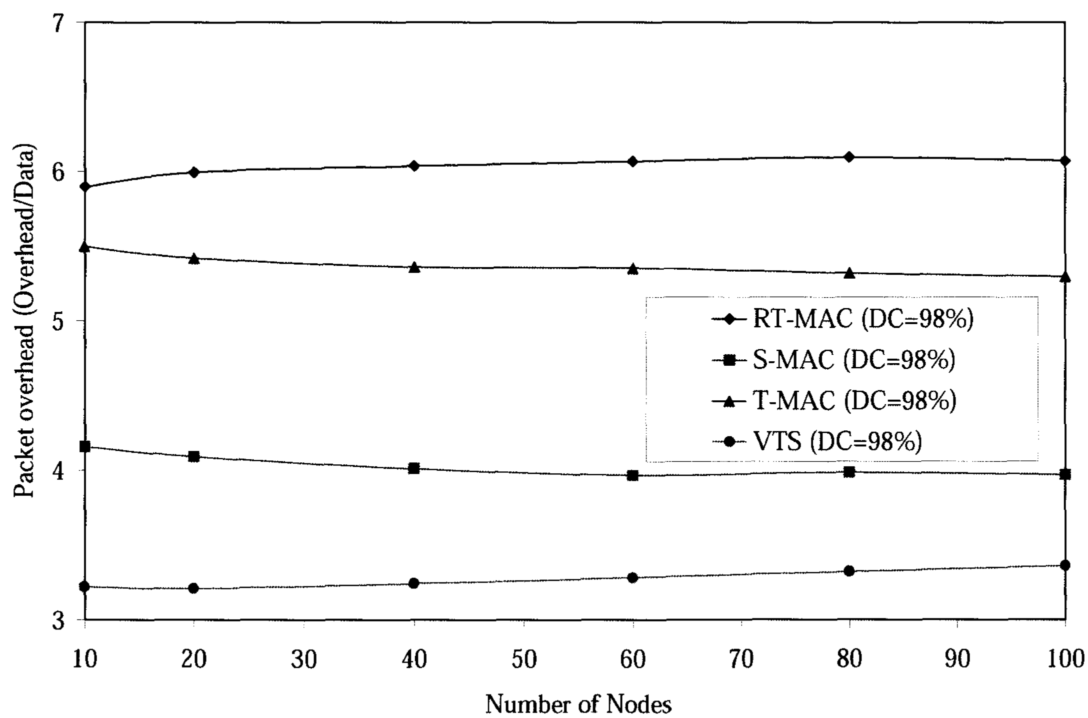

Figure 7.10: Packet overhead pattern at 98 percent duty cycle

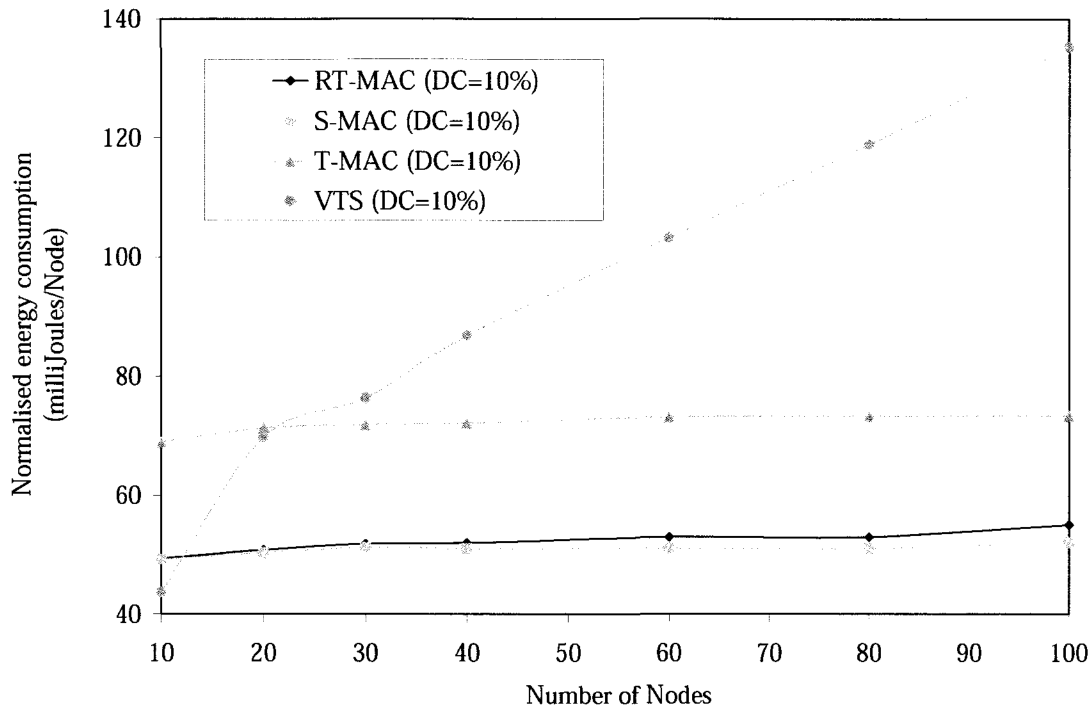

Figure 7.11: Normalized energy consumption pattern at 10 percent duty cycle

RT-MAC helps to reduce carrier sense duration and collisions significantly. Therefore, gain in terms of time saved due to lower contention duration and fewer collisions is more than the time taken by the extra control packets. Additionally, these figures show that the packet overhead is the lowest in VTS since it is a virtual TDMA based collision free protocol. However, as VTS still uses S-MAC (contention based) as the underlying protocol for the data packet transfer, it has a packet overhead close to 3 control packets per data packet due to RTS, CTS and ACK control packets being used for a data packet transmission by one hop.

Energy consumption includes the energy spent by all nodes in the network during transmission, reception, idle listening and sleep mode. Parameter values taken for energy consumption calculation are mentioned in Table 7.1. We observed energy consumption behavior of S-MAC, T-MAC and RT-MAC for a session duration of 25 seconds with 25 data packets transmission per session. However, in the case of VTS, we observed its energy consumption behavior until the end of all data packet transmissions. As shown in Figures 7.11 and 7.12, the energy consumption behavior of RT-MAC and S-MAC are

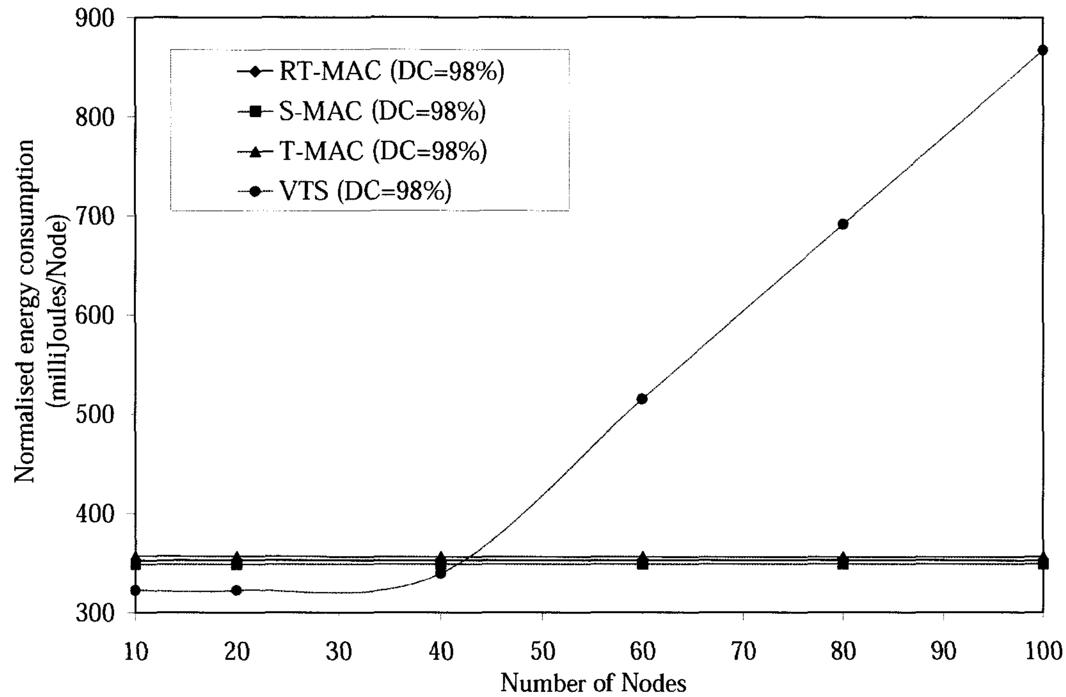

Figure 7.12: Normalized energy consumption pattern at 98 percent duty cycle

more or less the same at both low and high duty cycle operations. Thus, RT-MAC is able to provide delay guarantees without any significant increase in energy consumption.

