Stochastic Approach to Quality of Service in Data Gathering for Wireless Sensor Networks

Thesis submitted in partial fulfilment of the requirements

for the award of the degree of

Doctor of Philosophy

by

Debanjan Sadhukhan

Under the supervision of

Prof. S. V. Rao

Department of Computer Science and Engineering

Indian Institute of Technology Guwahati

Guwahati - 781039 Assam India

SEPTEMBER, 2017

Dedicated to

My sister, my wife and my parents

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. S. V. Rao for his valuable guidance, inspiration, and advice. His support and encouragement generously paved the way for my development as a research scientist. I benefited greatly from many fruitful discussions with my supervisors, which has changed my personality, ability and nature in many ways. I feel very privileged to have had the opportunity to learn from, and work with him. I am highly grateful to my Doctoral Committee members, Prof. D. Goswami, Dr. T. Venkatesh, and Dr. P. S. Mandal. Their comments and suggestions have truly deepened and widened my understanding of the problems I had worked on. I am thankful to Ministry of Human Resource Development, Government of India, for awarding me the research fellowship that gave me extremely good opportunities to broaden my research activities. I would also like to express my heartfelt gratitude to the director, the deans and other management of IIT Guwahati whose collective efforts have made this institute a place for world-class education and research. I am thankful to all the faculty members and the staff of the Department of Computer Science and Engineering for extending their cooperation in terms of technical and official support. I am obliged to the research scholars, M. Tech and B. Tech students of this institute with whom I had closely worked. During my stay here in Guwahati, I met a few friends who reside close to my heart like Sanjay, Deepak, Mayank, Shilpa, Shashi. I also spent a quality amount of time with friends like Sonia, Tushar, Nilkanta, Shirshendu. Last but not the least, I would like to express my gratitude to my parents for their constant

support and encouragement. Their motivation, assistance and guidance helped me to find the path for my future life.

18 September 2017 Debanjan Sadhukhan

Declaration

I certify that

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

18 September 2017 Debanjan Sadhukhan

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati - 781039 Assam India

Dr. S. V. Rao
Professor
Email : svrao@iitg.ernet.in
Phone : +91-361-2582358

Certificate

This is to certify that the work contained in this thesis entitled "**Stochastic Approach to Quality of Service in Data Gathering for Wireless Sensor Networks**" submitted by **Debanjan Sadhukhan**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, is a bona fide research work of our supervision and is worthy of consideration for the award of the degree of Doctor of Philosophy of the Institute.

Prof. S. V. Rao
18 September 2017
(Thesis Supervisor)

Abstract

Wireless sensor networks (WSN) can be categorized into event-driven, periodic and hybrid based on data-gathering techniques. Depending on the application requirements, quality of service (QoS) can be referred as deployment cost, lifetime, delay, jitter, packet loss, bandwidth, throughout etc. Generally, in order to save energy a sensor node switches on the communication device only when it is expected to send or receive any data packets. This strategy is known as sleep/wake scheduling technique. In synchronous sleep/wake scheduling, sensor nodes wake up synchronously whereas in asynchronous sleep/wake scheduling sensor nodes wake up independently. In a rare event detection application, asynchronous sleep/wake scheduling is energy efficient because of overhead involved for synchronization in synchronous sleep/wake scheduling. As sensor nodes wake-up independently, asynchronous sleep/wake scheduling increases delay. For such an application, satisfying delay constraint and lifetime requirement with minimum deployment cost is a challenging problem. Specifically, in a rare event-detection application, like fire or disastrous event detection, an event detected close to the central facility needs to reach faster than an event detected far away. In other words, there can be a large variation in QoS requirements for different areas within the FoI. In the first phase of our thesis, we address the problem of finding a minimum cost WSN for a set of spatially differentiated QoS requirements. First, we use a stochastic approach to calculate the expected e2e delay for a given sensor density. We use this analysis

to find the critical expected sensor density that satisfies given e2e delay constraint and lifetime requirement. Later we extended this analysis to find the minimum cost network for a set of spatially differentiated delay constraints and the lifetime requirement.

In order to minimize the delay incurred in asynchronous sleep/wake scheduling, anycasting based forwarding techniques are proposed in the literature where a sensor node forwards its data to any one of the nodes in its forwarding set. In anycasting strategies, the delay constraint may not always be satisfied as a result of clock-skew[1]. This is because, the clock-skew incurs additional delay during event reporting which may violate the delay constraint for a large number of packets. We find the critical wake-up rate to constrain the increase in end-to-end delay, as a result of clock-skew, within a given delay constraint ξ, by estimating the expected increase in end-to-end delay using a stochastic approach. We verify our mathematical analysis using Monte-carlo simulation. Further, simulation results in network simulator 2.34 (ns2) confirm the effectiveness of our approach.

In the next phase of our work we focus on periodic data-gathering. In periodic data-gathering, sensors send/receive data packets periodically and switch on the transceiver only during packet transmission to save energy. In order to transfer data-packets a sender-receiver pair establishes clock-synchronization. Exact clock-synchronization is impossible because of error present in synchronization protocols. In the absence of synchronization, clock-disagreement increases as time passes. To compensate any message loss, the sensor nodes wake up some time earlier, known as guard-time, before the scheduled wake-up time. This strategy consumes more energy especially for long periodic intervals. In this phase of our thesis, in order

[1]Clock-skew refers to the difference in actual and expected crystal oscillator frequency.

to increase the lifetime, we propose a multi-beacon guard method to decrease the energy consumption by minimizing the awake time of sender-receiver pair, using periodic switch on/off the receiver during the guard-time. Under the assumption that synchronization error follows the normal-distribution, we provide an optimal solution to determine the number-of-times the receiver needs to wake-up along with wake-up intervals to collectively minimize total energy consumption of the sender-receiver pair during a transmission. We validate our analysis using Monte-Carlo simulation and show the effectiveness in energy conservation using ns2 simulation.

In a hybrid data-gathering, a sensor network dynamically switches between event driven and periodic data gathering scheme. Two key QoS requirements in hybrid data-gathering are lifetime and critical event-reporting delay. In the last phase of our work, we propose a novel hybrid data-gathering protocol that minimizes the critical event reporting delay and maximizes the lifetime in a delay-constrained WSN. In our protocol, the sensor nodes generate periodic events and if a critical event is detected, the event information is forwarded to the base station within a strict delay constraint. We show the effectiveness of our protocol in ns2 simulation and compare our approach with recent hybrid data-gathering protocols.

Contents

List of Figures

List of Algorithms

List of Tables

List of Symbols

Symbols	Description
$A(c, C)$	Communication region of a sensor located at c with communication radius C
C	Communication radius
$CA(b_p, R_i, R_j)$	Circular annulus: The area between the boundaries of two concentric circles centered at b_p with radii R_i, R_j such that $R_j > R_i$
C_D	Circle with radius C with center at base-station
$CI(dis_i, \delta)$	Circle Intersect: The intersection of a circle with radius $R_j = dis_i - \delta, 0 < \delta \leq C$, and the effective forwarding region of communication of node i
$CIL^l(dis_i, \delta)$	Left sub-part of $CI(dis_i, \delta)$ after cutting by line l
$CIR^l(dis_i, \delta)$	Right sub-part of $CI(dis_i, \delta)$ after cutting by line l
clk_i	Clock-skew rate of node i
Δ	End-to-end delay constraint
$D_{i,k,w}$	Expected end-to-end delay of node i
$\delta d_i(\vec{f})$	Increase in expected one-hop delay for clock-skew
$d_i(\vec{f})$	Expected one-hop delay of node i

dis_i	Distance of node i from the base-station
$D_{k,w}$	Expected end-to-end delay
$d_{k,w}$	Expected one hop delay
EC_i	Effective forwarding area of communication
$\|\|EC_i\|\|$	Area of effective forwarding region of communication
ECL_i^l and	Left sub-part of EC_i after cut by line l
E_{mb}	Total energy consumption during transmission
ECR_i^l and	Right sub-part of EC_i after cut by line l
E_{r_mb}	Energy consumption of the receiver during transmission
E_{s_mb}	Energy consumption of the sender during transmission
E_{sw}	Transition energy required from sleep to awake state
E_{txdata}	Energy required to transmit data-packet
E_{rxdata}	Energy required to receive data-packet
E_{txack}	Energy required to transmit acknowledgement packet
E_{rxack}	Energy required to receive acknowledgement packet
E_{rxbcn}	Energy required to receive beacon packet
E_{txbcn}	Energy required to transmit beacon packet
E_{T_g}	Expected energy consumption for guard-time approach
$f_i(h) = j$	Forwarding policy of nodes i at h^{th} beacon

F_i	Forwarding set of node i
f_i	Forwarding policy of node i
$\vec{f}$	Global forwarding vector
L	lifetime constraint
N_i	Set of neighbors of node i
N_r	Number of beacon interval
N_{T_t}	Number of sub-transmission intervals
p_w	Probability of a node wakes up at h^{th} beacon signal
$P(W)$	Probability of event W
P_{idle}	Power required for idle listening
$P(X = x)$	Probability of sender wakes up at time x
Q	Initial energy
$q_{i,j}$	Probability of packet is forwarded to node j
$SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$	The j^{th} sectoral annulus $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$ of node i, between two concentric circles, centered at b_p with radii $\beta_{i_{j1}}, \beta_{i_{j2}}$, such that $\beta_{i_{j1}} > \beta_{i_{j2}}$, is defined as the intersection of the area between their boundaries and EC_i
T_{rtt}	Average rtt time
Th_{mb}	Threshold for multi-beacon
T_e	Duration of epoch

T_s	Synchronization interval
T_t	Sub transmission interval
t_I	Duration of beacon interval
t_D	Delay to transmit one packet
τ_p	Scheduled arrival time of message p
τ'_p	Actual arrival time of message p
$[-T_g, T_g]$	Duration of guard-time
w	Periodic wake-up rate
W_s	Expected waiting time of sender
ξ	Increase in delay for clock-skew constraint

List of Abbreviations

Terms	Abbreviations
ADCs	Analog to digital converters
CDMA	Code division multiple access
CH	Cluster head
CSD	critical sensor density
e2e	end-to-end
EEPROM	Electrically erasable programmable random access memory
FoI	Field of Interest
ns2	Network simulator 2.34
MAC	Medium access control
MBG	Multi-Beacon Guard
Otcl	Object oriented tool command language
PDF	Probability density function
QoS	Quality of Service

RTT	Round trip time
RBS	Receiver broadcasting synchronization
s/w	Sleep/wake
SDRAM	Synchronous dynamic random access memory
SRAM	Static random access memory
TDMA	Time division multiple access
TPSN	Timing sync protocol for sensor networks
WSN	Wireless Sensor Network

1

Introduction

A wireless sensor network (WSN) consists of several tiny sensors powered by batteries which can communicate with each other to monitor a field of interest (FoI). Recently, WSN attracted attention because of the variety of applications it supports. The application set of WSN includes environmental monitoring (temperature, pressure and precipitation measurement), tsunami detection, forest-fire detection, seismic event detection, intrusion detection for battlefield surveillance, home and office intelligence, and medical care, etc. Mostly in hostile environment, where manual placement is not possible, the sensor nodes are scattered from helicopters, such deployment is called random[1] [49].

The main purpose of a WSN is to collect relevant data from the environment and report to the user according to the application requirement. Based on data-gathering schemes, WSNs can be categorized intro event-driven, periodic and hy-

[1]Also known as stochastic deployment.

brid [16, 47]. In event-driven data-gathering applications, like intrusion detection, tsunami detection, forest-fire detection, and many more, the sensor nodes remain idle unless and until a critical event occurs in the vicinity, and once detected, the event information is forwarded to the base-station as early as possible. Whereas, in periodic data-gathering applications, like environmental monitoring, the sensor nodes generate periodic data which are gathered at the base-station. The sensor nodes switch between event-driven and periodic data-gathering according to the requirement or application specification in hybrid data gathering applications [43]. Consider a forest monitoring system where the sensor nodes are deployed to monitor several physical phenomena like temperature, humidity, and many more using a periodic approach. However, if any critical event like fire breaks out within the forest the event information is forwarded to the base-station as early as possible in order to minimize the damage. Similar applications are earthquake and volcanic event detection for environmental monitoring.

Depending on the application requirements, the quality of service (QoS) can be referred as the degree of reliability, timeliness, robustness, availability, security etc. provided by the corresponding WSN [88]. The QoS requirements can be classified into application specific and network specific. The application specific QoS requirements are coverage, exposure, lifetime, and deployment cost. Whereas the network specific QoS requirements are connectivity, scalability, delay, packet loss, bandwidth, reliability, fault-tolerant, jitter and throughput etc [95]. Although we classify the QoS requirements into application and network specific, but data-gathering techniques have significant implications on QoS requirements. For instance, in applications involve periodic data-gathering, jitter and end-to-end (e2e) delay may not be required whereas in event-driven data-gathering e2e delay plays a significant

role. Hence, while designing QoS requirements for an application, data-gathering model (event-driven, periodic or hybrid) plays a significant role [23].

1.1 Motivation and Contribution

Generally, sensor nodes are equipped with a limited, unattended battery power source. Hence, maximizing the lifetime remains one of the key QoS requirements while designing any WSN [40, 62]. Lifetime refers the duration of the time when the first node depletes its energy completely within the network. It can be noted that if energy consumption increases, lifetime decreases and vice versa. One way of reducing energy consumption is using sleep/wake (s/w) scheduling techniques where the communication device is switched on/off periodically. These techniques can be categorized into synchronous and asynchronous. In synchronous s/w scheduling, sensor nodes exchange synchronization messages and wake up synchronously to send/receive data-packets, whereas in asynchronous s/w scheduling techniques, sensor nodes wake-up independently [46].

In rare event-detection applications, like fire or disastrous event detection, an event detected close to the central facility needs to reach faster than an event detected far away. In other words, there can be a large variation in QoS requirements for different areas within the FoI. High priority areas are associated with stringent QoS requirements, whereas low priority areas are with relaxed QoS requirements. Specifically, the delay constraints of the higher priority areas are very less than delay constraints of the distant areas, with a fixed lifetime requirement of the whole network. Delay constraint refers to the expected time duration, for the first packet that detects the event, between the event is detected and the event information

reaches to the base-station. In these applications, the sensors remain idle most of the time until a critical event is detected. Hence, asynchronous s/w scheduling techniques are considered to be more energy efficient than synchronous s/w scheduling techniques because of extra energy consumption for synchronization. As because sensor nodes wake up independently in asynchronous s/w scheduling, a node intends to send a packet needs to wait until the forwarding node wakes up, which may increase overall e2e delay. In order to decrease this delay anycasting forwarding strategy, each node maintains a forwarding set and the packet is forwarded to the first node wakes up within the set, is proposed in the literature. Compared to the basic forwarding strategy, where a node maintains a single designated next-hop relaying node, anycasting strategy decreases expected one hop waiting time which in turn decreases expected e2e delay. In anycasting forwarding strategy asynchronous periodic wake-up pattern, every node independently wakes up after a periodic interval, minimizes the e2e delay further compared to Poisson wake-up pattern [45],[23]. Hence, in this thesis we assume that sensor nodes follow asynchronous, periodic wake-up pattern to conserve energy in event-driven data-gathering.

In such a scenario density of nodes becomes the governing factor for e2e delay and lifetime. This is because if density increases, the size of the forwarding set may increase, which may decrease expected one hop delay. If expected one-hop delay decreases then expected e2e delay decreases as well. The problem of finding the critical sensor density (CSD), minimum density that satisfies given constraints, is well addressed in the literature, but in the context of satisfying coverage and connectivity requirements [96, 97, 50, 93, 90, 98, 14, 21, 28]. Moreover, proposed anycasting forwarding techniques [102, 56, 18, 68, 69, 46, 45] in the literature are

not applicable to find CSD for given delay and lifetime requirements.

Moreover, if the periodic wake-up interval of the nodes in the forwarding set decreases, expected one hop delay may decrease which decreases expected e2e delay, but decreases the lifetime as well. Increasing density increases the overall cost of the network, which is not acceptable always as the cost being one of the primary QoS requirements while designing any WSN, and vice versa. Density refers to the number of sensors present in a unit area. Assuming sensor nodes are deployed following a random distribution, in the first contribution of our thesis we look into the problem of finding the minimum cost network for given spatially differentiated delay constraints. Cost is defined as the factor of the expected number of sensors present in the FoI times cost of each sensor. The expected number of nodes present in the FoI is the factor of the density times the area of FoI. It can also be noted here that density refers to the number of sensors present in the unit area. For a uniform random deployment with density λ, the overall cost of the network is defined as $\lambda \times ||A|| \times C$ where $||A||$ and C respectively denotes the area of FoI and cost of each sensor. We look into the problem of finding a minimum cost network for spatially differentiated delay constraints and the lifetime requirement. In other words,

Problem 1: In event-driven data-gathering, how to minimize the deployment cost for a given set of spatially differentiated delay constraints and the lifetime requirement, when nodes follow anycasting forwarding strategy across the FoI?

We use a stochastic approach to calculate the expected e2e delay for a given sensor density in a stochastic deployment. We use this analysis to first find the critical expected sensor density that satisfies given e2e delay constraint and lifetime requirement in a circular-shaped FoI. Later we extended this analysis to find the

minimum cost network for spatially differentiated delay constraints and lifetime requirement. We also validated our analysis using Monte-Carlo simulation and show the effectiveness of our approach using ns2 simulation.

Generally, sensor nodes are associated with imperfect crystal oscillator and as a result of this, clock-disagreement is a common phenomenon. The most governing factors for clock-disagreement are phase-offset and clock-skew. Phase-offset denotes the clock difference (or time difference) between two sensors at an instant of time. Whereas clock-skew refers to the difference in actual and expected crystal oscillator frequency, and depends on environmental factors like temperature, pressure, radiation, magnetic fields, etc. It is measured in terms of parts-per-million (ppm), where one ppm denotes the clock drift of one micro second (μs) in a second. The clock-skew rate of Mica Motes is up to 50 ppm. A typical range of clock-skew is generally up to 40 to 100 ppm [85]. As a result of clock-skew, the actual periodic wake-up interval may vary over time, which in turn increases the overall end-to-end delay.

In order to mitigate the effect of clock-skew, time synchronous protocols [80, 73, 30, 34] are proposed in the literature. Generally in the time synchronization protocols, sensor nodes estimate expected phase-offset and expected clock-skew to synchronize time. However, clocks may diverge as a result of estimation error involved in synchronization. Moreover, several environmental factors like temperature, pressure, radiation, magnetic fields, etc. may vary the clock-skew. Frequent synchronization may reduce the clock divergence, but it is not energy efficient for the rare event detection scenario. Clock-skew may vary the actual periodic wake-up interval over time and packets may violate e2e delay constraint. None of the anycasting strategies [18, 68, 69, 46, 45] considered the effect of clock-skew in e2e

delay to the best of our knowledge, that may lead to violate the delay-constraint in time-critical applications. Existing anycasting strategies [18, 68, 69, 46, 45] not considered the effect of clock-skew in e2e delay to the best of our knowledge, that may lead to violate the delay-constraint in time-critical applications. This limitation in the literature motivates us to find the critical wake-up rate that constrains the overall increase in e2e delay.

Problem 2: What is the critical wake-up rate to constrain the increase in e2e delay, as a result of heterogeneous clock-skew present in WSN, within a given threshold ξ, for event driven data-gathering, where ξ denotes the given threshold for the amount of allowed e2e delay increased for clock-skew?

In this contribution, first we estimate the increase in e2e delay as a result of the heterogeneous clock-skew present in the network and then we use this estimation to find the critical wake-up rate for all nodes to constrain the increase in the overall e2e delay within given threshold ξ. We showed the effectiveness of our approach by comparing the results with a recent anycasting based forwarding strategy in network simulator 2.34 (ns2).

Synchronous s/w scheduling techniques are more energy efficient than asynchronous s/w scheduling techniques in periodic data gathering [85]. Exact clock synchronization is required for synchronous s/w scheduling, which is very difficult to achieve because of the imperfect crystal oscillator. In order to circumvent the effect of clock-skew, time synchronization protocols are applied. Sensor nodes synchronize time by calculating expected phase offset and expected clock-skew. At the end of time synchronization, clocks may diverge as a result of estimation error involved in synchronization which follows normal distribution with zero mean [85].

Frequent synchronization may reduce the clock divergence, but it is not energy efficient for application with a long periodic interval. In our third contribution, we maximize the lifetime of the network, for periodic data-gathering in presence of clock-skew.

As a result of inaccuracy and non-determinism present in time synchronization protocol, nodes wake up earlier than the scheduled wake up time, to circumvent any message loss, known as guard time [85, 86]. But the energy consumption due to guard-time increases as time passes, which in-turn minimizes the lifetime. Existing protocols [85, 86] use guard-time to circumvent message loss which is not energy-efficient.

In this work, we propose *Multi-Beacon Guard* (MBG) method, where a sensor node wakes up multiple times within the guard-time if the length of the guard-time is more than a threshold, otherwise follow the simple guard-time approach to conserve energy. We derive an expression for energy consumption in multi-beacon approach during a data-packet transmission. We use this analysis to find the optimal number of times the receiver wakes up and the wake-up intervals that minimize the expected energy consumption of a sender-receiver pair. Further results obtained from ns2 simulation confirm the effectiveness of our approach.

Note that in this work our objective is to maximize the lifetime of the sensor network in presence of clock-skew. In other words, we focus on the following problem.

Problem 3: How to maximize the lifetime of the network in periodic data-gathering in the presence of clock-skew?

In hybrid data-gathering applications, like environmental monitoring with fire, earthquake and volcanic event detection, and many more, two key QoS require-

ments are lifetime and critical event-reporting delay. As critical events occur rarely the lifetime is governed mostly by the energy consumption during periodic data-gathering. In order to maximize the lifetime the sensor nodes follow synchronous, periodic s/w scheduling technique during periodic data-gathering. Network lifetime increases as the wake-up rate decreases. However, it increases the critical event-reporting delay, which is not desirable for critical events. Increasing wake-up rate decreases the overall lifetime during periodic data-gathering but decreases the time-critical event reporting delay. Two key QoS requirements, in these type of applications, are maximizing the network lifetime during periodic data-gathering while minimizing the time-critical event reporting delay. Hence, in the last contribution of our thesis we address the problem of maximizing lifetime while minimizing the critical event reporting delay in hybrid data-gathering.

To the best of our knowledge, existing hybrid data-gathering protocols [59, 75, 51] do not address the problem of minimizing the critical event-reporting delay while maximizing the lifetime. Hence, in the final contribution of our thesis, we propose a protocol where the sensor nodes generate periodic events and if a critical event is detected, the event information is forwarded to the base-station in a fastest possible way. In order to show the effectiveness, we compare the results with a recent hybrid data-gathering protocol in ns2 simulation.

The problem we address in the fourth contribution can be summarized as,

Problem 4: How to maximize the lifetime and minimize the critical event-reporting delay in a hybrid data-gathering?

Overall, in this thesis we focus on improving these QoS requirements in event-driven, periodic and hybrid data-gathering. In other words, we use stochastic ap-

proach to improve several QoS requirements and we believe that these analysis motivate to derive many more theoretical framework that helps to improve QoS further.

1.2 Organization of the Thesis

The rest of the thesis is organized as follows. In the next chapter we first briefly discuss about WSNs. Next we address related works with respect to the QoS requirements in data-gathering for WSNs. In chapter 3, first we estimate expected e2e delay for given sensor density and next we use this analysis to find the minimum cost network for spatially differentiated QoS requirements. In the presence of clock-skew the e2e delay constraint may not be always satisfied. Hence, in the chapter 4 we estimate the expected increase in e2e delay as a result of clock-skew and use this information to find a critical wake-up rate that constrains this expected increase within given threshold ξ. We propose *Multi-Beacon Guard* (MBG) method that minimizes the energy consumption during a transmission in the presence of clock-skew by multiple times waking up within the guard-time in chapter 5. In chapter 6, we propose a hybrid data-gathering protocol that minimizes the expected critical event-reporting delay and maximize the lifetime. Finally, the thesis ends with conclusions and discussion in Chapter 8.

2

Background and Related Works

In this chapter we first give a brief overview of hardware and software systems, simulators, and node deployment strategies in WSNs. The remainder of this chapter describes works done so far to improve QoS requirements for different data-gathering techniques.

2.1 Wireless Sensor Networks Hardware and Software Systems

In this section first we describe a general architecture of a WSN followed by an architecture of a sensor node. A WSN consists of several tiny sensors powered by batteries which can communicate with each other to monitor a FoI.

The sensor network can consist of one or more sink node(s). The sink node (also

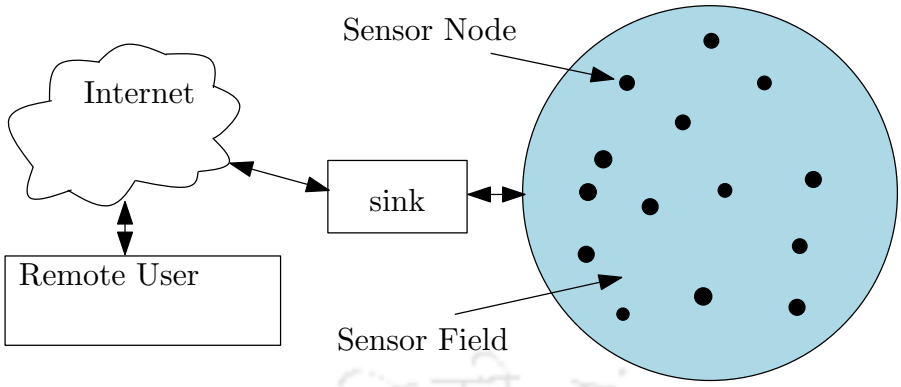


Figure 2.1: Basic architecture of a WSN

known as base-station) collects information regarding the environment from the sensors and forwards the same to a remote or local user. The remote user may be connected to the sink node by the Internet (see Fig.2.1).

2.1.1 Wireless Sensor Network: Hardware System

Generally, a sensor network consists of many sensor nodes. Here we describe the physical architecture of a sensor node. The physical architecture of a sensor node can be classified into four major units: processing, storage, communication, and sensing [96]. One can refer to Table 2.1 to see the major units of the architecture of some sensors. Moreover, the sensor node also has a power supply unit. The role of the power supply unit is to supply the energy needed for all units. Normally sensor nodes are battery powered and as the sensors are sometimes deployed in a hostile environment the battery is impossible to replace.

Processing Unit: The main job of the processing unit is to process the data received by the communication and sensing unit. The processors can be categorized into several types based on their tasks, like microprocessors and/or micro-controllers, low power digital signal processors, communication processors, and application

Table 2.1: Major units of the architecture of some sensors

Sensor	Major Units
Tmote	**Processing unit** : 8MHz Texas Instruments MSP430 microcontroller **Storage unit:** 10k RAM and 48k Flash **Communication unit:** 250kbps 2.4GHz IEEE 802.15.4 Chipcon Wireless Transceiver **Sensing unit:** Integrated Humidity, Temperature, and Light sensors
BTnode	**Processing unit:** Atmel ATmega128L(AVR RISC 8 MHz @ 8 MIPS) **Storage unit:** 180 Kbyte SRAM, 128 Kbyte Flash ROM, 4 Kbyte EEPROM **Communication unit:** Chipcon CC1000 operating in ISM Band 433-915 MHz **Sensing unit:** UART, SPI, I2C, GPIO, ADC, Clock, Timer, LEDs Standard Molex
CSIRO Fleck	**Processing unit:** Atmega128L, 8MHz **Storage unit:** 512K external memory **Communication unit:** Nordic 903 **Sensing unit:** Temperature, Light, Screw terminal for 4X digital i/o and 2X analog
MicaZ	**Processing unit:** ATMEGA 128 **Storage unit:** 4K RAM 128K Flash **Communication unit:** 802.15.4/ZigBee compliant RF transceiver **Sensing unit:** Large expansion connector
EYES	**Processing unit:** MSP 430F149 (5 MHz @ 16 Bit) **Storage unit:** 60 Kbytes of program memory and 4 Kbyte EEPROM **Communication unit:** RFM TR1001 hybrid radio transceiver **Sensing unit:** Compass, accelerometer, temperature sensor, light sensor, pressure sensor
Telos	**Processing unit:** Motorola HCS08 **Storage unit:** 4K RAM **Communication unit:** 250kbps 2.4GHz IEEE 802.15.4 **Sensing unit:** Large expansion connector

specific integrated circuits. One can refer to Table 2.1 for different processing unit for various sensors.

It can also be noted that generally the processors are equipped with a crystal oscillator which is used for the clock to implement the real time in sensor node. The rate at which the clock runs is determined by the angular velocity of the hardware oscillator. As a result of the imperfect oscillator or some external reasons, the oscillator frequency varies unpredictably, which is known as clock-skew [85].

The clock-skew also depends on environmental factors like temperature, pressure, radiation, magnetic fields etc. Generally, the clock-skew is measured in terms of parts-per-million (ppm), where one ppm is equivalent to drift of one micro sec-

ond (μs) in one second. The clock-skew rate of Mica Motes is up to 50 ppm. A typical range of clock-skew is generally up to 40 to 100 ppm [85]. Generally, the maximum clock-skew for a sensor is specified by the manufacturer. In a heterogeneous WSNs, where different sensor nodes are equipped with different processing units, maximum clock-skew for different sensors may vary over the given FoI.

Storage unit: The storage unit mostly consists of synchronous dynamic random access memory (SDRAM), electrically erasable programmable read-only memory (EEPROM), static random access memory (SRAM), and non-volatile (flash) memory. This unit is used to buffer received and sent messages. Depending on the message size and complexity the size of the storage unit varies. For example, in a multimedia sensor network [8], the storage unit is relatively high, compared to basic sensor networks. A comparison of different storage units for various types of sensors is given in Table 2.1.

Communication and sensing unit: The communication unit is used to communicate with other sensors and/or base-station. Generally, most of the sensors use 2.4 GHz band for communication. A comparison of different communication channels for different sensors are given in Table 2.1. Moreover, it can also be noted that the current generation of WSNs use mutihop communication which significantly reduces energy consumption.

The sensing unit interacts with the physical world around the sensor. Generally, the sensing unit is composed of a sensing subsystem and analog-to-digital (ADCs) converters. The sensing subsystem consists of different type of sensors, like temperature, pressure, light, humidity, etc., which produce analog signals. The ADCs convert the analog signals of the sensors into digital signals and forward to the

Table 2.2: Core interfaces of TinyOS

Interface	Description
ADC	Sensor hardware interface
Clock	Hardware clock
EEPROMRead/Write	EEPROM read and write
HardwareId	Hardware ID access
I2C	Interface to I2C bus
Leds	Red/yellow/green LEDs
MAC	Radio MAC laye
Mic	Microphone interface
Pot	Hardware potentiometer for transmit power
Random	Random number generator
ReceiveMsg	Receive Active Message
SendMsg	Send Active Message
StdControl	Init, start, and stop components
Time	Get current time
TinySec	Lightweight encryption/decryption
WatchDog	Watchdog timer control

processing unit [8]. Different sensing unit of different sensors are compared in Table 2.1.

2.1.2 Wireless Sensor Network: Software Systems

Operating system in the sensor network is much lighter than general purpose operating systems as because of sensor nodes have resource constraints in terms of processing power, memory size and energy. Here we discuss several operating systems developed so far to offer different solutions for sensor network applications.

TinyOS:

TinyOS [3] is a component based, event-driven programming model, developed in NesC language [35]. In practice, a TinyOS program is a graph of *components* which are independent entities and use TinyOS interfaces. The core TinyOS interfaces are given in Table. 2.2.

Each component is composed of three computational abstractions, which are *commands, events,* and *tasks.* Command initiates a service, for example initiating sensor reading, whereas an event is generated when the service is complete. Events can also be hardware interruption or any message arrival. Task is composed of several commands and events that may be executed at a certain time [53].

LiteOS:

It is an open source, interactive and a multi-threaded operating system designed specially for WSN devices. It supports Windows XP/Vista/7, MicaZ etc. LiteOS [2] supports many features like a hierarchical file-system with shell interface and UNIX-like commands, kernel support for multi-threading, debugging and dynamic memory etc.

The overall architecture of LiteOS can be divided into three parts, like LiteShell, LiteFS (file system), and the kernel. LiteShell is the UNIX like shell script runs on the base-station to interact with the user. LiteShell commands can be categorized into file commands, process commands, debugging commands, environment commands and device commands. File commands are similar to simple UNIX file commands (like *ls -l*). Process commands are used to run different applications on sensors. In order to help with debugging, the debugging commands are introduced in LiteOS. Examples of environment commands are *history* that displays the previous commands, *who* that shows the current user, *man* that generates command reference, and *echo* to display strings. The LiteOS shell interacts with the sensors using the device commands [22].

Contiki:

Contiki [1] is also an open source and event-driven operating system for networked and memory-constrained systems with a particular focus on low-power WSNs. It is developed in C language. One of the distinguishable feature of Contiki is that it can load and unload individual applications or services at run-time. Contiki also supports concurrency by using multi-threading [29]. A comparison between these operating systems is given in 2.3.

Table 2.3: Comparison of different operating systems

Feature	**TinyOS**	**LiteOS**	**Contiki**
Publication	2000	2008	2004
Language Support	nesC	LiteC++	C
Event Based Programming	Yes	Yes	Yes
Multi-Threading	Yes	Yes	Yes
Platform Support	Mica, Mica2, MicaZ, TelosB, Tmote, IRIS, Tinynode, Eyes	MicaZ, IRIS, AVR MCU	Tmote, TelosB, ESB, AVR MCU, MSP430 MCU
Simulator	TOSSIM, Power Tossim	Through AVRORA	Cooja, MSPSim, Netsim

2.2 Wireless Sensor Network Simulators

Any WSN simulator consists of several modules namely events, sensor, protocols and applications. Many different possible platforms are designed for simulating and testing of different WSNs.

2.2.1 NS2

NS-2 [5] is a discrete event simulator targeted at networking research. It first designed in 1989 and since then it is expanded to support almost every field in net-

working. Simulation for both wired and wireless network can be done using ns2. NS2 is written in C++ and OTcl (object oriented tool command language). C++ is used to design the internal mechanism and OTcl is used to set up the simulation by calling the objects and scheduling discrete events. NS2 provides a large number of C++ classes. Generally, simulation script contains Tcl commands which call these C++ classes. However, one can build own classes and refer them in Otcl script. At the end of the simulation, text-based simulation results are generated. One can interpret these results using NAM (network animator) or XGgraph [99].

NS-2 supports several wireless media that includes IEEE 802.15.4 which is widely used for WSN. Moreover, it also support battery models of sensor nodes. NS2 is extended to implement the energy consumption model of sensor network. After every operation, the remaining energy of each sensor can be viewed in output file. Sensor network based routing protocols like LEACH [38], TEEN [58], and PEGASIS [55], are added in NS2 for better performance evaluation [41].

2.2.2 TOSSIM

TOSSIM [7] is a discrete event simulator for TinyOS WSNs and developed in UC Berkeley. It is designed mainly for TinyOS application to be run on MICA Motes. It captures the behavior and interaction of a thousand of TinyOS motes. TOSSIM captures more realistic simulation results because it runs the same code in TinyOS that runs in actual sensor nodes. By replacing some components of TinyOS, TOSSIM translates hardware interrupts into discrete events [52].

2.2.3 GLoMoSim

Global Mobile Information System Simulator (GloMoSim) [6] is designed for large wireless and wired communication networks. GloMoSim is effective for hybrid networks that include wireless, wired and satellite based communication. In order reduce the execution time, GloMoSim uses PARSEC (PARallel Simulation Environment for Complex systems) simulation language, which is a C-based simulation language, developed by the Parallel Computing Laboratory at UCLA, for discrete event simulation. Moreover, GloMoSim introduces aggregation techniques to increase performance. GloMoSim is useful when a large number of nodes, from 100 to 100000, needs to be simulated in the FoI. As GloMoSim uses parallel programming, it achieves higher performance if the number of nodes increases.

