LOW DELAY AND LOW ENERGY CONTENTION BASED SYNCHRONOUS MAC PROTOCOLS FOR EVENT-DRIVEN WIRELESS SENSOR NETWORKS

Ripudaman Singh

LOW DELAY AND LOW ENERGY CONTENTION BASED SYNCHRONOUS MAC PROTOCOLS FOR EVENT-DRIVEN WIRELESS SENSOR NETWORKS

A

Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

By

RIPUDAMAN SINGH

DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781 039, ASSAM, INDIA

November, 2018

Certificate

This is to certify that the thesis entitled "**LOW DELAY AND LOW ENERGY CONTENTION BASED SYNCHRONOUS MAC PROTOCOLS FOR EVENT-DRIVEN WIRELESS SENSOR NETWORKS**", submitted by **Ripudaman Singh** (126102018), a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under our supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated:
Guwahati.

Dr. Brijesh K. Rai
Assistant Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

Dated:
Guwahati.

Prof. Sanjay K. Bose
Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

To my grand parents and parents

Acknowledgements

First and foremost, I feel it as a great privilege in expressing my deepest and most sincere gratitude to my supervisors Dr. Brijesh K. Rai & Prof. Sanjay K. Bose, for their excellent guidance throughout my PhD tenure. Their kindness, dedication and attention to the details has been a source of great inspiration to me. My heartfelt thanks to my supervisors for their unlimited support and patience that they have shown towards me. Their emphasis on clear communication helped me a lot. They have enriched my life in many significant ways. I thank them from the bottom of my heart for always being there with me during all kinds of distress.

I would like to thank my doctoral committee members: Prof. K. Rakhesh Singh, Dr. T. Venkatesh, and Dr. Kalpana Dhaka for sparing time out of their busy schedule to evaluate my progress and enrich this work with their valuable suggestions and feedbacks. I would also like to thank other faculty members for helping me with my research work in innumerable ways.

During my PhD, I found really wonderful lab-mates and fellow research scholars. I had never expected such sensitive and helping people existing in highly professional research institute like IITs. My sincere thanks to my seniors Dr. Sam Darshi and Dr. Brijesh Kumbhani for some of the extremely important discussions and feedbacks during the initial days of my PhD. I have no words to thank my friends Deepak, Arijit, Shivanshu, and Shashank Sir. These guys made my stay at IIT Guwahati memorable. I will cherish the time spent with them. I thank them all for their friendship.

I express my deep appreciation to my wife Neetu for her unconditional support and understanding for all these moments. This PhD would not have been possible without her help and support. I extend my love to my daughter Rinee. I extend deepest gratitude to my uncle Abhayveer for always standing by my side and supporting me with his thoughts, suggestions, and guidance not only during this PhD tenure but rather throughout my life.

Lastly, I extend my sincere thanks to all the staff members from EEE office and Academic office for helping me out in all sorts of ways during my stay at IITG.

Ripudaman Singh

Abstract

Monitoring of delay-sensitive events, such as fire and intrusions, in unsafe regions is an important class of wireless sensor network (WSN) applications. In event-driven WSNs, sensor nodes are densely deployed and they operate with a low duty-cycle to reduce their energy consumption in *idle listening*. The following two characteristics of duty-cycled contention based synchronous MAC protocols make them the very suitable for delay-sensitive event monitoring applications, (1) they are robust to dynamic topology changes, and (2) in such protocols, a data packet can travel multiple hops in one cycle. However, in the existing contention based synchronous MAC protocols, in a low duty-cycle WSN, very few nodes can schedule the forwarding of their data packets when multiple sensor nodes with data packets to send lie in the carrier sensing range of each other and a node cannot schedule a longer flow. Therefore, they encounter long end-to-end transmission delays (E2ETD) in reporting the detected event to the sink node in a dense multi-hop WSN.

In this thesis, we propose two new contention based synchronous MAC protocols and a new framework to reduce the E2ETD in event-driven dense multi-hop WSNs. The first protocol proposed by us reduces the E2ETD by allowing multiple nodes to forward their data packets through a single scheduled flow.

The new framework proposed in this thesis improves the E2ETD performance of implemented contention based synchronous MAC protocols by mapping their m (> 1) cycles data transmission processes in a single duration cycle. In this framework, dense deployment of sensor nodes is exploited to reduce idle listening.

Our second proposed protocol reduces E2ETD by providing multiple chances to each node to succeed in data transmission scheduling in a cycle. Algorithms proposed in this protocol help in reducing the energy consumption in transmission, reception, and *idle listening*.

Contents

List of Figures xvii

List of Tables xxi

List of Acronyms xxiii

List of Publications xxv

1 Introduction 1

1.1 Issues Related to Duty-Cycle MAC Protocols . . . 3

1.2 Motivation of the Present Work . . . 6

1.3 Thesis Contribution . . . 8

1.4 Thesis Organization . . . 9

2 Literature Survey: Review of Related Work 13

2.1 MAC Protocols for WSNs . . . 14

2.1.1 Contention-Free MAC Protocols . . . 14

2.1.2 Contention Based MAC Protocols . . . 16

2.1.2.1 Contention Based Asynchronous MAC Protocols . . . 18

2.1.3 Hybrid MAC Protocols . . . 19

2.2 Contention Based Synchronous MAC Protocols . . . 19

2.2.1 *Generic* Contention Based Synchronous MAC Protocols . . . 20

2.2.2 *Staggered* Contention Based Synchronous MAC Protocols . . . 29

2.3 Synchronization Protocols for WSNs . . . 34

3 A Low Delay Cross-Layer MAC Protocol for *k-covered* Event-Driven WSNs 37

3.1 Introduction . . . 38

3.2 Description of LDC-MAC . . . 40

3.2.1 Data Packet Transmission Flow Setup Process in DW . . . 40

3.2.2 DTS/DRS Assignment Process . . . 41
3.2.3 Process Followed by a SS Node and a PS Node for Data Packet Transmission . 42
3.2.4 Illustration of the Data Packet Transmission Process . . . 43
3.3 Results . . . 45
3.3.1 PDR Comparison . . . 46
3.3.2 AE2ETD Comparison . . . 48
3.3.3 AEC Comparison . . . 51
3.3.4 Network Life Comparison . . . 52
3.4 Conclusions . . . 53

4 A Novel Framework to Enhance the Performance of *Generic* Contention Based Synchronous MAC Protocols in Event-Driven WSNs 55

4.1 Introduction . . . 56
4.2 Description of the Proposed Framework . . . 57
4.2.1 Setup Phase . . . 58
4.2.2 Data Transmission Phase . . . 60
4.2.3 Illustration of Data Packet Transmission Process . . . 62
4.2.4 Addition/Deletion of Sensor Nodes in DTP . . . 63
4.3 Determination of Optimum m . . . 64
4.3.1 Determination of $AHD(m)$ . . . 64
4.3.2 Determination of $E[X_S(m, d(S, s_i))]$ . . . 66
4.3.3 Determination of $E[X_P(m, d(S, s_i))]$ . . . 67
4.3.4 Validation of m_{opt} Determination Process . . . 68
4.4 Results . . . 70
4.4.1 Results for RMAC and PRMAC Implementation . . . 72
4.4.2 Results for CL-MAC and LDC-MAC Implementation . . . 77
4.5 Conclusions . . . 79

5 A Joint Routing and MAC Protocol for Transmission Delay Reduction in Many-to-One Communication Scenarios for Event-Driven WSNs 81

5.1 Introduction . . . 82
5.2 Proposed Cycle Structure . . . 84
5.3 Synchronization and Proposed Changes in SYNC Packet . . . 84

5.4 Description of JRAM . . . 86
5.4.1 Setup Phase . . . 86
5.4.2 Data Transmission Phase . . . 91
5.4.3 Data Transmission Process . . . 92
5.4.4 Addition/Deletion of Sensor Nodes in DTP . . . 94
5.5 Determination of α' . . . 95
5.6 Determination of Optimum m . . . 96
5.7 Results . . . 99
5.7.1 Impact of m on JRAM's Performance . . . 100
5.7.2 Comparison of AE2ETD . . . 101
5.7.3 Comparison of PDR . . . 102
5.7.4 Comparison of AEC . . . 106
5.8 Conclusions . . . 106

6 Discussion and Future Work **109**

6.1 Summary of Contributions and Discussions . . . 110
6.2 Suggestions for Future Work . . . 111

A Determination of $\mathbf{A}^{\mathbf{R}}_{(0,\delta)} \cap \mathbf{A}_u$ **113**

List of Figures

1.1 Cycle structure followed by the network nodes in *generic* contention based synchronous MAC protocols. 5

1.2 Thesis organization. 11

2.1 Classification of MAC protocols proposed for WSNs. 15

2.2 Operation of a TDMA based MAC protocol. 16

2.3 Operation of a contention based MAC protocol with competition. 17

2.4 Classification of duty-cycled contention based MAC protocols proposed for WSNs. . . 17

2.5 Data packet transmission process of SMAC [37]. 21

2.6 Data packet transmission process of RMAC [42]. 23

2.7 Data packet transmission process of PRMAC [48]. 24

2.8 Data packet transmission process of DW-MAC [51]. 26

2.9 Illustration of EACK mechanism of CL-MAC [55]. 27

2.10 Illustration of multiple multi-hop flow scheduling feature of CL-MAC [55]. 28

2.11 *Staggered* wakeup schedules of the sensor nodes lying at $i-1$, i, and $i+1$ hops distance from the sink node. 30

2.12 Data packet transmission process of PDC [62]. 32

2.13 Cycle structure followed by the network nodes in CROP-MAC [64]. 33

2.14 Data packet transmission process of CROP-MAC [64]. 33

3.1 Network containing 50 randomly deployed sensor nodes and one sink node (S). 43

3.2 Data packet transmission process of LDC-MAC corresponding to the network scenario shown in Fig. 3.1. 44

3.3 Comparison of PDR in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. 47

3.4 Comparison of PDR in case of $A = 2400\ m$, $n = 225$, $t = 2$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. 47
3.5 Comparison of PDR in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 100\ m$, $CSR = 200\ m$, and $ST = 2000\ s$. 48
3.6 Comparison of AE2ETD in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. 49
3.7 Comparison of AE2ETD in case of $A = 2400\ m$, $n = 225$, $t = 2$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. 49
3.8 Comparison of AE2ETD in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 100\ m$, $CSR = 200\ m$, and $ST = 2000\ s$. 50
3.9 Comparison of AEC in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. 50
3.10 Comparison of AEC in case of $A = 2400\ m$, $n = 225$, $t = 2$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. 51
3.11 Comparison of AEC in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 100\ m$, $CSR = 200\ m$, and $ST = 2000\ s$. 51
3.12 Comparison of network life in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 100\ m$, $CSR = 200\ m$, and $ST = 2000\ s$. 52

4.1 Cycle structure followed by the network nodes in Setup Phase. 58
4.2 Partition of sensor nodes into DSs. 61
4.3 Cycle structure followed by the network nodes in (a) our proposed framework in DTP, and (b) existing *generic* contention based synchronous MAC protocols. 62
4.4 Illustration of data transmission process, in case of $m = 2$, when RMAC is implemented in the framework. 63
4.5 Depiction of various parameters used in m_{opt} determination. 65
4.6 Variations in $AHD\,(m)$ and $m.AHD\,(m)$ with m for $n = 400,\ 625,$ and 900. 69
4.7 Variations in AE2ETD with m for $n = 400,\ 625,$ and 900. 70
4.8 Impact of proposed framework on the PDR performance of (a) RMAC and (b) PRMAC. 74
4.9 Impact of our proposed framework on E2ETD performance of (a) RMAC and (b) PRMAC. 75
4.10 Impact of our proposed framework on AEC of (a) RMAC and (b) PRMAC. 76

4.11 Impact of our proposed framework on PDR performance of CL-MAC and LDC-MAC. 77

4.12 Impact of our proposed framework on E2ETD performance of CL-MAC and LDC-MAC. 78

4.13 Impact of our proposed framework on AEC of CL-MAC and LDC-MAC. 78

5.1 Proposed cycle structure. 84

5.2 Structures of (a) SW (b) DW and (c) DCW. 85

5.3 Cycle structure followed by the nodes in Setup Phase. 87

5.4 Overlapping TSs of different *grade* nodes in DTP. 91

5.5 In an *idle cycle*, (a) sleep and wakeup time instants of a sensor node having $G = j$ and $DSI = i$, and (b) sleep and wakeup time instants of a sink node. 92

5.6 Illustration of JRAM's data packet transmission process. 94

5.7 Representation of a sub-network considering that each network node has identical CR (R) in all the direction and network area doesn't have any obstacles. 96

5.8 Depiction of various parameters used in α' determination. 97

5.9 Variations in α' with m. 97

5.10 Variation in (a) PDR and (b) AE2ETD, with m. 101

5.11 Comparison of E2ETD in case of (a) $M = 2$, (b) $M = 3$, and (c) $M = 4$. 103

5.12 Comparison of PDR in case of (a) $M = 2$, (b) $M = 3$, and (c) $M = 4$. 104

5.13 Comparisons of AEC in case of (a) $M = 1$, (b) $M = 2$, and (c) $M = 3$. 105

A.1 Possible geometries of $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ corresponding to different values of δ. 115

List of Tables

3.1 Networking Parameters 46

3.2 Frame Sizes 46

4.1 Cycle Duration Related Parameters 72

4.2 Networking Parameters 72

4.3 Frame Sizes 72

5.1 Networking Parameters 99

5.2 Frame Sizes 100

5.3 Cycle Duration Related Parameters 100

Glossary

WSN	Wireless Sensor Network
ADC	Analog-to-Digital Converter
QoS	Quality of Service
E2ETD	End-to-End Transmission Delay
EOR	Event Occurrence Rate
MAC	Medium Access Control
OSI	Open Systems Interconnections
SW	Synchronization Window
DW	Data Window
SlpW	Sleep Window
DTS	Data Transmission Segment
DRS	Data Reception Segment
CSR	Carrier Sensing Range
CR	Communication Range
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
AEC	Average Energy Consumption
TDMA	Time Division Multiple Access
DIFS	Distributed Inter frame Space
SIFS	Short Inter frame Space
CCA	Clear Channel Access
CW	Contention Window
NAV	Network Allocation Vector
TxW	Transmit Window
RxW	Receive Window

RTR	Ready-to-Receive
ACK	Acknowledgement
EACK	Early Acknowledgement
GDSA	Grade Division and Schedule Assignment
RTS	Request-to-Send
CTS	Clear-to-Send
FRTS	Future Request-to-Send
PION	Pioneer
FSP	Flow Scheduling Packet
RTSD	Request-to-Send Data
CTSD	Clear-to-Send Data
PS	Primary Sender
SS	Secondary Sender
PAI	Packet Arrival Interval
PDR	Packet Delivery Ratio
DS	Disjoint Set
DSI	Disjoint Set Index
LT	Lookup Table
TS	Time Segment
DTP	Data Transmission Phase
RTFD	Request-to-Forward Data
CTFD	Clear-to-Forward Data

List of Publications

Journal Publications

1. Ripudaman Singh, Brijesh K. Rai and Sanjay K. Bose, "A low delay cross-layer MAC protocol for k-covered event driven wireless sensor networks," *IEEE Sens. Lett.*, vol. 1, no. 6, pp. 1-4, Dec. 2017.

2. Ripudaman Singh, Brijesh K. Rai and Sanjay K. Bose, "A novel framework to enhance the performance of contention-based synchronous MAC protocols," *IEEE Sens. J.*, vol. 16, no. 16, pp. 6447-6457, Aug. 2016.

3. Ripudaman Singh, Brijesh K. Rai and Sanjay K. Bose, "A joint routing and MAC protocol for transmission delay reduction in many-to-one communication paradigm for wireless sensor networks," *IEEE IoT J.*, vol. 4, no. 4, pp. 1031-1045, Aug. 2017.

International Conferences

1. Ripudaman Singh, Brijesh K. Rai and Sanjay K. Bose, "A low delay cross-layer contention based synchronous MAC protocol for a multi-hop WSN," *IEEE TENCON*, 2016, pp. 1821-1824.

2. Ripudaman Singh, Brijesh K. Rai and Sanjay K. Bose, "A contention based routing enhanced MAC protocol for transmission delay reduction in a multi-hop WSN," *IEEE TENCON*, 2017, pp. 398-402.

1

Introduction

Contents

1.1 Issues Related to Duty-Cycle MAC Protocols 3
1.2 Motivation of the Present Work 6
1.3 Thesis Contribution 8
1.4 Thesis Organization 9

A wireless sensor network (WSN) comprises a large number of small sized and low cost sensor nodes. A sensor node is typically made up of four basic units: the sensing unit, the processing unit, the communication unit, and the power unit. Sensing unit consist of two subunits, sensor and analog-to-digital converter (ADC). Based on the observed phenomenon, a sensor produces an analog signal, which is converted into the digital signal by the ADC and then sent to the processing unit. The processing unit is usually connected with a storage unit and it manages the procedures that enables a sensor node to collaborate with the other deployed nodes to execute the given sensing tasks. A sensor node communicates with the neighboring nodes through its communication unit. The sensing modules used in the sensing unit of a sensor node depend on the monitored events.

The applications of WSNs can be divided into two groups, data gathering applications and event-driven applications. In data gathering applications, such as monitoring of air quality, temperature and humidity, sensor nodes sense the environment and periodically send the sensed information to the sink node [1–4]. On the other hand, in event-driven applications, such as fire detection, target tracking, and intrusion detection, sensor nodes remain in *idle state* most of the time. However, when an event of interest occurs, multiple sensor nodes lying in its neighborhood may generate data packets to report the event to the sink node [1–5]. Monitoring of hazards, such as fire, intrusion and tsunami, in hostile environment is an important category of even-driven WSNs applications. In such applications, for better quality of service (QoS), a detected event should be reported to the sink node as quickly as possible [6, 7]. Note that the collected sensing data would be meaningless if these are not received by the sink node in time and the event may become even more serious. In event-driven WSNs, accurate information of the location of the event would also help in initiating prompt action [1]. This thesis is mainly focused on delay-sensitive event monitoring applications of WSNs in unsafe regions, such as fire and intrusion monitoring in the dense forest.

In the delay-sensitive event-driven WSNs, sensor nodes are densely deployed due to the following reasons. In the first case, using multi-sensor data fusion techniques, it is unlikely that a false event would be detected as the true event if three or more sensor nodes participate in the decision [8]. Secondly, the design of triangulation based positioning systems require that each point in the monitored region be covered by at least three sensor nodes for high position accuracy [9]. Thirdly, to cope with the sensor node failures (because of their potential fragility), the design of WSNs for hazard monitoring should be as reliable as possible since failed nodes cannot be easily diagnosed and replaced due to the

large size and/or unsafe nature of the monitored region.

The small size of a sensor node not only limits its battery capacity but also provides difficulties in recharging or replacing the battery [10]. In addition, dense deployment in large sized regions and the possibly hostile nature of the monitored region will make it difficult to replace nodes that do not have sufficient energy. Therefore, along with minimum end-to-end transmission delays (E2ETD), efficient utilization of the limited battery energy of a sensor node is also a critical requirement for long network life and uninterrupted network service in delay-sensitive event monitoring applications of WSNs in unsafe regions. In a WSN, energy may be wasted because of several reasons. A major reason for energy wastage is *collision*. When a node receives two or more packets simultaneously, these packets are termed as have *collided*, even when the packets overlap only partly. The receiver node discards all the corrupt packets and retransmissions of these packets are required, which increases energy consumption. The second reason is *overhearing*. This occurs when a node receives a packet that is not destined for it. The third cause is *control packet overhead*. Transmission and reception of a control packet require considerable amount of energy. Another reason for energy wastage is *idle listening*. This occurs when a node listens to an idle channel in order to receive possible traffic even though the channel is idle. For most of the sensor nodes, the power consumed in *idle listening* is comparable to that consumed in transmitting or receiving packets [11, 12]. For example, the idle: send: receive power ratios for MicaZ motes are 1.13:1:1.13 [13] while these ratios for Tmote Sky motes are 1.11:1:1.11 [14].

In a WSN, to reduce the energy-consumption in *idle listening*, each sensor node repeatedly switches its radio (communication unit) *off*/*on*. The ratio of the time spent by the radio of a sensor node in *on* and *off* state in a cycle is termed as duty-cycle of the sensor node. In *off* state (also known as *sleep state* of the sensor node), a radio can not transmit/receive any packet. In the hazards monitoring application of WSNs, due to the low event occurrence rate (EOR), sensor nodes prefer to operate with a low duty-cycle.

1.1 Issues Related to Duty-Cycle MAC Protocols

Medium access control (MAC) layer is a part of the link layer in the OSI (open systems interconnections) layer model. Main function of MAC is to coordinate access to and transmission over a medium which is common to several nodes. Radio of a sensor node is controlled by the MAC protocol

and it is one of the most energy-consuming units of sensor nodes [15]. Therefore, an efficient MAC protocol can provide low E2ETD with low energy consumption in delay-sensitive event monitoring applications of WSNs in unsafe regions. MAC protocols proposed for duty-cycled WSNs can be divided into three groups, contention-free [16–24], contention based [25–64], and hybrid [65–71]. In the contention-free MAC protocols, a node does not need to win the contention for medium access before transmitting its data packets. On the other hand, in a contention based MAC protocol, a node need to win the contention for medium access before transmitting its data packets. In hybrid MAC protocols, a node switches its medium access rule from contention based/contention-free to contention-free/contention based according to the current traffic load. The contention based MAC protocols are further divided into two kinds based on the sleep/wakeup schedules of the network nodes, i.e. asynchronous [25–36] and synchronous [37–64]. The following two characteristics of contention based synchronous MAC protocols make them the very suitable for delay-sensitive event monitoring applications in dense multi-hop WSNs. First, they are robust to dynamic topology changes due to the new node insertion and node and link failures. Second, in such protocols, a data packet can travel multiple hops in one cycle. For ease in understanding, in this thesis, we divide the existing contention based synchronous MAC protocols into two groups, *generic* [37–55] and *staggered* [56–64].

In the cross-layer *generic* contention based synchronous MAC protocols [42–55], all the neighbors of a node follow the same sleep/wakeup schedule. Therefore, a node can send and/or receive data packets to and/or from each of its one-hop neighbor node. As a result of this, a sensor node can report the detected event to any sink node if there exists a path in between the sensor node and each sink node. These protocols are well suited for WSNs deployed for monitoring multiple events in the same geographic region with the constraint that the two or more events must be reported to different sink nodes [55]. As can be seen in Fig. 1.1, the cycle structure followed by the network nodes in *generic* contention based synchronous MAC protocols contains three windows, a synchronization window (SW), a data window (DW), and a sleep window (SlpW). (In Fig. 1.1, the duration of one cycle is denoted by T_{CYCLE}.) In its SW, each node periodically executes a synchronization protocol [72–80]. The nodes schedule data packet transmission in their DW. In SlpW, generally, a node remains in *sleep state* to reduce *idle listening*. Therefore, in case of low-duty cycle, this cycle structure contains a small DW and a large SlpW. In the cross-layer *generic* contention based synchronous MAC protocols proposed in [42–55], the cross-layer routing information enables a node to schedule

multi-hop forwarding of its data packets in DW. The concerned nodes wakeup to transmit/receive data packets in the subsequent SlpW during their data transmission segment (DTS)/data reception segment (DRS). A node determines its DTS/DRS by employing a one-to-one *proportional mapping function* between the time interval during a DW and during the subsequent SlpW. Therefore, in these protocols, more than one nodes can forward their data packets when multiple sensor nodes with data packets to send lie in the carrier sensing range (CSR) of each other. Further, in [51–55], the duration of a DTS/DRS is γ times the transmission duration of the control packet used for sending the data transmission request in the DW where γ denotes the ratios of the SlpW and DW durations. That is, the duration of a node's DTS/DRS increases with a decrease in the duty-cycle. In hazard monitoring applications, such as fire, the size of a data packet is almost equal to the control packet used for sending the data transmission request in the DW because only few bytes are sufficient to report the detected information. In such a scenario, a sensor node can transmit/receive multiple data packets in its DTS/DRS. Moreover, in event-driven WSNs, a sensor node employs data compression [81–83] and/or data aggregation techniques [84, 85] to optimize the number and size of the data packets to be sent in a cycle. Therefore, generally, a node utilizes only a small portion of its DTS/DRS for transmitting/receiving data packets. However, in the existing cross-layer *generic* contention based synchronous MAC protocols, a node is not allowed to utilize the unused portion of its one-hop neighbor's DRS for forwarding its data packets if the node failed in scheduling and one of its one-hop neighbors succeed in scheduling in the same DW. In addition to this, these protocols do not assign DRS to the source node of any flow scheduled in a DW.

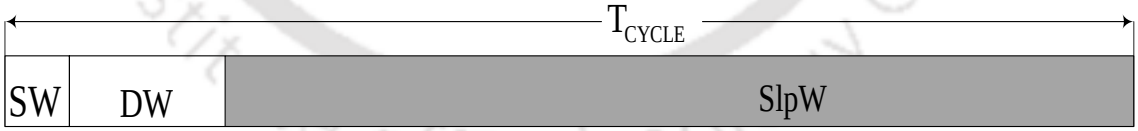


Figure 1.1: Cycle structure followed by the network nodes in *generic* contention based synchronous MAC protocols.

Staggered contention based synchronous MAC protocols [56–64] are proposed for many-to-one communication scenarios. Therefore, unlike *generic* contention based synchronous MAC protocols, in these protocols, a node can report its detected information only to the sink node. These protocols assign a *grade* to each network node. The *grade* of a node represents the minimum number of hops a data packet need to travel from the node to reach one of the sink nodes. Therefore, unlike *generic*

protocols, they do not require a separate routing protocol for assigning the next-hop forwarders to the sensor nodes. To alleviate the effect of periodic sleep on the E2ETD, these protocols utilize the *grade* information to assign *staggered* wakeup schedules to adjacent *grade* nodes. Because of this, a data packet can then travel in a pipelined fashion from source to sink. These protocols are much more energy-efficient than the *generic* contention based synchronous MAC protocols. However, in the existing *staggered* contention based synchronous MAC protocols, at most one node can forward its data packets in a cycle when multiple sensor nodes with data packets to send lie in the CSR of each other.

1.2 Motivation of the Present Work

Some geographic regions, such as disputed border areas or dense forests, are not safe for humans to monitor continuously. Therefore, the continuous monitoring of hazards, such as fire and intrusion, in these regions is an important class of WSN applications. For better QoS, in hazard monitoring applications in unsafe regions, sensor nodes should be deployed such that each point of the monitored region lies in the sensing range of three or more sensor nodes. In event-driven WSNs, for better reliability, multi-sensor data fusion techniques (e.g., [86–89]) are employed for decision making. In these techniques, it is unlikely that a false event would be detected as the true event if three or more sensor nodes participate in making the decision [8]. Due to the delay-sensitive nature of hazardous events, a detected event should be reported to the sink node as early as possible. Therefore, for better QoS, at least three sensor node should report the detected event to the sink node quickly. A WSN is termed as *k-covered* if each point of the monitored region lies in the sensing range of k or more sensor nodes [90, 91]. In a *k-covered* WSN, whenever an event of interest occurs, M $(\geq k)$ sensor nodes lying in the neighborhood of the event's location generate data packets to report the event to the sink node. Typically, a sensor node has a smaller sensing range than its communication range (CR). Therefore, two or more sensor nodes that detect the event may lie in the CR of each other. Furthermore, generally, the CSR of a sensor node is twice or more than its CR [47, 92]. Therefore, most of the sensor nodes that detect the event may lie in the CSR of each other.

Generally, in hazard monitoring applications of WSNs, EOR remains very low. Because of this, the sensor nodes in such applications prefer to operate with a low duty-cycle in order to enhance long network life. In order to provide the low E2ETD in an event-driven low duty-cycled dense multi-hop

WSN, a duty-cycled contention based synchronous MAC protocol needs to have following features. It should allow three or more sensor nodes to forward their data packets in the same cycle when multiple sensor nodes with data packets to send lie in the CSR of each other and it should allow a node to forward its data packets over a large number of hops in each cycle. As mentioned in above section, in low duty-cycled WSNs, the cycle structure followed by the sensor nodes in the cross-layer *generic* contention based synchronous MAC protocols contains a small size DW. In addition to this, in these protocols, CSMA/CA mechanism is adopted in order to resolve the contention for channel access. In this mechanism, a node can try to schedule data packet transmission only when no other node in its CSR is transmitting. Therefore, in a small size DW, very few nodes can schedule the forwarding of their data packets when multiple sensor nodes with data packets to send lie in the CSR of each other and a node cannot schedule a longer flow.

We noted the following three shortcomings/problems in existing contention based synchronous MAC protocols that are tackled in the schemes proposed by us.

(i) Existing cross-layer *generic* contention based synchronous MAC protocols do not allow a node (say, A) to utilize the flow scheduled by its one-hop neighbor (say, B), to forward its data packets even when the both A and B have data packets to send to the same final destination and node B has fewer data packets than what a node can send in its DTS (or a node can receive in its DRS). This shortcoming limits the number of nodes that can forward their data packets in the same cycle when multiple sensor nodes with data packets to send lie in the CR of each other. As a result of this, the E2ETD in reporting a detected event to the sink node increases.

(ii) As shown in Fig. 1.1, the cycle structure followed by the network nodes in the existing cross-layer *generic* contention based synchronous MAC protocols contains one SW, and only one pair of DW and SlpW. Therefore, if a node failed in scheduling in a DW, it waits for almost one entire cycle duration for the next chance to transmit. In addition to this, in a small size DW, a node can schedule forwarding of its data packets over only few hops. These shortcomings limit the number of nodes that can forward their data packets in the same cycle when multiple sensor nodes with data packets to send lie in the CSR of each other, and the number of hops that a data packet can travel in a cycle. As a result, the delay in reporting a detected event to the sink node increases.

(iii) In the existing *staggered* contention based synchronous MAC protocols, a node gets only one

chance to succeed in data transmission scheduling in a cycle. Therefore, in such protocols, at most one sensor node can forward its data packets in a cycle when multiple sensor nodes with data packets to send lie in the CSR of each other. Because of this, these protocols encounter longer delay in reporting a detected event to the sink node in dense multi-hop WSNs.

1.3 Thesis Contribution

This thesis aims at design of new low delay and low energy duty-cycled contention-based synchronous MAC protocols for event monitoring in the dense multi-hop WSNs. The proposed protocols improve E2ETD by providing more number of chances to a node to schedule the forwarding of its data packets in a cycle and/or allowing a node to schedule multi-hop forwarding of its data packets multiple times in a cycle. As in the existing duty-cycle contention based synchronous MAC protocols [42–55], we assume in this thesis that a separate protocol (e.g., [72–80]) is used to synchronize the clocks in the sensor nodes with required precision during the synchronization window (SW). The main contributions of this thesis are summarized next.

(a) A Low Delay Cross-Layer MAC (LDC-MAC) Protocol:
We propose LDC-MAC, a low delay cross-layer MAC protocol for *k-covered* event-driven WSNs. LDC-MAC is a cross-layer *generic* contention based synchronous MAC protocol. It has the following three novel features. First, it assigns an interference-free DRS to each node of every flow scheduled in a DW. Second, it allows a node to utilize the unused portion of its one-hop neighbors' DRS to forward its data packets if it failed to win the contention for medium access in the DW. Third, it allows multiple nodes to forward their data packets through the one scheduled flow. Results show that LDC-MAC provides significantly low E2ETD in an event-driven multi-hop WSN, at the cost of a very minor increase in average energy consumption (AEC).

(b) A Framework for *Generic* Contention Based Synchronous MAC Protocols:
We propose a novel framework to improve the E2ETD performances of the existing *generic* contention based synchronous MAC protocols in event-driven WSNs. This framework maps the m (> 1) cycle data transmission process of the implemented *generic* contention based synchronous MAC protocol in a single duration cycle. In this framework, dense deployment of sensor nodes is exploited to reduce *idle*

listening. An analytical technique is proposed to determine the m for the optimum E2ETD in a given network scenario. Results show that proposed framework significantly improves E2ETD performances of the existing cross-layer *generic* contention based synchronous MAC protocols, in event-driven dense multi-hop WSNs, at the cost of small increase in AEC.

(c) A Joint Routing and MAC (JRAM) Protocol:

We propose JRAM, a joint routing and MAC protocol for transmission delay reduction in the many-to-one communication scenarios for event-driven WSNs. JRAM is a *staggered* contention based synchronous MAC protocol. JRAM proposes a new cycle structure that provides multiple chances to each node to succeed in data packet transmission scheduling in every cycle. Low complexity distributed algorithms are proposed to reduce the energy-consumption in transmission, reception, and *idle listening*. These protocols also overcome the need of implementing a separate routing protocol by assigning the next-hop forwarders to the nodes. Results show that in an event-driven dense multi-hop WSN, JRAM provides significantly lower E2ETD than other existing *staggered* contention based synchronous MAC protocols.

1.4 Thesis Organization

This thesis is organized as follows. A structural organization of this thesis is shown in Fig. 1.2.

Chapter 1 (the current chapter) states the problems studied in this thesis and the motivation behind these problems. This chapter highlights the salient features of contention based synchronous MAC protocols and hazard monitoring applications of WSNs. This chapter also summarizes the thesis contributions and provides a brief outline of the thesis organization.

Chapter 2 reviews the current state of the art literature for work done in the areas relevant to this thesis and associated topics. Here, we discuss the different type of MAC protocols proposed for WSNs. Contention-free, contention based asynchronous, and hybrid MAC protocols are discussed in brief and the reasons why they are not particularly suitable for delay-sensitive event monitoring applications in dense multi-hop WSNs are mentioned. Contention based synchronous MAC protocols are discussed in detail. Shortcomings of recently proposed *generic* and *staggered* contention based synchronous MAC protocols are highlighted in the context of delay-sensitive event monitoring applications in dense multi-hop WSNs. We also discuss in brief the synchronization protocols proposed for WSNs and their

suitability for contention based synchronous MAC protocols.

Chapter 3 presents LDC-MAC, a low delay cross-layer MAC protocol for *k-covered* event-driven WSNs. In this, we briefly discuss the reasons behind the large size of a DTS/DRS along with the issues pertaining to its under-utilization. Followed by this, we describe the process adopted for interference-free DRS assignment to the nodes of a scheduled flow. With the help of an illustration, we then describe how the LDC-MAC protocol allows multiple nodes to forward their data packets through the same scheduled flow.

In **Chapter 4**, we present a novel framework to enhance the performance of existing *generic* contention based synchronous MAC protocols in event-driven WSNs. First, we discuss the technique adopted for partitioning the sensor nodes into disjoint sets (DSs) for exploiting the dense deployment of sensor nodes to reduce *idle listening*. Our proposed cycle structure is presented next. With the help of an illustration, we describe how our propose framework maps m cycles of the data transmission process of an implemented *generic* contention based synchronous MAC protocol in one cycle. Thereafter, we discuss the analytical technique proposed for determining the optimum value of m in a given network.

Chapter 5 presents JRAM, a joint routing and MAC protocol for reducing transmission delay in the many-to-one communication scenarios for event-driven WSNs. First, we discuss the proposed cycle structure. We then describe the technique proposed to reduce the energy-consumption in transmission, reception, and *idle listening*. With the help of an illustration, we describe how JRAM provides more than one chances to a node to succeed in data packets transmission scheduling in a cycle.

Chapter 6 concludes this thesis with a summary of the work done and includes some suggestions that may be investigated in future research.

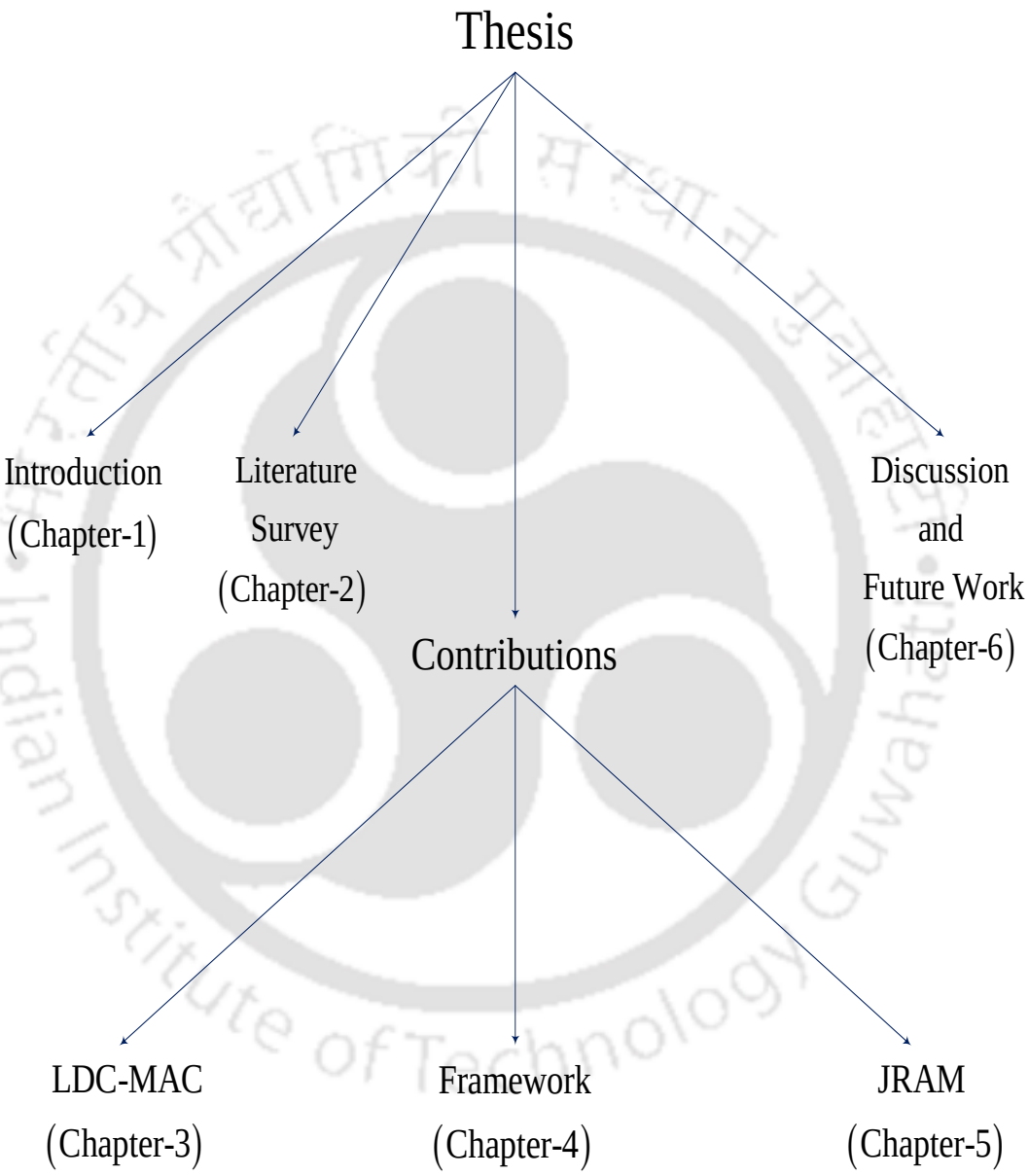


Figure 1.2: Thesis organization.

2

Literature Survey: Review of Related Work

Contents

2.1 MAC Protocols for WSNs 14
2.2 Contention Based Synchronous MAC Protocols 19
2.3 Synchronization Protocols for WSNs 34

In this chapter we present different categories of MAC protocols proposed for WSNs and review the literature under each category. The appropriateness of each category for delay-sensitive event monitoring applications in dense multi-hop WSNs is also discussed. Duty-cycled contention based synchronous MAC category is identified as the most suitable choice for delay-sensitive event monitoring applications in a dense multi-hop WSN. Basic and recently proposed duty-cycled contention based synchronous MAC protocols are discussed in detail. The factors that affects the E2ETD performance of a contention based synchronous MAC in an event-driven dense multi-hop WSN are also discussed. We also discuss in brief the synchronization protocols proposed for synchronizing the sleep and wakeup time instants of neighboring sensor nodes in a duty-cycled multi-hop WSN.

2.1 MAC Protocols for WSNs

In a WSN, multiple sensor nodes share a communication medium for transmitting their data packets. The wireless channel is broadcasting in nature; therefore, the transmission of a sensor node interferes with the packet reception at the other nodes within its interference range. In such scenarios, the MAC layer is primarily responsible for coordinating access to and transmission over a medium common to several nodes such that the packet loss due to interference is minimum. In addition to this, the radio is also controlled by the MAC protocol that is one of the most energy-consuming units of a sensor node [15]. Therefore, an efficient MAC protocol can provide low E2ETD in a dense multi-hop WSN with low energy-consumption. As can be seen in Fig. 2.1, the existing MAC protocols can be classified mainly into three groups, contention-free [16–24], contention based [25–64], and hybrid [65–71].

2.1.1 Contention-Free MAC Protocols

In contention-free MAC protocols [16–24], a fraction of the available medium resource (e.g., a time/frequency slot from a time frame/frequency band) is assigned to a node such that the node can transmit without collision utilizing its assigned resource. In such protocols, for medium resource assignment to a node, the knowledge of network topology and tight synchronization are the two basic requirements [15]. Time Division Multiple Access (TDMA) is a representative example of contention-free MAC protocols. In TDMA based MAC protocols [16–24], time is divided into frames and each frame is subdivided into slots. In each frame, a collision-free slot is assigned to every sensor node to transmit its data packets. The basic principal of TDMA is illustrated in Fig. 2.2, assuming that

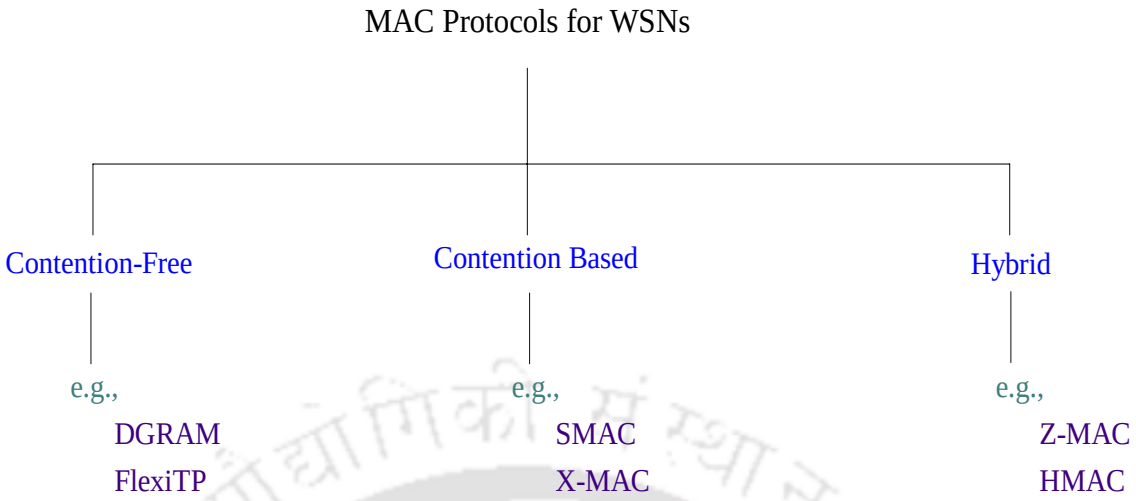


Figure 2.1: Classification of MAC protocols proposed for WSNs.