7.2.2 Multi-stream scenario simulation results

In multi-stream simulation, nodes are placed in an x-y grid. A user defined routing path is used to provide desired stream separation among parallel streams. In Figure 7.13, the packet transfer delay pattern is shown for zero (i.e., overlapping streams), one and two hops stream separation at 10 percent duty cycle. 2S(→,→), and 2S(→,←) signify that two streams are travelling in same and opposite directions, respectively in a x-y grid. 2HS signifies that there is a 2 hop separation between the parallel streams. In this figure, 25 packets are sent in each stream, and the packet transfer delay is recorded for each stream. The packet transfer delay is averaged across the streams involved. As discussed in Section 6.2, the parallel streams with two hops separation travel independently, thus, they have the lowest average packet transfer delay. In case of two overlapping streams or

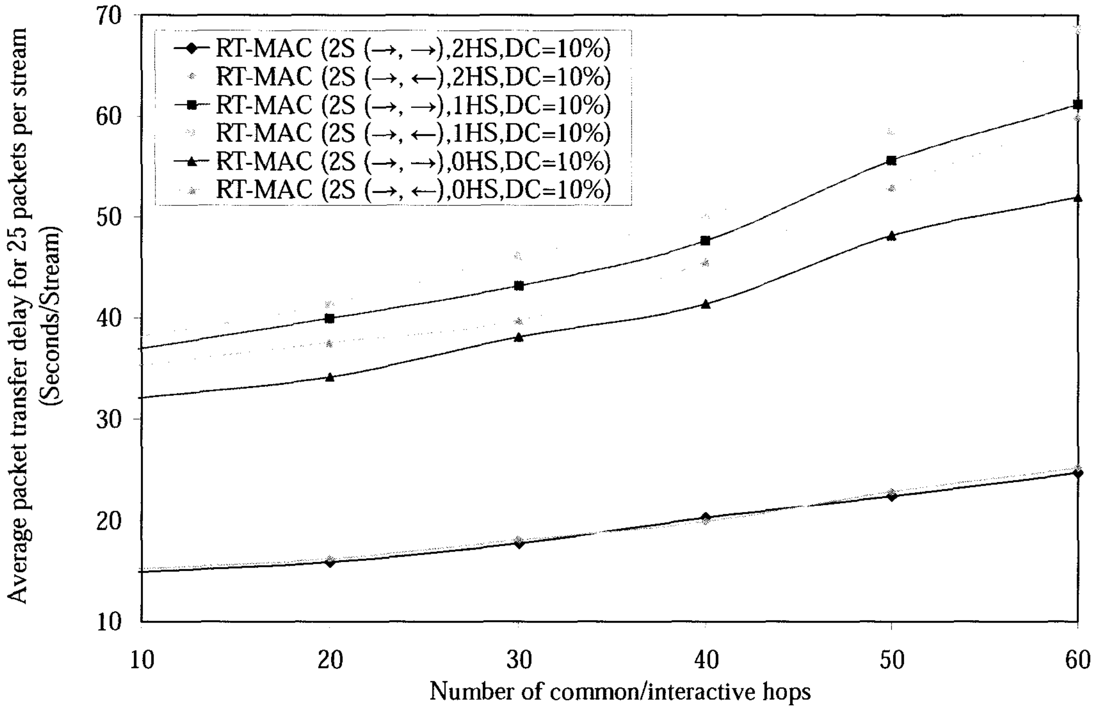

Figure 7.13: Average packet transfer delay pattern with two streams at 10 percent duty cycle

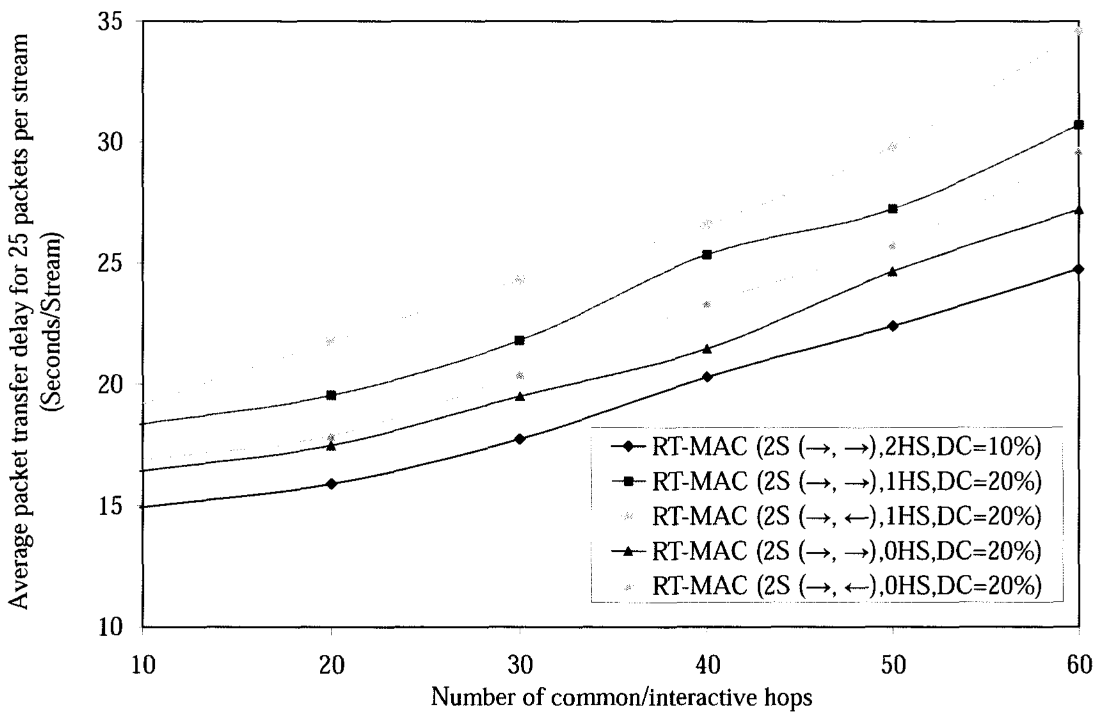

Figure 7.14: Average packet transfer delay pattern with two streams with increased duty cycle for interactive streams