2.3 Node Deployment Strategies in Wireless Sensor Network

Currently, sensor nodes are deployed on land, underground, and underwater. The deployment strategies refer to the technique followed while deploying nodes within the given FoI. Sensor nodes can be deployed deterministically or stochastically. In deterministic deployment strategy sensor nodes are placed manually. In applications like home or office surveillance, this type of strategy is preferred. Whereas, in stochastic deployment sensor nodes are scattered following a random process within the FoI. Random deployment is favorable in hostile scenarios where it is not possible to manually place the sensor. The example of random deployment strategies are uniform deployment and Poisson deployment where the sensor node are

deployed following uniform distribution and Poisson process, respectively [94].

2.4 Quality of Service Requirements for various Data-gathering Techniques

Many papers appeared in the literature managing QoS requirements in WSNs. Detailed survey of these approaches can be found in [23, 88, 95]. Depending on the application requirements, the QoS can be referred as the degree of reliability, timeliness, robustness, availability, security etc. provided by the corresponding WSN [88]. The QoS requirements can be classified into application specific, like coverage, lifetime, and deployment cost, and network specific, like connectivity, scalability, delay, packet loss, bandwidth, reliability, fault-tolerant, jitter and throughput etc [23]. In this subsection, we give a brief survey of various QoS requirements that are relevant in the context of our work for different data-gathering techniques.

2.4.1 Lifetime

Generally, sensor nodes are equipped with a limited, unattended battery power source. Hence, in any WSN application, extending the lifetime remains one of the key objectives along with satisfying other QoS requirements. Generally, energy consumption is reduced using s/w scheduling techniques where the communication device is switched off when no event is detected in the vicinity.

The s/w scheduling techniques can be further divided into synchronous and asynchronous s/w scheduling techniques. In synchronous s/w scheduling techniques,

sensor nodes associates slots for communication whereas in asynchronous s/w scheduling nodes wake-up independently. A sender-receiver pair wakes-up the same time to send/receive data-packets in synchronous s/w scheduling. In order to wake up at the same time, nodes exchange synchronization messages in synchronous s/w scheduling protocols.

In periodic data-gathering techniques, sensor nodes generate periodic data which are gathered at the control unit. The application set includes environmental monitoring, battlefield surveillance, and many more [94]. As senors are monitoring the environment continuously, tree-based forwarding strategies are often preferred. Compared to an arbitrary network topology where each sensor node needs to maintain routing table, the tree based strategy saves cost, in terms of energy consumption, of maintaining routing table. Moreover, using data-aggregation at the parent node decreases the amount of data-transmission and hence increases the lifetime in correlated data-gathering. Many efficient tree construction strategies are proposed in the literature [36, 31, 78, 100].

In [36], the authors studied the problem of constructing an efficient tree to send the aggregated information to the sink. The main objective is to construct an efficient tree that minimizes the total amount of data transmitted. Minimizing the total amount of data transmitted surely maximizes the lifetime. The author used a randomized technique that approximates the optimal tree. Enachescu et al. [31], considered a grid of sensors and propose a simple randomized tree construction scheme that achieves a constant factor approximation to the optimal tree. In [78], the authors presented a continuous data-gathering protocol that efficiently collects data while maintaining constant local state and making only local decisions. Khan and Pandurangan in [44] proposed a distributed algorithm that constructs an ap-

proximate minimum spanning tree (MST) in arbitrary networks.

In [87], Shroff et al. showed that the problem of finding the optimal data-gathering tree that maximizes lifetime for a given data-generation rate is NP-Complete. The authors proposed a tree construction method that starts from an arbitrary tree and iteratively reduces the load on bottleneck nodes (nodes likely to soon deplete their energy due to high degree or low remaining energy). This method completes in polynomial time and provides a near optimal solution.

Generally, in periodic data-gathering nodes exchange data packets at periodic intervals. Hence synchronous s/w scheduling techniques are energy efficient where every sender-receiver pair associates slots for communication [85]. Exact clock synchronization is required for synchronous s/w scheduling[1], which is very difficult to achieve because of the imperfect crystal oscillator [85]. The most governing factors for clock-disagreement are phase-offset[2] and clock-skew[3]. In order to circumvent the effect of clock-skew, time synchronization protocols are applied. Sensor nodes synchronize time by calculating expected phase offset and expected clock-skew. At the end of time synchronization, clocks may diverge as a result of estimation error involved in synchronization which follows normal distribution with zero mean [85].

Synchronization

In traditional synchronization schemes the sender periodically sends messages containing its current clock value to synchronize the time with the receiver. An extension of this approach is a two-way message passing to estimate the propagation

[1]This is because a sender-receiver pair needs to wake-up at the same time

[2]Phase-offset denotes the clock difference (or time difference) between two sensors at an instant of time.

[3]The clock-skew refers to the difference in actual and expected crystal oscillator frequency.

delay in order to increase the accuracy of clock-synchronization [61]. Generally, synchronization error occurs as a result of nondeterminism present in send time, channel access time, propagation time, and receive time. In order to remove this nondeterminism, Elson et al. proposed Receiver-Broadcast Synchronization (RBS) scheme [30], by broadcasting physical layer beacons. The neighbors compare the arrival time of these beacons and calculate phase-offset and clock-skew using least-square linear regression model. As a result of several rounds of message exchange RBS consumes a significant amount of energy. In Timing-sync Protocol for Sensor Networks (TPSN) [33], nodes exchange only two way time-stamped synchronization messages at medium access control (MAC) layer and calculate the phase offset. It successfully eliminates synchronization error caused by access time and propagation delay, but unable to estimate the clock-skew. In flooding time synchronization protocol [60], Maroti et al. combined TPSN and RBS to estimate the clock-skew using linear regression. In [34], Ganeriwal et al. proposed an energy efficient long term synchronization scheme by transmitting synchronization beacons at periodic intervals to maintain desired bound on synchronization error. The authors used the least square based scheme to calculate the clock-skew from a set of received beacons. It also adapts to change in clock drift and environmental conditions to achieve application-specific precision with high probability [34]. Similar approach is followed in [80, 73, 30, 34] to calculate expected phase shift and clock-skew among the neighbors. We only discuss the significant contributions in the literature, the detailed survey of these protocols can be found in [77, 48].

Although there are many synchronization protocols are available, none of them can completely remove the clock-disagreement. This is because of inaccuracy and non-determinism present in time synchronization protocol and some external fac-

tors that affects clock-skew. In order to mitigate this effect and circumvent any message loss nodes wake up earlier than the scheduled wake up time which is known as guard time. In [92], the authors used a fixed guard time to compensate any message loss. In the absence of time synchronization, the guard time tends to increase as time passes which in turn increases energy consumption. In order to conserve more energy, Wu et al. dynamically vary guard-time as time passes while satisfying given threshold message capture probability in the presence of normally distributed synchronization error [85].

If the time elapses from last-synchronization increases, the amount of energy consumption during guard-time increases rapidly. The use of periodic switch-on/off within the guard-time during data-collection is first proposed by Beber et al. [19] recently, to conserve the energy consumption during guard-time. In BailighPulse [19], a sensor node wakes up multiple times within the guard-time, such that every time it polls the channel for any activity. If any activity is present, data-transmission begins, else the node switches off its transceiver to save energy and wakes up at the next beacon interval.

In applications like intrusion detection, tsunami detection, forest-fire detection, volcanic event detection, and many more, the sensor nodes remain idle unless and until a critical event occurs in the vicinity, and once detected, the event information is forwarded to the base-station as early as possible [84]. This type of data-gathering is known as event-driven data-gathering. In event driven data-gathering, synchronous s/w scheduling is not energy efficient. Passing synchronization messages increases energy consumption, which may be a major dominating

Table 2.4: Comparison between different s/w scheduling techniques

Paper	Synchronous	Network Type	Objective	Solution
[36]	Yes	Tree based	Maximize lifetime	Randomized technique that approximates optimal tree
[31]	Yes	Tree based	Maximize lifetime	Randomized tree construction, achieves a constant factor approximation to the optimal tree
[78]	Yes	Tree based	Maximize lifetime	Distributed protocol makes local decisions
[44]	Yes	Minimum Spanning Tree	Minimize data transmission	Distributed protocol: approximate minimum spanning tree
[87]	Yes	Tree based	Maximizes lifetime	Polynomial time, near optimal solution
[92]	Yes	Tree/graph based	Minimizes energy Consumption in presence of clock-skew	Multiple time wakes-up within guard-time
[66]	No	Random forwarding	Minimizes critical event delay	Asynchronous long preamble-sampling based protocol
[20]	No	Random forwarding	Minimizes delay and maximizes lifetime	Asynchronous short and strobed preamble-sampling
[39]	No	Anycasting forwarding	Minimizes delay and maximizes lifetime	shortest path anycasting tree
[102, 56]	No	Anycasting forwarding	Minimizes delay and maximizes lifetime	Exploits geographical distance
[18, 68, 69]	No	Anycasting forwarding	Minimizes delay and maximizes lifetime	Minimizes hop-count information
[46]	No	Anycasting forwarding	Minimizes expected delay and maximizes lifetime	Optimal anycasting distance forwarding technique
[45]	No	Anycasting forwarding	Minimizes delay and maximizes lifetime	Delay optimal anycasting scheme

factor in rare event detection application. Whereas in asynchronous s/w scheduling techniques, since sensor nodes wake-up independently, no clock-synchronization is required with the neighboring nodes, which in turn decreases energy consumption [45]. Hence, in event-driven data-gathering asynchronous s/w scheduling techniques are considered to be more energy efficient.

An asynchronous preamble-sampling based protocol which exploits Low-Power-Listening (LPL), is proposed in [66]. Nodes periodically listen for a short time to decide ongoing transmission. If an event is detected, the sender attaches long preamble before transmitting data. LPL minimizes energy consumption when no event is detected, whereas consumes more energy due to long preambles. In order to minimize energy consumption for long preamble, short and strobed preambles based strategies are proposed [20], using additional low power radio, which increases sensor cost. In order to reduce the cost for additional low power radio, a similar strategy is proposed without the need of an ultra low power radio [72]. The trade-off between energy efficiency and latency associated with waking up the nodes, is also discussed in [72].

In hybrid data gathering schemes, the sensor nodes switch between event-driven and periodic according to the requirement or application specification. Consider a forest monitoring system where the sensor nodes are deployed to monitor several physical phenomena like temperature, humidity, and many more using a periodic approach. However, if any critical event like fire breaks out within the forest this event information is forwarded to the base-station as early as possible in order to minimize the damage. Similar application is environmental monitoring with earthquake or volcanic event detection, and many more [11, 9, 76].

Several works appeared in hybrid data-gathering to improve QoS. In [59], Arati Manjeshwar and Dharma P. Agarwal proposed a hybrid protocol APTEEN that can handle both periodic and event-driven queries. APTEEN uses soft and hard threshold to switch between event-driven and time-driven approach. Generally, the sensor nodes send data-packets if an event is detected. If no event is detected for sometime, the sensor nodes switch to time-driven approach and starts sending data-packets periodically. In order to prolong the lifetime further, in SINA [75] instead of all nodes joining the information-gathering process, only a set of nodes participate in the process and hence nodes are included selectively. In [51], a hybrid energy-efficient protocol was proposed that supports dynamic switching between event and time driven data-reporting scheme. The sensor nodes that are going to detect an event in the near future wake up pro-actively along with the nodes that are detecting the event.

2.4.2 Delay

QoS guarantees may be required to ensure that the occurrence of a high priority data is to be delivered to the destination with certain bounds on delay. This is referred as delay constraint. In this subsection, we briefly survey the delay-aware network protocols proposed in recent times. A detailed study of the literature can be found in [10].

Shanti and Sahoo [70] have proposed a periodic data gathering protocol, called Delay Guaranteed Routing and MAC (DGRAM) protocol, that provides deterministic delay guarantee. Their protocol requires a short beacon exchange phase to gather node location information. Then, it uses slot reuse technique to reduce

latency between two successive medium accesses by a sensor node, with a slot allocation strategy that does not require exchange of control messages. This technique makes the deployment self-configuring, then each node runs a short beacon exchange phase to learn about the topology of the network. It also allows sensors to go to sleep when they are not communicating to save energy. The authors have presented the method to assign time slots to sensor nodes and show how the slots are reused by nodes that are non-interfering. They have presented delay analysis of DGRAM to prove its delay bound. Their simulation results have shown that their analytical delay bound is always guaranteed by the protocol. Cuomo et al. [25] proposed a clustered-tree topology that improves energy consumption using aggregation and reduce the delay. In order to reduce the the delay the authors minimized the number of hops between the source and the sink. They have shown that their algorithm minimizes delay using simulation.

Bhuiyan et al. [17] minimized the delay by choosing nodes that are having light loads and incurring low delay during data forwarding towards the Base Station (BS). They considered that all nodes describe their congestion status and delay measurements by broadcasting periodic control data packets to allow neighboring nodes to utilize these data during route selection process. Hence the performance of their proposed scheme depends on the successful delivery of these control packets.

Lin and Van der Schaar [54] have proposed a multi-hop network to optimize the overall performance of several delay-sensitive applications under different network dynamics. Furthermore, they have investigated how nodes can independently learn the network dynamics online, based on their available information. This online learning strategy enables the users to adapt their cross-layer techniques in real time fashion to the changing environment so that the nodes can maximize the usefulness

of the delay-sensitive applications.

He et al. [37] have proposed the SPEED algorithm which is a stateless, localized routing algorithm for WSNs that provides real-time communication. The authors have assumed that nodes are aware of their location, routing can be done by selecting at each step, nodes which are closer to the destination than the actual one. Because SPEED only maintains immediate neighbor information, they have used a beaconing module to exchange neighbor locations.

Nandi and Yadav [64] have proposed a QoS aware MAC protocol for WSNs and its cross-layer extension to network layer to provide QoS to delay sensitive WSN scenarios. They have considered both event driven traffic which needs immediate attention and periodic reporting traffic. They have reduced the delay suffered by using three techniques; difference in sleep schedules (DSS) of nodes through dynamically regulating duty cycle based on utilization.

As mentioned earlier, in order to save energy nodes use s/w scheduling techniques. In event driven data-gathering, synchronous s/w scheduling is not energy efficient. Passing synchronization messages increases energy consumption, which may be a major dominating factor in rare event detection application.

In event-driven data-gathering, along with lifetime another most important QoS requirement is e2e delay. In asynchronous s/w scheduling, each node wakes up independently and waits for the next hop node to wake up before transmitting any packet which inturn increases one-hop delay. If one-hop delay increases, e2e delay increases as well.

In order to minimize the delay incurred by asynchronous strategies, anycasting based packet forwarding schemes are proposed in the literature. In anycasting

based packet forwarding schemes, each node maintains a set of candidate nodes to forward a data packet. Actually, data packet is forwarded to the first node wakes up within the forwarding set. Since each node maintains a set of forwarding nodes, compared to a single forwarding node in traditional approaches, anycasting strategy decreases expected waiting time significantly [46].

Several anycasting based packet forwarding schemes are discussed in the literature for wireless networks. In order to route a data-packet efficiently, shortest path anycasting tree is proposed in [39]. Compared to general shortest path tree, in the shortest path anycasting tree, instead of forwarding the packet to a single parent, every node maintains a set of multiple parents, and forwards the data-packet to the first node wakes up within this set. Geographical distance to the sink is exploited in [102, 56], to minimize the e-delay. Hop-count information is also used to minimize the delay in the routing path [18, 68, 69]. Moreover, in [63] the authors used both hop-count and power consumption metrics to reduce the overall cost of forwarding a data-packet from the source to the sink, which in turn increases overall lifetime of the network.

Kim et al. proposed an anycasting forwarding technique, in which neighboring nodes are added to the forwarding set only when they collectively minimize overall expected e-delay [46]. The authors developed an optimal anycasting forwarding technique to minimize the expected event-reporting delay for given wake-up rate in [46]. Next, the same authors studied the joint optimization problem to find the wake-up rate and forwarding policy that minimizes the delay and maximizes the lifetime. But when an actual critical event occurs, the packets may follow a longer route to the base station. In order to minimize path length, the same authors developed a delay optimal anycasting scheme [45], where nodes do not immediately

forward the packet, instead they wait for some time and then opportunistically forward only when expected delay involves for waiting is more. A comparison of different s/w scheduling techniques is given in Table. 2.4.

2.4.3 Overall Cost of the Network

Another important QoS requirement is minimizing overall cost of the network. Cost is defined as the factor of the expected number of sensors present in the FoI times cost of each sensor. The expected number of nodes present in the FoI is the factor of the density times the area of the FoI. It can also be noted here that density refers to the number of sensors present in the unit area. For a uniform random deployment with density λ, the overall cost of the network is defined as $\lambda \times ||A|| \times C$, where $||A||$ and C respectively denotes the area of FoI and cost of each sensor. In order to minimize the overall cost of the network, critical sensor density(CSD) is estimated. CSD refers to the minimum number of sensors required to satisfy given constraints.

Coverage in WSNs determines how well the FoI is covered. As sensor nodes are equipped with a limited, unattended battery power source, the maximum lifetime coverage problem emerged in the literature. For a given WSN, the problem of finding the set of sensors that provides coverage requirement and the time duration for the set of sensors to be active, such that the lifetime is maximum, is known as the maximum lifetime coverage problem.

Existing works focus on organizing the sensors into a number of scheduling subset such that each subset provides the desired lever of coverage. The work in [12, 89, 82] minimize the number of sensors in a subset by identifying redundant sensors. In order to guarantee both coverage and connectivity, Coverage Connectivity Protocol

(CCP) is proposed in [91] that shows if the communication range is at least twice the sensing range of a sensor, then connectivity is taken care of by the coverage. One can refer to [81] for a detailed survey of these techniques. The problem of finding CSD to satisfy a given coverage requirement is well addressed in literature. Zhang and Hou estimated CSD to cover a given FoI with high probability [96]. They extended it to show that grid based deployment requires lower density compared to uniform and Poisson distribution [97].

The problem of finding CSD for random deployment of nodes in a heterogeneous WSN is addressed in [50]. Yen et al. derived the expected coverage ratio for a randomly deployed WSN [93]. They mapped the stochastic coverage problem into set intersection problem and use integral geometry to find the expected coverage ratio. The theoretical results are verified using simulation results. In [90], the authors studied the problem of finding CSD to satisfy both coverage and connectivity requirement. The authors also showed a relation between coverage and connectivity. Specifically, they showed that if the communication range is at least twice of the sensing range, the FoI is inherently k-connected if it is k-covered.

These studies mostly focus on either full connectivity or k-connectivity in the network. The problem of finding CSD for partial connectivity is addressed by Cai et al. in [21]. Assuming that the sensor deployment follows Poisson distribution, the authors found the CSD of the sensor nodes for given partial connectivity requirement ρ, such that the probability of at least a fraction ρ sensors are connected in the network is high. The problem of finding the CSD for a fault-tolerant network is addressed in [28]. In a 3-D WSN, the problem of finding CSD that satisfies given coverage and connectivity requirements is addressed in [13]. In [65], the authors given an upper bound on the lifetime for given density and degree of coverage.

3

Minimum Cost Event Driven WSN with Spatial Differentiated QoS Requirements

3.1 Introduction

In a rare event driven data gathering application, like intrusion detection, tsunami detection and forest-fire detection, nodes remain idle for most of the time until an event occurs. Some of the key QoS requirements in these type of applications are the lifetime, e2e delay, and the overall deployment cost of the network. Lifetime refers to the duration of time before the first node depletes its energy completely it the network. E2e delay is the average delay between an event is detected and the event information reaches to the base-station. If we assume that N nodes are

deployed in the network such that each node is associated with cost C_{Node}, then the overall cost of the network is denoted as $C_{Net} = C_{Node} \times N$. Let us assume that nodes are deployed uniform randomly with rate λ. If λ increases, the average cost of the network C_{Net} may increase, and vice versa. In these type of networks, some of the key objectives are maximizing the lifetime, minimizing the e2e delay and minimizing the overall cost of the network, and satisfying the given coverage and connectivity requirement etc. Cost is defined as the factor of the expected number of sensors present in the FoI times cost of each sensor. The expected number of nodes present in the FoI is the factor of the density times the area of the FoI. It can also be noted here that density refers to the number of sensors present in the unit area. For a uniform random deployment with density λ, the overall cost of the network is defined as $\lambda \times ||A|| \times C$, where $||A||$ and C respectively denotes the area of FoI and cost of each sensor.

In applications like fire or disastrous event detection, an event detected close to the central facility needs to reach faster than an event detected far away. In other words, there can be a large variation in QoS requirements for different areas within the FoI. High priority areas are associated with stringent QoS requirements, whereas low priority areas are with relaxed QoS requirements. Assume the base-station along with the central facility is located in the middle of a convex-shaped FoI (refer Fig. 3.1). The area around the central facility is divided into circular rings with exponential e2e delay constraints like $1\,ms, 10\,ms, 100\,ms, 1\,s, 10\,s, 100\,s$, etc. If an event is detected in an area with darker shade, the event information must reach to the base-station much faster than an event is detected in areas with lighter shades. Designing a minimum cost network for such an application while satisfying the delay constraint for each area and the lifetime requirement of the

overall network is an interesting problem. Hence, in this chapter we are interested in the problem of *finding a minimum cost WSN for a given spatial differentiated e2e delay constraints and lifetime requirement in a convex-shaped FoI.*

Generally, sensor nodes are deployed in a rectangular, squared shaped or in a circular area. In order to provide a general solution that is applicable in all of these cases we considered the FoI to be a convex-shaped. An example for such a convex-shaped FoI is a when nodes are deployed in a $l \times l\ m^2$ area. We use a stochastic method to estimate the expected e2e delay for a given density and use this analysis to find the expected critical sensor density (CSD) that satisfies both the delay constraint and the lifetime requirements. Next we use the estimated CSD to find the minimum cost WSN.

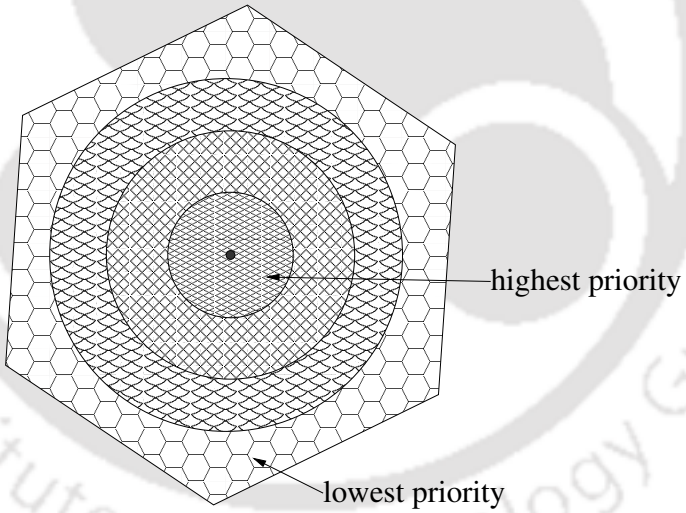


Figure 3.1: Different priority areas

In event-driven data-gathering, anycasting forwarding techniques are used to minimize the delay and maximize the lifetime. In anycasting forwarding strategy, density governs e2e delay and lifetime. With increasing the density, e2e delay may decrease for a given wake up rate. In the contrary, for a given e2e delay, increasing the density may result in increasing overall lifetime of the network. In order to

increase network lifetime, for a given e2e delay constraint, one can decrease the wake up rate by increasing the density. But, increasing the density increases the overall cost of the network. Hence, we address the problem: *what is the CSD that satisfies given delay constraint and lifetime requirement, when nodes follow anycasting forwarding strategy?* If the FoI is divided into k areas such that each area, A_i, has a delay constraint D_i, and lifetime constraint L, then the problem can be defined formally as,

$$\min_{\lambda_1,\lambda_2,\ldots,\lambda_k} \{\sum_{i=1}^{k} ||A_i|| \times \lambda_i \times C\} \quad \text{subject to}$$

$$D'_i \leq D_i \quad \forall i,$$

$$||A_i|| > 0,\ \lambda_i \geq \lambda^c > 0\,, D_i > 0 \,\forall i,$$

$$L' > L > 0,\ w > 0,\ C > 0, \tag{3.1}$$

where D'_i denotes the expected e2e delay in an area A_i, L' denotes the average lifetime, w denotes the wake-up rate and λ^c denotes the density requirement to satisfy coverage and connectivity. In order to minimize $\sum_{i=1}^{m} ||A_i|| \times \lambda_i \times C$, we minimize $||A_i|| \times \lambda_i \times C$ for all i. Hence, we re-write Eq. 3.1 as,

$$\min_{\lambda_i} \{||A_i|| \times \lambda_i \times C\} \quad \text{subject to}$$

$$D_i' \leq D_i,$$

$$||A_i|| > 0,\ \lambda_i \geq \lambda^c > 0\,, D_i > 0,$$

$$L' > L > 0,\ w > 0,\ C > 0, \tag{3.2}$$

for all i. We use a stochastic approach to estimate expected e2e delay for a given sensor density and use this analysis to find the CSD that satisfies given requirements in an area.

The problem of finding the CSD is well addressed in the literature, but in the context of satisfying coverage and connectivity requirements [96, 97, 50, 93, 90, 98, 14, 21, 28]. The authors in [65] derived an upper bound on the lifetime of the network, but for given spatial differentiated delay constraints and lifetime requirement, to the best of our knowledge, the problem of finding CSD is not addressed in the literature. Proposed anycasting forwarding techniques [102, 56, 18, 68, 69, 46, 45] in the literature are not applicable to find CSD for given delay and lifetime requirements.

3.2 Minimum Cost Network for Spatial Differentiated QoS

We use a stochastic approach to find the minimum cost network for a given delay constant of each area and the lifetime requirement of the whole network. First we

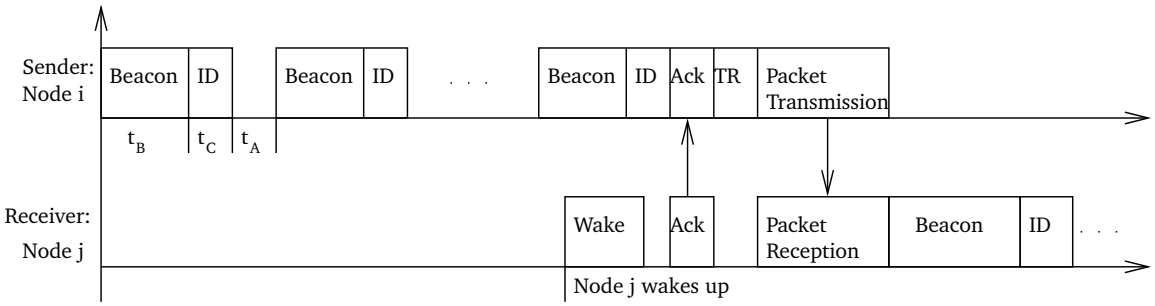


Figure 3.2: Packet forwarding protocol

estimate the CSD for given delay constraints and lifetime requirement, and use this analysis to find the minimum cost network. In order to find the CSD for given delay constraint and lifetime requirement, we first analyze the expected e2e delay for a given sensor density.

3.2.1 Expected e2e Delay

First we use a stochastic approach to estimate the expected e2e delay of a randomly chosen sensor i, located at a distance dis_i from the base-station in a circular-shaped FoI. Then we extend this analysis to estimate the maximum expected e2e delay of a convex-shaped FoI.

Expected e2e Delay for a Fixed Size Forwarding Set

We assume random uniform deployment of nodes in a circular-shaped FoI. We also assume that a sensor located at a point c, can only communicate perfectly within a circular region of radius C centered at c, which is denoted by $A(c, C)$. In our protocol before sending a data packet, a node sends a beacon signal of duration t_B, followed by an ID signal of duration t_C, and listens for acknowledgment of duration t_A (refer Fig. 3.2). If any node hears the beacon and the ID signal, it

sends acknowledgment only if it belongs to the forwarding set, else go to sleep and wakes up after $\frac{1}{w}$, where w denotes the asynchronous periodic wake up rate. Let $\{i_1, i_2, ..., i_k\}$ be the forwarding set of node i. The probability of any node in the forwarding set wakes up at h^{th} beacon signal is defined as $p_w = \frac{t_I}{1/w}$, if $h < \frac{1/w}{t_I}$, else 1, where $t_I = t_A + t_B + t_C$ denotes the beacon interval. Assume node $i_k \in F_i$ wakes up at h^{th} beacon signal. Note that, $1 \leq h \leq \frac{1/w}{t_I}$. We denote $h_{max} = \frac{1/w}{t_I}$.

Let W_h denotes the event that the packet is forwarded at h^{th} beacon. Note that the packet is forwarded at h^{th} beacon only if no node wakes up during $(h-1)$ beacons and atleast one node wakes up during h^{th} beacon. The probability of no node wakes up during $(h-1)$ beacons is $(1-p_w)^{k(h-1)}$ and the probability that atleast one node wakes up during h^{th} beacon is $(1-(1-p_w)^k)$.

Hence, the probability of the packet is forwarded after h^{th} beacon is,

$$P(W_h) = ((1-p_w)^{k(h-1)})(1-(1-p_w)^k). \tag{3.3}$$

Hence, expected one hop delay is given by

$$d_{k,w} = \sum_{h=1}^{\left\lfloor \frac{1/w}{t_I} \right\rfloor} P(W_h) * h + t_D, \tag{3.4}$$

where t_D denotes the transmission delay. The expected e2e delay of a node i is the sum of the expected one hop delay and the expected e2e delay of the nodes in its forwarding set. Since, every node in forwarding set has equal asynchronous

periodic wake up rate w, the probability of the packet is forwarded to any node is $\frac{1}{k}$, and the expected e2e delay of its forwarding set nodes is $\sum_{j=1}^{k} \frac{1}{k} * D_{i_j,k,w}$, for $1 \leq j \leq k$, where $D_{i_j,k,w}$ denotes the respective expected e2e delay of node i_j. Hence follows the lemma.

Lemma 1. *Let $\{i_1, i_2, ..., i_k\}$ be the forwarding set of node i. If $D_{i_j,k,w}$, denotes the respective expected e2e delay of node i_j, for $1 \leq j \leq k$, then the expected e2e delay of node i, such that $(i \neq j)$, is given by $D_{i,k,w} = d_{k,w} + \sum_{j=1}^{k} \frac{1}{k} * D_{i_j,k,w}$, where $d_{k,w} = \sum_{h=1}^{\lfloor \frac{1/w}{t_I} \rfloor} P(W_{h,k}) * h + t_D$.*

Expected e2e Delay of a Circular-shaped FoI

In this subsection, we use the numerical iteration technique to estimate the maximum of the expected e2e delay in a circular-shaped FoI with the base-station positioned at its center.

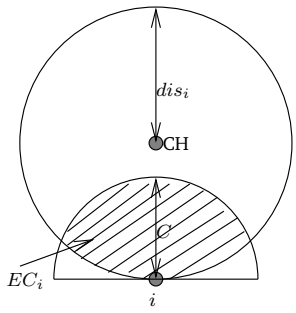
The overall e2e delay decreases if the neighboring nodes with the lower e2e delay are included in the forwarding set [46]. Increasing the number of nodes in the forwarding set decreases expected one-hop delay but may increase the expected e2e delay of its forwarding set nodes. Hence, in order to reduce the expected e2e delay of node i, only neighboring nodes which collectively minimizes the overall expected e2e delay are included in its forwarding set. The e2e delay of a node increases as the distance from the base-station increases for a given density. Hence, a linear search within the neighboring nodes, with higher priority to the nodes closer to the base-station efficiently selects the forwarding set, which minimizes overall expected e2e delay. The expected e2e delay of a given FoI is the expected e2e delay of the farthest node from the base-station.

The e2e delay of the nodes are estimated in the increasing order of their distance from the base-station. We estimate the e2e delay of the nodes closer to the base-station first, and use the results to estimate the e2e delay of their neighbors which are away from the base-station. Since the base-station is always awake, the e2e delay of the nodes within its communication range is equal to transmission delay t_D. We denote this *direct communication circle* by C_D.

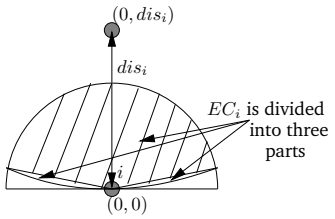
In order to estimate the maximum expected e2e delay, we divide the FoI into rings using concentric circles centered at base-station, and estimate the expected e2e delay for a randomly chosen node in each ring. First, we estimate the expected e2e delay of a randomly chosen node within the first ring using the e2e delay of nodes within the direct communication range of the base-station. Assuming base-station is positioned at b_p, we estimate the minimum expected e2e delay of a randomly chosen node i within the circular annulus $CA(b_p, C, C+\delta_1)$, such that the neighboring nodes closer to base-station belongs to C_D. The circular annulus $CA(b_p, R_i, R_j)$ between two concentric circles, centered at b_p with radii R_i and R_j, such that $R_j > R_i$, is defined as the area between their boundaries.

For a node i at a distance dis_i from the base station, the effective forwarding region of communication, EC_i, is the intersection of the open circular area with radius dis_i centered at the base-station and communication region of node i, as shown in Fig. 3.3(a). The following lemma quantifies the effective forwarding region of communication, which can be proved using simple geometry.

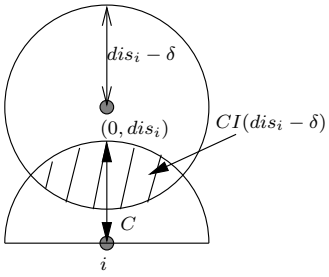
Lemma 2. *Consider a node i with communication range C and at a distance of dis_i, such that $dis_i > C$, from the base-station. The area of the effective forwarding region of communication of the node i is* $||EC_i|| = 2\cos^{-1}\left(\frac{C}{2dis_i}\right)C^2 + 2\cos^{-1}\left(1-\frac{C^2}{2{dis_i}^2}\right){dis_i}^2 -$



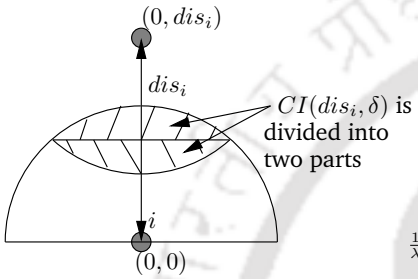
(a) Effective forwarding region of communication



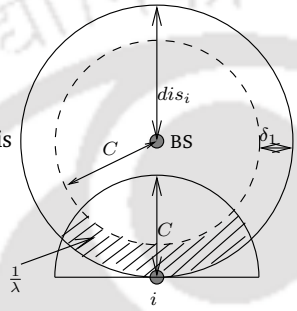
(b) Demonstrates how EC_i is estimated



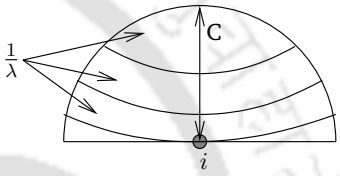
(c) Intersection of a circle and EC_i



(d) Demonstrates how $CI(dis_i, \delta)$ is estimated



(e) Only one node is expected in the shaded region



(f) EC_i is divided into equal area sectoral annuli.

Figure 3.3: Estimation of e2e delay in a circular-shaped FoI

$dis_i\sqrt{C^2 - \left(\frac{C^2}{2dis_i}\right)^2}$.

Proof. The effective forwarding region of communication, EC_i, is the intersection of the open circle with radius dis_i centered at base-station and closed circle with radius C centered at node i. The intersection points between the circle centered at $(0, dis_i)$ with radius dis_i and the circle centered at $(0, 0)$ with radius C are $\left(\pm\sqrt{C^2 - \left(\frac{C^2}{2dis_i}\right)^2}, \frac{C^2}{2dis_i}\right)$. The angle associated between these points for the circle centered at $(0, 0)$ with radius C is $2\cos^{-1}\left(\frac{C}{2dis_i}\right)$. Hence, the area associated for the circle centered at $(0, 0)$ with radius C is, $C^2 * 2\cos^{-1}\left(\frac{C}{2dis_i}\right)$. Note that EC_i is the sum of this area and the area associated with two symmetrical sectoral cut by the line joining $(0, 0)$ and

$\left(\sqrt{C^2-\left(\frac{C^2}{2dis_i}\right)^2}, \frac{C^2}{2dis_i}\right)$, $\left(-\sqrt{C^2-\left(\frac{C^2}{2dis_i}\right)^2}, \frac{C^2}{2dis_i}\right)$, respectively (see Fig. 3.3(b)). In order to calculate the area associated by the sectoral cut, we first calculate the area associated by the angle $\cos^{-1}\left(1-\frac{C^2}{2dis_i}\right)$ for the circle origin at $(0, dis_i)$ with radius dis_i, and subtract the area associated with the symmetric triangles $\left((0,dis_i),(0,0),\left(\pm\sqrt{C^2-\left(\frac{C^2}{2dis_i}\right)^2}, \frac{C^2}{2dis_i}\right)\right)$. We get the sum of two symmetrical sectoral cut as $2\cos^{-1}\left(1-\frac{C^2}{2{dis_i}^2}\right){dis_i}^2 - dis_i\sqrt{C^2-\left(\frac{C^2}{2dis_i}\right)^2}$. Hence, the effective forwarding region of communication is

$$EC_i = 2\cos^{-1}\left(\frac{C}{2dis_i}\right)C^2 + 2\cos^{-1}\left(1-\frac{C^2}{2{dis_i}^2}\right){dis_i}^2 - dis_i\sqrt{C^2-\left(\frac{C^2}{2dis_i}\right)^2}. \quad \square$$

The following lemma estimates the intersection of a circle and the effective forwarding region of communication for node i, which can be proved using simple geometry.

