

AN ENERGY-EFFICIENT REAL-TIME COORDINATION AND ROUTING  
FRAMEWORK FOR WIRELESS SENSOR ACTOR NETWORKS

A THESIS SUBMITTED TO  
THE GRDUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GHALIB ASADULLAH SHAH

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN  
COMPUTER ENGINEERING

JANUARY 2007

---

Prof. Dr. Canan ÖZGEN  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

---

Prof. Dr. Ayse KIPER  
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

---

Prof. Dr. Muslim BOZYIGIT  
Supervisor

**Examining Committee Members**

Prof. Dr. Semih Bilgen (METU, EEE) \_\_\_\_\_

Prof. Dr. Muslim Bozyigit (METU, CENG) \_\_\_\_\_

Prof. Dr. Payidar Genc (METU, CENG) \_\_\_\_\_

Prof. Dr. Ali YAZICI (TOBB, CE) \_\_\_\_\_

Prof. Dr. Adnan YAZICI (METU, CENG) \_\_\_\_\_

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Ghalib Asadullah SHAH

Signature :

# **ABSTRACT**

## **AN ENERGY-EFFICIENT REAL-TIME COORDINATION AND ROUTING FRAMEWORK FOR WIRELESS SENSOR ACTOR NETWORKS**

Ghalib Asadullah Shah

Ph.D., Department of Computer Engineering

Supervisor : Prof. Dr. Muslim Bozyigit

January 2007, 162 pages

In Wireless Sensor Actor Networks (WSANs), sensor nodes perform the sensing task and actor nodes take action based on the sensed phenomenon. The presence of actors in this configuration can not be benefited from, unless they are able to execute actions at right place and right time in the event region. The right place can be related to the accurate position of the sensor nodes. While, the right time is related to delivering the packets directly to the appropriate actors within the event specific response times. Hence, the efficient localization of sensor nodes, sensor-actor/actor-actor coordination and real-time routing is indispensable in WSANs. Furthermore, the limited energy levels and bandwidth of the state of art sensor nodes currently impose stringent requirements for low-complexity, low-energy, distributed coordination and cooperation protocols and their implementation.

In this study, we propose an integrated framework which addresses the issues of sensors localization, network configuration, data aggregation, real-time data delivery, sensor-actor/actor-actor coordination and energy saving mechanisms. The proposal incorporates novel approaches on three fronts; (1) timing-based sensors localization (TSL) algorithm to localize the sensor nodes relative to actors, (2) real-time coordination and routing protocols and (3) energy conservation. The distributed real-time coordination and routing is implemented in addressing and greedy modes routing. A cluster-based real-time coordination and routing (RCR) protocol operates in addressing mode. The greedy mode routing approach (Routing by Adaptive Targeting, RAT) is a stateless shortest path routing. In dense deployment, it performs well in terms of delay and energy consumption as compared to RCR. To keep the traffic volume under control, the framework incorporates a novel real-time data aggregation (RDA) approach in RCR such that the packets deadlines are not affected. RDA is adaptive to the traffic conditions and provides fairness among the farther and nearer cluster-heads. Finally, framework incorporates a power management scheme that eliminates data redundancy by exploiting the spatial correlation of sensor nodes.

Simulation results prove that the framework provides the real-time guarantees up to 95 % of the packets with lesser energy consumption of up to 33 % achieved using MEAC as compared to LEACH and SEP. The packet delivery ratio is also 60 % higher than that of *semi-automated architecture*. Furthermore the action accuracy is supported by TSL which restricts the localization errors less than 1 meter by tuning it according to the expected velocity of nodes and required accuracy.

**Keywords:** Wireless Sensor Actor Networks (WSAN), Localization of Sensor Nodes, Real-time Routing, Coordination, Exploiting Spatial Correlation

## ÖZ

### **Sensör ve Aktör Agları için Enerji-Etkin Kosum-Zamani Koordinasyon ve Yol verme Çerçevesi**

Sensör ve actor aglarında sensörler algılamayla, aktörler ise algılanan olaya karşılık verme veya müdahale etmeyle görevlidir. Bir ag birden fazla olaya müdahale için programlanmış olabilir. Bu durumda, sensörler ayrı ayrı birden fazla farklı algaç modüllü ve farklı enerji kaybetme hızlarından ötürü farklı enerji seviyeli olabilirler. Aktör varlığının kabul edilebilmesi için doğru müdahalelerin doğru zaman ve yerde yapılması beklenir. Doğru yer demek sensörlerin yerlerinin aktörlere göre bilinmesine, doğru zaman ise uygulamaya bağlı olarak algılanan olaylara karşılık verme zamanına uygunluk demektir. Bundan böyle, bu tür ağlarda yer belirleme, koordinasyon ve gerçek zamanlı yol verme teknikleri vazgeçilmezdir. Ayrıca, günümüz sensör ve aktörlerindeki enerji ve iletişim kapasitesi seviyeleri koordinasyon ve işbirliği protokolleri için çok önemli sınırlamalar oluşturmaktadırlar.

Bu çalışmada, sensör yerlerinin tespiti, ağ konfigürasyonu, veri birikimi, gerçek zaman veri iletimi, sensör-aktör/aktör-aktör koordinasyonu ve enerji tasarrufu mekanizmaları gibi konuları içeren tümleşik bir çerçeve önerilmektedir. Öneri, üç boyutta yeni yöntemler içermektedir; (1) zamana dayalı sensör yerlerinin tespiti (TSL), (2) yol verme algoritmaları, (3) enerji tasarrufu. TSL sensör düğümlerinin aktörlere göre yerlerini belirler. Dağıtık gerçek-zaman koordinasyon, düğüm ID ve bencil yol vermeye göre çalışmaktadır. Öbeklemeye bağlı gerçek-zaman koordinasyon ve yol verme protokolü (RCR) düğüm ID modunda çalışmaktadır. Bencil yol verme yöntemi (RAT), durumsuz en kısa bir yol

verme teknigidir. RAT, yksek yogunluklu aqlarda enerji bakımından diđer yntemlere gre daha iyi sonu vermektedir.

*To my late uncle and my mother.*



## **ACKNOWLEDGEMENTS**

It is my pleasure to express my sincere gratitude to my supervisor Prof. Muslim Bozyigit. I am grateful to him for leading me into this exciting research area and assisting me in all kinds of problems no matter they are academic or non-academic whenever I knocked his door.

I am extremely indebted to Prof. Buyurman Baykal and Associate Prof. Ozgur B. Akan for encouragement and insight in this new field. I am also obliged to thank Prof. Semih Bilgen for his invaluable feedback during all of my progress committees. I would also like to express my gratefulness to other jury members Prof. Payidar Genç, Prof. Ali Yazici and Prof. Adnan Yazici for their advice during the thesis defense. I also appreciate Associate Prof. Demet Aksoy for her invaluable feedback and support in the thesis.

I like to acknowledge NUST, National University of Science and Technology Pakistan, and TÜBİTAK, The Scientific and Technological Research Council of Turkey, for partially sponsoring this work.

Last, but not the least, I thank my family, particularly my elder brother Dr. Khalid Saifullah, for their understanding, love, and countless prayers during all of my endeavors. I am in depth to my dear wife Humaira and cute son Ali for their patience and understanding during this study. I also would like to thanks my friends, particularly Sajjad, Prof. Fakhar Mahmood and his family, for their company and continuous support.

With the bottom of my heart I am extremely gratified to almighty Allah for providing me opportunities and guidance in times of difficulties to reach to this milestone in my life.