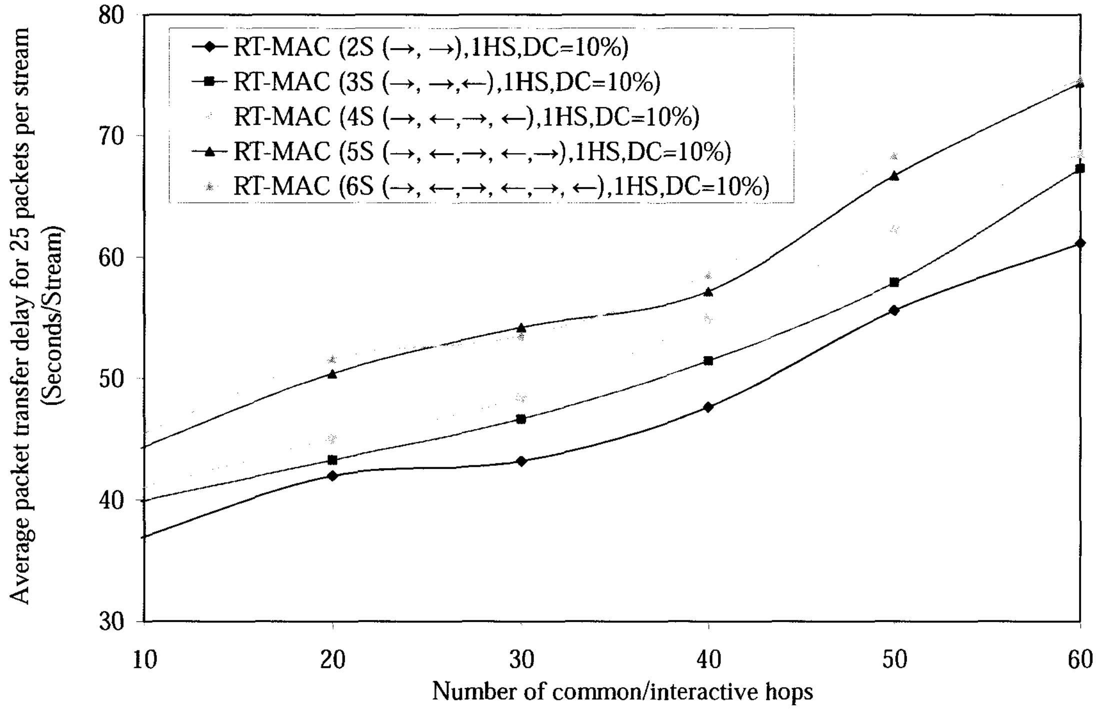

Figure 7.15: Average packet transfer delay pattern with multiple streams separated by one hop with 10 percent duty cycle

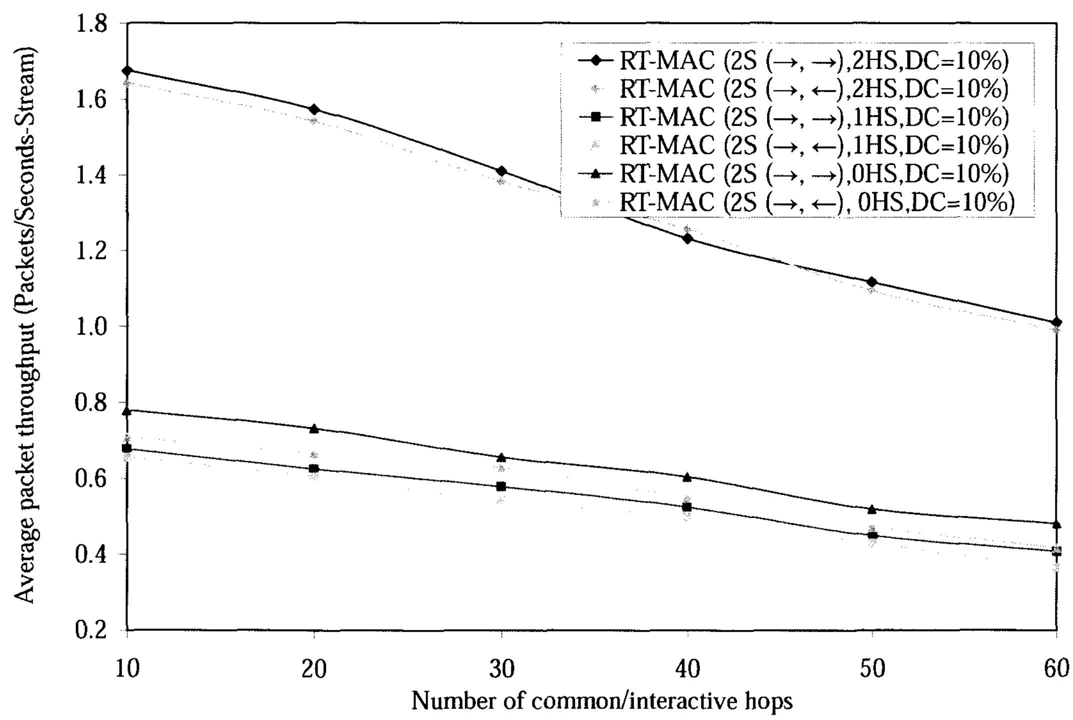

Figure 7.16: Average packet throughput pattern of RT-MAC at 10% duty cycle

two streams with one hop separation, the average packet transfer delay is almost doubled as compared to independent streams. It is because the data packet in one stream has to wait to ensure 4 hops separation from the nearest data packet in the other stream for the streams travelling in same direction, whereas, it has to ensure two hop separation in case of the streams travelling in opposite direction. This figure also shows that the delay is relatively higher for streams travelling in opposite direction. It is because the streams travelling in opposite direction need to maintain two hop separation communication pattern, thus, the frequency of usage of CC control packet used to maintain two hop separation is more than that of used to maintain 4 hops separation. It is also observed in this figure that streams with one hop separation have larger delay than that of the overlapping streams. It is because, in case of the streams with one hop separation, an additional CC transmission is involved in every 2 or 4 hops segment of the streams to inform the other stream about the successful transmission of a data packet in a stream by 2 or 4 hops.

In Figure 7.14, the duty cycle of the nodes in the network is doubled. Thus, it is observed that the average packet transfer delay is almost halved in overlapping streams and the streams with one hop separation; thus, it becomes comparable to the average packet transfer delay of the independent streams. This figure signifies that the delay guarantees of the individual streams can still be met when they share some of the nodes on the way by increasing the duty cycle of the shared node during run time.

Figure 7.15 shows the average packet transfer delay with 2, 3, 4, 5 and 6 parallel stream scenarios with one hop separation among streams at 10 percent duty cycle. This figure shows that the average packet transfer delay do not vary much with the increase in number of streams involved. It is because in RT-MAC, any stream is affected only by two parallel streams on either side of it, whereas the streams on both sides of a stream travel independently as they are separated from each other by two hops.

Figures 7.16 and 7.17 show the average packet throughput for 10 and 20 percent duty cycle respectively. As shown in Figure 7.16, the packet throughput decrease by almost half for the overlapping streams and the streams with one hop separation. It is because the streams are sharing nodes, thus, only one stream can travel at a time in a 2 or 4 hop segment. Similarly, the average packet throughput is almost doubled when duty cycle is