Lemma 3. *Consider a node i at a distance dis_i from the base station, with communication range C. The intersection of a circle centered at base-station (b_p) with radius $R_j = dis_i - \delta, 0 < \delta \leq C$, and the effective forwarding region of communication of node i(refer Fig. 3.3(c)), $CI(dis_i, \delta)$, is*

$$\begin{aligned} CI(dis_i,\delta) = {} & 2\cos^{-1}\left(\frac{\delta^2-c^2-2dis_i\delta}{2dis_iC}\right)C^2 \\ & - \frac{\delta^2-c^2-2dis_i\delta}{2dis_i}\sqrt{C^2-\left(\frac{\delta^2-c^2-2dis_i\delta}{2dis_i}\right)^2} \\ & + 2\cos^{-1}\left(\frac{dis_i-\frac{\delta^2-c^2-2dis_i\delta}{2dis_i}}{dis_i-\delta}\right)C^2 \\ & - \left(dis_i-\frac{\delta^2-c^2-2dis_i\delta}{2dis_i}\right)\sqrt{C^2-\left(\frac{\delta^2-c^2-2dis_i\delta}{2dis_i}\right)^2} \end{aligned} \tag{3.5}$$

Proof. Note that the intersection of a circle centered at base-station (b_p) with radius $R_j = dis_i - \delta, 0 < \delta \leq C$, and the effective forwarding region of communication of node i, EC_i, is the intersection of the circle centered at base-station with radius $(dis_i - \delta)$ and the circle centered at $(0,0)$ with radius C (refer Fig. 3.3(c)). The intersection points of the circle centered at base-station and the circle centered at node $(0,0)$ are $\left(-\sqrt{C^2 - \left(\frac{\delta^2 - 2dis_i\delta - C^2}{2dis_i}\right)^2}, \frac{\delta^2 - 2dis_i\delta - C^2}{2dis_i}\right)$ and $\left(\sqrt{C^2 - \left(\frac{\delta^2 - 2dis_i\delta - C^2}{2dis_i}\right)^2}, \frac{\delta^2 - 2dis_i\delta - C^2}{2dis_i}\right)$. The intersection of the circle centered at base-station with radius $(dis_i - \delta)$ and the circle centered at node $(0,0)$ with radius C is the union of the sectoral area of the circle centered at the base-station with radius $(dis_i - \delta)$ cut by the line joining the points $\left(-\sqrt{C^2 - \left(\frac{\delta^2 - 2dis_i\delta - C^2}{2dis_i}\right)^2}, \frac{\delta^2 - 2dis_i\delta - C^2}{2dis_i}\right)$ and $\left(\sqrt{C^2 - \left(\frac{\delta^2 - 2dis_i\delta - C^2}{2dis_i}\right)^2}, \frac{\delta^2 - 2dis_i\delta - C^2}{2dis_i}\right)$ and the sectoral area of the circle centered at node $(0,0)$ with radius C cut by the same line (see Fig. 3.3(d)). The sectoral area of the circle centered at node $(0,0)$ with radius C is $2\cos^{-1}\left(\frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_iC}\right)C^2 - \frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i}\sqrt{C^2 - \left(\frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i}\right)^2}$ and the sectoral area of the circle centered at the base-station with radius $(dis_i - \delta)$ is $2\cos^{-1}\left(\frac{dis_i - \frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i}}{dis_i - \delta}\right)C^2 - \left(dis_i - \frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i}\right)\sqrt{C^2 - \left(\frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i}\right)^2}$. Hence, intersection of the circle centered at base-station with radius $(dis_i - \delta)$ and the circle centered at node $(0,0)$ with radius C is given in Eq. 3.5. □

Let δ_1 denotes the maximum width of the circular annulus $CA(b_p, C, C + \delta_1)$, such that the effective forwarding region of communication EC_i, is expected to contain only neighbors, which are in the direct communication range of the base-station. Therefore, the expected number of nodes in the shaded region in Fig 3.3(e) of a node i is one, which is node i itself. Moreover, the expected area of the shaded

region in Fig 3.3(e) is $\frac{1}{\lambda}$, where λ denotes the node density. The expected maximum value of δ_1 can be found by solving the following equation,

$$EC_i - CI(dis_i, \delta_1) = \frac{1}{\lambda}. \tag{3.6}$$

We assume that Eq. 3.6 can be solved in constant time because it is a single variable equation. In the following lemma we estimate the minimum expected e2e delay of a node belongs to $CA(b_p, C, C + \delta_1)$.

Lemma 4. *The expected minimum e2e delay of a randomly chosen node i within the circular annulus $CA(b_p, C, C + \delta_1)$ such that the effective forwarding region of communication, EC_i, is expected to contain neighbors which can directly communicate with base-station, is $D_{i,k,w} = \sum_{h=1}^{\lfloor \frac{1/w}{t_I} \rfloor} p_{h,k,w} * h + t_D$, where $k = EC_i * \lambda - 1$ and λ denotes the density.*

Proof. Since $C < dis_j \leq C + \delta_1$ and the effective forwarding region of communication EC_i, is expected to contain neighbors which can directly communicate with the base-station, then the expected number of sensors belong to this area is $||EC_i|| * \lambda - 1$. Since these nodes have minimum e2e delay t_D, the overall e2e delay of node i decreases if all nodes are included in its forwarding set. Hence, the minimum expected e2e delay of node i is,

$$D_{i,k,w} = \sum_{h=1}^{\lfloor \frac{1/w}{t_I} \rfloor} p_{h,k,w} * h + t_D, \tag{3.7}$$

where $k = ||EC_i|| * \lambda - 1$. □

We gradually increase the distance from the base-station in steps of γ, and estimate the minimum expected e2e delay. We calculate the expected e2e delay of a random node i at a distance $C + \delta_1 + m\gamma$ for $m \in N$, using the estimated expected e2e delay of the nodes which are within the distance $C + \delta_1 + (m-1)\gamma$ from the base-station.

We divide the effective forwarding region of communication, EC_i, into several sectoral annuli such that every sectoral annulus is expected to contain only one node, as shown in Fig. 3.3(f), for $1 \leq j \leq k$, where $k = ||EC_i|| * \lambda - 1$. The j^{th} sectoral annulus $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$ of node i, between two concentric circles, centered at b_p with radii $\beta_{i_{j1}}, \beta_{i_{j2}}$, such that $\beta_{i_{j1}} > \beta_{i_{j2}}$, is defined as the intersection of the area between their boundaries and EC_i.

Let i_1 denotes the closest sectoral annulus to the base-station. $\beta_{i_{11}}$ is equal to $dis_i - C$. $\beta_{i_{12}}$ can be found by solving the following equation.

$$CI(dis_i, \beta_{i_{12}} - (dis_i - C)) = \frac{1}{\lambda}. \tag{3.8}$$

Note that $\beta_{i_{21}} = \beta_{i_{12}}$. Moreover, $\beta_{i_{j1}} = \beta_{i_{(j-1)2}}$, for $2 \leq j \leq k$. For an arbitrary i_j, $\beta_{i_{j2}}$ can be calculated by solving the following equation.

$$CI(dis_i, \beta_{i_{j2}} - \beta_{i_{j1}}) = \frac{1}{\lambda}, \tag{3.9}$$

for $1 \leq j \leq k$. We assume the equations 6.8 and 6.9 can be solved in constant time because these are single variable equations.

We first estimate the expected minimum e2e delay of a random node within j^{th} sectoral annulus, for $1 \leq j \leq k$, and use these to estimate the expected minimum e2e delay of a random node i.

Consider a random sectoral annulus $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$. Assume m_{i_1} be the largest integer such that $C + \delta_1 + m_{i_1}\gamma \leq \beta_{i_{j1}}$ and m_{i_2} be the smallest integer such that $C + \delta_1 + m_{i_2}\gamma \geq \beta_{i_{j2}}$. In order to calculate the expected e2e delay of a random node belongs to $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$, we use the expected e2e delay of nodes at distances $C + \delta_1 + (m_{i_1} + 1)\gamma, C + \delta_1 + (m_{i_1} + 2)\gamma, ..., C + \delta_1 + m_{i_2}\gamma$. The node can belong to any one of the areas induced by the intersection between $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$ and the ring formed by the circular annuli $CA(b_p, C+\delta_1+t\gamma, C+\delta_1+(t+1)\gamma)$, where $m_{i_1} \leq t \leq (m_{i_2} - 1)$. In fact, the estimated minimum expected e2e delay of a random node belongs to $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$ is proportional to the area induced by the intersection between $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$ and the ring formed by the corresponding circular annuli.

Area induced by $SA_{i,j}(C+\delta_1+m_{i_1}\gamma, C+\delta_1+(m_{i_1}+1)\gamma)$ is $CI(dis_i, dis_i-(C+\delta_1+(m_{i_1}+1)\gamma))-\frac{1}{\lambda}(k-j)$. Moreover, the area induced by $SA_{i,j}(C+\delta_1+t\gamma, C+\delta_1+(t+1)\gamma)$ for $m_{i_1} < t \leq (m_{i_2} - 1)$ is

$$||SA_{i,j}(C + \delta_1 + t\gamma, C + \delta_1 + (t + 1)\gamma)|| = CI(dis_i, dis_i - (C + \delta_1 + t\gamma)) - \sum_{s=1}^{t-1} \|SA_{i,j}(C + \delta_1 + s\gamma, C + \delta_1 + (s + 1)\gamma)\| - \frac{1}{\lambda}(k - j).$$

The probability of node i_j belongs to $SA_{i,j}(C + \delta_1 + t\gamma, C + \delta_1 + (t + 1)\gamma)$ is

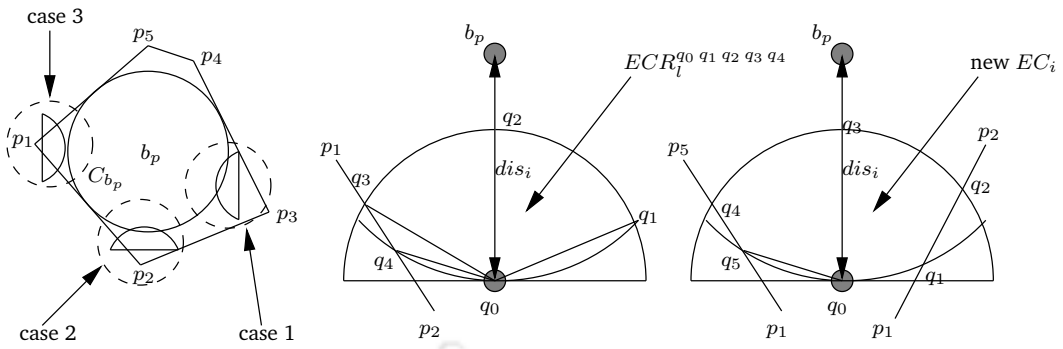
$||SA_{i,j}(C+\delta_1+t\gamma, C+\delta_1+(t+1)\gamma)||\lambda$. Let $D_{i_j,t\gamma}$ denotes minimum expected e2e delay of a random node within $SA_{i,j}(C+\delta_1+t\gamma, C+\delta_1+(t+1)\gamma)$. Hence, an upper bound for the expected e2e delay D_{i_j} of node i_j, is $\sum_{t=m_{i_1}}^{(m_{i_2}-1)} ||SA_{i,j}(C+\delta_1+t\gamma, C+\delta_1+(t+1)\gamma)||\lambda D_{i_j,t\gamma}$.

We use expected e2e delay D_{i_j} of a random node belongs to ${i_j}^{th}$ sectoral annulus, to find the minimum expected e2e delay of node i at a distance $dis_i = C+\delta_1+m\gamma$. Consider a randomly chosen node i at a distance $dis_i = C+\delta_1+m\gamma$ from the base-station. We divide the effective forwarding region of communication, EC_i, into $EC_i * \lambda - 1$ sectoral annuli such that every sectoral annulus is expected to contain one node. Assuming D_{i_j} denotes the estimated minimum expected e2e delay for a randomly selected node within circular annulus i_j, for $1 \leq j \leq (EC_i * \lambda - 1)$, which is calculated as shown earlier using the minimum expected e2e delay of nodes at distances $C+\delta_1+\gamma, C+\delta_1+2\gamma, ..., C+\delta_1+(m-1)\gamma$. A linear search over the nodes at every sectoral annulus, with higher priority given to nodes closer to the base-station effectively selects k' required number of nodes in the forwarding set, that minimizes overall e2e delay [46]. Hence, minimum expected e2e delay of node i is upper bounded by,

$$D_{i,k',w} = d_{k',w} + \sum_{j=1}^{k'} \frac{1}{k'} * D_{i_j}, \tag{3.10}$$

where $d_{k',w} = \sum_{h=1}^{\left\lfloor \frac{1/w}{t_I} \right\rfloor} P(W_{h,k'}) * h + t_D$. Hence, follows the theorem.

Theorem 1. *Assume D_{i_j} denotes the estimated minimum expected e2e delay for a randomly selected node within circular annulus $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$ of node i at distance*



(a) EC_i is intersected by FoI (b) case 2 (c) case 3

Figure 3.4: Estimation of new EC_i

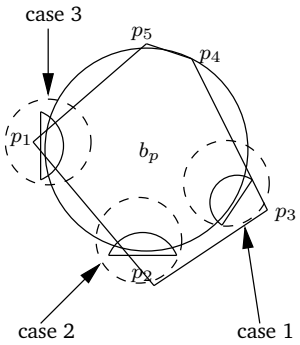
$dis_i = C + \delta_1 + m\gamma$ *from the base station, for* $1 \leq j \leq (EC_i * \lambda - 1)$. *Let* k' *nodes are included in the forwarding set. An upper bound on minimum expected e2e delay of node* i *is* $D_{i,k',w} = d_{k',w} + \sum_{j=1}^{k'} \frac{1}{k'} * D_{i_j}$, *where* $d_{k',w} = \sum_{h=1}^{\left\lfloor \frac{1/w}{t_I} \right\rfloor} P(W_{h,k'}) * h + t_D$.

In order estimate the maximum of minimum e2e delay in a circular-shaped FoI, we gradually increase the distance (such that $dis_i > \delta_1$) of a random node i from the base-station, by γ, and estimate the minimum expected e2e delay. This process continues till we reach the farthest point, which is at a distance equal to the radius of FoI. Overall procedure is depicted in Algorithm 1.

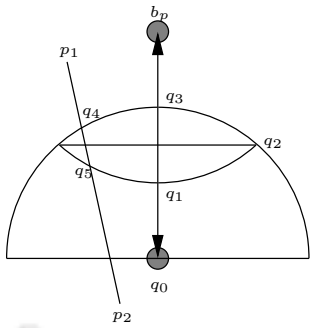
Algorithm 1: Estimates maximum e2e delay of a circular shaped FoI with radius r

Output: D_r: Maximum e2e delay in the FoI with radius r

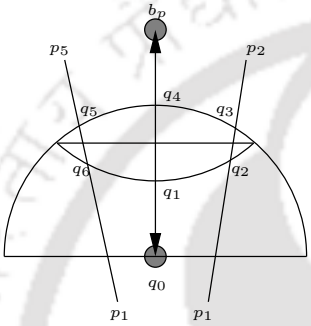
Estimate t_D, δ_1 using Lemma 4
Set $m = 1$ and the distance of node i from base-station $dis_i = \delta_1$
do
    Set $dis_i = m\gamma + \delta_1$
    Find the required number of nodes (k') that minimizes e2e delay in EC_i using linear search
    Estimate expected e2e delay $D_{i,k',w}$ using Theorem 1
    $m++$
while $dis_i \leq r$
$D_{i,k',w} = D_r$



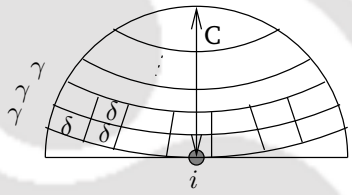
(a) CI is intersected by FoI



(b) case 2



(c) case 3



(d) Small parts within circular annulus

Figure 3.5: Estimation of e2e delay in a convex-shaped FoI

Estimation of e2e delay of a Convex-shaped FoI

Consider a convex shaped FoI $A =< p_1, p_2, p_3, p_4, p_5 >$, as shown in Fig. 3.4(a), with base-station located at b_p. Let r_A be the radius of the largest inscribed circle C_{b_p} inside A. The maximum expected e2e delay of C_{b_p} can be found using Algorithm 1. The steps involved to calculate the expected e2e delay of a node $i \in A - C_{b_p}$ is similar to subsection 3.2.1 except the boundary of the FoI intersects EC_i and $CI(dis_i, \delta)$.

The effective forwarding region of communication of the node i is the intersection of EC_i and A. There are three cases depending on the number of edges of the convex polygon A intersecting EC_i (refer Fig. 3.4(a)).

Case 1: If no edge is intersecting, then the effective forwarding region of commu-

nication is same as EC_i.

Case 2: If the effective forwarding region of communication is cut by only one edge of A as shown in Fig. 3.4(b), then it can be estimated as follows. Assume $\overline{a\,b}$ and $\widehat{a\,b\,c}$ respectively denotes the line segment joining points a and b, and the arc joining points a, b and c in counter clock-wise direction. Hence, the line segment $\overline{p_1\,p_2}$ cuts EC_i at q_3 and q_4. Let (x_i, y_i) be the co-ordinates of q_i. The effective forwarding region of communication, $ECR_i^{q_0\,q_1\,q_2\,q_3\,q_4}$, is enclosed between the line segment $\overline{q_3\,q_4}$, arc $\widehat{q_4\,q_0\,q_1}$, and arc $\widehat{q_1\,q_2\,q_3}$. Hence, the area $||ECR_i^{q_0\,q_1\,q_2\,q_3\,q_4}||$ is

$$\begin{aligned}||ECR_i^{q_0\,q_1\,q_2\,q_3\,q_4}|| = 2\cos^{-1}\left(1-\frac{C^2}{2{dis_i}^2}\right){dis_i}^2 - dis_i\sqrt{C^2-\left(\frac{C^2}{2dis_i}\right)^2} \\ + \left(cos^{-1}\left(\frac{y_3}{C}\right)+cos^{-1}\left(\frac{C}{2dis_i}\right)\right)C^2 + \triangle(q_3,q_4,q_0) \\ + \left(\cos^{-1}\frac{dis_i-y_4}{dis_i}\right){dis_i}^2 - \left(\frac{1}{2}x_4 dis_i\right),\end{aligned} \tag{3.11}$$

which can be verified by simple geometry.

Case 3: If EC_i is cut by two line segments of two different sides of FoI as shown in Fig. 3.4(c), then the effective forwarding area of communication is enclosed between the line segment $\overline{q_4\,q_5}$, arc $\widehat{q_5\,q_0\,q_1}$, line segment $\overline{q_1\,q_2}$, and arc $\widehat{q_2\,q_3\,q_4}$, which can also be found using similar method.

Next we estimate the area of the intersection of a circle centered at the base-station and the effective forwarding region of communication of a random node. In a similar way, there are three possible cases depending on the number of edges of the polygon A intersecting $CI(dis_i, \delta)$ (see Fig. 3.5(a)).

Case 1: When $CI(dis_i, \delta)$ is not intersected by an edge of A, then it is same as $CI(dis_i, \delta)$.

Case 2: If $CI(dis_i, \delta)$ is intersected by one edge (refer Fig. 3.5(b)), then the area is enclosed between the line segment $\overline{q_4\, q_5}$, arc $\widehat{q_5\, q_1\, q_2}$, and arc $\widehat{q_2\, q_3\, q_4}$. Using simple geometry this area can be calculated.

Case 3: Similarly, if $CI(dis_i, \delta)$ is cut by two line segments of two different sides of FoI (see Fig. 3.5(c)), then the intersected area is bounded by the line segment $\overline{q_5\, q_6}$, arc $\widehat{q_6\, q_1\, q_2}$, line segment $\overline{q_2\, q_3}$, and arc $\widehat{q_3\, q_4\, q_5}$.

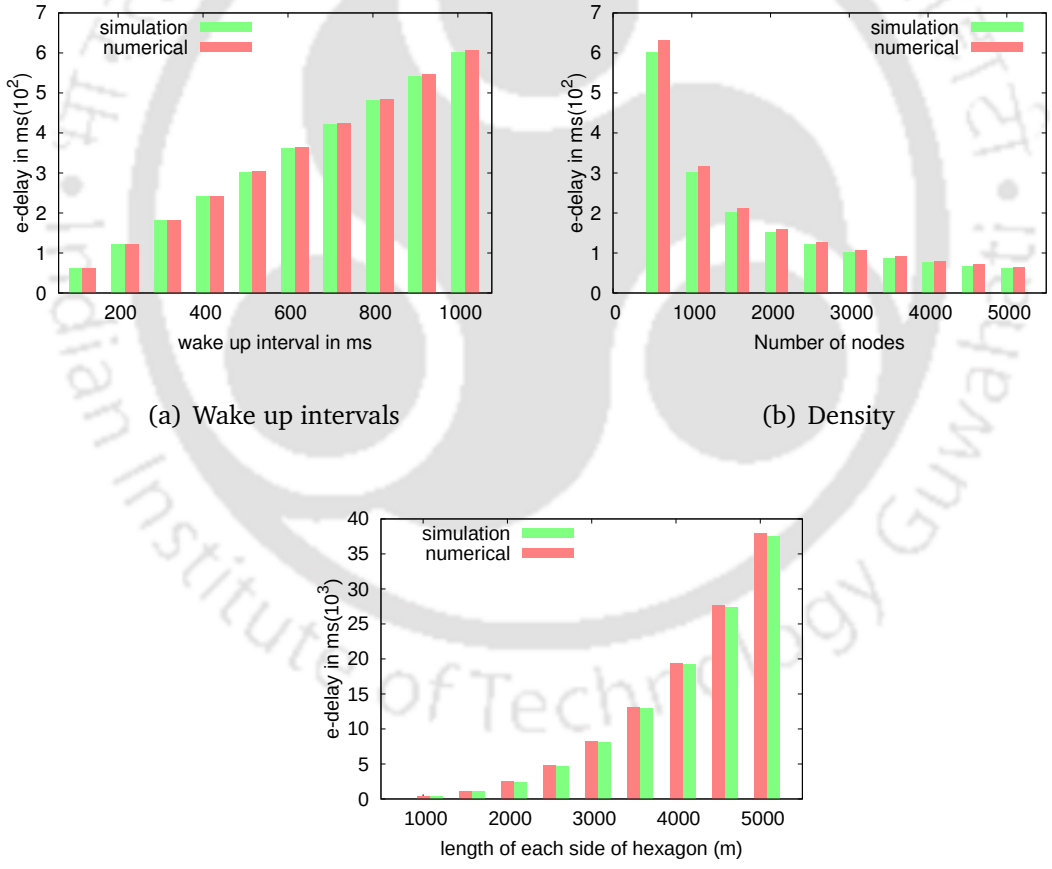
Now we can calculate the expected e2e delay of a random node $i \in A - C_{b_p}$. Depending on the position of the node i, the expected e2e delay changes within the same circular annulus, because the effective forwarding region of communication and the e2e delay of the next-hop nodes change within the same circular annulus. As mentioned earlier, the FoI is divided into several circular annuli in the steps of γ. In order to calculate the e2e delay of a random node within each sectoral annulus, we divide each sectoral annulus into several small parts of size δ as shown in Fig. 3.5(d). Assuming the expected e2e delay D_{δ_l}, of the node within the sub part δ_l of size $||\delta_l||$, within the j^{th} sectoral annulus of size $||SA_{i,j}||$ is known, one can find the expected e2e delay, $D_{i,j}$, of a random node within j^{th} sectoral annulus, using the following equation.

$$D_{i,j} = \sum_{\forall l} \frac{||\delta_l||}{||SA_{i,j}||} \times D_{\delta_l}. \tag{3.12}$$

The overall procedure is depicted in Algorithm 2.

Algorithm 2: Estimates maximum e2e delay of a convex-shaped FoI

1 Using Algorithm 1 estimate e2e delay of C_{b_p}
2 Set $m = 1, p = 1$ and $dis_i =$ radius of C_{b_p}
3 Set γ and δ respectively to be the steps of circular annulus and the steps of small part within each circular annulus
4 **do**
5 **do**
6 Estimate e2e delay of a node belongs within the small part of step δ
7 **while** *for all small parts of step* δ *within the circular annulus* $(dis_i + m\gamma, dis_i + m\gamma + 1)$
8 $m++$
9 **while** *for all circular annulus within the FoI*
10 Select the maximum e2e delay



(a) Wake up intervals

(b) Density

(c) FoI

Figure 3.6: Validating the estimated expected e2e delay

3.2.2 Validation of the Analysis using Monte-Carlo Simulation

In order to validate our analysis, in this subsection we numerically evaluate the expected e2e delay using Algorithm 2 for given density and wake-up rate, and compare with simulation results. Specifically, this section validates the analysis (and equations) by picking up its input from the same distribution used for the analysis, and run it for a large number of times to show that the analysis obtained is correct. We uniform randomly deploy 1000 nodes, with communication range of $200m$, in a circular area with radius 1000 m, and calculate the average e2e delay from the farthest node during each experiment. Note that the density is $\frac{1}{\pi \times 1000}$ nodes per m^2 area. We repeat such experiment by changing the seed of uniform distribution and redeploy the nodes.

Each experiment consists of several tests, where in each experiment we calculate the average e2e delay. Each test consists of the following things. Note that every node has equal wake-up interval as w. Let h_i denotes that node i wakes up at h^{th} beacon where h_i is uniformly distributed within $[1, h_{max}]$. In other words, node i wakes up at h_i beacon signal which is selected from a set of uniform randomly distributed integers within $[1, h_{max}]$. The forwarding set of node i, F_i, contains the neighboring nodes that are closer to the base-station and also within the communication range. Assume i_f denotes the node with the farthest distance from the base-station. The time when the critical event is generated at node i_f is selected from a set of uniform randomly distributed numbers within $[1, h_max]$. Let t_{i_f} denotes the time when the event is detected. Moreover, assume $i_{f_k} \in F_{i_f}$ is the forwarding node such that the difference between $h_{i_{f_k}}$ and t_{i_f} is minimum. In other words, i_{f_k} wakes up at the earliest after t_{i_f}. Hence, one hop delay between the event is detected

to the event is forwarded to the next hop node is $h_{i_{f_k}} - t_{i_f}$. In similar ways, we calculate the delay to reach the packet to the base station. We repeat such tests for 10,000 times, re-assign $h_i \forall i$, t_{i_f} and calculate the average e2e delay.

For simplicity, we place the base-station in the middle of the FoI. We repeat the experiment for 100 times. The average e2e delay obtained from the simulation results along with the numerical estimation are shown in Fig. 3.6. Our numerical estimation is close to the simulation results for various scenarios.

Impact of wake up interval: For different wake-up intervals, average e2e delays are shown in Fig. 3.6(a). We assume fixed density as $\frac{1}{\pi \times 1000}$ nodes per m^2 area. When the wake up interval increases, expected waiting time before sending a packet also increases. This in turn increases e2e delay. Also note that the estimated expected e2e delay is always higher than the average e2e delay obtained from the simulation.

Impact of density: Average e2e delays are shown in Fig. 3.6(b) while varying the number of nodes in the FoI, for a fixed wake-up interval $500\,ms$. If the number of nodes in the forwarding set increases, expected one hop delay decreases, which in turn decreases expected e2e delay. Though the difference in simulation and numerical estimation decreases as we increase the density, but the percentage of over estimation on the expected e2e delay is almost same.

Impact of FoI: The average maximum e2e delays, while varying the size of FoI are shown in Fig. 3.6(c), for a fixed wake up interval $500\,ms$ and fixed density $\frac{1}{\pi \times 1000}$ nodes per m^2 area. As the radius of circular-shaped FoI increases, e2e delay also increases rapidly. This is because, if the size of FoI increases, the number of hops from the farthest node increases as well, which in turn increases the e2e delay.

3.2.3 Minimum Cost Network for Lifetime and Delay Constraint

First we find the CSD for each area and then use this information to find the minimum cost network. Assume sensor nodes are deployed with the initial energy Q and consume average energy E_w during a wake up interval and the energy required for wake up. Moreover, E_{setup} denotes the required energy for setup and initial things before the wake-up process starts. The average wake up rate $w = \frac{Q-E_{setup}}{LE_w}$, where L is required lifetime constraint. The unit of L is same as the unit of time. Consider an example where w is 2 per second, $Q = 1000\,mJ$,$E_{setup} = 1\,mJ$, and $E_w = 10\,\mu J$. For these values, L is $49.95 \times 10^3\,s$.

If the maximum expected e2e delay D'_i in the area is greater than the delay constraint D_i, it is necessary to increase the density. The minimum sensor density λ_m is defined as the density required to satisfy the coverage requirement, which can be found using the methods described in [96, 97, 65]. In order to find the CSD λ required to satisfy given delay D_i, and lifetime constraint L, we formulate the problem as follows.

$$\min_{\lambda} \{||A|| * \lambda * C\} \quad subject\ to$$
$$D'_i \leq D_i, \lambda_m \leq \lambda_c, \quad and\ D_e, L, \lambda_c > 0,$$

where C denotes the cost of a sensor.

In order to find an upper bound λ_u for the CSD that satisfies given delay constraint D_i and lifetime requirement L, we exponentially increase the density λ from λ_m and find λ_u that satisfies the delay constraint using section 3.2.1, with the average wake

up rate $w = \frac{Q - E_{setup}}{LE_w}$. Next we use binary search between λ_u and $\frac{\lambda_u}{2}$ to find the CSD λ that minimizes the overall deployment cost.

The convex shaped FoI consists of k regions with delay constraint D_i for region A_i, for $1 \leq i \leq k$ (see Fig. 3.1). In order to find the minimum cost network, we find the CSD λ_i, for each A_i, for $1 \leq i \leq k-1$, using Algorithm 1, and for the area A_k, λ_k can be found using the method described in subsection 9.

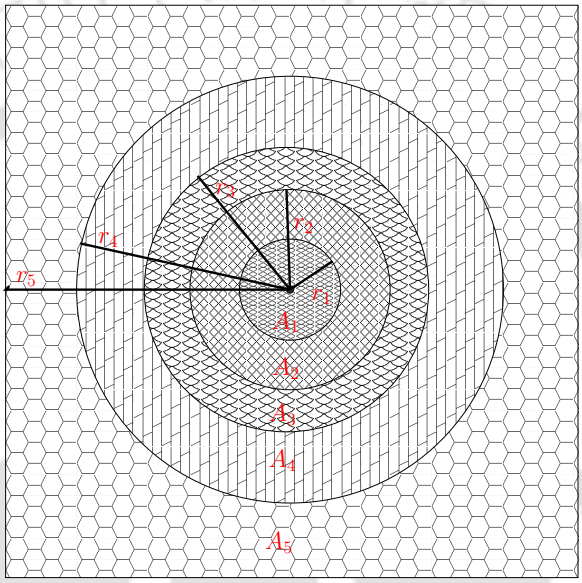


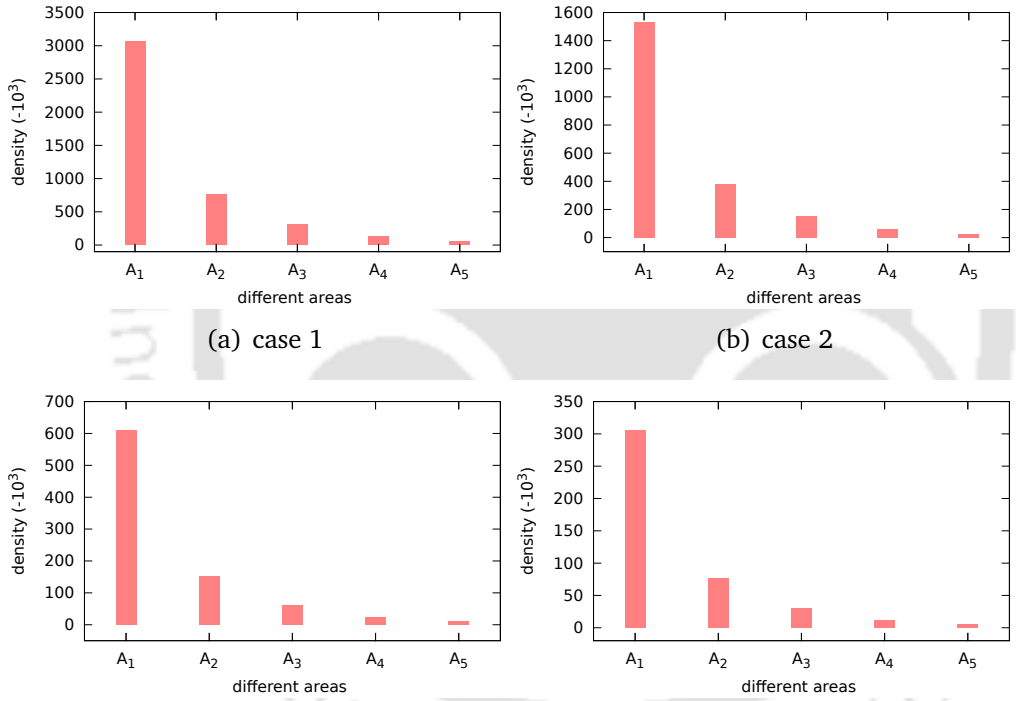
Figure 3.7: Simulation scenario

3.3 Simulation Results

In this section we show the effectiveness of our approach using ns2 simulation. We assume that the base-station (and the center facility) located in the middle of a square shaped FoI with side length $2\,r_5$. The FoI consists of five different areas A_1, A_2, A_3, A_4, and A_5, defined by radius, r_1, r_2, r_3, r_4, and r_5, as shown in Fig. 3.7. Moreover, the areas A_1, A_2, A_3, A_4, and A_5, are associated with delay constraints D_1, D_2, D_3, D_4, and D_5, respectively. We consider nine cases as shown in Table 3.1.