# TABLE OF CONTENTS

PLAGIARISM .....	iii
ABSTRACT .....	iv
ÖZ.....	vi
ACKNOWLEDGEMENTS .....	ix
TABLE OF CONTENTS .....	x
CHAPTER	
1. INTRODUCTION .....	1
1.1 Design Challenges.....	4
1.2 Our Contributions .....	8
1.3 Overview of ERCR Framework .....	12
1.3.1 Timing-based Sensors Localization (TSL) .....	13
1.3.2 Self-configuring Clustering Protocol: MEAC .....	13
1.3.3 Delay-constrained Energy-Aware Routing (DEAR) .....	14
1.3.4 Sensor-Actor/Actor-Actor Coordination.....	15
1.3.5 Routing by Adaptive Targeting (RAT) .....	16
1.3.6 Real-time Data Aggregation (RDA).....	16
1.3.7 Exploiting Energy-aware Spatial Correlation .....	17
1.4 Energy Model .....	17
1.5 Simulations .....	18
1.6 Thesis Organization .....	20
2. TIMING-BASED SENSOR LOCALIZATION (TSL) IN WSANs .....	21
2.1 Introduction .....	21
2.2 Related Work.....	23
2.2.1 Range-based Localization .....	23
2.2.2 Range-free Localization .....	24
2.3 TSL Approach .....	26
2.3.1 Overview of the Algorithm .....	27
2.3.2 Pulse Transmission in TSL .....	29
2.3.3 Node Localization .....	32

2.3.4	Collaborative Localization.....	37
2.3.5	Accuracy in TSL and Parameter Selection .....	38
2.4	Performance Evaluation .....	39
2.5	Summary .....	41
3.	MULTI-EVENT ADAPTIVE CLUSTERING PROTOCOL (MEAC) .....	47
3.1	Introduction .....	47
3.2	Heterogeneity Model.....	48
3.3	MEAC Protocol.....	52
3.3.1	Optimal Number of Clusters ( $k_{opt}$ ) .....	53
3.3.2	Optimal Cluster Size ( $N_{opt}$ ).....	57
3.3.3	Cluster Formation.....	58
3.3.4	Inter-Clusters Connectivity .....	63
3.4	Performance Evaluation .....	63
3.5	Summary .....	69
4.	REAL-TIME COORDINATION AND ROUTING (RCR) .....	70
4.1	Introduction .....	70
4.2	Delay-constrained Energy Aware Routing (DEAR) Protocol .....	71
4.2.1	Network Model .....	71
4.2.2	Establishing Routes.....	72
4.2.3	Sink-based DEAR (S-DEAR).....	73
4.2.4	Alternative Path Selection .....	75
4.3	Sensor-Actor (S-A) Coordination .....	76
4.4	Real-time Data Aggregation (RDA) .....	77
4.5	Actor-Actor (A-A) Coordination .....	79
4.5.1	Event Targeting .....	80
4.5.2	Example Scenario.....	83
4.6	Performance Evaluation .....	83
4.6.1	S-A Coordination .....	85
4.6.2	A-A Coordination.....	87
4.6.3	Aggregation.....	87
4.7	Summary .....	91
5.	RAT: ROUTING BY ADAPTIVE TARGETING .....	93
5.1	Introduction .....	93
5.2	Delay-constrained Geographical (DC-GEO) Routing Mechanism .....	95
5.2.1	Delay Measurement .....	95
5.2.2	Delay-constrained Forwarding Set (DFS).....	96
5.2.3	Dual-Forwarding Recovery.....	99

5.2.4	DC-GEO Overheads.....	103
5.3	Location-aided Integrated Pull/Push (IPP) Coordination .....	103
5.3.1	Sensor-Actor (S-A) Coordination .....	104
5.3.2	Actor-Actor (A-A) Coordination .....	107
5.4	Performance Evaluation .....	108
5.5	Summary .....	113
6.	EXPLOITING ENERGY-EFFICIENT SPATIAL CORRELATION.....	114
6.1	Introduction .....	114
6.2	Related Work.....	115
6.3	Cluster-based Spatial Correlation of Sensor Nodes.....	117
6.3.1	Correlation Model .....	117
6.3.2	Gridiron Spatial Correlation (GSC) .....	119
6.3.3	Scheduling Nodes.....	121
6.4	Performance Evaluation .....	123
6.5	Summary .....	127
7.	CONCLUSION AND FUTURE WORK .....	128
7.1	Conclusion .....	128
7.2	Future Work .....	130
	BIBLIOGRAPHY .....	131
	CURRICULUM VITAE .....	144

## LIST OF TABLES

1.1	MICA Mote Specification .....	4
1.2	Radio characteristics used in simulation model. ....	18
5.1	Number of Nodes vs Event Radius .....	109

## LIST OF FIGURES

1.1	Architecture of WSAN .....	2
1.2	Architecture of ERCR framework showing the interaction of components incorporated. ....	13
2.1	Architecture of TSL .....	27
2.2	Packet format of <i>reference pulse</i> .....	29
2.3	Transmission pattern of <i>reference pulse</i> by 4 actors. ....	33
2.4	Location estimation of a sensor node. ....	35
2.5	Algorithm of Node Localization.....	43
2.6	Collaborative location estimation of a sensor node receiving pulses from 3 actors. ....	44
2.7	Location errors with fixed values of $\mathfrak{I}$ when the nodes are silent.....	44
2.8	Location errors with fixed values of $\mathfrak{I}$ when nodes are generating traffic at 2kbps. ....	45
2.9	Location errors with different values of $\eta$ . Maximum nodes velocity is 50 m/s and $\mathfrak{I}$ is adjusted according to $\eta$ and $\nu$ .....	45
2.10	Location errors and communication cost with different values $f$ set according to $\eta$ and $\nu$ .....	45
2.11	Location errors with fixed values of $\mathfrak{I}$ for different number of nodes.....	46
2.12	Location errors of nodes with maximum velocity of 10 m/s and varying $\mathfrak{I}$ set according to $\eta$ & $\nu$ where Fig 2.12(a), 2.12(b) and 2.12(c) show errors for $\nu = 1m/s$ , $\nu = 2m/s$ and $\nu = 3m/s$ respectively. ....	46
3.1	Heterogeneity due to number of $m$ nodes. ....	50
3.2	Network model to formulate the optimal clusters. Fig 3.2(a) represents the model to find the probability of DN nodes. Fig 3.2(b) illustrates the model of routing packets from cluster-heads to the sink node.....	53
3.3	Cluster formation in SEP and MEAC for different values of number of nodes. ....	56
3.4	Size of clusters in the network of 100 nodes deployed in $100 \times 100 m^2$ field. ....	58

3.5	Cluster-head election procedure. ....	62
3.6	Energy consumption comparison among MEAC, LEACH and SEP in the presence of heterogeneity due to energy for $\alpha = 0$ in 3.6(a), $m = 20%$ , $\alpha = 1$ in 3.6(b) and $m = 20%$ , $\alpha = 3$ in 3.6(c). ....	64
3.7	MEAC stable period in heterogeneous WSNs for different values of $\lambda_{rate}$ ...	65
3.8	Throughput comparison among MEAC, LEACH and SEP. ....	66
3.9	Energy consumption in heterogeneous WSN ( $k = 30%$ , $\varphi = 1$ , $m = 20%$ , $\alpha = 1$ ) for uniform 3.9(a), and non-uniform 3.9(b) deployment of nodes. ....	67
3.10	Average reporting rate in heterogeneous WSN ( $q = 30%$ , $\varphi = 1$ , $m = 20%$ , $\alpha = 1$ ) for uniform 3.10(a), and non-uniform 3.10(b) deployment of nodes. ....	68
4.1	Decomposition of graph G into the minimized acyclic subgraph $\bar{G}$ within the region G. ....	73
4.2	Degree of aggregation applied by cluster-heads at different route lengths from the sink node at $\rho_o = 3$ samples/sec. ....	79
4.3	Positioning actor toward uncovered source. ....	81
4.4	Algorithm of action coverage of a source. ....	82
4.5	A-A coordination to target the event area. ....	83
4.6	150 sensor nodes and 4 actors randomly deployed in $100 \times 100$ and action range of actors set to 25 meters. ....	84
4.7	Packet delivery ratio in <i>semi-automated architecture</i> and <i>automated architecture</i> for 2 samples/second triggered by 3 phenomenon nodes of event radius 50 meters. ....	86
4.8	200 sensor nodes and 4 actors randomly deployed in $150 \times 150$ . ....	86
4.9	Performance for different number of sensor nodes with 4 actor nodes randomly deployed. ....	88
4.10	Performance for different number of actor nodes deployed with 200 sensor nodes in $150 \times 150$ . ....	89
4.11	Event reporting delay for different values of $\tau$ . ....	89
4.12	Mean aggregation achieved at different values of $\tau$ . ....	90
4.13	Event reporting delay with different values of aggregation achieved applied for different values of $\tau$ . ....	90
4.14	Event delivery ratio for different values of $\tau$ . ....	91
4.15	Mean energy consumption for different values of $\tau$ . ....	91
5.1	Delay in DC-GEO vs pure greedy routing. ....	98
5.2	Algorithm of the DC-GEO protocol for selecting forwarding node. ....	102