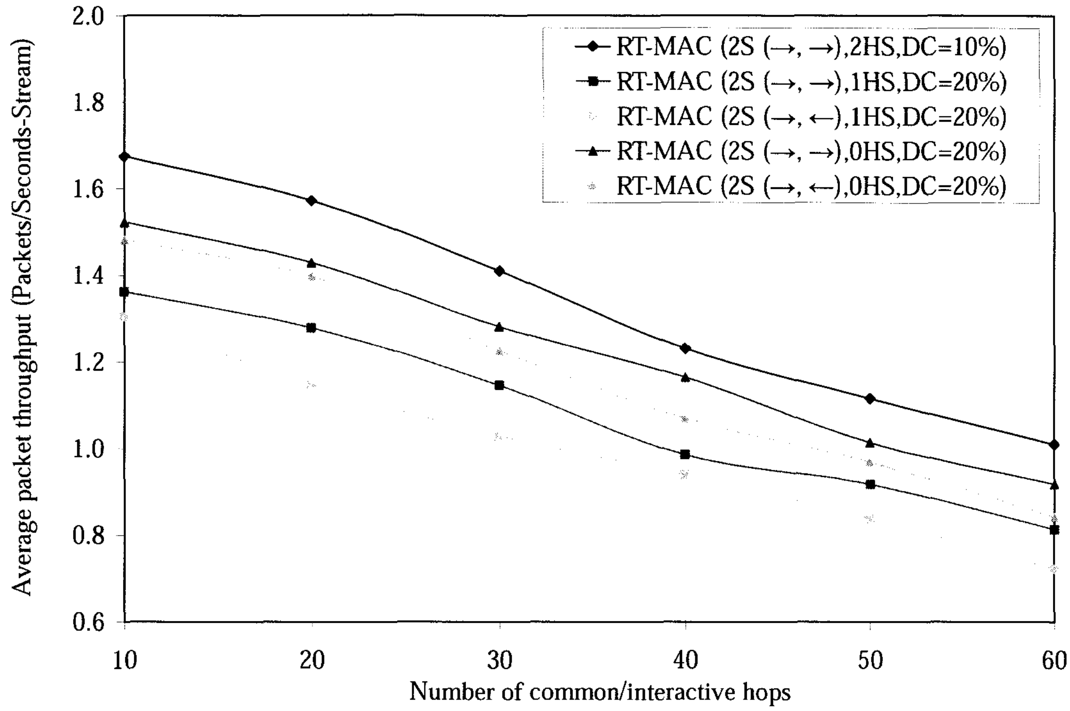

Figure 7.17: Average packet throughput pattern of RT-MAC with increased duty cycle for interactive streams

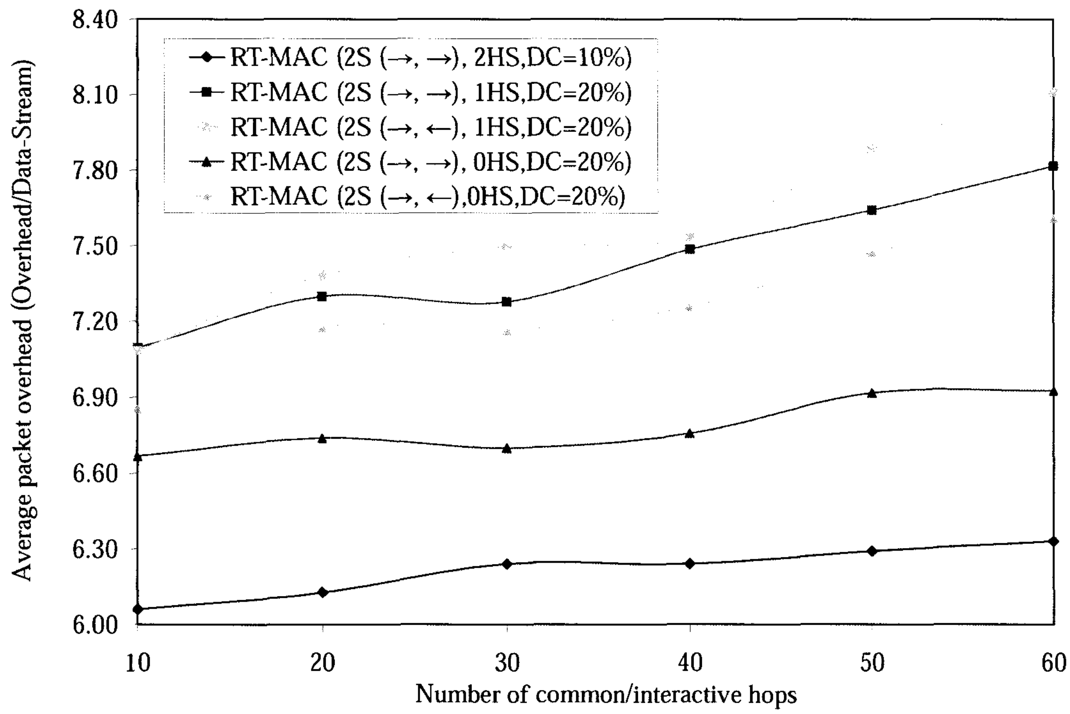

Figure 7.18: Average packet overhead pattern of RT-MAC with increased duty cycle for interactive streams

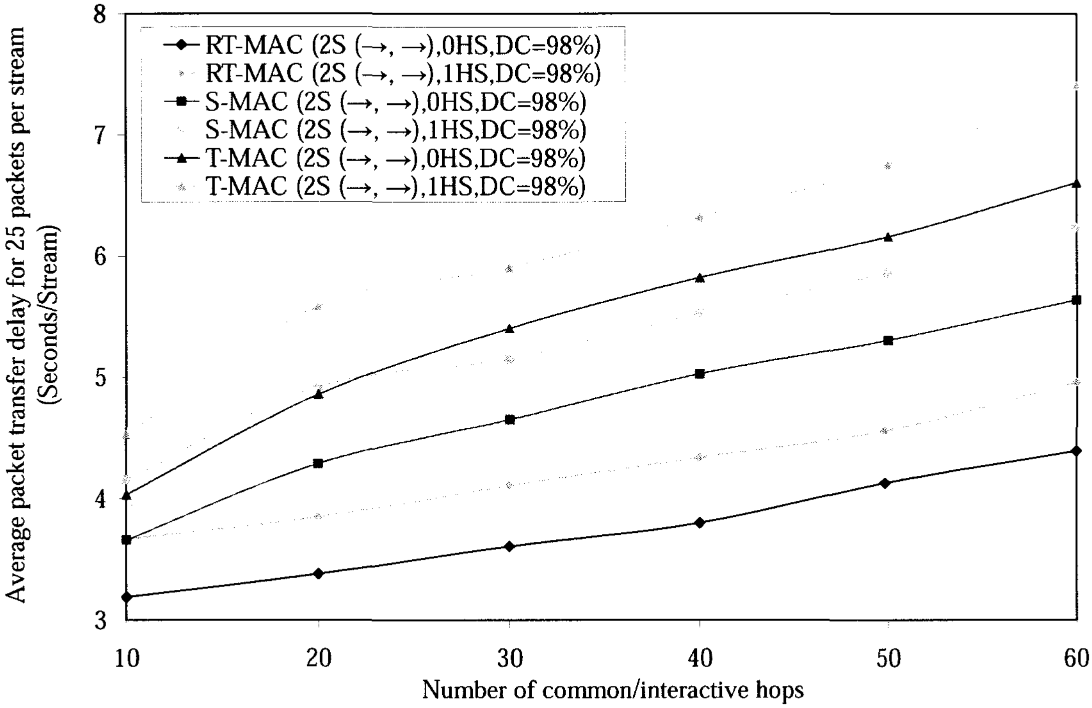

Figure 7.19: Average packet transfer delay pattern with interactive streams at 98 percent duty cycle