Table 3.1: Different cases for simulation

	A_1	A_2	A_3	A_4	A_5	w
	$r_1 = 200$	$r_2 = 400$	$r_3 = 800$	$r_4 = 1600$	$r_5 = 3200$	
Case 1	$D_1 = 1\,ms$	$D_2 = 5\,ms$	$D_3 = 25\,ms$	$D_4 = 125\,ms$	$D_5 = 625\,ms$	$10\,s$
Case 2	$D_1 = 1\,ms$	$D_2 = 5\,ms$	$D_3 = 25\,ms$	$D_4 = 125\,ms$	$D_5 = 625\,ms$	$5\,s$
Case 3	$D_1 = 1\,ms$	$D_2 = 5\,ms$	$D_3 = 25\,ms$	$D_4 = 125\,ms$	$D_5 = 625\,ms$	$2\,s$
Case 4	$D_1 = 1\,ms$	$D_2 = 5\,ms$	$D_3 = 25\,ms$	$D_4 = 125\,ms$	$D_5 = 625\,ms$	$1\,s$
Case 5	$D_1 = 1\,ms$	$D_2 = 6\,ms$	$D_3 = 36\,ms$	$D_4 = 216\,ms$	$D_5 = 1296\,ms$	$1\,s$
Case 6	$D_1 = 1\,ms$	$D_2 = 7\,ms$	$D_3 = 49\,ms$	$D_4 = 343\,ms$	$D_5 = 2401\,ms$	$1\,s$
Case 7	$D_1 = 1\,ms$	$D_2 = 8\,ms$	$D_3 = 64\,ms$	$D_4 = 512\,ms$	$D_5 = 4096\,ms$	$1\,s$
Case 8	$D_1 = 1\,ms$	$D_2 = 9\,ms$	$D_3 = 81\,ms$	$D_4 = 729\,ms$	$D_5 = 6.56\,s$	$1\,s$
Case 9	$D_1 = 1\,ms$	$D_2 = 10\,ms$	$D_3 = 100\,ms$	$D_4 = 1\,s$	$D_5 = 10\,s$	$1\,s$

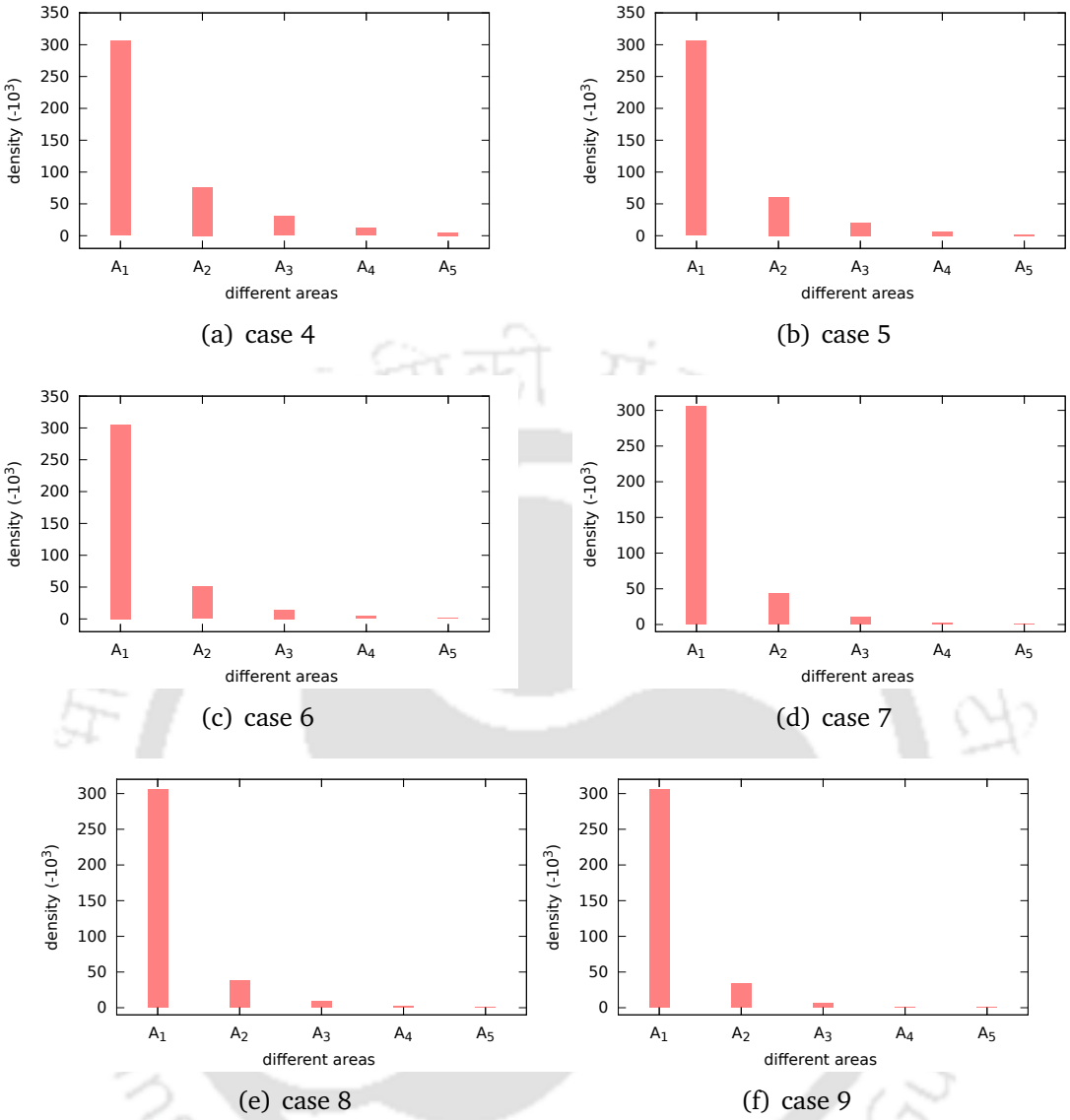


(a) case 1 (b) case 2

(c) case 3 (d) case 4

Figure 3.8: Density for differentiated QoS cases: varying wake-up interval

We also assume that the sensor nodes follow anycasting forwarding strategy [46]. Also note that, we run the simulation for 20 times to calculate the average CSD that satisfies the given constraint. The unit of density is number of sensors per unit area. Other parameters used in the simulation are given in Table 5.2.



(a) case 4 (b) case 5

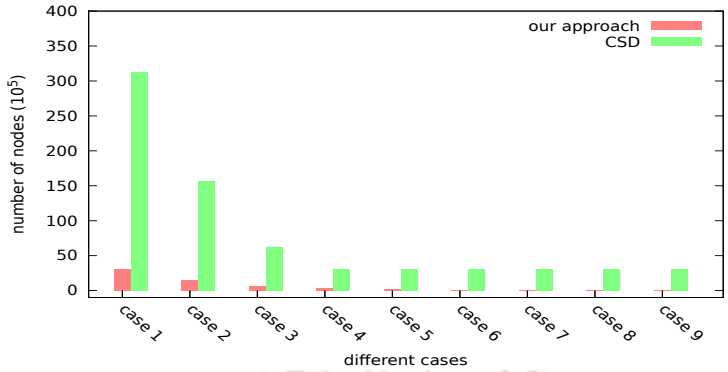
(c) case 6 (d) case 7

(e) case 8 (f) case 9

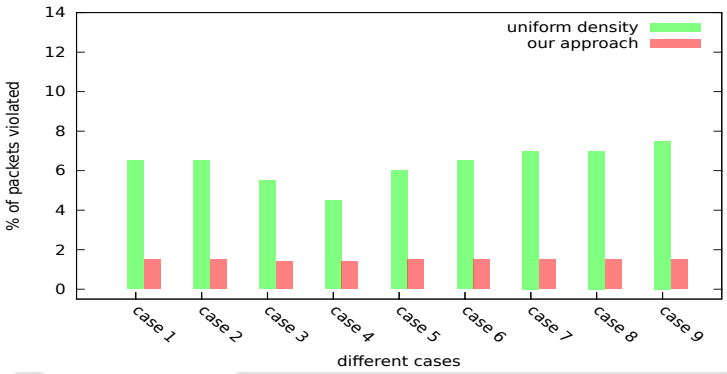
Figure 3.9: Density for differentiated QoS cases: varying rate of change of delay constraint

Table 3.2: Simulation parameters

Communication range	$100\,m$
Data rate	19.2 kbps
Transmission Power	19.5 mW
Receiving/Idle Power	13.0 mW
Data packet length	8 bytes
Control packet length	3 byte
Wireless Media	802.15.4



(a) cost



(b) packets violated delay constraint

Figure 3.10: Comparing different QoS with other approaches

3.3.1 Densities Vs Delay Constraint

Here we show that there exist substantial difference in densities for given delay constraints. Different CSDs for different cases are shown in Fig. 3.8,3.9. If the delay increases, the density decreases exponentially. This is because, if the density decreases, one hop delay increases which in turn increases e2e delay. In other words, lower density is sufficient to satisfy higher delay.

It can be noted from Fig. 3.8 that if the wake-up interval (and the lifetime) decreases, then CSDs for corresponding areas decrease as well. If the wake-up interval decreases, expected number of nodes in the forwarding set increases, which decreases one-hop and e2e delay. As a result, for a fixed delay constraint, one can

relax the density if wake-up interval decreases. Note that, if FoI remains same, the rate at which density changes only depends on the rate at which delay constraint changes. This can be observed in Fig. 3.8, where the rate at which delay constraint increases remains same, even though the wake-up interval decreases. In the contrary, if the rate at which delay constraint changes, increases (refer Fig. 3.9), the rate at which density changes also increase.

3.3.2 Minimizing Network Cost

In order to show the effectiveness in minimizing the cost, we compare the total number of nodes required to satisfy delay and lifetime constraint in our approach with the uniform critical sensor density (CSD) approach, and show the results in Fig. 3.10(a). We fix the uniform density across the FoI that satisfies given delay and lifetime constraints in all areas using the methods described in 3.2.3. Note that, if we assume the closest area is with highest priority (as in Table. 3.1), then in uniform CSD approach the sensor nodes are deployed across the FoI with the same density as the density required to satisfy delay constraint in the closest area to the base-station.

It can be observed from the figure that our approach minimizes the total number of nodes significantly in all cases. Moreover, saving in the network cost increases if wake-up interval increases. If wake-up interval increases, one-hop delay increases, which in turn increases e2e delay. Hence, for a fixed delay constraint, density must be increased to satisfy given delay constraint and lifetime requirement. In addition, if the wake-up interval increases, the rate at which density changes across all areas, increases as well. Hence, savings in our approach is higher for higher wake-up

interval if other parameters are fixed. Also note that, if the rate of change of delay constraint increases, our approach reduces cost further. This is because if this rate increases, the difference in the density increases which can be seen in Fig. 3.9.

3.3.3 Satisfying delay and lifetime constraint

We fix the density across the area and show the effectiveness for satisfying delay and lifetime constraints, in our approach. In order to compare the results, we deploy the same number of nodes using uniform density across the FoI and show the percentage of packets violated the delay constraint if wake-up interval (for given lifetime constraint) is fixed as 1 s. In Table. 3.3 we show the different densities for different cases in uniform-deploy and our approach. Note that, in both of the approaches the total number of sensors deployed is same. Hence, the cost remains same in both of the approaches. The results are shown in Fig. 3.10(b). Compared to uniform density, our approach performs better in satisfying delay constraints.

3.4 Conclusion

In this work, we estimated the maximum expected e2e delay for convex-shaped FoI with the given density and used this analysis to find the CSDs that satisfy given spacial differentiated delay constraint and lifetime requirement. As a result of the imperfect oscillator or some external reasons, the frequency of the crystal oscillator embedded in each sensor varies unpredictably, which is known as clock-skew [85]. Because of clock-skew the wake-up rate may vary unpredictably which may increase overall e2e delay. As a result, the delay constraint may not always be satisfied. In

Table 3.3: Density for different cases ('U'-uniform approach, 'O'-our approach)

	A_1	A_2	A_3	A_4	A_5	Approach
Case 1	0.297	0.297	0.297	0.297	0.297	U
Case 1	3.0570	0.7642	0.3056	0.1222	0.0488	O
Case 2	0.148	0.148	0.148	0.148	0.148	U
Case 2	1.5284	0.3820	0.1527	0.0610	0.0243	O
Case 3	0.059	0.059	0.059	0.059	0.059	U
Case 3	0.6113	0.1527	0.0610	0.0243	0.0097	O
Case 4	0.029	0.029	0.029	0.029	0.029	U
Case 4	0.3056	0.0763	0.0305	0.0121	0.004	O
Case 5	0.018	0.018	0.018	0.018	0.018	U
Case 5	0.3056	0.0610	0.0203	0.0067	0.0022	O
Case 6	0.012	0.012	0.012	0.012	0.012	U
Case 6	0.3056	0.0508	0.0144	0.0040	0.0011	O
Case 7	0.010	0.010	0.010	0.010	0.010	U
Case 7	0.3056	0.0436	0.0108	0.0026	0.0006	O
Case 8	0.008	0.008	0.008	0.008	0.008	U
Case 8	0.3056	0.0381	0.0084	0.0018	0.00035	O
Case 9	0.007	0.007	0.007	0.007	0.007	U
Case 9	0.3056	0.0339	0.0067	0.0012	0.0002	O

the next chapter, our objective is to constrain this increase in delay as a result of clock-skew within given threshold.

4

Satisfying end-to-end delay Constraint in Event-driven data-gathering in Presence of Clock-skew

4.1 Introduction

In the previous chapter, we worked on the problem of finding minimum cost network for event-driven data-gathering applications like tsunami, forest fire and seismic event detection where the sensors remain idle until an (rare) event occurs and once detected the event information needs to be forwarded to base-station within

strict delay constraint.

In particular, we estimated the CSD for given delay constraint and lifetime requirement. However, the delay constraint may not always be satisfied because of clock-skew[1]. In order to circumvent the effect of clock-skew, several time synchronization protocols are proposed in the literature.

Generally in the time synchronization protocols, sensor nodes estimate expected phase-offset [2] and expected clock-skew to synchronize time. However, clocks may diverge as a result of estimation error involved in synchronization. Moreover, several environmental factors like temperature, pressure, radiation, magnetic fields, etc. may vary the clock-skew. Frequent synchronization may reduce the clock divergence, but it is not energy efficient for the rare event detection scenario. Clock-skew may vary the actual periodic wake-up interval over time and packets may violate e2e delay constraint. None of the anycasting strategies [18, 68, 69, 46, 45] considered the effect of clock-skew in e2e delay to the best of our knowledge, that may lead to violate the delay-constraint in time-critical applications. This limitation in the literature motivates us to find the critical wake-up rate that constrains the overall increase in e2e delay.

The advancement in sensor technology allows the sensors to be heterogeneous, in terms of different maximum clock-skew associated with them. Also note that, when clock-skew is very low (sensors with the high precision hardware clock), the effect of clock-skew on overall e2e delay becomes negligible. However, if the clock-skew is high, may be beyond an acceptable threshold, it may pose severe consequences. Hence, we assume that the amount of e2e delay increases as a result of clock-skew,

[1]As a result of imperfect oscillator or some external reasons, the hardware oscillator frequency in a sensor node varies unpredictably, which is known as clock-skew [85].

[2]Phase-offset refers to the difference in time between two clocks at a time instance.

beyond an acceptable threshold ξ, is a given constraint. In this chapter, we address the following problem *"maximize the lifetime while controlling the wake-up rate to constrain the increase in e2e delay, as a result of heterogeneous clock-skew present in WSN, within a given threshold ξ ?"*. Assuming N nodes are deployed, we can formally write our objective as

$$\begin{aligned} &\max_{w_1,w_2,...,w_N} \{max(L_i)\forall i\} \quad \text{subject to} \\ &L_i = \frac{Q_i}{E_w \times w_i}, \forall i, \\ &\delta D_i(\vec{f}) \leq \xi, \forall i, \\ &Q_i > 0, E_w > 0, w_i > 0, \end{aligned} \tag{4.1}$$

where L_i denotes the lifetime of node i, Q_i denotes the available energy at node i for event-driven operation, E_w denotes the average amount of energy consumption (if no event detected) during a wake-up and w_i denotes the wake-up rate. The estimation of increase in expected e2e delay as a result of clock-skew of node i, $\delta D_i(\vec{f})$ for given forwarding vector$\vec{f}$, is shown in the following sections. Note that if $max(\delta D_i(\vec{f})) < \xi$, then $\delta D_i(\vec{f}) < \xi, \forall i$. In order to maximize $max(L_i)\forall i$, we maximize L_i for all i. Hence Eq. 4.1 can be written as,

$$\max_{w_i} \{L_i\} \quad \text{subject to}$$
$$L_i = \frac{Q_i}{E_w \times w_i}, \forall i,$$
$$max(\delta D_i(\vec{f})) < \xi,$$
$$Q_i > 0, E_w > 0, w_i > 0, \tag{4.2}$$

for all i. As Q_i and E_w are constant for node i, this problem is same as minimizing w_i, hence we can write the Eq. 4.2 as,

$$\min_{w_i} \{w_i\} \quad \text{subject to}$$
$$max(\delta D_i(\vec{f})) < \xi,$$
$$w_i > 0. \tag{4.3}$$

We estimate the increase in e2e delay as a result of heterogeneous clock-skew using stochastic analysis. We use this estimation to find the critical wake-up rate for all nodes to constrain the increase in overall e2e delay within given threshold ξ. We validated the analysis using Monte-carlo simulation. We also shown the effectiveness of our approach by simulation using ns2.

4.2 Minimizing Delay in Anycasting Forwarding as a Result of Clock-skew

In this section, first we derive an expression for the expected increase in expected e2e delay in delay-optimal anycasting forwarding scheme [45] and use this analysis to find the critical wake-up rate, $\bar{r}_i$ for each node i in the network, that constrain the increase in overall e2e delay to within given threshold ξ. Since our approach uses delay optimal anycasting forwarding policy, we briefly discuss the method for the sake of completeness.

Assume, node i needs to forward a data-packet. The node i first sends a beacon signal of duration t_B, followed by an ID signal of duration t_C, and listens for acknowledgment of duration t_A as shown in Fig.3.2, where $t_I = t_A + t_B + t_C$ denotes the duration of an epoch. The node i keeps on sending the beacon signal until it receives an acknowledgement.

Let N_i denotes the set of neighbors of node i. Every node $j \in N_i$ follows a periodic wake-up rate r_j and forwarding policy to satisfy the given delay constraint. If j wakes up and hears h^{th} beacon and ID signal, then node j can either choose to receive the packet by sending its ID during t_A, or go back to sleep and wakes up at the next interval. If node $j \in F_i$ where F_i denotes the forwarding set of node i, then it sends an acknowledgement otherwise it goes back to sleep and wakes up at the next interval. The s/w interval for the node j is determined by the s/w scheduling policy chosen by the node j.

Here we briefly define the forwarding policy of node i, the details of which can be found in [45]. Before going into the details, note that only the neighbors are added

in the forwarding set which collectively minimizes overall e2e delay. But in practice, when an actual event occurs the event information packet may follow a longer route. In order to minimize the path length, nodes do not immediately forward the packet, instead they wait for some time and then opportunistically forward only when expected delay involve for waiting is more. Hence, the forwarding policy may change for different beacons. Let $f_i(h)$ be the forwarding policy of the node i at h^{th} beacon signal. If $f_i(h) = j$ then the node i sends the packet to any node with rank higher or equal to the rank of node j in beacon interval h. The nodes in the forwarding set are given ranks according to their distance from the base-station. Hence, if a node is closer to the base-station, it has higher rank. For ease of understanding, we re-define the notations in Table 4.1.

4.2.1 Expected Increase in End-to-end Delay

Let $F_i \subseteq N_i$ be the forwarding set of i, where N_i is the set of neighbors of node i. The probability of any node $j \in F_i$ wakes up at h^{th} beacon signal is

$$p_{j,h} = \begin{cases} \frac{t_I}{1/r_j - (h-1)t_I} & \text{if } h \leq \left\lceil \frac{1/r_j}{t_I} \right\rceil, \\ 1 & \text{otherwise,} \end{cases} \tag{4.4}$$

where $t_I = t_A + t_B + t_C$.

In order to find an expression for the expected increase in e2e delay as a result of the clock skew, we first need to find an expression for the expected increase in one hop delay with given forwarding strategy. Let $f_i =< f_i(1), f_i(2), ..., f_i(\infty) >$ be the forwarding policy of node i such that $f_i(h) = j$. The packet is forwarded at the h^{th} beacon signal, only if the packet is not forwarded during $h-1$ beacons and any

Table 4.1: Shows details of the notations

clk_j	clock-skew rate of node j
Δ	e-delay constraint
$d_i(f)$	expected one-hop delay of node i
$d'_i(f)$	expected one hop delay with clock-skew
$\delta d_i(f)$	increase in expected one-hop delay as a result of clock-skew
$\delta D_i(f)$	increase in expected e-delay as a result of clock-skew
f	global forwarding vector
f_i	forwarding policy of node i
$f_i(h)$	forwarding policy of i at h^{th} beacon
F_i	forwarding set of node i
h_{max}	minimum number of beacons for which the packet is surely forwarded
h'_{max}	minimum number of beacons for which the packet is surely forwarded in presence of clock-skew
N	set of all nodes in the given network
N_i	neighbors of node i
$p_{j,h}$	probability of node $j \in F_i$ wakes up at h^{th} beacon
$p'_{j,h}$	probability of any node j wakes up at h^{th} beacon with clock-skew clk_j
$P_i(h\|f_i(h) = j)$	probability of packet is forwarded at h^{th} beacon
$P_i(h\|f_i(h) = j)$	probability of packet is forwarded at h^{th} beacon in presence of clock-skew
$q_{i,j}$	probability of a packet is forwarded to node j
$q_{i,j}(h)$	probability of sending a packet to node j at h^{th} beacon
r_i	optimal wake-up rate for node i
r'_i	minimum wake-up rate due to clock-skew clk_i
r''_i	critical wake-up rate to satisfy delay constraint
ξ	threshold for increase in e-delay as a result of clock-skew
t_A	duration of acknowledgment signal
t_B	duration of beacon signal
t_C	duration of ID signal
t_I	duration of an epoch

node with equal or higher rank of j wakes up at h^{th} beacon signal. The probability of any node with equal or higher rank of j, wakes up at the beacon signal h, is $1 - \prod_{k=1}^{j}(1 - p_{k,h})$. Hence, the probability of the packet is forwarded at h^{th} beacon to the next hop node, such that $f_i(h) = j$ is

$$P_i(h|f_i(h) = j) = (\prod_{l=1}^{h-1}(1 - P_i(l|f_i(l) = q))) \times (1 - \prod_{k=1}^{j}(1 - p_{k,h})). \quad (4.5)$$

Moreover, if $f_i(h) = j$ for some h such that $i \neq j$, $k \leq j$ and $h > \frac{1/r_k}{t_I}$, then the packet is surely forwarded at h^{th} beacon. The minimum value of h for which the packet is surely forwarded is denoted by h_{max}. Therefor, $\sum_{h=1}^{h_{max}} P_i(h|f_i(h) = j) \geq 1$. The expected one hop delay of the node i, $d_i(\vec{f})$, for given global forwarding vector $\vec{f} = (f_1, f_2, ..., f_N)$, is the sum of the duration W_i, duration of the expected number of beacons it waits, and transmission time. The expected number of beacons the node i waits is given by $\sum_{h=1}^{h_{max}} P_i(h|f_i(h) = j)(h)t_I$. Hence, the expected one hop delay of the node i, $d_i(\vec{f})$, is

$$d_i(\vec{f}) = \sum_{h=1}^{h_{max}} P_i(h|f_i(h) = j)(h)t_I + t_D,$$
$$\text{where } \left(\sum_{h=1}^{h_{max}-1} P_i(h|f_i(h) = j)\right) \leq 1 \text{ and } \left(\sum_{h=1}^{h_{max}} P_i(h|f_i(h) = j)\right) > 1 \quad (4.6)$$

The value of h_{max} can be calculated by gradually increasing h and selecting the minimum h that satisfies $\sum_{h=1}^{h_{max}} P_i(h|f_i(h) = j) \geq 1$.

In homogeneous clock-skew, every node is associated with equal maximum clock-skew rate in the network. Whereas, in case of heterogeneous clock-skew present in the network, node j is associated with clock-skew rate upto clk_j. Hence, change in the wake up interval as a result of clock-skew of node j is in $\left[-clk_j \times \frac{1}{r_j}, clk_j \times \frac{1}{r_j}\right]$ and wake-up rate is in the range $\left[\frac{r_j}{1+clk_j}, \frac{r_j}{1-clk_j}\right]$. If wake-up rate is greater than or equal to r_j, there is no increase in e2e delay. On the other hand, the maximum increase in e2e delay occurs for the wake-up rate $r'_j = \frac{r_j}{1+clk_j}$. The equations corre-

sponding to Eq. 4.4 and 4.5 for r'_j are given in Eq. 4.7 and 4.8, respectively.

$$p'_{j,h} = \begin{cases} \frac{t_I}{1/r'_j} & \text{if } h \leq \left\lceil \frac{1/r'_j}{t_I} \right\rceil, \\ 1 & \text{otherwise.} \end{cases} \tag{4.7}$$

$$P'_i(h|f_i(h)=j) = (\prod_{l=1}^{h-1}(1 - P'_i(l|f_i(l)=q))) \times (1 - \prod_{k=1}^{j}(1-p'_{k,h})) \tag{4.8}$$

Accordingly, we can find h'_{max} for the given forwarding set and the new expected one hop delay $d_i(\vec{f})$ is,

$$d'_i(\vec{f}) = \sum_{h=1}^{h'_{max}} P'_i(h|f_i(h)=j)(h)t_I + t_D, \quad \text{where } \left(\sum_{h=1}^{h'_{max}-1} P'_i(h|f_i(h)=j)\right) \leq 1 \text{ and } \left(\sum_{h=1}^{h'_{max}} P'_i(h|f_i(h)=j)\right) > 1 \tag{4.9}$$

Assume forwarding policy for every node remains same although the wake-up rate changes as a result of clock-skew. Hence, the increase in expected one hop delay as a result of the clock skew is upper bounded by,

$$\delta d_i(\vec{f}) = d_i(\vec{f}) - d'_i(\vec{f}). \tag{4.10}$$

In optimal anycasting forwarding technique, the packet is forwarded to node j at beacon signal h only if $f_i(h) = k$ and all nodes with higher priority than j are asleep,

where $j \leq k$. Hence, the probability of sending the packet to node j at h^{th} beacon is given by

$$q_{i,j}(h) = \begin{cases} \prod_{l=1}^{j-1} \left(1 - p'_{l,h}\right) \; p'_{j,h}, \text{if } f_i(h) = k \text{ where } j \leq k, \\ 0 \qquad\qquad\qquad\qquad \text{otherwise.} \end{cases} \tag{4.11}$$

The probability of a packet is forwarded to node j within h'_{max} beacon signal is

$$q_{i,j} = \frac{\sum_{h=1}^{\left\lfloor \frac{h'_{max}}{t_I} \right\rfloor} q_{i,j}(h)}{\sum_{k \in F_i} \sum_{h=1}^{\left\lfloor \frac{h'_{max}}{t_I} \right\rfloor} q_{i,k}(h)}. \tag{4.12}$$

The expected increase in e2e delay as a result of clock-skew is the sum of expected increase in one hop delay and expected increase in next hop e2e delay. The expected increase in next hop e2e delay is the sum of the product of the probability of the packet is forwarded to the next hop node and expected increase in the next hop e2e delay. Every node i calculates its own expected increase in e2e delay as a result of the clock-skew as

$$\delta D_i(\vec{f}) = \left(\sum_{j \in F_i} q_{i,j} \times \delta D_j(\vec{f}) \right) + \delta d_i(\vec{f}). \tag{4.13}$$

4.2.2 Critical Wake-up Rate

In this subsection we use the expected increase in e2e delay as a result of the clock-skew to find the critical wake-up rate r''_i for each node i, to constrain the increase in delay within ξ. In other words, our method drastically decreases the number of

data-packets with overall e2e delay more than $\Delta + \xi$.

If the increase in e2e delay as a result of clock-skew is less than the threshold ξ, then the wake-up rate remains unchanged. If the increase is more, then the critical wake-up rate is calculated for every node i to satisfy the given threshold ξ.

The nodes closer to the base station encounter less increase in delay compared to the nodes far from the base station. Let k be the node with the maximum delay increase as a result of clock-skew. That is, $\delta D_k(\vec{f}) \geq \delta D_i(\vec{f})$ for all i in the network. If the packet emitting from node k satisfies ξ, then any other packet is expected to satisfy ξ as well. In order to constrain this delay within ξ, every node wakes up more frequently.

Peripheral nodes broadcast their own expected e2e delay increase as a result of clock-skew to the base-station. The base-station waits for some time to receive the expected e2e delay increase from the nodes at the periphery of the FoI. If the maximum e2e delay increase is more than ξ, the base-station broadcasts a message to revise the wake-up rate of all the nodes in the network. A node is on the periphery of the FoI if no other node's forwarding set contains this node. Since, each node maintains a set of nodes to which it is in their forwarding set, it can determine whether it is a periphery node or not.

The nodes in the direct communication range of the base-station set their expected e2e delay increase for clock-skew as zero. Every other node i waits to receive the expected e2e delay increase from the nodes in its forwarding set to calculate its own expected e2e delay increase ($\delta D_i(\vec{f})$). Every node calculates its own expected e2e delay only when the expected e2e delay increase of all the nodes in its forwarding set is received.

Algorithm 3: Every node i calculates new wake-up rate r_i''

if *Base station* $\in F_i$ **then**
  $\delta D_i(\vec{f}) = 0$
else
  if $\delta D_j(\vec{f})$ *for all* $j|j \in F_i$ *received* **then**
    $$\delta D_i(\vec{f}) = \left(\sum_{j \in F_i} q_{i,j} \times \delta D_j(\vec{f}) \right) + \delta d_i(\vec{f})$$
if *there exist a node* j *such that* $i \in F_j$ **then**
  Forwards $\delta D_i(\vec{f})$ to all $\{j|i \in F_j\}$
else
  i sends $\delta D_i(\vec{f})$ to base station
if *Receives message from base-station to revise wake-up rate* **then**
  Set $r_i'' = \frac{r_i}{1 - c l k_i}$

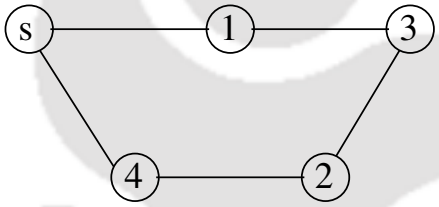


Figure 4.1: Simple example demonstrates our approach

Example 1: We consider a simple network as shown in Fig. 4.1 [45] to demonstrate our analysis. Let $t_I = 1$ and $t_D = 2$ for all nodes, and $1/r_1 = 1/r_2 = 1/r_3 = 50t_I$ and $1/r_4 = 3t_I$. Node $1, 2,$ and 4 respectively contains {s,4} and {s} in their corresponding forwarding set. Since the sink is always awake, $f_1(1) = f_4(1) =$s. Moreover, f_1 and f_4 are defined only for the first beacon. Whereas $f_2(1) = f_2(2) = ... = f_2(50) = 4$ as node 4 is the only forwarding node for 2. Node 3 contains node $1, 2$ in its forwarding set and its forwarding strategy is $f_3(1) = f_3(2) = ... = f_3(41) = 2$, and $f_3(42) = ... = 1$. The nodes are in direct communication range of base-station, that is 4 and 1 set $\delta D_4(\vec{f}) = \delta D_1(\vec{f}) = 0$ and forwards these values to node 2 and 3 respectively. When node 2 receives this information from node 4, it calculates $\delta D_2(\vec{f}) = 0 + \delta d_2(\vec{f}) = 0.000179$. Node 3 calculates $q_{3,2} = 0.44$, and

$q_{3,1} = 0.56$, and subsequently $\delta D_3(\vec{f}) = 0.44 * 0 + 0.66 * 0.000179 + \delta d_3(\vec{f}) = 0.002118$. Let $\xi = 1\,ms$, hence we get $\max_i(\delta D_i(\vec{f})) > \xi$. New wake up interval $1/r'_1 = 1/r'_2 = 1/r'_3 = (50 - 50 * 0.0001)t_I = 49.0001\,t_I$ for node $1, 2, 3$ and $1/r'_4 = 2.9997 t_I$ for node 4.

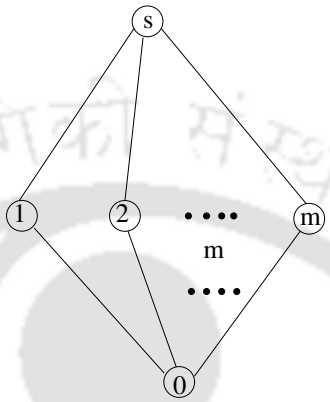


Figure 4.2: Validating our approach

4.2.3 Validation of the Analysis

In order to validate the expressions for the expected increase in one hop delay (Eq. 4.10) and e2e delay (Eq. 4.13) we evaluate the corresponding equations numerically, and compare with respective simulation results. Specifically, this section validates the equations by picking up its input from the same distribution used for the analysis, and run it for a large number of times to show that the analysis obtained is correct.

Expected Increase in One-hop Delay: Here we validate Eq. 4.10. For simplicity, we consider a simple network shown in Fig. 4.2. For the simulation results, a Monte Carlo simulation is conducted which is used to estimate the probability from a large number of experiments. Assume node 0 is sending critical data-packet to node s. Here, we validate the increase in one-hop delay of the critical data-packet generated

at node 0.

Each experiment consists of the following things. For simplicity, we assume that each node wake up at an interval of $w = 10$ seconds. Note that $f_i(h)$ denotes the forwarding policy of node $i \forall i$ at $h^{th} \forall h$ beacon signal. Let h_i denotes that the node i is scheduled to wake up at h^{th} beacon where h_i is uniformly distributed within $[1, h_{max}]$. In other words, the node i is scheduled to wake up at h_i beacon signal which is selected from a set of uniform randomly distributed integers within $[1, h_{max}]$. Hence, the forwarding nodes of node 0, node $1, 2, ..., m$, are scheduled to wake-up at $h_1, h_2, ..., h_m$ beacons. In order to incorporate the clock-difference for clock-skew, actual wake-up time of a node i are picked from a set of normally distributed random numbers with mean h_i and standard deviation $w \times clk_i$. The time when the critical event is generated at node 0 is selected from a set of uniform randomly distributed numbers within $[1, h_max]$. Let t_0 denotes the time when the event is detected. Moreover, assume $0_k \in F_0$ is the forwarding node such that the difference between h_{0_k} and t_0 is minimum. Here note that h'_{0_k} is the actual wake-up time which incorporates the effect of clock-skew and h_{0_k} denotes the scheduled wake-up time. Hence, one hop delay, with and without incorporating clock-skew, between the event is detected to the event is forwarded to the next hop node are respectively $h'_{0_k} - t_0$ and $h_{0_k} - t_0$. Hence we can calculate the one-hop delay increase for clock-skew. In similar ways, one can calculate the increase in e2e delay for clock-skew to reach the packet to the base station although the details of it is given in next sub-section. We repeat such experiments for 10,000 times by re-assign $h_i \forall i$, $h'_i \forall i$, t_{i_f} and calculate the average one-hop delay increase.

In actual simulation we set the number of nodes in forwarding set as $m = 10$. We vary the (homogeneous) clock-skew rate from 10 to 100 and compare the re-

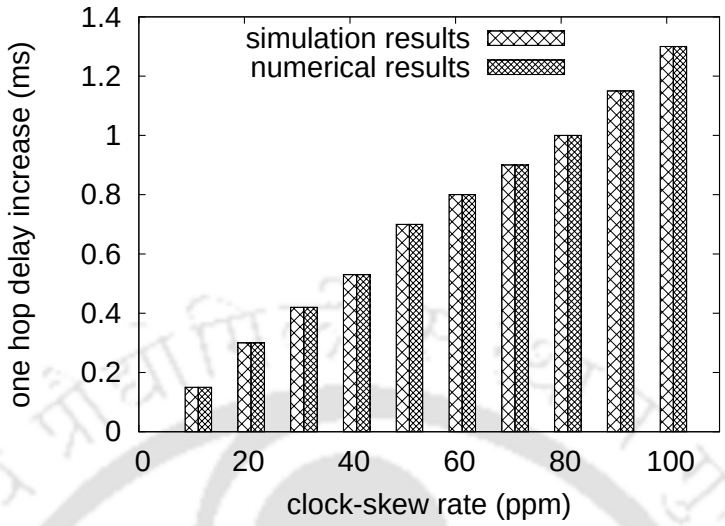


Figure 4.3: Validating expected increase in one-hop delay

sults obtained from simulation as shown in Fig.4.3. Note that, the numerical and simulation results match with no visible error bars. Moreover, increasing clock-skew increases one-hop delay because as clock-skew increases, wake-up rate may decrease which may increase expected increase in one-hop delay.

Validating Expected Increase in e2e delay: In order to verify the expression for an increase in expected e2e delay, we evaluate Eq. 4.13 numerically and compare it with that obtained in a simulation. We repeat the experiments several times where each experiment consists of several tests. In each experiment, we deploy 100 nodes uniformly at random in $1000 \times 1000\ m^2$ (rectangular) area with $200\ m$ communication range and calculate the average e2e delay from the farthest node during each experiment. We repeat such experiment by changing the seed of uniform distribution and redeploy the nodes. The sensor nodes follow optimal anycasting forwarding policy [45], with periodic wake-up interval 10 seconds. The farthest node is expected to have maximum increase in e2e delay as a result of clock-skew.