5.3	Packet forwarding from source to destination actor in the presence of void region in a highly non-uniform deployment of nodes. ....	102
5.4	Average response time in RAT. ....	109
5.5	Average number of hops for different action range of actors in RAT. ....	110
5.6	Delivery ratio in RAT. ....	110
5.7	Unattended requests in RAT. ....	111
5.8	Average energy consumption in RAT. ....	111
5.9	Energy and delay comparison at different subscription period. ....	112
6.1	Spatial resolution control mechanism employed by the cluster-head. ....	120
6.2	Spatial resolution in cluster zone. ....	121
6.3	State transition of member nodes in exploiting GSC. ....	123
6.4	Event reporting rate of active nodes. ....	124
6.5	Active nodes for different values of correlation region $\theta$ . ....	125
6.6	Distortion in sensor readings at different values of correlation region $\theta$ ....	125
6.7	Average packet delay observed at different values of correlation region. ....	125
6.8	Packet delivery ratio of active nodes at different values of correlation region. ....	126
6.9	Energy consumed per packet delivery at different values of correlation region. ....	126



# CHAPTER 1

## INTRODUCTION

Wireless sensor and actor networks (WSANs) have emerged as a new computing class that is composed of a large number of sensor nodes as well as, relatively, small number of actor nodes which are embedded in their operating environments. All of these nodes are often distributed over wide geographical delta or located in remote and largely inaccessible regions [1]. Sensor nodes are tiny, resource-constrained devices capable of sensing, computation and communication. On the other hand, actors are mostly mobile, resource-rich devices capable of making decisions by themselves and executing actions on certain input received from the sensor nodes.

Generally, the sensor nodes are densely deployed in the field in a random fashion to detect various events and report them to the sink node, which is responsible for the monitoring of the field. In practice, it is not only important to monitor the environment, but also to react to it. The recent advances in the field of wireless networks has made it possible to include actuator nodes in the field which react appropriately in the environment on occurrence of certain events. This allows us to instrument, observe, and respond to the physical world on scales of space and time that was impossible in the past.

Typically, the architecture of WSAN consists of sensors which sense the phenomena, a sink that collects the data from the sensors to process and actors that act upon the commands sent by the sink. In the literature, such architecture is known as *semi-automated architecture*. An architecture in which sensor nodes send information to the actor nodes

directly without the involvement of sink node is called an *automated architecture* [1] as depicted in Fig 1.1. Apparently, the communication path in *semi-automated architecture* introduces significant delay, which is not acceptable for delay-sensitive applications. For example, consider a military application where sensors in the battlefield will detect the movement of red forces and send the information to the sink which is situated in a remote command and control station. The sink then sends action commands to actors for taking some measures in the reported area. In this case unnecessary delay is introduced due to sensor-sink communication which could be prevented if the actors can take localized actions without the involvement of the sink.

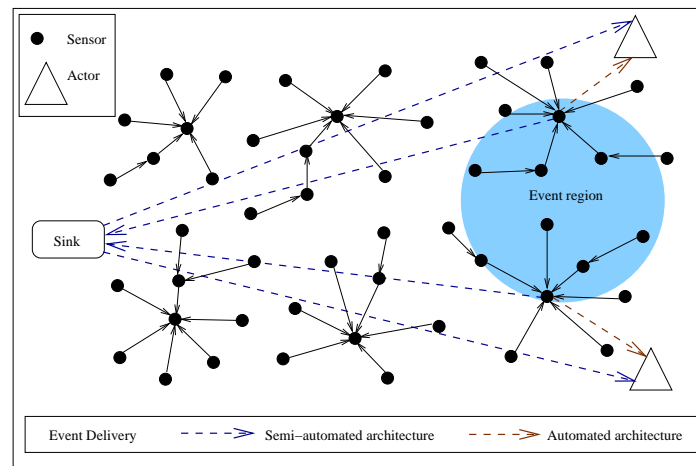


Figure 1.1: Architecture of WSAN

Regardless of the applications, the common feature of all WSANs is that central entity which performs the functions of data collection and coordination may not be necessary [2]. Hence, unlike wireless sensor networks (WSNs) where the communication takes place between sensors and the sink, in WSANs, new networking phenomena called sensor-actor and actor-actor communications may occur. Sensor-actor communication provides the transmission of event features from sensors to actors. After receiving event information, actors need to communicate with each other in order to perform the appropriate action on the event area. However, so as to provide effective acting, both sensor-actor and actor-actor communications should be based on the coordination of sensors and actors. Furthermore,

due to the following unique characteristics of WSANs and its dissimilarities with respect to WSNs, new challenges exist to address efficient communication requirements among sensors and actors in WSANs.

- **Nodes heterogeneity:** WSAN is composed of heterogeneous nodes where the sensor nodes are energy constrained stationary devices as compared to the mobile energy rich actor nodes.
- **Real-time requirement:** The issue of real-time communication is more important in WSANs since actions are performed on the environment after the sensing occur. The right action can not be performed unless the event report is made available to the actor nodes within certain time period.
- **Coordination:** The main communication paradigm in WSANs is sensor-actor as opposed to sensor-sink in WSNs. Furthermore due to the existence of multiple actors, multiple destinations are possible which makes the life of sensor nodes more difficult. Sensor nodes need to coordinate with the multiple actors to select the appropriate actor (possibly single destination among multiple).
- **Actor's mobility:** In most of WSANs applications, actors are presumably mobile. That is, the destination may change its position during the course of the environment monitoring. Hence, requiring the routing protocol to repair the route in order to successfully relay the event data.

There is the need to tackle the issues inherited by WSAN due to its recent development apart from the challenges faced in WSNs. These issues are discussed in detail in Section 1.1. Thereafter, we provide a brief overview of the existing studies in this domain and highlight the contributions of our framework in order to address these issues in Section 1.2. A short overview of the proposed framework and operations of the components incorporated in it are summarized in Section 1.3.

## 1.1 Design Challenges

Several design challenges present themselves to the designers of wireless sensor and actor network applications. The limited resources available to individual sensor nodes implies designers must develop highly distributed, fault-tolerant, and energy efficient applications in a small memory-footprint. Consider the latest-generation MICAz [103] sensor node. MICAz motes are equipped with an Atmel128L [104] processor capable of a maximum throughput of 8 millions of instructions per second (MIPS) when operating at 8 MHz. It also features an IEEE 802.15.4/Zigbee compliant RF transceiver, operating in the 916 MHz/2.4 GHz globally compatible industrial scientific medical (ISM) band, a direct spread-spectrum radio resistant to RF interference, and a 38 kbps data transfer rate. The MICAz runs on TinyOS [105] and is compatible with existing sensor boards that are easily mounted onto the mote. A partial list of specifications given by the manufacturers of the MICA mote is presented in Table 1.1.