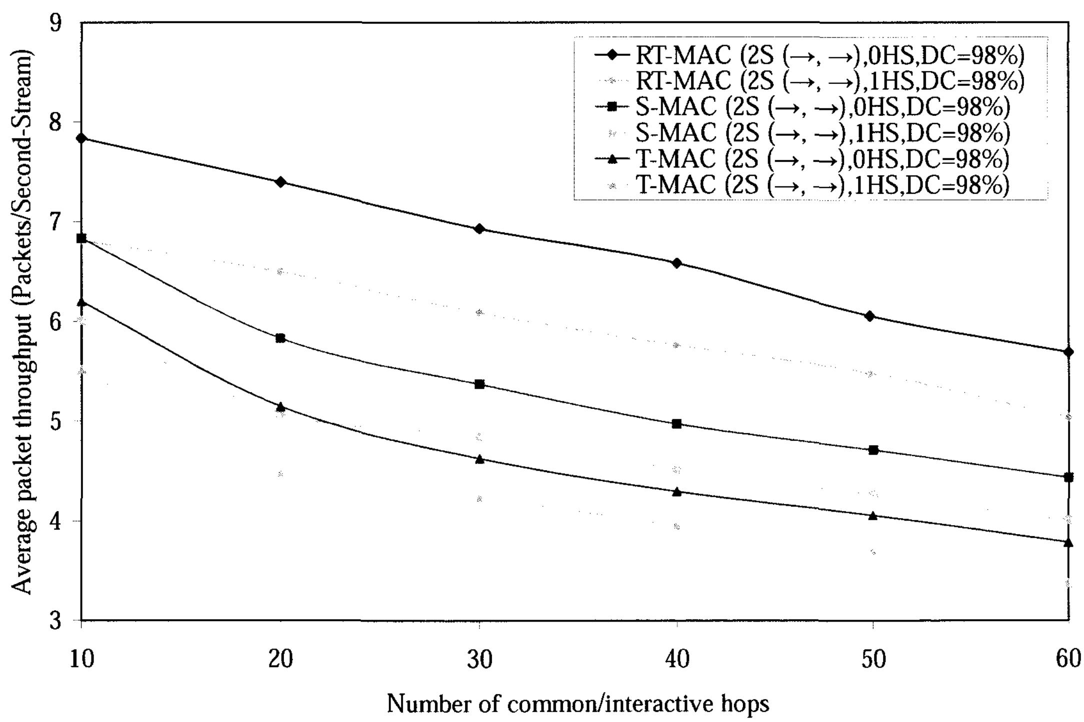

Figure 7.20: Average packet throughput pattern with interactive streams at 98 percent duty cycle

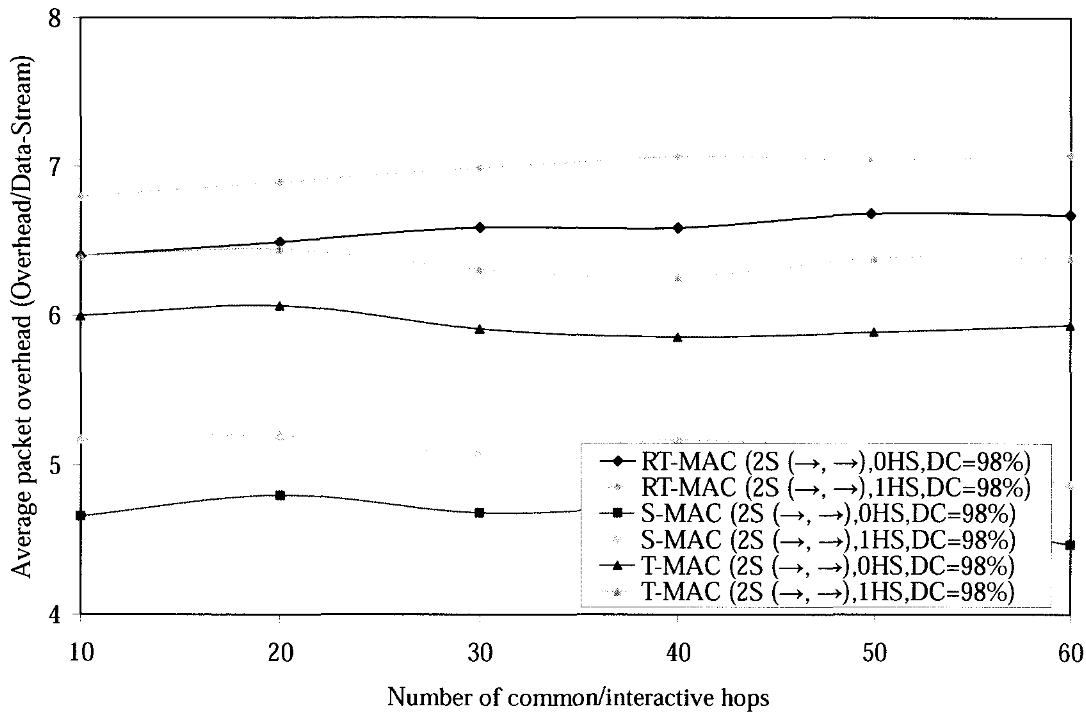

Figure 7.21: Average packet overhead pattern with interactive streams at 98 percent duty cycle

increased to 20 percent, as shown in Figure 7.17. Figure 7.18 shows the average packet overhead pattern of RT-MAC at 20 percent duty cycle. This figure shows that the packet overhead is higher for the parallel streams travelling in opposite direction with one hop separation. As explained above for Figure 7.13, the parallel streams travelling in opposite direction with one hop separation uses more number of CC control packet, therefore, it shows maximum control overhead as compared to other cases.

Figures 7.19 and 7.20 compared the average packet transfer delay pattern and the average packet throughput pattern of RT-MAC with S-MAC and T-MAC protocols at 98 percent duty cycle. These figures show that RT-MAC has lower average packet transfer delay and higher average packet throughput than S-MAC and T-MAC protocols. It is because of lesser number of collisions and maximum spatial channel reutilization in RT-MAC due to its feedback based MAC strategy.

Figure 7.21 shows that the average packet overhead in RT-MAC is higher than S-MAC and T-MAC protocols due to usage of CC control packet in RT-MAC. As explained in

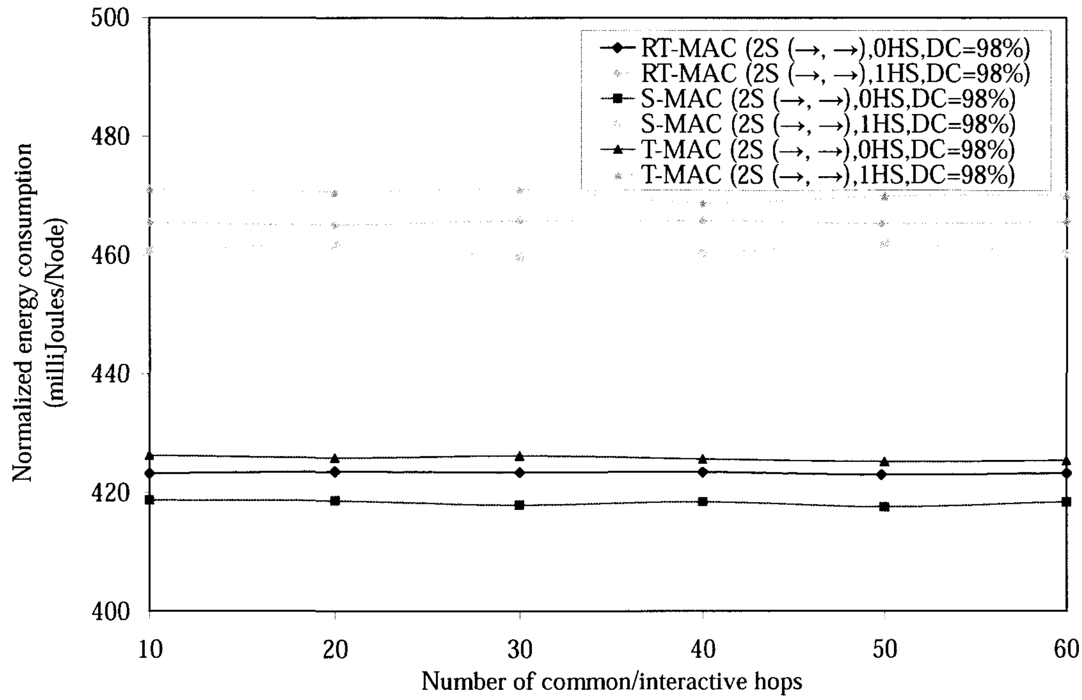

Figure 7.22: Normalized energy consumption pattern with interactive streams at 98 percent duty cycle

Section 7.2.1, the negative effect of the increase in control overhead in RT-MAC is compensated by the reduction in the number of collisions, lower contention duration and better spatial channel reutilization. Thus, an increase in average packet overhead does not increase the average packet transfer delay in RT-MAC. Figure 7.22 shows the normalized energy consumption pattern with interactive streams for RT-MAC, S-MAC and T-MAC protocols at 98 percent duty cycle. This figure signifies that RT-MAC achieves its timing objectives in multi-stream scenario with the energy performance comparable to S-MAC and T-MAC protocols.