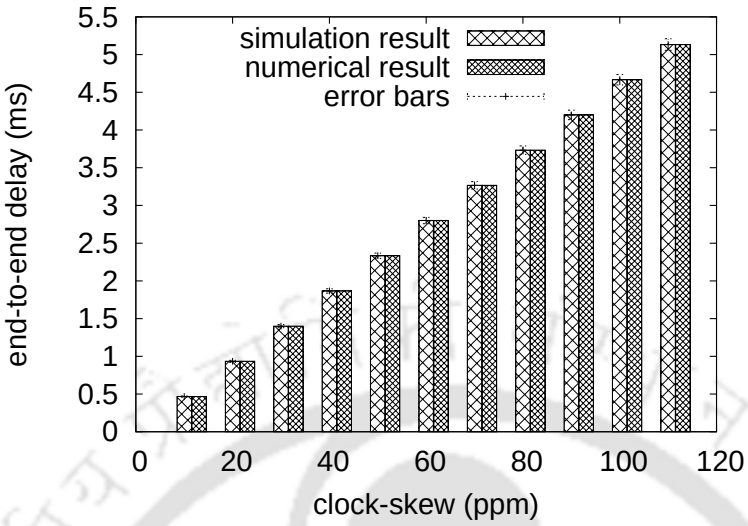


Figure 4.4: Validating expected increase in e2e delay

Each experiment consists of several tests where each test consists of the following things. Note that every node has equal wake-up interval as w. Let h_i denotes that node i wakes up at h^{th} beacon where h_i is uniformly distributed within $[1, h_{max}]$.

Hence, we select the farthest node and generate an event to calculate the maximum increase in e2e delay.

We repeat the experiment 100 times by generating multiple events and calculate the average increase in e2e delay for different clock-skew (see Fig. 4.4). Increasing clock-skew increases expected e2e delay because as clock-skew increases expected one-hop delay increases which in turn increases expected e2e delay. Although the size of error-bar increases as clock-skew increases, the percentage of size of error-bars for e2e delay remains same as we increase clock-skew.

Validating Expected Increase in e2e delay while varying FoI: In order to show the affect of our analysis in a large FoI, in Fig. 4.5 we varied the length of the

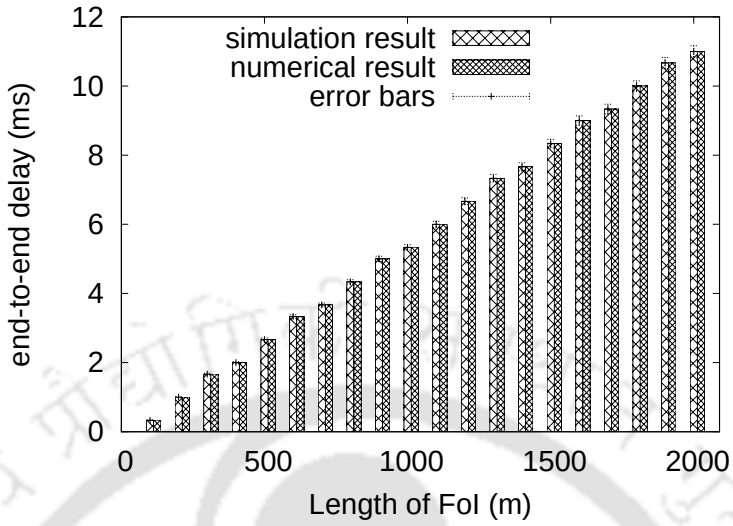


Figure 4.5: Validating expected increase in e2e delay

(square-shaped) FoI and show the average increase in e2e delay. We set the clock-skew as 100 ppm. Note that, if FoI increases e2e delay increases as well. This is because, for a large FoI the packets traverse through more number of hops, which increases overall increase in e2e delay for clock-skew. It can also be noted that, for a large FoI the error bars are more visible. As the clock-skew follows normal distribution with zero mean, increasing FoI increases the standard deviation of the sum of these normal variables, and hence increases the error bars.

4.3 Simulation Results

Table 4.2: Shows details of the parameters used for ns2-based simulation

Communication range	$250\,m$
Data rate	19.2 kbps
Transmission/Receiving Power	13.0 mW
Idle Power	13.0 mW
Data packet length	8 bytes
Control packet length	2 byte

We now verify the effectiveness of optimal anycasting forwarding technique [45] with our approach using ns2. For simulation we deploy 100 nodes in 1000*1000 m^2 area using a random uniform distribution. The sensor nodes calculate the periodic wake-up rate and optimal anycasting forwarding policy during the configuration phase, for given delay constraint Δ [45].

We set data-packet size as 8 Byte and control packet size as 3 Byte [85]. We use Mica 2 motes parameters used in [57],[85]. We set the receive and idle power to $13.0\,mW$ [85]. Other parameters used in simulation are given in Table 4.2. We define the lifetime of the network as the lifetime of the first node that depletes its energy completely.

In order to show effectiveness, we compare the percentage of packets follows delay constraint in optimal s/w scheduling [45] and in our strategy. Moreover, we also compare the lifetime of network, control packets overhead, and packet delivery ratio, in optimal s/w scheduling [45] and in our strategy.

Packets follow delay constraint: We show the percentage of packets violate delay constraint $(\Delta + \xi)$, for different data generation interval in Fig. 4.6, with the worst clock-skew $100\,ppm$. Whereas, in Fig. 4.7 we set wake-up interval $10\,s$ and compare the percentage of packets violate delay constraint. The percentage of packets do

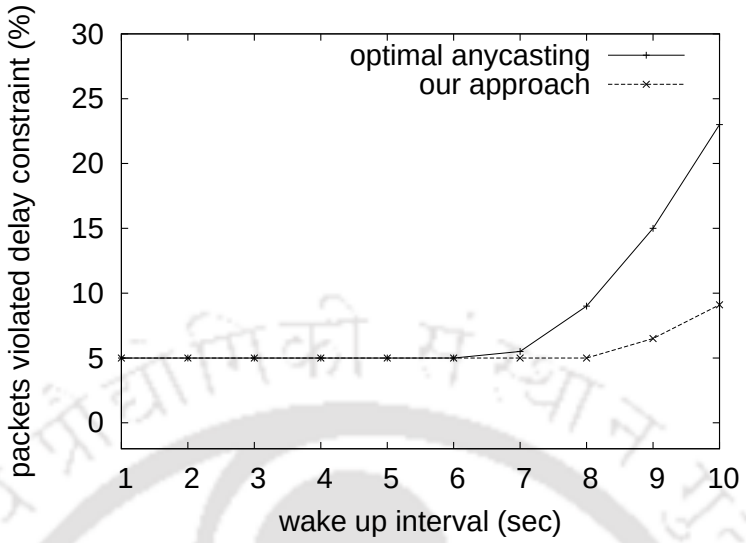


Figure 4.6: Monitoring e2e delay while varying wake-up interval with fixed $\xi = 1ms$

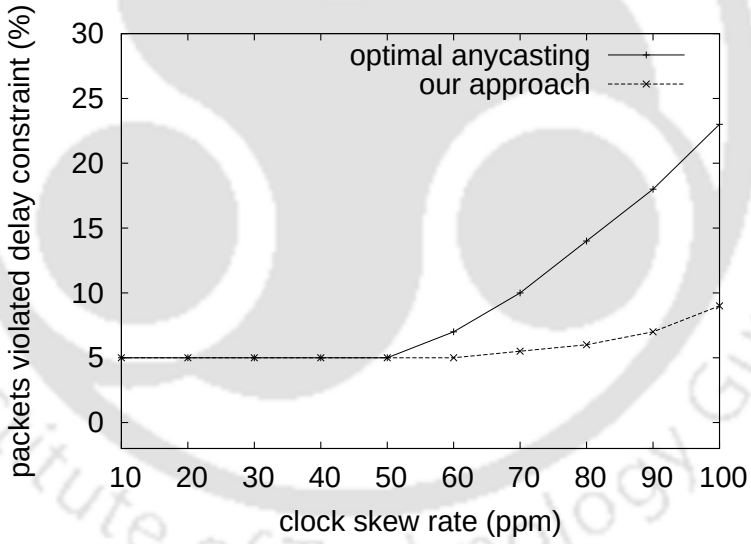


Figure 4.7: Monitoring e2e delay while varying clock-skew with fixed $\xi = 1ms$

not follow delay constraint $\Delta + \xi$ increases, as we increase wake up interval or clock skew in conventional approach, after a certain interval. Whereas using our approach, the percentage of packets violate delay constraint is less. This is because, when ξ is fixed, increasing wake-up interval or clock-skew, increases expected one-

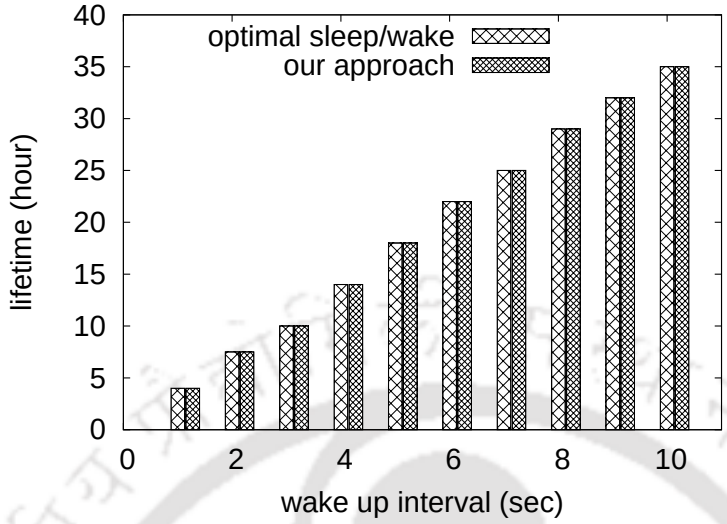


Figure 4.8: Monitoring lifetime while varying wake-up interval with fixed $\xi = 1ms$

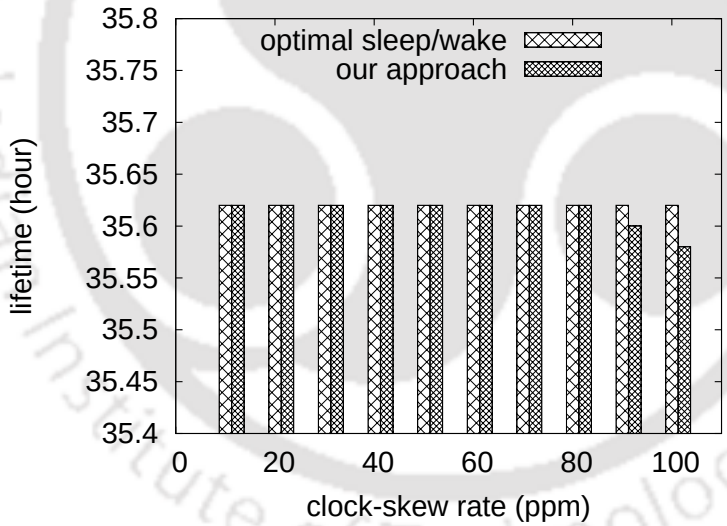


Figure 4.9: Monitoring lifetime while varying clock-skew with fixed $\xi = 1ms$

hop delay increase as a result of clock-skew, which in turn increases the percentage of packets violated delay constraint $\Delta + \xi$.

Lifetime of network: It can be observed from Fig. 4.8 that when the wake-up interval increases the overall lifetime in our approach as well as conventional ap-

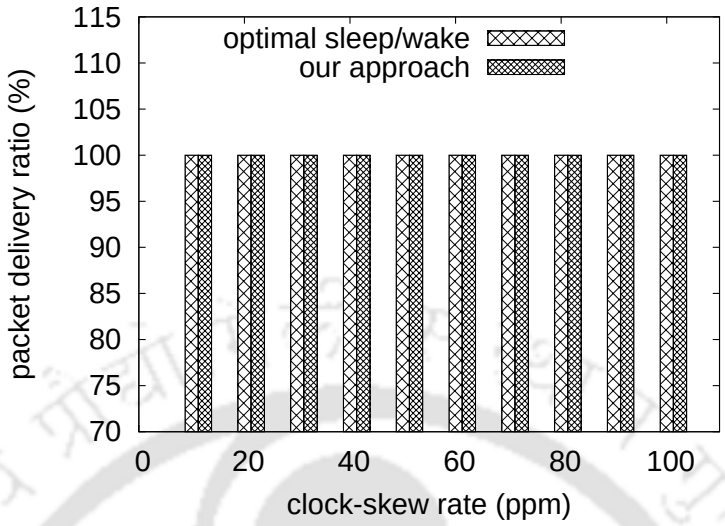


Figure 4.10: Monitoring percentage of packet delivery ratio

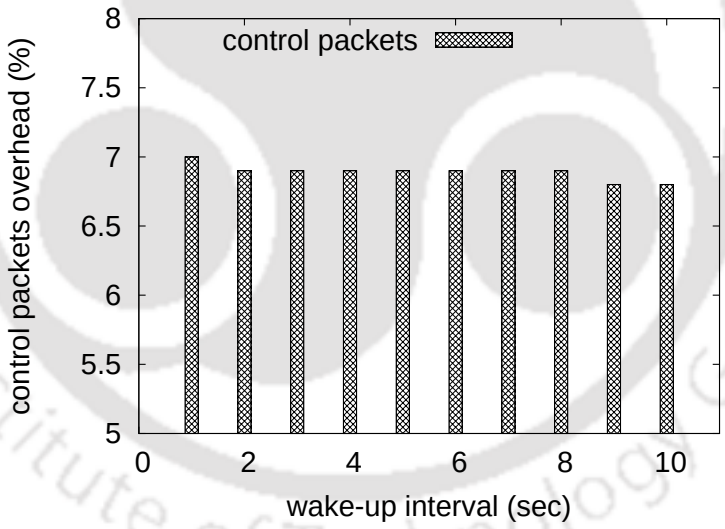


Figure 4.11: Monitoring control packets overhead

proach increases as well. This is because increasing wake-up interval, increases the lifetime of every node, which increases the overall network lifetime. Moreover, when wake-up interval is set $10\,s$ across the network, the lifetime decreases very marginally in our approach when clock-skew increases(4.9). Although it is not

clearly visible in the graph, with 100 ppm clock skew rate and 10 seconds wake-up interval, the lifetime decreases by approximately 0.1%. This is because, in our approach we increase the wake-up rate to satisfy the delay constraint $\Delta + \xi$, which decreases the overall lifetime of the network.

Packet delivery ratio: Fig.4.10 shows percentage of data packets delivered when an event is detected, while varying clock-skew with wake-up interval 10s. It can be noted that, the percentage of packet delivery ratio in both our approach and conventional approach remains same ($\approx 100\%$ when rare events occur).

Control packets overhead: Amount of control packets exchanged for our approach depends only on the number of nodes and network topology and forwarding policy. Wake-up interval, clock-skew does not affect the number of control packets. In Fig.4.11, we show that the percentage of control packets in our approach remains almost same when wake up interval increases.

4.4 Conclusion

In this work, we provided a solution to constrain the additional delay incurs as a result of the clock skew within given bound ξ. Our approach decreases this additional delay while maximizing the lifetime. In the following chapter we are interested in improving QoS for periodic data-gathering. More precisely, we look into the problem of maximizing the lifetime in periodic data-gathering in the presence of clock-skew.

5

Maximizing lifetime in Periodic Data-gathering in Presence of Clock-skew

5.1 Introduction

In previous works, we used stochastic approaches to QoS in event-driven data-gathering. Whereas, in this work we are interested in periodic data-gathering. In applications like environmental monitoring [84], fire detection [94], battlefield surveillance [94], and many more, sensor nodes generate periodic data which are gathered at the control unit, for overall monitoring of the FoI in a WSN. One key QoS requirement in these type of applications is the lifetime of the overall network. In periodic data-gathering, nodes exchange data packets at periodic intervals, and

every sender-receiver pair associates slots for communication. In order to save energy, the sender-receiver pair wakes up at the same time to exchange data packets and remains asleep otherwise, known as the synchronous s/w scheduling technique. Generally in synchronous s/w scheduling, nodes exchange control messages to negotiate slots for communication at periodic intervals [74, 67, 101, 42]. Further energy can be conserved by forming hierarchical structures like a tree based or cluster based during data-forwarding [86].

Exact clock synchronization is required for synchronous s/w scheduling[1], which is very difficult to achieve because of the imperfect crystal oscillator [85]. As a result of inaccuracy and non-determinism present in time synchronization protocol, nodes wake up earlier than the scheduled wake up time, to circumvent any message loss, known as guard time.

As a result of non-determinism present in synchronization, guard time increases as time passes which in turn increases power consumption. Especially in the dense sensor network, when multiple nodes forward data to a single forwarding node (like in cluster based strategies), the forwarding node needs to apply guard time for each sender. Hence, additional energy consumption for guard time increases rapidly with the increasing number of senders associated with the forwarding node. In optimal s/w scheduling [85, 86], the receiver keeps its transceiver on during guard-time until it receives a data-packet, which in turn drains out energy quickly for longer guard-time. Though it is mentioned in [19] that the actual wake-up time follows normal distribution, but it is not considered while calculating the wake-up interval between consecutive wakes up within the guard-time. In practice, the distribution of actual wake up time has implication in the wake-up pattern within the guard-

[1]This is because a sender-receiver pair needs to wake-up at the same time

time and energy consumption can be further minimized by considering this phenomenon. These limitations in the literature motivate us to determine an energy efficient s/w scheduling technique to conserve additional energy in guard time. In this chapter, we propose *Multi-Beacon Guard* [MBG] method, where the sensor nodes wake-up multiple times within the guard-time if the length of the guard-time is more than a threshold, otherwise follow the simple guard-time approach to conserve energy.

5.2 Multi-Beacon Guard

In existing approaches, the receiver keeps its transceiver on during guard-time, and waits for the sender to wake-up. The energy consumption is directly proportional to waiting time. In order to reduce the energy consumption we propose that the receiver wakes up multiple times within the guard-time. In MBG, if the length of guard-time is more than the optimal threshold, multi-beacon approach is followed, else simple guard-time is followed. In *multi-beacon* approach, guard-time is divided into unequal intervals. At the beginning of each interval, the receiver wakes up, sends a beacon message, and waits for an average round trip time (RTT) for an acknowledgment from the sender. If the sender is awake and receives a beacon message, it starts data transmission. If the data packet is received, acknowledgment is sent to confirm. In the contrary, if the sender is not awake, the receiver goes to sleep state after RTT and wakes up at the beginning of the next interval and follows the same procedure. The receiver reduces the total waiting time by going to sleep mode after each unsuccessful data transfer attempt in multi-beacon approach. Once the sender wakes up at the scheduled wake-up time, its transceiver remains active

till it sends the message successfully. The total energy consumption is minimized by reducing total waiting time of the sender and the receiver collectively during a packet transmission.

5.2.1 Wake-up Pattern of Receiver

If τ_p and τ_p' respectively denotes the scheduled and actual arrival time of message p, then as a result of non-determinism present in the system and measurement, τ_p' follows normal distribution with standard deviation σ_p [85]. In order to successfully receive the message the receiver is awake during the interval $[\tau_p - T_g, \tau_p + T_g]$, where $[-T_g, T_g]$ (or $2T_g$) denotes the duration of guard-time. Let X be the event that the sender wakes up and $P(X = x)$ denotes the probability of the sender wakes up at time x. In order to make $\int_{-\infty}^{-T_g} P(X = x)dx + \int_{T_g}^{\infty} P(X = x)dx \simeq 0$, we choose $\delta > 0$ to be very small such that $P\left(|\tau_p - \tau_p{}'| \geq \sqrt{\frac{1}{\delta}}\sigma_p\right) \leq \delta$ and $T_g \geq \frac{\sqrt{\frac{1}{\delta}}\sigma_p}{2}$, using Chebyshev's inequality [71]. Note that for simplicity we use Chebyshev's inequality, one can use Chernoff's inequality for tighter bound [24]. Under this truncated definition $P(X = x) \sim N(\tau_p, \sigma_p^2)$, where $N(\tau_p, \sigma_p^2)$ denotes the normal distribution with mean τ_p and standard deviation σ_p, $\tau_p - T_g \leq x \leq \tau_p + T_g$ else 0. For simplicity we assume $\tau_p = 0$. Since the sender's wake up time follows normal distribution, the receiver needs to wake up more frequently at the mean of the distribution to minimize the expected energy consumption of the sender and the receiver. Hence, the sender must have equal expected waiting time between the two consecutive wake up of the receiver. In other words, the area under the normal curve between the two consecutive wake up of the receiver is equal (see Fig. 5.1).

Assuming the area under the probability distribution function (PDF) is unity and

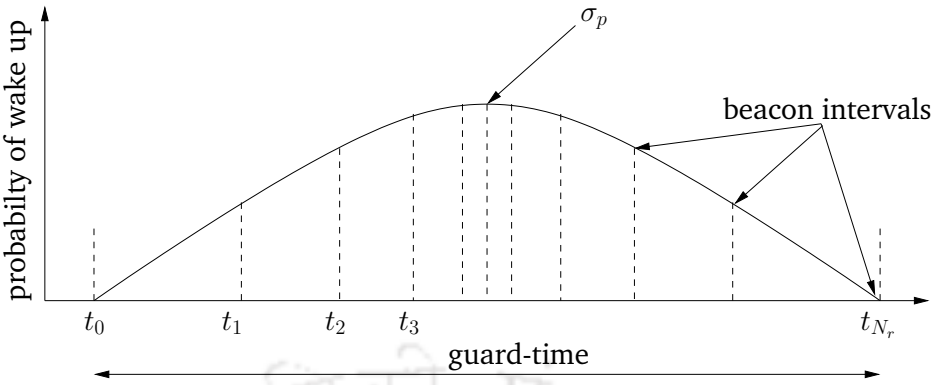


Figure 5.1: Wake-up pattern of receiver within guard-time

t_1,t_2,t_3,...,t_{N_r} denote the exact wake up times of the receiver then the area under PDF between t_k and t_{k+1} is $\frac{1}{N_r}$ for $1 \leq k \leq N_r - 1$. That is,

$$\int_{-T_g}^{t_1} P(X = x)dx = \int_{t_k}^{t_{k+1}} P(X = x)dx = \frac{1}{N_r}, \tag{5.1}$$

for $1 \leq k \leq (N_r - 1)$. Hence, t_i (as shown in Fig. 5.1) can be found using standard normal distribution for $1 \leq i \leq N_r$, as

$$t_i = \sum_{n=0}^{\infty} (2\pi)^{\frac{2n+2}{2}} \frac{C_{2n+1}}{(2n+1)!} \left(i\frac{1}{N_r\sigma} - \frac{1}{2} \right)^{2n+1}, \tag{5.2}$$

where $C_{n+1} = \sum_{j=0}^{n-1} \binom{n}{j+1} C_j C_{n-j}$, and $C_1 = 1$ [27].

The overall procedure is explained in Fig. 5.2. At the beginning of the beacon interval, the receiver switches on its transceiver and sends the beacon messages such that beacons are sent more frequently close to the scheduled wake-up time of the sender. As a result of clock-skew, sender may wake-up sometime later of scheduled wake-up time. If the sender receives the beacon then data-transmission begins.

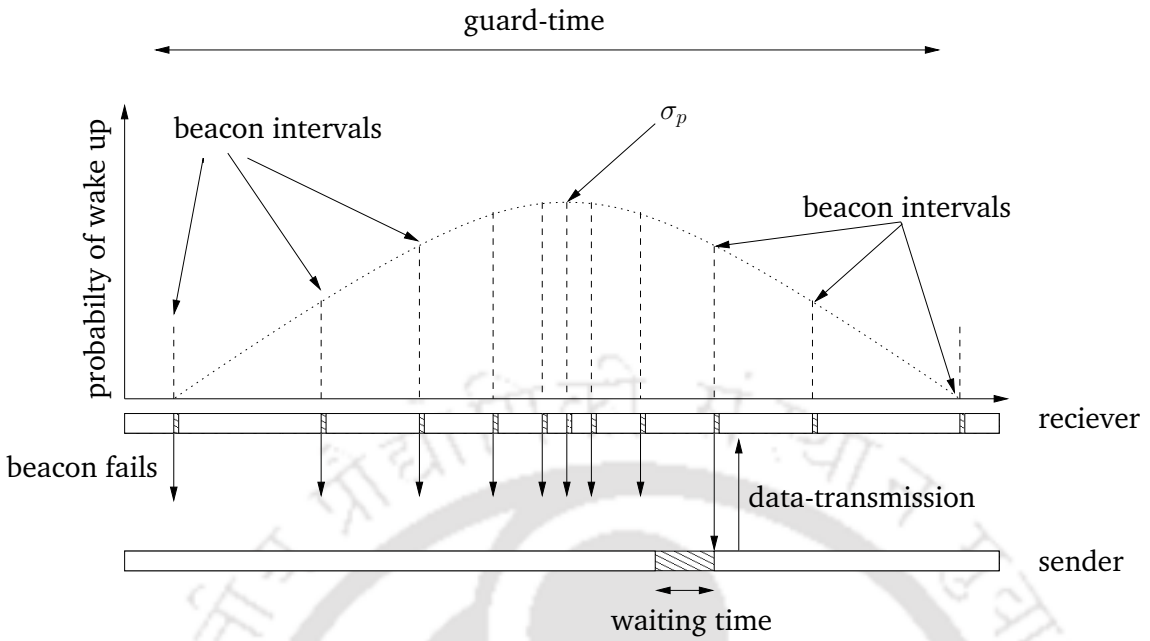


Figure 5.2: Overall procedure of MBG

5.2.2 Minimizing Transmission Energy

Using the wake-up pattern obtained in the previous subsection, in this subsection we show how it reduces energy consumption during a packet transmission. In order to show the effectiveness in energy saving, we first derive expression for expected energy consumption, using the expected number-of-times the receiver wakes up and the expected waiting time of the sender, which are respectively derived in the following lemmas.

Lemma 5. *Assume the actual wake-up time of sender follows normal distribution and the guard-time is divided into N_r intervals. The expected number of time the receiver wakes up for a successful packet transmission is $\frac{N_r+1}{2}$.*

Proof. The receiver wakes up, sends the beacon message, and waits for RTT to receive data from the sender, at times $t_i, 1 \leq i \leq N_r$. The receiver has to wake up $i+1$ times if the sender wakes up in the interval $(t_i, t_{i+1}]$, assuming propagation

delay is negligible. The expected number-of-times the receiver wakes up is,

$$\begin{aligned} & P(X = x : x \leq t_1) \times 1 \\ & + \sum_{i=1}^{N_r - 1} P(X = x : t_i > x \leq t_{i+1}) \times (i+1) \\ & = \frac{N_r + 1}{2}. \end{aligned}$$

□

Lemma 6. *Assume the actual time of the sender follows the normal distribution within the interval* $[\tau_p - T_g, \tau_p + T_g]$*, where* τ_p*,* T_g *respectively denotes the scheduled wake-up time of the sender and the half of the guard-time. The expected waiting time of a sender for a successful packet transmission is* $\frac{T_g}{N_r}$.

Proof. If the area under the normal curve is equally divided into n_r regions, then the expected waiting time of the sender is,

$$\begin{aligned} W_s = & \int_{-T_g}^{t_1} P(X = x)(t_1 - x)dx \\ & + \sum_{i=1}^{N_r - 1} \int_{t_i}^{t_{i+1}} P(X = x)(t_{i+1} - x)dx. \end{aligned}$$

If N_r is odd, we can rewrite the equation as,

$$
\begin{aligned}
W_s = & \int_{-T_g}^{t_1} P(X=x)(t_1-x)dx \\
& + \int_{t_{n_{r-1}}}^{t_{N_r}} P(X=x)(t_1-x)dx \\
& + \sum_{i=1}^{\lfloor \frac{N_r}{2} \rfloor - 1} \int_{t_i}^{t_{i+1}} P(X=x)(t_{i+1}-x)dx \\
& + \sum_{i=1}^{\lfloor \frac{N_r}{2} \rfloor - 1} \int_{t_{N_r-(i+1)}}^{t_{N_r-i}} P(X=x)(t_{N_r-(i)}-x)dx \\
& + \int_{t_{\lfloor \frac{N_r}{2} \rfloor}}^{t_{\lfloor \frac{N_r}{2} \rfloor + 1}} P(X=x)(t_{\lfloor n_{r/2} \rfloor + 1} - x)dx. \qquad (5.3)
\end{aligned}
$$

In order to prove this lemma, we first show that $\int_{t_i}^{t_{i+1}} P(X = x)(t_{i+1} - x)dx + \int_{t_{n_r-(i+1)}}^{t_{N_r-i}} P(X = x)(t_{N_r-i} - x)dx = \frac{1}{N_r} |t_i - t_{i+1}|$.

$$
\begin{aligned}
&\int_{t_i}^{t_{i+1}} P(X=x)(t_{i+1}-x)dx \\
&\quad + \int_{t_{nr-(i+1)}}^{t_{N_r-i}} P(X=x)(t_{N_r-i}-x)dx \\
&= \int_{t_i}^{t_{i+1}} P(X=x)(t_{i+1})dx - \frac{-\sigma}{\sqrt{2\pi}}\left(e^{\frac{-{t_{i+1}}^2}{2\sigma^2}} - e^{\frac{-{t_i}^2}{2\sigma^2}}\right) \\
&\quad + \int_{t_{n_r-(i+1)}}^{t_{n_r-i}} P(X=x)(t_{n_r-i})dx \\
&\quad - \frac{-\sigma}{\sqrt{2\pi}}\left(e^{\frac{-{t_{n_{r-i}}}^2}{2\sigma^2}} - e^{\frac{-{t_{n_r-(i+1)}}^2}{2\sigma^2}}\right), \\
&\quad \text{by substituting } P(X=x) = \tfrac{1}{\sigma_p\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma_p^2}}
\end{aligned}
$$

Since the function is symmetric on both side of mean, and $\|t_i - t_{i+1}\| = \left\|t_{n_r-(i+1)} - t_{n_r-(i)}\right\|$, $|t_{N_r-i}| = |t_i|$ for $i \leq \left\lfloor \frac{N_r}{2} \right\rfloor - 1$. The above equation can be rewritten as,

$$
\begin{aligned}
&t_{i+1}\int_{t_i}^{t_{i+1}} P(X=x)dx + t_{N_r-i}\int_{t_{nr-(i+1)}}^{t_{N_r-i}} P(X=x)dx \\
&= t_{i+1}\frac{1}{N_r} + t_{N_r-i}\frac{1}{N_r}, \text{ by substituting } P(X=x) = \tfrac{1}{\sigma_p\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma_p^2}} \\
&= \frac{1}{N_r}\left|t_{i+1} - t_i\right|. \qquad (5.4)
\end{aligned}
$$

Since the midpoint of the middle interval $[t_{\left\lfloor \frac{N_r}{2} \right\rfloor}, t_{\left\lfloor \frac{N_r}{2} \right\rfloor+1}]$ is mean of PDF, we can write $\int_{t_{\left\lfloor \frac{N_r}{2} \right\rfloor}}^{t_{\left\lfloor \frac{N_r}{2} \right\rfloor+1}} P(X=x)(t_{\left\lfloor \frac{N_r}{2} \right\rfloor+1} - x)dx$ of Eq.5.3 as,

$$\int_{t_{\lfloor \frac{N_r}{2} \rfloor}}^{0} P(X=x)(t_{\lfloor \frac{N_r}{2} \rfloor+1})dx - \int_{t_{\lfloor \frac{N_r}{2} \rfloor}}^{0} P(X=x)(x)dx$$
$$+ \int_{0}^{t_{\lfloor \frac{N_r}{2} \rfloor+1}} P(X=x)(t_{\lfloor \frac{N_r}{2} \rfloor+1})dx$$
$$- \int_{0}^{t_{\lfloor \frac{N_r}{2} \rfloor+1}} P(X=x)(x)dx$$

$$= t_{\lfloor \frac{N_r}{2} \rfloor+1} \int_{t_{\lfloor \frac{N_r}{2} \rfloor}}^{0} P(X=x)dx - \frac{-\sigma}{\sqrt{2\pi}} \left(1 - e^{\frac{-t_{\lfloor \frac{N_r}{2} \rfloor}^2}{2\sigma^2}} \right)$$
$$+ t_{\lfloor \frac{N_r}{2} \rfloor+1} \int_{0}^{t_{\lfloor \frac{N_r}{2} \rfloor+1}} P(X=x)dx$$
$$- \frac{-\sigma}{\sqrt{2\pi}} \left(e^{\frac{-(t_{\lfloor \frac{N_r}{2} \rfloor+1})^2}{2\sigma^2}} - 1 \right),$$

by substituting $P(X=x) = \frac{1}{\sigma_p\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma_p^2}}$

As we know that the function is symmetric in both side of the mean, hence $\left| t_{\lfloor \frac{N_r}{2} \rfloor+1} \right| = \left| t_{\lfloor \frac{N_r}{2} \rfloor} \right|$ and the above expression becomes,

$$\begin{aligned}
&\int_{t_{\lfloor \frac{N_r}{2} \rfloor}}^{t_{\lfloor \frac{N_r}{2} \rfloor + 1}} P(X = x)(t_{\lfloor \frac{N_r}{2} \rfloor + 1} - x)dx \\
&= t_{\lfloor \frac{N_r}{2} \rfloor + 1} \int_{t_{\lfloor \frac{N_r}{2} \rfloor}}^{0} P(X = x)dx \\
&\quad + t_{\lfloor \frac{N_r}{2} \rfloor + 1} \int_{0}^{t_{\lfloor \frac{N_r}{2} \rfloor + 1}} P(X = x)dx, \\
&= t_{\lfloor \frac{N_r}{2} \rfloor + 1} \times \frac{1}{N_r}.
\end{aligned} \tag{5.5}$$

Note that $t_{\lfloor \frac{N_r}{2} \rfloor + 1}$ can be written as $\left| 0 - t_{\lfloor \frac{N_r}{2} \rfloor} \right|$. Similarly, $\int_{-T_g}^{t_1} P(X = x)(t_1 - x)dx + \int_{t_{n_{r-1}}}^{t_{N_r}} P(X = x)(t_1 - x)dx = \frac{1}{N_r} |t_1 - T_g|$. Hence Eq.5.3 can be written as,

$$\begin{aligned}
W_s &= \frac{1}{N_r} |t_1 - T_g| + \frac{1}{N_r} \left(\sum_{i=1}^{\lfloor \frac{N_r}{2} - 1 \rfloor} |t_i - t_{i+1}| \right) \\
&\quad + \frac{1}{N_r} \left| t_{\lfloor \frac{N_r}{2} \rfloor} - 0 \right|, \text{by substituting Eq.5.4,5.5.} \\
&= \frac{T_g}{N_r}.
\end{aligned}$$

Similarly, it can be proved for N_r is even. □

The total expected energy consumption E_{mb} during transmission is the sum of expected energy consumption of the receiver E_{r_mb} and the sender E_{s_mb}. Every time the receiver wakes up, it transmits a beacon message and waits for RTT time to receive a data packet from the sender. After successfully receiving the data packet it sends acknowledgment. Hence, the expected energy consumption of a receiver, using the notations given in Table 5.1, is

$$E_{r\_mb} = (E_{sw} + E_{txbcn} + P_{idle} \times T_{rtt}) \times \frac{n_r + 1}{2} + E_{rxdata} + E_{txack}. \quad (5.6)$$

Whereas, the sender wakes up and waits for a beacon message from the receiver to transmit the data packet. Hence the expected energy consumption of the sender is

$$E_{s\_mb} = E_{sw} + P_{idle} \times \frac{T_g}{n_r} + E_{rxbcn} + E_{txdata} + E_{rxack}. \quad (5.7)$$

Hence, the total expected energy consumption is

$$E_{mb} = (E_{sw} + E_{txbcn} + P_{idle}T_{rtt})\frac{n_r + 1}{2} + E_{rxdata} + E_{txack} + E_{sw} + P_{idle}\frac{T_g}{n_r} + E_{rxbcn} + E_{txdata} + E_{rxack}. \quad (5.8)$$

The following lemma minimizes the expression for expected energy consumption E_{mb}.