Table 1.1: MICA Mote Specification

CPU Speed	8 MHz
Program Flash	128K bytes
Measurement Flash	512K bytes
SRAM	4k bytes
RF	916 MHz/ 2.4 GHz (ISM)
Data Rate	38.4 Kbits/sec
Range	1000 ft
Tx Power	25 mA
Rx Power	8 mA
Sleep Power	< 1 $\mu$ A
Battery	2 x AA (1 year each)

Due to low energy, bandwidth and computation, WSANs inherits various issues in establishing the network and functioning effectively and efficiently.

1. The most essential requirement of applications in WSANs at any level of communication is the energy conservation. Due to unattended mode of sensor nodes, sensor nodes are required to configure efficiently. Henceforth, with the collaboration of

sensor nodes or sensor-sensor coordination, the event data can be efficiently reported to the interested clients.

2. In WSANs, sensors are closely coupled to the physical world and, therefore, they must be made aware of their geographical position to represent the area under observation. Such information is mandatory for actors to actively react in the region of interest. Intuitively, the validity of event readings highly depends on the exact location information of the events that must be reported along with the event features. In this context, the location accuracy of sensor nodes is locally essential with respect to the actors. In order to make sensor and actor networks operate effectively, a low-cost and accurate localization algorithm is imperative. Actors are usually location-aware nodes equipped with some localization hardware. However, the location of sensor nodes is unknown and is an essential requirement in WSANs.
3. In addition to energy constraints as in WSN, WSAN imposes timing constraints in the form of end-to-end deadlines. Therefore, in-time packet delivery is the first and foremost goal of the routing protocols in WSANs since actors need in-time event information to perform appropriate actions. A real-time routing protocol is indispensable that would route data through the lower-delay paths.
4. In order to reduce the communication delay, event reporting policy adopted in WSNs should be modified such that packets are routed directly to the actors rather than the sink node. That is, the protocol needs to implement automated architecture in order to reduce the packet delay which is higher in semi-automated architecture due to the involvement of the sink node. However, this requires sensor-actor coordination so that the event is reported to the right actor capable of responding to it.
5. An actor receiving sensor readings may not always be the right choice to respond. It is either due to inaccessible region or overlapping action coverage of two or more actors. The former case is possible because actors have specific action range and

actor-actor coordination is required to relocate one of the actors toward such regions. In the latter case, actor-actor coordination is required in order to balance the load or selecting the actor which can best perform in overlapping region. Intuitively, actor-actor coordination is imperative for the reliable and efficient response to the event.

6. In WSANs, nodes, especially actors can be mobile. For example, robots used in the distributed robotics application or soldiers which are equipped with data transceivers in a battlefield are usually mobile. Therefore, protocols developed for WSANs should support the mobility of actor nodes.
7. WSANs are mostly designed to monitor and respond in various hostile environments. In order to increase the reliability of applications in such environments, sensors are deployed densely in the field. However, the dense deployment results in huge volume of traffic and creates hot spots in the network because the event data is synchronous by nature triggering all the nodes at once. The raw sensed data is typically forwarded to a sink or actors for processing which contains redundant event reports and unnecessarily consumes the scarce energy resource of the sensor nodes. An important energy saving mechanism for sensor nodes is to exploit in-network data aggregation [46, 47, 48]. The main idea of in-network data aggregation is to eliminate unnecessary packet transmission by filtering out redundant sensor data and/or by performing an incremental assessment of the semantic of the data, e.g. picking the maximum temperature reading.
8. Although aggregation is a useful energy conserving approach, it provides estimated results that may not be appropriate for applications demanding intolerable distortion in the event readings. Therefore, applications may require a certain degree of redundancy, which is not possible with aggregation approach due to its inflexibility of controlling the redundancy dynamically. Furthermore, all the nodes in network

remain active unnecessarily participating in observing the phenomenon that may be otherwise avoided by exploiting spatial correlation. Hence, an alternative approach in this context is to exploit spatial correlation deactivating some of the nodes generating redundant information according to information reliability. This decreases the number of transmissions and increases the lifetime of the network.

There have been considerable efforts to solve the issues partially like nodes configuration and routing problem in wireless sensor networks [34, 53, 54, 56, 58, 59, 60]. However, these protocols do not consider the heterogeneity of WSN. Moreover, none of these protocols provide sensor-actor/actor-actor coordination. Recently, coordination mechanisms [49, 50, 51, 65] are proposed for WSN. The framework in [49] is an event-based reactive model of clustering. Cluster formation is triggered by an event so that clusters are created on-the-fly. The in-time packet delivery in terms of reliability is operated by the actor nodes. Therefore, sensor nodes react slowly to late traffic waiting for the feedback from the actor nodes in order to speed-up the delivery. Hence, the coordination framework is not suitable for time-critical events. Moreover, cluster to actor routing is done using greedy geographical approach. A packet forwarding node finds the next hop node according to the greedy approach failing to do so results into a packet loss as the packets enters into a void region. Since the work assumes that the network is dense therefore it does not propose any void region prevention or recovery mechanism.

In [50], the sensor field is divided into maps, where each map is represented by a sensor node which detects an event the earliest. For building a map, nodes which have detected an event but not reported yet, flood the event detection message. It applies aggregation hierarchically in the map and the representative node of the map collects data from all the nodes and reports it to the actuator. This is also an event-driven nodes configuration and aggregation that introduce high latency. This approach becomes more inefficient in terms of delay and energy when the event center moves frequently that requires rebuilding maps in the field. Moreover, it does not provide any mechanism to route data according

to the specific response time i.e. no support of real-time routing. Delay energy-aware routing (DEAP) [65] is a geographical-based routing protocol proposed for WSNs that conserves the energy of nodes by an adaptive energy management scheme. It switches the node to sleep mode if the queue is empty and back to active mode when the packets are buffered. It extends the nodes active duration when there are buffered packets to avoid from latency. However, there is no explicit in-time delivery model to support real-time traffic and also does not provide any actor-actor coordination mechanism. Consequently, there exists no unified solution which addresses the real-time routing, sensor-actor/actor-actor coordination for heterogeneous WSNs.

## 1.2 Our Contributions

To address the above issues, we present energy-efficient real-time coordination and routing (ERCR) framework which provides an integrated solution comprising of sensors localization, sensor-sensor, sensor-actor and actor-actor coordination as well as real-time aggregation and routing in the *semi-automated architecture* and *automated architecture*. The contributions of the framework are summarized as follow

**Localization of Sensor Nodes:** A great deal of solutions [4]-[26] are proposed for solving localization problem in wireless sensor networks. However, a unified solution for sensors and actors is not addressed in these studies. Recently, a localization protocol [3] is presented for WSNs, which uses RSSI to calculate distance between beaconing actors and sensors. Clearly, the RSSI-based approaches are highly hostile to external environment due to multipath reflections, non line-of-sight conditions, and other shadowing effects that leads to erroneous distance estimates [21]. In order to localize the sensor nodes relative to actors, we present an efficient Timing-based Sensor Localization (TSL) algorithm for WSN. In TSL, sensor nodes determine their distance from actors by using propagation time and speed of RF signal. The heuristic of TSL is that it computes



the propagation time following some timing patterns as described in Chapter 2 without synchronizing the sensor nodes. Moreover, it is adaptive to the mobility of sensor nodes and can be configured according to the required localization accuracy in order to avoid the overheads raised due to high velocity.

**Sensor-sensor coordination:** In order to configure the heterogeneous sensor nodes, framework provides a cluster-based configuration so called a multi-event adaptive clustering protocol (MEAC) in which sensor nodes coordinate with each other to form clusters. The details of the operations of MEAC are given in Chapter 3. Among a group of nodes, a node is elected as a cluster-head, which has higher energy level but lower traffic rate. Generally, the clustering protocols [34, 36] focus on the currently available energy of the nodes and periodically reorganize clusters to do energy balancing. However, this strategy is not practical when the nodes are sending traffic at different rates due to different events characteristics. If all the nodes have the same probability to become cluster heads then the nodes reporting events at higher rate will eventually loose their energy earlier than the others resulting into smaller network lifetime. Therefore, the load on a node, that is, its data rate is used as a key factor for cluster-head selection. Hence, MEAC distributes the energy usage of nodes by adapting to the residual energy and multiple events in the field in order to increase the stable period of the network.