7.3 Summary of the chapter

This chapter presents simulation results of RT-MAC protocol in single-stream as well as multi-stream scenario. The performance of RT-MAC is compared with other real-time and general purpose MAC protocols. It is observed that RT-MAC provide a lower end-to-end delay deadline than VTS, S-MAC and T-MAC with the energy performance comparable to these protocols.

The next chapter concludes this research work. Some future research directions are also given in the next chapter.

Chapter 8: Conclusion and future work

This thesis presents RT-MAC in both single-stream and multi-stream scenario. The main contribution of this research is to provide a soft real-time MAC protocol that guaranties bounded and minimum end-to-end delay with no hardware assumptions and is capable to work with random network topology. RT-MAC support operation of MAC with and without duty cycle. Duty cycle mode is generally desired when energy conservation is also a goal along with timing consideration. RT-MAC can vary duty cycle during run time, which facilitates it to provide delay guarantees in multi-stream scenario. RT-MAC is capable of working at much lower duty cycle operations than other contention based protocols, which can increase the network lifetime substantially when there are no event reporting scenarios. RT-MAC is useful for a variety of event driven and periodic soft real-time WSN applications. This thesis presents state analysis that facilitates determination of the lower end-to-end delay bound as well as providing greater insight into the operations of RT-MAC protocol. It also shows that RT-MAC is a highly fault tolerant protocol. Discrete Markov Chain analysis of RT-MAC shows the behavior of the protocol in some realistic conditions where fault may also occur.

In this research, it is found that a feedback mechanism enables RT-MAC to provide end-to-delay guarantees as well as a lower end-to-end packet transfer delay for a soft real-time WSN application without any unusual increase in energy consumption. Simulation results show that delay and energy behaviors of RT-MAC and S-MAC are similar. However, unlike S-MAC, RT-MAC is additionally able to provide delay guarantees, which makes it a real-time protocol. It is also observed that being a contention based real-time MAC protocol, RT-MAC is able to provide lower delay bounds and lower energy consumption as compared to VTS, which is based on the TDMA scheme. Specifically, RT-MAC is two times faster at lower duty cycle operations, and seven times faster at higher duty cycle operations than VTS protocol with 20 hop network. Simulation study shows that RT-MAC provides higher and more stable packet throughput than S-MAC and T-MAC protocol. It is observed that the packet throughput of RT-MAC is higher by 8 and 21 percent respectively, than the packet throughput of S-MAC and T-MAC protocol at higher packet arrival rate, i.e., higher than 20 packets per seconds.

In this thesis, it is observed that feedback based MAC schemes are more preferable to provide real-time communication as compared to reservation based MAC schemes. It is because feedback based MAC schemes provide more definitive and predictability to data transmission pattern. It is also observed that contention based MAC schemes are better for soft real-time communication in WSN (as compared to scheduling based MAC schemes) due to their flexibility in changing duty cycle during run time and better scalability.

In the future, Markov analysis in multi-stream case needs to be done. It is expected to be computationally complex due to the large number of possible states in multi-stream mode of RT-MAC. RT-MAC can be extended to provide service differentiation at MAC layer. For this, the capability of varying transmitting power of sensor nodes during run time can be added in RT-MAC to reduce or increase number of hops between a source and destination node pair in a stream to adjust end-to-end delays as per the priority of data packets. Varying transmission power is possible in newer sensor node. Here, the effect of increased transmission range on the neighboring node needs to be properly analyzed. A prototype implementation of RT-MAC protocol will be done on sensor nodes.

References

[1] Abdelzaher, T.F., Prabh, S., and Kiran, R., "On real-time capacity limits of multihop wireless sensor networks", Proceedings of Real-Time Systems Symposium, 2004, pp. 359-370.

[2] Afonso, J.A., Rocha, L.A., Silva, H.R., and Correia, J.H., "MAC protocol for low-power real-time wireless sensing and actuation", IEEE International Conference on Electronics, Circuits, and Systems, art. no. 4263600, 2006, pp. 1248-1251.

[3] Akyildiz I. F., Su W., Sankarasubramaniam Y., and Cayirci E., "A survey on sensor networks, IEEE Communications Magazine", Vol. 40, No. 8, 2002, pp. 102-114.

[4] Akyildiz I. F., Su W., Sankarasubramaniam Y., and Cayirci E., "Wireless sensor networks: a survey", Computer Networks, Vol. 38, No. 4, March 2002, pp. 393-422.

[5] Ali, M., Bohm, A., and Jonsson, M., "Wireless Sensor Networks for Surveillance Applications – A Comparative Survey of MAC Protocols", The Fourth International Conference on Wireless and Mobile Communications, ICWMC 2008, 2008, pp. 399-403.

[6] Arampatzis, T., Lygeros, J., Manesis, S., "A survey of applications of wireless sensors and wireless sensor networks", Proceedings of the 20th IEEE International Symposium on Intelligent Control, ISIC 2005, and the 13th Mediterranean Conference on Control and Automation, MED 2005, 2005, art. no. 1467103, pp. 719-724.

[7] Baroudi, U., "EQoSA: Energy and QoS aware MAC for wireless sensor networks", 9th International Symposium on Signal Processing and its Applications, ISSPA 2007, 2007, art. no. 4555601, pp. 1-4.

[8] Bharathidasan, A., and Ponduru, V.A.S., "Sensor Networks: An Overview", Technical report, Dept. of Computer Science, University of California at Davis, 2002. [Online], Available: wwwcsif.cs.ucdavis.edu/~bharathi/sensor/survey.pdf.

[9] Bhardwaj, M., Garnett, T., and Chandrakasan, A.P., Upper bounds on the lifetime of sensor networks, IEEE International Conference on Communications 3, 2001 pp. 785-790.

[10] Caccamo,M., Zhang, L. Y., Sha, L., and Buttazzo, G., "An implicit prioritized access protocol for wireless sensor networks", IEEE Real-Time System Symposium RTSS 2002, 2002, pp. 39-48.

[11] Caccamo,M., and Zhang, L. Y., "The capacity of implicit EDF in wireless sensor networks", 15th IEEE Euromicro Conference on Real-Time Systems ECRTS 2003, 2003, pp. 267-275.

[12] Capella, J.V., Ors, R., Bonastre, A., and Serrano, J.J., "New challenges in wireless sensor networks: Fault tolerance and real time", Proceedings of the IEEE International Conference on Industrial Technology, art. no. 1600852, 2005, pp. 1385-1391.

[13] Channa, M.I., and Memon, I., "Real time traffic support in wireless sensor networks", Student Conference on Engineering Sciences and Technology, SCONEST 2005, art. no. 4382896.

[14] Chee-Yee C., and Kumar, S.P., "Sensor networks: evolution, opportunities, and challenges", Proceedings of the IEEE, Vol. 91, No. 8, August 2003, pp. 1247-1256.

[15] Chen, J., Zhu, P., and Qi, Z., "PR-MAC: Path-oriented Real-time MAC protocol for wireless sensor network", Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4523, 2007, pp. 530-539.

[16] Chen, Z., and Khokhar, A., "Self Organization and Energy Efficient TDMA MAC Protocol by Wake up for Wireless Sensor Networks", First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON 2004, 4-7 October 2004, pp. 335-341.

[17] Culler, D., Estrin, D., and Srivastava, M., "Overview of Sensor Networks", Published by the IEEE Computer Society, Vol. 37, No. 8, 2004, pp. 41-49.

[18] Czapski, P.P., "A survey: MAC protocols for applications of wireless sensor networks", IEEE Region 10 Annual International Conference, Proceedings/TENCON, art. no. 4142132, 2006.

[19] Demirkol, I., Ersoy, C., and Alagöz, F., "MAC protocols for wireless sensor networks: A survey", IEEE Communications Magazine, Vol. 44, No. 4, 2006, pp. 115-121.