(1) the network contains three nodes A, B, and C; (2) all the three nodes are in the communication range (CR) of each other; and (3) each node can transmit its data packets directly to the sink node (S). In Fig. 2.2, a frame contains three slots, Slot 1, Slot 2, and Slot 3. Slot 1, Slot 2, and Slot 3 are allocated to the Node A, Node C, and Node B, respectively. Each node can transmit its data packet in the assigned slot without suffering from collision. In this way, TDMA based MAC protocols ensure collision free data packet transmissions.

The existing work on TDMA protocols is mainly focused on the algorithms of collision-free slot assignment to the network nodes and techniques of efficient utilization of assigned time slots. In the TDMA protocols proposed for WSNs [16–24], duty-cycling is adopted for reducing the energy-consumption in *idle listening*. In a duty-cycled TDMA protocol, a sensor node remains awake during its each data reception slot while, in its data transmission slot, a sensor node wakes up only if it has data packets to send. In all other slots of each frame, a sensor node remains in *sleep state*. In addition to this, there is no energy consumption in *overhearing* and transmissions do not suffer from collisions. Therefore, the main and most important advantage of TDMA over CSMA is low power consumption. Further, in TDMA, a node can transmit in its assigned slot without collision. As a results of this, TDMA protocols provide low and predictable E2ETD, and high PDR in high EOR scenarios. These features of TDMA protocols make them very suitable for periodic data gathering applications of WSNs where sensor nodes sense their environment at fixed time intervals and forward the sensed data to the sink node.

However, the TDMA protocols are less suitable than CSMA for low EOR delay sensitive event monitoring applications in dense multi-hop WSNs due to the following limitations. TDMA protocols do not adopt well to network topology changes which are quite frequent in sensor networks due to insertion of new nodes, and links and nodes failure. The nodes require highly accurate synchronization. Moreover, large E2ETD is encountered in reporting the detected event to the multi-hop sink node due to low bandwidth utilization.

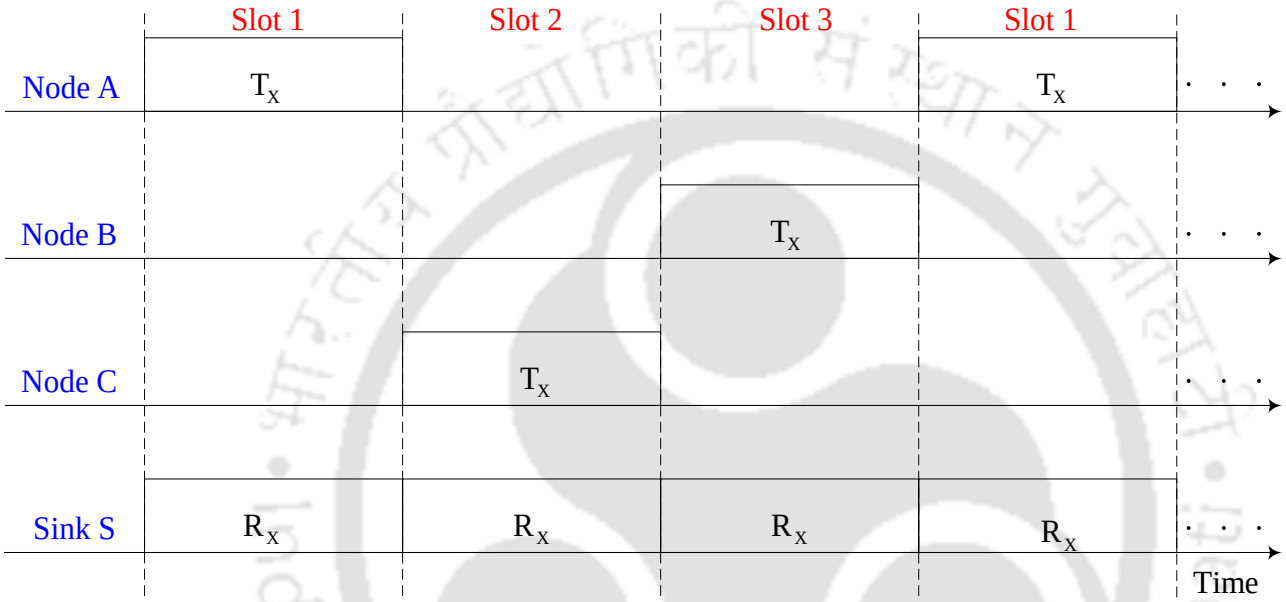


Figure 2.2: Operation of a TDMA based MAC protocol.

2.1.2 Contention Based MAC Protocols

In contention based MAC protocols, a sensor node need to win the contention for channel access before sending its data packets. Generally, the carrier sense multiple access with collision avoidance (CSMA/CA) mechanism is adopted in order to resolve the contention. The basic principal of CSMA/CA is illustrated in the Fig. 2.3. When a node intends to transmit a data packet, it performs carrier sensing to check the current status of the channel. In case, the channel is idle and it remains idle for distributed inter frame space (DIFS), then the node sets a random back-off timer and transmit the request-to-send (RTS) packet if the medium remains idle till the expiration of the timer. The node starts transmitting the data packet only after receiving the clear-to-send (CTS) packet from the intended receiver, as a response to the send RTS packet. The CSMA/CA mechanism is easy in implementation as it does not require the knowledge of the network topology. For example, in Fig.

2.3, the nodes s_1, s_2, and s_3 trying to send their data packets at the same time instant and they lie in the carrier sensing range (CSR) of each other. After observing the medium idle for DIFS duration, each node sets its back-off timer. As shown in Fig. 2.3, the node s_1 wins the contention for channel access as its timer expires before the other two contenders. Note that the collision occurs when the back-off timer of two or more contenders expire at the same time instant.

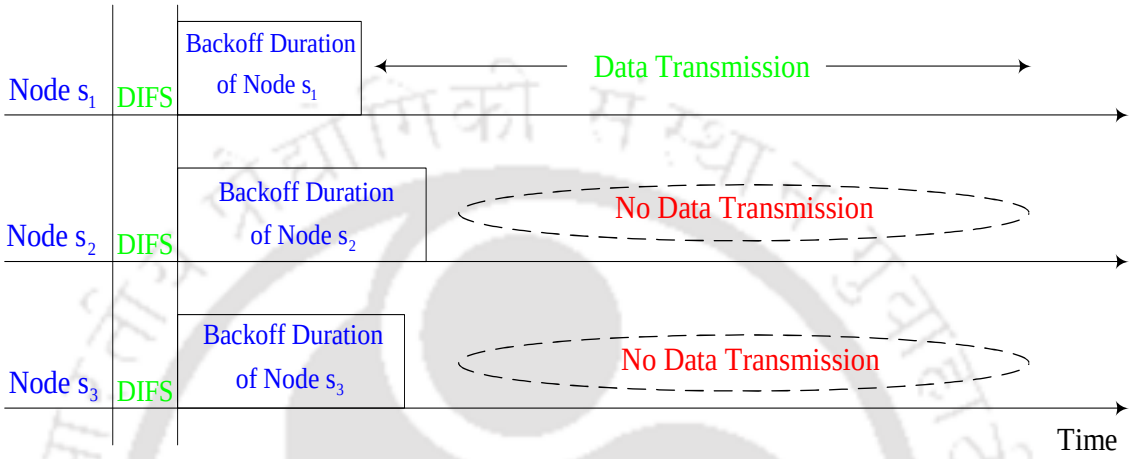


Figure 2.3: Operation of a contention based MAC protocol with competition.

In the contention based MAC protocol proposed for WSNs, duty-cycling technique is adopted for the long network life. In this technique, each sensor node periodically goes in *sleep state* to reduce the energy-consumption in *idle listening*. As can be seen in Fig. 2.4, the existing duty-cycled contention based MAC protocols can be divided into two groups on the basis of the sleep/wakeup schedule followed by the sensor nodes, named as, asynchronous [25–36] and synchronous [37–64].

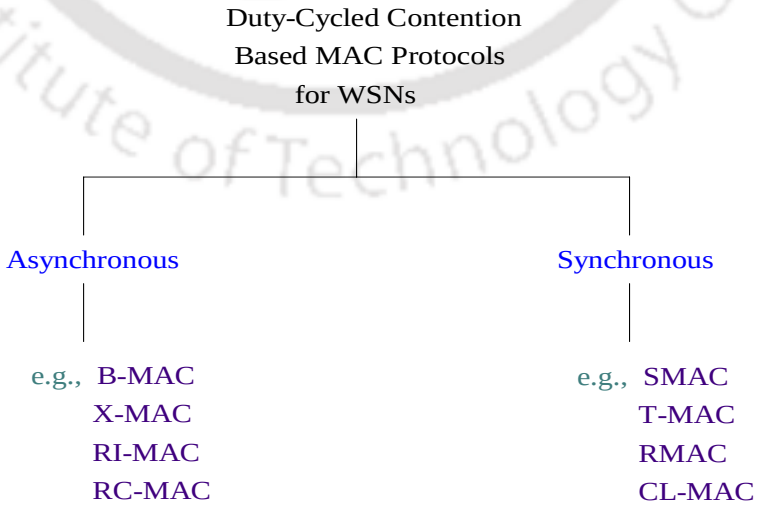


Figure 2.4: Classification of duty-cycled contention based MAC protocols proposed for WSNs.

2.1.2.1 Contention Based Asynchronous MAC Protocols

In asynchronous MAC protocols [25–36], each deployed sensor node autonomously chooses its sleep/wakeup schedule. These protocols do not employ a separate synchronization protocol to synchronize the sleep/wakeup schedule of neighboring sensor nodes. Therefore, in such protocols, a node is not aware about the time-offset in between its own sleep and wakeup time instants and its one-hop neighbors' sleep and wakeup time instants. However, all the network nodes follow the cycle structure which contains a small size wakeup window and a large size sleep window (SlpW). Moreover, the durations of wakeup window, SlpW and cycle (T_{CYCLE}) are same in the cycle structure followed by each sensor node. The AEC and E2ETD performance of an asynchronous MAC protocol mainly depends on the technique used for establishing the communication between the two nodes. The existing research on asynchronous MAC protocols is mainly focused on the techniques of minimizing the energy consumption and delay in establishing the communication between a sender and its intended next-hop receiver.

The research starts with using a long preamble to establish communication between the two nodes. Berkeley MAC (BMAC) [25] improved this preamble sampling method by proposing a better outlier detection based clear channel access (CCA) algorithm. After this, four approaches have been proposed for reduction in energy-consumption and/or latency in establishing the communication between two nodes. First approach is to include the useful information (e.g., next-hop receiver address) in the preamble [26–30]. The main objective of this method is to reduce the energy-consumption in *overhearing*. The other three methods include schedule learning [31], receiver-initiated low power probing (LPP) [32–35], and predictive-wakeup [36]. The main objective of these three methods is to reduce the latency in establishing the communication between two nodes.

In comparison to TDMA protocols, duty-cycled contention based asynchronous MAC protocols are easy to implement as they do not require information on the network topology and a separate synchronization protocol. However, due to lack of synchronization in sleep/wakeup schedules of neighboring nodes, in such protocols, a sensor node cannot schedule multi-hop forwarding of its data packets after winning the contention for channel access. Therefore, these protocols provide larger E2ETD in reporting a detected event to the multi-hop distance sink which makes such protocols less suitable for delay-sensitive event monitoring in a dense multi-hop WSN.

2.1.3 Hybrid MAC Protocols

In several WSN applications, network traffic load varies over a wide range with the time. The contention-free MAC protocols [16–24] are suitable mainly under high network traffic load because the transmission of a node in its assigned slot do not suffer from collision. Under low traffic load conditions, they provide larger E2ETD due to the low bandwidth utilization. On the other hand, contention based MAC protocols [25–64] are suitable mainly under low traffic load due to the low collision probability and efficient bandwidth utilization. The E2ETD performance of a contention based MAC protocol degrades with an increase in the network traffic load because an increase in the network traffic load increases the packet collision probability and reduces the bandwidth utilization.

Unlike the purely contention based and contention-free MAC protocols, in hybrid MAC protocols [65–71], a node switches its medium access rule from contention based/contention-free to contention-free/contention based according to the current traffic load. Basically, a hybrid MAC protocol is a contention-free MAC protocol. To avoid the increase in E2ETD due to the low bandwidth utilization under low traffic load, a hybrid MAC protocol switches its MAC technique from contention-free to contention based. The adaptability of the hybrid MAC protocols toward the variation in network traffic load make these protocols suitable for the WSN applications where network traffic load varies with the time. Most of the hybrid MAC protocols proposed for WSNs [65–71] are the combination of TDMA and CSMA/CA techniques.

A hybrid MAC protocol inherits the shortcomings of contention-free techniques such as, knowledge of network topology and requirement of highly accurate synchronization for contention-free slot assignment. Therefore, these techniques are not suitable for delay sensitive event monitoring in a dense multi-hop WSN where network topology changes frequently due to the nodes failure, insertion of new nodes and links failure.

2.2 Contention Based Synchronous MAC Protocols

In the duty-cycling contention based synchronous MAC protocols [37–64], each node contains information about the time-offset in between its own sleep and wakeup time instants and each of its one-hop neighbors' sleep and wakeup time instants. In such protocols, each node periodically follows a synchronization protocol to maintain the fixed time-offset (which is a user defined parameter) in between its own sleep and wakeup time instants and each of its one-hop neighbors' sleep and wake

time instants. (The synchronization protocols proposed for WSNs are discussed in brief in Section 2.3.) For ease in understanding, in this thesis, we divide the existing contention based synchronous MAC protocols into two groups, *generic* [37–55] and *staggered* [56–64].

2.2.1 *Generic* Contention Based Synchronous MAC Protocols

In the *generic* contention based synchronous MAC protocols, a node can report the detected event to the any sink node if there exists a path in between the sensor node and each sink node. These protocols are well suited for WSNs deployed for monitoring multiple events in the same geographic region with the constraint that the two or more events must be reported to different sink nodes [55]. In this section, we discuss the basic and recently proposed *generic* contention based synchronous MAC protocols in detail.

Sensor MAC (SMAC) [37]: SMAC is the first *generic* contention based synchronous MAC protocol proposed for WSNs. In SMAC, each network node follows the cycle structure shown in Fig. 1.1 where cycle duration (T_{CYCLE}) is divided into three windows, named as, synchronization window (SW), data window (DW) and sleep window (SlpW). A node wakes up at the beginning of its SW and goes to sleep at the beginning of its SlpW if it is not transmitting/receiving any packet. In its SW, each network node periodically follows a synchronization protocol so that it can synchronize its sleep and wakeup time instants with the sleep and wakeup time instants of its one-hop neighbors. A node with data packets to send sets a timer for DIFS + $rand(CW_D).T_{\text{slot}}$ duration at the beginning of its DW and senses the medium. Here, $rand\,(CW_D)$ denotes the randomly chosen slot of the contention window used in DW (CW_D) and T_{slot} denotes the duration of one slot of CW_D. In case, the node did not sense any ongoing communication till the expiration of its timer, it sends request-to-send (RTS) packet to its next-hop receiver and wait for the response. If the intended receiver node is ready for receiving the data packets, it replies with the clear-to-send (CTS) packet. After receiving the CTS from the intended receiver node, the sender sends it data packets. RTS and CTS, each packet contains a duration field, which indicates how long the remaining transmission will be. When a node receives RTS/CTS packet destined to the other node, it records this value in a variable termed as network allocation vector (NAV) and sets a timer for it. At NAV timer expiration, the node resets its NAV to zero. When a node has data packet to send, it first checks its NAV. A non-zero value of NAV indicates that the medium is busy. The data transmission process of SMAC is illustrated in Fig. 2.5 assuming that (1) the network contains two sensor nodes (s_1 and s_2) and one sink node (S); (2) each sensor

node has one data packets to send to S in the same cycle; (3) all the three nodes (s_1, s_2, and S) are in the CSR of each other; and (4) sensor node s_2 won the contention for channel access.

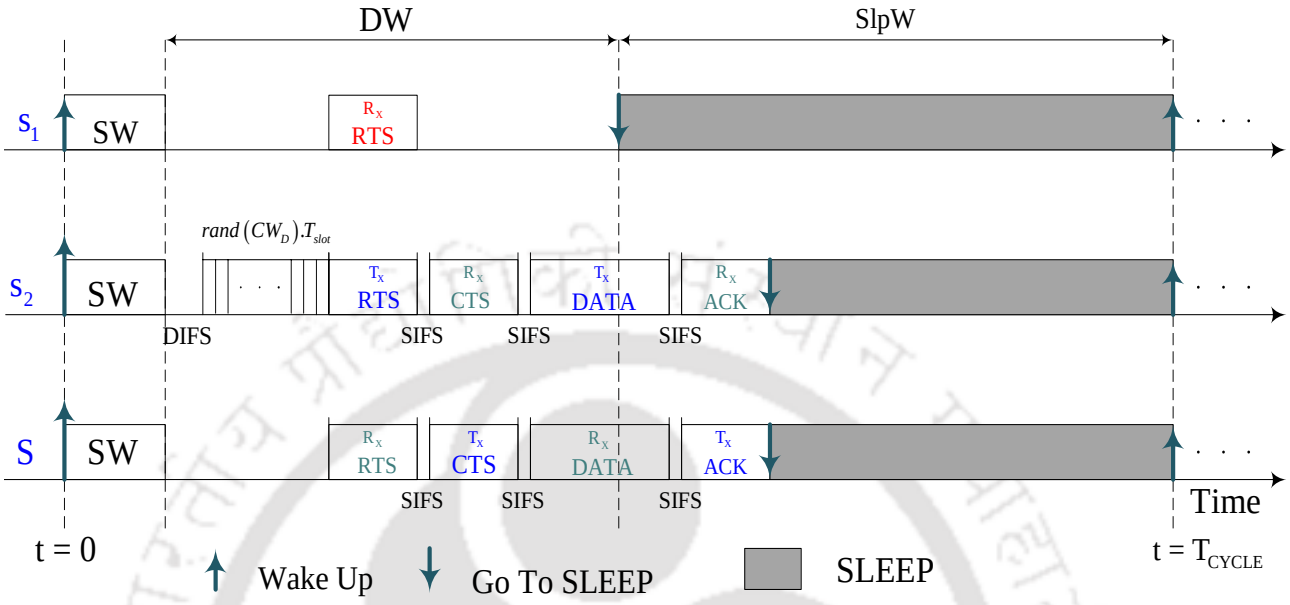


Figure 2.5: Data packet transmission process of SMAC [37].

In SMAC, a data packet can travel at the most one hop in a cycle which results in large E2ETD in a multi-hop WSN. To overcome this shortcoming, adaptive listening technique was introduced in [38]. In [38], a data packet can travel up to two hops in a cycle.

Variable Load Adaptive MAC (VLA-MAC) [39]: VLA-MAC is based on SMAC and it improves SMAC by reducing the E2ETD and energy-consumption. It operates into two modes based on traffic-load, named as, burst transmission mode and normal transmission mode. In VLA-MAC, a node estimates the network traffic load on the basis of packet inter-arrival time in its queue. When a node has more than one data packets in its queue, the node enters in burst transmission mode to transmit multiple data packets in the same cycle. Under low traffic-load, it allows a node to schedule data packet transmission in SW so that the neighboring nodes can go in sleep at the beginning of DW to reduce their energy consumption in *idle listening*.

Timeout MAC (T-MAC) [40]: In SMAC, a node wastes its energy in *idle listening* during the entire DW if it did not transmit/receive any RTS/CTS during DW. Unlike SMAC, T-MAC allows a sensor node to go in *sleep state*, before the beginning of its SlpW, if no *activation event* occurred within TA duration where $TA = C + R + T$. (Here, C, R and T are the length of the contention interval, transmission duration of one RTS packet, and the short time between the end of the RTS

packet and the beginning of the CTS packet, respectively.) In T-MAC, packet reception and packet collision are considered as *activation events*.

By allowing a node to goes in sleep before the beginning of its SlpW may cause a break for packet forwarding. In T-MAC, two schemes are proposed to solve this early sleeping problem, named as, future request-to-send (FRTS) and full-buffer priority. The MAC protocol proposed in [41] also adopted the similar approach for reducing the *idle listening*. However, the techniques proposed in [41] for solving the early sleeping problem is different from that of [40]. In [40,41], a data packet can travel up to three hops in a cycle.

Routing Enhanced MAC (RMAC) [42]: For further improvement in E2ETD in a multi-hop scenario, RMAC accommodates the scheduling process and data transmission process in separate windows. RMAC exploits the cross-layer routing information to enable a sensor node to schedule multi-hop forwarding of one data packet in the DW. For this, RMAC introduce a new control packet, named as pioneer (PION), which contains all the fields included in an RTS and a CTS in SMAC, such as the addresses of sender, next-hop receiver, previous-hop receiver, and duration of the transmission. In addition to this, a PION also contains some cross-layer information: the final destination address of the current flow and hop distance of the sender from the source node of the current flow.

A node with data packet to send, after winning the contention for channel access, forwards a PION along the data forwarding path. For the next-hop receiver, a PION behaves as the data transmission request (i.e., as a RTS), while, for the previous hop receiver, a PION behaves as the confirmation to the recently sent request (i.e., as a CTS). On the basis of its hop distance from the source node of current flow, each sensor node that transmit a PION in DW determines its data packet transmission and/or reception time instants in SlpW. Data packet transmission process of RMAC is illustrated in Fig. 2.6 assuming that the sensor node s_1 has two data packets to send to the sink node S through the sensor node s_2. As can be seen in Fig. 2.6, the node lying at i^{th} hop distance from the source node of its schedule flow forwards its data packet at the time instant $t_{SLEEP} + i.u$ where $u = T_{DATA} + \text{SIFS} + T_{ACK} + \text{SIFS}$ and time instant t_{SlpW} denotes the beginning of SlpW (as shown in Fig. 2.6). (Here, T_{DATA} and T_{ACK} denote the transmission durations of one data and one ACK packet, respectively.)

In [43], Cho and Bahk derived an analytical expression to determine the duty-cycle value that minimizes the overall energy-consumption in RMAC while satisfying a given delay bound.

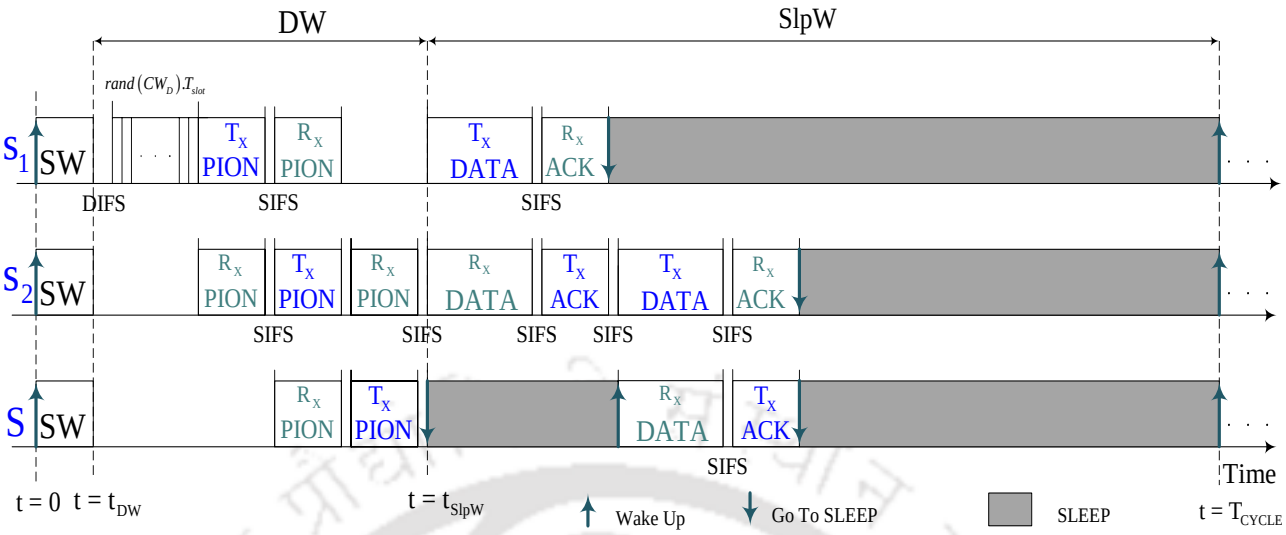


Figure 2.6: Data packet transmission process of RMAC [42].

Look Ahead Scheduling (LAS-MAC) [44]: As in RMAC, in LAS-MAC, scheduling process is accommodated in DW and the data packet transmission is accommodated in SlpW. However, the control packet used in LAS-MAC for scheduling in DW, named as LAS-RTS, does not contain the address of previous hop receiver. Therefore, in LAS-MAC, a node can schedule a longer flow than in RMAC in the same DW duration. Moreover, in LAS-MAC, the data packet plays the role of both data and ACK. The downstream node of the sender considers the received data packet as data while the upstream node of the sender considers it as the ACK of the recently sent data packet. This reduces the energy consumption in control overhead. A similar approach is also adopted in LCO-MAC [45] and AM-MAC [46] to reduce the energy-consumption in control overhead.

Hop Extended MAC (HE-MAC) [47]: HE-MAC is based on RMAC and it improves RMAC by enabling a node to schedule longer flows than RMAC in the same DW duration. It uses the control packet, named as EXP (Explorer), for scheduling the multi-hop forwarding of a data packet in the DW. Other than the information included in the PION, the EXP contains the information about the maximum number of hops it can travel through in the current cycle as *maxHop*.

HE-MAC exploits the fact that a node lying two or more hops away from the sender can sense the ongoing communication. It allows a node to change its internal state from IDLE to RTR (ready-to-receive) if it senses ongoing communication and the time remaining in the beginning of its SlpW is less than $2.T_{EXP} + \text{SIFS}$. (Here, T_{EXP} denotes the transmission duration one EXP packet.) The nodes that are in RTR state do not go to sleep at the beginning of their SlpW so that they can receive

and/or transmit EXP in SlpW. As a result of this, in HE-MAC, a node can schedule a longer flow in the same DW duration than RMAC. Therefore, in a multi-hop WSN, HE-MAC provides lower E2ETD than RMAC. HE-MAC allows a node to change its state from RTR to sleep and follow the sleep/wakeup schedule if it did not transmit/receive any EXP till $t_{SlpW} + 2.(T_{EXP} + SIFS)$.

Pipelined Routing Enhanced MAC (PR-MAC) [48]: In RMAC and [44–47], a node can schedule multi-hop forwarding of only one data packet in a cycle. This leads to increase in E2ETD and AEC when a node has multiple data packets to send to the same final destination in the same cycle.

PR-MAC enables a node to schedule multi-hop forwarding of its multiple data packets in a cycle. For this, unlike RMAC, PR-MAC added two new fields in the PION to accommodate the number of data packets the sender wants to send to its downstream node and the number of data packets the sender can receive from its upstream node. The data packet transmission process of PR-MAC is illustrated in Fig. 2.7 assuming that the sensor node s_1 has two data packets to send to the sink node S through the sensor node s_2. As can be seen in Fig. 2.7, the node lying at i^{th} hop distance from the source node of its schedule flow forwards its j^{th} data packet at the time instant $t_{SlpW} + i.u + (j-1).T_p$ where T_p is known as retransmission period.

Hop extended PRMAC (HE-PRMAC) [49] adopted the RTR mechanism proposed in HE-MAC to improve PR-MAC.

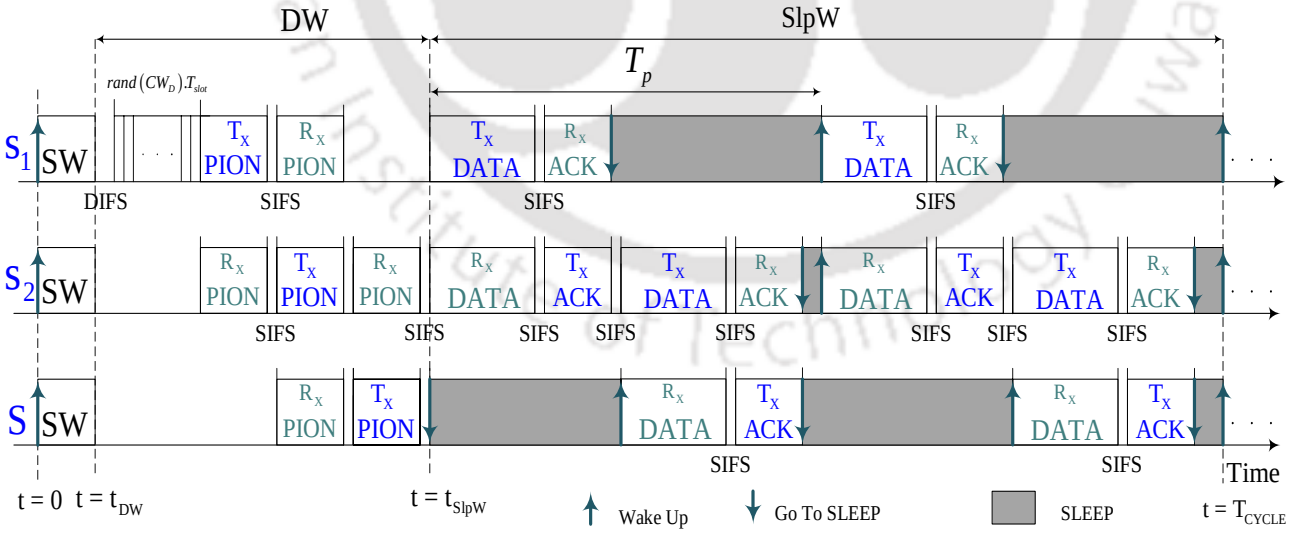


Figure 2.7: Data packet transmission process of PRMAC [48].

BulkMAC [50]: In RMAC and [44–49], a sensor node can setup only one flow in a cycle. This results in larger E2ETD when a node has data packets to send to multiple destinations in the same cycle.

BulkMAC enables a node to setup two types of flow, single-hop multiple receiver flow and multi-hop flow. In a multi-hop flow, a node can forward more than one of its data packets over multiple hops. On the other hand, in a single-hop multiple receiver flow, a node can forward its data packets to more than one one-hop neighbors in the same cycle. Unlike PR-MAC, in BulkMAC, a node forwards its data packets successively (i.e., with $T_p = 0$) to its next-hop receiver node.

Demand Wakeup MAC (DW-MAC) [51]: In the duty-cycled cross-layer *generic* contention based synchronous MAC protocols proposed in [42,44–50], each node of every flow scheduled in a DW determines its data transmission segment (DTS) and/or data reception segment on the basis of its hop distance from the source node of its flow. Therefore, in [42, 44–50], the source node of each flow setup in every DW transmits its first data packet at the beginning of its SlpW (i.e., two or more nodes lying in the CSR of each other cannot forward their data packets in the same cycle). As a result of this, these protocols provides larger E2ETD when multiple nodes with data packets to send lie in the CSR of each other.

Unlike [42, 44–50], DW-MAC assigns DTS and/or DRS to a node by employing a one-to-one *proportional mapping function* between the time interval during a DW and during the subsequent SlpW. As a result of this, in DW-MAC, two nodes lying in the CSR of each other can forward their data packets in the same cycle. As in BulkMAC, in DW-MAC, a node forwards its data packets successively to its next-hop receiver node in its DTS. The control packet used in DW-MAC for scheduling in DW is termed as scheduling frame (SCH). A SCH contains the duration of the transmission, and addresses of sender, receivers, and final destination. Data packet transmission process of DW-MAC is illustrated in Fig. 2.8 assuming that the sensor node s_1 has two data packets to send to the sink node S through the sensor node s_2. In Fig. 2.8, $T_{DTS/DRS}$ denotes the duration of one DTS/DRS; T_i^D and T_i^S $\left(= \left(T_{SlpW}/T_{DW}\right).T_i^D\right)$ denote the time instants when sensor node s_i starts transmitting SCH and data packet, respectively. In [52], Nguyen and Ji derived an analytical expression to determine the optimum value of the *proportional mapping function* that will guarantees no collisions at any intended receiver as well as the minimum E2ETD.

Aggregation MAC (AG-MAC) [53]: AG-MAC is based on DW-MAC and it improves DW-MAC by reducing the energy-consumption in control overhead. Unlike DW-MAC, in AG-MAC, a sensor node aggregates several queued packets, which are destined to the same destination, into a single packet before adding the MAC layer header to it.

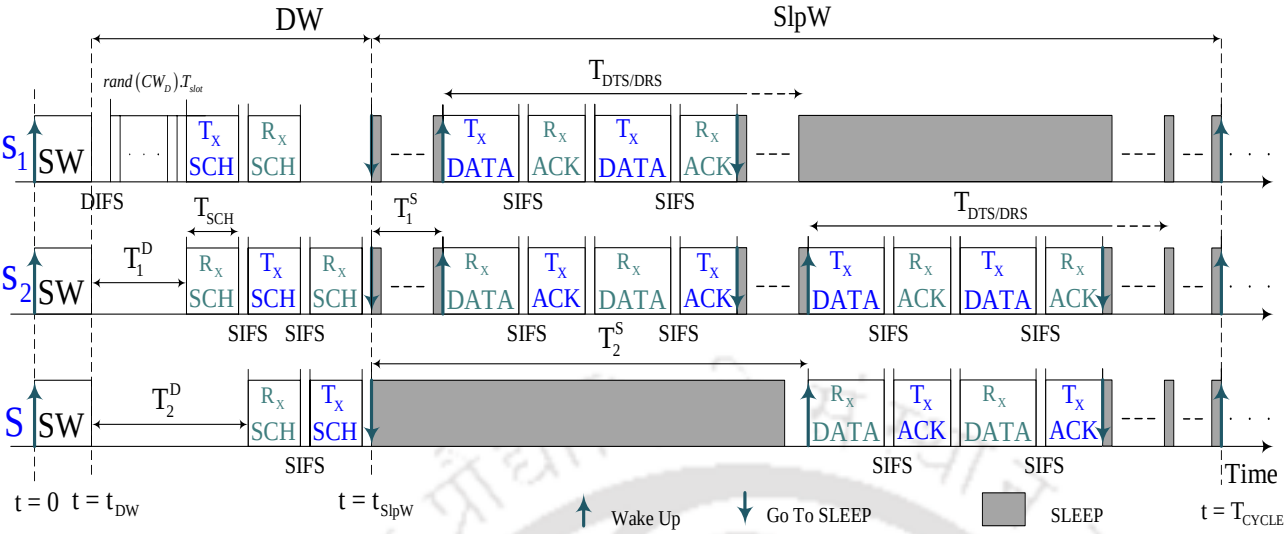


Figure 2.8: Data packet transmission process of DW-MAC [51].

Adaptive Scheduling MAC (AS-MAC) [54]: AS-MAC is based on DW-MAC and it improves DW-MAC by reducing the energy-consumption in *idle listening*. Unlike DW-MAC, in AS-MAC, the wakeup duration of a node in DW is variable and it depends on the traffic load. In case of high traffic load, a node remains awake during the entire DW so that more nodes can forward their data packets in the same cycle. On the other hand, in case of low traffic load, AS-MAC allows a node to go in sleep before the beginning of its SlpW so that the node can reduce its energy-consumption in *idle listening*.

Cross-Layer MAC (CL-MAC) [55]: CL-MAC is based on DW-MAC and it improves DW-MAC by adding two new features, named as, early acknowledgement (EACK) mechanism and multiple multi-hop multi packet flow scheduling feature. CL-MAC is mainly proposed for heterogeneous WSNs where sensor nodes are deployed to monitor multiple events in the same geographic region and different events need to be reported to different sink nodes. EACK mechanism allows a sink node to send confirmation for the received data packet transmission request at the beginning of its DRS so that the sink can schedule reception of data packets with more number of sensor nodes. In addition to this, EACK mechanism also allows a sensor node to send confirmation for the received data packet transmission request at the beginning of its DRS if the remaining DW duration is not sufficient to send the confirmation. In this way, with the help of EACK, a sensor node can schedule one-hop longer flow in the same duration DW than DW-MAC.

In CL-MAC, the control packet used for scheduling in DW is termed as a flow setup packet (FSP). As in [42,44–52], the intended downstream node considers the received FSP as data packet transmission

request while the intended upstream node considers it as confirmation for the recently sent data packet transmission request. As in DW-MAC, in CL-MAC, a node determines its DTS/DRS by employing a one-to-one *proportional mapping function* between the time interval during a DW and during the subsequent SlpW. The unique feature of an FSP is that a node can send data transmission request for multiple different destinations through one FSP. The FSP used for sending the data transmission request for l destinations is denoted by FSP_l. Transmission duration of FSP_l is l times the transmission duration of FSP_1. An FSP_l reserves l segments in SlpW and the duration of each segment is equal to the $(T_{FSP_1}.T_{SlpW})/T_{DW}$ where T_{FSP_1} denotes the transmission duration of one FSP_1. In case of $l > 1$, a node determines its DRS on the basis of its position in the received FSP_l.

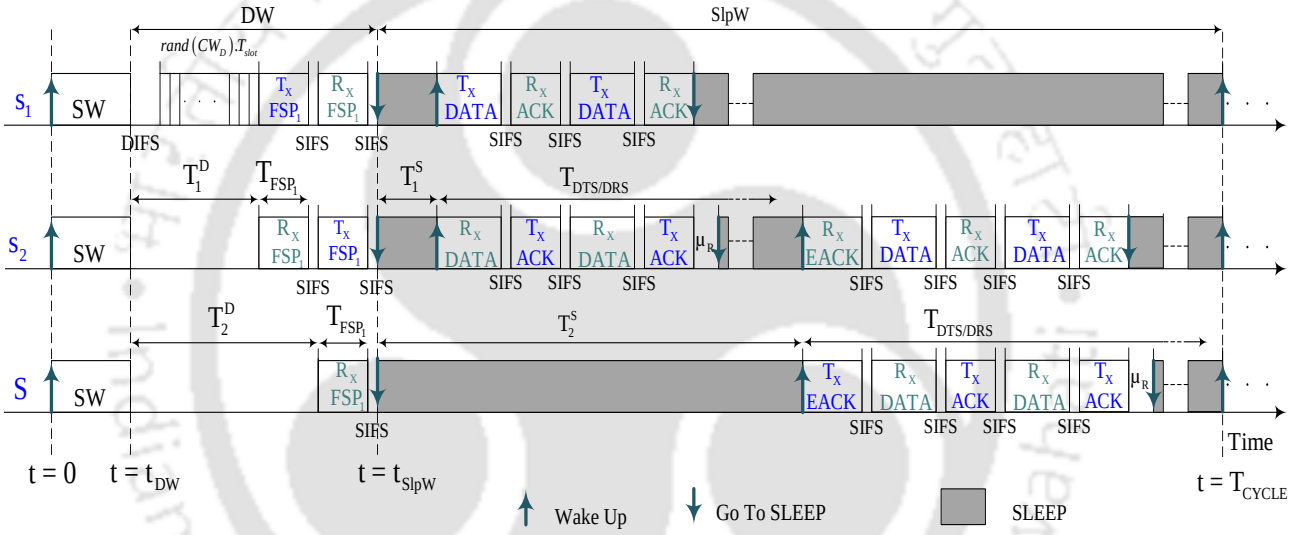


Figure 2.9: Illustration of EACK mechanism of CL-MAC [55].

EACK mechanism of CL-MAC is illustrated in Fig. 2.9, where sensor node s_1 forwards its two data packets to the sink node S, through the sensor node s_2. As can be seen in Fig. 2.9, the sink node sends its confirmation for the data transmission request received in DW at the beginning of its DRS through EACK. It may also be noted that a node goes to sleep if it did not start receiving data packet within μ_R (receive timeout) seconds from the last ACK reception. In Fig. 2.9, T_i^D and T_i^S $\left(= (T_{SlpW}/T_{DW}).T_i^D\right)$ denote the time instants when sensor node s_i starts transmitting FSP_1 and data packet, respectively.

Multiple multi-hop flow scheduling process of CL-MAC is shown in Fig. 2.10 assuming that the sensor node s_1 has one data packet for node A and one data packet for node B in the same cycle. Further, s_1 sends packet to node A and node B through the node s_2 and s_3, respectively. After

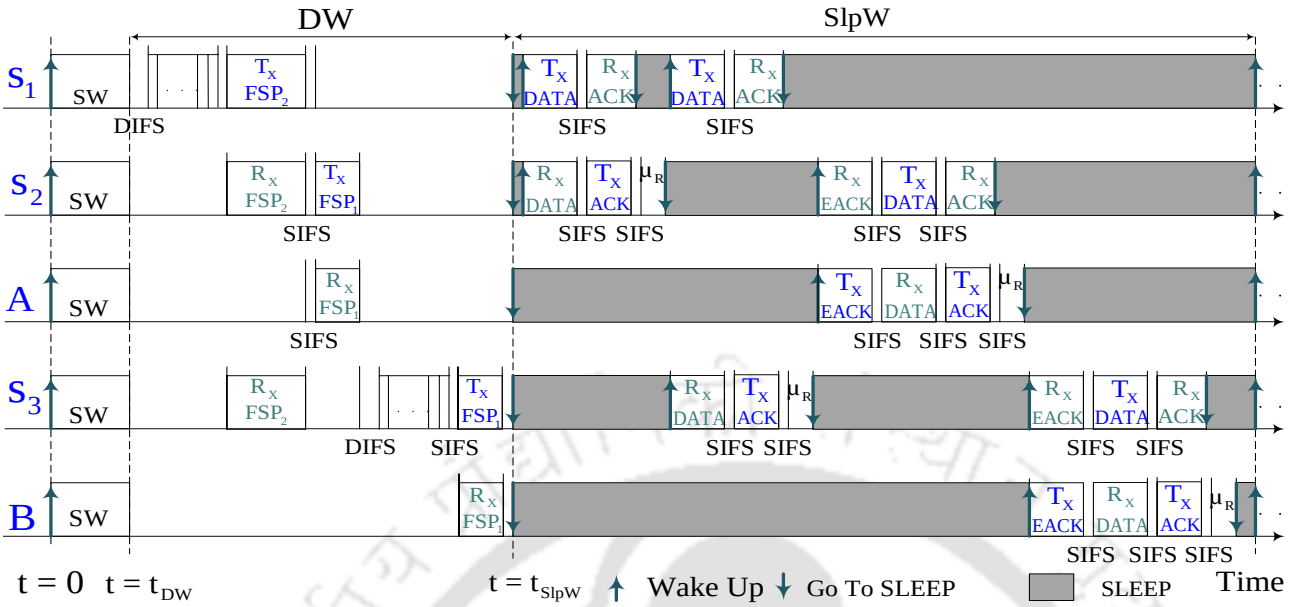


Figure 2.10: Illustration of multiple multi-hop flow scheduling feature of CL-MAC [55].

winning the contention for channel access, s_1 sends FSP_2 with first destination address set to s_2 and second destination address set to s_3. That is the priority is given to the node s_2. After SIFS duration, the node s_2 sends FSP_1 to the node A. On the other hand, node s_3 sets its NAV equal to the $T_{FSP_1} + \text{DIFS} + \text{SIFS}$ and sets NAV timer for this duration. At NAV timer expiration, s_3 sends FSP_1 to the node B. At the beginning of its DRS, a node receives its data packet.