**Real-time routing:** Framework achieves the event specific delay bound  $\tau$  of packets through the delay-constrained energy aware routing (DEAR) protocol. The value of  $\tau$  may differ for different events. The DEAR protocol, described in Section 4.2, exploits the hierarchical cluster-based configuration of nodes to relay the packets. The selection of forwarding node is based on the packet delay as well as the balance energy consumption of sensor nodes. The routing decisions in DEAR are not based on the geographical location of the forwarding nodes as exploited in greedy mode routing protocols [49, 50, 65]. The greedy mode routing protocol becomes very cumbersome and incurs extra overheads for irregular or non-uniform deployment of sensor nodes requiring recovery mode operations

in void regions. The proposed solutions [49, 50, 65] even do not provide any recovery algorithm. Moreover, in ERCR, cluster-heads take localized decision about the in-time packet delivery unlike the coordination framework [49] in which sensor nodes wait for the actors to speed up the routing. Hence, DEAR responds to the late traffic locally and does not suffer in void region problem.

**Sensor-actor coordination:** We assume that the sensor nodes have *a priori* knowledge of the actors action range. Action range represents the area in which an actor can reliably execute actions. It is assumed to be same for all the actors. In ERCR, only the cluster-heads are responsible of coordinating with the actor nodes. A cluster-head selects an actor for coordination which can react in the clustered region and is attending least number of sensor nodes as explained in Section 4.3. Although this seems a very simple coordination solution but it is realistic and practical.

**Data aggregation:** The main idea of in-network data aggregation is to eliminate unnecessary packet transmission by filtering out redundant sensor data to conserve energy. It is argued that aggregation extends the queuing delay at the relaying nodes and becomes more challenging due to coexistence of mixed-type traffic resulting from concurrent events. ERCR incorporates a very simple yet practical real-time data aggregation (RDA) method. In RDA, only cluster-heads aggregate data from their member nodes and keep aggregating data as long as the end-to-end delay constraint is not violated as described in Section 4.4. Moreover, it does not require building aggregation trees [46, 47, 48] since the nodes have already been configured in the form of clusters. It is worthwhile to note that RDA achieves fairness implicitly since nearer cluster-heads have more time for aggregation due to shorter path delay and thereby reduce the traffic significantly near the destination.

**Actor-actor coordination:** ERCR considers two scenarios in order to trigger actor-actor coordination; overlapping region and inaccessible region. Generally, the existing studies [49, 50] consider only the first scenario and provide an optimal solution to react in

overlapping region. They have not discussed actor-actor coordination from the perspective of inaccessible region. However, ERCR moves an actor receiving request toward the inaccessible region if possible or trigger actor-actor coordination to relocate some other actor in that region. Hence, in ERCR, actors initiate A-A coordination at both occasions that enables the actors to execute certain actions no matter the reporting region is in their range or not.

**Integrated Pull/Push Coordination:** In order to overcome the inefficiencies of pure pull or pure push data dissemination models, an integrated Pull/Push Coordination approach is incorporated in the framework. Hybrid Push/Pull coordination is studied in [64, 70, 71]. These studies are proposed for monitoring and control purposes in WSN that has a single sink. However, in WSAN, an efficient coordination mechanism is required to overcome the mobility of multiple destination actors. Hence, the framework also realizes the mobility of actors in IPP in addition to the multiple destination actors in WSAN.

**Controlling Data Redundancy:** Spatial correlation of sensor nodes has been exploited in the literature [75]- [84] to eliminate data redundancy for energy conservation. The field is divided into correlation regions and a single node called a representative node is selected in each correlation region to report the event readings. However, the existing approaches do not consider the residual energy of nodes in selecting the representative nodes and therefore do not efficiently exploit the spatial correlation. In this framework, we propose an energy-aware spatial correlation of sensor nodes based on the clustering protocol as given in Chapter 6. Cluster-heads exploit spatial correlation in their regions independently to keep the member nodes active subject to the information reliability and balanced energy consumption of correlated nodes.

## 1.3 Overview of ERCR Framework

The architecture of the framework is shown in Fig. 1.2. The first and foremost job of the sensor nodes is to get their spatial position which is essential for their identification. This is obtained by the timing-based sensors localization (TSL) protocol at the lower level of the framework. Henceforth, sensors are ready to sense the phenomenon and put the location information in the event reports to represent the environment. At next level, the framework offers two different novel approaches for relaying data; routing by adaptive targeting (RAT) that works in greedy mode and delay-constrained energy-aware routing (DEAR) which employs addressing (node ID) mode routing. Greedy routing is generally encouraged [49, 58, 65] because it follows the shortest possible path and in turn conserves the nodes energy. However, greedy mode routing is inefficient when the deployment is non-uniform and there exist holes in the greedy routes.

RAT incorporates an integrated pull-push (IPP) data delivery mechanism in which actors publish their interest while sensors subscribe the actors to coordinate with them. It is important to note that each sensor needs to coordinate with the actors independently. On the other hand, RCR employs DEAR as a routing protocol and configures the sensor nodes in the form of clusters using multi-event adaptive clustering (MEAC) protocol. Cluster-heads then perform aggregation on the data received from their members, coordinate with the actors and make the routing decision themselves by taking into account the event-specific deadline. Eventually, it conserves the energy of member nodes and extends the life of entire network. At top of these layers, energy-aware spatial correlation of sensor nodes is exploited in order to control the number of originating packets. In the rest of this section, we briefly describe the operations of each of the components.

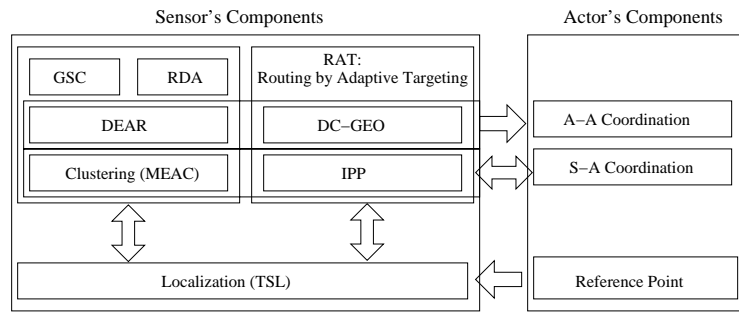


Figure 1.2: Architecture of ERCR framework showing the interaction of components incorporated.

### 1.3.1 Timing-based Sensors Localization (TSL)

TSL localizes the sensor nodes with respect to actors in order to avoid the independent localization errors of sensors and actors. Actors are location-aware nodes equipped with some localization hardware. However, sensor nodes are localized with reference to actors. Sensor nodes determine their distance from the actors by using propagation time and speed of RF signal. Actors actively broadcast *reference pulses* in a pattern of intervals defined according to the mobility of sensor nodes and the required level of localization accuracy as described in Section 2.3.2. These reference pulses carry the interval numbers in which they were transmitted. The interval numbers are then exploited by the sensor nodes to calculate the start time of the pulses locally. The locally estimated start time is then used to determine the propagation time.