[20] Egea-lópez, E., Vales-alonso, J., Martínez-sala, A.S., García-haro, J., Pavón-mariño, P., and Bueno delgado, M.V., "A wireless sensor networks MAC protocol for real-time applications, Personal and Ubiquitous Computing", Volume 12, Issue 2, February 2008, pp. 111-122.

[21] Egea-lópez, E., Vales-alonso, J., Martinez-sala, A.S., García-haro, J., Pavón-mariño, P., and Bueno-delgado, M.V., "A real-time MAC protocol for wireless sensor

networks: Virtual TDMA for Sensors (VTS)", Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 3894 LNCS, 2006, pp. 382-396.

[22] Ergen, S.C. and Varaiya, P., "PEDAMACS: power effcient and delay aware medium access protocol for sensor networks", IEEE Transections on Mobile Computing Vol. 5, No. 7, 2006, pp. 920-930.

[23] Francomme, J., Mercier, G., and Val, T., "A simple method for guaranteed deadline of periodic messages in 802.15.4 cluster cells for control automation applications", IEEE Conference on Emerging Technologies and Factory Automation, ETFA 2006, September 2006, pp. 270-277.

[24] Ganeriwal, S., Kumar, R., and Srivastava, M. B., "Timing-sync protocol for sensor networks", 1st international conference on Embedded networked sensor systems, November 05-07, 2003, pp. 138 - 149.

[25] Gebali F., Analysis of Computer and Communication Networks, Chapters 3 and 4, Springer Publishers, New York, USA,2008.

[26] Halkes, G.P., Van dam, T., and Langendoen, K.G., "Comparing energy-saving MAC protocols for wireless sensor networks", Mobile Networks and Applications Vol. 10, No. 5, 2005, PP. 783-791.

[27] Hoesel, L. V., and Havinga, P., "A Lightweight Medium Access Protocol(LMAC) for Wireless Sensor Networks:Reducing Preamble Transmissions and Transceiver State Switches", INSS 2004, 2004.

[28] Kalidindi, R., Ray, L., Kannan, R., and Iyengar, S., "Distributed energy aware MAC layer protocol for wireless sensor networks", International Conference on Wireless Networks, ICWN 2003, 23-26 June 2003, pp. 282-286.

[29] Kim, J., Lim, J.-H., Pelczar, C., and Jang, B., "RRMAC: A sensor network MAC for real time and reliable packet transmission", Proceedings of the International Symposium on Consumer Electronics, ISCE 2008, April 2008, art. no. 4559491, pp. 1-4.

[30] Kim, J., Lim, J.-H., Pelczar, C., and Jang, B., "Sensor network for traffic safety", IEEE Vehicular Technology Conference, 2008, art. no. 4525886, pp. 3052-3056.

[31] Kim, T.H., and Choi, S., "Priority-based delay mitigation for event-monitoring IEEE 802.15.4 LR-WPANs", IEEE Communication Letters, Vol. 10, No. 3, 2006, pp. 213-215.

[32] Kleinrock L., and Tobagi, F. A., "Packet switching in radio channels: Part 2-The hidden node problem in carrier sense multiple access modes and the busy tone solution", IEEE Transactions on Communication, vol. COM-23, no. 12, December 1975, pp. 1417-1433.

[33] Koubâa, A., Alves, M., Nefzi, B., and Song, Ye-Q., “Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks”, proceedings of the 5th International Workshop on Real-Time Networks, RTN 2006, 2006, Dresden, Germany.

[34] Koubâa, A., Alves, M., and Tovar, E., "Time Sensitive IEEE 802.15.4 Protocol", Sensor Networks and Configurations: Fundamentals, Techniques, Platforms, and Experiments, Book chapter 2, Ed. N. P. Mahalik, Springer Berlin Heidelberg, Germany, 2007, pp. 19-49.

[35] Kredo II, K., Mohapatra, P., “Medium access control in wireless sensor networks”, Computer Networks: The International Journal of Computer and Telecommunications Networking, vol.51, no.4, March 2007, pp. 961-994.

[36] Krohn, A., Beigl, M., Decker, C., and Zimmer, T., “Tomac- real-time message ordering in wireless sensor networks using the MAC layer”, In Proceedings of the 2nd International Workshop on Networked Sensing Systems, INSS 2005, 2005.

[37] Lewis, F. L., “Wireless Sensor Networks”, Chapter 4 in Smart Environments: Technologies, Protocols, and Applications, John Wiley Publishers, Ed. D.J. Cook and S.K. Das, New York, 2004.

[38] Li, H., Shenoy, P., and Ramamritham, K., “Scheduling messages with deadlines in multi-hop real-time sensor networks”, Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2005, 2005, pp. 415-425.

[39] Li, H., Shenoy, P., and Ramamritham, K., “Scheduling Communication in Real-Time Sensor Applications”, Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium, Vol. 10, RTAS 2004, pp. 10-18.

[40] Li, Y., Chen, C. S., Song, Ye-Q., Wang, Z., “Real-time QoS support in wireless sensor networks: a survey”, 7th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems, FeT 2007.

[41] Lu, G., Krishnamachari, B., and Raghavendra, C.S., “An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks”, Proceedings of the 18th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2004, 2004, pp. 3091-3098.

[42] Lu, G., Krishnamachari, B., and Raghavendra, C.S., “An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks”, Wireless Communications and Mobile Computing, Vol. 7, No. 7, 2007, pp. 863-875

[43] Manshaei, M.H., Cantieni, G.R., Barakat, C., and Turletti, T., “Performance analysis of the IEEE 802.11 MAC and physical layer protocol”, Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM 2005, 13-16 June 2005, pp.88-97.

[44] Niberg, T., Dulman, S., Havinga, P., Hoesel, L.V., and Wu, J., "Collaborative Algorithms for Communication in Wireless Sensor Networks". Ambient Intelligence:Impact on Embedded Systems, Kluwer Academic Publishers, November 2003.

[45] Onda K., Ohshimaa T., Nakayamaa M., Fukudaa K., and T. Arakia, "Thermal behavior of small lithium-ion battery during rapid charge and discharge cycles", Journal of Power Sources, Vol. 158, No. 1, 14 July 2006, pp. 535-542.

[46] Prabh, K.S., and Abdelzaher, T.F., "On scheduling and real-time capacity of hexagonal wireless sensor networks", Euromicro Conference on Real-Time Systems, 2007, art. no. 4271688, pp. 136-145.

[47] Puccinelli, D., and Haenggi, M., "Wireless sensor networks: applications and challenges of ubiquitous sensing", IEEE Circuits and Systems Magazine, Vol. 5, No. 3, 2005, pp.19-31.

[48] Raghunathan, V., Ganeriwal, S., and Srivastava, M., "Emerging techniques for long lived wireless sensor networks", IEEE Communications Magazine, Vol. 44, No. 4, April 2006, pp. 108-114.

[49] Ray, S., Carruthers, J.B., and Starobinski, D., "RTS/CTS-Induced Congestion in Ad Hoc Wireless LANs", Proceedings of IEEE Wireless Communications and Networking Conference, WCNC 2003, 2003, pp. 1516-1521

[50] Ray, S., Starobinski, D., and Carruthers, J. B., "Performance of wireless networks with hidden nodes: A queueing-theoretic analysis", Elsevier Journal of Computer Communications - Special Issue Performance Wireless LANs, PANs, Ad-Hoc Networks, vol. 28, no. 10, June 2005, pp. 1179-1192.

[51] Ray, S., and Starobinski, D., "On false blocking in RTS/CTS-based multihop wireless networks", IEEE Transactions on Vehicular Technology, vol. 56, no. 2, 2007, pp. 849-862.

[52] Ren, B., Xiao, J., Ma, J., and Cheng, S., "An energy-conserving and collision-free MAC protocol based on TDMA for wireless sensor networks", Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 3794, 2005, pp. 603-612.