Lemma 7. *The expression for the expected energy consumption in the multi-beacon approach given in Eq.5.8 is convex and the optimal value occurs at*

Table 5.1: Notations

Symbol	Description
E_{sw}	Transition energy required from sleep to awake state
E_{txdata}/E_{rxdata}	Energy required to transmit/receive data packet (in μJ)
E_{txack}/E_{rxack}	Energy required to transmit/receive acknowledge packet (in μJ)
E_{txbcn}/E_{rxbcn}	Energy required to transmit/receive beacon packet (in μJ)
T_{rtt}	Average rtt time between any two neighboring node (in Sec)
P_{idle}	Power required for idle listening (in mW)

$$N_r = \sqrt{\frac{2P_{idle}(T_g)}{E_{sw} + E_{txbcn} + P_{idle}T_{rtt}}}, \tag{5.9}$$

where $N_r, P_{idle}, T_g, E_{sw}, E_{txbcn}, T_{rtt}$ *respectively denote the number-of-times the receiver wakes up, power required for idle listening, guard time, transition energy required from sleep to awake state, energy required to transmit a beacon and average RTT.*

Proof. The function derived in Eq. 5.8, is convex because it is second order differentiable, $\frac{\mathrm{d}}{\mathrm{d}N_r}(\frac{\mathrm{d}}{\mathrm{d}N_r}(E_{mb})) = P_{idle} * \frac{1}{{n_r}^3} > 0$ since $P_{idle}, n_r > 0$. We can find the optimal value of E_{mb} occurs at,

$$N_r = \sqrt{\frac{2P_{idle}(T_g)}{E_{sw}+E_{txbcn}+P_{idle}T_{rtt}}}, \tag{5.10}$$

from the equation $\frac{\mathrm{d}}{\mathrm{d}N_r}(E_{mb}(n_r)) = 0$. □

5.2.3 Threshold for multi-beacon

If the guard time is too short, the energy required for the multi-beacon approach for control packets may be more than that of guard-time approach. In this subsection we find the threshold for T_g beyond which the multi-beacon is energy efficient than the guard-time approach. We denote the threshold for the multi-beacon approach by Th_{mb}.

Lemma 8. *The expected energy consumption in guard-time approach is*

$$E_{T_g} = (E_{sw} + E_{txdata} + E_{rxack}) + (E_{sw} + P_{idle} \times T_g + E_{rxdata} + E_{txack}), \quad (5.11)$$

using the notations given in Table 5.1.

Proof. In guard time approach, the receiver is awake during the whole guard-time and the sender sends the data packet as soon as it wakes up. As mentioned earlier wake up times of the sender follows the normal distribution within the guard time. If x denotes the instant when the sender wakes up, then the amount of time the receiver waits is $T_g + x$, where $-T_g \leq x \leq T_g$. Hence, the expected amount of time the receiver waits is

$$
\begin{aligned}
W_{r\_T_g} &= \int_{-T_g}^{T_g} P(X=x)(T_g+x)dx, \\
&= T_g \int_{-T_g}^{T_g} \left(\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}}\right) dx \\
&\quad + \int_{-T_g}^{T_g} \left(\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}}\right) x \; dx, \\
&\qquad \text{by substituting } P(X=x) = \frac{1}{\sigma_p\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma_p^2}} \\
&= T_g.
\end{aligned}
$$

The receiver sends the acknowledgment after receiving the data packet from the sender and switches off the transceiver. Hence, the expected energy consumption of the receiver using the notations given in Table 5.1, is $E_{r\_T_g} = E_{sw} + P_{idle} \times (W_{r-T_g}) + E_{rxdata} + E_{txack}$. The energy consumption of sender is $E_{s\_T_g} = E_{sw} + E_{txdata} + E_{rxack}$. Hence, the total expected energy consumption is

$$
\begin{aligned}
E_{T_g} &= (E_{sw} + E_{txdata} + E_{rxack}) \\
&\quad + (E_{sw} + P_{idle}(W_{r\_T_g}) + E_{rxdata} + E_{txack}).
\end{aligned} \tag{5.12}
$$

□

The threshold for the multi-beacon approach can be found by solving the inequality $E_{T_g} > E_{mb}$. That is,

$$T_g(P_{idle} - \frac{P_{idle}}{N_r}) > E_{rxbcn} + \{E_{sw} + E_{txbcn} + P_{idle}T_{rtt}\}\frac{n_r + 1}{2} - E_{sw}.$$

By substituting Eq.5.10, gives the following inequality

$$T_g > \frac{\left(c_3 + \sqrt{c_2}\right)^2}{2{P_{idle}}^2} \tag{5.13}$$

where $c_2 = c_3^2 + 4P_{idle}(E_{rxbcn} + \frac{1}{2}(c_1) - E_{sw})$, $c_3 = c_1\frac{1}{2}\sqrt{\frac{p_{idle}\times 2}{c_1}} + \sqrt{\frac{P_{idle}c_1}{2}}$, and $c_1 = (E_{sw} + E_{txbcn} + P_{idle}T_{rtt})$.

Lemma 9. *Energy consumption in the multi-beacon approach is less than the guard-time approach when $T_g > Th_{mb}$, where T_g and Th_{mb} respectively denotes half of the guard time and the threshold for the multi beacon approach.*

In order to optimize the energy consumption in MBG, simple guard-time strategy is followed if $T_g \leq Th_{mb}$. Otherwise, multi-beacon approach is followed.

5.2.4 Validation of Analysis

In order to verify the expressions for expected energy consumption and the threshold for the multi-beacon approach, we evaluate corresponding equations numerically, and respectively compare with simulation results. For the simulation results, a Monte Carlo simulation is conducted, which is used to estimate the probability from a large number of experiments. Each experiment consists of several tests. In

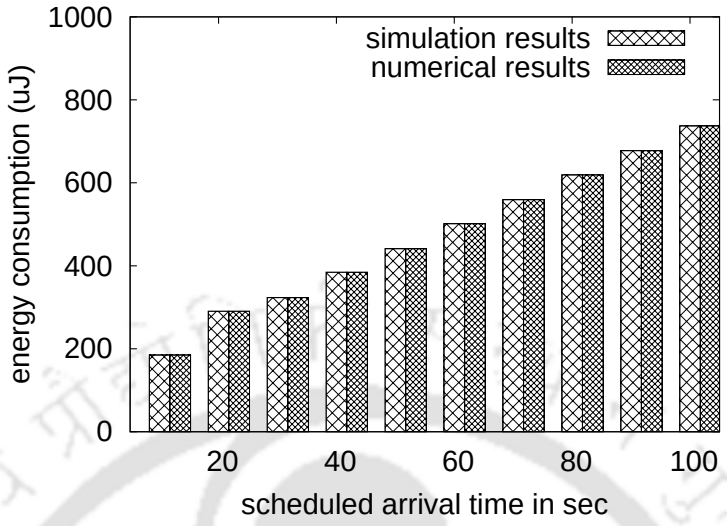


Figure 5.3: Validation of expected energy consumption in MBG

each test a sender's wakes up time is picked from the normal distribution with standard deviation $\sigma_p = 1.00021\sigma_0\sqrt{\frac{1}{N_s} + \frac{1}{N_s}\frac{(\tau_p - \overline{T(j)})^2}{\overline{T^2(j)} - (\overline{T(j)})^2}}$, where N_s denotes number of pairs of time instants exchanged during synchronization between node i and j or $((t_i(k), t_j(k)), k = 1...N_s,)$, and $\overline{T(j)} = \frac{\sum_{k=1}^{N_s} t_j(k)}{N_s}, \overline{T^2(j)} = \frac{\sum_{k=1}^{N_s} t^2{}_j(k)}{N_s}$, and σ_0 denotes standard deviation of synchronization error [85]. We set $N_s = 2$ and $\sigma_0 = 36.5\mu s$ as shown in [85]. We repeat the test for 10000 times and calculate the mean. We repeat the experiment for 10000 times and the results with 95% confidence level are given in Fig. 5.3,5.4. The error bars are not visible in the plot.

Fig. 5.3 shows the average energy consumption[2] during a transmission for different τ_p. For numerical results we evaluate Eq. 5.8 when $T_g > Th_{mb}$, otherwise Eq. 5.12. The energy consumption increases as time lapses from the last synchronization because of the increasing guard-time. Moreover, the simulation and numerical results are matching for different values of τ_p.

[2]The total average energy during a transmission between a sender-receiver pair.

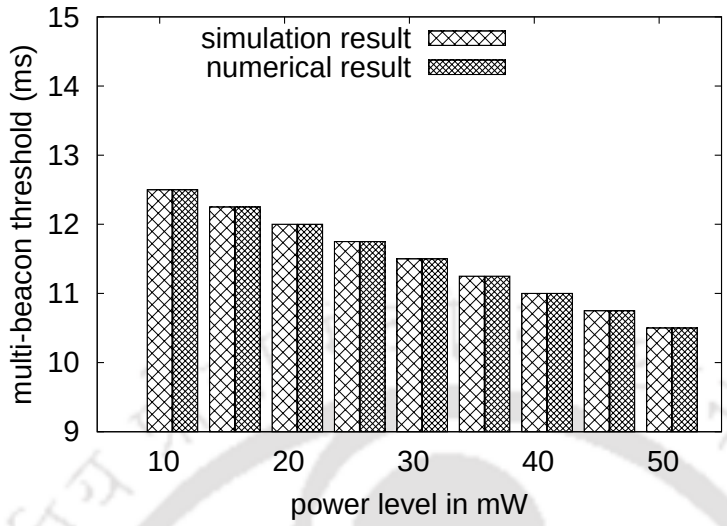


Figure 5.4: Validation of multi-beacon threshold

In order to validate the threshold for the multi-beacon approach through simulation, we first compute the threshold for a fixed power level, by varying T_g[3]. We show the comparison between simulation results and numerical evaluation of Eq. 5.13 in Fig. 5.4 for different power levels. The amount of energy consumed in multi-beacon approach per unit sleep time increases as power level increases. The transceiver of the receiver in multi-beacon approach does not remain on continuously, as compared to the guard-time approach. Hence, the multi-beacon threshold decreases as power level increases, which can be observed in Fig. 5.4. Also note that the simulation and numerical results are matching. Moreover, the simulation results are with 95% confidence level though error bars are not visible in the plot. Energy consumption in MBG and guard-time approach is given in Fig. 5.5 for fixed power level $10\,mW$. One can see clearly that energy conservation increases with T_g, when $T_g > Th_{mb}$.

[3]A fixed idle, transmit, and receive power consumption is called power level.

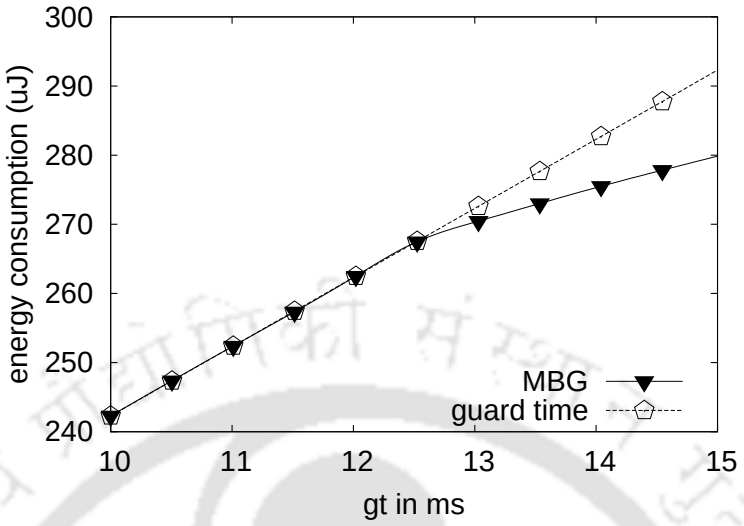


Figure 5.5: Energy consumption in MBG and guard-time approach with fixed power level 10 mW

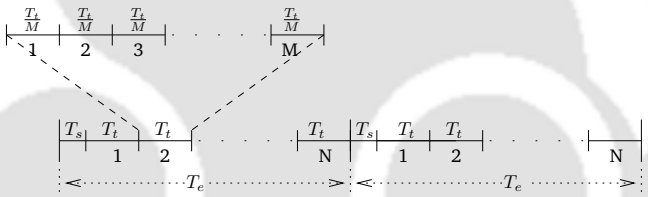


Figure 5.6: Data-generation model

5.3 Expected Energy Conservation Estimation

In previous subsection we derived and validated the expressions for the expected energy consumption in guard-time and MBG using the wake-up pattern obtained in the subsection 5.2.1. Whereas in this subsection, we extend our analysis to show how it save energy for a particular data-generation model. Note that, though we show the effectiveness of our approach in saving energy for a particular data-generation model, our approach is applicable for any synchronized periodic data-gathering protocol.

We assume the data-generation model given in [85, 86, 83, 79, 67], where the time is divided into constant duration epochs (T_e) such that each epoch begins with synchronization interval T_s, followed by transmission interval. Each transmission interval consists of one (or more) sub-transmission interval(s) T_t, such that in each sub-transmission interval each sender sends messages to the receiver, i.e. $T_e = T_s + N_{T_t}T_t, N \geq 1$, where N_{T_t} is the number of sub-transmission interval. Each sub-transmission interval is split into M equal slots, one for each sender (see Fig. 5.6). If sender i is communicating with the receiver in q^{th} sub-transmission interval T_q, then time elapsed from last synchronization is,

$$T_{i,q} = \frac{T_t}{M}(i-1) + T_t(q-1). \tag{5.14}$$

Using the implementation of RBS [30] given in [85], the standard deviation σ_p of actual arrival time of the message p is given by,

$$\sigma_p = \sqrt{\frac{{\sigma_0}^2}{{a_{ij}}^2}\left[\frac{1}{N_s} + \frac{1}{N_s}\frac{\left(\tau_p - \overline{T(j)}\right)^2}{\overline{T^2(j)} - \left(\overline{T(j)}\right)^2}\right]}, \tag{5.15}$$

where N_s denotes the number of pairs of time instants exchanged during synchronization between node i and j or $((t_i(k), t_j(k)), k = 1...N_s,)$, and $\overline{T(j)} = \frac{\sum_{k=1}^{N_s} t_j(k)}{N_s}, \overline{T^2(j)} = \frac{\sum_{k=1}^{N_s} {t^2}_j(k)}{N_s}$, and σ_0 denotes the standard deviation of synchronization error [85].

Note that $\overline{T(j)}$ denotes the average of N_s time instants of the receiver. Hence, $(\tau_p - \overline{T(j)}) \geq T_{i,q}$, that is $(\tau_p - \overline{T(j)}) = T_{i,q} + \epsilon$, where $0 \leq \epsilon \leq T_s$. Moreover, the

standard deviation σ_p' in terms of $T_{i,q}$ can be written as,

$$\sigma_p{}'(T_{i,q}) = \sqrt{\frac{{\sigma_0}^2}{{a_{ij}}^2}\left[\frac{1}{N_s} + \frac{1}{N_s}\frac{(T_{i,q})^2}{\overline{T^2(j)} - \left(\overline{T(j)}\right)^2}\right]} \leq \sigma_p. \tag{5.16}$$

Also note that, when $\sigma_p{}'(T_{i,q})$ is given, T_g can be calculated as shown in the section 5.2.1. The expected energy conservation during a data-packet transmission E_{cdata}, in multi-beacon approach can be given by,

$$\begin{aligned} E_{cdata} &= E_{T_g} - E_{mb}, \\ &= E_{s\_w} - (E_{sw} + E_{txbcn} + P_{idle}T_{rtt})\frac{N_r+1}{2} \\ &\quad + E_{rxbcn} + P_{idle}\left(\frac{\sqrt{\frac{1}{\delta}}\sigma_p}{2} - \frac{\sqrt{\frac{1}{\delta}}\sigma_p}{2N_r}\right), \end{aligned}$$

by substituting $\sigma_p'(T_{i,q})$ in place of σ_p we get

$$\begin{aligned} E_{cdata} \geq E_{s\_w} &- (E_{sw} + E_{txbcn} + P_{idle} \times T_{rtt})\frac{N_r+1}{2} \\ &+ E_{rxbcn} \\ &+ P_{idle}\left(\frac{\sqrt{\frac{1}{\delta}}\sigma_p{}'(T_{i,q})}{2} - \frac{\sqrt{\frac{1}{\delta}}\sigma_p{}'(T_{i,q})}{2N_r}\right). \end{aligned}$$

By substituting Eq. 5.16 in place of $\sigma_p{}'(T_{i,q})$ in the above equation we get,

$$
\begin{aligned}
E_{cdata}(T_{i,q}) = E_{s\_w} - (E_{sw} + E_{txbcn} + P_{idle}T_{rtt})\frac{N_r+1}{2} \\
+ E_{rxbcn} \\
+ \frac{P_{idle}\sqrt{\frac{1}{\delta}}\sqrt{\frac{{\sigma_0}^2}{{a_{ij}}^2}\left[\frac{1}{N_s} + \frac{1}{N_s}\frac{(T_{i,q})^2}{T^2(j) - \left(\overline{T(j)}\right)^2}\right]}}{2} \\
- \frac{P_{idle}\sqrt{\frac{1}{\delta}}\sqrt{\frac{{\sigma_0}^2}{{a_{ij}}^2}\left[\frac{1}{N_s} + \frac{1}{N_s}\frac{(T_{i,q})^2}{T^2(j) - \left(\overline{T(j)}\right)^2}\right]}}{2N_r} \\
\leq E_{cdata}. \qquad (5.17)
\end{aligned}
$$

When time elapsed from last synchronization increases, the standard deviation σ_p of actual message arrival time increases, which in turn increases the length of guard-time. When T_g reaches to Th_{mb}, multi-beacon approach starts conserving energy. In other words, for a given T_t and M, if there exists $(T_{n,m})$, where $(T_{n,m})$ denotes n^{th} sender's time interval in m^{th} sub-transmission interval in an epoch, such that $T_g > Th_{mb}$ and $T_g \leq Th_{mb}$ for $(T_{n-1,m})$ if $n > 1$, else $T_g \leq Th_{mb}$ for $(T_{M,m-1})$ if $n = 1$, then expected energy conservation for node i in multi-beacon approach in an epoch is at least,

$$
\begin{aligned}
E_{cepoch}(i) = & \sum_{s=n}^{M} E_{cdata}(T_{s,m}) \\
& + \sum_{l=m+1}^{N_{T_t}} \sum_{k=1}^{M} E_{cdata}(T_{k,l}), \qquad \text{if } n > 1, \\
= & \sum_{l=m}^{N_{T_t}} \sum_{k=1}^{M} E_{cdata}(T_{k,l}), \qquad \text{if } n = 1.
\end{aligned}
\tag{5.18}
$$

5.4 Maximizing Network Lifetime

Hierarchical clustering strategies are known to be energy efficient in periodic data-gathering. Hence, in this section we analyze the effect of MBG to maximize the lifetime of a hierarchical clustering network. We consider the base-station is located at level 0 (highest level), and each cluster contains single cluster head (CH) and multiple cluster members [38, 26]. Neighboring clusters use orthogonal frequency channel to avoid collisions. We use the data-aggregation model described in [86]. If a CH j at level k contains m cluster members (labeled as $1...m$) then the aggregated message is

$$
\chi_j^k = r^k \left(\left(\sum_{i=1}^{m} \chi_i^{k+1} \right) + ml_j \right) + c^k, \tag{5.19}
$$

where ml_j denotes the message length of the node j, $r^k \leq 1$, and c^k corresponds to the overhead of aggregation.

In order to maximize the lifetime of the network, we need to measure the lifetime of a node. The lifetime of node i can be given as, $L(i) = \frac{Q_i - E_{con}(i)}{E_{epoch}(i)} epoch_{dur}$, where $Q_i, E_{con}(i), E_{epoch}(i)$, and $epoch_{dur}$ respectively denotes the initial energy, energy required during configuration phase, and energy consumption per epoch, and duration of epoch, for node i.

Each epoch consists of several sub-transmission intervals. In each sub-transmission interval, a sensor node receives packets from lower level nodes and sends aggregated packets to the higher level node. Energy consumption per epoch is the sum of energy consumption during synchronization, data-reception, and transmission. If E_{rx} and E_{tx} respectively denotes receiving and transmitting energy for a unit length packet, then total energy consumption during an epoch of a node i is given as

$$E_{epoch}(i) = E_{sync}(i) + \sum_{l=1}^{N_{T_t}} \left(\sum_{j=1}^{m} \left(\chi_j^{k+1} E_{rx} + E_{txack} \right) + \chi_i^k E_{tx} + E_{rxack} \right). \quad (5.20)$$

5.20 where $E_{sync}(i)$ denotes the amount of energy consumption for synchronization and N_{T_t} denotes the number of sub-transmission intervals. As the time lapses from synchronization increases, guard-time gt increases. Hence, CH awake-time before scheduled transmission also increases. Therefore, we need to consider additional energy consumption during this gt, in-order to find the total energy consumption. If node i is sending data in l^{th} sub-transmission interval, then the guard-time $[-gt(T(i,l)), gt(T(i,l))]$ can be calculated using the Chebyshev inequality discussed in section 5.2.1. Note that, expressions for expected energy consumption in multi-beacon and guard-time, for a sender-receiver pair is given in Eq. 5.8 and 5.12,

respectively.

Hence, expected energy consumption in MBG, excluding energy consumption for packet transmission and reception, of a receiver and sender during $[-gt(T(i,l)), gt(T(i,l))]$ is given by Eq. 5.21 and 5.22, respectively.

$$\mathrm{E}_{rx}^{MBG,\mathrm{gt(T(i,l))}} = \begin{cases} (E_{sw} + E_{txbcn} + P_{idle}T_{rtt})\frac{N_r+1}{2}, & \text{if } \mathrm{gt(T(i,l))} > \mathrm{TH_{mb}} \\ \mathrm{E_{sw}}+\mathrm{P_{idle}gt(T(i,l))}, & \text{otherwise.} \end{cases} \tag{5.21}$$

$$\mathrm{E}_{tx}^{MBG,\mathrm{gt(T(i,l))}} = \begin{cases} E_{sw} + P_{idle}\frac{gt(T(i,l))}{n_r} + E_{rxbcn}, & if\ \mathrm{gt(T(i,l))} > \mathrm{TH_{mb}} \\ \mathrm{E_{sw}}, & \text{otherwise.} \end{cases} \tag{5.22}$$

By substituting Eq. 5.21 and 5.22 in Eq. 5.20 we get total energy consumption for a node i at level k in a hierarchical clustering network as

$$\mathrm{E}_{epoch}^{MBG}(i) = E_{sync}(i)$$
$$\sum_{l=1}^{N_{T_t}} \left(\sum_{j=1}^{m_i} \left(\chi_j^{k+1} E_{rx} + \mathrm{E}_{rx}^{MBG,\mathrm{gt(T(j,l))}} + E_{txack} \right) + \chi_i^k E_{tx} + \mathrm{E}_{tx}^{MBG,\mathrm{gt(T(i,l))}} + E_{rxack} \right) \tag{5.23}$$

Similarly, total energy consumption for a node i at level k in a hierarchical clustering network, with guard-time approach, is given by

$$\mathrm{E}_{epoch}^{GT}(i) = E_{sync}(i) + \sum_{l=1}^{N_{T_t}} \left(\sum_{j=1}^{m_i} \left(\chi_j^{k+1} E_{rx} + \mathrm{E}_{rx}^{GT,\mathrm{gt(T(j,l))}} + E_{txack} \right) + \chi_i^k E_{tx} + \mathrm{E}_{tx}^{GT,\mathrm{gt(T(i,l))}} + E_{rxack} \right) \tag{5.24}$$

5.4.1 Dynamic Clustering Network

In dynamic clustering, the CHs change over time to increase the overall network lifetime. For simplicity we assume, the nodes use the fixed power-transmission model and the cluster members and cluster-heads communicate directly to the CH and base-station, respectively [38]. Hence, the maximum value of k is 2 because its a three level hierarchy. The time is divided into several rounds and in each round a cluster-head is selected randomly to collect data and forward to the base-station, such that in each r rounds, each node becomes CH once and remains non-CH for $r-1$ rounds. Let $e_{con}^{CH}(i)$ and $e_{con}^{NCH}(i)$ respectively denotes the amount of energy consumption during configuration for node i being the CH and non-CH.

Hence, energy consumption during r rounds is given in Eq. 5.27, where $E_{epoch,CH}^{MBG,dy}(i)$ and $E_{epoch,NCH}^{MBG,dy}(i)$ denotes energy consumption of node i as a CH and non-CH, as given in Eq. 5.25 and Eq. 5.26, respectively (both obtained from Eq. 5.23), and r_e denotes number of epochs in each round.

The energy consumption of a CH node and a cluster node are respectively given by the following equations.

$$\mathrm{E}^{MBG,dy}_{epoch,CH}(i) = E^{dy}_{sync,CH} + \sum_{l=1}^{N_{T_t}} \left(\sum_{j=1}^{m_i} \left(\chi^2_j E_{rx} + \mathrm{E}^{MBG,\mathrm{gt(T(j,l))}}_{rx} + E_{txack} \right) + \chi^1_i E_{tx} + \mathrm{E}^{MBG,\mathrm{gt(T(i,l))}}_{tx} + E_{rxack} \right) \tag{5.25}$$

$$\mathrm{E}^{MBG,dy}_{epoch,NCH}(i) = E^{dy}_{sync,NCH} + \sum_{l=1}^{N_{T_t}} \left(\chi^2_j E_{rx} + \mathrm{E}^{MBG,\mathrm{gt(T(j,l))}}_{rx} + E_{txack} \right) + \chi^1_i E_{tx} + \mathrm{E}^{MBG,\mathrm{gt(T(i,l))}}_{tx} + E_{rxack}. \tag{5.26}$$

Hence, energy consumption during r rounds is given by

$$E^{MBG}_{dy}(i) = \left(r_e E^{MBG,dy}_{epoch,CH}(i) + e^{CH}_{con}(i) \right) + (r-1) \left(r_e E^{MBG,dy}_{epoch,NCH}(i) + e^{NCH}_{con}(i) \right) \tag{5.27}$$

The expected lifetime of node i is given as

$$L^{MBG}_{dy}(i) = \frac{Q_i}{E^{MBG}_{dy}(i)} r_{dur}, \tag{5.28}$$

where r_{dur} denotes the duration of r rounds. Hence, the expected lifetime of the network is given by

$$L_{dynamic}^{MBG} = \min\left\{L_{dy}^{MBG}(i)\right\}, \quad \forall i. \tag{5.29}$$

Similarly, we can calculate the expected lifetime of the network for guard-time (L_{dy}^{GT}). Hence, the expected lifetime increased using MBG is denoted as

$$L^{*} = L_{dy}^{MBG} - L_{dy}^{GT}. \tag{5.30}$$

5.5 Simulation

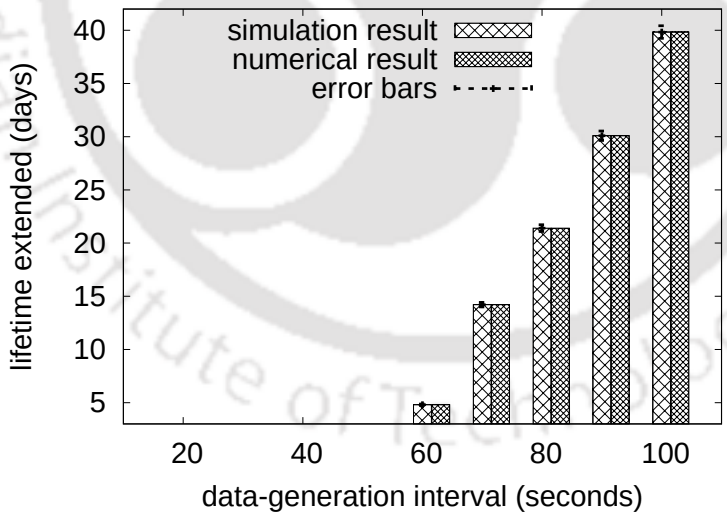


Figure 5.7: Validating expression for lifetime extended for $n = 2$

We deploy 100 nodes in $500 \times 500\ m^2$ area using a random uniform distribution and simulated MBG using the network ns2. We use Mica 2 motes [4] parameters

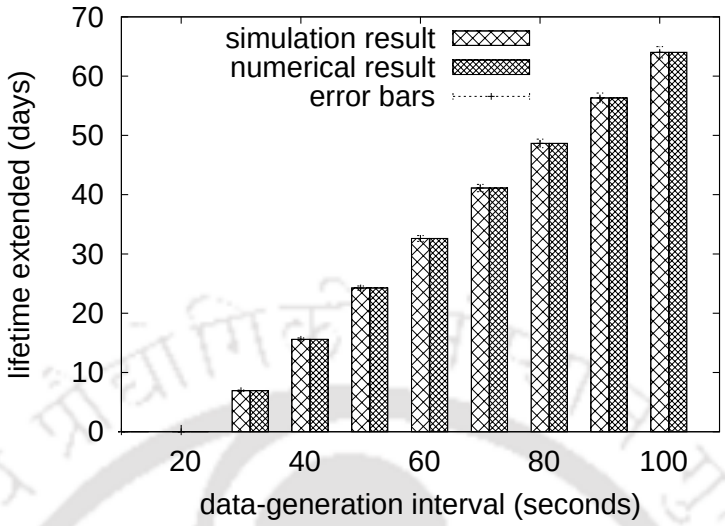


Figure 5.8: Validating expression for lifetime extended for $n = 5$

along with the power requirements shown in [57],[85]. Other parameters used in simulation are given in Table 5.2. We define the lifetime of the network as the lifetime of the first node that depletes its energy completely.

Table 5.2: Simulation Parameters

Name	Value
Area under simulation	$500 \times 500\ m^2$
Communication range	$250\,m$
Radio transceiver type	$CC\,1000$
Data rate	19.2 kbps
Initial energy	1000 J
Receiving/Idle Power	13.0 mW
Transmission Power	19.5 mW
Energy required from sleep to awake	22.0 μJ
Data packet length	8 Byte
Beacon/Ack packet length	3 Byte
Synchronization protocol	RBS
Wireless media	802.11

Nodes use LEACH [38] for clustering formation. We set the number of clusters in each round as 4% of the total nodes. The cluster-head collects data-packets from cluster-members, aggregate the packets ($r = 0.5$) and forwards the resultant

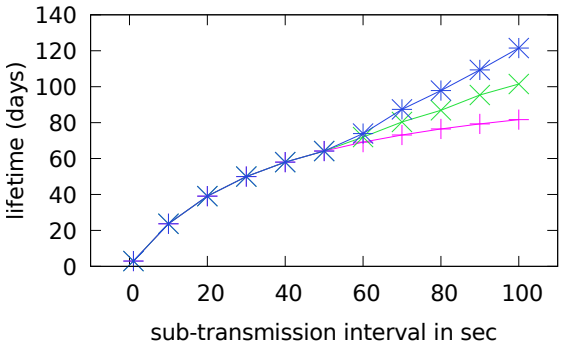


Figure 5.9: Lifetime ($N_{T_t} = 2$)

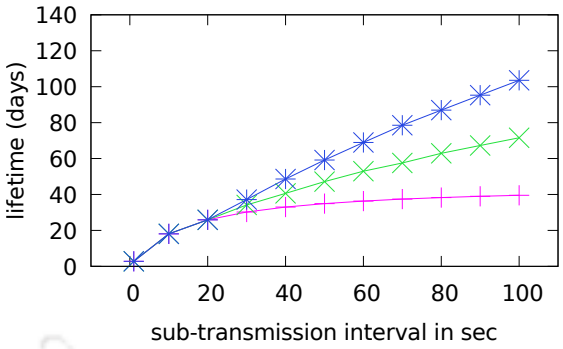


Figure 5.10: Lifetime ($N_{T_t} = 5$)

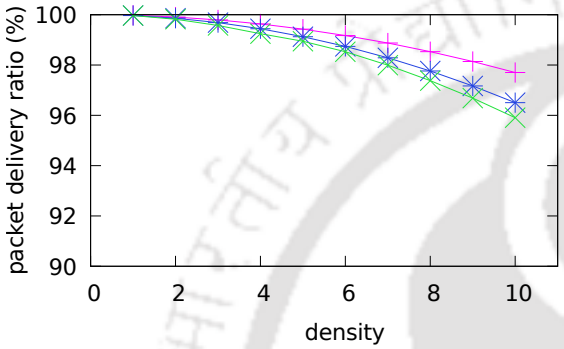


Figure 5.11: Delivery ratio ($N_{T_t} = 2$)

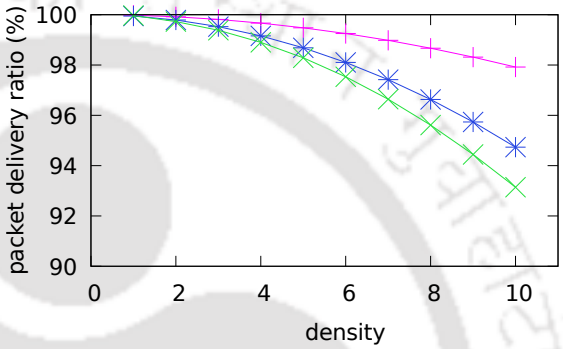


Figure 5.12: Delivery ratio ($N_{T_t} = 5$)

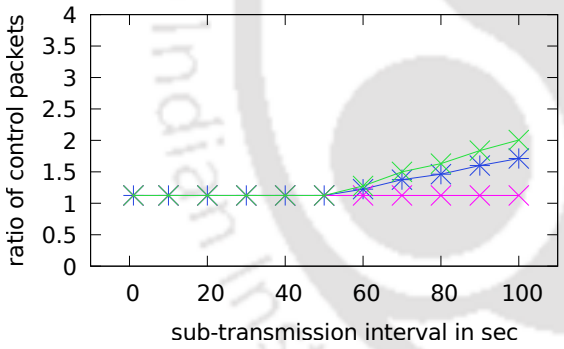


Figure 5.13: Overhead ($N_{T_t} = 2$)

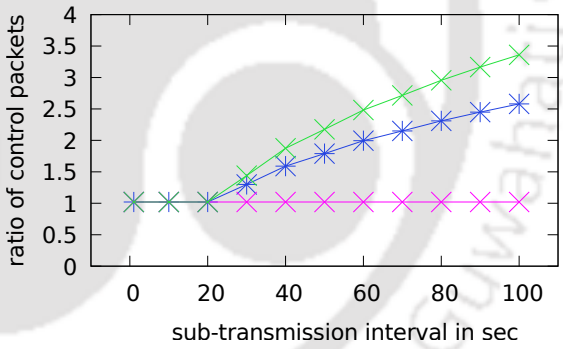


Figure 5.14: Overhead ($N_{T_t} = 5$)

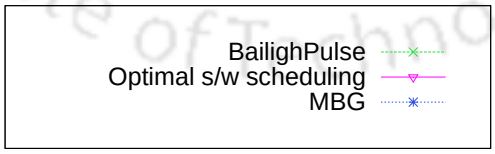


Figure 5.15: legend

packets to the base-station. For simplicity, we assume that sensor nodes use fixed-level power transmission model. In order to avoid collision, time division multiple

access (TDMA) and code division multiple access (CDMA) based approaches are respectively used for intra and inter cluster communication. For synchronization we use RBS [30] using the implementation described in [85].

5.5.1 Validation of Expected Increase in Lifetime

In order to validate the expression for lifetime extension using MBG, we evaluate Eq. 5.30 numerically, and compare with simulation results. For the simulation results, a Monte Carlo simulation is conducted, which is used to estimate the probability from a large number of experiments. In each experiment, we randomly deploy the sensor nodes and calculate the lifetime of the network, for MBG and optimal s/w scheduling [85] which uses guard-time approach. We repeat the experiment for 100 times and calculate the average increase in the lifetime. Each experiment consists of several tests. In each test, a sender wake-up time is selected using similar methods shown in Section 5.2.4. We conduct such tests 100 times and calculate the average lifetime. The results with 95% confidence levels along with error bars, for $N_{T_t} = 2$ and $N_{T_t} = 5$, are shown in Fig. 5.7 and 5.8, respectively.