### 1.3.2 Self-configuring Clustering Protocol: MEAC

MEAC achieves energy efficiency by considering three design factors; (1) limiting the number of clusters ( $k$ ) in the network, (2) electing an appropriate node to function as a cluster-head, and (3) reducing the frequency of clusters reformation. To limit the number of clusters, it calculates the optimal number of clusters by using the given topology information i.e. dimension of the field, number of sensor nodes and their transmission radius. Nodes are then restricted to form approximately  $k$  number of clusters. Second,

cluster formation is based on the weighting equation which includes the nodes energy, data rate and density of nodes. These parameters are given weight according to the application needs as explained in Section 3.3.3. Nodes exchange their aggregated weights with the neighbors and a node announces itself as a cluster-head subject to the highest aggregated weight. Hence, MEAC distributes the energy usage of nodes by adapting to the different energy levels of nodes as well as multiple events (data rates). Moreover, cluster formation is not triggered periodically rather it defines threshold on weight and a cluster-head is withdrawn only if its weight goes down to threshold.

### **1.3.3 Delay-constrained Energy-Aware Routing (DEAR)**

ERCR incorporates DEAR protocol to deliver packets from the source clusters to the target nodes (Sink/Actors) within the given time constraint. DEAR provides real-time routing support that works on top of the clustering protocol MEAC. It establishes a backbone network by integrating the forward tracking and backtracking mechanism. Path from single/multiple hop members to cluster-heads is established during cluster formation. While the route from cluster-heads to destination sink/actor is initiated by the destination nodes in backtracking manner as given in Section 4.3. Once routes are established, event flows from cluster-heads to cluster-heads through some intermediate *gateway* nodes. The backbone provides the cluster-heads all the potential routes toward the destination sink or actor through the intermediate cluster-heads. The *gateway* selection criterion is then based on the packet deadline with efficient energy consumption. An energy efficient *gateway* does not merely mean the low cost link but a link that can satisfy the delay constraint and it balances the energy consumption on all the outgoing links. ERCR supports both the *semi-automated* as well as *automated architecture* of WSANs. In the presence of the sink, it adapts to the centralized version of DEAR (S-DEAR) to coordinate with the actors through the sink if required. On the other hand, when there is no sink or ignoring its presence, it provides the distributed version of DEAR (A-DEAR) for coordination among

sensors and actors.

### 1.3.4 Sensor-Actor/Actor-Actor Coordination

In order to initiate sensor-actor coordination in ERCR, the mobile actors, periodically or upon location change, broadcast S-A Coordination information (*SACInfo*) beacon. The *SACInfo beacon* contains the location of the actor, number of sensors being attended and energy level. ERCR permits only cluster-heads to coordinate with the actor nodes. A cluster-head selects an actor for coordination which can react in the clustered region and is attending least number of sensor nodes. As a result, a dynamic *responsibility clusters* are formed in the event region. *In a responsibility cluster, cluster-head collects data from their members and route to the actor responsible of responding in their regions.* Since, the decision of coordination is based mainly on the location of actors, therefore the coordination is revised if an actor moves. That's why, actors broadcast *SACInfo beacon* upon location change in addition to the periodical broadcast.

In actor-actor coordination, ERCR considers two scenarios; first, actors are insufficient in the field to cover the whole region and second, the operating actors are more than required resulting into overlapping action region. In case of inaccessible region, an actor receiving request either moves itself toward such region if possible or trigger actor-actor coordination to relocate some actor in that region. This movement is possible only if the moving actor can attend its current request as well as the request from the inaccessible region. In case of overlapping region, an actor receiving request might be busy in responding to other requests or have shortage of resources. Therefore, it initiates actor-actor coordination to meet the event response time as well as increasing the reliability of applications.

### **1.3.5 Routing by Adaptive Targeting (RAT)**

Generally, when sensors detect an event (a change in the environment), they communicate with each other to confirm the event or generate new data based on the event, which is reported to the information collector (sink/actor). There are two ways to disseminate the event data, namely push-based and pull-based or querying approach. Using push policy, event information is sent out to the sink without explicit requests. In contrast, with pull-based approach events are reported in response to explicit requests received from sink/actors. In our study we refer to the communication initiated by the actors to disseminate their location and their acting capabilities as pull communication.

The geographical-based routing by adaptive targeting (RAT) protocol allows sensor-to-actor communication and dynamic coordination of actors in response to emergencies. RAT comprises two component; Delay-constrained geographical-based routing (DC-GEO) and Integrated Pull/Push (IPP) coordination presented in Section 5.3. DC-GEO relays the packets in greedy mode such that delay constraint can be met as well as energy consumption of forwarding nodes is balanced. In IPP, actor nodes subscribe to specific events of their interest in the field and sensor nodes disseminate the event readings to subscribed actor for a time period of subscription life. As a result, actor nodes do not require sending a query every time they need event readings. Similarly, sensor nodes push the data as long as there is a subscribed actor interested in the observed event.

### **1.3.6 Real-time Data Aggregation (RDA)**

The aggregation approach implemented by ERCR is dynamic in nature and adapts according to the traffic conditions. RDA performs aggregation only at the cluster-heads. Cluster-heads, knowing the communication delay of the destination actors obtained through DEAR protocol, keep aggregating data from their member nodes only during the auxiliary time. This is the fraction of time available for which the communication delay is lesser than the response time as described in detail in Section 4.4. Intuitively, nearer cluster-head



experiencing lower routing delay gets more auxiliary time performing more aggregation as compared to the farther cluster-heads. This avoids the channel capturing of nearer nodes and leaves the bandwidth for the other nodes. Hence, RDA does not affect the packets deadlines and implicitly achieves fairness.

### **1.3.7 Exploiting Energy-aware Spatial Correlation**

The framework proposes gridiron mechanism exploiting the spatial correlation (GSC). GSC is based on the clustering protocol in which cluster-heads are responsible of exploiting spatial correlation of their member nodes and selecting the appropriate member nodes to remain active for observing the phenomenon. This mechanism is based on the distortion tolerance and the energy of member nodes. The width of the correlation region is calculated by the sink/actor according to the current event reports received from the sensor nodes and information reliability. This value is then propagated to all the cluster-head to adjust the number of active nodes accordingly. Each cluster-head divides its clustered region into correlation regions and selects a representative node in each correlation region, which is closer to the center of correlation region and has higher residual energy. Hence, the whole field is efficiently represented by a subset of active nodes which perform the task well equal to that of the all deployed nodes.

## **1.4 Energy Model**

In our energy model, we take into account the energy consumed in transmitting data while ignoring the other consumption factors like processing and sleep mode. So the study always focuses on energy conservation in terms of minimizing the transmission energy to consider the efficient solution . It is due to the fact that the energy expenditure in data processing is much less compared to data communication. The example described in [95], effectively illustrates this disparity. Assuming Rayleigh fading and fourth power

Operation	Energy Dissipated
Transmitter/Receiver Electronics ( $E_{elec}$ )	50nJ/bit
Data Aggregation ( $E_{DA}$ )	5nJ/bit/report
Transmit Amplifier ( $E_{amp}$ )	10pJ/bit/ $m^2$

Table 1.2: Radio characteristics used in simulation model.

distance loss, the energy cost of transmitting 1 *KB* a distance of 100 *m* is approximately the same as that for executing 3 million instructions by a 100 million instructions per second (*MIPS*)/*W* processor [1].

An accurate model for transmission energy consumption per bit at the physical layer is  $E = 2E_{elec} + \beta d^\alpha$ , where  $E_{elec}$  is a distance independent term that takes into account overheads of transmitter/receiver electronics, while  $\beta d^\alpha$  accounts for the radiated power necessary to transmit one bit over a distance  $d$  between source and destination.  $\alpha$  is the exponent of the path loss ( $2 \leq \alpha \leq 5$ ) and  $\beta$  is a constant [*Joule/(bits.l $^\alpha$ )*]. We use minimum transmission energy (MTE) routing approach which requires multihop routing protocols.