In several WSN applications (e.g., fire detection, radioactive radiation detection, etc.), size of data packet is almost equal to the control packet FSP_1 as few bytes are required to represent the detected information. Further, to reduce the *idle listening*, in a low EOR WSN, sensor nodes prefer to operate with low duty-cycle (i.e., with the higher value of $T_{\text{SlpW}}/T_{\text{DW}}$). Moreover, in event-driven WSNs, each sensor node uses data aggregation [81–83] and/or data compression [84,85] techniques to optimize the size and number of data packets to be sent in a cycle. Therefore, in CL-MAC, most of the time, a sensor node utilizes only a small fraction of its DTS/DRS in transmitting/receiving the data packets. Furthermore, the multiple flow scheduling feature of CL-MAC does not help much in E2ETD improvement in the low EOR scenarios. The reasons are (1) due to low EOR, the probability of a node having data packets to send for more than one destination in the same cycle is low, (2) as can be seen in Fig. 2.10, this feature is beneficial mainly when DW is large so that the last nodes of the flows scheduled in DW do not lie in the CSR of each other.

Thus, utilizing the synchronization and routing layer information, in the *generic* cross-layer MACs [42, 44–54], a node can schedule multi-hop forwarding of its data packets in one cycle. As a result of this, these protocols provide lower E2ETD than the contention based asynchronous MACs and contention-free (TDMA) MACs in low EOR delay-sensitive event reporting in a dense multi-hop WSN. Further, in *generic* MACs, each node periodically broadcasts a *synchronization message* containing its current sleep/wakeup schedule. Therefore, these protocols quickly adopt the effects of network topology changes due to insertion of new nodes and nodes failure. These features make such protocols a more suitable choice than the TDMA and contention based asynchronous MACs, for low EOR delay-sensitive event monitoring in a dense multi-hop WSN. Furthermore, mainly due to its EACK mechanism, CL-MAC provides lower E2ETD than other existing *generic* cross-layer MACs in a multi-hop WSN. Above features of *generic* cross-layer MACs, limitations of CL-MAC, and the requirement of as low as possible E2ETD in delay-sensitive event reporting motivates for the *generic* cross-layer MAC that can further reduce the E2ETD in event reporting in a dense multi-hop WSN.

2.2.2 *Staggered* Contention Based Synchronous MAC Protocols

In the cross-layer *generic* contention based synchronous MAC protocols, such as [42,44–55], all the neighbors of a sensor node follow the same sleep/wakeup schedule. Therefore, in these protocols, the maximum number of hops traveled by a data packet in a cycle depends on the size of DW (or duty-cycle). *Staggered* contention based synchronous MAC protocols [56–64] are proposed to overcome this constraint. Unlike *generic* contention based synchronous MAC protocols, in these protocols, a node can report its detected information only to the sink node. In this section, we discuss the basic and recently proposed *staggered* contention based synchronous MAC protocols in detail.

DMAC [56]: DMAC is the first *staggered* contention based synchronous MAC protocol proposed for WSNs. As can be seen in Fig. 2.11, the cycle structure followed by the sensor nodes in DMAC contains three windows - receive window (RxW), transmit window (TxW) and sleep window (SlpW). In receive window, a node can receive one data packet from its upstream node. In transmit window, a node can send one data packet to its downstream node if it succeeds in medium contention. In SlpW, each node remains in sleep to save energy. The duration of the transmit window and the receive window are equal and sufficient to communicate one data packet. As can be seen in Fig. 2.11, a node that lies at i^{th} hop distance from the sink node follows this cycle structure such that its TxW completely overlaps

with RxW of the nodes lying at $i-1$ hops distance from the sink node and its RxW completely overlaps with the TxW of the nodes lying at $i+1$ hops distance from the sink node. In Fig. 2.11, the sensor node s_i lies at i hops distance from the sink node (S). In most WSN applications, the size of the data packet is almost equal to the control packets, RTS/CTS. Therefore, to remove unnecessary overhead, DMAC does not use RTS/CTS packets.

In DMAC, a node can forward more than one data packet in the same cycle. For this, DMAC includes a one-bit field, named as *more data flag*, in the MAC header. In case, a node has more than one data packets to send to its downstream node or it receives a data packet with *more data flag* set, it sends data packet with *more data flag* set. A node sends ACK with *more data flag* if it receives data packet with *more data flag* set.

MERLIN [57] extended DMAC by employing a multicast upstream and multicast downstream approach for relaying the packet to and from the sink (or gateway) node. Robust Multi-Pipelined Scheduling (RMS) [58] analyzed the impact of unreliable link quality on the E2ETD performance of single predetermined routing path with a *staggered* wakeup scheduling approach. In [56–58], it assumed that the local synchronization is enough to maintain the assigned *staggered* wakeup schedules and some of the existing synchronization protocols could meet their requirements. However, they do not provide details about how to implement the synchronization protocol.

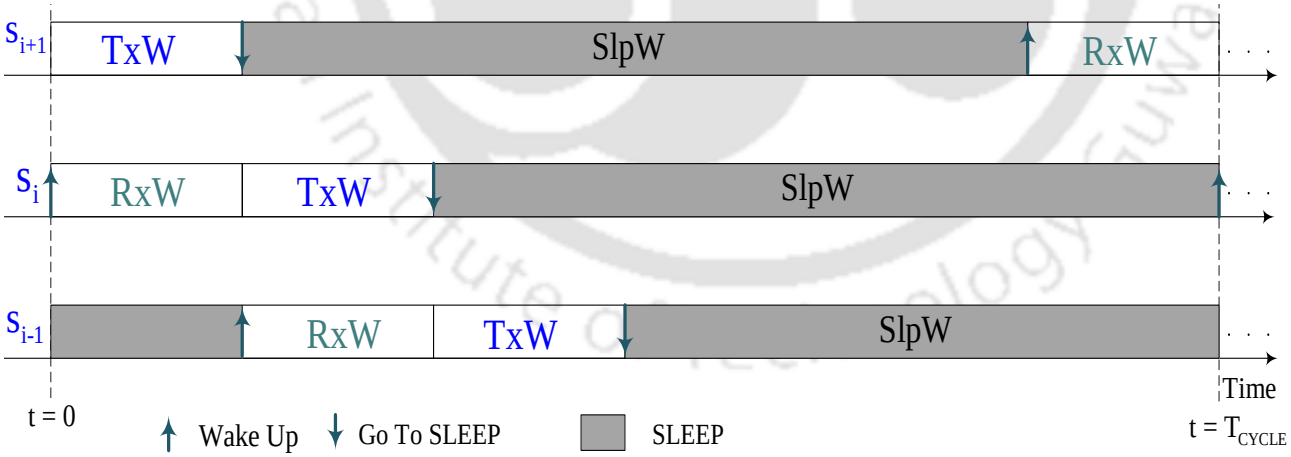


Figure 2.11: *Staggered* wakeup schedules of the sensor nodes lying at $i-1$, i, and $i+1$ hops distance from the sink node.

PMAC [59]: PMAC is based on DMAC but, unlike DMAC, PMAC does not require a separate synchronization protocol. In PMAC, for *staggered* wakeup schedule assignment, in the initial *grade division and schedule assignment (GDSA) process*, after setting its own *grade* to zero, the sink node

broadcasts a packet (termed as GRADE) containing *grade* 1. *Grade* of a node represents its hop distance from the sink node. If a node receives a GRADE packet containing *grade* i and its current *grade* is greater than i, then it sets its current *grade* to i, determines its schedule based on the current *grade*, and then broadcasts a GRADE packet containing *grade* $i+1$. An i^{th} *grade* node determines its schedule such that its TxW and RxW completely overlaps with the RxW and TxW of the nodes having *grade* $i-1$ and $i+1$, respectively.

PRI-MAC [60] modifies and improves the *GDSA* process of PMAC by adding more information on the current *grade*, state and state duration of the sender in the GRADE packet. (Here, current state and state duration of a node denotes its current window and the time spent by the node in the current window, respectively.) Compared to PMAC, these enhancements allow PRI-MAC to reduce the synchronization error in the schedules initially assigned to the nodes. However, in both PMAC and PRI-MAC, due to lack of a periodic schedule synchronization mechanism, the synchronization between the schedules progressively decrease over time. Reduced pipelined forwarding MAC (RP-MAC) [61] improved PRI-MAC by reducing the energy-consumption in *idle listening* and *control packets overheads*.

Pipelined Data Collection (PDC) [62]: PDC is based on PRI-MAC. It proposed a periodic synchronization mechanism to maintain the wakeup schedules assigned in the *GDSA* process. For this, in PDC, each node periodically follows the PRTS/PCTS unicast handshake with its parent node. The same handshake is also the initial part of the process followed by a node to transmit a data packet to its parent node (i.e., PRTS/PCTS/DATA/ACK). Further, in PDC, whether a node follows the handshake as per its periodic requirement for schedule synchronization or as per its immediate requirement for data transmission, it does so at the beginning of its TxW. Therefore, there is the possibility that, in the same cycle, a node (say, A) needs to follow the handshake for data transmission and another node of the same *grade* (say, B) which lies in the CSR of A, needs to follow the handshake as per its periodic need for schedule synchronization. In that case, node A cannot send data packet to its parent node if node B succeeds in its medium contention and would result in an increase in the transmission delay. In addition, as in [59–61], a node can transmit and/or receive only one data packet in a cycle. Data packet transmission process of PDC is shown in Fig. 2.12, assuming that (1) the sensor node s_2 has one data packet to send to the sink node S and the sensor node s_i lies at i^{th} hops distance from the sink node S where $i \in [1, 2]$.

Adaptive data collection (ADC) [63] protocol, following the synchronization technique proposed in PDC, further improved the PDC with the help of free addressing and dynamic duty-cycle. In ADC, a node can forward more than one data packet in a cycle.

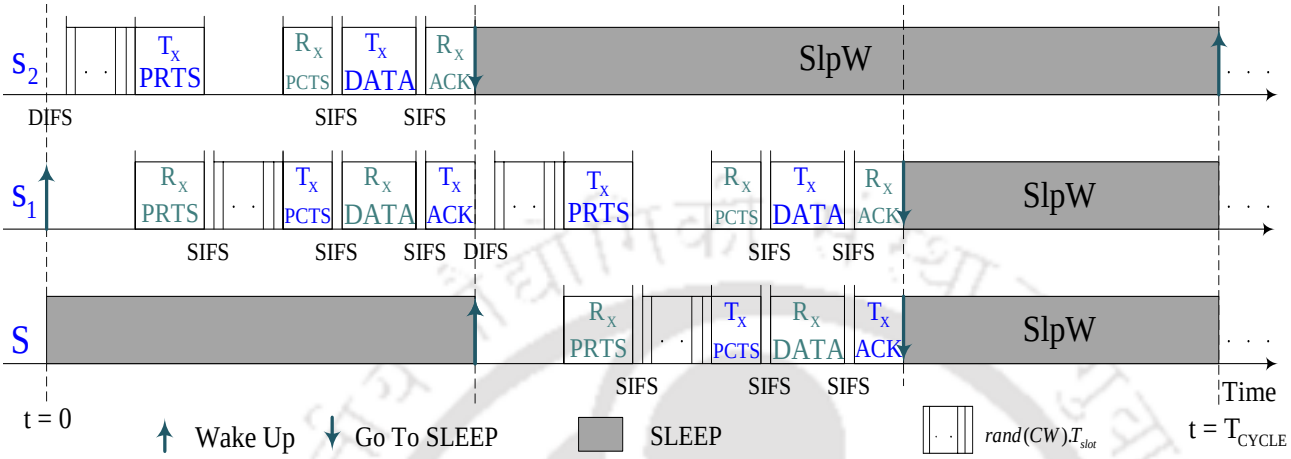


Figure 2.12: Data packet transmission process of PDC [62].

CROP-MAC [64]: To avoid the delay in data packet transmission due to the interference of synchronization traffic, as can be seen in Fig. 2.13, CROP-MAC proposed a cycle structure which contains separate windows for the data packet transmission scheduling (DW_2), data packet reception scheduling (DW_1), periodic SYNC packet broadcast (SW_2), and SYNC packet reception (SW_1). As can be seen in Fig. 2.14, an i^{th} grade node follows this cycle structure such that its SW_1 and DW_1 completely overlaps with the SW_2 and DW_2, respectively, of the nodes having grade $i+1$. As in [42, 44–55], CROP-MAC assumes that a flooding based synchronization protocol is used to synchronize the clocks of neighboring sensor nodes with the required precision during SW_1 and SW_2.

Further, as in ADC, CROP-MAC adopts a dynamic duty-cycle to enable a node to schedule the transmission of multiple pending data packets to its next-hop forwarder in one cycle. CROP-MAC replaces RTS/CTS with the RTSD (request-to-send data)/CTSD (confirmation-to-send data). RTSD and CTSD, each packet contains addresses of sender and receiver. Data packet transmission process of CROP-MAC is shown in Fig. 2.14, assuming that (1) the sensor node s_2 has two data packets to send to the sink node S and (2) the sensor node s_i lies at i^{th} hops distance from the sink node S where $i \in [1, 2]$. In Fig. 2.14, T_1 denotes the time required for avoiding collision between the data packet and the control packet (RTSD/CTSD) and T_2 denotes the time required to avoid collision between the data packets.

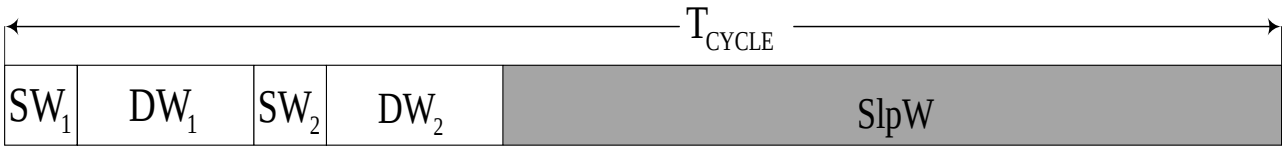


Figure 2.13: Cycle structure followed by the network nodes in CROP-MAC [64].

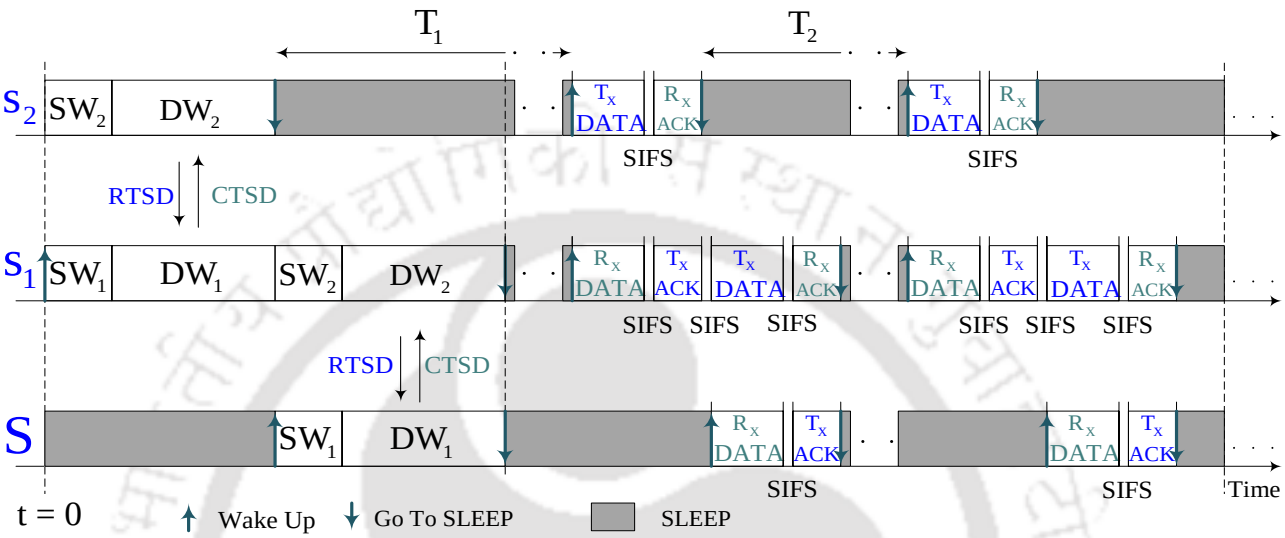


Figure 2.14: Data packet transmission process of CROP-MAC [64].

However, as in [59–63], the CROP-MAC has the following shortcomings: 1) at the most one node can schedule data packet transmission in a cycle when multiple nodes with data packets to send lie in the CSR of each other; 2) a node gets only one chance for trying to schedule data packet transmission in a cycle; and 3) the sink node can receive data packets from at the most one sensor node in a cycle. The above three shortcomings limit the E2ETD performance of these protocols in an event driven dense multi-hop WSN.

Unlike *generic* MACs, in *staggered* MACs, a node randomly chooses one of its adjacent lower grade node, as its next-hop forwarder. Therefore, they do not require an additional protocol to assign the next-hop forwarder to the nodes. Moreover, due to staggered wakeup schedules of adjacent grade nodes, staggered MACs provide lower E2ETD than the *generic* MACs in a multi-hop converge-cast communication scenario. However, unlike *generic* MACs, in *staggered* MACs, a sensor node can forward its data packets only to its adjacent lower grade nodes. Therefore, *staggered* MACs are not useful for WSNs deployed for monitoring multiple events in the same geographic region with the constraint that the two or more events must be reported to the different sink nodes.

2.3 Synchronization Protocols for WSNs

Each sensor node deployed in a WSN has its own clock that frequently drifts apart due to the imperfections of the clock oscillator. Since the drift may not be same for the different nodes, the clocks of the neighboring sensor nodes may not always remain synchronized even when they are initially perfectly synchronized. The lack of synchronization in the sleep and wakeup time instants of neighboring sensor nodes in a multi-hop WSN leads to inefficient and inaccurate operation of a duty-cycled contention based synchronous MAC protocol.

Protocols proposed for synchronizing the clocks of neighboring sensor nodes can be divided into two groups, flooding based protocols [72–76] and fully distributed protocols [77–80]. In the fully distributed protocols, each sensor node interacts only with its one-hop neighbors in a peer-to-peer fashion. On the other hand, in the flooding based protocols, one or more nodes behave as the time reference for the remaining deployed nodes. Flooding Time Synchronization Protocol (FTSP) [72] is a representative example of flooding based synchronization protocols. In FTSP, each *reference node* periodically floods the current value of its clock into the network through the *synchronization message*. At each new *synchronization message* reception, every sensor node establishes a linear relationship between the *reference node's* clock and its own clock using least-squares regression technique. This linear relationship helps a sensor node in predicting the future clock value of the *reference node*. At the expiration of its broadcast timer, a node broadcasts this predicted reference clock value to its neighbors through the *synchronization message* so that all the deployed sensor nodes can follow the same sleep/wakeup schedule. It is shown through the experimental results that, in a multi-hop WSN, FTSP provides 0.5 μs per hop average synchronization error. The synchronization accuracy of FTSP in a multi-hop WSN is further improved by the subsequently proposed flooding based synchronization protocols, such as [74–76] etc. The flooding based synchronization protocols are easy in implementation, and robust to network topology changes and nodes and links failure. Therefore, in most of the existing duty-cycle contention based synchronous MAC protocols, flooding based synchronization protocols are used to synchronize the sleep and wakeup time instants of neighboring sensor nodes.

In most of the existing duty-cycled contention based synchronous MAC protocols (e.g., [42, 44–55] etc.), *synchronization message* is termed as SYNC packet. The cycle structure followed by the network

nodes in the existing *generic* contention based synchronous MAC protocols contains a synchronization window (SW) so that each node can periodically broadcast a SYNC packet in its SW without affecting the data packet transmission process. In the protocols propose in Chapter 3 and 4 of this thesis, a sink node behaves as the *reference node* and each node periodically broadcasts a SYNC packet in its SW. Unlike [42, 44–55], as in [64], the cycle structure followed by the network nodes in the protocol propose in Chapter 5 contains separate SWs for the periodic SYNC broadcast to its adjacent higher *grade* nodes and for the SYNC packet reception from its adjacent lower *grade* nodes. The reason for this is that the wakeup schedules of adjacent *grade* nodes in [64] and the protocol propose in Chapter 5 are *staggered*. The flooding based synchronization techniques make the duty-cycled contention based synchronous MACs robust to network topology changes and nodes failure.

3

A Low Delay Cross-Layer MAC Protocol for *k-covered* Event-Driven WSNs

Contents

3.1	Introduction	38
3.2	Description of LDC-MAC	40
3.3	Results	45
3.4	Conclusions	53

In this chapter[1], we propose and present LDC-MAC, a low delay cross-layer MAC protocol for *k-covered* event-driven WSNs. LDC-MAC is a cross-layer *generic* contention based synchronous MAC protocol. It reduces the end-to-end transmission delay (E2ETD) by increasing the number of nodes that can forward their data packets in the same cycle when multiple nodes with data packets to send lie in the CR of each other. To achieve this, LDC-MAC assigns an interference free DRS to each node of every flow scheduled in a DW and efficiently utilizes the assigned DRS. As in [51–55], LDC-MAC assigns DTS and/or DRS to a node by employing a one-to-one *proportional mapping* function between the time interval during a DW and during the subsequent SlpW. The DRS utilization strategy of LDC-MAC is inspired by the spectrum utilization strategy in cognitive radio networks (CRNs) where a user can utilize the spectrum allocated to another user (known as authorized user) when the authorized user is not utilizing it.

3.1 Introduction

In an event-driven *k-covered* WSN, an event is detected by the M ($\geq k$) sensor nodes. Each sensor node that detects an event, generates data packets to report the event to the sink node. In delay-sensitive event monitoring applications of WSNs, a detected event should be reported to the sink node as soon as possible. Therefore, all the M sensor nodes try to forward their data packets to the sink node in the same cycle. In general, a sensor node has a larger communication range (CR) than its sensing range. Therefore, out of M such sensor nodes, two or more sensor nodes may lie in the CR of each other. In addition to this, in an event-driven WSN, a large number of sensor nodes report their detected events to the same sink node. Because of this, multiple sensor nodes lying in the CR of a sink node may have data packets to send to the sink node in the same cycle.

As mentioned in Chapter 2, due to its DTS/DRS assignment technique and early acknowledgement (EACK) mechanism, CL-MAC [55] provides lower E2ETD than the other existing *generic* contention based synchronous MAC protocols [37–55] when multiple sensor nodes lying in the carrier sensing range (CSR) of each other try to forward their data packets in the same cycle. In CL-MAC, the duration of one DTS/DRS is equal to γ times the transmission duration of the control packet used for sending data transmission request. (Recall that γ denotes the ratio of SlpW and DW durations.)

[1]The work reported in this chapter has been presented in the following journal publication:
Ripudaman Singh, Brijesh K. Rai, and Sanjay K. Bose, "A low delay cross-layer MAC protocol for k-covered event driven wireless sensor networks," *IEEE Sens. Lett.*, vol. 1, no. 6, pp. 1-4, Dec. 2017.

In event-driven WSNs, event occurrence rate (EOR) remains low most of the time. Therefore, in such WSNs, sensor nodes prefer to operate with a low duty-cycle (i.e., prefer to follow the cycle structure having small size DW and large size SlpW) as it reduces the energy consumption in *idle listening*. With a low duty-cycle, a large value of γ makes the DTS/DRS size long enough to transmit/receive multiple data packets. However, in WSNs, a sensor node utilizes data compression [81–83] and/or data aggregation [84, 85] techniques to optimize the size and number of data packets to be forwarded. Therefore, in CL-MAC, most of the time, a sensor node has only a few data packets to send compared to what a node can receive in its DRS. As a result, most of the time, a large portion of a node's DRS remains unused. Following are the main shortcomings of CL-MAC.

(i) It does not assign DRS to the source node of any flow scheduled in the DW.

(ii) A sensor node does not utilize the unused portion of its one-hop neighbor node's DRS to forward its data packets if it fails in channel contention in the DW.

In this chapter, we propose a low delay cross-layer MAC protocol, named as LDC-MAC, for *k-covered* event-driven WSNs. The features that make LDC-MAC better than the CL-MAC are the following.

(i) It assigns an interference free DRS to each node of every flow scheduled in the DW. The DRS of each node is followed by its DTS.

(ii) It enables the source node of each flow scheduled in the DW to receive data packets in its DRS from the one-hop neighbors which failed in medium contention in the same DW. Such a node forwards its own data packets and the data packets received in the DRS to its next-hop forwarder.

(iii) It also enables a sensor node (say, s_j) to receive data packets from a one-hop neighbor (say, s_t) in the unused portion of its DRS if s_t failed in scheduling in the DW and s_j receives data transmission request as an intended receiver from a sensor node (say, s_i) in the same DW. In this case, s_j forwards the data packets received from s_i and s_t to its next-hop receiver node. (Here, unused portion of s_j's DRS refer to the portion of DRS which remains after receiving data packets from s_i.)

These novel features overcome the two shortcomings of CL-MAC mentioned earlier and allow more number of sensor nodes than CL-MAC to forward their data packets in the same cycle when multiple

sensor nodes, with data packets to send, lie in the CR of each other. This reduces the waiting time of data packets in the queue of sensor nodes allowing LDC-MAC to provide lower E2ETD than CL-MAC.

3.2 Description of LDC-MAC

In this section we describe LDC-MAC in detail with the following assumptions (1) the network contains n sensor nodes (s_1, $s_2, \ldots,$ s_n) and t sink nodes which are randomly deployed, (2) all the nodes are stationary and the CR of each node is the same, (3) an event is detected by the M sensor nodes, (4) all nodes are following the same sleep/wakeup schedule (as mentioned in Chapter 2, in contention based synchronous MAC protocols, a separate protocol is used to synchronize the sleep/wakeup schedules of network nodes) and (5) each sensor node is aware of its next-hop forwarder corresponding to each sink node.

We start with the process followed by a node to schedule the multi-hop forwarding of its data packets in DW, followed by the description of DTS/DRS assignment process and the data transmission process. An illustration of the data transmission process is also included.

3.2.1 Data Packet Transmission Flow Setup Process in DW

Each sensor node, which has data packets to send, schedules a timer of DIFS $+ rand\,(CW_D)\,.T_{slot}$ duration at the beginning of its DW, and senses the medium. (Here, $rand\,(CW_D)$ and T_{slot} represent the randomly chosen slot of contention window used in DW (CW_D) and duration of one slot of CW_D, respectively.) Such a node sends a data transmission request to its intended next-hop receiver node if it did not sense any ongoing communication until the expiration of its timer.

A node sends data transmission request through the control packet FSP (flow setup packet) [55] which contains addresses of sender, next-hop receiver, previous hop receiver and final destination. The next-hop receiver node interprets a received FSP as the data transmission request, whereas, the previous-hop receiver node interprets it as the confirmation for its recently sent data transmission request. A node, that receives FSP as an intended next-hop receiver, forwards the data transmission request (FSP) to its next-hop receiver if it is not the final destination and the remaining duration of DW is sufficient to send FSP. In this way, a node schedules multi-hop forwarding of its data packets in its DW.

In case a node overhears FSP before the expiration of its timer, it cancels its timer and goes in sleep for next $T_{FSP} + 2$.SIFS seconds where T_{FSP} denotes the transmission duration of one FSP packet. It

then wakes up and again schedules a timer for DIFS + $rand\,(CW_D)\,.T_{slot}$ duration if the remaining duration of DW is more than the DIFS + $rand\,(CW_D)\,.T_{slot} + T_{FSP}$ duration.

For ease of understanding, in a SlpW, we term a node as a primary sender (PS) node if it is a part of any flow scheduled in the DW of the same cycle. (Note that, each node, that either sends a FSP in the DW or receives a FSP as an intended next-hop receiver in the DW, is the part of a scheduled flow.) On the other hand, in a SlpW, we term a node as a secondary sender (SS) node if it has data packets to send and it is not a part of any flow scheduled in the DW of the same cycle. Unlike the existing cross-layer *generic* contention based synchronous MAC protocols [42–55], the distinguishing feature of LDC-MAC is that it enables a SS node to send its data packets in the DRS of its one-hop neighbor PS node. It is this feature of LDC-MAC that results in its better performance than the existing cross-layer *generic* contention based synchronous MAC protocols.

3.2.2 DTS/DRS Assignment Process

As in [51–55], in LDC-MAC, DTS/DRS are determined on the basis of (a) the time instant when a node starts to transmit/receive data transmission request in DW and (b) γ (the ratio of the durations of SlpW and DW).

DTS/DRS assignment for PS nodes: After sending FSP in DW, a sensor node (s_i) determines the beginning of its $\gamma.T_{FSP}$ duration DTS (in the subsequent SlpW) as,

$$\mathrm{t}_{\mathrm{Tx}}^{\mathrm{DATA}}\left(s_i\right) = \mathrm{t}_{\mathrm{SlpW}} + \gamma.\left(\mathrm{t}_{\mathrm{Tx}}^{\mathrm{FSP}}\left(s_i\right) - \mathrm{t}_{\mathrm{DW}}\right), \tag{3.1}$$

where, $\mathrm{t}_{\mathrm{Tx}}^{\mathrm{FSP}}\left(s_i\right)$ denotes the time instant when s_i starts transmitting FSP in the DW, and time instants t_{DW} and $\mathrm{t}_{\mathrm{SlpW}}$ denote the beginning of DW and SlpW, respectively.

On the other hand, on FSP reception as an intended next-hop receiver in DW, the receiver node (s_j) determines the beginning of its $\gamma.T_{\mathrm{FSP}}$ duration DRS (in the subsequent SlpW) as,

$$\mathrm{t}_{\mathrm{Rx}}^{\mathrm{DATA}}\left(s_j\right) = \mathrm{t}_{\mathrm{SlpW}} + \gamma.\left(\mathrm{t}_{\mathrm{Rx}}^{\mathrm{FSP}}\left(s_j\right) - \mathrm{t}_{\mathrm{DW}}\right), \tag{3.2}$$

where, $\mathrm{t}_{\mathrm{Rx}}^{\mathrm{FSP}}\left(s_j\right)$ denotes the time instant when s_j start receiving FSP in the DW.

As mentioned in the earlier section, if a sensor node s_i sends FSP as a source node, the node s_i did not sense any ongoing communication in between the time instant $\mathrm{t}_{\mathrm{Tx}}^{\mathrm{FSP}}\left(s_i\right) - \mathrm{DIFS}$ and time instant $\mathrm{t}_{\mathrm{Tx}}^{\mathrm{FSP}}\left(s_i\right)$ in DW and s_i will not observe any ongoing communication in between the time instant $\mathrm{t}_{\mathrm{SlpW}} + \gamma.\left(\mathrm{t}_{\mathrm{Tx}}^{\mathrm{FSP}}\left(s_i\right) - \mathrm{DIFS} - \mathrm{t}_{\mathrm{DW}}\right)$ and the time instant $\mathrm{t}_{\mathrm{SlpW}} + \gamma.\left(\mathrm{t}_{\mathrm{Tx}}^{\mathrm{FSP}}\left(s_i\right) - \mathrm{t}_{\mathrm{DW}}\right)$ in the

subsequent SlpW. Unlike [51–55], in LDC-MAC, we assign this γ.DIFS duration segment (between time instants $t_{SlpW}+\gamma.\left(t_{Tx}^{FSP}(s_i)-DIFS-t_{DW}\right)$ and $t_{SlpW}+\gamma.\left(t_{Tx}^{FSP}(s_i)-t_{DW}\right)$) to the source node as its DRS. Therefore, the source node (s_i) determines the beginning of its γ.DIFS duration DRS (in the subsequent SlpW) as,

$$t_{Rx}^{DATA}(s_i)=t_{SlpW}+\gamma.\left(t_{Tx}^{FSP}(s_i)-DIFS-t_{DW}\right). \tag{3.3}$$

DTS assignment for SS nodes: After overhearing FSP, a SS node follows Algorithm 3.1 to choose an appropriate next-hop receiver node and to determine the chosen node's DRS. A SS node determines the chosen node's DRS as its DTS. In Algorithm 3.1, the process is described assuming that s_t failed in scheduling in DW and it overhears the FSP send by s_i to s_j in the same DW. In this algorithm, s_t chooses s_i as its next-hop receiver only when s_j does not lie in its CR. This happens because s_j's hop distance from sink node is less than that of s_i. DRS of s_i is determined based on whether s_i is the source node or not. (Note that a zero previous hop receiver address in the received FSP indicates that the sender is the source node.)

Algorithm 3.1 Process followed by the SS node s_t to determine the beginning of its DTS $\left(t_{Tx}^{DATA}(s_t)\right)$.

\# **Input:** $t_{Rx}^{FSP}(s_t)$, γ, t_{SlpW}, t_{DW}
\# **Output:** $t_{Tx}^{DATA}(s_t)$
1: **if** s_j is one-hop neighbor of s_t **then**
2: $\quad t_{Tx}^{DATA}(s_t) \leftarrow t_{SlpW}+\gamma.\left(t_{Rx}^{FSP}(s_t)-t_{DW}\right)$
3: **else if** s_i is source node **then**
4: $\quad t_{Tx}^{DATA}(s_t) \leftarrow t_{SlpW}+\gamma.\left(t_{Rx}^{FSP}(s_t)-DIFS-t_{DW}\right)$
5: **else**
6: $\quad t_{Tx}^{DATA}(s_t) \leftarrow t_{SlpW}+\gamma.\left(t_{Rx}^{FSP}(s_t)-T_{FSP}-SIFS-t_{DW}\right)$
7: **end if**

3.2.3 Process Followed by a SS Node and a PS Node for Data Packet Transmission

At the beginning of its DTS, a PS node follows RTS/CTS handshake with its next-hop receiver node and sends its data packets after this. A request-to-send (RTS) [37] packet contains the addresses of its sender and receiver, and the time required by the sender to send its data packets. A clear-to-send (CTS) [37] packet contains addresses of its sender and receiver, and the time requested in RTS for data transmission.

On the other hand, at the beginning of its DTS, a SS node sets a timer for DIFS+$rand\left(CW_D\right).T_{slot}$ duration and senses the medium. On RTS/CTS reception before timer expiration, a SS node cancels

its timer and goes in the sleep mode for the duration given in the received packet. It then wakes up and again sets a timer for $rand\,(CW_D)\,.T_{slot}$ duration if the remaining duration of its DTS is sufficient to send a data packet. If a SS node did not sense any ongoing communication until the expiration of its timer, it follows RTS/CTS handshake with its chosen next-hop receiver node and sends its data packets.

3.2.4 Illustration of the Data Packet Transmission Process

In this section, we give an example to describe the process followed by a sensor node to forward its data packet.

Network Model: To illustrate the data transmission process, we consider the network shown in Fig. 3.1. This network contains n $(= 50)$ randomly deployed sensor nodes $(s_1,\ s_2,\ \ldots,\ s_n)$ and one sink node (S). In this network (a) s_1, s_2 and s_3 lie in the CR of each other; (b) s_1 and s_2 each has one data packet to send to the sink S; (c) s_1 and s_2 do not lie in the CR of S; and (d) s_3 and S are next-hop receivers of s_1 and s_3, respectively.

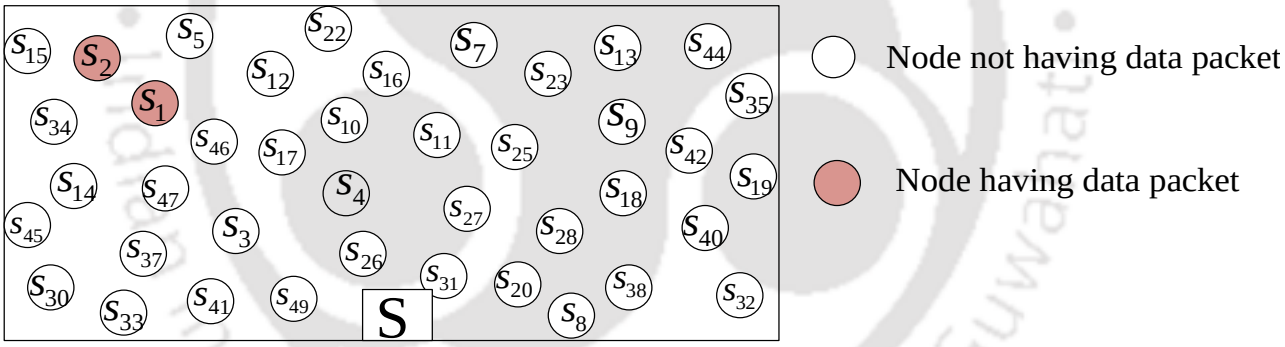


Figure 3.1: Network containing 50 randomly deployed sensor nodes and one sink node (S).

Data Packet Transmission Flow Setup in DW: Here, we assume that the node s_1 wins the contention for channel access (Fig. 3.2) and sends its FSP to the node s_3. Since, s_3 is not the final destination, it sends FSP to its next-hop receiver (S). The sensor node s_2 loses the contention for channel access to s_1 and it overhears the FSP sent by the sensor node s_1.

DTS/DRS assignment: After sending FSP, PS nodes s_1 and s_2 follow (3.1) to determine the beginning of their $\gamma.T_{FSP}$ duration DTS. On FSP reception as an intended next-hop receiver, s_3 and S follow (3.2) to determine the beginning of their $\gamma.T_{FSP}$ duration DRS. The source node s_1 follows (3.3) to determine the beginning of its γ.DIFS duration DRS. In Fig. 3.2, it may be noted that the

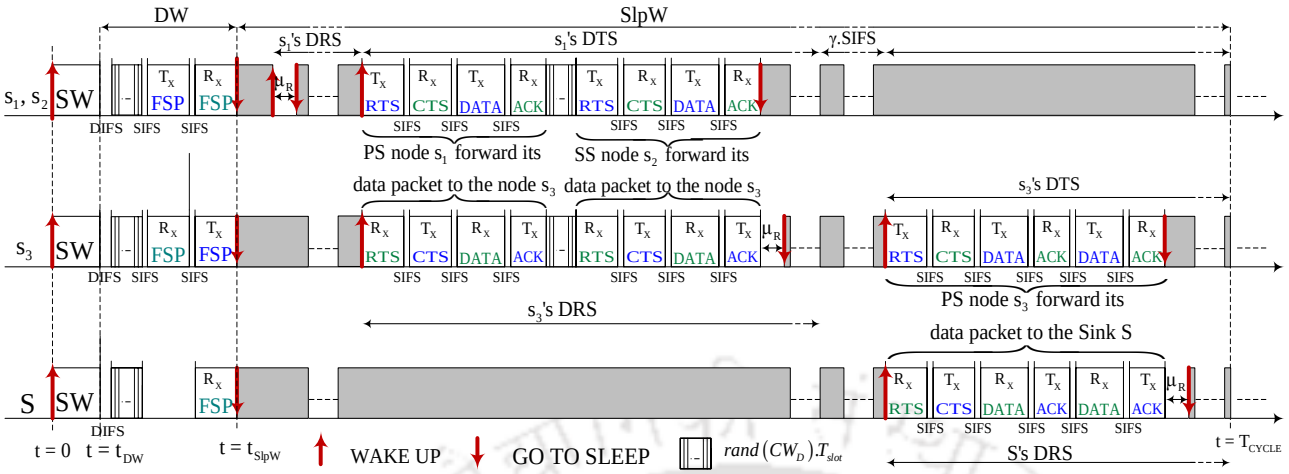


Figure 3.2: Data packet transmission process of LDC-MAC corresponding to the network scenario shown in Fig. 3.1.

DTS of a PS node overlaps with the DRS of its intended next-hop receiver node.

Following Algorithm 3.1, based on the received FSP, the SS node s_2 chooses the sensor node s_3 as its next-hop receiver and determines s_3's DRS as its DTS.

Data Transmission in SlpW: As can be seen in Fig. 3.2, the source node s_1 wakes up at the beginning of its DRS to receive data packets from the SS nodes. However, there may be a case where there is no SS to send data packets. In such a case, the receiver node goes to sleep after the μ_R seconds, from the beginning of its DRS, where $\mu_R = \text{DIFS} + rand\,(CW_D)\,.T_{slot} + T_{RTS}$. (Here, T_{RTS} denotes the transmission duration one RTS packet.)

At the beginning of s_3's DRS, the SS node s_2 sets a $\text{DIFS} + rand\,(CW_D)\,.T_{slot}$ duration timer and senses the medium while the PS node s_1 initiates RTS/CTS handshake with s_3. As shown in Fig. 3.2, after RTS/CTS handshake, s_1 sends its data packet to s_3. (Note that priority is given to the PS node to send its data packets.)

On overhearing the RTS sent by s_1 to s_3, the node s_2 cancels its timer and goes to sleep to avoid overhearing and *idle listening* during the data transmission process. It then wakes up and sets a timer for $rand\,(CW_D)\,.T_{slot}$ duration. At timer expiration, it follows the RTS/CTS handshake with s_3 and then sends its data packet. As can be seen in Fig. 3.2, a receiver node goes to sleep if it does not start to receive a packet within μ_R seconds after its last ACK.

In the DRS of S, after RTS/CTS handshake, PS node s_3 sends both its data packets to S. In this way, in one cycle, data packets of both the nodes are received by S. In contrast, in the other existing

cross-layer *generic* contention based synchronous MAC protocols, such as [42–55] etc., at least two cycles will be required for this (one cycle each for s_1 and s_2).

3.3 Results

In this section, we use the ns-2.35 simulator to compare LDC-MAC and CL-MAC [55], in terms of their PDR, average E2ETD (AE2ETD), average energy consumption (AEC), and network life. For this, we construct a network by randomly deploying n sensor nodes (s_1, $s_2, \ldots,$ s_n) in a uniform fashion, to monitor the events in an $A \times A$ meter^2 area. In this network, an event is detected by a cluster of M sensor nodes (s_1, $s_2, \ldots, s_M$). This network contains t sink nodes and i^{th} ($i \in [1, t]$) sink node is denoted by S_i. In case of $t = 1$, S_1 is located at $(A/2, A/2)$ while, for $t = 2$, S_1 and S_2 are located at $(0, A)$ and $(A, 0)$, respectively. All the sensor nodes and the sink node(s) are stationary. Each node is assumed to have one omnidirectional antenna and uses the two-ray ground reflection radio propagation model. As in CL-MAC [55], we also assume that the nodes follow a single sleep/wakeup schedule and routing information is available to them corresponding to the shortest path.

Performance is analyzed in two different scenarios. In the first scenario, we set $n = 900$, $A = 1800$, and $t = 1$. In this scenario, a cluster of $M = 6$ sensor nodes is randomly chosen from the deployed 900 sensor nodes as the source nodes, and performance is analyzed by varying the packet arrival interval (PAI) at the source nodes. In the second scenario, we set $n = 225$, $A = 2400$, and $t = 2$. In this scenario, two clusters, each having $M = 4$ sensor nodes, are randomly chosen from the deployed 225 sensor nodes as the source nodes. The nodes of one cluster forward their data packets to the sink node S_1 and the nodes of the other cluster forward their data packets to the sink node S_2. Performance is analyzed by varying the packet arrival interval (PAI) at the source nodes.