[53] Ross, S. M., Introduction to Probability Models, Ninth Edition, Academic press Publishers, 2006, Chapter 4.

[54] Rowe, A., Mangharam, R., and Rajkumar, R., "RT-Link: A time-synchronized link protocol for energy-constrained multi-hop wireless networks", 3rd Annual IEEE Communications Society on Sensor and Adhoc Communications and Networks, SECON 2006, 2006, art. no. 4068297, pp. 402-411.

[55] Singh, B.K., and Tepe, K.E., "A Novel Real-time MAC Layer Protocol for Wireless Sensor Network Applications", IXth IEEE International Conference on Electro/Information Technology, EIT 2009.

[56] Singh, B.K., and Tepe, K.E., "Feedback based Real-time MAC (RT-MAC) protocol for wireless sensor networks", IEEE Global Communications Conference, GLOBECOM 2009.

[57] Singh, B.K., and Tepe, K.E., "Feedback based MAC Protocol for Real-time Data Streaming in Wireless Sensor Networks", IEEE/ACM Transactions on Networking, Submitted in August 2009.

[58] Singh, B.K., and Tepe, K.E., "Feedback based solution to false blocking problem at MAC layer in Wireless Sensor Networks", IEEE Communications Letters, Submitted in January 2010.

[59] Stankovic, J.A., Abdelzaher, T.F., Lu, C., Sha, L., and Hou, J.C., "Real-Time Communication and Coordination in Embedded Sensor Networks", Proceedings of the IEEE, Vol. 91 No. 7, 2003, pp. 1002-1022.

[60] Stewart, W.J., Introduction to Numerical Solutions of Markov Chains, Princeton University, Press, Princeton, New Jersey, 1994.

[61] Trivedi, K.S., Probability and Statistics with Reliability, Queuing, and Computer Science Applications, 2nd Edition, Wiley-Interscience Publishers, 2001, Chapter 7.

[62] Van dam, T., and Langendoen, K., "An Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks", 1st international conference on Embedded networked sensor systems, SenSys'03, 5-7 November 2003, pp. 171-180.

[63] Varga A., "The OMNeT++ discrete event simulation system", European Simulation Multiconference ESM'2001, 2001, Prague, Czech Republic.

[64] Varga, A., and Hornig, R., An overview of the OMNeT++ simulation environment, 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, March 03-07, 2008, Marseille, France, Art. 60.

[65] Warland, J., and Varaiya, R., "High-Performance Communication Networks, Moragan Kaufmann", San Francisco, 2000.

[66] Watteyne, T., Augé-blum, I., and Ubéda, S., "Dual-Mode Real-Time MAC protocol for Wireless Sensor Networks: a Validation/Simulation Approach", ACM International Conference Proceeding Series 138, art. no. 1142683, Proceedings of the First International Conference on Integrated Internet Ad hoc and Sensor Networks, InterSense '06, 2006.

[67] Watteyne, T., and Augé-blum, I., "Proposition of a hard real-time MAC protocol for wireless sensor networks", Proceedings of IEEE Computer Society's Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, MASCOTS 2005, 2005, art. no. 1521180, pp. 533-536.

[68] Woodward, M. E., Communication and Computer Networks, IEEE Computer Society Press, Los Alamitos, CA, 1994.

[69] Xu, N., "A Survey of Sensor Network Applications". [Online], Available: http://courses.cs.tamu.edu/rabi/cpsc617/resources/sensor%20nw-survey.pdf.

[70] Xue, Y., Ramamurthy, B., and Vuran, M.C., "A service-differentiated real-time communication scheme for wireless sensor networks", 33rd IEEE Conference on Local Computer Networks, LCN 2008, October 2008, art. no. 4664276, pp. 748-755.

[71] Ye, W., Heidemann, J., and, Estrin, D., "An energy-efficient MAC protocol for wireless sensor networks", Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 3, INFOCOM 2002, 23-27 June 2002, pp. 1567-1576.

[72] Ye, W., Heidemann, J. and, Estrin, D., "Medium access control with coordinated adaptive sleeping for wireless sensor networks", IEEE/ACM Transactions on Networking, Vol. 12, No. 3, June 2004, pp. 493-506.

[73] Yick J., Mukherjee B., and Ghosal D., "Wireless sensor network survey, Computer Networks", Vol. 52, No. 12, August 2008, pp. 2292-2330.

[74] Zhang, Y. , Zheng, S., and Xiong, S., "A Scheduling Algorithm for TDMA-Based MAC Protocol in Wireless Sensor Networks", First International Workshop on Education Technology and Computer Science, ETCS 2009, 2009, vol. 3, pp.148-151.

[75] Crossbow wireless sensor networks site (October 2009). [Online]. Available: http://www.xbow.com/.

[76] EYES project site (October 2009). [Online]. Available: http://www.eyes.eu.org/index.htm.

[77] IEEE Standard 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)-(2003) (October 2009). [Online]. Available: http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf.

[78] OMNeT++ Community site (October 2009). [Online]. Available: http://www.omnetpp.org/.

[79] OMNET++ report. (November 2009). [Online]. Available: http://www.sce.carleton.ca/faculty/wainer/students/DEVSnet/_private/Omnet%20report.doc

[80] RIAC website (November 2009). [Online]. Available: http://www.theriac.org/DeskReference/viewDocument.php?id=95.

[81] Tutorial of OMNET++. (November 2009). [Online]. Available: http://personal.stevens.edu/~hli5/TutorialofOMNET.htm

[82] Zigbee Alliance Home Page (October 2009). [Online]. Available: http://www.zigbee.org.

Appendix A: List of publication

Journal papers

[C1] Brajendra Kumar Singh, and Kemal Ertugrul Tepe, "Feedback based MAC Protocol for Real-time Data Streaming in Wireless Sensor Networks", IEEE/ACM Transactions on Networking, [First phase reviews are received. The updated paper is to be submitted soon for the next review phase].

[C2] Brajendra Kumar Singh, and Kemal Ertugrul Tepe, "Feedback based solution to false blocking problem at MAC layer in Wireless Sensor Networks", IEEE Communications Letters, Submitted in May 2010.

[C3] Brajendra Kumar Singh, and Kemal Ertugrul Tepe, "Multi-stream support in Real-Time MAC (RT-MAC) protocol for Wireless Sensor Networks", To be submitted, Thesis chapter No. 6 and 7.

[C4] "A survey on real-time MAC protocols for Wireless Sensor Networks ", To be submitted, Thesis chapter No. 2.

Conferences papers

[C5] B.K. Singh and K.E. Tepe, "Feedback based Real-time MAC (RT-MAC) protocol for wireless sensor networks", IEEE Global Communications Conference, GLOBECOM 2009.

[C6] Brajendra Kumar Singh and Kemal Ertugrul Tepe, "A Novel Real-time MAC Layer Protocol for Wireless Sensor Network Applications", IXth IEEE International Conference on Electro/Information Technology, EIT 2009.

[C7] Izhar Ahmed, K.E. Tepe and B. K. Singh, "Reliable Coverage Area Based Link Expiration Time (LET) Routing Metric for Mobile Ad Hoc Networks", First International Conference on Ad Hoc Networks, ADHOCNETS 2009.

[C8] "Markov Analysis of RT-MAC protocol for Wireless Sensor Networks in single-stream scenario", To be submitted, Thesis chapter No. 5.

VITA AUCTORIS

Brajendra Kumar Singh was born in 1974 in India. He graduated from Jiwaji University in 1996, Gwalior in India, where he obtained Bachelors of Engineering (B.E.) in Electronics. He joined Indian Space Research Organization (ISRO), Ahmedabad in India, as Engineer in 1998. He obtained Masters of Engineering degree from Rajiv Gandhi Technical University, Bhopal in India in 2001. Since January 2006, he is on leave from ISRO to pursue his doctoral research in the department of Electrical and Computer Engineering at the University of Windsor, Ontario, Canada.