## 1.5 Simulations

The emergence of wireless sensor networks brought many open issues to network designers. Traditionally, the three main techniques for analyzing the performance of wired and wireless networks are; analytical methods, computer simulation, and physical measurement. However, because of energy limitation of sensor networks, decentralized collaboration, fault tolerance, algorithms for sensor networks tend to be quite complex and usually defy analytical methods that have been proved to be fairly effective for traditional networks. Furthermore, few sensor networks have come into existence, for there are still many unsolved research problems, so measurement is virtually impossible. It appears that simulation is the only feasible approach to the quantitative analysis of sensor networks.

*ns-2* [106] is an open-source simulation tool that runs on Linux. It is a discreet event

simulator targeted at networking research and provides substantial support for simulation of routing, multicast protocols and IP protocols, such as UDP, TCP, RTP and SRM over wired and wireless (local and satellite) networks. It has many advantages that make it a useful tool, such as support for multiple protocols and the capability of graphically detailing network traffic. Additionally, *ns-2* supports several algorithms in routing and queuing. LAN routing and broadcasts are part of routing algorithms. Queuing algorithms include fair queuing, deficit round-robin and FIFO. NS2 is available on several platforms such as FreeBSD, Linux, SunOS and Solaris [108]. NS2 also builds and runs under Windows.

We implemented all the protocols in *ns-2.26* version running over Fedora Core 2 Linux system. When simulating the network environment, scenario based simulations will be used. Each of these scenarios will have different variables aside from node number, including the following:

- Simulation area (meters)
- Node density (nodes/m<sup>2</sup>)
- Movement of nodes (m/s<sup>2</sup>)

Considering the above variables, some of the measurables that shall be plotted include:

- Energy consumption of sensor nodes
- Location accuracy
- Delay
- Packet deadline miss-ratio
- Throughput

## 1.6 Thesis Organization

In Chapter 2, a novel timing-based sensor localization algorithm is presented which localizes the mobile sensor nodes with reference to the actor nodes. TSL is adaptive to the mobility of sensor nodes and can be tuned to achieve the desired accuracy.

The multi-event adaptive clustering (MEAC) protocol is presented in chapter 3 before the RCR protocol since it provides the basis for the energy efficient real-time coordination and routing protocol. The MEAC protocol is more adaptive to the heterogeneity parameters of WSNs such as energy and multiple event reporting rates and significantly improves the network lifetime.

Chapter 4 describes the design and implementation of real-time coordination and routing protocol. It includes the sensor-actor coordination, delay-constrained energy aware routing for in-time data delivery, real-time data aggregation and actor-actor coordination.

Chapter 5 presents the geographical based routing protocol RAT which relays the packets in greedy mode while considering their deadlines as well. A heuristic integrated pull/push data delivery model is also incorporated to provide energy efficient sensor-actor coordination. An adaptive targeting solution is also included in RAT in order to attend the emergency in the field through actor-actor coordination.

An energy conserving scheme is presented in Chapter 6 which exploits the spatial correlation of densely deployed sensor nodes. It is also based on the clustering protocol MEAC in which cluster-heads select appropriate representative nodes for sensing the whole field while disabling the activity of others according to the required distortion tolerance.

Finally, Chapter 7, summarizes the thesis and draw conclusions of our achievements and provides inroad to further investigate the related issues in this research area.

## CHAPTER 2

### TIMING-BASED SENSOR LOCALIZATION (TSL) IN WSANs

#### 2.1 Introduction

The emergence of wireless sensor actor networks raises many system design challenges since sensors are closely coupled to the physical world; they must be made aware of their geographical position to represent the area under observation. Such information is mandatory for actors to actively respond to the region of interest. A large variety of applications in WSANs ranging from military to health have been emerged where the location information is considered vital along with the event readings. Actor nodes require the exact location of the sources in order to identify the location where an event originates and effectively act upon the event. For instance, in case of fire, sensors relay the exact origin and intensity of the fire to water sprinkler actors so that the fire can easily be extinguished before being spread uncontrollable. By knowing the exact location of the explosion of gases in mines, the miners can be evacuated before the collapse of mines. In health applications, location awareness facilitates doctors with the information of nearby medical equipments and personnel in a smart hospital.

In order to make sensor and actor networks operate effectively, a low-cost and accurate localization algorithm is imperative. Actors are usually location-aware nodes equipped with some localization hardware. However, the location of sensor nodes is unknown and is an essential requirement in WSANs. The size and cost factors mainly preclude the

reliance of the mobile sensor nodes on GPS receivers in a network comprising of nodes in the order of hundreds to thousands or even more [20]. On the other hand, GPS-free localization algorithms for WSNs [11], [14], [26] also exist in the literature which are estimation-based approaches localizing the nodes more imprecisely. As a result, localizing sensors and actors independently, the individual location errors in sensors and actors are accumulated to large inaccuracy, which severely hampers the application performance.

Furthermore, most of the existing solutions do not consider the mobility of sensor nodes and thereby are inappropriate for localizing the mobile sensor nodes in WSANs. Localization of mobile sensor nodes is studied in [16], [17]. In [16], some additional seed nodes are deployed to localize the sensor nodes where the current position of mobile node is predicted on the basis of previous observations. However, the accuracy highly depends on the density and velocity of seed nodes. This is another estimation-based approach among a long list of statistical approaches which do not provide the degree of accuracy needed for action precision in WSANs. The results provided in this study reveal that it has intolerable location errors, i.e., above 20% of transmission radius of nodes. Consider a war scenario where a fire gun requires the precision of less than a half meter if the target is the soldier of red forces and precision of a few meters is required in order to hit the tank of enemy forces.

A great deal of solutions [4]-[26] are proposed for solving localization problem in wireless sensor networks. However, a unified solution for sensors and actors is not addressed in these studies. Recently, a localization protocol [3] is presented for WSANs which uses RSSI to calculate distance between beaconing actors and sensors. Clearly, the RSSI-based approaches are highly hostile to external environment due to multipath reflections, non line-of-sight conditions, and other shadowing effects that lead to erroneous distance estimates [21].

In this study, we propose a precise Timing-based Sensor Localization (TSL) as a reliable, accurate and cost-effective algorithm that neither requires extra hardware (GPS/ultrasonic

receivers or anchor nodes) [9], [18], [19], [20], [21] nor uses estimation approach [11], [14], [26] and localizes the sensors with respect to actors. The detailed overview of the approach is provided in Section 2.3.1.

## **2.2 Related Work**

The existing localization solutions in the literature can be categorized into two groups; range-based and range-free techniques. In this section, we investigate the various approaches adopted by these solutions and their drawbacks.

### **2.2.1 Range-based Localization**

In range-based localization solutions, the absolute distance (range) between a reference node and a localizing node can be estimated by using one of these approaches; received signal strength, the time-of-flight of communication signal and angle of arrival of the received signal. However, the accuracy of such estimation depends on the surrounding environment and usually relies on complex hardware [22].

#### **Received Signal Strength Indicator (RSSI)**

This technique requires the knowledge of transmitter power, the path loss model and the power of the received signal. Based on the known transmit power and the path loss model, the effective propagation loss can be calculated. Theoretical and empirical models [19], [20] are used to translate this loss into a distance estimate. This method has been used mainly for RF signals. The major drawback of this method is that multipath reflections, non line-of-sight conditions, and other shadowing effects might lead to erroneous distance estimates [21].

### **Timing-based methods (ToA, TDoA)**

This approach records the time-of-arrival (ToA) i.e. propagation time of the beaconing signal [9], [18] or Time-Difference-of-Arrival (TDoA) [24] of two signals that might be RF-signal and another signal of low propagation speed from a set of reference points. The propagation time can be directly translated into distance, based on the known signal propagation speed (speed of light). A timing-based position scheme (TPS) [24] is proposed that takes the TDOA of two RF beacons and apply trilateration to compute the intersection of three circles of beaconing stations. However, due to the high propagation speed of wireless signals, a small measurement error causes a large error in the distance estimate [21]. Hence, for densely deployed sensor nodes, localization techniques using TDoA measurements need to use a signal that has a smaller propagation speed than wireless, such as ultrasound [22]. Although this gives fairly accurate results, it requires additional hardware at the sensor nodes to receive the ultrasound signals and make it range dependent due to the limitation of ultrasound transmission.