To measure PDR and AE2ETD, each source node generates data packets during the entire simulation time, with the given PAI. To measure the AEC, in Figs. 3.9 and 3.10, each source node generates only 30 data packets during the entire simulation time (ST), with the given PAI. On the other hand, in Fig. 3.11, AEC is measured by allowing each source node to generates data packets during the entire ST, with the given PAI. The results shown in the Figs. 3.3, 3.4, 3.6, 3.7, 3.9, and 3.10 are obtained by taking average over 40 simulations with different seeds and the duration of each simulation is kept at 600 seconds. 95 % confidence intervals are also calculated and shown in the figures by error bars. On the other hand, the results shown in the Figs. 3.5, 3.8, 3.11, and 3.12 are obtained by taking average

over 60 simulations and duration of each simulation is kept at 2000s. Networking parameters are the same as those in RMAC [42], as shown in Table 3.1. Frame sizes are same as those in CL-MAC [55] and shown in Table 3.2. The durations of SW, DW, and SlpW are equal to 52 ms, 100 ms, and 14.844 s, respectively.

Table 3.1: Networking Parameters

Bandwidth	20 Kbps	Communication Range	250m
Idle Power	0.45 W	Carrier Sensing Range	550 m
Sleep Power	0.05 W	DIFS	10 ms
Tx Power	0.5 W	SIFS	5 ms
Rx Power	0.5 W	Slots in CW of DW	64
T_{slot}	1 ms	Slots in CW of SW	31

Table 3.2: Frame Sizes

Frame Type	Size	Frame Type	Size
FSP/FSP_1	12 bytes	RTS/CTS/SYNC	9 bytes
DATA	50 bytes	EACK/ACK	10 bytes

3.3.1 PDR Comparison

Fig. 3.3 shows comparison of the PDR of LDC-MAC and CL-MAC when $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. Similarly, Fig. 3.4 shows comparison of the PDR of LDC-MAC and CL-MAC in case of $A = 2400\ m$, $n = 225$, $t = 2$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. PDR is defined as the ratio of total number of packets successfully received at the sink node(s) to the total number of packets generated at the source nodes during the simulation time. In each case, the PDR of LDC-MAC is more than that of CL-MAC. The reason for this is that in LDC-MAC, in its DRS, a node can receive data packets from both the PS node and the SS nodes. On the other hand, in CL-MAC, in its DRS, a node can receive data packets only from the PS node. Consequently, in LDC-MAC, the sink node receives more data packets per cycle than CL-MAC. It may be noted that, with an increase in PAI, the difference in PDRs increases. This happens because an increase in PAI reduces the number of data packets generated by a source node in one cycle duration. As a result of this, a PS node can receive more number of data packets from the SS nodes in its DRS.

In Fig. 3.3 and 3.4, at PAI = 6.0 s, PDR of LDC-MAC is almost 14.0% and 6.0%, respectively, more than that of CL-MAC.

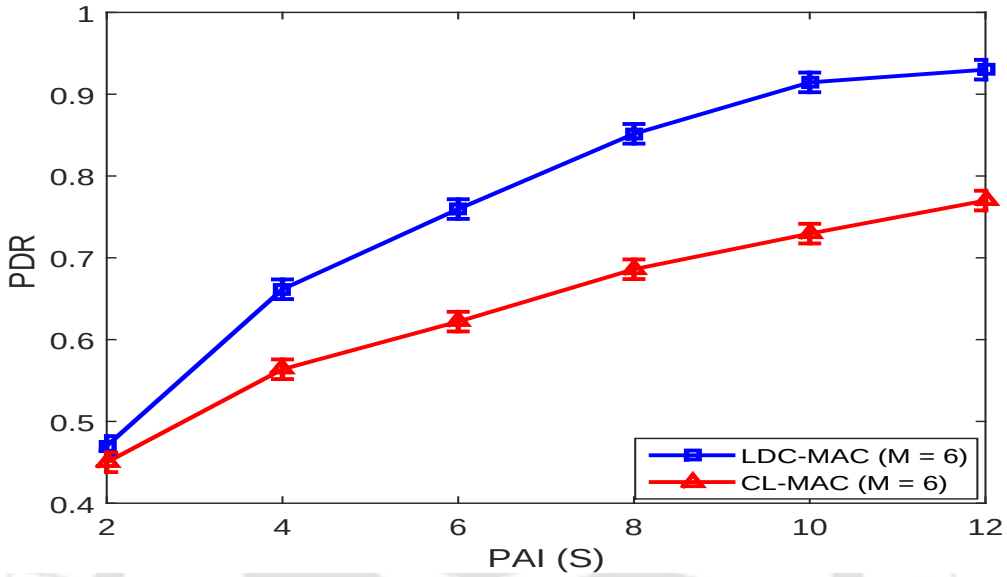


Figure 3.3: Comparison of PDR in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$.

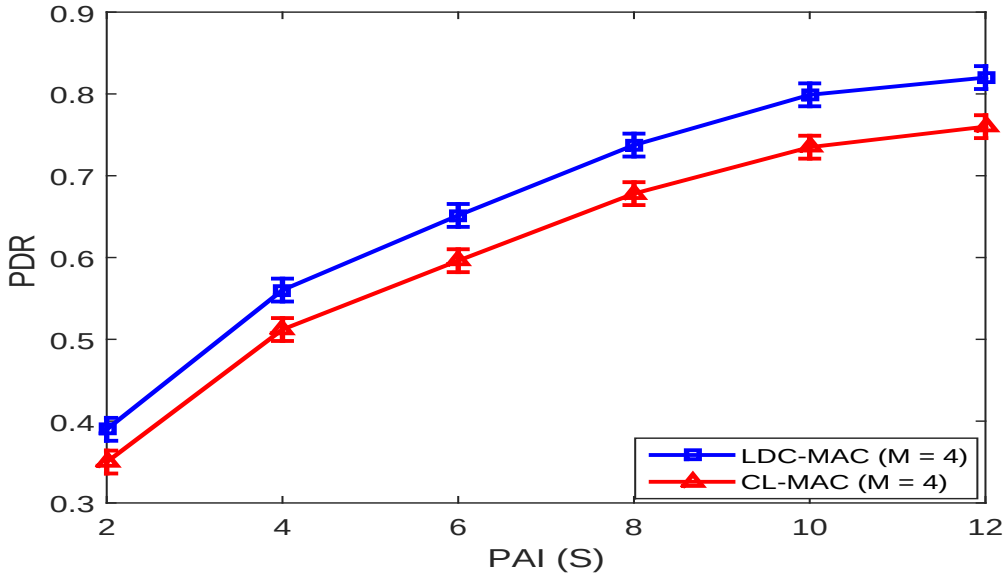


Figure 3.4: Comparison of PDR in case of $A = 2400\ m$, $n = 225$, $t = 2$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$.

Further, from Figs. 3.3 and 3.5, it can be observed that the reduction in the CR of nodes (or, equivalently, node density) results in the reduction in the difference in PDR of LDC-MAC and CL-

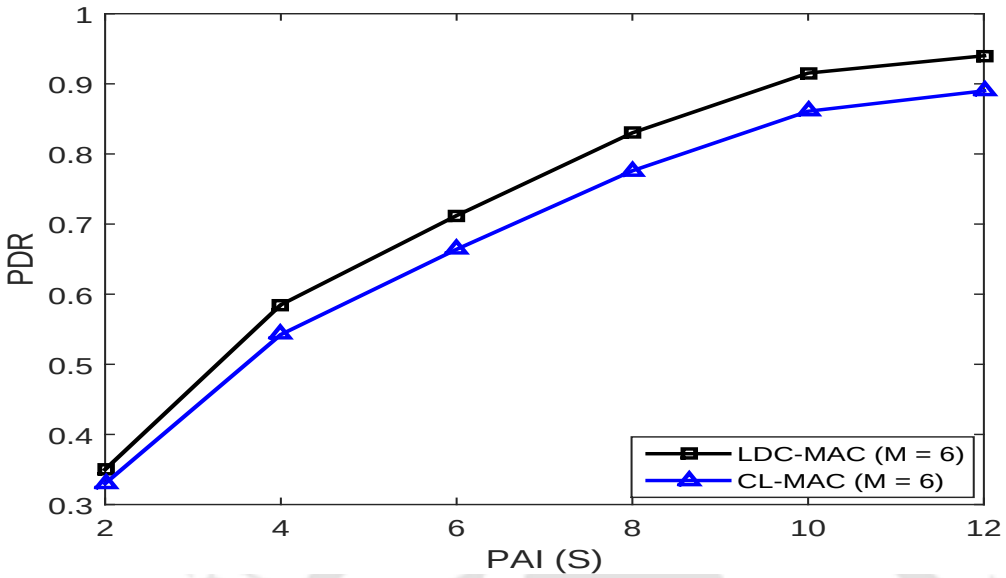


Figure 3.5: Comparison of PDR in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 100\ m$, $CSR = 200\ m$, and $ST = 2000\ s$.

MAC. In Fig. 3.5, at PAI = 6.0 s, PDR of LDC-MAC is almost 8.0% more than that of CL-MAC. It happens because the additional features of LDC-MAC can help in improving the E2ETD and PDR only when the sensor nodes having data packets to send to the same final destination lie in the CR of each other. For a given node density, with the reduction in CR, the number of nodes that lie in the CR of a sensor node reduces. As a result of this, probability that the two or more nodes having data packets to send to the same final destination lie in the CR of each other also reduces. Therefore, the improvement in the performance of LDC-MAC over CL-MAC reduces with a reduction in the CR.

3.3.2 AE2ETD Comparison

Fig. 3.6 shows comparison of the AE2ETD when $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. Similarly, Fig. 3.7 shows comparison of the AE2ETD in case of $A = 2400\ m$, $n = 225$, $t = 2$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. The E2ETD of a data packet is equal to the time difference between the time instant when it is received at the sink and the time instant when it is generated at the source. AE2ETD is the average of the E2ETD of all the data packets received at the sink during the simulation run. We find that in each case, the AE2ETD of LDC-MAC is less than that of CL-MAC. The reason for this once again is that in LDC-MAC, a node can send data packets both as a PS node and as a SS node. In contrast, in CL-MAC, a node can

send data packets only as a PS node. It may be noted that, with an increase in PAI, the difference in AE2ETDs increases. This happens because an increase in PAI increases the number of data packets a PS node can receive from the SS nodes in its DRS. In Fig. 3.6 and 3.7, at PAI = 6.0, AE2ETD of LDC-MAC is almost 57.0% and 21.0%, respectively, less than that of CL-MAC. Further, from

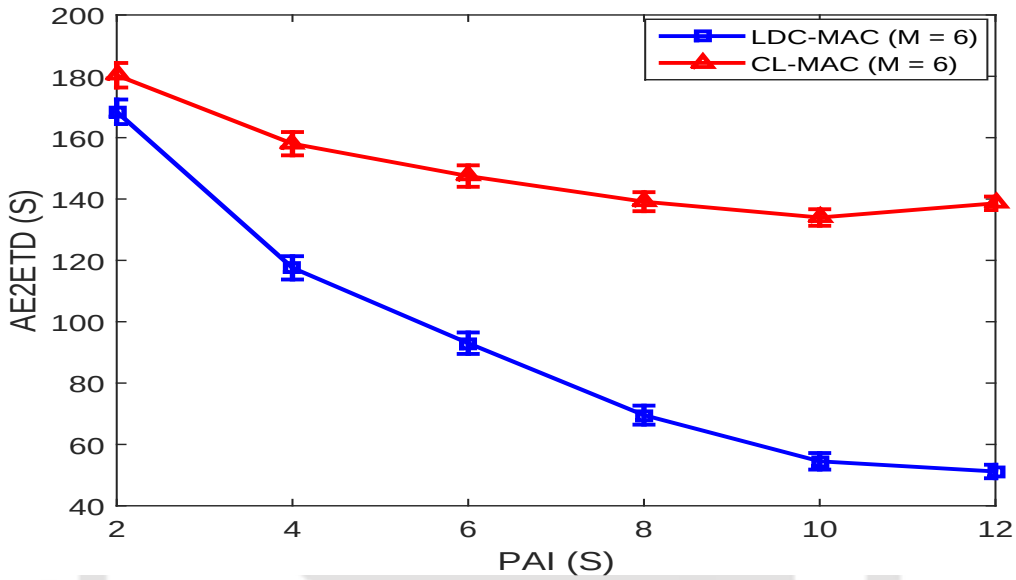


Figure 3.6: Comparison of AE2ETD in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$.

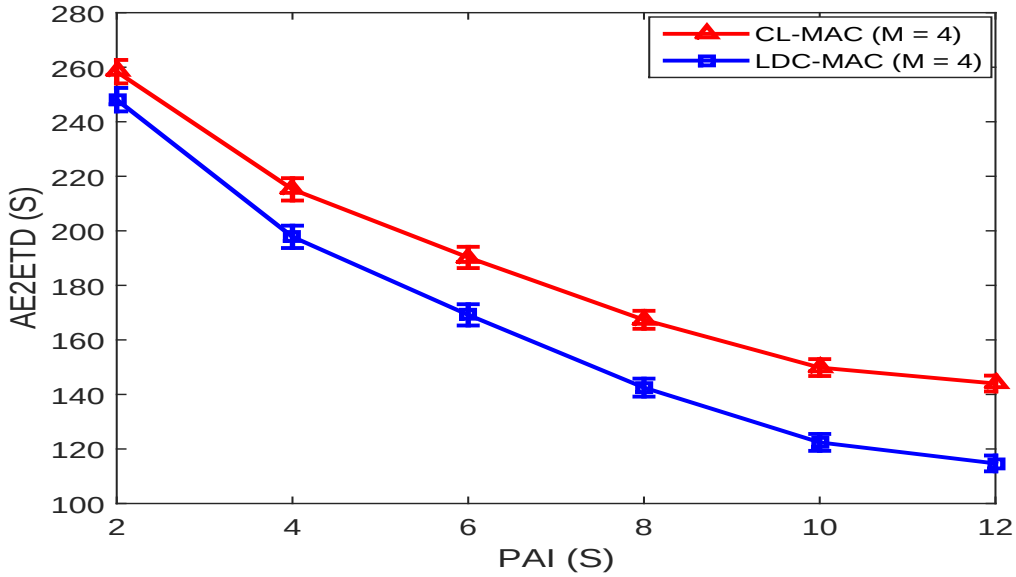


Figure 3.7: Comparison of AE2ETD in case of $A = 2400\ m$, $n = 225$, $t = 2$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$.

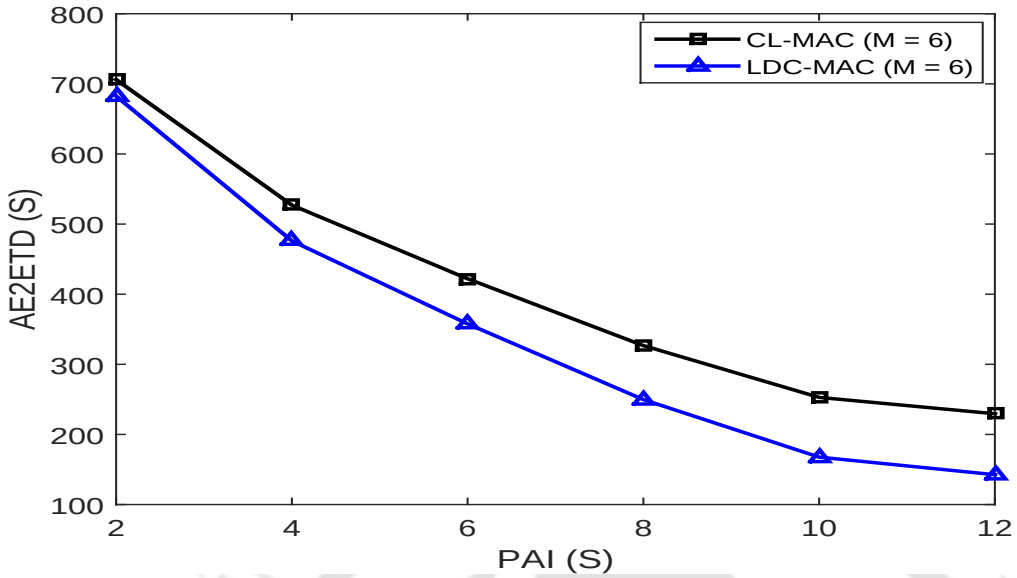


Figure 3.8: Comparison of AE2ETD in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 100\ m$, $CSR = 200\ m$, and $ST = 2000\ s$.

Figs. 3.6 and 3.8, it can be observed that reduction in the CR of nodes results in the decrease in the difference between the AE2ETD of LDC-MAC and CL-MAC. In Fig. 3.8, at PAI = 6.0, AE2ETD of LDC-MAC is almost 31.0% lesser than that of CL-MAC. It happens because, with a reduction in the CR, the number of nodes lying in the CR of a node reduces.

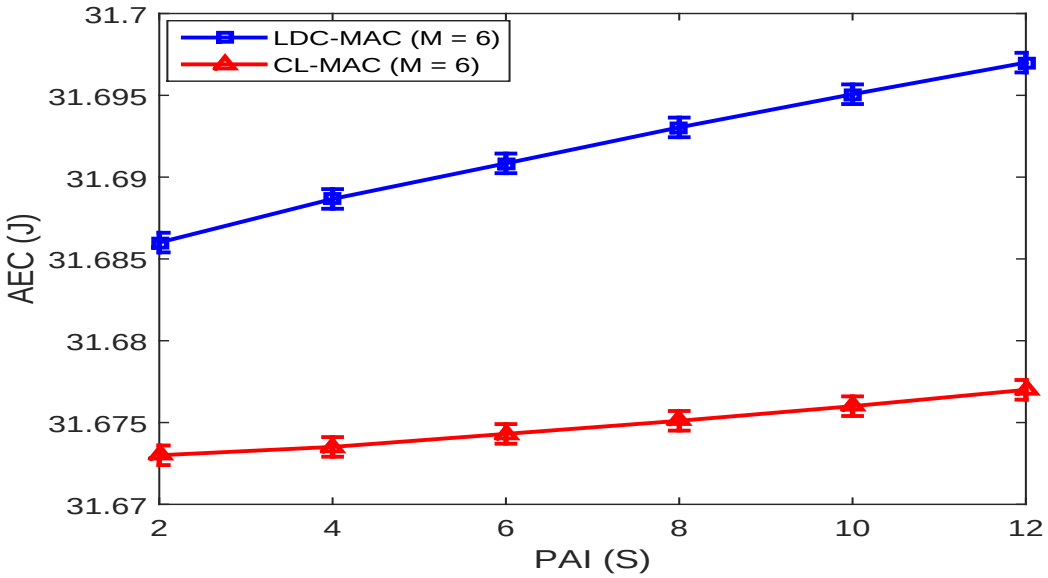


Figure 3.9: Comparison of AEC in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$.

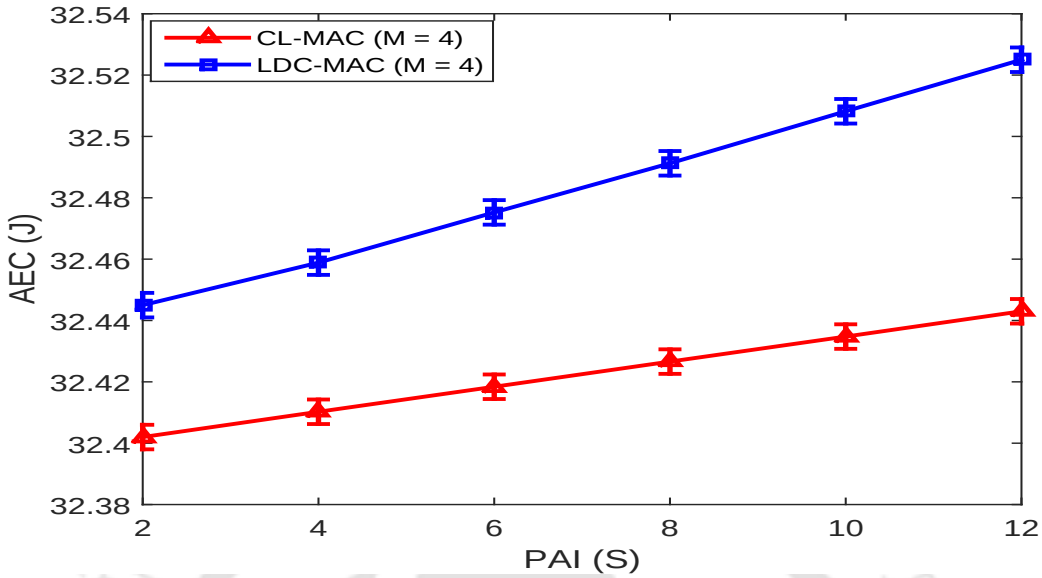


Figure 3.10: Comparison of AEC in case of $A = 2400\ m$, $n = 225$, $t = 2$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$.

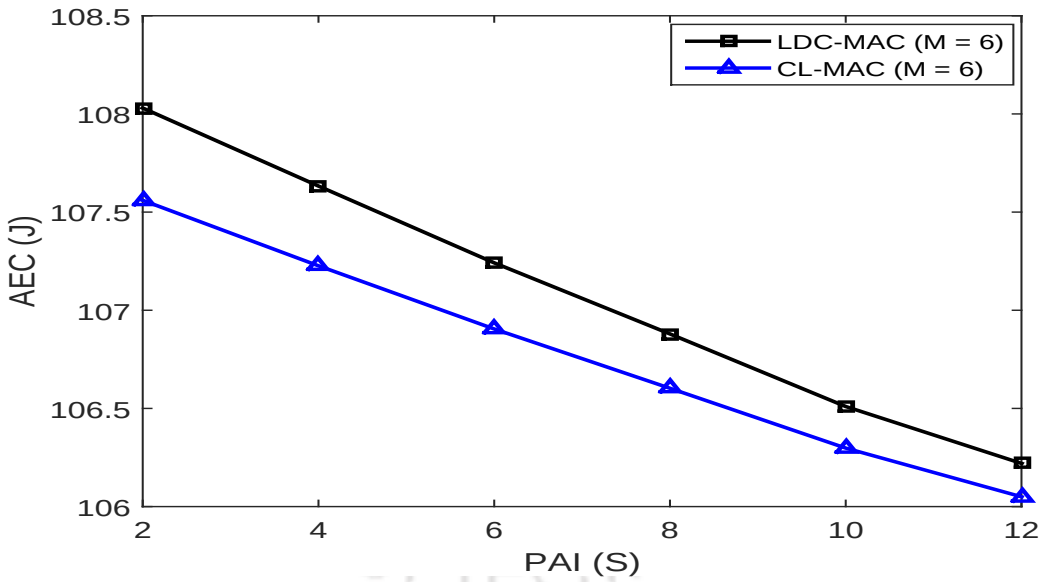


Figure 3.11: Comparison of AEC in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 100\ m$, $CSR = 200\ m$, and $ST = 2000\ s$.

3.3.3 AEC Comparison

Fig. 3.9 shows comparison of the AEC when $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. Similarly, Fig. 3.10 shows comparison of the AEC in case of $A = 2400\ m$, $n = 225$, $t = 2$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 600\ s$. AEC is defined

as the total energy consumed during the simulation divided by the total number of sensor nodes in the network. In each case, there is a very small ($< 0.3\%$) difference in the AEC of LDC-MAC and CL-MAC. A possible reason for this is the RTS/CTS handshake process that is followed in LDC-MAC by a PS and a SS before data packet transmission. It may be noted that, with an increase in PAI, AEC of each protocol increases. The reason is that an increase in PAI, increases the transmission and/or reception of control packet (FSPs) in DW and RTS/CTS handshakes in SlpW.

Fig. 3.11 shows comparison of the AEC when $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 2000\ s$. As in the above two cases, from Fig. 3.11, it can be noted that there is a very small ($< 0.3\%$) difference in the AEC of LDC-MAC and CL-MAC.

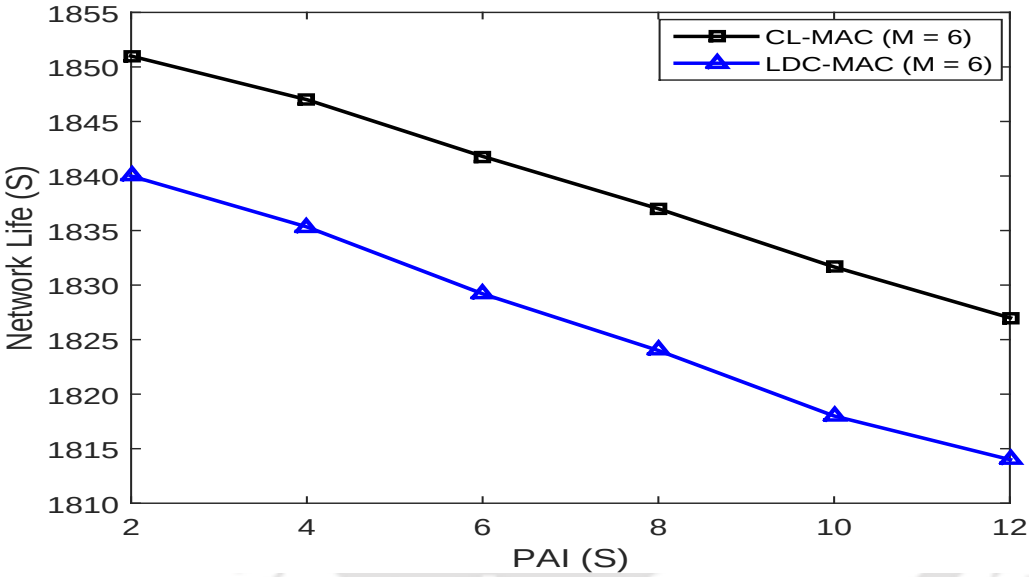


Figure 3.12: Comparison of network life in case of $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 100\ m$, $CSR = 200\ m$, and $ST = 2000\ s$.

3.3.4 Network Life Comparison

Fig. 3.12 shows comparison of the network life when $A = 1800\ m$, $n = 900$, $t = 1$, $CR = 250\ m$, $CSR = 550\ m$, and $ST = 2000\ s$. The network life time is defined as the life of the sensor node which runs out of energy at the earliest. For network life comparison, we randomly choose a cluster of $M = 6$ sensor nodes as source nodes. For the sake of fair comparison of network life, we restrict each source node to generate only 100 data packets during the entire simulation time (ST), with the given PAI. In our simulations, we consider that the initial energy of each sensor node is 110 J. In Fig. 3.12, results are averaged over 60 simulations with different seeds and duration of each simulation is

kept at 2000s.

From Fig. 3.12, it can be noted that the network life in case of CL-MAC is slightly more than that of LDC-MAC. It happens because, in LDC-MAC, a PS node can receive data packets from the PS node and/or SS nodes in the same cycle. As a result of this, in LDC-MAC, the nodes behaving as the PS nodes consumes more energy than that consumed by the same node in case of CL-MAC. In Fig. 3.12, at PAI = 6.0s, the network life in case of CL-MAC is almost 0.7% more than that of LDC-MAC which is insignificant in comparison to the gains in terms of E2ETD and PDR. Moreover, in a densely deployed WSN, dying of a few nodes may not affect network connectivity at all.

3.4 Conclusions

This chapter presented a low delay cross-layer MAC (LDC-MAC) protocol for *k-covered* event driven WSNs. LDC-MAC assigns DRS to each node that either sends data transmission request (FSP) or receives FSP as an intended next-hop receiver node, in its DW. It enables a node, which failed in data packet transmission flow setup in DW, to forward its data packets utilizing the unused portion of its one-hop neighbor's DRS. These features increase the number of nodes that can forward their data packet in the same cycle when multiple nodes, with data packet to send, lie in the CR of each other. As a result, both E2ETD and PDR are better than those of other existing cross-layer *generic* contention based synchronous MAC protocols, at the cost of a very marginal increase in the AEC.

4

A Novel Framework to Enhance the Performance of *Generic* Contention Based Synchronous MAC Protocols in Event-Driven WSNs

Contents

4.1	Introduction	56
4.2	Description of the Proposed Framework	57
4.3	Determination of Optimum m	64
4.4	Results	70
4.5	Conclusions	79

In this chapter[1], we propose a framework which improves the E2ETD performance of existing *generic* contention based synchronous MAC protocols by exploiting the dense deployment of sensor nodes in the network. Our framework improves the E2ETD performance of the implemented MAC protocol by increasing the number of hops a data packet can travel in a cycle and increasing the number of nodes that can forward their data packets in the same cycle when multiple nodes with data packets to send lie in the CR and/or CSR of each other. For this, in the propose framework, sensor nodes are partitioned into m disjoint sets (DSs) and the cycle structure followed by the network nodes in existing contention based synchronous MAC protocols is modified such that the modified structure contains $m-1$ more pairs of DW and SlpW. Sleep/wakeup schedules to the sensor nodes are assigned such that the propose framework maps the m cycles of the data transmission process of the implemented MAC protocol in a single duration cycle. An analytical technique is also proposed to determine the m for the optimum E2ETD in a given network scenario.

4.1 Introduction

As mentioned in Chapter 1, for better quality of service (or reliability), the sensor nodes in a WSN should be deployed in a way such that each location of the monitored region lies in the sensing range of at least three sensor nodes. In such a WSN, an event is detected by the multiple sensor nodes that generally lie in the CR and/or CSR of each other. The E2ETD performance of a cross-layer *generic* contention based synchronous MAC, in a dense multi-hop WSN, mainly depends on the following two factors - (1) the number of nodes that can forward their data packets in the same cycle when multiple nodes with data packets to send lie in the CR and/or CSR of each other, and (2) the number of hops that a data packet can travel in one cycle.

The cross-layer *generic* contention based synchronous MAC protocol LDC-MAC, presented in Chapter 3, reduces the E2ETD by increasing the number of nodes that can forward their data packets in the same cycle when multiple nodes with data packets to send lie in the CR of each other. However, as in other existing cross-layer *generic* contention based synchronous MACs, in LDC-MAC, a data packet can not travel more than $\lceil (T_{\mathrm{DW}} - \mathrm{DIFS}) \,/\, (T_{\mathrm{C}} + \mathrm{SIFS}) \rceil + 2$ hops in a cycle where C denotes the of the control packet used for sending data transmission request in the DW (e.g., PION [42], FSP [55] etc.) and T_C denotes the transmission duration of this control packet. (Recall that T_{DW}

[1]The work reported in this chapter has been presented in the following journal publication:
Ripudaman Singh, Brijesh K. Rai, and Sanjay K. Bose, "A novel framework to enhance the performance of contention-based synchronous MAC protocols," *IEEE Sens. J.*, vol. 16, no. 16, pp. 6447-6457, Aug. 2016.

denotes the duration of one DW.) This constraint limits the E2ETD performance of a cross-layer *generic* contention based synchronous MAC in a multi-hop scenario. The main reasons for this are as follows - (1) the cycle structure followed by the network nodes contains only one pair of DW and SlpW; and (2) all the network nodes follow a single sleep/wakeup schedule (i.e., all the neighbors of a sensor node follow same sleep/wakeup schedule).

In this chapter, we propose a framework to overcome the effect of above shortcomings on E2ETD performance of a cross-layer *generic* contention based synchronous MAC protocol. In this framework, we partition the network nodes into m disjoint sets (DSs) of almost equal size and modify the cycle structure followed by the existing *generic* contention based synchronous MAC protocols. In the modified cycle structure, cycle duration is divided into m equal segments and one SW, where each segment is further subdivided into DW and SlpW. Compared to the existing cycle structure, our proposed modified structure contains $m-1$ more pairs of DW and SlpW, in the same duration cycle. We keep all the nodes of i^{th} $(1 \leq i \leq m)$ DS awake during SW and DW of i^{th} segment in a cycle. In addition to this, we allow a node of the i^{th} DS to wakeup during the DW of j^{th} segment, where $j \neq i$, only when the node has data packets to send. With this, a node gets m chances to forward its data packets in a cycle, which leads to reduction in E2ETD. In our proposed framework, for data packet forwarding, in each segment, every node follows the data transmission scheduling process of the MAC protocol implemented in the framework. The optimum performance of our proposed framework depends on the optimum value of m, which in turn mainly depends on the density of nodes in the network. In a dense network, which is usually the case in WSNs, this approach significantly improves E2ETD and PDR performance of a *generic* contention based synchronous MAC protocol.

4.2 Description of the Proposed Framework

In this section we describe our proposed framework in detail. We start with the Setup Phase, where we describe the process adopted for partitioning the sensor nodes into m equal sized DSs. This is followed by the data transmission phase where we discuss the modified cycle structure and the sleep/wakeup schedule followed by the sensor nodes of the i^{th} $(1 \leq i \leq m)$ DS. We then describe the data transmission process and illustrate it with an example.

4.2.1 Setup Phase

The objective of this phase is to assign a disjoint set index (DSI), from $[1, m]$, to each sensor node deployed in the network area, such that (1) the number of sensor nodes corresponding to each DS is almost the same and (2) the node density corresponding to each DS is almost the same throughout the network area. It is assumed that before the start of the Setup Phase, the network has the following status.

(i) n sensor nodes (s_1, $s_2, \ldots,$ s_n) are randomly deployed, in a uniform fashion, in the network area. The network contains t sink nodes.

(ii) A unique address is assigned to each sink node and each sensor node from $[1, t]$ and $[t+1, t+n]$, respectively.

(iii) DSI of each sink node and each sensor node are set to 0 and 1, respectively.

(iv) All the nodes follow the cycle structure shown in Fig. 4.1 with the same sleep/wakeup schedule. In Fig. 4.1, cycle duration is divided into two windows, named as, SW and SlpW. In each cycle, every node wakes up at the beginning of its SW and goes to sleep at the beginning of its SlpW.

(v) Synchronization is achieved and maintained by using a flooding based synchronization protocol [72–76]. For this, (1) one of the sink nodes behaves as the *reference node*; (2) each node periodically (at the interval of N_{SC} cycles) broadcasts a synchronization (SYNC) packet in its SW; and (3) each node periodically updates its sleep/wakeup schedule based on the recently received SYNC packets. Along with the information required to maintain synchronization by the used synchronization protocol, in our proposed framework a SYNC packet also contains current DSI of the sender.

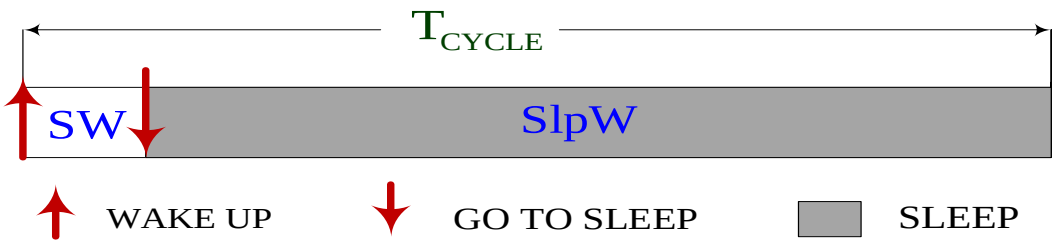


Figure 4.1: Cycle structure followed by the network nodes in Setup Phase.

In the Setup Phase, at the time of SYNC broadcast and after receiving a SYNC packet from a sensor node, each sensor node follows Algorithm 4.1 and Algorithm 4.2, respectively, to update its Lookup Table (LT). The objective of Algorithm 4.1 and Algorithm 4.2 is to maintain the same number of nodes corresponding to each DS in the neighborhood of each sensor node. This partitions the sensor nodes into m DSs of almost equal size. Moreover, it leads to an almost uniform node density corresponding to each DS. For $i = 1,\ 2, \ldots,\ n$, LT of the sensor node s_i has the following information.

(i) Address and current DSI of the sensor nodes from which it is periodically receiving SYNC packets.

(ii) Current value of node_DS$_j^{s_i}$ (total number of nodes in LT of s_i that have $DSI = j$) for $j = 1,\ 2, \ldots, m$.

In Algorithm 4.1, the sender node (s_i) first determines the DS from $[1, m]$ that have minimum number of nodes in its LT. It then checks its current DSI and changes it such that total number of nodes corresponding to each DS become almost the same in its LT. After this, it broadcasts the SYNC packet containing its address, current DSI, and the timing information required to maintain synchronization by the synchronization protocol that is used.

Algorithm 4.1 Process followed by the sensor node s_i at the time of SYNC packet broadcast.

node_DS$_j^{s_i}$: total number of nodes in LT of s_i that have $DSI = j$
DS$_{\min}^{s_i}$: DS having minimum number of nodes in the LT s_i
DSI $[s_i]\,[s_t]$: current DSI of node s_t in the LT of node s_i
1: Determine DS$_{\min}^{s_i}$
2: **if** more than one DSs are eligible for DS$_{\min}^{s_i}$ **then**
3: randomly choose one of them as DS$_{\min}^{s_i}$
4: **end if**
%Check to change the current DSI%
5: **if** DSI$[s_i][s_i] \neq$ DS$_{\min}^{s_i}$ AND
node_DS$_{\text{DSI}[s_i][s_i]}^{s_i}$ − node_DS$_{\text{DS}_{\min}^{s_i}}^{s_i} \geq 2$ **then**
6: node_DS$_{\text{DSI}[s_i][s_i]}^{s_i} \leftarrow$ node_DS$_{\text{DSI}[s_i][s_i]}^{s_i} - 1$
7: DSI$[s_i][s_i] \leftarrow$ DS$_{\min}^{s_i}$
8: node_DS$_{\text{DSI}[s_i][s_i]}^{s_i} \leftarrow$ node_DS$_{\text{DSI}[s_i][s_i]}^{s_i} + 1$
9: **end if**

In Algorithm 4.2, at each SYNC packet reception, a sensor node (s_t) first checks whether the sender node (s_i) is a newly deployed node or an existing node. In case of a newly deployed node, it adds the sender in its LT and increases the total number of nodes corresponding to the sender's DSI by 1. In case it is an existing node, it checks whether the sender has changed its DSI after the previous SYNC

Algorithm 4.2 Process followed by the sensor node s_t after receiving SYNC packet broadcast by s_i.

$\text{node\_DS}_j^{s_t}$: total number of nodes in LT of s_t that have $DSI = j$
$\text{DSI}\,[s_t]\,[s_i]$: current DSI of node s_i in the LT of node s_t
$\text{DSI}_{\text{SYNC}}^{s_i}$: DSI of node s_i in its most recently broadcasted SYNC packet
1: **if** $s_i \notin$ LT **then**
2: $\quad \text{DSI}[s_t][s_i] \leftarrow \text{DSI}_{\text{SYNC}}^{s_i}$
3: $\quad \text{node\_DS}_{\text{DSI}[s_t][s_i]}^{s_t} \leftarrow \text{node\_DS}_{\text{DSI}[s_t][s_i]}^{s_t} + 1$
4: **end if**
5: **if** $s_i \in$ LT **then**
6: $\quad$ **if** $\text{DSI}_{\text{SYNC}}^{s_i} \neq \text{DSI}[s_t][s_i]$ **then**
7: $\quad\quad \text{node\_DS}_{\text{DSI}[s_t][s_i]}^{s_t} \leftarrow \text{node\_DS}_{\text{DSI}[s_t][s_i]}^{s_t} - 1$
8: $\quad\quad \text{DSI}[s_t][s_i] \leftarrow \text{DSI}_{\text{SYNC}}^{s_i}$
9: $\quad\quad \text{node\_DS}_{\text{DSI}[s_t][s_i]}^{s_t} \leftarrow \text{node\_DS}_{\text{DSI}[s_t][s_i]}^{s_t} + 1$
10: $\quad$ **end if**
11: **end if**

broadcast or not. If there is a change, then it reduces the total number of nodes corresponding to the current DSI by 1, updates the current DSI of the sender, and increases the total number of nodes corresponding to the updated DSI by 1. Each node remains in Setup Phase for λ cycles duration.

To analyze the impact of the pre-defined parameter λ on the maximum deviation in the size of a DS from the mean value, we consider a network in which 225 sensor nodes are randomly deployed in a uniform fashion in an 2400×2400 meter^2 area and a sink is placed at $(0, 0)$. We performed extensive simulations for $\lambda \in [300, 1000]$, $m \in [2, 6]$ and $N_{SC} = 50$, using the ns-2.35 simulator. In these simulations we use the networking parameters given in Table 4.2. For $\lambda = 300$, it is observed that the maximum deviation in the size of a DS from the mean value, for each m, is less than 6.0% of the mean value. Moreover, for $\lambda \in [600, 1000]$, this deviation becomes almost constant and its value is less than 5.0% of the mean value, for each m.

Fig. 4.2 shows the performance of Algorithm 4.1 and Algorithm 4.2 in case of $\lambda = 600$ cycles. It can be noticed that the sizes of all the DSs are almost equal corresponding to each m. Moreover, for each m, the maximum deviation in the size of a DS from the mean value is less than 5.0% of the mean value.

4.2.2 Data Transmission Phase

At the end of the Setup Phase $\left(\lambda^{\text{th}} \text{ cycle}\right)$, each node restructures its current cycle structure, from Fig. 4.1 to Fig. 4.3 (a), and enters its data transmission phase (DTP). As can be seen in Fig. 4.3 (a), cycle duration (T_{CYCLE}) is divided into m equal time segments (TSs) and one SW, and each

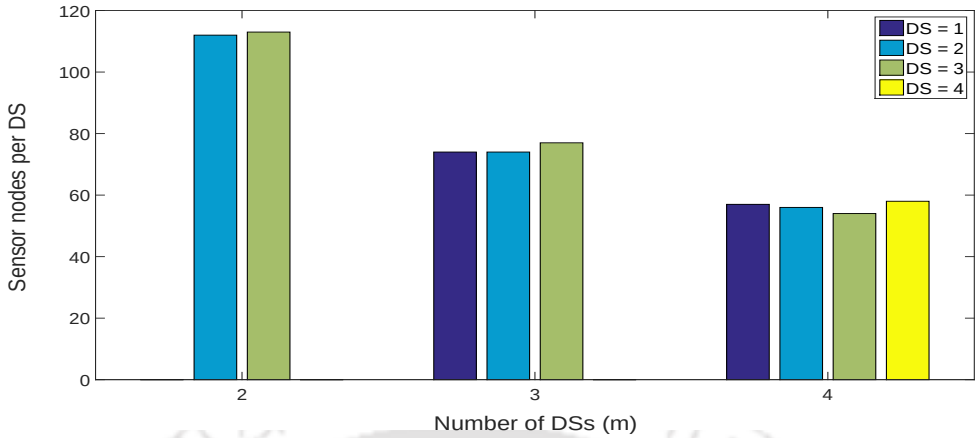


Figure 4.2: Partition of sensor nodes into DSs.

TS is further subdivided in DW and SlpW. Note that for $m = 1$, the proposed cycle structure (Fig. 4.3 (a)) is identical to that followed by existing *generic* contention based synchronous MAC protocols (Fig. 4.3 (b)). In Fig. 4.3 (a), notation TS_i is used to represent the i^{th} $(1 \leq i \leq m)$ TS. Each sensor node that have $DSI = i$, is identified as *primary member* in TS_i and remains awake during the DW of TS_i in each cycle. On the other hand, each node that have $DSI = j$, is identified as *secondary member* in TS_i if $j \neq i$. A sensor node wakes up as a *secondary member* in a DW only when it has data packet to send. Unlike the sensor nodes, all the sink nodes remains awake during the DW of each TS. A sensor/sink node wakes up in a SlpW only to transmit and/or receive data packets. As in Setup Phase, in SW, all the network nodes remain awake and each node broadcasts SYNC once every N_{SC} cycles, containing the information required to maintain synchronization by the synchronization protocol used, and the address and current DSI of the sender. It is to be noted that, unlike in the Setup Phase, in DTP, a sensor node neither follow Algorithm 4.1 at the time of SYNC broadcast nor follow Algorithm 4.2 after receiving a SYNC packet from a sensor node.