It can be observed that when the data-generation interval increases, the expected lifetime extension increases as well. This is because, when data-generatiom interval increases, length of guard-time increases as well, which in turn increases energy consumption and decreases the lifetime in conventional strategy, whereas MBG saves energy and increases lifetime. Similarly, when N_{T_t} increases, the lifetime extension also increases because increasing N_{T_t} increases guard-time. If guard-time is more, lifetime extension using MBG over optimal s/w scheduling increases.

5.5.2 Comparison with different approaches

In this sub-section we compare MBG with BailighPulse [19] and optimal s/w scheduling [85]. Optimal s/w scheduling uses the simple guard-time approach whereas BailighPulse [19] uses equal intervals multibeacon approach.

Lifetime: In Fig. 5.9 and 5.10, we show the effectiveness in increasing network lifetime of MBG over optimal s/w scheduling[85] and BailighPulse [19], by varying data-generation interval. If data-generation interval increases, guard-time increases which decreases the overall lifetime. Hence, MBG increases the lifetime rapidly if data-generation interval increases over optimal s/w scheduling[85] (which uses simple guard-time technique) as shown in Fig. 5.9-5.10. Moreover, compared to BailighPulse [19] which uses uniform beacon interval, MBG increases the lifetime because of less number of wake-up intervals. Also note that, if the number of data-generation interval (N_{T_t}) increases, the guard-time increases as well. Hence, MBG is effective in extending the network lifetime for higher values of N_{T_t} (see Fig. 5.9,5.10).

Packet delivery ratio: Fig. 5.11, Fig.5.12 show that the percentage of the packet delivery ratio decreases marginally in MBG and BailighPulse [19] as the density increases. Increasing density increases the amount of beacons exchanged during an epoch (both in MBG and BailighPulse [19]), which in turn increases the probability of collision with data-packets. Moreover, as the number of beacon packets increases more in BailighPulse [19] than MBG, as guard-time increases, BailighPulse [19] suffers from higher packet loss. This phenomenon is more visible for the higher value of N_{T_t} because as N_{T_t} increases the number of control packets exchanged also increases as well.

Delay: There is no deviation in the e2e delay for MBG, BailighPulse [19] and optimal s/w scheduling, since these protocols follow TDMA based data-packet transmission.

Ratio of control and data packets: In MBG and BailighPulse [19], as the guard-time increases amount of beacons exchanged increases significantly because of multi-beacon approach. In Fig. 5.13,5.13 we compare the number of control packets generated for a data-packet, to measure the control packets overhead, while varying data-generation interval in MBG, BailighPulse [19] and optimal s/w scheduling. Because of multi-beacon approach, the ratio of control and data-packets increases as data-generation interval increases, in both MBG and BailighPulse [19].

5.6 Conclusion

In this work, we proposed the *Multi-Beacon Guard* [MBG] method where the sensor nodes follow the multi-beacon approach (with unequal intervals) if the length of guard-time is more, otherwise follow the simple guard-time approach to conserve energy. We studied the problem of energy efficient transmission between a sender-receiver pair and extended this analysis to maximize the lifetime of a hierarchical WSN. Further simulation results confirms the effectiveness of our approach. Till now we have discussed about improving QoS requirements for event-driven and periodic data-gathering. Whereas in the next chapter we discuss about improving QoS requirements for hybrid data-gathering.

6

Minimizing Critical Event Delay and Maximizing Lifetime in a novel Delay-constrained Hybrid Data-gathering Protocol

6.1 Introduction

In time driven data-gathering applications, like environmental monitoring, sensor nodes generate periodic data. Whereas, in event-driven data-gathering applications, like intrusion detection, tsunami detection, forest-fire detection, and many more, the sensor nodes remain idle until a critical event is detected in the vicinity, and once detected, the event information is forwarded as early as possible to the

base-station. In the previous chapters, we showed how to improve QoS for event-driven and periodic data-gathering, whereas in this chapter we focus on improving QoS for hybrid data-gathering. In hybrid data gathering schemes, the sensor network switches between event-driven and time-driven according to the requirement or application specification.

In application like forest monitoring system, the sensor nodes are deployed to monitor several physical phenomena like temperature, humidity, and many more, using a time driven approach. However, if a critical event like fire is detected, this information needs to be forwarded to the base-station as early as possible, in order to minimize the damage. Similar applications in environmental monitoring are earthquake, tsunami or volcanic event detection. In these applications, sometimes it is also desirable that the critical data-packet must reach to the base-station within a strict delay-constraint.

Mostly related protocols with our protocol [59, 75, 51] are not designed to minimize the critical event reporting delay as well as maximize the network lifetime in a hybrid WSN. Energy-efficient hybrid protocol [51] gathers data pro-actively from the environment, but minimizing the delay of the critical-event information is not addressed. This motivates us to design a protocol that minimizes the critical event reporting delay as well as maximizes the network lifetime during periodic data-gathering in a hybrid WSN.

The contributions of this chapter are as follows. We propose a novel, hybrid, and dynamic data-gathering protocol to minimize the critical event reporting delay and maximize the lifetime. Using ns-2 simulation we show that our protocol decreases the critical event-reporting delay compared to recent protocols. We also extend our

analysis for a critical-delay constrained hybrid-data gathering network. We estimate the critical wake-up rate that satisfies given critical delay constraint and maximizes the lifetime. We validate our analysis using Monte-Carlo simulation.

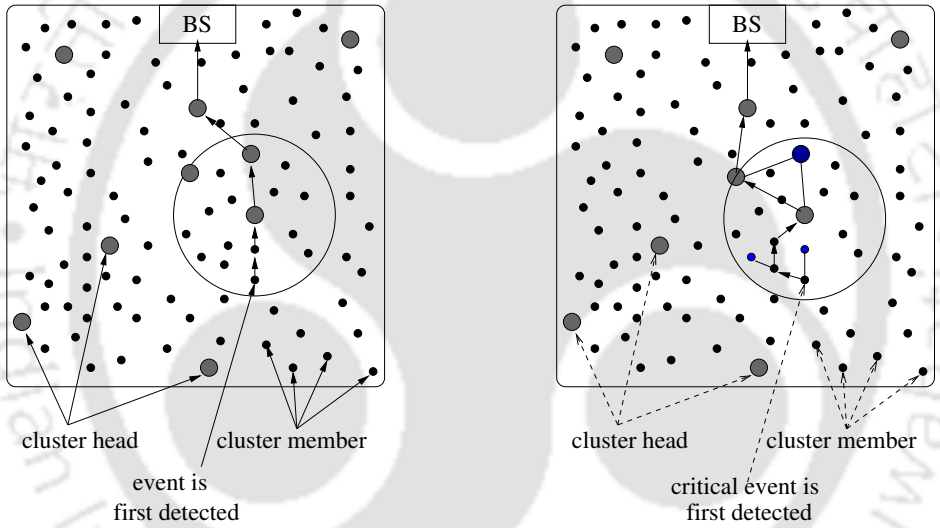
6.2 Minimizing Critical Delay and Maximizing Lifetime

Our proposed hybrid protocol switches dynamically between time-driven and event-driven data gathering schemes. Under normal conditions, the sensor nodes follow time-driven data-gathering using hierarchical clustering network. Whenever a critical event is detected, the corresponding nodes change their state to event driven. If for sometime, no more critical event is detected in the vicinity, the sensor nodes change their state back to time driven.

Our hybrid protocol consists of configuration phase and data-gathering phase. Since, critical events are rare and the network collects time driven data most of the time, our protocol follows the multi-level hierarchical clustering strategy for energy efficiency [15]. In multi-level hierarchical clustering strategy, sensor nodes are divided into clusters in each level of hierarchy. The cluster heads at level l sends their data to the cluster heads at level $l + 1$. Base-station receives the data at the highest level.

During the configuration phase, the CH selection, cluster formation and slots allocation are done. In order to conserve energy consumption, sensor nodes use multi-hop communication (like the shortest path tree) for while forwarding the data to the cluster heads [15]. In order to improve energy conservation further, the sensor

nodes follow synchronous s/w scheduling technique. We assume the sensor nodes follow TDMA based synchronous scheduling where each node wakes up at its corresponding slot to send the periodic data to the next hop node [32]. Let w denotes the periodic wake-up rate interval such that every node i wakes up after every $\frac{1}{w}$ time in its corresponding slot. Once a periodic event is detected, the sensor nodes forward the corresponding data-packet using the multi-level, multi-hop clustering technique at their corresponding slots till the data-packet reaches to the base-station which is at the highest level of the hierarchy (refer 6.1(a)).



(a) Periodic data-gathering (b) Event-driven data-gathering

Figure 6.1: Demonstrates the path a packet follows till it reaches to the base-station in a two level hierarchical network

Note that, if the wake-up rate decreases, the lifetime increases, but the expected e2e delay incurred for critical packet increases as well. In the contrary, if the periodic wake-up rate increases, expected critical event delay decreases at a cost of decreasing lifetime. In order to decrease the delay without decreasing the lifetime, sensor nodes follow anycasting forwarding technique while sending the critical data-packets. Consider a node i which belongs to the FoI and F_i denotes the for-

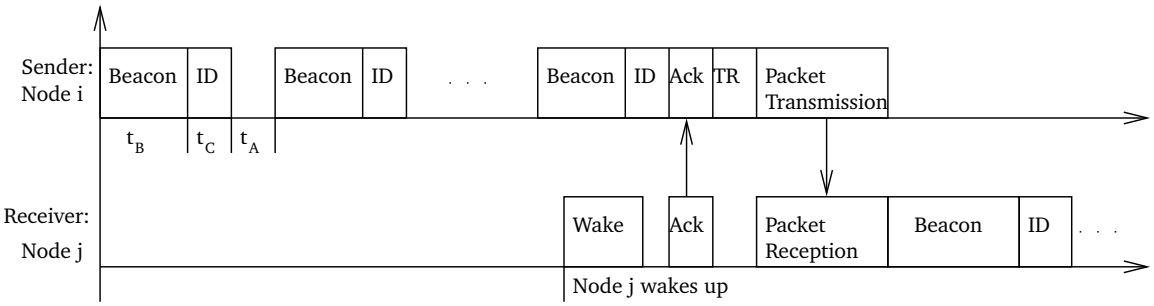


Figure 6.2: Packet forwarding protocol

warding set of node i. If any node within the forwarding set F_i wakes up, the critical event packet is immediately forwarded to this node. Compared to the designated forwarding node scheme, anycasting scheme surely minimizes expected one hop delay, which in turn minimizes the expected event reporting delay. A path followed by a packet contains in a two level hierarchical network is shown in Fig. 6.1(b). During the configuration phase the forwarding set, F_i, for each node i is constructed. The neighboring nodes of node i which are closer to the base-station are included in F_i.

If a critical event is detected, the node i changes its state from time-driven to event-driven and start transmitting a special beacon signal, followed by its own ID, using different CDMA channel, and waits for an acknowledgement from any node $n_{i_j} \in F_i$ (see Fig. 6.2). If an acknowledgement is received from node n_{i_j}, then node i sends the data-packet. If the node n_{i_j} detects a special beacon signal followed by ID of node i, then node n_{i_j} changes its state to event-driven and sends acknowledgement to node i and waits for the data-packet. If data-packet contained critical-event information is received, then the node n_{i_j} transmits a special beacon signal and follows similar procedure to forward this data-packet to the next hop node. This procedure continues until the packet reaches to the CH. CHs use similar strategy to forward the data. If no special beacon signal is detected for some-time sensor nodes switch back to time-driven state and starts periodic data-gathering.

Increasing wake-up rate decreases the critical event reporting delay with a cost of decreasing lifetime and vice versa. For a given critical delay constraint, we need to estimate the expected critical wake-up rate that satisfies the constraint and increases lifetime.[1]

6.3 Maximizing Lifetime

Assuming N nodes are deployed, we can formally write our objective as

$$\begin{aligned} &\max_{w^c} \{min(L_i), \quad \forall i\} \quad \text{subject to} \\ &L_i = \frac{Q_i}{w^p E^p + w^c E^c}, \quad \forall i, \\ &z \times w^c \geq w^p > 0, \quad \forall i, z \in \mathbb{Z}^+ \\ &max(D_i) \leq D, \quad \forall i, \\ &Q_i > 0, E^p, E^c > 0, \forall i, \end{aligned} \tag{6.1}$$

where w^c and w^p respectively denotes critical and periodic wake-up rate, L_i denotes the lifetime of node i, Q_i denotes the available energy for the data-gathering phase, E_p, E_c denotes the energy required during a periodic and critical wake-up, D_i denotes the expected critical delay of the node i and D denotes critical delay constraint. Note that the $max(D_i)$ denotes the maximum of expected end-to-end delay for event-driven and it should be less than D.

The estimation of $max(D_i)$ is shown in the following section. In order to maximize

[1]We assume that the delay incurs in periodic data packet is not a constraint.

$max(L_i) \forall i$, we maximize L_i for all i. Hence Eq. 6.1 can be written as,

$$\begin{aligned} &\max_{w^c} \{L_i\} \quad \text{subject to} \\ &L_i = \frac{Q_i}{w^p E^p + w^c E^c}, \\ &z \times w^c \geq w^p > 0, z \in \mathbb{Z}^+ \\ &max(D_i) \leq D, \\ &Q_i > 0, E^p, E^c > 0, \end{aligned} \tag{6.2}$$

for all i. As Q_i, w^p, E^p, E^c are constants for node i, this problem is same as minimizing w^c, hence we can write the Eq. 4.2 as,

$$\begin{aligned} &\min_{w^c} \{w^c\} \quad \text{subject to} \\ &z \times w^c \geq w^p > 0, z \in \mathbb{Z}^+ \\ &max(D_i) \leq D, \\ &Q_i, E^p, E^c > 0. \end{aligned} \tag{6.3}$$

We denote this wake-up rate as expected critical wake-up rate. In the following section we show how this can be estimated.

6.3.1 Expected Critical wake-up rate

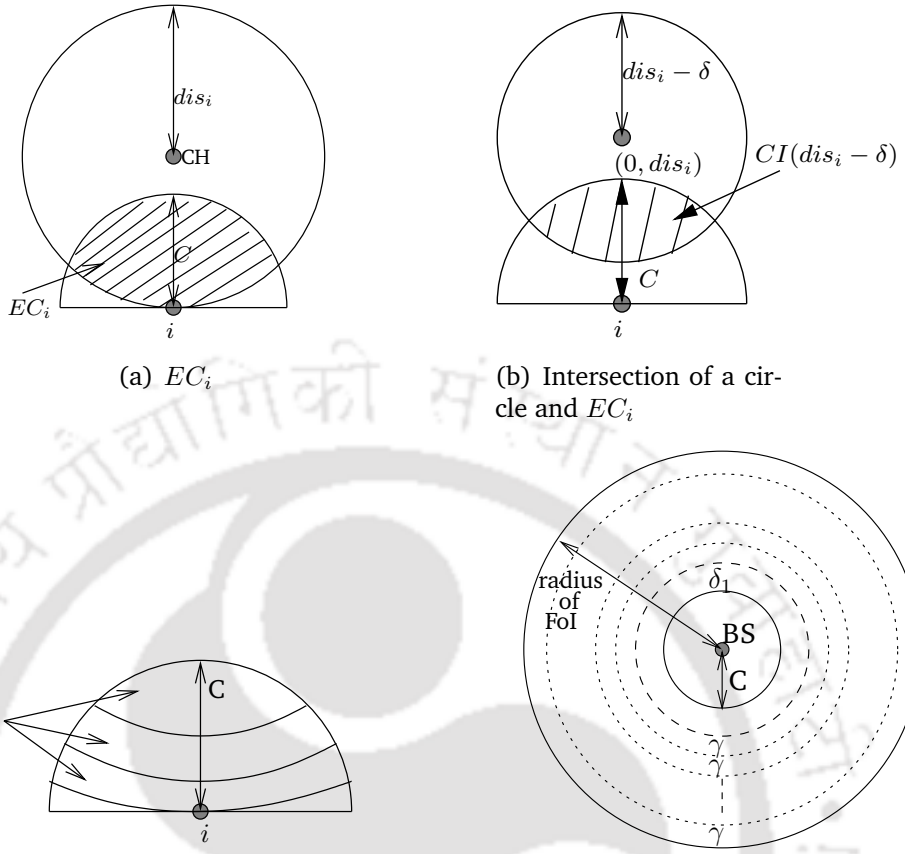
In this section we estimate the critical wake-up rate that satisfies critical e-delay constraint in a cluster. The critical e-delay in a hierarchical level is the maximum of the critical e-delays in its clusters. The critical e-delay of the FoI is the sum of the critical e-delays in the corresponding hierarchical levels. Hence, for a given wake-up rate we can estimate the critical e-delay in the FoI and use this information to find the critical wake-up rate.

In order to estimate the critical wake-up rate in a cluster, we first estimate the expected critical event reporting delay in a cluster and use this strategy to estimate the critical wake-up rate. Later we show how one can extend this strategy to find the critical wake-up rate for the given FoI.

6.3.2 Expected Critical Event Reporting Delay in a Cluster

For simplicity and ease of analysis, we assume a fixed cluster-based network where cluster-heads are fixed in clusters. Later we show how this strategy can be extended for a dynamic cluster based network where nodes change their turn to become the CHs.

In this subsection, we estimate the maximum expected critical event reporting delay using the numerical iteration technique for a given number of nodes and wake-up rate. Note that, if a critical event is detected, the corresponding packet is forwarded to the nearest cluster-head using anycasting strategy, and then the cluster-head forwards the same using anycasting technique, and this process continues till the packet reaches to the base-station. Hence, the expected critical event-



(a) EC_i

(b) Intersection of a circle and EC_i

(c) EC_i is divided into equal area sectoral annuli.

(d) Estimating critical event reporting delay of Cl

Figure 6.3: Estimating critical event reporting delay

reporting delay is the sum of the expected critical event-reporting delays at every level of hierarchy. We estimate the expected critical event reporting delay from the lowest to the highest level of hierarchy.

The expected critical event reporting delay in a level of hierarchy is the maximum of the expected critical event reporting delays in its clusters. Assuming the field of interest (FoI) is divided into convex-shaped clusters, here we estimate the maximum of expected critical event reporting delay of a cluster Cl in the lowest level of hierarchy. This method can be extended for each cluster and for each level of hierarchy to find the maximum of expected critical event-reporting delay. Let us

assume C denotes the communication range of a cluster member node. Maximum expected critical event reporting delay of the nodes in a cluster is estimated in the increasing order of their distance from the cluster-head. We estimate the critical event reporting delay of the nodes closer to the cluster-head first, and use the results to estimate the critical event reporting delay of their neighbors which are away from the cluster-head.

For simplicity we assume that the shape of the cluster is circular and the CH is located at its center. Later we extend this strategy for an convex-shaped cluster with CH positioned in arbitrary position. Let r_C denotes the radius of the maximum enclosing circle, C_q centered at the CH. We first estimate the maximum critical event reporting delay of a node in C_q. Later we extend this method to estimate the critical event reporting delay of a node $i \in (Cl - C_q)$.

Critical Delay in Maximum Enclosing Circle:

Let w and w_{CH} respectively denotes the wake-up rate such that every cluster member node and the cluster head node wakes up after every $\frac{1}{w}$ and $\frac{1}{w_{CH}}$ during time-driven approach respectively. Since the cluster head wakes up after every $\frac{1}{w_{CH}}$, the expected waiting time of a node which can directly communicate with the cluster head is $\frac{w_{CH}}{2}$. Hence, the critical event reporting delay of the nodes that can directly communicate to the cluster-head is $(\frac{w_{CH}}{2} + t_D)$, where t_D denotes the transmission delay. This circular area is referred as direct communication circle, C_D (refer Fig. 6.3(d)).

If a critical event is detected by a node $i \in (C_q - C_D)$, the event information is forwarded using anycasting technique (refer Fig. 6.3(d)). Before sending a data

packet containing critical event information, a node sends a beacon signal of duration t_B, followed by an ID signal of duration t_C, and listens for acknowledgment of duration t_A. A node sends acknowledgement if it hears the beacon and belongs to F_i. If $F_i = \{i_1, i_2, ..., i_k\}$ then the probability of a node wakes up within F_i at h^{th} beacon is $p_w = \frac{t_I}{1/w}$, if $h \leq \lceil \frac{1/w}{t_I} \rceil$, else 1, where $t_I = t_A + t_B + t_C$ denotes the beacon interval.

Let W denotes the event for the set of forwarding modes waking up at their respective beacons. The probability of W is denoted as $P(W) = (p_w)^k$. Let X denotes the event such that for the first $h-1$ beacons no nodes wakes up, then at h^{th} beacon j number of nodes wake up, during the last $h_{max} - h$ beacons remaining $k-j$ nodes wake up. Moreover, $h_{max} = \frac{1/w}{t_I}$ denotes the total number of beacons. The probability of the event $X, P(X)$, is

$$P(X) =^k C_j (h_{max} - h)^{k-j} (p_w)^k. \tag{6.4}$$

Let W_h be the event that denotes the packet is forwarded after h beacons. The probability of the event W_h is

$$P(W_h) = \sum_{j=1}^{k} {}^k C_j (h_{max} - h)^{k-j} (p_w)^k. \tag{6.5}$$

Hence the expected one hop delay for critical event reporting is

$$d_{k,w} = \sum_{h=1}^{\left\lfloor \frac{1/w}{t_I} \right\rfloor} P(W_h) * h + t_D, \tag{6.6}$$

where t_D is the transmission delay. The expected critical event reporting delay is the summation of the expected one hop delay and expected critical event reporting delay of the next hop node. As the probability of the packet is forwarded to a node in F_i is $\frac{1}{k}$ we get the following lemma for expected critical event reporting delay.

Lemma 10. *Let $\{i_1, i_2, ..., i_k\}$ is the forwarding set of node i. If $D_{i_j,k,w}$, denotes the expected critical event reporting delay of node i_j, for $1 \leq j \leq k$, respectively, then the expected critical event reporting delay of the node $i : (i \neq j)$, is $D_{i,k,w} = d_{k,w} + \sum_{j=1}^{k} \frac{1}{k} * D_{i_j,k,w}$, where $d_{k,w} = \sum_{h=1}^{\left\lfloor \frac{1/w}{t_I} \right\rfloor} P(W_{h,k}) * h + t_D$.*

Note that, the critical event information packet is forwarded to any node that is closer to the cluster-head. We denote this area as the effective forwarding region of communication, EC_i. Consider a node i, located $(0, 0)$, at a distance dis_i from the cluster-head. Note that EC_i is the intersection of the communication region of node i and the open circular area with radius dis_i centered at the cluster-head (see Fig. 6.3(a)).

Lemma 11. *Let the node i located $(0, 0)$, at a distance dis_i from the cluster-head located $(0, dis_i)$ (see Fig. 6.3(a)). Area of the effective forwarding region of communication of the node i with communication range C such that $dis_i > C$, is $||EC_i|| = 2\cos^{-1}\left(\frac{C}{2dis_i}\right)C^2 + 2\cos^{-1}\left(1 - \frac{C^2}{2{dis_i}^2}\right){dis_i}^2 - dis_i\sqrt{C^2 - \left(\frac{C^2}{2dis_i}\right)^2}$.*

Above lemma can be proved using simple geometry. Note that for simplicity we assume the node i located $(0, 0)$ and the cluster-head located $(0, dis_i)$. Lemma 11

can be extended for node i located (x_i, y_i) with distance dis_i from an arbitrary located cluster-head. Moreover, the expected number of nodes in F_i, $k = \lambda \times ||EC_i|| - 1$, since $i \in EC_i$.

In order to estimate the critical event reporting delay of a node, we divide C_q into several rings of concentric circles (centered at cluster-head) in the steps of γ. Next we estimate the expected critical reporting delay for a random node in each ring in increasing order of distance. Using the expected critical event reporting delay of the nodes within the distance $C + (m-1)\gamma$ from the cluster head we estimate the expected critical event reporting delay of a randomly chosen node i at distance $C + m\gamma$ for $m \in N$.

Consider a node i at a distance $C + m\gamma$ for $m \in N$ from the cluster-head. In order to estimate the expected critical event reporting delay, we first estimate the intersection of EC_i and a circle centered at cluster-head.

Lemma 12. *Consider a node i located $(0, 0)$, at a distance dis_i from the cluster head located $(0, dis_i)$, with communication range C, such that $dis_i > C$. The intersection of a circle centered at cluster head (q) with radius $R_j = dis_i - \delta, 0 < \delta \leq C$, and the effective forwarding region of communication of node i, $CI(dis_i, \delta)$, is given by Eq. 6.7 (see Fig. 6.3(b)).*

$$
\begin{aligned}
CI(dis_i, \delta) = {} & 2\cos^{-1}\left(\frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i C}\right) C^2 \\
& - \frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i}\sqrt{C^2 - \left(\frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i}\right)^2} \\
& + 2\cos^{-1}\left(\frac{dis_i - \frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i}}{dis_i - \delta}\right) C^2 - \left(dis_i - \frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i}\right) \\
& \times \sqrt{C^2 - \left(\frac{\delta^2 - c^2 - 2dis_i\delta}{2dis_i}\right)^2}
\end{aligned} \tag{6.7}
$$

The above lemma can be proved using geometry. Next we divide EC_i into several sectoral annuli such that it is expected that every sectoral annulus contains only one node (refer Fig. 6.3(c)). for $1 \leq j \leq k$, where $k = ||EC_i|| * \lambda - 1$. The j^{th} sectoral annulus $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$ of node i, between two concentric circles, centered at q with radii $\beta_{i_{j1}}, \beta_{i_{j2}}$, such that $\beta_{i_{j1}} > \beta_{i_{j2}}$, is defined as the intersection of EC_i and the area between their boundaries.

Let the closest sectoral annulus to the cluster-head be i_1. $\beta_{i_{11}}$ is equal to $dis_i - C$. One can find $\beta_{i_{12}}$ by solving Eq. 6.8.

$$
CI(dis_i, \beta_{i_{12}} - (dis_i - C)) = \frac{1}{\lambda}. \tag{6.8}
$$

Note that $\beta_{i_{21}} = \beta_{i_{12}}$. Moreover, $\beta_{i_{j1}} = \beta_{i_{(j-1)2}}$, for $2 \leq j \leq k$. For an arbitrary i_j, $\beta_{i_{j2}}$ can be found by solving equation as follows.

$$CI(dis_i, \beta_{i_{j2}} - \beta_{i_{j1}}) = \frac{1}{\lambda}, \tag{6.9}$$

for $1 \leq j \leq k$. As the equations 6.8 and 6.9 are single variable equations, we assume they can be solved in constant time.

First we estimate the expected critical event reporting delay of a randomly chosen node within j^{th} sectoral annulus, for $1 \leq j \leq k$, and then use these to estimate the expected critical event reporting delay of a random node i.

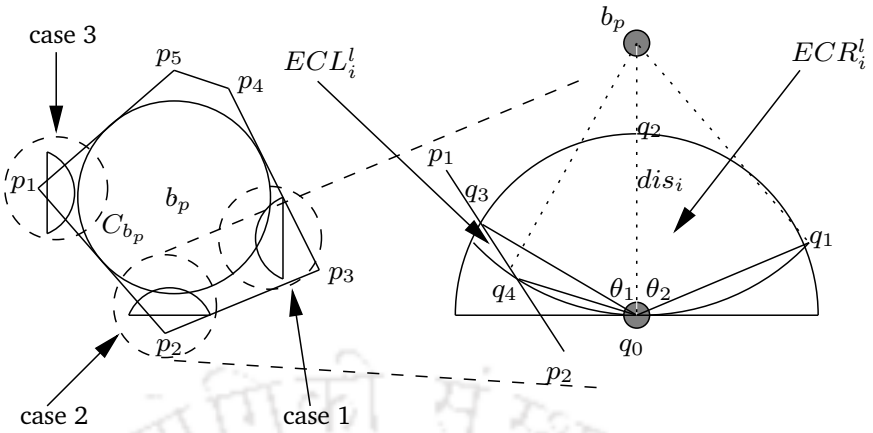
Consider a randomly chosen sectoral annulus $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$. Assume m_{i_1} is the largest integer such that $C + \delta_1 + m_{i_1}\gamma \leq \beta_{i_{j1}}$ and m_{i_2} is the smallest integer such that $C+\delta_1+m_{i_2}\gamma \geq \beta_{i_{j2}}$. We use the expected critical event reporting delay of nodes at distances $C + \delta_1 + (m_{i_1} + 1)\gamma, C + \delta_1 + (m_{i_1} + 2)\gamma, ..., C + \delta_1 + m_{i_2}\gamma$, to estimate the expected critical event reporting delay of a randomly chosen node belongs to $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$. Note that the estimated minimum expected critical event reporting delay of a randomly chosen node belongs to $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$ is proportional to the area induced by the intersection between $SA_{i,j}(\beta_{i_{j1}}, \beta_{i_{j2}})$ and the ring formed by the corresponding circular annuli.

Area induced by $SA_{i,j}(C + \delta_1 + m_{i_1}\gamma, C + \delta_1 + (m_{i_1} + 1)\gamma)$ is $CI(dis_i, dis_i - (C + \delta_1 + (m_{i_1} + 1)\gamma)) - \frac{1}{\lambda}(k - j)$. The area induced by $SA_{i,j}(C + \delta_1 + t\gamma, C + \delta_1 + (t + 1)\gamma)$ for $m_{i_1} < t \leq (m_{i_2} - 1)$ is given in Eq. 6.10.

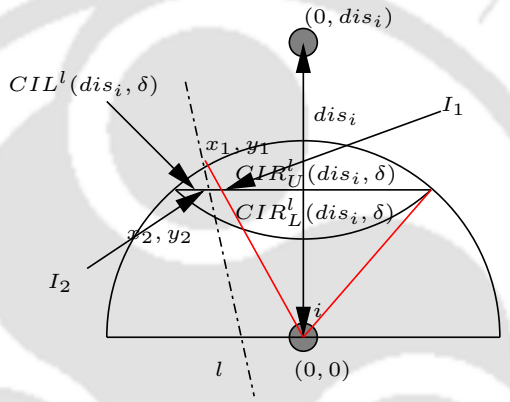
$$||SA_{i,j}(C+\delta_1+t\gamma, C+\delta_1+(t+1)\gamma)|| = CI(dis_i, dis_i-(C+\delta_1+t\gamma)) - \sum_{s=1}^{t-1} \|SA_{i,j}(C+\delta_1+s\gamma, C+\delta_1+(s+1)\gamma)\| - \frac{1}{\lambda}(k-j). \quad (6.10)$$

Probability of node i_j belongs within $SA_{i,j}(C+\delta_1+t\gamma, C+\delta_1+(t+1)\gamma)$ is $||SA_{i,j}(C+\delta_1+t\gamma, C+\delta_1+(t+1)\gamma)||\lambda$. Let $D_{i_j,t\gamma}$ denotes the minimum expected critical event reporting delay of a random node within $SA_{i,j}(C+\delta_1+t\gamma, C+\delta_1+(t+1)\gamma)$. Hence, the expected critical event reporting delay D_{i_j} of node i_j, is $\sum_{t=m_{i_1}}^{(m_{i_2}-1)} ||SA_{i,j}(C+\delta_1+t\gamma, C+\delta_1+(t+1)\gamma)||\lambda D_{i_j,t\gamma}$.

We use expected critical event reporting delay D_{i_j} of a random node belongs to ${i_j}^{th}$ sectoral annulus, to find the minimum expected critical event reporting delay of node i at a distance $dis_i = C+\delta_1+m\gamma$. Consider a randomly chosen node i at a distance $dis_i = C+\delta_1+m\gamma$ from the cluster-head. We divide the effective forwarding region of communication, EC_i, into $EC_i * \lambda - 1$ sectoral annuli such that it is expected that every sectoral annulus contains only one node. Assuming D_{i_j} denotes the estimated expected critical event reporting delay for a randomly selected node within circular annulus i_j, for $1 \leq j \leq (EC_i * \lambda - 1)$, which is calculated as shown earlier using the expected critical event reporting delay of nodes at distances $C+\delta_1+\gamma, C+\delta_1+2\gamma, ..., C+\delta_1+(m-1)\gamma$. The expected critical event reporting delay of node i can be calculated using Lemma 10.



(a) EC_i'



(b) New $CI(dis_i, \delta)$

Figure 6.4: Critical event reporting delay of a convex-shaped cluster

Critical Event Delay of a Convex Shaped Cluster

In order to find the expected end-to-end delay of a node $i \in (Cl - C_q)$, we first find the effective forwarding region of communication of the node i. Let $< p_1, p_2, p_3, p_4, p_5 >$ denotes the vertices of a convex shaped cluster (refer Fig. 6.4(a)). The effective forwarding region of communication of the node i is the intersection of EC_i and Cl. There are three cases depending on the number of edges of the convex polygon Cl intersecting EC_i as shown in the Fig. 6.4(a). Using simple geometry one can find the new area of effective forwarding region of communication, $||EC_i'||$. Moreover $k = ||EC_i'|| \times \lambda - 1$

In similar way, the area of intersection of a circle centered at cluster-head and the effective forwarding region of communication of a random node can be estimated since there are three possible cases depending on the number of edges of the polygon CL intersecting $CI(dis_i, \delta)$ (see Fig. 6.4(b)).

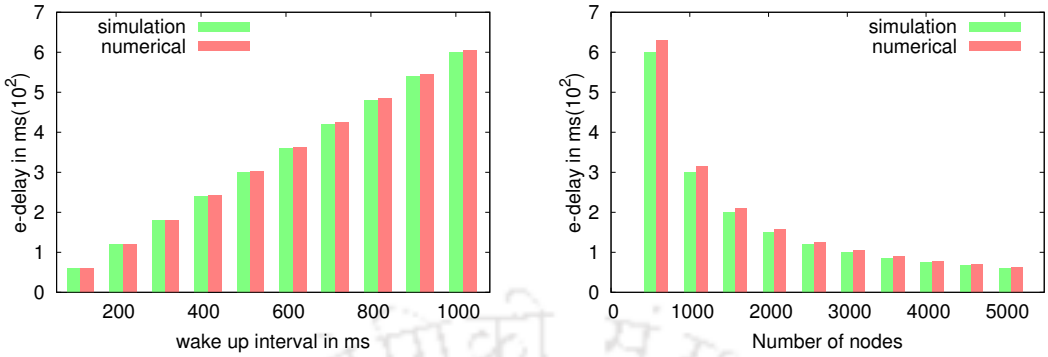
Now we are ready to estimate the expected critical event reporting delay of a random node $i \in (Cl - C_q)$. Depending on the position of the node i, the expected critical event reporting delay changes within the same circular annulus. This is because as the position changes, $||EC_i'||$ and critical event reporting delay of the next-hop nodes change as well. As mentioned earlier, Cl is divided into several circular annuli in the steps of γ. In order to estimate the delay of a random node within each sectoral annulus, one can use the similar strategy in Sec. 6.3.2 except that each circular annulus is subdivided into several small parts (refer Fig. 6.3(d)).

Overall Critical Event Reporting Delay

In order to find the critical event reporting delay in a hierarchical level, we find the cluster with maximum critical event reporting delay. The overall event reporting delay in the FoI is the summation of the critical event reporting delays in each hierarchical level.

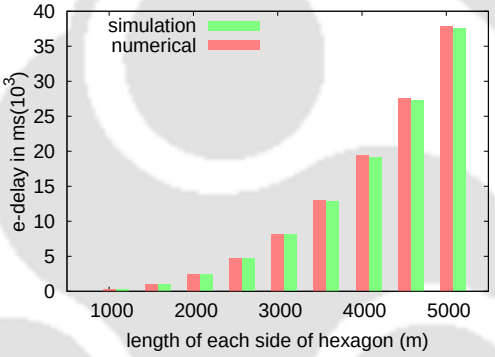
6.3.3 Critical Wake-up rate

Note that the critical wake-up interval is less than the delay constraint and be denoted by r_w^{max}. Assuming r_w^{min} to be zero we use binary search between r_w^{max} and r_w^{min} to find the minimum r_w that satisfies given delay constraint, using the method described in subsection 6.3.2, and increases lifetime.