Whether it is a *primary member* or *secondary member*, any node with data packets to send, tries to contend for the medium at the beginning of DW to schedule its data packet transmission. A successful scheduling is followed by the actual data packet transmission process in the subsequent SlpW. The process followed for data packet transmission scheduling during DW and for data packet transmission during SlpW, depends on the implemented MAC protocol in the framework. Allowing a node to schedule data transmissions both as the *primary member* as well as a *secondary member* has the following advantages: 1) it ensures that the overall network connectivity remains the same for

$m > 1$ as it is for $m = 1$; and 2) a data packet can travel almost equal to m cycles of the existing structure in one restructured cycle improving both E2ETD and PDR.

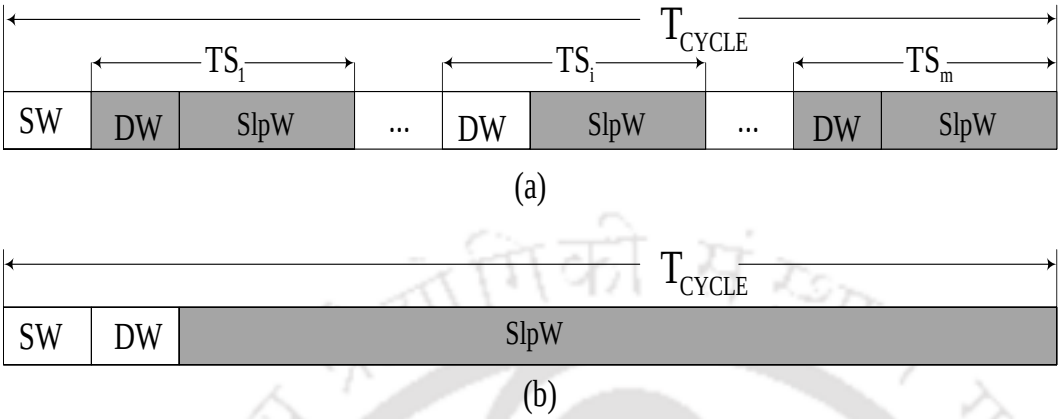


Figure 4.3: Cycle structure followed by the network nodes in (a) our proposed framework in DTP, and (b) existing *generic* contention based synchronous MAC protocols.

4.2.3 Illustration of Data Packet Transmission Process

Although all the existing *generic* contention based synchronous MAC protocols (such as, RMAC [42], PRMAC [48], CLMAC [55] etc.), can be easily implemented in our proposed framework, we illustrate the process here for the RMAC [42] implementation (shown in Fig. 4.4). For this, we consider a network where n sensor nodes (s_1, $s_2, \ldots,$ s_n) are randomly deployed in a uniform fashion in the network area. The network contains a sink node S. In Fig. 4.4, data transmission process is illustrated with the following assumptions (1) the sensor nodes are partitioned into two almost equal size DSs (i.e., $m = 2$); (2) sensor node density corresponding to each DS is almost same throughout the network area; (3) each sensor node is aware about its next-hop receiver node corresponding to every DS; (4) a sensor node can schedule at the most two hops forwarding of its data packet in one DW; (5) only the sensor node s_1 has a data packet to send to the sink node S and it lies at four hops distance from S. We use the notation $s_i\,(j, l)$ to indicate that node s_i belongs to j^{th} DS and is located at l hops away from S.

At the beginning of DW of TS_1, the node $s_1\,(1, 4)$ tries to contend for the medium. After successful medium contention, it schedules two-hop forwarding of its data packet. Corresponding to this scheduling, at the beginning of SlpW of TS_1, data packet travels from $s_1\,(1, 4)$ to $s_2\,(1, 3)$, and then from $s_2\,(1, 3)$ to the last node in the scheduled flow (i.e., $s_3\,(1, 2)$). Note that $s_3\,(1, 2)$ belongs to DS = 1, therefore, at the beginning of DW of TS_2, it wakes up as a *secondary member* and schedules

the forwarding of received data packet to the sink S. In SlpW of TS_2, the data packet arrives at the sink node S through the sensor node $s_4\,(2,1)$. (In Fig. 4.4, it may be noted that the nodes follow the scheduling process of the protocol implemented in the framework, i.e. RMAC.)

In this way, the data packet travels from s_1 to S in two TSs (i.e., within the cycle duration in this scenario). In contrast, for the same durations DW and $\mathrm{T_{CYCLE}}$, with the existing cycle structure (Fig. 4.3 (b)), RMAC require at least two cycles for this, as a node can schedule at the most two-hop forwarding of its data packet in one DW. In this way, our proposed framework maps the data transmission process of $m\,(=2)$ cycles of an existing *generic* contention based synchronous MAC in one cycle of the same duration.

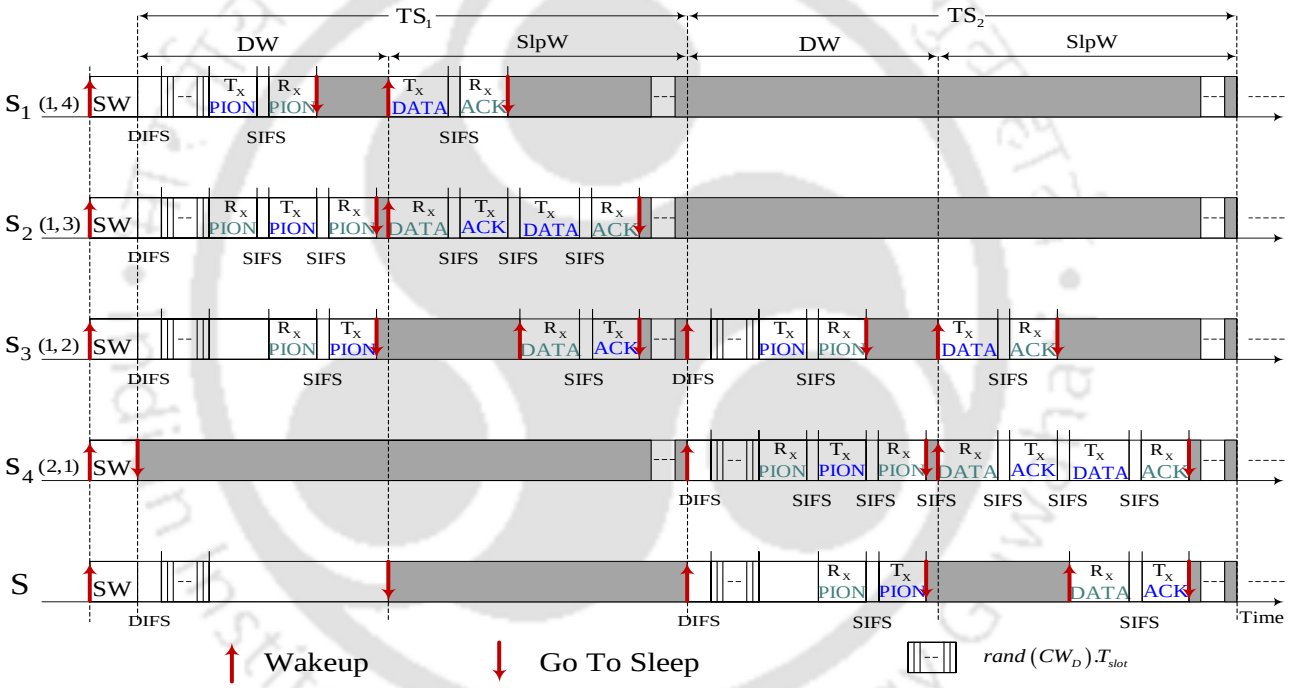


Figure 4.4: Illustration of data transmission process, in case of $m = 2$, when RMAC is implemented in the framework.

4.2.4 Addition/Deletion of Sensor Nodes in DTP

In the proposed framework, a node deployed after the setup phase may also be easily added in the system. Such a node first waits for N_{SC} cycles to receive a SYNC packet from each neighboring node. After each SYNC packet reception, node follows Algorithm 4.2 to add the sender in its LT. At the end of the N_{SC}^{th} cycle, the node chooses a DS having the minimum number of nodes as its current *DSI* and follows the sleep/wakeup schedule of the chosen DS. Similarly, expired nodes can also be

easily removed. To do this, each node removes the information of an already joined node if it doesn't receive any SYNC from it in the previous $N_{SC}.\chi$ cycles. (Here, χ is a suitably chosen, user defined positive integer.)

4.3 Determination of Optimum m

In this section, we determine the value of m which minimizes E2ETD. We term this value of m as the optimum value of m and denote it as m_{opt}. We choose E2ETD for m_{opt} determination as E2ETD would be the the most important performance metrics. Moreover, minimizing the value of E2ETD also leads to maximum PDR.

We determine m_{opt} with the following assumptions: 1) A sensor node chooses its next hop forwarder corresponding to the minimum end-to-end hops distance. (Note that this assumption corresponds to the geographic routing protocols of [93–96]. This is preferred in WSNs due to their low complexity and low energy consumption features.) 2) Sensor nodes are randomly deployed in a uniform fashion in the network area. 3) Density of sensor nodes of each of the m DSs is also the same and uniform throughout the network area. 4) In DW duration, a node can schedule average κ hop forwarding of its data packets. 5) Each node is equipped with one omnidirectional antenna and all the nodes transmit with the same transmission power (notation R is used to represent the communication radius of a node).

Recall that, when network nodes are divided into m DSs, the proposed framework can provide m chances to a sensor node to forward its data packets in each cycle. Let $AHD\,(m)$ be the average distance traveled by a data packet per hop towards its destination when the network nodes are divided into m DSs. Thus, in a cycle, a data packet can travel an average distance of $\kappa.m.AHD\,(m)$ towards its destination. This indicates that the E2ETD decreases with the increase in $\kappa.m.AHD\,(m)$. Therefore, for the given size of DW (i.e., for the fix value of κ), m which maximizes $m.AHD\,(m)$ corresponds to the minimum E2ETD.

4.3.1 Determination of $AHD\,(m)$

To determine $AHD\,(m)$, we consider a network where n $(s_1,\ s_2, \ldots,\ s_n)$ sensor nodes are randomly deployed in a uniform fashion in the network area. The network contains a sink node S. In this network, n' $(s_1,\ s_2, \ldots,\ s_{n'})$ sensor nodes are lying at more than one hop distance from S where $n' < n$. We randomly choose a sensor node s_i $(1 \leq i \leq n')$ which is at $d\,(S, s_i)$ distance from the sink S. (Note that

the notation $d(S, s_l)$ is used to represent the distance between the sensor node s_l $(1 \leq l \leq n)$ and sink node S.) For the sake of simplicity, we consider that the sensor node s_i is located at O $(0,0)$ and the sink S is located at P $(0,0)$, as shown in Fig. 4.5. In Fig. 4.5, C_1 represents the circle centered at O with radius R and C_2 represents the circle centered at P with radius $d(S, s_i) - h$, where h is a variable used for varying the radius of circle C_2. Let A_1 and A_2 be the areas of C_1 and C_2, respectively; and the notation $A(x_1, x_2)$ represents the portion of $A_1 \cap A_2$ lying between the $x = x_1$ and $x = x_2$ lines. In Fig. 4.5, C_1 and C_2 intersect at $x = \alpha$ (say) line, where α can be shown to be,

$$\alpha = \frac{R^2 - h^2 + 2d(S, s_i)h}{2d(S, s_i)}. \tag{4.1}$$

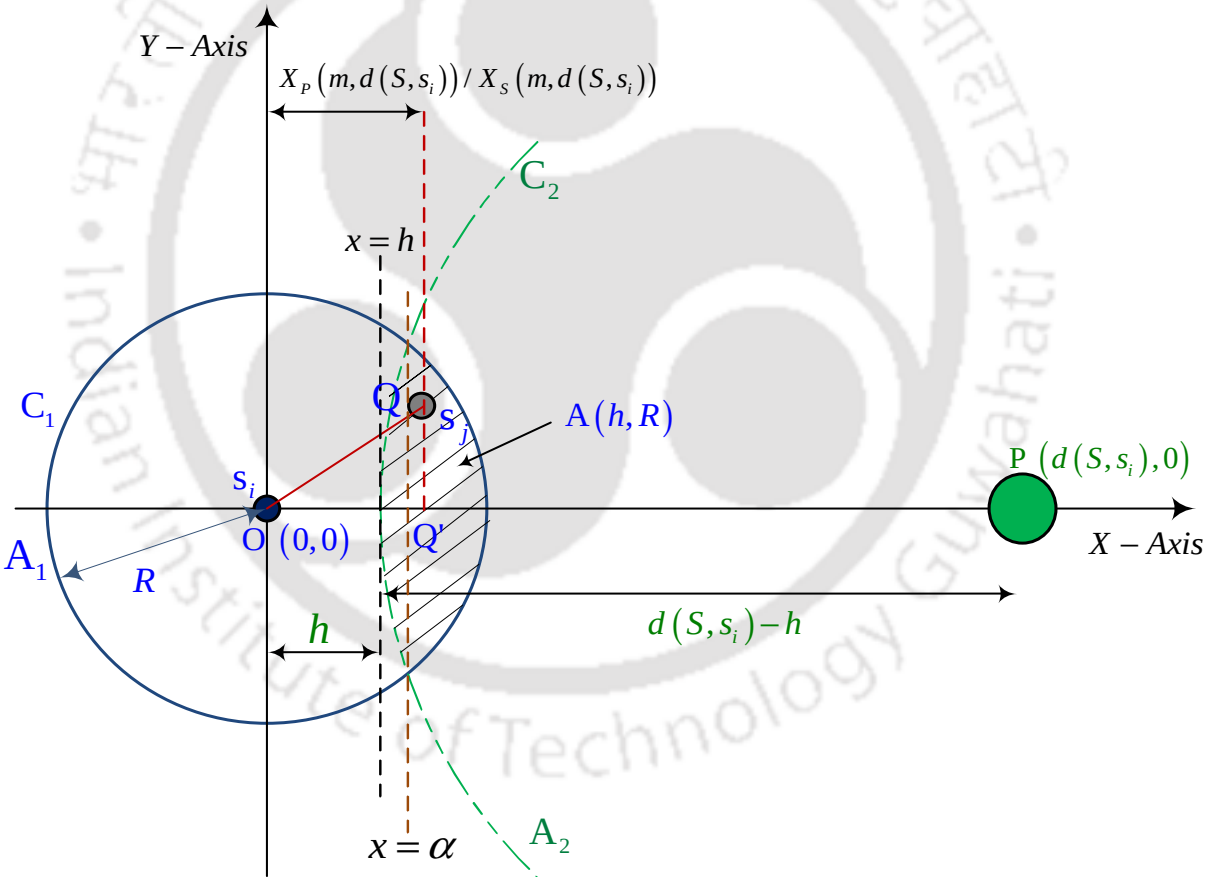


Figure 4.5: Depiction of various parameters used in m_{opt} determination.

Let s_j $(1 \leq j \leq n)$ be the next-hop forwarder of s_i which is located at Q (a, b) where $-R \leq a, b \leq R$. In Fig. 4.5, OQ represents the distance between the sensor nodes s_i and s_j and OQ' represents the projection of OQ on OP. Let X_P $(m, d(S, s_i))$ and X_S $(m, d(S, s_i))$ be the random variables representing the lengths of the OQ', in case, s_i is the *primary member* and *secondary member*,

respectively. In our proposed framework, irrespective of the implemented protocol, in each cycle, a sensor node gets one chance as a *primary member* and $m-1$ chances as a *secondary member* to forward its data packets. Further, when a sensor node schedules forwarding of its data packet as a *secondary member*, in the first hop, the data packet travels an average $E\left[X_S\left(m\right)\right]$ distance towards the sink node and, in the each of the remaining hops, it travels an average $E\left[X_P\left(m\right)\right]$ distance towards the sink node. $E\left[X_P\left(m\right)\right]$ and $E\left[X_S\left(m\right)\right]$ are given as,

$$\mathrm{E}\left[\mathrm{X_p}\left(m\right)\right] = \sum_{i=1}^{n'} \frac{\mathrm{E}\left[\mathrm{X_p}\left(m, d\left(S, s_i\right)\right)\right]}{n'} \tag{4.2}$$

and

$$\mathrm{E}\left[\mathrm{X_s}\left(m\right)\right] = \sum_{i=1}^{n'} \frac{\mathrm{E}\left[\mathrm{X_s}\left(m, \left(S, s_i\right)\right)\right]}{n'}, \tag{4.3}$$

respectively.

Here, $E\left[X_P\left(m, d\left(S, s_i\right)\right)\right]$ and $E\left[X_S\left(m, d\left(S, s_i\right)\right)\right]$ represent the means of random variable $X_P(m, d(S, s_i))$ and $X_S(m, d(S, s_i))$, respectively. Furthermore, when a sensor node schedules forwarding of its data packet as a *primary member*, in each hop, the data packet travels an average $E\left[X_p\left(m\right)\right]$ distance towards the sink node. Thus, in each cycle, a data packet can travel $m.\kappa-(m-1)$ hop with the per hop distance $E\left[X_P\left(m\right)\right]$ and $m-1$ hop with the per hop distance $E\left[X_S\left(m\right)\right]$, towards the sink node. Therefore, $AHD\left(m\right)$ can be shown as,

$$AHD\left(m\right) = \left(\frac{m.\kappa-(m-1)}{m.\kappa}\right).E\left[X_P\left(m\right)\right] + \left(\frac{m-1}{m.\kappa}\right).E\left[X_S\left(m\right)\right]. \tag{4.4}$$

4.3.2 Determination of $E\left[X_S\left(m, d\left(S, s_i\right)\right)\right]$

To determine $E\left[X_S\left(m, d\left(S, s_i\right)\right)\right]$, we propose Algorithm 4.3 considering that, in Fig. 4.5, s_i is a *secondary member*. In this case, s_i and its next-hop forwarder s_j lie in the different DSs. Therefore, $A_1.\rho\left(m\right) \geq 1$ ensures the existence of s_j in the CR of s_i where $\rho\left(m\right)$ represents the node density corresponding to the each DS. In Algorithm 4.3, our approach is based on the assumption that the node density corresponding to each DS is the same and is uniform throughout the network area. In the proposed approach, we first determine the probability of existence of the s_j in the CR of s_i. In case $A_1.\rho\left(m\right) > 1$, we determine the value of parameter h $(\in [-R, R])$ such that $A\left(h, R\right).\rho\left(m\right) = 1$. (Recall that variation in h results in the variation in the radius of circle C_2 (i.e., $d-h$).) $A\left(h, R\right).\rho\left(m\right) = 1$ ensures that the area $A\left(h, R\right)$ contains one node of each DS. From Fig. 4.5, it can be observed that, for each

$h \in [-R, R]$, nodes lying in the area $A(h, R)$ are closer to S than the nodes lying in the remaining area of C_1. This ensures that the chosen next-hop forwarder s_j lies in the $A(h, R)$. Using this h in Eq. (4.1), we determined α. Finally, using h and α, we determine the expectation (mean) of the length of the OQ' as $E[X_S(m, d(S, s_i))]$. This can be shown to be,

$$E[X_S(m, d(S, s_i))] = \int_{h}^{\alpha}\int_{-\beta}^{\beta} \rho(m)\, x dy dx + \int_{\alpha}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \rho(m)\, x dy dx, \tag{4.5}$$

where, $\beta = \sqrt{(d(S, s_i) - h)^2 - (x - d(S, s_i))^2}$.

On the other hand, in case of $A_1.\rho(m) \leq 1$, due to the identical and uniform density of sensor nodes of each DS throughout the network area, $E[X_S(m, d(S, s_i))]$ will be zero. This shows that, in case of $A_1.\rho(m) \leq 1$, data packet transmission by a *secondary member* does not significantly help in forwarding the data packet towards S.

Algorithm 4.3 $E[X_S(m, d(S, s_i))]$ determination.

```
# Input: n', d(S, s_i) ∀ i ∈ [1, n'], R, m, and ρ(m)
# Output: E[X_S(m, d(S, s_i))] ∀ i ∈ [1, n']
for i ← 1 : 1 : n' do
  if A_1.ρ(m) > 1 then
    Determine h such that A(h, R).ρ(m) = 1
    Determine α corresponding to this h using Eq. 4.1
    Use ρ(m), d(S, s_i), and above determined h and α in Eq. (4.5) to determine
    E[X_S(m, d(S, s_i))]
  else if A_1.ρ(m) ≤ 1 then
    E[X_S(m, d(S, s_i))] ← 0
  end if
end for
```

4.3.3 Determination of $E[X_P(m, d(S, s_i))]$

To determine $E[X_P(m, d(S, s_i))]$, we propose Algorithm 4.4 considering that, in Fig. 4.5, s_i is a *primary member*. In this case, s_i and its next-hop forwarder s_j lie in the same DSs. Therefore, $A_1.\rho(m) \geq 2$ ensures the existence of s_j in the CR of s_i. As in Algorithm 4.3, in Algorithm 4.4, our approach is based on the assumption that the node density corresponding to each DS is identically uniform throughout the network area. In the proposed approach, we first determine the probability of existence of the s_j in the CR of s_i. In case $A_1.\rho(m) > 2$, we search for $h \in (0, R]$ such that $A(h, R).\rho(m) = 1$. Note that, for $h \in (0, R]$, $A(h, R).\rho(m) = 1$ ensures that the area $A(h, R)$ contains s_i's next-hop forwarder s_j. In case no such h exist, we search for $h \in [-R, 0]$ such that

$A(h,R).\rho(m) = 2$. It is to be noted that, for $h \in [-R, 0]$, $A(h,R).\rho(m) = 2$ ensures that the area $A(h,R)$ contains both s_i and its next-hop forwarder s_j. Using this h in Eq. (4.1), we determined α. Finally, using h and α, we determine the expectation (mean) of the length of the OQ' as $E[X_P(m, d(S, s_i))]$. This can be shown to be,

$$E[X_P(m, d(S, s_i))] = \int_{h}^{\alpha}\int_{-\beta}^{\beta} \rho(m)\, x dy dx + \int_{\alpha}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \rho(m)\, x dy dx, \tag{4.6}$$

where $\beta = \sqrt{(d(S, s_i) - h)^2 - (x - d(S, s_i))^2}$.

On the other hand, in case of $A_1.\rho(m) \leq 2$, due to the identical and uniform density of sensor nodes of each DS throughout the network area, $E[X_P(m, d(S, s_i))]$ will be zero. This shows that, in case of $A_1.\rho(m) \leq 2$, data packet transmission by a *primary member* does not significantly help in forwarding the data packet towards S.

Algorithm 4.4 $E[X_P(m, d(S, s_i))]$ determination.

```
# Input: n', d(S, s_i) ∀ i ∈ [1, n'], R, m, and ρ(m)
# Output: E[X_P(m, d(S, s_i))] ∀ i ∈ [1, n']
for i ← 1 : 1 : n' do
  if A_1.ρ(m) > 2 then
    if A(0, R) ρ(m) > 1 then
      Determine h ∈ (0, R] such that A(h, R).ρ(m) = 1
    else
      Determine h ∈ [−R, 0] such that A(h, R).ρ(m) = 2
    end if
    Determine α corresponding to this h using Eq. 4.1
    Use ρ(m), d(S, s_i), and above determined h and α in Eq. (4.6) to determine
    E[X_P(m, d(S, s_i))]
  else if A_1.ρ(m) ≤ 2 then
    E[X_P(m, d(S, s_i))] ← 0
  end if
end for
```

4.3.4 Validation of m_{opt} Determination Process

To analyze the performance of our proposed approach, we randomly deploy n sensor nodes in a uniform fashion in 3000×3000 meter^2 area, and place a sink (S) at $(2700, 2700)$. We set the value of κ and R equal to 3 and 180 meter, respectively. Fig. 4.6 shows variations of $AHD(m)$ and $m.AHD(m)$ with m for $n = 400,\ 625,$ and 900. For each value of n, $AHD(m)$ decreases with an increase in m. This happens due to the decrease in $\rho(m)$ with an increase in m. On the other hand,

$m.AHD\,(m)$ first increases with m, achieves peak value and then starts to decrease. The value of m that maximizes $m.AHD\,(m)$ is considered as m_{opt}. In Fig. 4.6, it may be observed that in case of $n = 400,\ 625,$ and $900,\ m_{opt}$ are 2, 3 and 4, respectively.

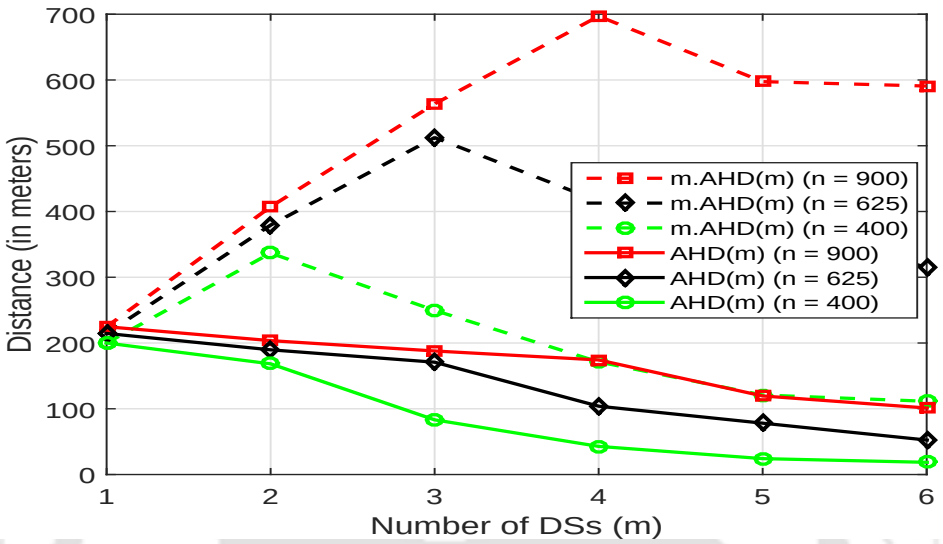


Figure 4.6: Variations in $AHD\,(m)$ and $m.AHD\,(m)$ with m for $n = 400,\ 625,$ and 900.

To verify the obtained value of m_{opt} with the help of simulations, we implement CL-MAC [55] protocol in our proposed framework using ns-2.35 simulator. For each value of n, we randomly choose 10 sensor nodes from the sensor nodes those lie at multi-hop distance from the sink S as the source nodes. We consider that each chosen source node generates one data packet to send to the sink S. The minimum time interval between the data packet generation instance of any two source node is 200s. This ensures that at any instant of time there is at the most one data packet to be forwarded in the network. The E2ETD of a data packet is equal to the time difference between the time instant when it is received at the sink and the time instant when it is generated at the source. AE2ETD is the average of the E2ETD of all the data packets received at the sink during the simulation run. Duty-cycle related parameters used in simulations are given in Table 4.1. Networking parameters are same as those in RMAC [42] (shown in Table 4.2) and frame size related parameters are similar to CL-MAC (shown in Table 4.3). In our simulations, each node is assumed to have one omnidirectional antenna and uses the two-ray ground reflection radio propagation model. As in CL-MAC [55], we also assume that the nodes follow a single sleep/wakeup schedule and routing information is available to them corresponding to the shortest path. We set the communication range of each sensor node equal to 180 meter. Fig. 4.7 shows the variation of AE2ETD with m for $n = 400,\ 625,$ and 900.

In Fig. 4.7, AE2ETD at each value of m is averaged over 15 simulations with different seeds, each lasting for 3300 seconds. From Fig. 4.7, it can be observed that for $n = 400,\ 625,$ and 900, minimum AE2ETD is obtained at $m = 2,\ 3$ and 4, respectively. This shows that the simulation and analytical approach provides the same m_{opt} for each n. It can be noted that, with the increase in node density (i.e., n), m_{opt} increases. Moreover, improvement in AE2ETD is better at the higher values of m_{opt}. The reason behind existence of the optimum m_{opt} is that for $m > 1$, E2ETD is affected by two factors

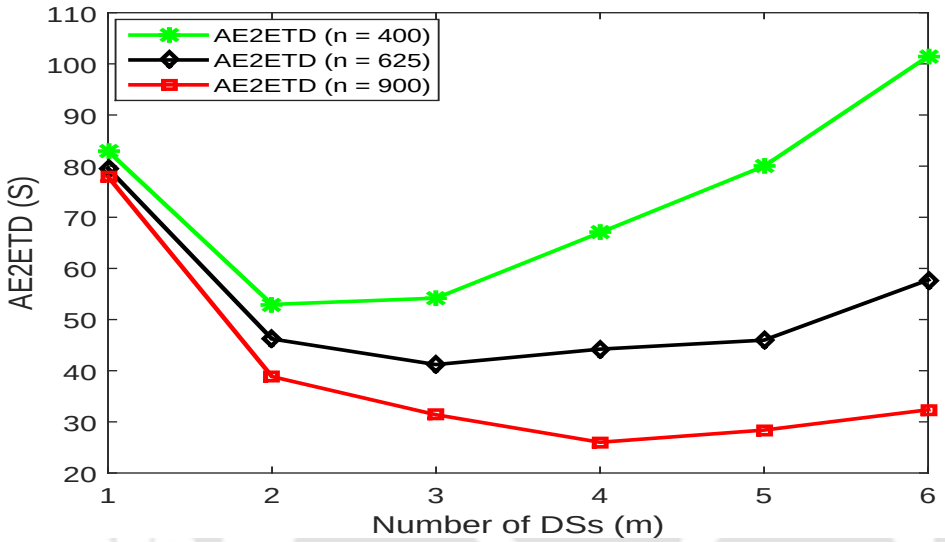


Figure 4.7: Variations in AE2ETD with m for $n = 400,\ 625,$ and 900.

- firstly the number of segments in a cycle and secondly the node density per DS. Increase in number of time segments per cycle improves E2ETD, while decrease in nodes per DS increases E2ETD due to decrease in $AHD\,(m)$ (as shown in Fig. 4.6). Recall that, with the increase in m, the number of time segments per cycle increases, while the number of nodes per DS decreases. For $m \leq m_{opt}$, the first factor dominates and therefore performance improves with increasing m and reaches its best value. For $m > m_{opt}$, the second factor starts to dominate and therefore, E2ETD starts to increase.

4.4 Results

We evaluate the performance of the proposed framework using the ns-2.35 simulator. For performance evaluation, we implement RMAC [42], PRMAC [48], CL-MAC [55], and the protocol presented in Chapter 3 (LDC-MAC) in our proposed framework.

For RMAC and PRMAC implementation, we consider the network where n sensor nodes are

randomly deployed in a uniform fashion in 1800×1800 meter^2 area and a sink node (S) is placed at $(0, 0)$. The network contains two source nodes, A and B, which are placed at $(1750, 1750)$ and $(1750, 50)$, respectively. In this case, we set the CR (R) of each network node equal to 180 meter. For PDR and E2ETD evaluation, we keep both the sources active such that each source generates data packets during the entire simulation time with packet arrival interval (PAI) varying from 4.0s to 9.0s in steps of 1.0s. In the case of AEC, only source A is active which generates 50 packets with each aforementioned PAI. In case of E2ETD and PDR, the results are averaged over 40 simulations with different seeds, each lasting for 300.0s. In the case of AEC, the results are averaged over 35 simulations with different seeds, each lasting for 1500.0s. We follow the technique proposed in Section 4.3 to determine the m_{opt}, which occurs at $m = 9$ and $m = 13$ for $n = 625$ and $n = 900$, respectively.

For CL-MAC and LDC-MAC implementation, we consider the network where 225 sensor nodes are randomly deployed in a uniform fashion in 2400×2400 meter^2 area and a sink node (S) is placed at $(1250, 1250)$. In this case, we set the CR (R) of each network node equal to 250 meter. A cluster of six sensor nodes is randomly chosen from the deployed 225 sensor nodes as the source nodes. To measure PDR and AE2ETD, each source node generates data packets during the entire simulation time, with the PAI varying from 2.0s to 12.0s in steps of 2.0s. To measure the AEC, each source node generates only 30 data packets during the entire simulation time, with each aforementioned PAI. In case of each performance metrics, results are averaged over 40 simulations with different seeds and the duration of each simulation is kept at 600.0s. For this network scenario, with the help of the technique proposed in Section 4.3, we obtained $m = 3$ as m_{opt}.

In our simulations, each node is assumed to have one omnidirectional antenna and uses the two-ray ground reflection radio propagation model. As in [42, 48, 55], we also assume that the nodes follow a single sleep/wakeup schedule and routing information is available to them corresponding to the shortest path. Cycle duration related parameters used in simulations are given in Table 4.1. Networking parameters are same as those in [42, 48] (shown in Table 4.2). Frame sizes are same as those in RMAC [42], PRMAC [48], and CL-MAC [55] (shown in Table 4.3). We add a 4-bit sized field in the SYNC packet to accommodate the current DS number of the sender. As can be seen in Table 4.1, to compensate the increased size of SW due to the sender's DS number accommodation in the SYNC packet, we reduce the size of DW such that $T_{SW} + T_{DW}$ duration remains the same for both $m = 1$ and $m > 1$. In case of each protocol, 95.0% confidence intervals are calculated for every metric

and are shown in the figures by error bars. In simulation results, $m = 1$ represents the original MAC protocol, while $m > 1$ represents the MAC protocol implemented in our proposed framework when the network nodes are partitioned into m DSs.

Table 4.1: Cycle Duration Related Parameters

Protocol	m	$\mathbf{T_{SW}}$ (ms)	$\mathbf{T_{DW}}$ (ms)	$\mathbf{T_{SlpW}}$ (ms)	$\mathbf{T_{CYCLE}}$ (s)
RMAC/PRMAC	1	55.2	117.0	9827.80	10.0
RMAC/PRMAC	> 1	55.5	116.7	$(T_{CYCLE} - T_{SW} - m.T_{DW})/m$	10.0
CL-MAC/LDC-MAC	1	55.2	100.0	14844.80	15.0
CL-MAC/LDC-MAC	> 1	55.5	99.7	$(T_{CYCLE} - T_{SW} - m.T_{DW})/m$	15.0

Table 4.2: Networking Parameters

Bandwidth	20 Kbps	Communication Range (R)	180/250 meter
Idle Power	0.45 W	Carrier Sensing Range	550 meter
Sleep Power	0.05 W	DIFS	10 ms
Tx Power	0.5 W	SIFS	5 ms
Rx Power	0.5 W	Slots in CW of DW	64
T_{slot}	1 ms	Slots in CW of SW	31

Table 4.3: Frame Sizes

Frame Type	Size	Frame Type	Size
FSP/FSP_1	12 bytes	RTS/CTS	9 bytes
PION (RMAC)	14 bytes	PION (PRMAC)	16 bytes
DATA	50 bytes	EACK/ACK	10 bytes

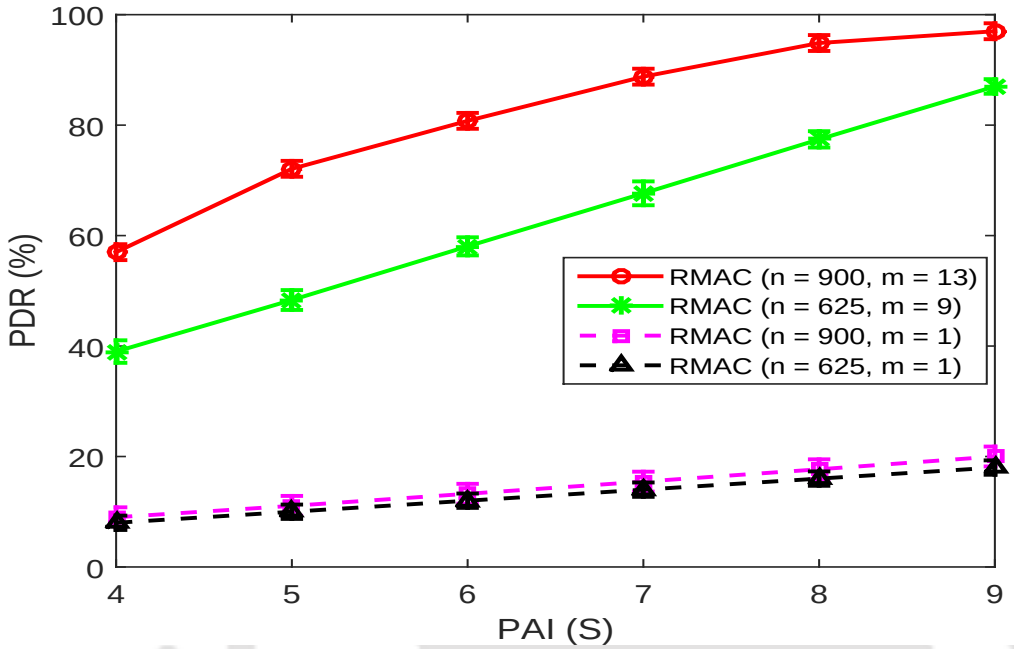
4.4.1 Results for RMAC and PRMAC Implementation

Fig. 4.8 shows the impact of the proposed framework on the PDR performance of the RMAC and PRMAC protocols. PDR is defined as the ratio of the total number of packets successfully received at the sink to the total number of packets generated at the source nodes during the entire simulation time. From Fig. 4.8 (a) and 4.8(b), it may be noted that the PDR of both the protocols increases

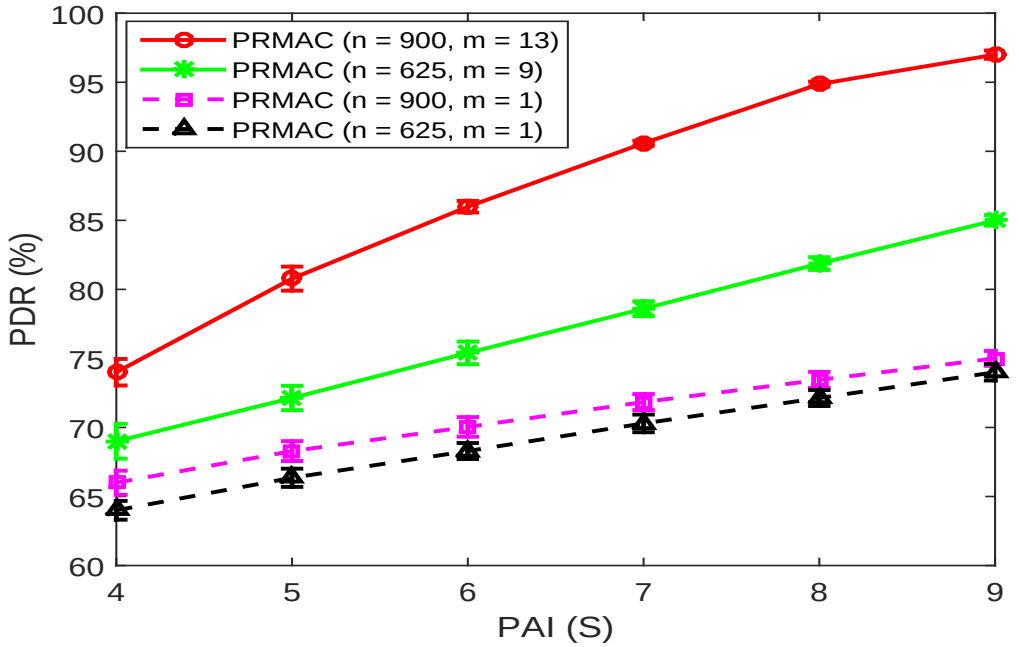
with an increase in PAI for each combination of m and n. This happens because an increase in PAI reduces the total number of data packets generated in one cycle. Further, in both the protocols, for each n, PDR increases with an increase in m. The reason for this is that in case of $m > 1$, a sensor node can schedule data packet transmission m times in each cycle and a sink node can receive data packets from m different sensor nodes in each cycle. On the other hand, in case of $m = 1$, a sensor node can schedule data packet transmission only once in each cycle and a sink node can receive data packets from at most one sensor node in each cycle. In case of $n = 625$, at PAI = 6.0s, our proposed framework increases the PDR of RMAC and PRMAC by factors of (almost) 4.75 and 1.1, respectively. Referring to Fig. 4.8, it can also be observed that increasing the node density also increases the PDR because of the increasing value of m_{opt}. In case of $n = 900$, at PAI = 6.0s, our proposed framework increases the PDR of RMAC and PRMAC by factors of (almost) 7.15 and 1.3, respectively.

Fig. 4.9 shows the impact of our proposed framework on the AE2ETD (average end-to-end transmission delay) performance of RMAC and PRMAC. Note that, in Fig. 4.9, AE2ETD is the average of the E2ETD of both the sources. The E2ETD of a source is defined as the difference between the arrival time of its first data packet at the sink and the generation time of the same data packet at the source. Referring to Fig. 4.9, for each value of n, AE2ETD of both the protocols reduces with an increase in m. The reason behind this improvement is that in the case of $m > 1$, a node can schedule data packet transmission m times in each cycle. As a result of this in one cycle, a data packet can travel almost equal to the m cycles of $m = 1$. In Fig. 4.9, in case of $n = 625$, at PAI = 6.0s, our proposed framework reduces the AE2ETD of RMAC and PRMAC by (almost) 82.0% and 75.0%, respectively. Referring to Fig. 4.9, it can also be observed that increasing the node density also reduces the E2ETD because of increasing value of m_{opt}. In Fig. 4.9, in case of $n = 900$, at PAI = 6.0s, our proposed framework reduces the AE2ETD of RMAC and PRMAC by (almost) 87.0% and 82.0%, respectively.

Fig. 4.10 shows comparison of AEC. AEC is defined as the total energy consumed during the simulation divided by the total number of sensor nodes in the network. In case of $m = 13$, sensor nodes are partitioned into 13 DSs of almost equal size. Therefore, density of active sensor nodes in DW of each segment (or total nodes per DS) reduces by the factor of 1/13 compared to $m = 1$. This leads to a small increase in the number of end-to-end hop count with increasing m, and therefore, results in small increase in AEC with m. Referring to Fig. 4.10, in both the protocols, AEC in case of $m = 13$ is almost 0.50% more than that in case of $m = 1$. AEC of PRMAC is slightly less than that

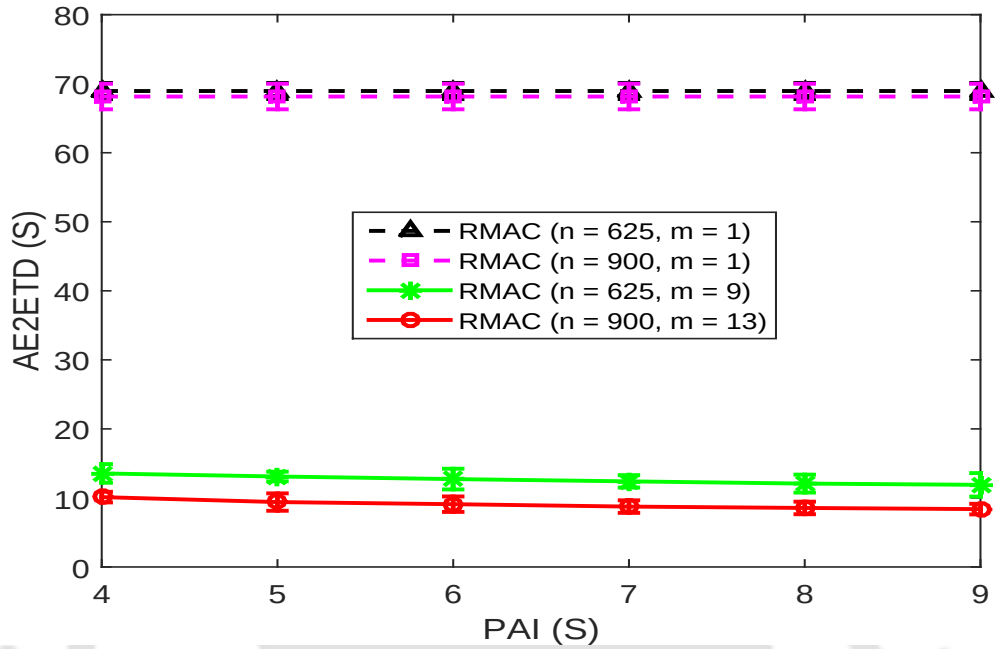


(a)

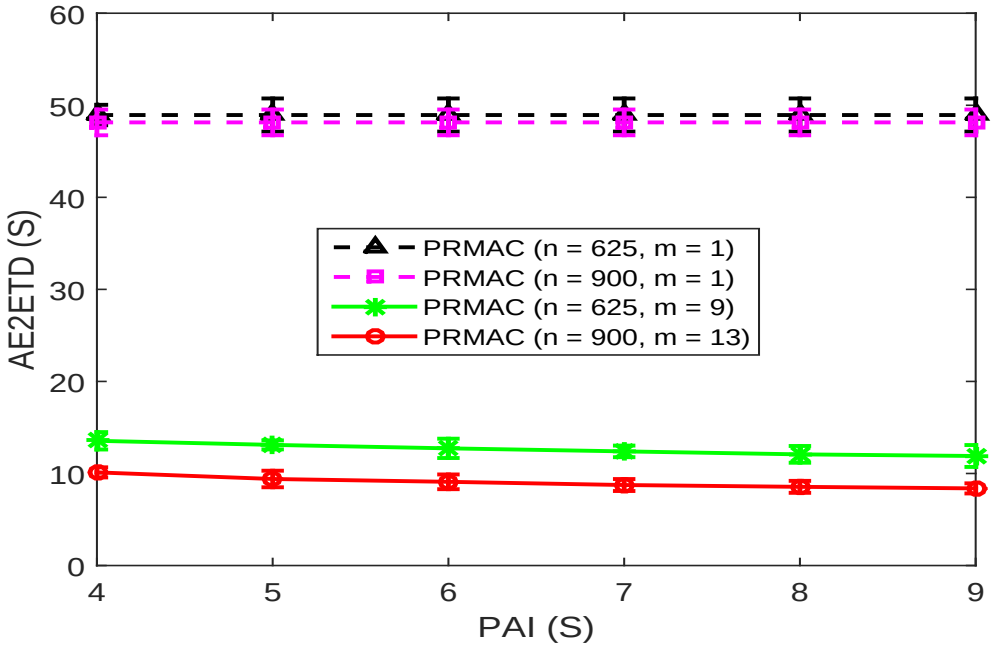


(b)

Figure 4.8: Impact of proposed framework on the PDR performance of (a) RMAC and (b) PRMAC.

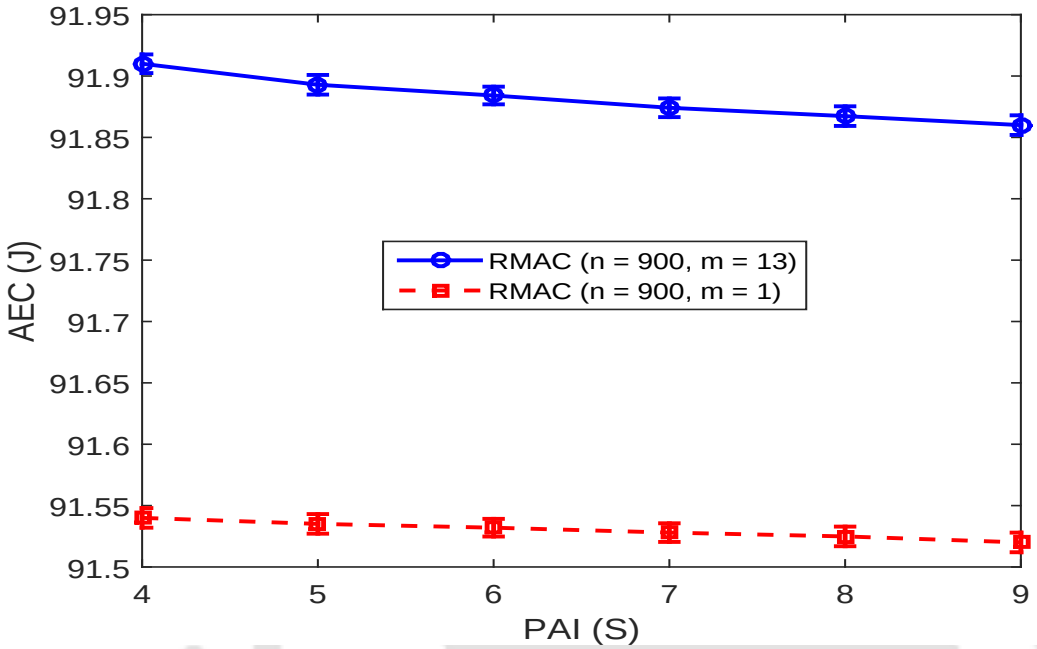


(a)

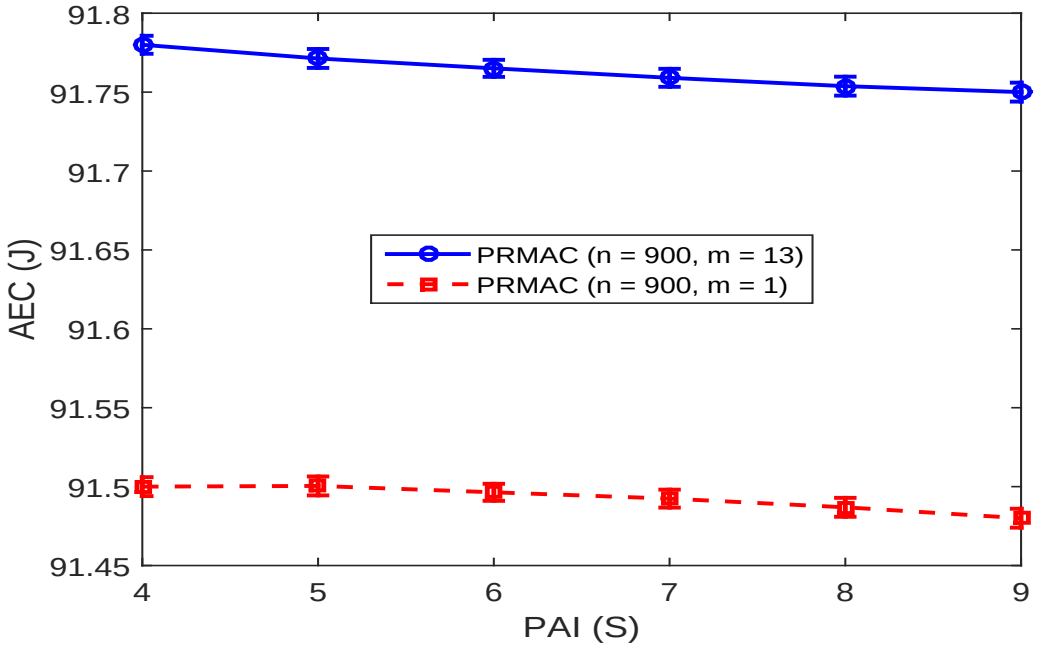


(b)

Figure 4.9: Impact of our proposed framework on E2ETD performance of (a) RMAC and (b) PRMAC.



(a)



(b)

Figure 4.10: Impact of our proposed framework on AEC (a) RMAC and (b) PRMAC.

for RMAC due to the formers' multi-hop multi-packet transmission scheduling ability, which reduces the communication overhead incurred by control packets.

4.4.2 Results for CL-MAC and LDC-MAC Implementation

Fig. 4.11 shows the impact of the proposed framework on the PDR performance of the CL-MAC and LDC-MAC protocols. From Fig. 4.11, it may be noted that the PDR of both the protocols increases with an increase in PAI for each m. This happens because an increase in PAI reduces the total number of data packets generated in one cycle. Further, in both the protocols, PDR increases with increasing m. The reason for this is that in our proposed framework, a sensor node can transmit data packets in each of the m segments of every cycle and a sink node can receive data packets in each of the m segments of every cycle. On the other hand, in both CL-MAC and LDC-MAC, a sensor node can transmit data packets only once in each cycle and, generally, a sink node receives data packets from one sensor node in each cycle. Referring to Fig. 4.11, in case of PAI = 6.0s, our proposed framework increases the PDR of CL-MAC and LDC-MAC by (almost) 10.0% and 6.0%, respectively. In Fig. 4.11, it may be noted that in case of $m = 3$, at PAI = 6.0s, the PDR of our proposed protocol (LDC-MAC) is (almost) 6.0% more than that of CL-MAC.

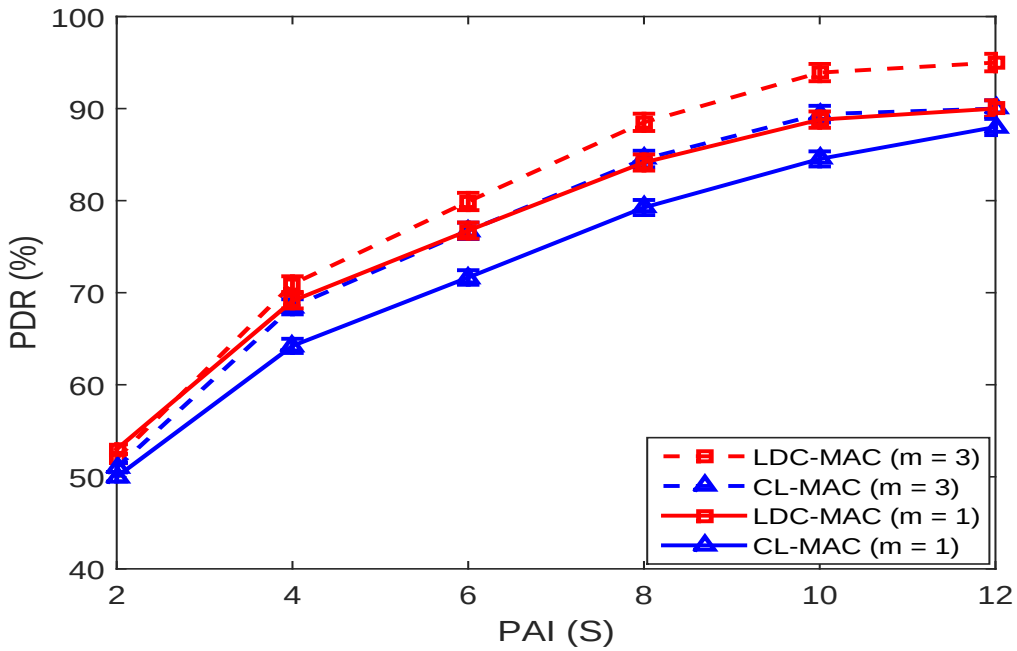


Figure 4.11: Impact of our proposed framework on PDR performance of CL-MAC and LDC-MAC.

Fig. 4.12 shows the impact of our proposed framework on the AE2ETD performance of CL-MAC

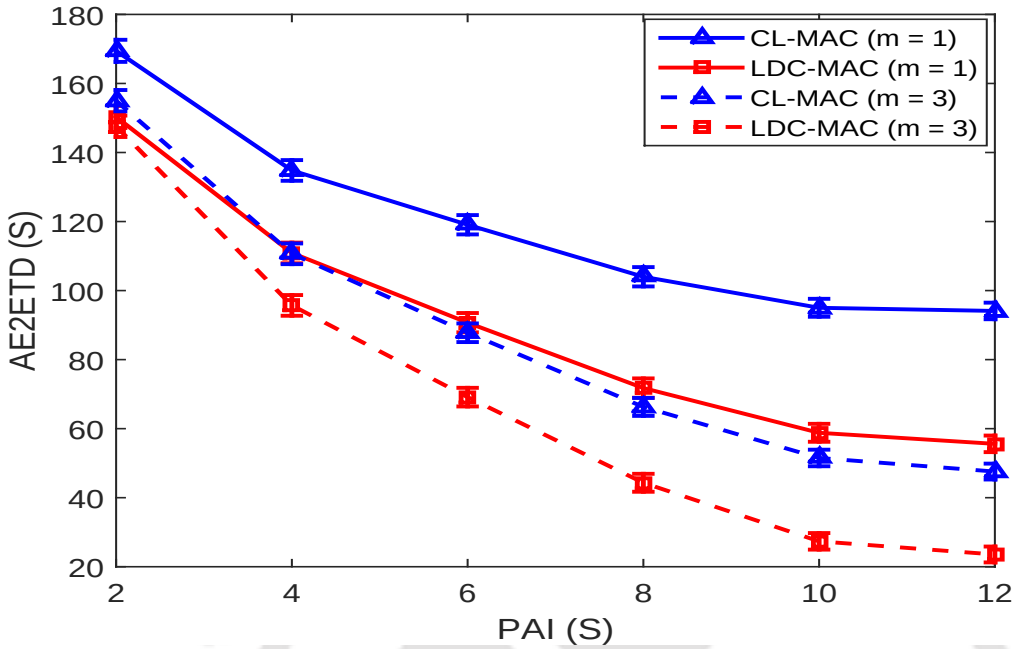


Figure 4.12: Impact of our proposed framework on E2ETD performance of CL-MAC and LDC-MAC.

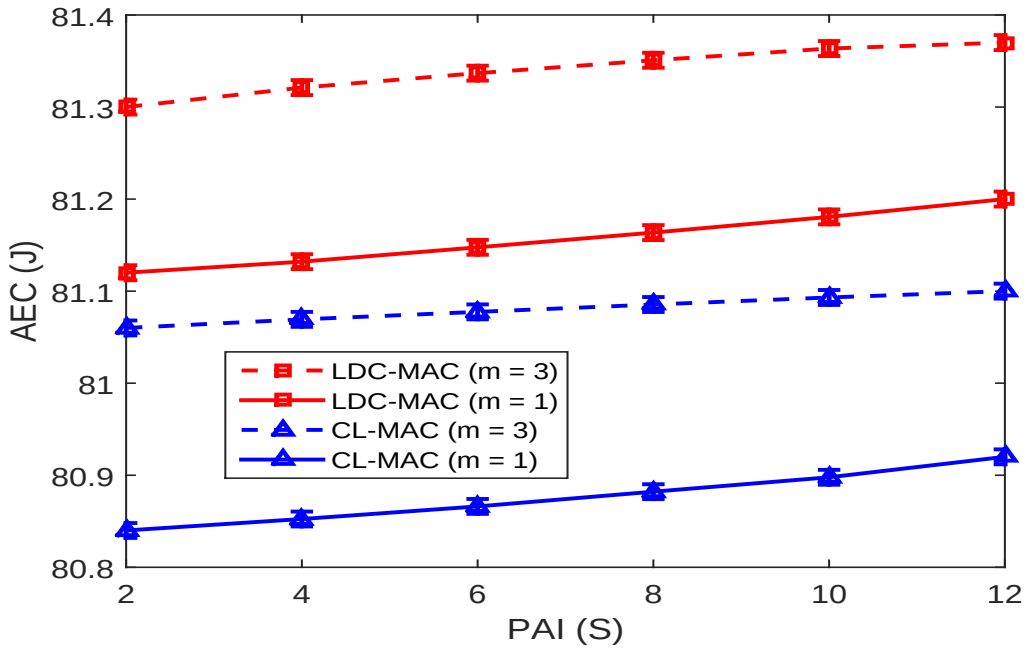


Figure 4.13: Impact of our proposed framework on AEC of CL-MAC and LDC-MAC.

and LDC-MAC. AE2ETD is the average of the E2ETD of all the data packets received at the sink during the simulation run. (Recall that E2ETD of a data packet is equal to the difference between the time instant when it is received at the sink node and the time instant when it is generated at

the source node.) Referring to Fig. 4.12, AE2ETD of both the protocols reduce with an increase in m. The reason behind this improvement is that in our proposed framework, a sensor node can transmit data packets in each time segment of every cycle. One the other hand, in both CL-MAC and LDC-MAC, a sensor node can transmit data packet only once in a cycle. As a result of this, in our proposed framework, in one cycle a data packet can travel almost equal to the m cycles of $m = 1$. Referring to Fig. 4.12, in case of PAI = 6.0s, our proposed framework reduces the AE2ETD of CL-MAC and LDC-MAC by (almost) 38.0% and 56.0%, respectively. In Fig. 4.12, it may be noted that in case of $m = 3$, at PAI = 6.0s, our proposed protocol (LDC-MAC) gives 50.0% less AE2ETD than that of CL-MAC.

Fig. 4.13 shows impact of our proposed framework on the AEC of LDC-MAC and CL-MAC. In our proposed framework, sensor nodes are partitioned into m DSs of almost equal size. Therefore, density of active sensor nodes in DW of each segment (or total nodes per DS) reduces by the factor of $1/m$ compared to CL-MAC and LDC-MAC (i.e., $m = 1$). This leads to a small increase in the number of end-to-end hop count with increasing m, which therefore results in a very marginal increase in AEC with m. Referring to Fig. 4.13, in case of $m = 3$, AEC of LDC-MAC and CL-MAC are slightly ($\approx 0.3\%$) more than that for $m = 1$.

4.5 Conclusions

In this Chapter, we have proposed a novel framework to improve the E2ETD and PDR performance of the existing *generic* contention based synchronous MAC protocols without increasing the duty-cycle (DC). To achieve this, in the network setup phase, we partitioned the deployed sensor nodes into m disjoint sets (DSs) with the help of our proposed low complexity distributed algorithms. These algorithms work such that each sensor node has equal number of sensor nodes in each of the m DSs in its neighborhood. This also results in an almost same and uniform sensor node density corresponding to each DS throughout the network area. We proposed a cycle structure that contains $m-1$ more pairs of DW and SlpW than the cycle structure used in the existing contention based synchronous MAC protocols. Sensor nodes follow this cycle structure such that, for the same values of T_{DW}, T_{SW}, and T_{CYCLE}, in our proposed framework and in an existing contention based synchronous MAC protocol, each sensor node operates with the same DC.

In the data transmission phase (DTP), we allow each sensor node to schedule data packet trans-

mission in every DW. As a result of this, in our proposed framework, in one cycle, a data packet can travel almost equal to the total distance traveled by a data packet in m cycles in the implemented protocol. Our proposed framework easily accommodates each sensor node deployed in DTP in the system, without interrupting the functioning of the network. The optimum performance of our proposed framework depends on using the optimum value of m (m_{opt}) which would vary with the sensor node density, the location of the sink node and the communication radius of the sensor nodes. To determine m_{opt}, we proposed an analytical technique and validated it through extensive ns-2.35 based simulations.

We implemented RMAC, PRMAC, CL-MAC, and LDC-MAC protocols in our proposed framework using the ns-2.35 simulator. Through extensive simulations, it is observed that our proposed framework significantly improves the E2ETD and PDR performance of each implemented protocol, at the cost of very small increase in AEC. For RMAC and PRMAC, we also analyzed the impact of node density on the performance of our proposed framework and noted that an increase in node density results in reduction in the E2ETD and increase in the PDR. It is also observed that the MAC protocol proposed in previous chapter (LDC-MAC) provides lower E2ETD and higher PDR than CL-MAC when both the protocols are implemented in the proposed framework for the same network scenario.

On the basis of our simulation results we can conclude that the scalable approach proposed here would be an effective technique to improve E2ETD and PDR in a densely deployed WSN, while keeping the AEC almost constant.

5

A Joint Routing and MAC Protocol for Transmission Delay Reduction in Many-to-One Communication Scenarios for Event-Driven WSNs

Contents

5.1	Introduction	82
5.2	Proposed Cycle Structure	84
5.3	Synchronization and Proposed Changes in SYNC Packet	84
5.4	Description of JRAM	86
5.5	Determination of α'	95
5.6	Determination of Optimum m	96
5.7	Results	99
5.8	Conclusions	106

In LDC-MAC (presented in Chapter 3) and other similar studies [42–55], all the neighbors of a sensor node follow the same sleep/wakeup schedule so that a sensor node can forward its data packet to each of its neighbors. As a result of this, a sensor node can report the detected event to any sink node if there exists a path in between the sensor node and each sink node. This feature makes these protocols well suited for scenarios where sensor nodes are deployed to monitor more than one events in the same geographic region and different events need to be reported to different sink nodes. In these protocols, the maximum number of hops traveled by a data packet in a cycle is proportional to the size of DW. Therefore, in case of small sized DW, in a multi-hop scenario, these protocols will incur larger E2ETDs in reporting the detected event to the sink node. It may be noted that an increase in DW size results in the increase in AEC due to increase in the *idle listening*. The framework presented in Chapter 4 significantly improves the E2ETD performance of such protocols in a densely deployed multi-hop WSN, at the cost of a very small increase in AEC. However, as in LDC-MAC and [42–55], in this framework, maximum number of hops traveled by a data packet in a cycle also depends on the DW duration.

In this chapter[1] we present JRAM, a joint routing and MAC protocol for transmission delay reduction in many-to-one communication scenarios for event-driven WSNs. In a many-to-one communication scenario, each sensor node reports the detected events to the sink node closest to it. Unlike in LDC-MAC and in the framework proposed in Chapter 4 and [42–55], in JRAM, the E2ETD does not depend on the DW size and a sensor node does not require a separate routing protocol to determine its appropriate next-hop forwarder. As a results of this, JRAM provides low E2ETD with low energy consumption.

5.1 Introduction

In most WSN applications, sensor nodes are deployed to monitor either only one event in the given geographic region or multiple events in the same geographic region but an event is not restricted to be report to a particular sink node (i.e., an event can be reported to any sink node). In such a scenario, a sensor node is expected to report the detected event to the nearest sink node as this may reduce E2ETD and AEC for low event occurrence rate (EOR). Therefore, a large number of sensor nodes may

[1]The work reported in this chapter has been presented in the following journal publication:
Ripudaman Singh, Brijesh K. Rai, and Sanjay K. Bose, "A joint routing and MAC protocol for transmission delay reduction in many-to-one communication paradigm for wireless sensor networks", *IEEE IOT J.*, vol. 4, no. 4, pp. 1031 - 1045, Aug. 2017.

report their detected events to the same sink node thereby resulting in a many-to-one communication scenario.

The contention based synchronous MAC protocols proposed to reduce the E2ETD in a many-to-one communication scenario (e.g., [59–64]) assigns a *grade* to each network node. This represents the minimum number of hops a packet needs to travel from the node before arriving at one of the sink nodes. On the basis of this *grade* information, [59–64] assign a *staggered* wakeup schedule to the nodes in an adjacent *grade* and the same schedule to nodes of the same *grade*. This *staggered* wakeup scheduling enables the forwarding of a data packet from source to the sink in a pipelined fashion and results in improved E2ETD. However, the E2ETD and PDR performance of [59–64] may degrade drastically with an increase in M when M sensor nodes lie in the carrier sensing range (CSR) of each other and try to forward their data packet in the same cycle. This happens because of the following shortcomings - (1) in such protocols, a sink node can receive data packet from the at most one sensor node in a cycle and (2) they provide only one chance to a sensor node to succeed in data packet transmission scheduling in a cycle. Therefore, the existing *staggered* wakeup schedule based MAC protocols may not be very suitable for monitoring delay-sensitive events in a dense multi-hop WSN where an event is detected by the more than one sensor nodes that may lie in the CSR of each other.

In this chapter, we propose JRAM, a joint routing and MAC protocol for transmission delay reduction in many-to-one communication scenarios for event-driven WSNs. Our main contributions can be summarized as follows.

(i) JRAM uses a novel cycle structure which provides m chances to a node to succeed in data transmission scheduling in a cycle and enables a sink node to receive data packets from m different sensor nodes in the same cycle.

(ii) JRAM partitions the network nodes into m disjoint sets (DSs) to reduce the *idle listening* period of a node in a cycle, without affecting the network connectivity.

(iii) JRAM assigns four parameters to each network node for partitioning the nodes into DSs and to *stagger* the wakeup schedules of adjacent *grade* nodes. With reference to the structure of the *synchronization message* (SYNC packet) in flooding based synchronization protocols, such as [[72–76] etc., four new fields are added in the SYNC packet, which is periodically broadcast. These additional fields accommodate the parameters of a sender so that a node deployed after

the Setup Phase can be quickly accommodated in the system. They also allow an existing node to periodically update its lookup table (LT).

(iv) To reduce the energy-consumption, JRAM uses distributed algorithms which make a particular sequence of the parameters assigned to a sensor node unique within the communication range (CR) of the sensor node. This sequence is used as the address of the sensor node, after the Setup Phase, to overcome the overhead of the added fields.

5.2 Proposed Cycle Structure

Our proposed cycle structure is shown in Fig. 5.1, where T_{CYCLE} represents the duration of one cycle. As can be seen in Fig. 5.1, a cycle is divided into N equal time segments (TSs), where the i^{th} TS is represented by TS_i. TS_1 and TS_2, each contains one SW and m DWs. For $j \in [1, 2]$, we use the notations SW_j and DW_{ji} to represent the SW and i^{th} DW of TS_j, respectively. For $j \in [3, N]$, each TS_j contains m data communication windows (DCWs), and the i^{th} DCW of TS_j is represented by DCW_{ji}. The structure of SW, DW, and DCW are shown in Fig. 5.2 (a), 5.2 (b), and 5.2 (c), respectively. The duration of each TS (T_{TS}) is equal to the $max(T_{SW} + m.T_{DW}, m.T_{DCW})$ where T_{SW}, T_{DW} and T_{DCW} are the durations of one SW, one DW and one DCW, respectively. We use the notation T_{rd} to represent the difference in $max(T_{SW} + m.T_{DW}, m.T_{DCW})$ and $min(T_{SW} + m.T_{DW}, m.T_{DCW})$. It may be noted that, in Fig. 5.1, we have depicted the scenario by considering that $m.T_{DCW}$ is larger than $T_{SW} + m.T_{DW}$. (In Fig. 5.2, CW_S and CW_D are the contention windows (CWs) used in SW and DW, respectively; and the notation T_X denotes the transmission duration of the packet X.)

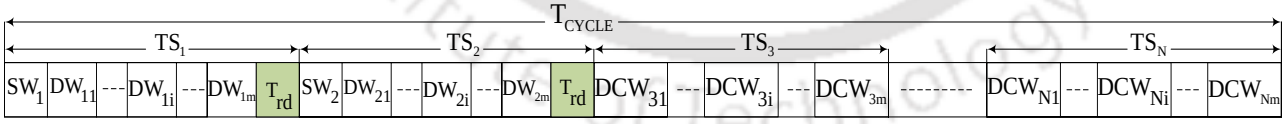


Figure 5.1: Proposed cycle structure.

5.3 Synchronization and Proposed Changes in SYNC Packet

We use the low duty-cycle synchronization protocol (LDSP) [76] to synchronize our multi-hop WSN. For this, in JRAM, sink nodes behave as the *reference nodes*. Each network node periodically (at the interval of N_{SC} cycles) broadcasts the following three timing information through the SYNC packet - (1) its current sleep/wakeup schedule, (2) the estimated current sleep/wakeup schedule of its

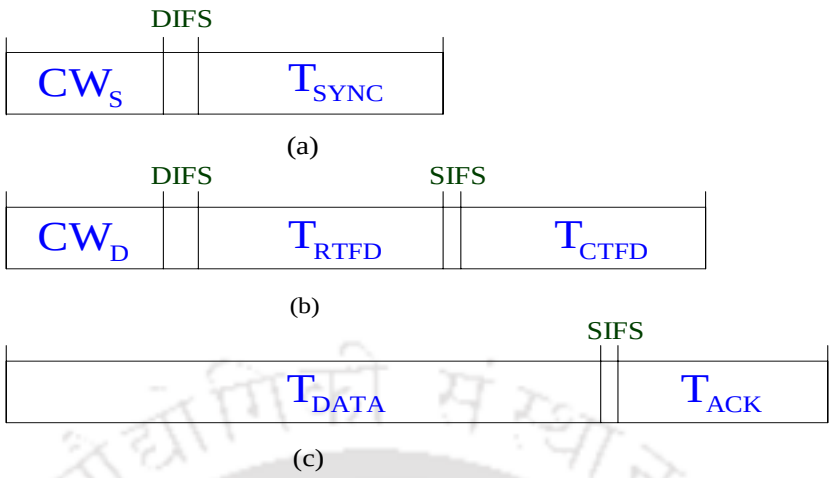


Figure 5.2: Structures of (a) SW (b) DW and (c) DCW.

reference node and (3) its estimated current clock drift rate with respect to the *reference node*. Note that unlike the other network nodes, in the case of a *reference node*, the *reference node's* estimated current sleep/wakeup schedule is the same as the sender's current sleep/wakeup schedule and the sender's estimated current clock drift rate with respect to the *reference node* is 0. In LDSP, based on the two most recently received SYNC packets from its parent node, a sensor node estimates the following - (1) its *reference node's* current sleep/wakeup schedule and (2) its own current clock drift rate with respect to the *reference node*. Other than the LDSP, other existing flooding based synchronization protocols [72–75] can also be used in JRAM to synchronize the multi-hop WSN, as they have the same basic requirements. However, because of its parallel synchronization mechanism, we prefer to use LDSP here.

Unlike LDSP, in JRAM, along with the three timing information, a SYNC packet also contains the current values of the following four parameters assigned to the sender.

(i) *Associated Sink (AS):* This represents the sink whose communication hop distance from the node is minimum.

(ii) *Grade (G):* This represents the communication hop distance of the node from *AS*.

(iii) *Subgrade (SG):* This parameter is assigned to divide the network nodes, which have the same *AS* and *G*, into groups.

(iv) *Disjoint Set Index (DSI):* This parameter is assigned to partition the network nodes into m disjoint sets (DSs).

5.4 Description of JRAM

In this section we describe JRAM in detail. We start with the Setup Phase, where we assign four parameters to each network node for partitioning the nodes into m disjoint sets (DSs) and for *staggering* the wakeup schedules of adjacent *grade* nodes. This is followed by the data transmission phase where we assign *staggered* wakeup schedules to the sensor nodes and discuss the sleep/wakeup schedule followed by the sensor nodes of the i^{th} $(1 \leq i \leq m)$ DS. We then describe the data transmission process and illustrate it with an example.

5.4.1 Setup Phase

The Setup Phase is divided into two parts, Setup Phase - 1 and Setup Phase - 2. It is assumed that before the start of Setup Phase - 1, the network has the following status.

(i) n sensor nodes (s_1, $s_2, \ldots,$ s_n) are randomly deployed, in a uniform fashion with density ρ nodes/meter^2, in the network area. The network contains t sink nodes.

(ii) A unique address is assigned to each sink node and each sensor node from $[1, t]$ and $[t+1, n+t]$, respectively.

(iii) All the sensor nodes and all the sink nodes are stationary.

(iv) All four parameters (i.e., AS, G, SG, and DSI) of each sensor node are initialized with -1 in its LT.

(v) AS of each sink node is initialized with its own address in its LT. Other three parameters (i.e., G, SG, and DSI) of each sink node are initialized with 0 in its LT.

(vi) All the sensor nodes are in their active (wakeup) state.

(vii) Sink nodes are following the cycle structure shown in Fig. 5.3 with the synchronized sleep/wakeup schedule. As shown in Fig. 5.3, the cycle duration is divided into two windows: synchronization window (SW) and sleep window (SlpW). In one SW, a node can broadcast only one SYNC packet. In each cycle, a sink node wakes up at the beginning of its SW and goes to sleep at the beginning of its SlpW.

(viii) All the t sink nodes are able to control their transmit power.

(ix) Each sensor node transmits every packet with the same transmit power (the corresponding CR is R).

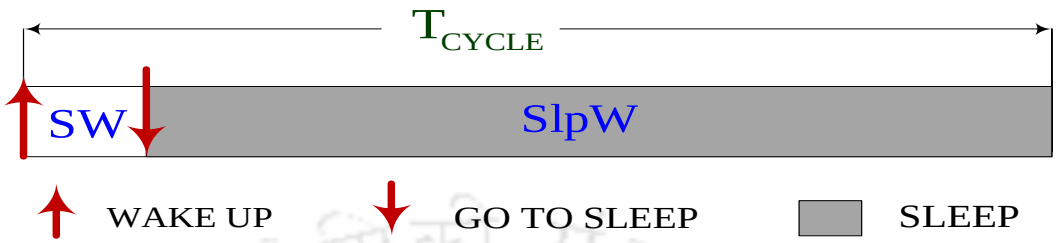


Figure 5.3: Cycle structure followed by the nodes in Setup Phase.

Setup Phase - 1: The objective of this phase is to assign the appropriate AS, G and SG to each sensor node. To initiate the process, each sink node broadcasts SYNC packets, one in each of its SW, for β' $(= \lceil 1/\alpha' \rceil)$ successive cycles. (That is, each sink node broadcasts β' SYNC packets in β' successive cycles.) Here, α' $(\in (0, 1])$ is an appropriately chosen user defined parameter whose purpose and determination process are described in Section 5.5. For $i \in [1, \beta']$, the i^{th} SYNC packet broadcast by a sink contains its current AS, G, DSI, three timing information, address, and $SG = i$. That is, for $1 \leq i, j \leq \beta'$, the i^{th} and j^{th} $(i \neq j)$ SYNC packet broadcast by a sink contains only different SG values which are i and j, respectively. A sink node broadcasts its i^{th} $(i \in [1, \beta'])$ SYNC packet with the transmission power corresponding to the CR $= i.\alpha'.R$. Each sink node repeats its β' SYNC packets broadcast process at the interval of N_{SC} $(>> \beta')$ cycles (i.e., once in every N_{SC} cycles, each sink node broadcasts β' SYNC packets). Note that a node follows the CSMA/CA technique to broadcast the SYNC packet.

After receiving the first SYNC packet, from any sensor/sink node, a sensor node sets the sender as its parent and starts to follow the cycle structure shown in Fig. 5.3, with the received sleep/wakeup schedule of the *reference node*. A sensor node, after receiving first two SYNC packets from its parent, sets a timer for $(\beta' - 2 + rand\,(N_{SC} - \beta'))\,.T_{CYCLE}$ duration and waits for timer expiration. (Here, $rand\,(N_{SC} - \beta')$ represents a randomly chosen number in $[1, N_{SC} - \beta']$.) On timer expiration, a sensor node tries to broadcast one SYNC packet in its SW. This randomized waiting time reduces/avoids contention in medium access. After broadcasting the first SYNC packet, a sensor node periodically (at the interval of N_{SC} cycles) broadcasts one SYNC packet in its SW (i.e., unlike a sink node, a sensor node broadcasts only one SYNC packet in N_{SC} cycles). Each SYNC packet broadcast by a sensor node contains its current AS, G, SG, DSI, three timing information, and address. Unlike sink nodes,

a sensor node broadcasts each SYNC packet with the transmission power corresponding to CR = R.

Further, on each SYNC packet reception from any sensor/sink node, a sensor node executes Algorithm 5.1 to update its LT. In Algorithm 5.1, the following three steps are executed to update the LT.

Step 1: If it is the first SYNC packet received from the sender then add the sender in the LT with the received AS, G, SG and DSI values.

Step 2: In case the sender already exists in the LT but it has changed one or more of its parameters since the previous SYNC broadcast, then update the changed parameter(s) in the LT.

Step 3: The sensor node sets its current AS, G, and SG equal to the received AS, received $G+1$, and received SG, respectively, if one or more of the following happen - (a) its current G is -1; (b) current G − received $G > 1$; and (c) current G − received $G = 1$ and current SG − received $SG > 0$.

There are two objectives of **Step 3**. The first is to assign the minimum G and the second is to assign the minimum SG corresponding to the current G. In **Step 3**, (a) occurs when the node is in its initial state and receives its first SYNC packet. Thereafter, (b) and (c) help a node in rectifying the error in its current G and current SG values, respectively. The functioning of rule (a), rule (b), and rule (c) of **Step 3** can be understood with the help of following example. Let the sensor node s_t receives SYNC packet containing $AS = 1$, $G = 6$ and $SG = 4$, as its first ever received SYNC packet. Following the rule (a), s_t sets its $AS = 1$, $G = 7$ and $SG = 4$. Let the second packet received by s_t contains $AS = 1$, $G = 6$, and $SG = 2$. Since, the current SG $(= 4)$ is greater than the received SG and current G is equal to the received $G + 1$, therefore, following the rule (c), node s_t sets its $AS = 1$, $G = 7$ and $SG = 2$. Now corresponding to the current G, current SG is also minimum. In this way, based on the received SYNC packets, **Step 3** ensures that (i) the current G is minimum; and (ii) corresponding to the current G, current SG is also minimum.

Each node spends λ_1 $(>> N_{SC})$ cycles in this phase, considering its first SYNC packet broadcast cycle as the first cycle. In this way, Setup Phase - 1 associates each sensor node with a sink termed as AS, and assigns G, SG, and sleep/wakeup schedule with respect to the AS.

Setup Phase - 2: In this phase, the sensor nodes are divided into m DSs by assigning a DSI, from $[1, m]$, to each sensor node. The $DSIs$ are assigned such that the $\langle AS, G, SG, DSI\rangle$ sequence of each sensor node become unique within its LT (or CR). In this way, the sensor nodes can use their $\langle AS, G, SG, DSI\rangle$ sequence as an address to reduce the SYNC packet size. For this, at the end of its setup phase-1 (i.e., ${\lambda_1}^{\text{th}}$ cycle), each sensor node randomly chooses a DSI from $[1, m]$ to replace

Algorithm 5.1 Process followed by the sensor node s_t, to update its LT, after receiving SYNC packet from a sensor node s_i.

```
# AS[s_t][s_i] : current AS of node s_i in the LT of node s_t
# G[s_t][s_i] : current G of node s_i in the LT of node s_t
# SG[s_t][s_i] : current SG of node s_i in the LT of node s_t
# DSI[s_t][s_i] : current DSI of node s_i in the LT of node s_t
# AS_SYNC^{s_i} : AS of node s_i in its most recently broadcasted SYNC packet
# G_SYNC^{s_i} : G of node s_i in its most recently broadcasted SYNC packet
# SG_SYNC^{s_i} : SG of node s_i in its most recently broadcasted SYNC packet
# DSI_SYNC^{s_i} : DSI of node s_i in its most recently broadcasted SYNC packet
# Step 1
if s_i ∉ LT then
    AS[s_t][s_i] ← AS_SYNC^{s_i}
    G[s_t][s_i] ← G_SYNC^{s_i}
    SG[s_t][s_i] ← SG_SYNC^{s_i}
    DSI[s_t][s_i] ← DSI_SYNC^{s_i}
end if
# Step 2
if s_i ∈ LT AND it changed its one or more parameters
   after previously broadcasted SYNC then
    update the changed parameter(s) in the LT
end if
# Step 3
if G[s_t][s_t] = −1 OR (G[s_t][s_t] − G_SYNC^{s_i}) > 1 OR
   (G[s_t][s_t] = G_SYNC^{s_i} + 1 AND SG[s_t][s_t] > SG_SYNC^{s_i}) then
    AS[s_t][s_t] ← AS_SYNC^{s_i}
    G[s_t][s_t] ← G_SYNC^{s_i} + 1
    SG[s_t][s_t] ← SG_SYNC^{s_i}
end if
```

its current DSI in the LT, without interrupting its periodic SYNC broadcast process. As in Setup Phase-1, in Setup Phase-2, following each SYNC packet reception, a sensor node follows Algorithm 5.1 to update its LT. Unlike Setup Phase-1, in Setup Phase-2, at the time of SYNC broadcast, a sensor node follows Algorithm 5.2. In this algorithm, first, a sensor node checks whether there is any other sensor node corresponding to its $\langle AS, G, SG, DSI\rangle$ sequence in the LT. If such a node exists, then the node searches for the DSI, which can make its $\langle AS, G, SG, DSI\rangle$ sequence unique within its LT. In case, more than one such $DSIs$ exists, then the node randomly chooses one of them as its current DSI and afterwards broadcasts its SYNC packet. Each node spends λ_2 ($>> N_{SC}$) cycles in this phase, considering the $(\lambda_1 + 1)^{\text{th}}$ cycle as the first cycle.

We have examined the operation of the Setup Phase with the help of the ns-2.35 simulator. For this, we randomly deploy $n = 900$ sensor nodes in a uniform fashion in a 1800×1800 meter^2 area and

Algorithm 5.2 The process followed by the sensor node s_i before each SYNC packet broadcast.

```
if ⟨AS, G, SG, DSI⟩ sequence of s_i is not unique in its LT then
    Search for DSI j ∈ [1, m] such that the sequence ⟨AS, G, SG, DSI⟩ become unique in the LT
    if more than one such j exists then
        Randomly choose one of them as its current DSI
    end if
end if
```

place a sink at the center. In our simulations, we consider that each sensor node and each sink node is equipped with one omnidirectional antenna; and use the two-ray ground reflection radio propagation model. The networking parameters used in the simulations are the same as those in RMAC [42] and given in Table 5.1. In addition, we consider $m = 4$, $\alpha' = 0.075$, $N_{SC} = \lceil 1.2\pi\varphi^2\rho \rceil$ where φ denotes the CSR of a sensor node. (Note that the parameter α' is determined following the process described later in Section 5.5.) In case of $\lambda_1 \geq 4N_{SC}$, it is observed that the AS, G, and SG values of each sensor node remain unchanged. In addition, in case of $\lambda_1 \geq 4N_{SC}$ and $\lambda_2 \geq 2N_{SC}$, it is observed that the $\langle AS, G, SG, DSI \rangle$ sequence of each sensor node is unique within its LT. This happens because the value of parameter α' is chosen such that there can be at most $m-1$ other sensor nodes, in the CR of a sensor node s_i $(1 \leq i \leq n)$, which have AS, G, and SG equal to the s_i's AS, G, and SG, respectively. Therefore, there will always be at least one DSI that can make the $\langle AS, G, SG, DSI \rangle$ sequence of the sensor node s_i unique in its LT. Based on these observations, we recommend the use of $\lambda_1 = 4N_{SC}$ and $\lambda_2 = 2N_{SC}$.

Even though the values of λ_1 and λ_2 are the same for each node, all the nodes cannot however terminate their Setup Phase at the same time. The reason for this is that all the nodes cannot start their Setup Phase-1 in the same cycle. Recall that the Setup Phase-1 of a node starts when it broadcasts its first SYNC packet. In general, the nodes lying at φ or less distance from each other cannot transmit SYNC packet in the same cycle. In the network considered, we observe in our simulations that each node is able to complete its Setup Phase within $15N_{SC}$ cycles where the cycle in which sink node broadcast its first SYNC packet, is considered to be the first cycle and λ_1 and λ_2 are set equal to $4N_{SC}$ and $2N_{SC}$, respectively. Furthermore, in case of $\lambda_1 = 4N_{SC}$ and $\lambda_2 = 2N_{SC}$, the G of each node is less than 7. Therefore, with 10 bits, we can represent the $\langle AS, G, SG, DSI \rangle$ sequence of any sensor/sink node. On the other hand, unique address assignment to the network nodes also requires 10 bits as there are 901 nodes in the network. That is, using the $\langle AS, G, SG, DSI \rangle$ sequence of a node as its address we can overcome the overhead of the additional fields added in the SYNC

packet to accommodate the AS, G, SG, and DSI.