(a) Wake up intervals

(b) Density



(c) FoI

Figure 6.5: Validating the estimated expected e2e delay

6.3.4 Validation of the Analysis using Monte-Carlo Simulation

In order to validate our analysis, in this subsection we numerically evaluate the expected critical e2e delay using the method described in subsection 6.3.2 for given density and wake-up rate, and compare with simulation results. Specifically, this section validates the analysis (and equations) by picking up its input from the same distribution used for the analysis, and run it for a large number of times to show that the analysis obtained is correct. We uniform randomly deploy 1000 nodes, with communication range of $200m$, in a circular FoI with 1000m radius, and calculate

the average e2e delay from the farthest node during each experiment. Note that the density of such a network is $\frac{1}{\pi \times 1000}$. We repeat such experiment by changing the seed of uniform distribution and redeploy the nodes.

Each experiment consists of several tests, where in each experiment we calculate the average critical e2e delay. Each test consists of the following things. Note that every node has equal wake-up interval as w. Let h_i denotes that node i wakes up at h^{th} beacon where h_i is uniformly distributed within $[1, h_{max}]$. In other words, node i wakes up at h_i beacon signal which is selected from a set of uniform randomly distributed integers within $[1, h_{max}]$. The forwarding set of node i, F_i, contains the neighboring nodes that are closer to the CH and also within the communication range. Assume i_f denotes the node with the farthest distance from CH. The time when the critical event is generated at node i_f is selected from a set of uniform randomly distributed numbers within $[1, h_max]$. Let t_{i_f} denotes the time when the event is detected. Moreover, assume $i_{f_k} \in F_{i_f}$ is the forwarding node such that the difference between $h_{i_{f_k}}$ and t_{i_f} is minimum. In other words, i_{f_k} wakes up at the earliest after t_{i_f}. Hence, one hop delay between the event is detected to the event is forwarded to the next hop node is $h_{i_{f_k}} - t_{i_f}$. In similar ways, we calculate the delay to reach the packet to the CH. We repeat such tests for 10,000 times, re-assign $h_i \forall i$, t_{i_f} and calculate the average critical e2e delay.

For simplicity, we place the cluster-head at the center of the FoI. We repeat the experiment for 100 times. The average e2e delay obtained from the simulation results along with the numerical estimation are shown in Fig. 3.6. Our numerical estimation is close to the simulation results for various scenarios.

Impact of wake up interval: For different wake-up intervals, average e2e delays

are shown in Fig. 6.5(a). When the wake up interval increases, expected waiting time before sending a packet also increases. This in turn increases e2e delay. Also note that the estimated expected e2e delay is always higher than the average e2e delay obtained from the simulation.

Impact of density: Average e2e delays are shown in Fig. 6.5(b), while varying the number of nodes in the FoI, for a fixed wake-up interval $500\,ms$. If the number of nodes in the forwarding set increases, expected one hop delay decreases, which in turn decreases expected e2e delay. Though the difference in simulation and numerical estimation decreases as we increase the density, but the percentage of over estimation on the expected e2e delay is almost same.

Impact of FoI: The average maximum e2e delays, while varying the size of FoI are shown in Fig. 6.5(c), for a fixed wake up interval. If the size of FoI increases, the number of hops from the farthest node increases as well, which in turn increases e2e delay.

6.4 Simulation Results

We deploy 200 nodes in $500 \times 500\ m^2$ area using a random uniform distribution and simulated our hybrid protocol using the network ns2. We use Mica 2 motes [4] parameters along with the power requirements shown in [57],[85]. Other parameters used in simulation are given in Table 6.1. We define the lifetime of the network as the lifetime of the first node that depletes its energy completely.

For cluster formation, sensor nodes use energy efficient hierarchical clustering algorithm [15], in which the nodes use multi-hop communication for intra and

Table 6.1: Simulation Parameters

Name	Value
Area under simulation	$500 \times 500\ m^2$
Communication range	$250\,m$
Radio transceiver type	$CC\,1000$
Data rate	19.2 kbps
Initial energy	1000 J
Receiving/Idle Power	13.0 mW
Transmission Power	19.5 mW
Energy required from sleep to awake	22.0 μJ
Data packet length	8 Byte
Beacon/Ack packet length	3 Byte
Synchronization protocol	RBS
Wireless media	802.11

inter-cluster communication. We set the number of clusters in each round as 4% of the total nodes. The cluster-head collects data-packets from cluster-members, aggregate the packets and forwards the resultant packets to the base-station. In order to avoid collision, TDMA and CDMA based approaches are respectively used for intra and inter cluster communication. For synchronization we use RBS [30] using the implementation described in [85].

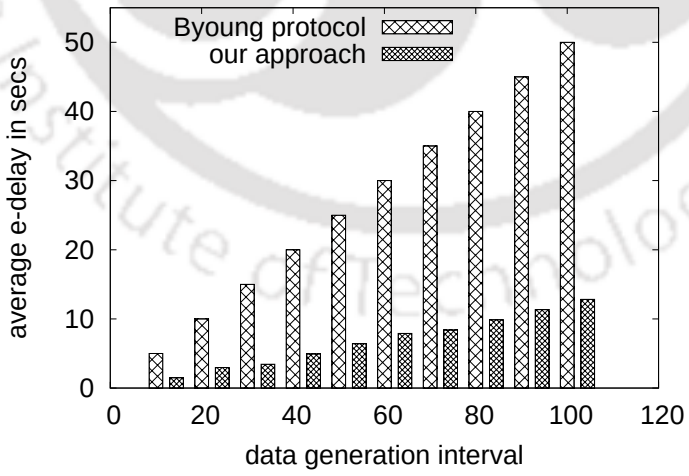


Figure 6.6: Data generation interval

In order to show the effectiveness of our approach, we compare average criti-

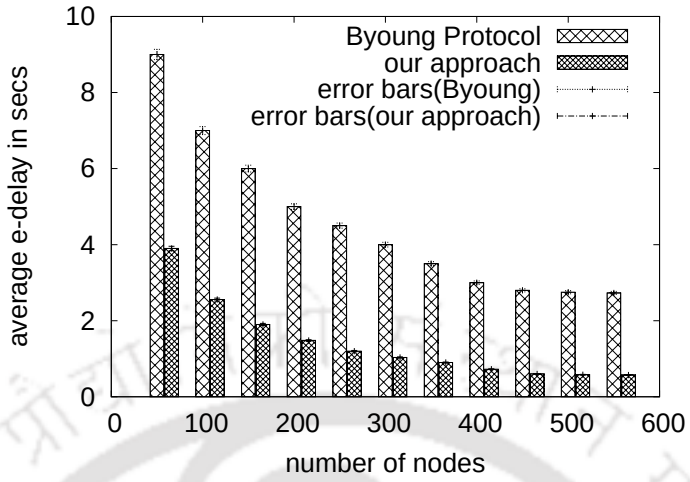


Figure 6.7: Number of nodes

cal event reporting delay, lifetime and packet loss rate by varying data-generation rate, density and size of FoI in both our hybrid protocol and energy-efficient hybrid protocol [51] and show the results obtained in ns2 simulation in the following subsection.

6.4.1 Average critical event reporting delay

In order to calculate the critical event delay in each experiment, we generate the location and the time of the critical event by following uniform random distribution within the FoI. We repeat such an experiment 1000 times and calculate the average critical event delay. It can be noted that compared to energy-efficient hybrid protocol [51], our approach minimizes the delay significantly in different scenarios.

Impact of periodic data generation interval: The average critical event delay for different data generation intervals are shown in Fig. 6.6. As data generation interval increases, average critical event delay for both the methods increases as

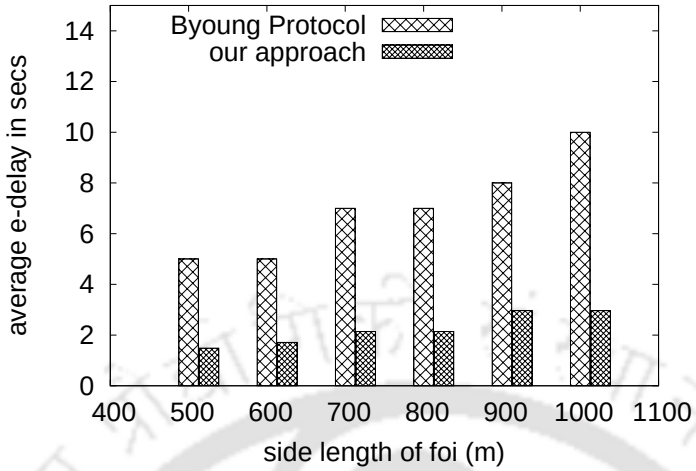


Figure 6.8: Side length of the rectangular FoI

well. This is because, if data-generation interval increases, expected one-hop delay increases, which in-turn increases critical event reporting delay in both hybrid and our approach. Also note that, the average critical delay in case of our approach is much lower than hybrid approach. This is because, our protocol uses anycasting strategy which minimizes the critical event reporting delay significantly. Moreover, if an event is detected, in energy efficient hybrid approach the nodes forward their data packet to the designated forwarding nodes whereas in our approach the data-packet is forwarded using anycasting strategy, which minimizes the critical event-reporting delay significantly.

Impact of number of node: In order to show the effectiveness of our approach at different densities, we varied the number of nodes deployed in the FoI. For a fixed data generation interval of 10 seconds, the results are shown in Fig. 6.7 along with 95% error bars (not clearly visible). Note that, increasing the number of nodes in the given FoI, decreases the average one hop delay, which in turn decreases average critical event reporting delay, in both energy efficient hybrid protocol and our

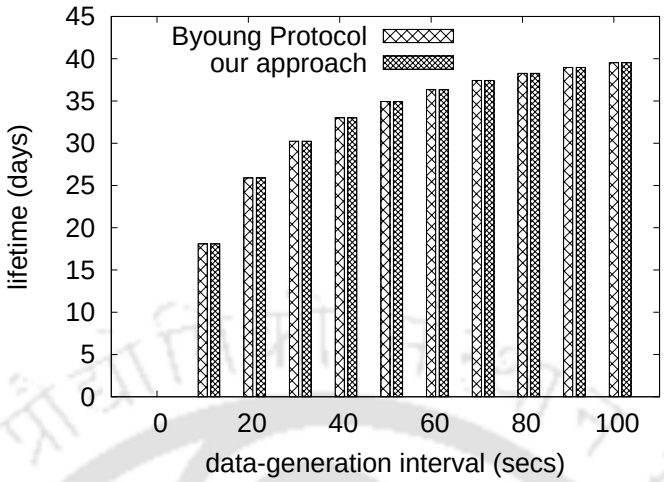


Figure 6.9: Comparing lifetime

approach. Moreover, as our protocol uses anycasting strategy, the average critical event reporting delay decreases significantly. Moreover, when the number of nodes is more than a threshold (450 in Fig. 6.7), amount of delay decreases as number of nodes increases rarely and becomes almost constant.

Impact of FoI: The average critical event reporting delays for different sizes of rectangular FoI are shown in Fig. 6.8. In order to vary the size of the FoI, we vary the side length of the FoI. We set the data generation interval as $10\,s$ and number of nodes as 200. As the size of the FoI increases, maximum hop count for intra and inter cluster communication increases as well, which lead to increase in average critical event reporting delay. Moreover, because of anycasting our protocol significantly decreases the critical event delay as the size of area increases.

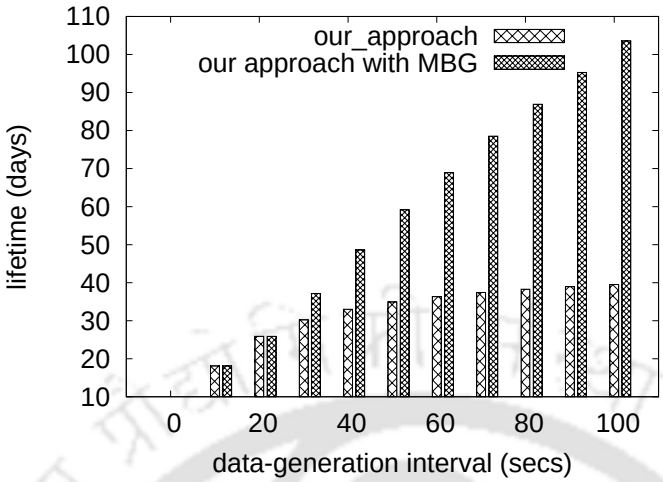


Figure 6.10: Comparing lifetime of our approach with MBG

6.4.2 Average lifetime

In Fig. 6.9, we compare the average network lifetime while increasing data generation interval. As it can be noted that when data generation interval increases lifetime increases as well. The lifetime is mostly affected by periodic data generation interval, as we assumed that critical events occur rarely. As both our protocol and energy efficient hybrid protocol uses similar methods during periodic data-gathering, the lifetime in both the approaches matches closely. Moreover, when MBG is applied during each periodic wake-up, the lifetime decreases significantly as shown in Fig. 6.10. This is because, during every wake-up as a result of guard-time, energy consumption increases rapidly when data generation interval increases. Whereas, MBG saves this extra energy consumption as shown in the figure.

It can also be noted that the lifetime reaches a limit. Actually, if the data-generation interval crosses a threshold, amount of energy saving during a transmission using MBG becomes almost constant. This is because, energy consumption in MBG

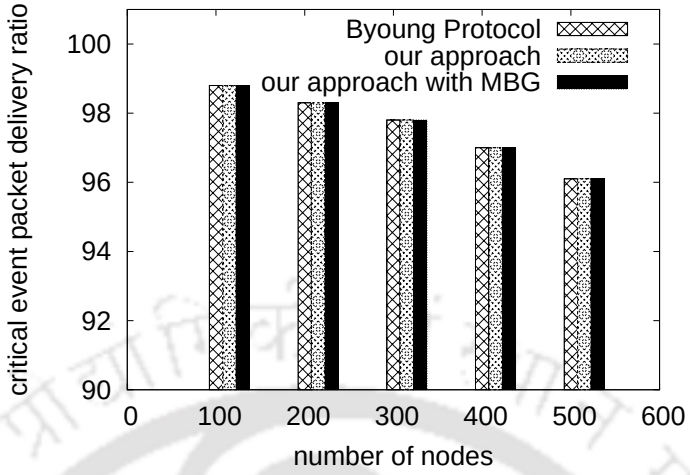


Figure 6.11: Percentage of packets delivered

mostly govern by the energy consumption to transmit the beacon messages. If data-generation interval is very high, the increase in energy for beacon messages is very less compared to the increase in energy for guard-time.

6.4.3 Packet delivery ratio

We set data-generation interval as 1 seconds in $500 \times 500\, m^2$ FoI. We compare the packet delivery ratio in energy efficient hybrid [51] protocol, and our hybrid protocol for different densities by varying number of nodes (see Fig. 6.11). If the number of nodes increases, the probability of a collision with an event reporting and data gathering packet increases as well, and thus decreases the packet delivery ratio marginally. Moreover, the critical packet delivery ratio closely matches in both the methods.

6.5 Conclusion

In this work, we proposed a protocol that dynamically switches between event-driven and periodic data gathering and shown the effectiveness of our approach over conventional approach by comparing several network parameters. Our protocol is effective especially for time-critical event detection in hybrid data-gathering applications. In the next chapter we conclude the thesis with pointers to the future works.

7

Conclusions and Future Works

This thesis has made four major research contributions to improve various QoS requirements in data-gathering techniques for WSNs with stochastic deployment of sensors. Mostly we concerned about maximizing the lifetime, minimizing the e2e delay and minimizing the deployment cost for various data-gathering schemes.

Chapter 3 and chapter 4 deals with improving QoS requirements for event-driven data-gathering. In chapter 3, we estimated the maximum expected e2e delay for a given sensor density and used this analysis to estimate the CSDs that satisfy given delay constraint and lifetime requirement. Our analysis is based on the assumption that sensor nodes are deployed following uniform stochastic distribution. Using our analysis one can successfully deploy sensor nodes in a convex-shaped FoI which satisfies given e2e delay constraint and lifetime requirement. We further extend this analysis to find a minimum cost network for given spatially differentiated delay constraints and lifetime requirement. But this delay constraint may not be always

satisfied as a result of clock-skew. In order to constrain this increase in e2e delay for clock-skew within a given threshold ξ, we provided a solution in chapter 4. Our approach decreases this additional delay while maximizing the lifetime. We estimated the increase in e-delay as a result of heterogeneous clock-skew using stochastic analysis. We use this estimation to find the critical wake-up rate for all nodes to constrain the increase in overall e-delay within given threshold ξ.

In chapter 5, we concentrated on improving QoS requirements for periodic data-gathering. In order to circumvent the message loss, the receiver uses guard-time in the presence of clock-skew during a transmission. However, this method drains out of energy if the length of guard-time is more. In this chapter, we proposed the *MBG* method where the sensor nodes follow the multi-beacon approach if the length of guard-time is more, otherwise follow the simple guard-time approach to conserve energy. In multi-beacon approach, the guard-time is divided into unequal intervals and a sensor node wakes up at the beginning of these intervals. After each unsuccessful attempt in multi-beacon approach the senor nodes save energy by going back to sleep state again. We studied the problem of energy efficient transmission between a sender-receiver pair and extended this analysis to maximize the lifetime of a hierarchical cluster-based WSN. Further simulation results confirms the effectiveness of our approach.

In hybrid data-gathering, the sensor network switches between event-driven and periodic. In applications like environmental monitoring with disastrous critical events (tsunami or earthquake) detection, the sensor nodes use periodic data-gathering for most of the time and if a critical event is detected the event information must reach to the base-station as early as possible. For such applications, some of the most important QoS requirements are the critical event reporting de-

lay and the lifetime of the overall network. In chapter 4, we propose a protocol that minimizes the critical event reporting delay by using anycasting technique and maximizes the lifetime during periodic data-gathering. In order to show the effectiveness in the presence of clock-skew, we used MBG during periodic data-gathering and shown how it increases the lifetime further. The effectiveness of our approach is shown in ns2 simulation results.

7.1 Future Work Directions

This thesis gives rise to a number of important research directions for improving QoS in various data-gathering schemes.

- In event-driven data-gathering, our work in chapter 3 minimizes the overall cost of the network with spatially differentiated delay constraint and lifetime requirement. Our approach saves more cost for a lesser lifetime requirement and a higher rate of change of delay constraint. Moreover, our analysis can be extended to 3-dimensional heterogeneous WSNs with different communication range.

- We assumed a single base-station is located within the given FoI. Whereas, it would be interesting to see how our analysis can be extended to find the minimum cost network for a given FoI where multiple base-stations are located at multiple points and each base-station is allotted with a different e2e delay requirement. In the presence of clock-skew, the actual delay constraint may not be always satisfied. Whether the analysis provided in chapter 3 can be extended for polygon shaped FoI with multiple base-stations is an interesting

question.

- Chapter 6 proposed a protocol that maximizes the lifetime and minimizes the critical event reporting delay in a hybrid data-gathering. But sometimes the critical event reporting delay constraint is given. Hence, in our future work, we will design a protocol that maximizes the lifetime in a critical event delay constrained hybrid data-gathering technique for WSNs.

- Anycasting plays an importing role for minimizing the e2e delay in WSN. Whereas, anycasting technique is not addressed till now in mobile WSNs. In a delay constrained mobile event-driven WSN, anycasting technique can be applied to decrease the overall e2e delay. In our future work we devise an efficient delay-constrained protocol for such a network that maximizes the overall lifetime as well.

Bibliography

[1] Contiki, www.sics.se/contiki.

[2] Liteos, www.tinyos.net/.

[3] Tinyos, www.tinyos.net/.

[4] www.eol.ucar.edu/isf/facilities/isa/internal/crossbow.

[5] www.isi.edu/nsnam/ns/ns-largesim.html.

[6] www.pcl.cs.ucla.edu/projects/glomosim/.

[7] www.tinyos.stanford.edu/tinyos-wiki/index.php/tossim.

[8] I. F. Akyildiz, T. Melodia, and K. R. Chowdury. Wireless multimedia sensor networks: A survey. *IEEE Wireless Communications*, 14(6):32–39, 2007.

[9] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[10] I. Al-Anbagi, M. Erol-Kantarci, and H. T. Mouftah. A survey on cross-layer quality-of-service approaches in WSNs for delay and reliability-aware applications. *IEEE Communications Surveys & Tutorials*, 18(1):525–552, 2016.

[11] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.

[12] H. M. Ammari and S. K. Das. Joint k-coverage and hybrid forwarding in duty-cycled three-dimensional wireless sensor networks. In *Proceedings of the IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 170–178. IEEE, 2008.

[13] H. M. Ammari and S. K. Das. Critical density for coverage and connectivity in three-dimensional wireless sensor networks using continuum percolation. *IEEE Transactions on Parallel and Distributed Systems*, 20(6):872–885, 2009.

[14] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *Proceedings of the ACM International Symposium on Mobile Ad-hoc Networking and Computing*, pages 131–142. ACM, 2006.

[15] S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of IEEE International Conference on Computer Communications*, pages 1713–1723. IEEE, 2003.

[16] D. Bhattacharyya, T. Kim, and S. Pal. A comparative study of wireless sensor networks and their routing protocols. *Sensors*, 10(12):10506–10523, 2010.

[17] M. M. Bhuiyan, I. Gondal, and J. Kamruzzaman. Codar: Congestion and delay aware routing to detect time critical events in wsns. In *Proceedings of the International Conference on Information Networking*, pages 357–362. IEEE, 2011.

[18] S. Biswas and R. Morris. ExOR: opportunistic multi-hop routing for wireless networks. *ACM SIGCOMM Computer Communication Review*, 35(4):133–144, 2005.

[19] W. Bober and C. J. Bleakley. BailighPulse: A low duty cycle data gathering protocol for mostly-off wireless sensor networks. *Computer Networks*, 69:51–65, 2014.

[20] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the International conference on Embedded networked sensor systems*, pages 307–320. ACM, 2006.

[21] H. Cai, X. Jia, and M. Sha. Critical sensor density for partial connectivity in large area wireless sensor networks. *ACM Transactions on Sensor Networks*, 7(4):35:1–35:23, 2011.

[22] Q. Cao, T. Abdelzaher, J. Stankovic, and T. He. The liteos operating system: Towards unix-like abstractions for wireless sensor networks. In *Proceedings of the International Conference On Information Processing in Sensor Networks*, pages 233–244. IEEE, 2008.

[23] D. Chen and P. K. Varshney. QoS support in wireless sensor networks: A survey. In *Proceedings of the International Conference on Wireless Networks*, pages 1–7, 2004.

[24] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.

[25] F. Cuomo, A. Abbagnale, and E. Cipollone. Cross-layer network formation for energy-efficient ieee 802.15. 4/zigbee wireless sensor networks. *Ad Hoc Networks*, 11(2):672–686, 2013.

[26] P. Ding, J. Holliday, and A. Celik. Distributed energy-efficient hierarchical clustering for wireless sensor networks. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems*, pages 322–339. Springer, 2005.

[27] D. E. Dominici. The inverse of the cumulative standard normal probability function. *Integral Transforms and Special Functions*, 14:281–292, 2003.

[28] K. Dorling, G. G. Messier, S. Valentin, and S. Magierowski. Minimizing the net present cost of deploying and operating wireless sensor networks. *IEEE Transactions on Network and Service Management*, 12(3):511–525, 2015.

[29] A. Dunkels, B. Gronvall, and T. Voigt. Contiki: A lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the IEEE International Conference on Local Computer Networks*, pages 455–462. IEEE, 2004.

[30] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, pages 147–163. ACM, 2002.

[31] M. Enachescu, A. Goel, R. Govindan, and R. Motwani. Scale free aggregation in sensor networks. In *Proceedings of the Algorithms and Experiments for Sensor Systems*, pages 71–84. Springer, 2004.

[32] S. Gandham, Y. Zhang, and Q. Huang. Distributed time-optimal scheduling for convergecast in wireless sensor networks. *Computer Networks*, 52(3):610–629, 2008.

[33] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems*, pages 138–149. ACM, 2003.

[34] S. Ganeriwal, I. Tsigkogiannis, H. Shim, V. Tsiatsis, M. B. Srivastava, and D. Ganesan. Estimating clock uncertainty for efficient duty-cycling in sensor networks. *IEEE/ACM Transactions on Networking*, 17(3):843–856, 2009.

[35] D. Gay, P. Levis, R. Von Behren, M. Welsh, et al. The nesc language: A holistic approach to networked embedded systems. In *Proceedings of the ACM Special Interest Group on Programming Languages*, pages 1–11. ACM, 2003.

[36] A. Goel and D. Estrin. Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk. *Algorithmica*, 43(1-2):5–15, 2005.

[37] T. He, J. A Stankovic, C. Lu, and T. Abdelzaher. Speed: A stateless protocol for real-time communication in sensor networks. In *Proceedings of the International Conference on Distributed Computing Systems*, pages 46–55. IEEE, 2003.

[38] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless sensor networks. In *Proceedings of the Hawaii International Conference on System Sciences*, page 10, 2000.

[39] W. Hu, N. Bulusu, and S. Jha. A communication paradigm for hybrid sensor/actuator networks. *International Journal of Wireless Information Networks*, 12(1):47–59, 2005.

[40] C. Hua and T. S. P. Yum. Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks. *IEEE/ACM Transactions on Networking*, 16(4):892–903, 2008.

[41] T. Issariyakul and E. Hossain. *Introduction to network simulator NS2*. Springer Science & Business Media, 2011.

[42] U. Jang, S. Lee, and S. Yoo. Optimal wake-up scheduling of data gathering trees for wireless sensor networks. *Journal of Parallel and Distributed Computing*, 72(4):536–546, 2012.

[43] A. Kemal and Y. Mohamed. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, 2005.

[44] M. Khan and G. Pandurangan. A fast distributed approximation algorithm for minimum spanning trees. *Distributed Computing*, 20(6):391–402, 2008.

[45] J. Kim, X. Lin, and N. B. Shroff. Optimal anycast technique for delay-sensitive energy-constrained asynchronous sensor networks. *IEEE/ACM Transactions on Networking*, 19(2):484–497, 2011.

[46] J. Kim, X. Lin, N. B. Shroff, and P. Sinha. Minimizing delay and maximizing lifetime for wireless sensor networks with anycast. *IEEE/ACM Transactions on Networking*, 18(2):515–528, 2010.

[47] H. Kour and A. K. Sharma. Hybrid energy efficient distributed protocol for heterogeneous wireless sensor network. *International Journal of Computer Applications*, 4(6):1–5, 2010.

[48] S. M. Lasassmeh and J. M. Conrad. Time synchronization in wireless sensor networks: A survey. In *Proceedings of the IEEE Southeast Conference*, pages 242–245. IEEE, 2010.

[49] L. Lazos and R. Poovendran. Stochastic coverage in heterogeneous sensor networks. *ACM Transactions on Sensor Networks*, 2(3):325–358, 2006.

[50] L. Lazos and R. Poovendran. Stochastic coverage in heterogeneous sensor networks. *ACM Transactions on Sensor Networks*, 2(3):325–358, 2006.

[51] B. D. Lee and K. H. Lim. An energy-efficient hybrid data-gathering protocol based on the dynamic switching of reporting schemes in wireless sensor networks. *IEEE Systems Journal*, 6(3):378–387, 2012.

[52] P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: Accurate and scalable simulation of entire tinyos applications. In *Proceedings of the International Conference on Embedded Networked Sensor Systems*, pages 126–137. ACM, 2003.

[53] P. Levis, S. Madden, J. Polastre, R. Szewczyk, et al. *Tinyos: An operating system for sensor networks*. Springer, 2005.

[54] Z. Lin and M. van der Schaar. Autonomic and distributed joint routing and power control for delay-sensitive applications in multi-hop wireless networks. *IEEE Transactions on Wireless Communications*, 10(1):102–113, 2011.

[55] S. Lindsey and C. S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. In *Proceeding of the Aerospace Conference*, pages 3–8. IEEE, 2002.

[56] S. Liu, F. Kai-Wei, and P. Sinha. CMAC: An energy efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks. In *Proceedings of the IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 11–20, 2007.

[57] J. Ma, W. Lou, Y. Wu, X. Y. Li, and G. Chen. Energy efficient TDMA sleep scheduling in wireless sensor networks. In *Proceedings of IEEE International Conference on Computer Communications*, pages 630–638. IEEE, 2009.

[58] A. Manjeshwar and D. P. Agrawal. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In *Proceedings of the International Parallel & Distributed Processing Symposium*, page 189, 2001.

[59] A. Manjeshwar and D. P. Agrawal. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *Proceedings of the IEEE International Parallel and Distributed Processing Symposium*, page 48, 2002.

[60] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi. The flooding time synchronization protocol. In *Proceedings of the International Conference on Embedded Networked Sensor Systems*, pages 39–49. ACM, 2004.

[61] D. L. Mills. Internet time synchronization: the network time protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, 1991.

[62] X. Min, S. Wei-Ren, J. Chang-Jiang, and Z. Ying. Energy efficient clustering algorithm for maximizing lifetime of wireless sensor networks. *AEU-International Journal of Electronics and Communications*, 64(4):289–298, 2010.

[63] N. Mitton, D. Simplot-Ryl, and I. Stojmenovic. Guaranteed delivery for geographical anycasting in wireless multi-sink sensor and sensor-actor networks. In *Proceedings of the Annual IEEE Conference on Computer Communications*, page 000, 2009.

[64] S. Nandi and A. Yadav. Cross layer adaptation for qos in wsn. *ArXiv Preprint ArXiv:1110.1496*, 2011.

[65] G. Paillard and V. Ravelomananana. Limit theorems for degree of coverage and lifetime in large sensor networks. In *Proceedings of the Conference on Computer Communications*, pages 2011–2019. IEEE, 2008.

[66] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems*, pages 95–107. ACM, 2004.

[67] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1):63–78, 2006.

[68] M. Rossi and M. Zorzi. Integrated cost-based MAC and routing techniques for hop count forwarding in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 6(4):434–448, 2007.

[69] M. Rossi, M. Zorzi, and R. R. Rao. Statistically assisted routing algorithms (SARA) for hop count based forwarding in wireless sensor networks. *Wireless Networks*, 14(1):55–70, 2008.

[70] A. Sahoo and S. Chilukuri. Dgram: a delay guaranteed routing and mac protocol for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(10):1407–1423, 2010.

[71] J. G. Saw, M. C. K. Yang, and T. C. Mo. Chebyshev inequality with estimated mean and variance. *The American Statistician*, 38(2):130–132, 1984.

[72] C. Schurgers, V. Tsiatsis, and M.B. Srivastava. STEM: Topology management for energy efficient sensor networks. In *Proceedings of the IEEE Conference on Aerospace*, pages 1099–1108. IEEE, 2002.

[73] M. L. Sichitiu and C. Veerarittiphan. Simple, accurate time synchronization for wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking*, pages 1266–1273, 2003.

[74] M.L. Sichitiu. Cross-layer scheduling for power efficiency in wireless sensor networks. In *Proceedings of the IEEE International Conference on Computer Communications*, pages 1740–1750. IEEE, 2004.

[75] C. Srisathapornphat, C. Jaikaeo, and C. C. Shen. Sensor information networking architecture. In *Proceedings of the International Workshops on Parallel Processing*, pages 23–30. IEEE, 2000.

[76] M. Srivastava, R. Muntz, and M. Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments. In *Proceedings of the International Conference on Mobile Computing and Networking*, pages 132–138. ACM, 2001.

[77] B. Sundararaman, U. Buy, and A. D. Kshemkalyani. Clock synchronization for wireless sensor networks: A survey. *Ad Hoc Networks*, 3(3):281–323, 2005.

[78] N. Thepvilojanapong, Y. Tobe, and K. Sezaki. On the construction of efficient data gathering tree in wireless sensor networks. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 648–651. IEEE, 2005.

[79] T. Van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems*, pages 171–180. ACM, 2003.

[80] J. Van Greunen and J. Rabaey. Lightweight time synchronization for sensor networks. In *Proceedings of the ACM International Conference on Wireless Sensor Networks and Applications*, pages 11–19. ACM, 2003.

[81] B. Wang. Coverage problems in sensor networks: A survey. *ACM Computing Surveys*, 43(4):32:1–32:53, 2011.

[82] M. K. Watfa and S. Commuri. An energy efficient and self-healing 3-dimensional sensor cover. *International Journal of Ad Hoc and Ubiquitous Computing*, 3(1):33–47, 2008.

[83] Y. Wei, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the IEEE International Conference on Computer Communications*, pages 1567–1576. IEEE, 2002.

[84] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, 2006.

[85] Y. Wu, S. Fahmy, and N. B. Shroff. Optimal sleep/wake scheduling for time-synchronized sensor networks with QoS guarantees. *IEEE/ACM Transactions on Networking*, 17(5):1508–1521, 2009.

[86] Y. Wu, S. Fahmy, and N. B. Shroff. Sleep/wake scheduling for multi-hop sensor networks: Non-convexity and approximation algorithm. *Ad Hoc Networks*, 8(7):681–693, 2010.

[87] Y. Wu, Z. Mao, S. Fahmy, and N. B. Shroff. Constructing maximum-lifetime data-gathering forests in sensor networks. *IEEE/ACM Transactions on Networking*, 18(5):1571–1584, 2010.

[88] F. Xia. QoS challenges and opportunities in wireless sensor/actuator networks. *Sensors*, 8(2):1099–1110, 2008.

[89] J. Xiao, C. Qi, and L. Shu. Research of coverage control algorithm in three-dimensional wireless sensor network based on energy efficiency. In *Proceedings of the Chinese Control and Decision Conference*, pages 3363–3368. IEEE, 2012.

[90] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks*, 1(1):36–72, 2005.

[91] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks*, 1(1):36–72, 2005.

[92] W. Ye, F. Silva, and J. Heidemann. Ultra-low duty cycle MAC with scheduled channel polling. In *Proceedings of International Conference on Embedded Networked Sensor Systems*, pages 321–334. ACM, 2006.

[93] L. H. Yen, C. W. Yu, and Y. M. Cheng. Expected k-coverage in wireless sensor networks. *Ad Hoc Networks*, 4(5):636–650, 2006.

[94] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.

[95] M. A. Yigitel, O. D. Incel, and C. Ersoy. QoS-aware MAC protocols for wireless sensor networks: A survey. *Computer Networks*, 55(8):1982–2004, 2011.

[96] H. Zhang and J. Hou. On deriving the upper bound of α-lifetime for large sensor networks. In *Proceedings of the ACM International Symposium on Mobile Ad-hoc Networking and Computing*, pages 121–132. ACM, 2004.

[97] H. Zhang and J. Hou. Is deterministic deployment worse than random deployment for wireless sensor network? In *Proceedings of the IEEE International Conference on Computer Communications*, pages 1–13. IEEE, 2006.

[98] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, 1(1-2):89–124, 2005.

[99] J. Zhang, W. Li, D. Cui, X. Zhao, and Z. Yin. The NS2-based simulation and research on wireless sensor network route protocol. In *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4. IEEE, 2009.

[100] Y. Zhang and Q. Huang. A learning-based adaptive routing tree for wireless sensor networks. *Journal of Communications*, 1(2):12–21, 2006.

[101] Y. Zhao, J. Wu, F. Li, and S. Lu. Vbs: maximum lifetime sleep scheduling for wireless sensor networks using virtual backbones. In *Proceedings of the IEEE International Conference on Computer Communications*, pages 1–5. IEEE, 2010.

[102] M. Zorzi and R. R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance. *IEEE Transactions on Mobile Computing*, 2(4):349–365, 2003.

Publications

Journals

- **D. Sadhukhan**, S. V. Rao. Effect of Clock-skew in Event Driven, Delay Constrained Heterogeneous WSN with Anycast. Wireless Personal Communications (Springer), 2017 **(Accepted)** [Chapter 4]
- **D. Sadhukhan**, S. V. Rao. MBG: Energy Efficient Multi-Beacon Guard Method for Periodic Data Gathering in Time Synchronized WSN. Computer Networks (Elsevier), 2017 (Under Review). [Chapter 5]
- **D. Sadhukhan**, S. V. Rao. Minimum Cost Event Driven WSN with Spatial Differentiated QoS Requirements. Wireless Netowrks, (Springer), 2017, (Under Review). [Chapter 3]

Conference

- **D. Sadhukhan**, S. V. Rao. Critical Sensor Density for Event-driven Data-gathering in Delay and Lifetime Constrained WSN. In Proceedings of IFIP Wired/Wireless Internet Communications, held in Thessaloniki, Greece, May 25-27th 2016, also appeared in LNCS springer. [Chapter 2]