5.4.2 Data Transmission Phase

After $(N - (G \ mod \ N)) .T_{TS}$ duration from the end of Setup Phase, a node having *grade* G starts to follow the cycle structure shown in Fig. 5.1, to enter its data transmission phase (DTP). We introduce the T_{TS} duration difference in the adjacent *grade* nodes schedule so that the TS_2 segment of each j^{th} *grade* sensor node completely overlaps with the TS_1 segment of $(j-1)^{\text{th}}$ *grade* sensor nodes, as shown in the Fig. 5.4. This makes the wakeup schedule of the adjacent *grade* nodes a *staggered* one, and enables forwarding of a data packet in a pipelined fashion from a source to a sink.

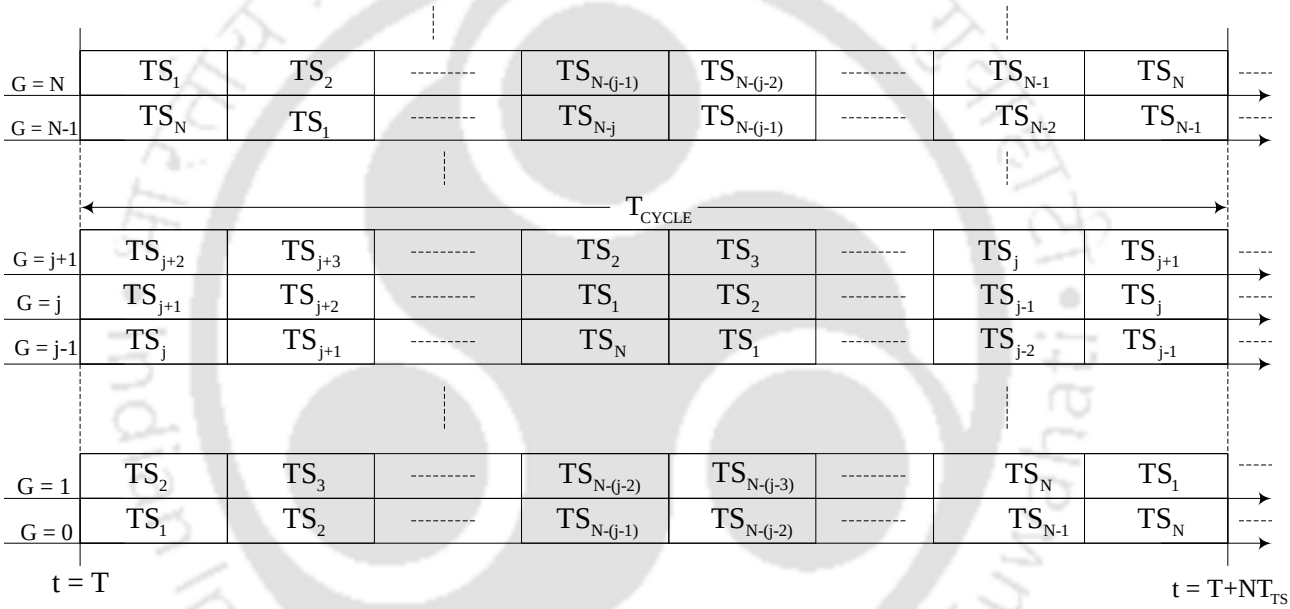


Figure 5.4: Overlapping TSs of different *grade* nodes in DTP.

For schedule synchronization as well as topology control and maintenance, a sensor node wakes up in each of its SW_2 to receive the SYNC packet, and periodically (at the interval of N_{SC} cycles) wakes up in its SW_1 to broadcast a SYNC packet. Unlike Setup Phase, in DTP, (1) at each SYNC packet reception, a sensor node follows only **Step 1** of Algorithm 5.1, and (2) the SYNC packet broadcast by a sensor node contains its address and three timing information. Note that, in DTP, the current $\langle AS, G, SG, DSI \rangle$ sequence of the sender is used as its address.

For $i \in [1, m]$, a sensor node is allowed to schedule data packet transmission in each of its DW_{2i} so that it can get multiple (m) chances to succeed in scheduling. In case a sensor node does not have data packet to send it remains in sleep in each of its DW_{2i}. On the other hand, a sensor node with

$DSI = i$ is allowed to schedule reception of data packets only in its DW_{1i}. This implies that, in an *idle cycle*, each sensor node with $DSI = i$ wakes up only during its SW_2 and DW_{1i} where *idle cycle* is defined as the cycle in which the node neither receives any packet nor tries to transmits any packet. In Fig. 5.5 (a), we show the wakeup and sleep instants of a sensor node having $G = j$ and $DSI = i$, in an *idle cycle*.

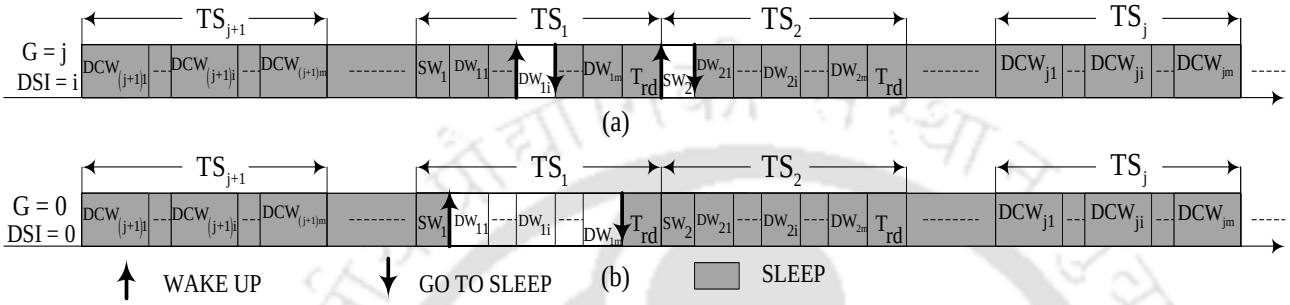


Figure 5.5: In an *idle cycle*, (a) sleep and wakeup time instants of a sensor node having $G = j$ and $DSI = i$, and (b) sleep and wakeup time instants of a sink node.

Unlike the sensor nodes, a sink node remains awake during each of its DW_{1i} so that it can schedule reception of data packets with m different sensor nodes in the same cycle. In addition, as in the Setup Phase, a sink node periodically (at the interval of N_{SC} cycles) wakes up during its SW_1 to transmit the j^{th} $(1 \leq j \leq \beta')$ SYNC packet. Note that a sink broadcasts its j^{th} SYNC packet with the transmission power corresponding to the $CR = j\alpha' R$. Unlike the Setup Phase, in DTP, the j^{th} SYNC packet broadcast by a sink node contains three timing information, and the $\langle AS, 0, j, 0\rangle$ sequence as its address. In Fig. 5.5 (b), we show the wakeup and sleep instants of a sink node in an *idle cycle*.

5.4.3 Data Transmission Process

In this section, we describe the process followed by a sensor node to transmit its data packets. A sensor node with data packets to send first tries to schedule during its DW_{21}. In case, the sensor node fails in scheduling in DW_{2j}, it tries in $DW_{2(j+1)}$ of the same cycle where $1 \leq j \leq m-1$. In this way, a sensor node gets m chances to schedule data packet transmission in each cycle. Recall that, in its LT, each sensor node contains the current AS, G, SG, and DSI of each of its one-hop neighbors. For scheduling in DW_{2i} $(1 \leq i \leq m)$, a sensor node with *grade* G chooses a $G-1$ *grade* node which has $DSI = i$ and the lowest SG, as its next-hop forwarder. A sensor node follows the steps given next for scheduling in DW_{2i}.

Step 1: At the beginning of DW_{2i}, the sensor node randomly chooses a slot of its contention window

(CW_D). We use the notations $rand(CW_D)$ and T_{slot} to represent the randomly chosen slot of CW_D and duration of one slot of CW_D, respectively.

Step 2: The sensor node sets a timer for DIFS + $rand(CW_D)$ .T_{slot} duration and senses the medium where DIFS is the distributed inter frame space. In case, it does not sense any ongoing communication until the timer expiration, it sends request-to-forward data (RTFD) packet to its next-hop forwarder. RTFD is a control packet which contains the sender's address, the receiver's address and the number of data packets the sender wants to send to the receiver. Otherwise, it cancels the timer and goes into sleep for the remaining duration of the DW_{2i}.

Step 3: If the next-hop forwarder is ready-to-receive, it sends confirmation-to-forward data (CTFD) packet. Similar to RTFD, CTFD is also a control packet which contains the sender's address and the receiver's address.

Step 4: Successful reception of CTFD at the intended receiver is considered to be successful scheduling.

In case, a sensor node schedules in its DW_{2i}, it transmits l^{th} data packet in its $DCW_{(2+pl)i}$ where $l \in [1, \lfloor (N-(2+p'))/p \rfloor]$, $p \geq \lceil (\varphi + 2R)/R \rceil$ and $p' \geq \lceil \varphi/R \rceil$. This interval of p TSs is required to avoid the collisions between the successively sent data packets of a sensor node and to avoid the collision of a data packet with SYNC/RTFD/CTFD packets. On the other hand, if a sensor/sink node schedules reception of data packets in its DW_{1i}, it receives the l^{th} data packet in its $DCW_{(1+pl)i}$. That is, a sensor node can transmit and/or receive N' data packets in a cycle only when $N = 2 + p' + N'p$. In this way, in JRAM, a sensor node can transmit and/or receive multiple data packets in a cycle.

In Fig. 5.6, we illustrate the data transmission scheduling process of JRAM, assuming that (1) n sensor nodes (s_1, $s_2, \ldots$, s_n) are randomly deployed in the network in a uniform fashion; (2) the network contains one sink node S; (3) the sensor nodes are partitioned into two DSs; (4) sensor nodes s_1 and s_2 lie in the CR of each other and have the same *grade*, $G = 2$; (5) s_1 and s_2 each has two data packets to send to the sink S which is at two hop distance; (6) the current $DSIs$ of sensor nodes s_3 and s_4 are 1 and 2, respectively, and both have the same *grade*, $G = 1$; (7) s_1, s_2, s_3, and s_4 lie in the CR of each other; and (8) current DSI of s_1 and s_2 are 2 and 1, respectively.

At the beginning of their DW_{21}, both s_1 and s_2 try to win the contention for channel access. As can be seen in Fig. 5.6, sensor node s_1 wins the contention for channel access and schedules transmission of both its data packets to the sensor node s_3, and after that both s_1 and s_3 go into sleep. During

this time, after sensing an ongoing communication, the sensor node s_2 goes into sleep. Note that, in JRAM, a sensor node is allowed to schedule in $DW_{2(i+1)}$ if it fails in scheduling in DW_{2i} where $1 \leq i \leq m-1$. Therefore, at the beginning of its DW_{22}, s_2 wakes up and schedules transmission of both its data packets with the sensor node s_4. Following the same process, the sensor nodes s_3 and s_4 schedule further transmission of data packets, to be received from s_1 and s_2, respectively, with the sink S. In JRAM, each sink node remains awake during its each DW_{1i} where $1 \leq i \leq m$. Therefore, the sensor nodes s_3 and s_4 succeed in scheduling during DW_{22} and DW_{21}, respectively, of the same cycle.

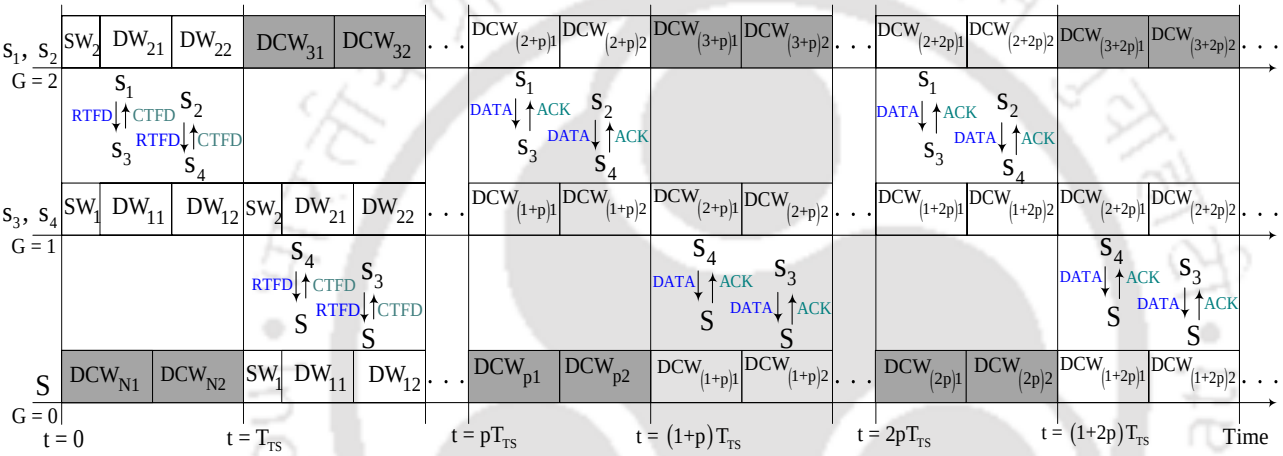


Figure 5.6: Illustration of JRAM's data packet transmission process.

It may be noted in Fig. 5.6, that a sensor node which schedules data packet transmission in DW_{2i}, transmits its l^{th} data packet in its $DCW_{(2+pl)i}$ of the same cycle, and the sensor/sink node that schedules data packet reception in DW_{1i}, receives l^{th} data packet in its $DCW_{(1+pl)i}$ of the same cycle. In this way, in JRAM, multiple (m) sensor nodes lying in the CR of each other can transmit data packets in the same cycle even though they have the same *grade*.

5.4.4 Addition/Deletion of Sensor Nodes in DTP

In the proposed protocol, a new sensor node, which has to be deployed later, i.e., after the Setup Phase, can also be easily added in the system. Such a node, first sets its AS, G, SG, and DSI values equal to -1 in its LT. Then, in the next N_{SC} cycles, it receives SYNC packets from its neighboring nodes. At each SYNC packet reception, the sensor node follows Algorithm 5.1. At the end of the N_{SC}^{th} cycle, it chooses a DSI such that its $\langle AS, G, SG, DSI \rangle$ sequence become unique within its LT. In case, there are more than one such $DSIs$, then the node randomly chooses one of them as its current

DSI and starts to follow the sleep/wakeup schedule followed by the other sensor nodes which have the same values of AS, G, and DSI. It randomly chooses one of the nodes having *grade*, one less than its own *grade*, as its parent node. Similarly, an expired node can also be easily removed. To do this, each node removes the information of an already joined node from its LT if it has not received any SYNC packet from it in the previous $\chi.N_{SC}$ cycles. (Here, χ is a suitably chosen, user defined positive integer).

5.5 Determination of α'

In this section, we discuss the importance of the parameter α' and the way it should be determined. Recall that the Setup Phase-1 associates each sensor node with a sink, referred to as its AS, and assigns G and SG with respect to the AS. As a result, the complete network is divided into multiple (equal to the number of sinks) subnetworks. As in AIMRP [1], for the sake of mathematical tractability, we assume that the transmission range of each node is identical in all the directions and the network area does not contain any obstacle. Under such an assumption, a subnetwork can be represented by annular rings, having thickness $\alpha' R$, centered at the sink, as shown in Fig. 5.7. In this representation, all the nodes lying in the same ring have same values of AS, G, and SG. Recall that, in Setup Phase-2, our objective is to assign the different $DSIs$ to all the sensor nodes which lie in the CR of each other and have same values of AS, G, and SG. This is possible only when the parameter α' is chosen such that each sensor node in a ring has less than m (number of DSs) sensor nodes in its CR, in the same ring. To determine α' in case of m DSs, we consider the subnetwork model shown in Fig. 5.8. In Fig. 5.8, each annular ring is centered at the sink located at P $(0,0)$ and the thickness of each annular ring is $\gamma' R$ where γ' $(\in (0,1))$ is a variable. Notations R_u and A_u represent the u^{th} ring and u^{th} ring's area, respectively. All the sensor nodes lying in a ring have same AS, G, and SG. Let $A^R_{(a,b)}$ be the communication area of a node lying at (a,b) with identical CR (R) in all the directions. In Fig. 5.8, $A^R_{(0,\delta)} \cap A_u$ represents the area of R_u that lies in the CR of a sensor node located at Q $(0,\delta)$. Therefore, for $\delta \in [(u-1)\,\gamma' R, u\gamma' R]$, $\left(A^R_{(0,\delta)} \cap A_u\right)\rho \leq m$ ensures that each sensor node of R_u has less than m sensor nodes of R_u in its CR. Moreover, for $\delta \in [(u-1)\,\gamma' R, u\gamma' R]$ and $u \in \left[1, U_{\gamma'}\right]$, $\left(A^R_{(0,\delta)} \cap A_u\right)\rho \leq m$ ensures that each sensor node has less than m sensor nodes of its ring in its CR where $U_{\gamma'}$ denotes the total number of rings in the subnetwork when the width of each ring is $\gamma' R$.

We propose Algorithm 5.3 to determine the maximum value of $\gamma' \in (0,1]$ which satisfies the

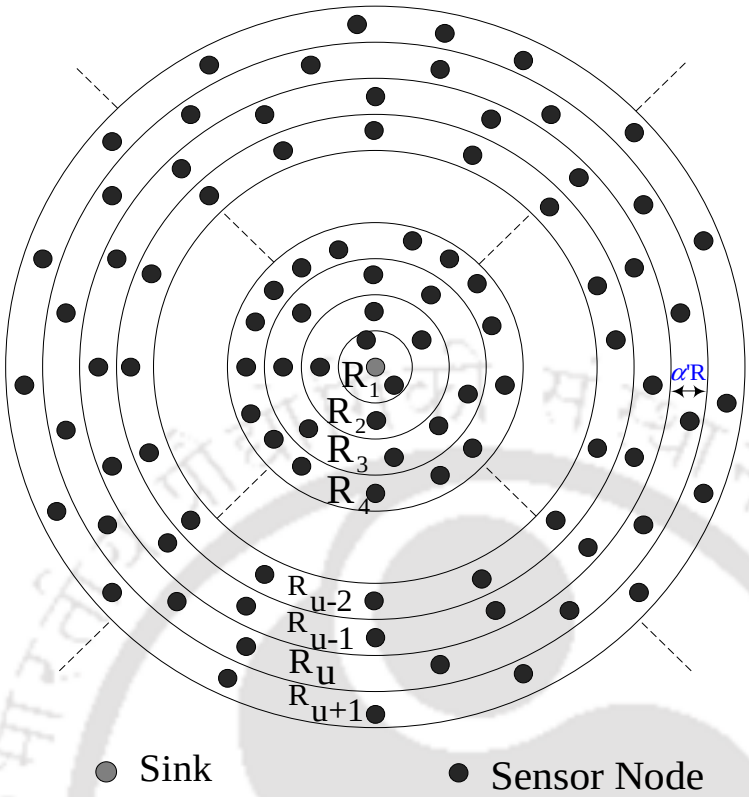


Figure 5.7: Representation of a sub-network considering that each network node has identical CR (R) in all the direction and network area doesn't have any obstacles.

constraint $\left(A^R_{(0,\delta)} \cap A_u\right) \rho \leq m$ for each combination of $\delta \in [(u-1)\,\gamma' R, u\gamma' R]$ and $u \in \left[1, U_{\gamma'}\right]$. Note that this γ' corresponds to the α' in case the sensor nodes are divided into m DSs. We choose the maximum value of γ', as that minimizes β' (recall that $SG = \beta' = \lceil 1/\alpha' \rceil$) and reduces the size of the $\langle AS, G, SG, DSI \rangle$ sequence of a node.

Fig. 5.9 shows variation of α' with m, in case of $\rho = 2.77 \times 10^{-4}$ nodes/meter^2, $\Delta_1 = 10^{-2}$, $\Delta_2 = 10^{-1}$, $R = 250$ meter and $U_{\gamma'} = 100$. It can be observed that α' increases with increasing m.

5.6 Determination of Optimum m

As mentioned in Chapter 3, in a *k-covered* WSN, an event is detected by M $(\geq k)$ sensor nodes that may lie in the CSR of each other. At event detection, a sensor node generates data packets, with the pre-defined packet generation interval, to report the event to the sink node. Because of this, M sensor nodes, lying in the CSR of each other may, have data packets to send to the same sink node in the same cycle.

In JRAM, in case of $m = X$, X sensor nodes can forward their data packets in the same cycle

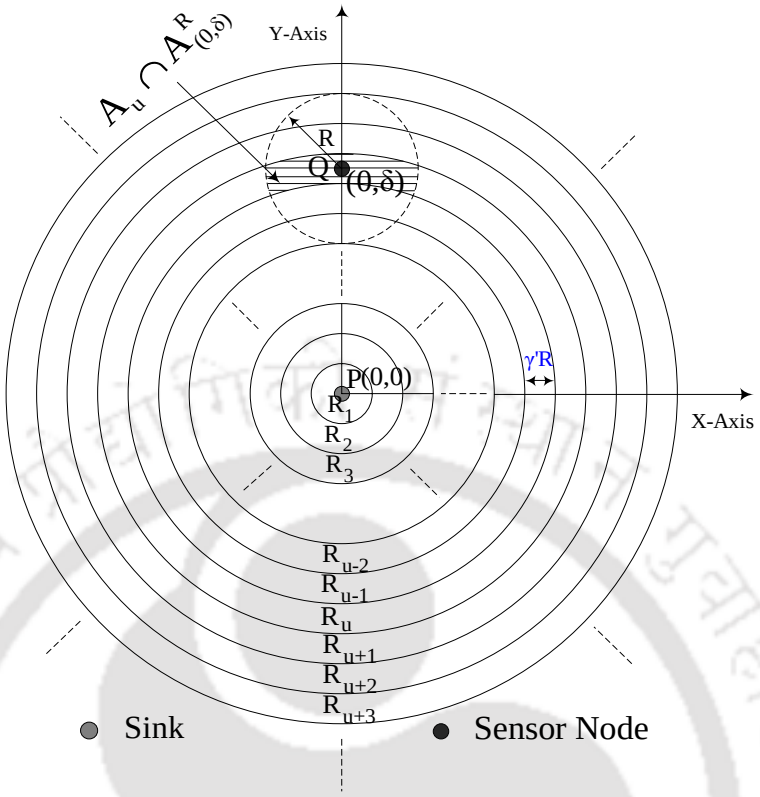


Figure 5.8: Depiction of various parameters used in α' determination.

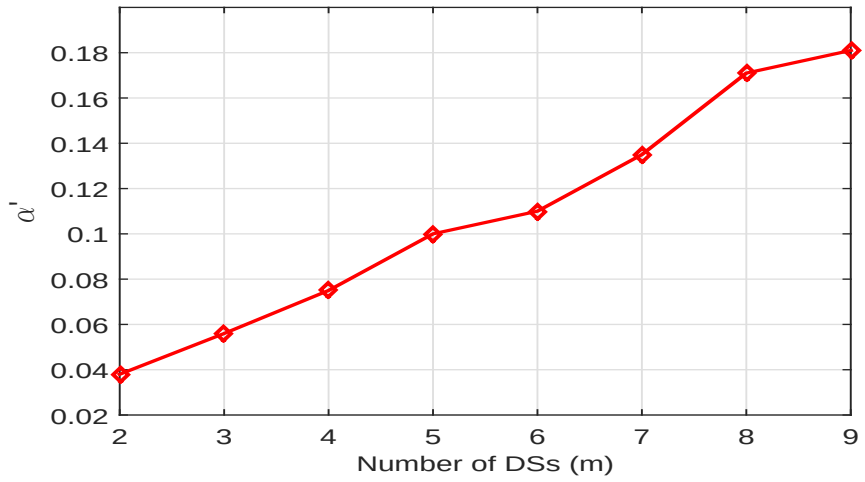


Figure 5.9: Variations in α' with m.

even when they lie in the CSR of each other. Therefore, m should be greater than or equal to the M, so that all the M sensor nodes can forward their data packets in the same cycle. Since, E2ETD of a data packet depends on the time spent by the data packet in the queue of source and relay nodes. Therefore, N should be chosen such that a sensor node can forward all the data packets generated

Algorithm 5.3 Determination of α'.

```
# Input: m, R, ρ, Δ1, Δ2 (Δ1, Δ2 ∈ (0,1))
# Output: α'
for γ' ← Δ1 : Δ1 : 1 do
    i ← 0
    for u ← 1 : 1 : U_γ' do
        for δ ← (u−1)γ'R : γ'RΔ2 : uγ'R do
            Determine A^R_(0,δ) ∩ A_u (given in Appendix A)
            if (A^R_(0,δ) ∩ A_u) ρ ≤ m then
                i ← i + 1
            end if
        end for
    end for
    % For given γ', i == U_γ' (⌊1/Δ2⌋ + 1) ensures that
    for each combination of u ∈ [1, U_γ'] and δ ∈ [(u−1)γ'R, uγ'R], (A^R_(0,δ) ∩ A_u) ρ ≤ m %
    if i == U_γ' (⌊1/Δ2⌋ + 1) then
        α' ← γ'
    end if
end for
```

by it in the previous (e.g., i^{th}) cycle in the next ($(i+1)^{\text{th}}$) cycle. Therefore, the E2ETD would be optimum if (1) $m = M$; (2) N is chosen such that the data packets generated by a source node in one cycle duration are exactly equal to the number of data packets a sensor node can forward in one cycle; and (3) each sensor node contains M or more adjacent lower grade sensor nodes in its CR.

Further, as can be seen in Fig. 5.1, with an increase in m, N reduces. As a result of this, with an increase in m, the number of data packets a sensor node can forward in a cycle reduces. Furthermore, in JRAM, in case of $m = X$, an i^{th} *grade* sensor node can get X chances to succeed in data transmission scheduling only when it has X or more $(i-1)^{\text{th}}$ grade sensor nodes in its CR. However, for $i > 1$, each i^{th} *grade* sensor node does not contain the same number of $(i-1)^{\text{th}}$ *grade* sensor nodes in its CR. In general, a sensor node having higher SG value contains less number of $(i-1)^{\text{th}}$ *grade* sensor nodes in its CR than the sensor node having a relatively lower SG value. Furthermore, with an increase in X, the number of i^{th} *grade* sensor nodes that have X or more $(i-1)^{\text{th}}$ *grade* sensor nodes in their CR, reduces. This reduces the effectiveness of the $m = X$ strategy drastically as all the sensor nodes are not able to utilize all the X DWs for data transmission scheduling. For example, in Fig. 5.10, corresponding to $X = 2$, the E2ETD and PDR are better than for $X = 3$ or 4 when $M \leq 6$ and $\rho = 2.77 \times 10^{-4}$ nodes/meter^2. Therefore, it seems appropriate to set $m = 2$ or 3, as the number

of i^{th} *grade* sensor nodes that have X or more $(i-1)^{\text{th}}$ *grade* sensor nodes in their CR, significantly reduces with increasing X.

5.7 Results

In this section, we study the performance of JRAM using the ns-2.35 simulator. We compare JRAM's performance with the PDC [62], ADC [63] and CROP-MAC [64] protocols which are specifically designed to reduce E2ETD in many-to-one communication scenarios for WSNs. For this, we randomly deployed 900 sensor nodes in a uniform fashion in an 1800×1800 meter^2 area and placed a sink at the center, i.e., (900,900). It is assumed that an event is detected by the cluster of M sensor nodes. In simulations, the cluster of M sensor nodes is randomly chosen from the deployed 900 sensor nodes as the source nodes, and each source node generates constant bit rate (CBR) traffic. The packet generation instants of all the M source nodes are synchronized for each packet arrival interval (PAI). In simulations, each node is assumed to have one omnidirectional antenna and uses the two-ray ground reflection radio propagation model. Networking parameters used in simulations are shown in Table 5.1. Frame size and cycle duration parameters are given in Tables 5.2 and Table 5.3, respectively. Notation JRAM ($m = X$) represents the JRAM when the network nodes are divided into X DSs, and the cycle structure followed by the each network node contains one SW and X DWs in each of the first two TS and X DCWs in each of the remaining $N-2$ TSs. For AE2ETD (average E2ETD) and PDR, results are averaged over 40 simulation runs with different seeds, each lasting for 600 seconds. In case of AEC (average energy-consumption), results are averaged over 50 simulation runs with different seeds, each lasting for 1500 seconds. In Fig. 5.12, 5.13, and 5.14, 95.0% confidence intervals are calculated and are shown in the figures by error bars.

Table 5.1: Networking Parameters

Bandwidth	20 Kbps	Communication Range (R)	250 meter
Idle Power	0.45 W	Carrier Sensing Range	550 meter
Sleep Power	0.05 W	DIFS	10 ms
Tx Power	0.5 W	SIFS	5 ms
Rx Power	0.5 W	Slots in CW of DW	64
T_{slot}	1 ms	Slots in CW of SW	31

Table 5.2: Frame Sizes

Frame Type	Size (bytes)
RTSD (CROP-MAC)	6
CTSD (CROP-MAC)	5
SYNC (CROP-MAC)	10
PRTS (PDC)/RTF (ADC)	11
PCTS (PDC)/CTF (ADC)	8
RTFD/CTFD (JRAM)	5
SYNC (JRAM)	9
DATA (CROP-MAC/JRAM/PDC)	50
ACK (CROP-MAC/JRAM/PDC)	9

Table 5.3: Cycle Duration Related Parameters

Protocol	SW (ms)	DW (ms)	DCW (ms)	T_{CYCLE} (s)
PDC	-	227.4	-	15.0
CROP-MAC	56.0	94.6	-	15.0
JRAM	55.2	93.0	63.2	15.0

5.7.1 Impact of m on JRAM's Performance

Fig. 5.10 (a) shows the impact of m on PDR performance of JRAM where PDR is defined as the ratio of total number of packets successfully received at the sink to the total number of packets generated at the source nodes during the simulation time. Fig. 5.10 (b) shows the impact of m on AE2ETD performance of JRAM. The E2ETD of a data packet is equal to the time difference between the time instant when it is received at the sink and the time instant when it is generated at the source. AE2ETD is the average of the E2ETD of all the data packets received at the sink during the simulation run. In Fig. 5.10 (a) and Fig. 5.10 (b), M is varied from 1 to 6 and PAI at each source node is taken as 2.0s. In case of $M = 2$, JRAM ($m = X$) has highest PDR and lowest AE2ETD corresponds to $X = 2$. This happens because increasing X reduces the number of data packets that a sensor node can send in a cycle. Furthermore, in case of each X, with increasing M, the PDR reduces and AE2ETD increases. The reason is that in JRAM ($m = X$), in case of each X, the sink can receive a fixed number of data packets per cycle while an increase in M increases the number of data packets generated in a cycle. Therefore, in case of each X, an increase in M results in a decrease in PDR and increase in AE2ETD. It should also be noted that, with increasing X, both the rate of decrease in PDR and the rate of

increase in AE2ETD reduces. This happens because the increase in X increases the total number of data packets that the sink can receive in each cycle. However, for $M > 2$, neither is the PDR highest nor is the AE2ETD lowest in case of $X = M$. This happens because, with increasing X, the number of i^{th} $(i > 1)$ *grade* sensor nodes that have X or more $(i-1)^{\text{th}}$ *grade* node in their CR, reduces.

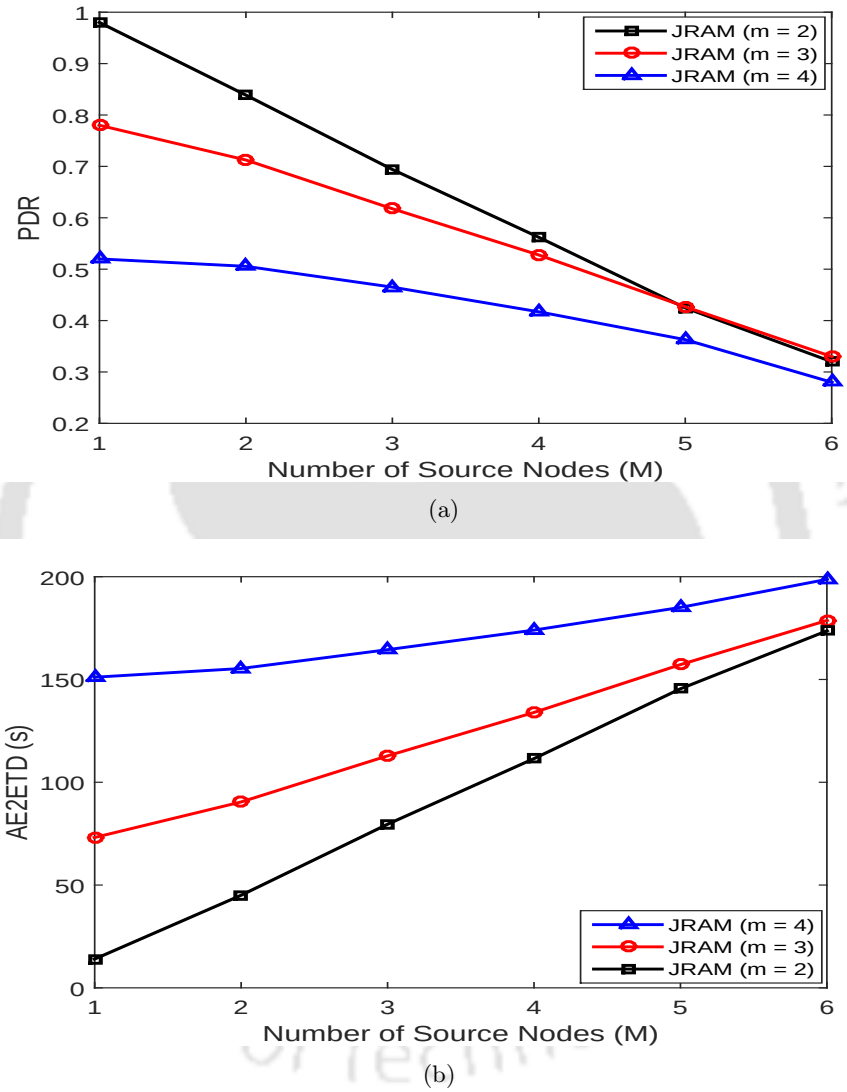


Figure 5.10: Variation in (a) PDR and (b) AE2ETD, with m.

5.7.2 Comparison of AE2ETD

Fig. 5.11 (a), 5.11 (b), and 5.11 (c) show comparison of AE2ETD of PDC, ADC, CROP-MAC, and JRAM $(m = 2)$ in case of $M = 2$, 3, and 4, respectively. In each case, the AE2ETD of JRAM $(m = 2)$ is the lowest. This happens because in JRAM $(m = 2)$, the sink can receive data packets

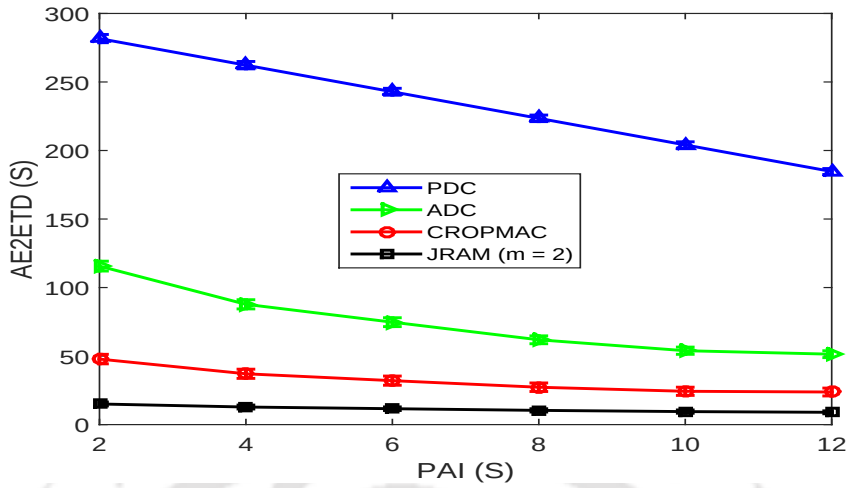
from two sensor nodes in each cycle. On the other hand, in PDC, ADC, and CROP-MAC, the sink can only receive data packet from one sensor node in a cycle. Moreover, with the reduction in PAI as well as with the increase in M, the AE2ETD of all the three protocols increase. The reason for this is that, in case of each protocol, the sink can only receive a fixed number of data packets per cycle while, both the reduction in PAI as well as the increase in M, increase the number of data packets generated in a cycle.

In each case, AE2ETD of CROP-MAC is less than the ADC. This happens because, as in JRAM ($m = 2$), in CROP-MAC the scheduling process and the synchronization process are accommodated in separate time windows. Moreover, in each case, AE2ETD of ADC is less than the PDC. The reason for this is that, as in JRAM ($m = 2$) and CROP-MAC, in ADC, a sensor node can transmit and/or receive multiple data packets in a cycle. In Fig. 5.11 (a), 5.11 (b), and 5.11 (c), at PAI = 2.0s, AE2ETD of JRAM ($m = 2$) are respectively (almost) 68.0%, 37.0%,and 36.0% less compared to CROP-MAC.

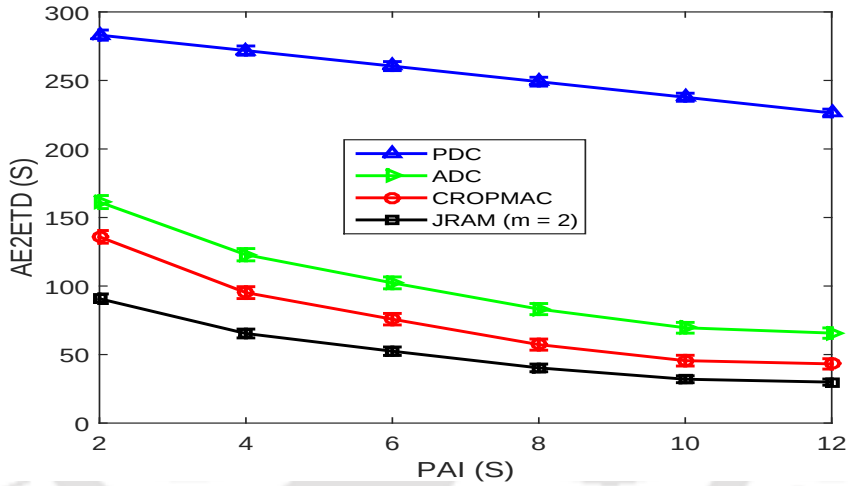
5.7.3 Comparison of PDR

Fig. 5.12 (a), 5.12 (b), and 5.12 (c) show comparison of PDR of PDC, ADC, CROP-MAC, and JRAM ($m = 2$) in case of M = 2, 3, and 4, respectively. In each case, the PDR of JRAM ($m = 2$) is highest. This happens because in JRAM ($m = 2$), two sensor nodes can send data packets to the sink in each cycle. As a result, in each cycle, the sink receives more data packets than PDC, ADC and CROP-MAC. Moreover, the PDR of each protocol reduces with the reduction in PAI as well as with the increase in M. The reason for this is that the reduction in PAI and the increase in M both increase the total number of data packets generated in a cycle while, in each protocol, the sink can receive a fixed number of data packets per cycle.

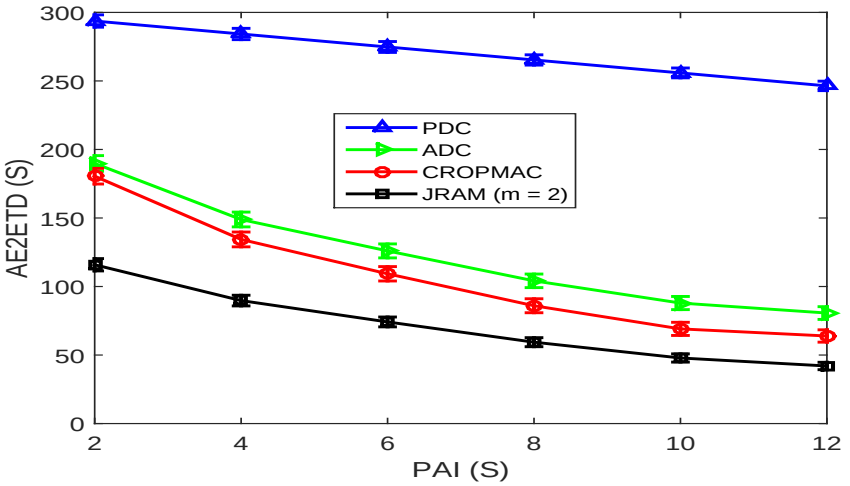
As in JRAM ($m = 2$), in CROP-MAC and ADC, a sensor node can transmit and/or receive multiple data packets per cycle. As a result, PDR of CROP-MAC and ADC is higher than the PDC. Further, in each case, PDR of CROP-MAC is higher than the ADC. The reason for this is that, unlike PDC and ADC, in CROP-MAC, the scheduling and the synchronization processes are accommodated in two different time windows. In Fig. 12 (a), 12 (b), and 12 (c), at PAI = 2.0s, PDR of JRAM ($m = 2$) is respectively (almost) 13.0%, 11.0%,and 11.0% more than that of CROP-MAC.



(a)

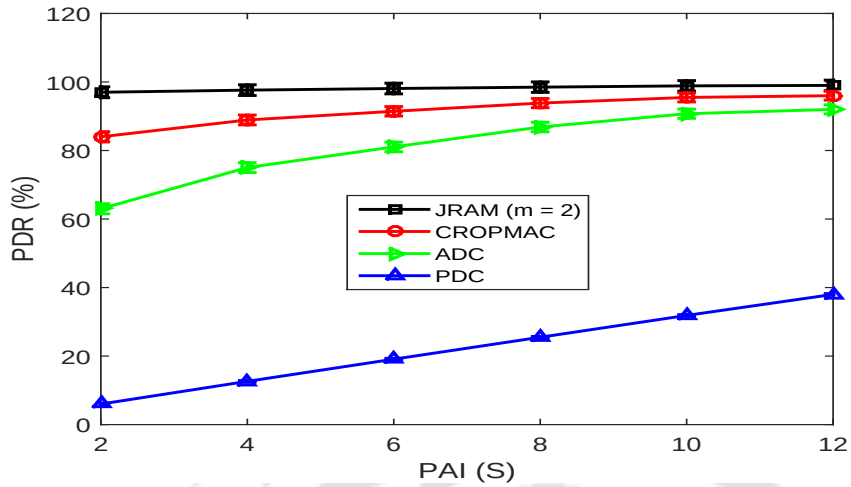


(b)

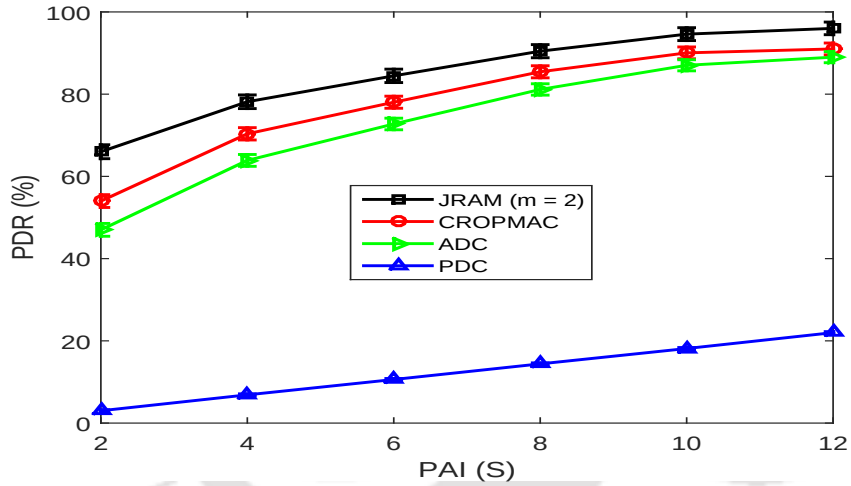


(c)

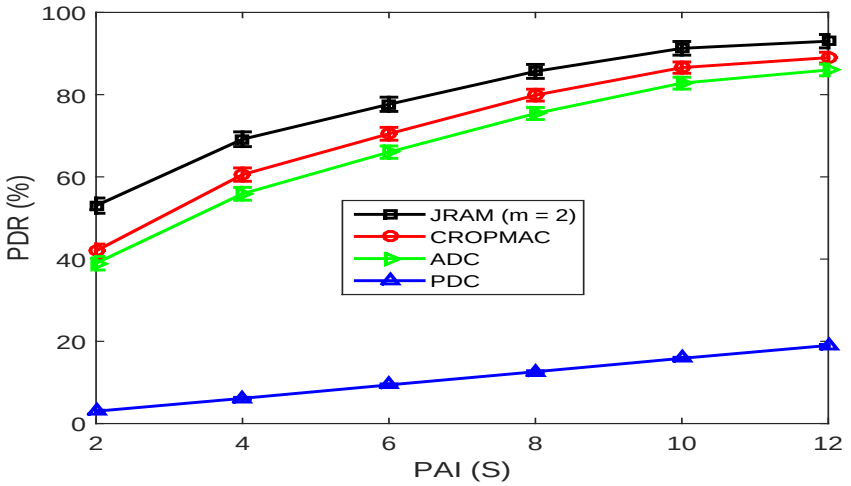
Figure 5.11: Comparison of E2ETD in case of (a) $M = 2$, (b) $M = 3$, and (c) $M = 4$.



(a)

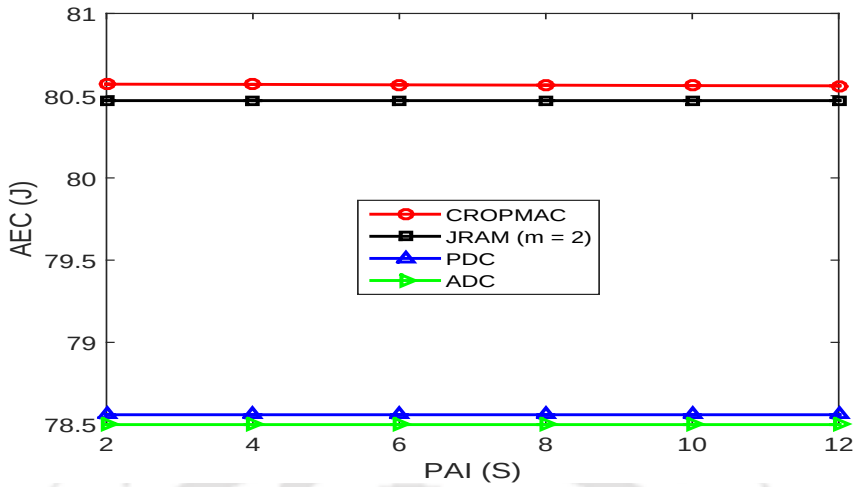


(b)

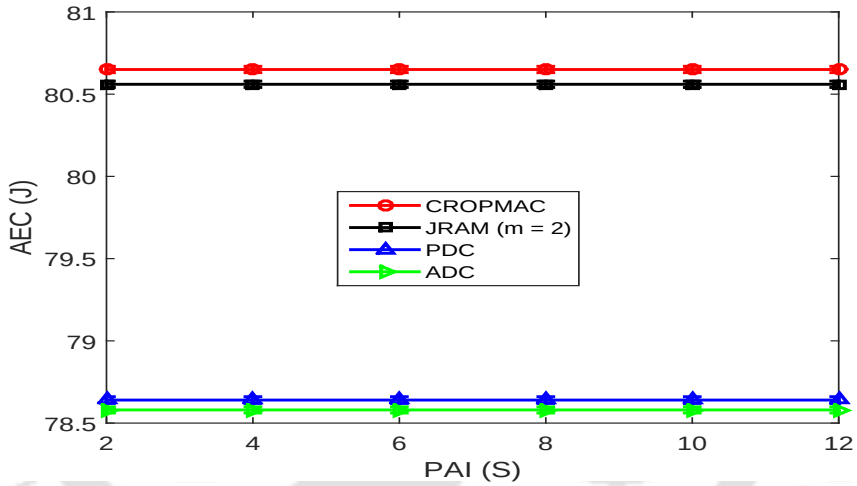


(c)

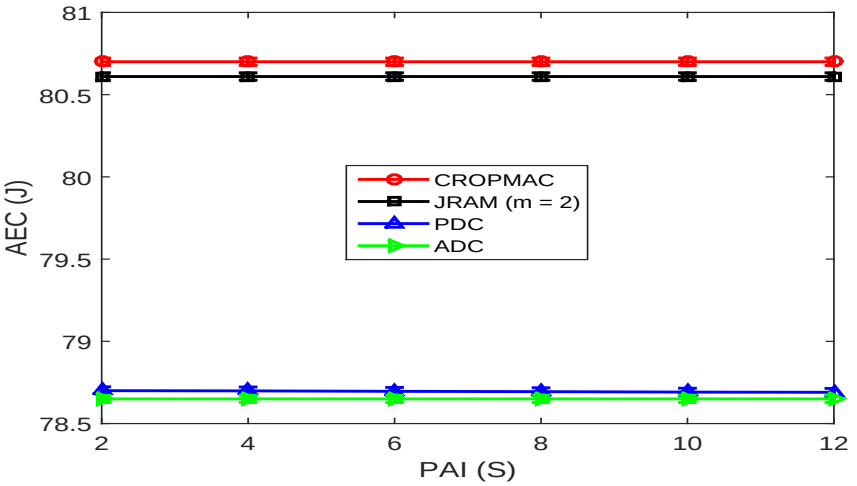
Figure 5.12: Comparison of PDR in case of (a) $M = 2$, (b) $M = 3$, and (c) $M = 4$.



(a)



(b)



(c)

Figure 5.13: Comparisons of AEC in case of (a) $M = 1$, (b) $M = 2$, and (c) $M = 3$.

5.7.4 Comparison of AEC

Fig. 5.13 (a), 5.13 (b), and 5.13 (c) show comparisons of the average energy consumption (AEC) of PDC, ADC, CROP-MAC and JRAM ($m = 2$) in case of $M = 1$, 2, and 3, respectively. In all the three cases, the PAI is varied from 2.0s to 12.0s in steps of 2.0s. AEC is defined as the total energy consumed during the simulation divided by the total number of nodes in the network. In case of $M = 1$, the source node generates 100 data packets with each PAI. While, in case of $M = 2$ and $M = 3$, each source node generates 50 and 33 data packets, respectively, with each PAI.

In all the three cases, at each PAI, AEC of JRAM ($m = 2$) and CROP-MAC are higher than the PDC and ADC. The reason is that, in JRAM ($m = 2$) and CROP-MAC, separate time windows are used to accommodate the scheduling process and the synchronization process. This results in an increase in AEC due to increase in *idle listening* duration of a node in a cycle. A possible reason for the difference in the AEC of PDC and ADC is the collision reduction in ADC due to the adaptive duty cycle feature. In addition to this, in case of each protocol, AEC increases with the increase in M. This happens because the increase in M results in increasing collisions. Moreover, in all the three cases, it can also be observed that the AEC of CROP-MAC is marginally higher than that of JRAM ($m = 2$) at each PAI. This happens due to the following two reasons: 1) In JRAM ($m = 2$), size of SW is less than that of CROP-MAC, as the node address ($\langle AS, G, SG, DSI \rangle$ sequence) overcomes the need of an additional field to accommodate the current *grade* of the sender and; 2) In JRAM ($m = 2$), DW size is also small compared to that of CROP-MAC as the control packet RTFD is smaller in size than the control packet RTSD [64]. In Fig. 5.13 (a), at PAI = 6.0s, AEC of PDC is 2.37% less compared to that of JRAM ($m = 2$). It is also observed that the difference in AEC of JRAM ($m = 2$) and PDC in case of $M > 1$ is almost equal to that in case of $M = 1$. This small increase in AEC in JRAM ($m = 2$), as compared to ADC, is insignificant in comparison to the significant gain in AE2ETD (Section 5.7.3) and PDR (Section 5.7.4).

5.8 Conclusions

In this chapter, we have proposed JRAM, a joint routing and MAC protocol for E2ETD and PDR improvement in many-to-one communication scenario for WSNs. JRAM exploits the dense deployment of sensor nodes in the network to improve the E2ETD and PDR. For this, JRAM uses a novel cycle structure which provides a sensor node m chances for data transmission scheduling in each cycle and

enables a sink node to receive data packets from m different sensor nodes in each cycle. This cycle structure also allows a sensor node to transmit multiple data packets in each cycle. These features of proposed cycle structure results in the increase in the number of sensor nodes that can forward their data packets in the same cycle when multiple sensor nodes with data packets to send lie in the carrier sensing range (CSR) of each other. Through extensive ns-2.35 based simulations, it is observed that, in case of $m = 2$, JRAM provides lowest E2ETD and highest PDR. To reduce the energy-consumption in *idle listening*, in JRAM, sensor nodes are partitioned into m disjoint sets (DSs) and sleep/wake schedules to the sensor nodes are assigned such that a sensor node can schedule the reception of data packets with at the most one sensor node in each cycle.

In JRAM, each node periodically broadcasts the information required for topology and synchronization maintenance, and to quickly accommodate a node deployed after setup phase in the system. Moreover, JRAM reduces the energy consumption in periodic broadcast of such information by making a particular sequence of information usable as the address of the node, with the help of the proposed distributed algorithms. Our ns-2.35 simulator based simulation results suggest that JRAM is a better solution as an integrated MAC and routing protocol to fulfil the crucial requirement of low E2ETD in reporting delay sensitive events to the sink in many-to-one communication scenarios.

6

Discussion and Future Work

Contents

6.1 Summary of Contributions and Discussions 110
6.2 Suggestions for Future Work 111

In this thesis, we address issues concerning WSNs used for hazard monitoring applications in unsafe regions. We propose low delay and low energy contention based synchronous MAC protocols for delay-sensitive event monitoring in dense multi-hop WSNs. This chapter presents the summery of the thesis and also discusses some of the possible future directions in which this work can be extended. The main contributions of this thesis are summarized in Section 6.1 and possible future work are outlined in Section 6.2.

6.1 Summary of Contributions and Discussions

In Chapter 3, we proposed LDC-MAC, a low delay cross-layer MAC protocol for *k-covered* event-driven WSNs. LDC-MAC improved E2ETD and PDR by exploiting the fact that, in an event-driven low duty-cycled WSN, most of the time a sensor has fewer number of data packets to send than what it can send in its DTS. LDC-MAC has the following two unique features. First, it assigns an interference-free DRS to each node of every flow scheduled in a DW. Second, it allows a node to utilize the unused portion of its neighboring nodes' DRS to forward its data packets if it failed to win the contention for medium access in DW. These features allow multiple nodes to forward their data packets through the same scheduled flow. As a result of this, more number of nodes can forward their data packets in the same cycle when multiple nodes with data packets to send lie in the CR of each other. LDC-MAC is analyzed through the extensive ns-2.35 based simulations for the event-driven applications in the dense multi-hop WSNs. Result showed that LDC-MAC provides significantly low E2ETD and high PDR than the CL-MAC at the cost of a very small increase in the energy-consumption.

In Chapter 4, we proposed a framework to improve the E2ETD and the PDR performances of the existing *generic* contention based synchronous MAC protocols in an event-driven dense multi-hop WSN. In this framework, in the network Setup Phase, sensor nodes are partitioned into m disjoint sets (DSs) such that the node density corresponding to each DS remains almost same and uniform throughout the network area. A new cycle structure is followed by the network nodes in the data transmission phase. This cycle structure contains m equal size time segments (TSs) and one SW where each TS contains one DW and one SlpW. A sensor node is allowed to send its data packets in each TS. As a result of this, m cycles of the data transmission process of the implemented *generic* contention based synchronous MAC protocol are mapped in one (same) duration cycle. To reduce *idle listening*, this framework allows a sensor node to receive data packets only in one TS in each cycle. This

framework easily accommodates the effects of node failure/deletion and new node insertion without re-executing the network setup phase. An analytical technique is proposed to determine the m for the optimum E2ETD in a given network scenario. The proposed analytical technique is validated through extensive ns-2.35 based simulations in different network scenarios. We tested the performance of our proposed framework for LDC-MAC and three other existing cross-layer *generic* contention based synchronous MAC protocols using the ns-2.35 simulator. Simulation results showed that in an event-driven dense multi-hop WSN, the proposed framework significantly improves the E2ETD and PDR of each implemented protocol, at the cost of a small increase in the AEC.

In Chapter 5, we proposed JRAM, a joint routing and MAC protocol for transmission delay reduction in many-to-one communication scenarios for event-driven WSNs. JRAM is a *staggered* contention based synchronous MAC protocol. In the network Setup Phase, JRAM assigns four parameters to each network node for partitioning the sensor nodes into m disjoint sets (DSs) and *stagger* the wakeup schedules of adjacent *grade* nodes. (*Grade* represents the minimum number of hops a packet needs to travel from the node before arriving at one of the sink nodes.) These parameters also help a node to determine its next-hop forwarders without using a separate routing protocol. These parameters are assigned such that the nodes lying in the CR of each other can have at the most three parameters value the same. However, the nodes that do not lie in the CR of each other may have all the four parameters value same. In data transmission phase, each node uses a sequence of its current parameters as its address for unicast communication. This address reusability feature reduces the size of the address field. JRAM used a new cycle structure which provides m chances to a node to succeed in data transmission scheduling in a cycle and enables a sink node to receive data packets from m different sensor nodes in the same cycle. The partitioning of sensor nodes into m DSs helps in reducing the *idle listening* period of a sensor node in a cycle. Our result showed that in an event-driven dense multi-hop WSN, the JRAM provides significantly low E2ETD and high PDR, in comparison to the existing *staggered* contention based MAC protocols.

6.2 Suggestions for Future Work

In this thesis we have addressed some of the basic issues related to the application of contention based synchronous MAC protocols for delay-sensitive event monitoring in dense multi-hop WSNs. The work reported in this thesis can be extended in following directions.

The techniques proposed in this thesis are evaluated only through the ns-2.35 based simulations. A discrete time Markov chain (DTMC) model can be developed for each technique as an alternate method to analyze the performance.

In this thesis we evaluated our proposed techniques only for event monitoring applications. Their suitability can also be analyzed for data gathering applications.

In Chapter 4, the analytical technique proposed for optimum m determination is developed assuming that the transmission range of each sensor node is same. Therefore, this technique may not be suitable for optimum m determination in a heterogeneous WSN where all the sensor nodes do not have same communication range (CR). By developing an analytical technique for optimum m determination in heterogeneous WSNs, the proposed framework can be utilized for performance enhancement of *generic* contention based synchronous MAC protocols in heterogeneous WSNs.

JRAM's performance can be further improved by allowing the multiple sensor nodes to forward their data packets through the same scheduled flow. Extension of JRAM for heterogeneous WSNs could also be an interesting future work.

A

Determination of $\mathbf{A}^{\mathbf{R}}_{(0,\delta)} \cap \mathbf{A}_u$

In this section, we drive analytical expressions to evaluate the $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ which is required in Algorithm 5.3 of Chapter 5, for α' determination, and shown in Fig. 5.8. It can be observed that corresponding to each combination of $\gamma' \in (0,1]$, $u \in \left[1, \mathrm{U}_{\gamma'}\right]$ and $\delta \in [(u-1)\gamma' R, u\gamma' R]$, geometry of $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ can be represented by one of the six subfigures of Fig. A.1. We drive analytical expression to evaluate the $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ corresponding to each subfigure, and represent the expression of i^{th} case by $A_i(u)$. However, we omit the details of the integration involved in determining $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$. Let $\mathrm{C}^{d'}_{(e_1,e_2)}$ be the circle centered at (e_1, e_2) and having radius d'. In Fig. A.1, (x_1, y_1) and (x_4, y_1) are the intersection points of $\mathrm{C}^{R}_{(0,\delta)}$ and $\mathrm{C}^{u\gamma' R}_{(0,0)}$; and (x_2, y_2) and (x_3, y_2) are the intersection points of $\mathrm{C}^{R}_{(0,\delta)}$ and $\mathrm{C}^{(u-1)\gamma' R}_{(0,0)}$.

Case 1: For $R - \delta \geq u\gamma' R$, $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ is shown in Fig. A.1 (a) and can be given as,

$$A_1(u) = \pi\left(u^2 - (u-1)^2\right)\left(\gamma' R\right)^2. \tag{A.1}$$

Case 2: For $(u-1)\gamma' R \leq (R-\delta) < u\gamma' R$, $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ is shown in Fig. A.1 (b) and can be given as,

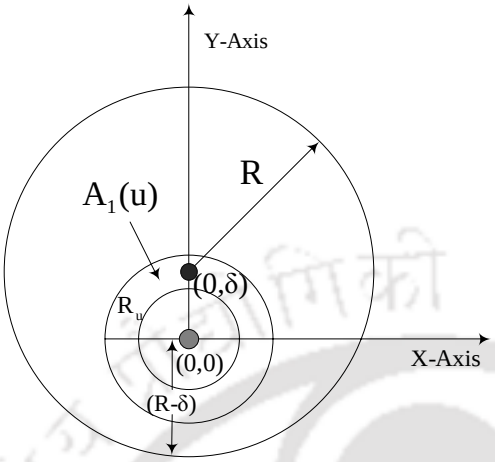
$$A_2(u) = \frac{A_1(u)}{2} + A_{22}(u), \tag{A.2}$$

where,

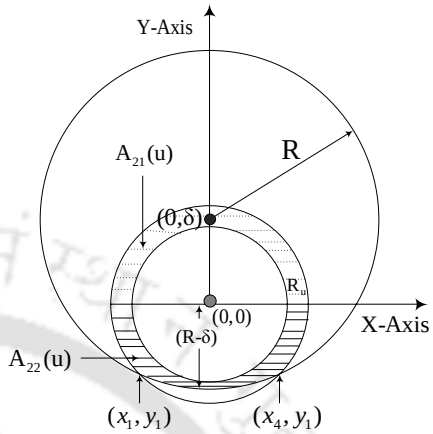
$$\begin{aligned} A_{22}(u) = &- x_4\sqrt{(u\gamma' R)^2 - x_4^2} - \left(u\gamma' R\right)^2 \sin^{-1}\left(\frac{x_4}{u\gamma' R}\right) \\ &- 2\delta x_4 + x_4\sqrt{R^2 - x_4^2} + R^2 \sin^{-1}\left(\frac{x_4}{R}\right) \\ &+ \frac{A_1(u)}{2}. \end{aligned} \tag{A.3}$$

Case 3: For $0 < (R-\delta) < (u-1)\gamma' R$ and $y_1 \leq 0$, $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ is shown in Fig. A.1 (c) and can be given as,

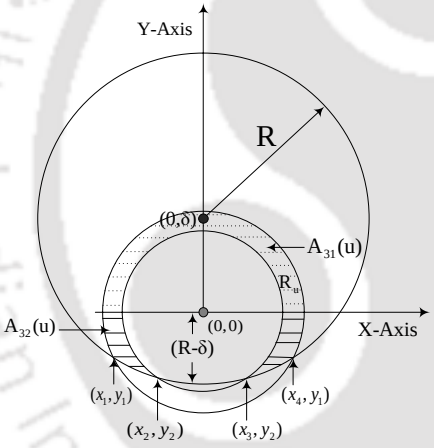
$$A_3(u) = \frac{A_1(u)}{2} + A_{32}(u), \tag{A.4}$$



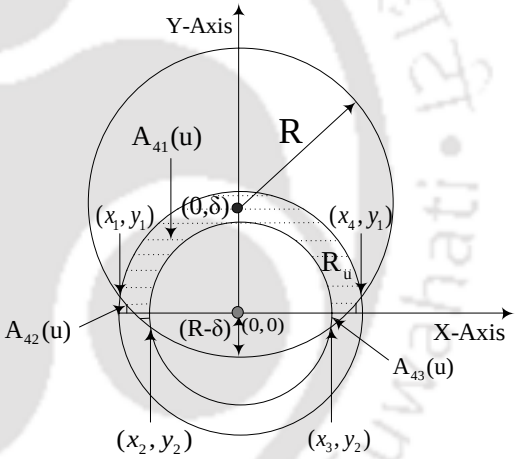
(a) Case1: $R - \delta \geq u\gamma' R$.



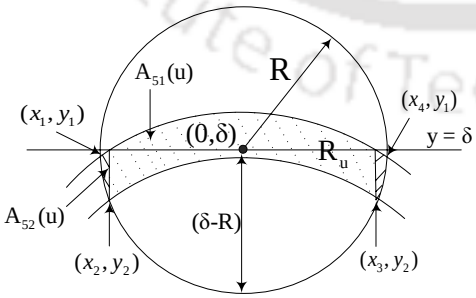
(b) Case2: $(u-1)\gamma' R \leq (R-\delta) < u\gamma' R$.



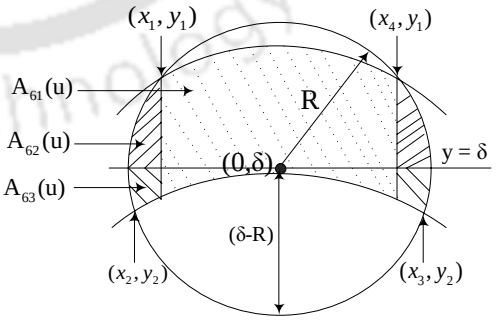
(c) Case3: $0 < (R-\delta) < (u-1)\gamma' R$ and $y_1 \leq 0$.



(d) Case4: $0 < (R-\delta) < (u-1)\gamma' R$, $y_1 > 0$, and $y_2 < 0$.



(e) Case5: $0 \leq y_2 \leq \delta$ and $y_1 \leq \delta$.



(f) Case6: $0 \leq y_2 \leq \delta$, and $y_1 > \delta$.

Figure A.1: Possible geometries of $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ corresponding to different values of δ.

where

$$A_{32}(u) = 2\delta x_3 - x_3\sqrt{R^2 - x_3^2} - R^2\sin^{-1}\left(\frac{x_3}{R}\right) + x_3\sqrt{(u-1)^2(\gamma' R)^2 - x_3^2} + (u-1)^2(\gamma' R)^2\sin^{-1}\left(\frac{x_3}{(u-1)\gamma' R}\right) + A_{22}(u). \tag{A.5}$$

Case 4: For $0 < (R-\delta) < (u-1)\gamma' R$, $y_1 > 0$, and $y_2 < 0$, $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ is shown in Fig. A.1 (d) and can be given as,

$$A_4(u) = \frac{A_1(u)}{2} + 2\left(A_{43}(u) - A_{42}(u)\right), \tag{A.6}$$

where

$$2\left(A_{43}(u) - A_{42}(u)\right) = A_{32}(u) + 2x_4\sqrt{(u\gamma' R)^2 - x_4^2} - \pi(u\gamma' R)^2 + 2(u\gamma' R)^2\sin^{-1}\left(\frac{x_4}{u\gamma' R}\right). \tag{A.7}$$

Case 5: For $0 \le y_2 < \delta$, and $y_1 \le \delta$, $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ is shown in Fig. A.1 (e) and can be given as,

$$A_5(u) = A_{51}(u) + 2A_{52}(u). \tag{A.8}$$

$$A_5(u) = A_4(u) + \pi(u-1)^2(\gamma' R)^2 - 2x_3\sqrt{(u-1)^2(\gamma' R)^2 - x_3^2} - 2(u-1)^2(\gamma' R)^2\sin^{-1}\left(\frac{x_3}{(u-1)\gamma' R}\right). \tag{A.9}$$

Case 6: For $0 \le y_2 < \delta$, and $y_1 > \delta$, $\mathrm{A}^{\mathrm{R}}_{(0,\delta)} \cap \mathrm{A}_u$ is shown in Fig. A.1 (f) and can be given as,

$$A_6(u) = A_{61}(u) + 2\left(A_{62}(u) + A_{63}(u)\right). \tag{A.10}$$

$$A_6(u) = A_5(u) - 2x_4\sqrt{R^2 - x_4^2} - 2R^2\sin^{-1}\left(\frac{x_4}{R}\right) + \pi R^2. \tag{A.11}$$

Bibliography

[1] S. Kulkarni, A. Iyer, and C. Rosenberg, "An address-light integrated MAC and routing protocol for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 4, pp. 793-806, Aug. 2006.

[2] B.-D. Lee and K.-H. Lim, "An energy-efficient hybrid data-gathering protocol based on the bynamic switching of reporting schemes in wireless sensor networks," *IEEE Syst. J.*, vol. 6, no. 3, pp. 378-387, Sept. 2012.

[3] Y. Hu, Y. Niu, J. Lam, and Z. Shu, "An energy-efficient adaptive overlapping clustering method for dynamic continuous monitoring in WSNs," *IEEE Sens. J.*, vol. 17, no. 3, pp. 834-847, Feb. 2017.

[4] N. Bouabdallah, M. E. Rivero-Angeles, and B. Sericola, "Continuous monitoring using event-driven reporting for cluster-based wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3460-3479, Sept. 2009.

[5] P. Huang, C. Wang, and L. Xiao, "RC-MAC: A receiver-centric MAC protocol for event-driven wireless sensor networks," *IEEE Trans. Computers*, vol. 64, no. 4, pp. 1149-1161, Apr. 2015.

[6] W. Shen, T. Zhang, F. Barac, and M. Gidlund, "PriorityMAC: A priority-enhanced MAC protocol for critical traffic in industrial wireless sensor and actuator networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 824-835, Feb. 2014.

[7] P. Suriyachai, U. Roedig, and A. Scott, "A survey of MAC protocols for mission-critical applications in wireless sensor networks," *IEEE Commun. Surv. Tuts.*, vol. 14, no. 2, pp. 240-264, 2012.

[8] L. A. Klein, "A boolean algebra approach to multiple sensor voting fusion," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 2, pp. 317-327, Apr. 1993.

[9] D. Nicules and B. Nath, "Ad-hoc positioning system (APS) using AoA," *in Proc. IEEE INFOCOM*, 2003, pp. 1734-1743.

[10] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 115-121, Apr. 2006.

[11] M. Doudou, D. Djenouri, and N. Badache, "Survey on latency issues of asynchronous mac protocols in delay-sensitive wireless sensor networks," *IEEE Commun. Surv. Tuts.*, vol. 15, no. 2, pp. 528-550, 2013.

[12] S. S. Ray, I. DEmirkol, and W.Heinzelman "ADV-MAC: Advertisement-based MAC protocol for wireless sensor networks," *in Proc. MSN*, 2010, pp. 265-272.

[13] "Crossbow Technology Inc," [online] Available: http://www.xbow.com/.

[14] "Sentilla Corporation," [online] Available: http://www.sentilla.com/.

[15] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Commun. Surv. Tuts.*, vol. 12, no. 2, pp. 222-248, 2010.

[16] W. L. Lee, A. Datta, and R. Cardell-Oliver, "FlexiTP: A flexible-schedule-based TDMA protocol for fault-tolerant and energy-efficient wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 6, pp. 851-864, June 2008.

[17] C. Shanti and A. Sahoo, "DGRAM: A delay guaranteed routing and MAC protocol for wireless sensor networks," *IEEE Mobile Comput.*, vol. 9, no. 10, pp. 1407-1423, Oct. 2010.

[18] G. M. Shafiullah, A. Thompson, P. J. Wolfs, and S. Ali, "Energy-efficient TDMA MAC protocol for wireless sensor networks applications," *in Proc. ICCIT*, 2008, pp. 85-90.

[19] G. M. Shafiullah, S. A. Azad, and A. B. M. S. Ali, "Energy-efficient wireless MAC protocols for railway monitoring applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 649-659, June 2013.

[20] T.-H. Hsu and P.-Y. Yen, "Adaptive time division multiple access-based medium access control protocol for energy conserving and data transmission in wireless sensor networks," *IET Commun.*, vol. 5, no. 18, pp. 2662-2672, Dec. 2011.

[21] W. Zhao and X. Tang, "Scheduling sensor data collection with dynamic traffic patterns," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 789-802, Apr. 2013.

[22] J. Ma, W. Lou, Y. Wu, X.-Y. Li, and G. Chen, "Energy efficient TDMA sleep scheduling in wireless sensor networks," *in Proc. IEEE INFOCOM*, 2009, pp. 630-638.

[23] S. Ray, I. Demirkol, and W. Heinzelman, "Supporting bursty traffic in wireless sensor networks through a distributed advertisement-based TDMA protocol (ATMA)," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 959-974, May 2013.

[24] A. N. Alvi, S. H. Bouk, S. H. Ahmed, M. A. Yaqub, M. Sarkar, and H. Song, "BEST-MAC: Bitmap-assisted efficient and scalable TDMA-based WSN MAC protocol for smart cities," *IEEE Access*, vol. 4, pp. 312-322, 2016.

[25] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," *in Proc. SenSys*, 2004, pp. 95-107.

[26] A. Bachir, D. Barthel, M. Heusse, and A. Duda, "Micro-frame preamble MAC for multihop wireless sensor networks," *in Proc. IEEE ICC*, 2006, pp. 3365-3370.

[27] X. Shi and G. Stromberg, "SyncWUF: An ultra low-power MAC protocol for wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 1, pp. 115-125, Jan. 2007.

[28] E.-Y. A. Lin, J. M. Rabaey, and A. Wolisz, "Power-efficient rendez-vous schemes for dense wireless sensor networks," *in Proc. IEEE ICC*, 2004, pp. 3769-3776.

[29] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," *in Proc. SenSys*, 2006, pp. 307-320.

[30] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, and P. Pinto, "A wireless sensor MAC protocol for bursty data traffic," *in Proc. IEEE PIMRC*, 2007, pp. 1-5.

[31] A. E.-Hoiydi and J.-D. Decotignie, "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks," *in Proc. ALGOSENSORS*, 2004, pp. 18-31.

[32] R. Musaloiu-E, C.-J. M. Liang, and A. Terzis, "Koala: Ultra-low power data retrieval in wireless sensor networks," *in Proc. IPSN*, 2008, pp. 421-432.

[33] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," *in Proc. SenSys*, 2008, pp. 1-14.

[34] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," *in Proc. SenSys*, 2010, pp. 1-14.

[35] P. Huang, C. Wang, L. Xiao, and H. Chen, "RC-MAC: A receiver-centric medium access control protocol for wireless sensor networks," *in Proc. IWQoS*, pp. 1-9, 2010.

[36] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," *in Proc. IEEE INFOCOM*, 2011, pp. 1305-1313.

[37] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," *in Proc. IEEE INFOCOM*, 2002, pp. 1567-1576.

[38] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493-506, June 2004.

[39] H. Liu, G. Yao, J. Wu, and L. Shi, "An adaptive energy-efficient and low-latency MAC protocol for wireless sensor networks," *J. Commun. Netw.*, vol. 12, no. 5, pp. 510-517, Oct. 2010.

[40] T. V. Dam and K. Langendoen, "An adaptive energy efficient MAC protocol for wireless sensor networks," *in Proc. SenSys*, 2003, pp. 171-180.

[41] S. Ray, I. Demirkol, and W. Heinzelman, "ADV-MAC: Analysis and optimization of energy efficiency through data advertisements for wireless sensor networks," *Ad Hoc Netw.*, vol. 9, no. 5, pp. 876-892, July 2011.

[42] S. Du, A. K. Saha, and D. Johnson, "RMAC: A routing enhanced duty cycle MAC protocol for wireless sensor networks," *in Proc. IEEE INFOCOM*, 2007, pp. 1478-1486.

[43] K.-T. Cho and S. Bahk, "Duty cycle optimization for a multi hop transmission method in wireless sensor networks," *IEEE Commun. Lett.*, vol. 14, no. 3, pp. 269-271, March 2010.

[44] J. Kim, K. Park, and D. Park, "An energy-efficient scheduling MAC protocol for wireless sensor networks," *in Proc. IEEE CCNC*, 2007, pp. 655-659.

[45] K. Nguyen and Y. Ji, "LCO-MAC: A low latency low control overhead MAC protocol for wireless sensor networks," *in Proc. MSN*, 2008, pp. 271-275.

[46] K. Nguyen and Y. Ji, "AM-MAC: an energy efficient, adaptive multi-hop mac protocol for sensor networks," *in Proc. IWCMC*, 2010, pp. 432-436.

[47] K.-T. Cho and S. Bahk, "HE-MAC: Hop extended MAC protocol for wireless sensor networks," *in Proc. IEEE GLOBCOM*, 2009, pp. 1-6.

[48] T. Canli and A. Khokhar, "PRMAC: Pipelined routing enhanced MAC protocol for wireless sensor networks," *in Proc. IEEE ICC*, 2009, pp. 1-5.

[49] S. Khaliq and S. Henna, "HE-PRMAC: Hop extended pipelined routing enhanced MAC protocol for wireless sensor networks," *in Proc. INTECH*, 2016, pp. 392-397.

[50] T. Canli, M. Hefeida, and A. Khokhar, "BulkMAC: A cross-layer based MAC protocol for wireless sensor networks," *in Proc. IWCMC*, 2010, pp. 442-446.

[51] Y. Sun, S. Du, O. Gurewitz, and D.B. Johnson, "DW-MAC: a low latency energy efficient demand-wakeup MAC protocol for wireless sensor networks," *in Proc. MobiHoc*, 2008, pp. 53-62.

[52] K. Nguyen and Y. Ji, "Achieving Minimum Latency in Multi-Hop MAC Protocol for Wireless Sensor Networks," *in Proc. IEEE VTC spring*, 2011, pp. 1-5.

[53] K. Nguyen, U. Meis, and Y. Ji, "An energy efficient high throughput MAC protocol using packet aggregation," *in Proc. IEEE GLOBECOM*, 2011, pp. 1236-1240.

[54] Y.Z. Zhao, M. Ma, C.Y. Miao, and T.N. Nguyen, "An energy-efficient and low-latency MAC protocol with adaptive scheduling for multi-hop wireless sensor networks," *Comput. Commun.*, vol. 33, no. 12, pp. 1452-1461, July 2010.

[55] M. S. Hefeida, T. Canli, and A. Khokhar, "CL-MAC: A cross-layer MAC protocol for heterogeneous wireless sensor networks," *Ad Hoc Netw.*, vol. 11, no. 1, pp. 213-225, Jan. 2013.

[56] G. Lu, B. Krishanamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," *in Proc. IPDPS*, 2004, pp. 224-231.

[57] A. G. Ruzzelli, G. M. Ohare, and R. Jurdak, "MERLIN: Cross-layer integration of MAC and routing for low-duty cycle sensor networks," *Ad Hoc Netw.*, vol. 6, no. 8, pp. 1238-1257, Nov. 2008.

[58] Y. Cao, S. Guo, and T. He, "Robust multi-pipeline scheduling in low duty-cycled wireless sensor networks," *in Proc. IEEE INFOCOM*, 2012, pp. 361-369.

[59] F. Tong, W. tang, R. Xie, L. Shu, and Y. Kim, "PMAC: A cross-layer duty cycle MAC protocol towards pipelining for wireless sensor networks," *in Proc. ICC*, 2011, pp. 1-5.

[60] F. Tong, M. Ni, L. Shu, and J. Pan, "A pipelined-forwarding, routing-integrated and effectively-identifying mac for large-scale wsn," *in Proc. IEEE GLOBCOM*, 2013, pp. 225-230.

[61] H. Khanh, C. Ock, and M. Kim, "RP-MAC: A cross-layer duty cycle MAC protocol with a reduced pipelined-forwarding feature for wireless Sensor Networks," *in Proc. IWCMC*, 2015, pp. 1469-1474.

[62] F. Tong, R. Zhang, and J. Pan, "One handshake can acheive more: An energy-efficient, practical pipelined data collection for duty-cycled sensor networks," *IEEE Sens. J.*, vol. 16, no. 9, pp. 3308-3322, May 2016.

[63] F. Tong and J. Pan, "Adaptive data collection with free addressing and dynamic duty-cycling for sensor networks," *Mobile Network and Applications*, vol. 22, no. 5, pp. 983-994, Oct. 2017.

[64] R. Singh and S. Chouhan, "A cross-layer MAC protocol for contention reduction and pipelined flow optimization in wireless sensor networks," *in Proc. IEEE RETIS*, 2015, pp. 58-63.

[65] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *IEEE/ACM Trans. on Netw.*, vol. 16, no. 3, pp. 511-524, June 2008.

[66] H. Wang, X. Zhang, F. Nait-Abdesselam, and A. Khokhar, "Cross-layer optimized MAC to support multihop QoS routing for wireless sensor networks," *IEEE Trans. Veh. Technol.*, Vol.59, no.5, pp. 2556-2563, June 2010.

[67] M. Arifuzzaman, M. Matsumoto. and T. Sato, "An Intelligent Hybrid MAC With Traffic-Differentiation-Based QoS for Wireless Sensor Networks," *IEEE Sens. J.*, vol. 13, no. 6, pp. 2391-2399, June 2013.

[68] H. Le, J. van Eck, and M. Takizawa, "An efficient hybrid medium access control technique for digital ecosystems," *IEEE Trans. Ind. Electron.*, vol. 60, no. 3, pp. 1070-1076, Mar. 2013.

[69] L. Sitanayah, C. J. Sreenan, and K. N. Brown, "A hybrid MAC protocol for emergency response wireless sensor networks," *Ad Hoc Networks*, vol. 20, no. 9, pp. 77-95, Sept. 2014.

[70] M. H. S. Gilani, I. Sarrafi, and M. Abbaspour, "An adaptive CSMA/TDMA hybrid MAC for energy and throughput improvement of wireless sensor networks," *Ad Hoc Netw.*, vol. 11, no. 4, pp. 1297-1304, June 2013.

[71] N. S. Kurian and B. Priya, "EHMBA: An energy efficient hybrid MAC multihop broadcast protocol for asynchronous duty-cycled wireless sensor networks," *in Proc. ICICES*, 2014, pp. 1-6.

[72] M. Marti, B. Kusy, G. Simon, and A. Ldeczi, "The flooding time synchronization protocol," in Proc. SenSys, 2004, pp. 39-49.

[73] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with Glossy," *in Proc. IPSN*, 2011, pp. 73-84.

[74] K. S. Yildirim and A. Kantarci, "Time synchronization based on slow-flooding in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 244-253, Jan. 2014.

[75] K. S. Yildirim and O. Grcan, "Efficient Time Synchronization in a Wireless Sensor Network by Adaptive Value Tracking," *IEEE Trans. Wireless Commun.*, vol 13, no. 7, pp. 3650-3664, July 2014.

[76] H. Huang, J. Yun, and Z. Zhong, "Scalable clock synchronization in wireless networks with low-duty-cycle radio operations," *in Proc. IEEE INFOCOM*, 2015, pp. 1-9.

[77] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," *in Proc. SenSys*, 2003, pp. 138-149.

[78] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," *in Proc. IEEE Decision Control*, 2007, pp. 2289-2294.

[79] P. Sommer and R. Wattenhofer, "Gradient clock synchronization in wireless sensor networks," *in Proc. IPSN*, 2009, pp. 37-48.

[80] L. Schenato and F. Fiorentin, "Average timesynch: A consensus-based protocol for time synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878-1886, Sept. 2011.

[81] X. Deng and Y. Yang, "Online adaptive compression in delay sensitive wireless sensor networks," *IEEE Trans. Computers*, vol. 61, no. 10, pp. 1429-1442, Oct. 2012.

[82] C. Caione, D. Brunelli, and L. Benini, "Compressive sensing optimization for signal ensembles in WSNs," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 382-392, Feb. 2014.

[83] N. Q. V. Hung, H. Jeung, and K. Aberer, "An evaluation of model-based approaches to sensor data compression," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2434-2447, Nov. 2013.

[84] M. Bagaa, Y. Challal, A. Ksentini, A. Derhab, and N. Badache, "Data aggregation scheduling algorithms in wireless sensor networks: solutions and challenges," *IEEE Commun. Surv. Tuts.*, vol. 16, no. 3, pp. 1339-1368, 2014.

[85] R. Rajagopalan and P. K. Varshney, "Data-aggregation techniques in sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 8, no. 4, pp. 48-63, 2006.

[86] M. Munz, M. Mhlisch, and K. Dietmayer, "Generic centralized multi sensor data fusion based on probabilistic sensor and environment models for driver assistance systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 1, pp. 6-17, 2010.

[87] S. Din, A. Ahmad, A. Paul, M. M. U. Rathore, and J. Gwanggil, "A cluster-based data fusion technique to analyze big data in wireless multi-sensor system," *IEEE Access*, vol. 5, pp. 5069-5083, 2017

[88] Y. Yuan and M. Kam, "Distributed decision fusion with a randomaccess channel for sensor network applications," *IEEE Trans. Instrum. Meas.*, vol. 53, no. 4, pp. 1339-1344, Aug. 2004.

[89] X. Zhai, H. Jing, and T. Vladimirova, "Multi-sensor data fusion in Wireless Sensor Networks for Planetary Exploration," *in Proc. AHS*, 2014, pp. 188-195.

[90] H. M. Ammari, "CSI: An energy-aware cover-sense-inform framework for k-covered wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 4, pp. 651-658, Apr. 2012.

[91] H. M. Ammari and S. Das, "A study of K-coverage and measures of connectivity in 3D wireless sensor networks," *IEEE Trans. Computers*, vol. 59, no. 2, pp. 243-257, Feb. 2010.

[92] K.-T. Cho and S. Bahk, "Optimal hop extended MAC protocol for wireless sensor networks," *Computer Netw.*, vol. 56, no. 4, pp. 1458-1469, Mar. 2012.

[93] M. Zorzi and R. R. Rao, "Geographic random forwarding (geraf) for ad hoc and sensor networks: Multihop performance," *IEEE Trans. Mobile Comput.*, vol. 2, no. 4, pp. 337-348, Oct. 2003.

[94] L. Cheng, J. Niu, J. Cao, S. Das, and Y. Gu, "Qos aware geographic opportunistic routing in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1864-1875, Jul. 2014.

[95] H. Huang, G. Hu, F. Yu, and Z. Zhang, "Energy-aware interference-sensitive geographic routing in wireless sensor networks," *IET Commun.*, vol. 5, no. 18, pp. 2692-2702, Dec. 2011.

[96] H. Zhang and H. Shen, "Energy-efficient beaconless geographic routing in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 6, pp. 881-896, June 2010.