### **Angle-of-Arrival (AoA)**

Special antenna (omni-directional) configurations are used to estimate the angle of arrival at which signals are received from a beacon node and use simple geometric relationships to calculate node positions [21]. The main drawback of this technique for terrestrial systems is the possibility of error in estimating the directions caused by multi-path reflections.

## **2.2.2 Range-free Localization**

To overcome the limitations of the range-based localization schemes, many range-free solutions [4], [5], [6], [7], [11], [14], [25] [26] have been studied. These solutions estimate the location of sensor nodes by, either, exploiting the radio connectivity informa-



tion among neighboring nodes, or applying some probabilistic approach. Among existing range-free localization approaches, Comparison of Received Signal Strength Indicator (ROCRSSI) [4] uses ring overlapping to estimate nodes location. The rings can be generated by comparison of the signal strength a sensor node receives from a specific anchor and the signal strength other anchors receive from the same anchor. Then it calculates the intersection area of these rings (or circles) and takes the gravity of this area as its estimated location. The ring overlapping, compared to triangle-overlapping in APIT [5] generates small intersection area and results in more accurate location estimation. The accuracy of APIT depends on the number of anchors, where by utilizing the combinations of anchor positions, the diameter of the estimated area in which a node resides can be reduced, to provide a good location estimate. However, the correctness of ROCRSSI is based on the assumption that in a certain range of direction, with the increase of distance between a sender and a receiver, the signal strength at the receiver decreases monotonically. It is highly dependent on the surrounding environment and density of nodes that may lead to inaccurate distance estimates [8], [21].

In mobile-assisted localization [6], a roving human or robot wanders through an area, collecting distance information between the nodes and itself. The challenge is to design movement strategies that produce a globally rigid structure of known distances among the static nodes. The pairwise node distances resulting from this strategy can then be fed into a Anchor-Free Localization (AFL), which computes an initial coordinate assignment to all the nodes, using the radio connectivity information alone. This approach is highly dependent on the movement of mobile and degree of network and produce inaccurate results in a field where the movement of mobile is restricted and the deployment of sensor nodes is non-uniform.

A statistical beacon-less approach [7] assumes that sensors are deployed in the form of groups and all the groups are deployed at different known regions. A node can estimate its position by observing group membership of its neighbors. The limitation of this approach

is the accuracy and deployment of sensor nodes in the form of groups (possibly manually) that may not be applied in some applications. Another range-free solution DV-hop [26] was proposed that exploits the connectivity of the nodes and determine the distance from anchor nodes by multiplying the hop count with an average hop distance. A drawback of DV-hop is that it fails for highly irregular network topologies, where the variance in actual hop distances is very large [13].

Hence, the above localization schemes have been proposed in WSNs which are not well suited for WSANs. WSNs has been widely used for monitoring the environment and the accuracy of localization can be compromised to some extent. In contrary, the exact location of the sensor nodes in WSANs assist the actors in executing precise actions. Intuitively, the location accuracy of sensor nodes is indispensable with respect to actors. Localizing the sensors and actors independently produce large positioning errors causing imprecise actions; thereby minimizing the effectiveness of actors in the field. Secondly, using actors as mobile beacons do not require to embed the additional anchor nodes or special mobile beaconing nodes as reference points and the impact of inaccurate location measurement of actors becomes null. Thirdly, the mobility of sensor nodes necessitates the need of an adaptive unified localization algorithm for WSANs that can be fine tuned to eliminate the errors caused by the mobility.

## **2.3 TSL Approach**

In this section, we present timing-based sensor localization (TSL) algorithm for WSANs. It exploits the actor nodes, inherently equipped with localization mechanisms such as GPS, as mobile beaconing reference stations. Both the sensors and actors are assumed to be mobile and at least four actor nodes are assumed to exist in the field that use long-range transmission such that all the sensor nodes receive the transmission as illustrated in Fig 2.1. TSL is also adaptive to different velocity of mobile nodes and can be configured

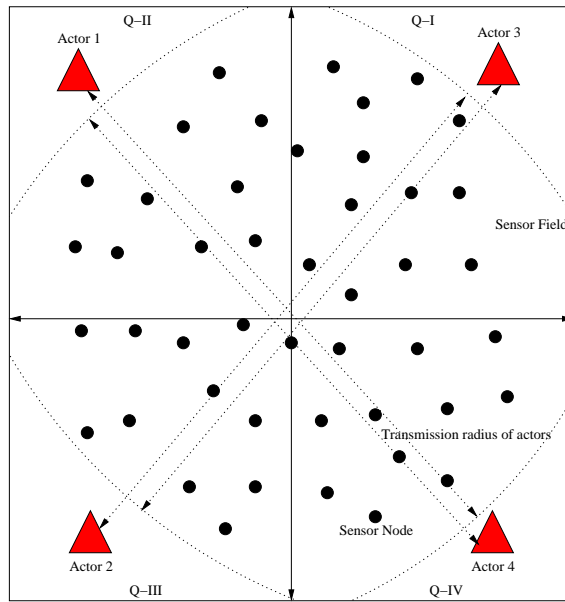


Figure 2.1: Architecture of TSL

according to the required accuracy in order to avoid the overheads raised due to high velocity. Hence, the objective of the protocol is to localize the mobile sensor nodes relative to actors with least energy consumption and achieve high accuracy by limiting the distance errors within tolerable limits.

### 2.3.1 Overview of the Algorithm

In TSL, sensor nodes determine their distance from the actors by measuring the propagation time of the *reference pulses* which are actively broadcast by the actors. Usually the propagation time is measured by employing either round-trip-time mechanism [27] or establishing time base (synchronizing the nodes) [9], [18]. However, the heuristic of TSL is the local measurement of the pulse delay at the localizing node without involving the reference node.

The transmission of pulses is scheduled according to the predefined pattern as described in Section 2.3.2. This pattern is formed by dividing the certain *localization time period* ( $\mathfrak{J}$ ) into discrete time units (fixed number of intervals) set according to the mobility of sensor nodes and required accuracy. The value of  $\mathfrak{J}$  reflects the possible topological

changes either due to the mobility of nodes or incremental deployment of nodes. For network of stationary nodes having no future redeployment of sensor nodes, the localization time can be set to infinity to avoid from the potential overheads. On the other hand,  $\mathfrak{J}$  is set to small value if the expected velocity of nodes is high. In such case, we need to re-localize the mobile nodes by transmitting the pulses in the network. If we aim to keep the distance error less than some tolerable value, it means that a pulse must be received by the sensor node before the traveled distance approaches to the tolerable distance error. Hence, the selection of  $\mathfrak{J}$  plays critical role in achieving the desired accuracy, which is further discussed in Section 2.3.5.

In order to avoid the potential errors in measuring the propagation time possibly caused either due to the synchronization algorithm or clock drifts, the sensor nodes compute the start time of the pulses locally by using the interval information relayed by the actors. This information include; the interval number in which the pulse is transmitted and some lag time elapsed since the pulse was scheduled to its transmission. Hence, making the approach independent to clocks of the actors. Actors send such interval information in the pulses and therefore, sensors can infer the start time to compute the propagation time of the pulses locally that remains almost same until the actor does not move. We take the mean of all the measured values in calculating distance from actor to sensor node. However, if the actor moves then the propagation time of the pulse measured by the sensor node is changed. Therefore, the actor should somehow relay its mobility information so that the sensor node neglects the observations obtained before the actor or its own mobility in order to compute the correct mean value.

A sensor node hearing these pulses calculates distance from at least four actors. Henceforth four overlapping rings are drawn centering at the actors with radii equal to the calculated distances of a node from the respective actors. These rings coincides at the position of the sensor nodes. As a result, we obtain two coinciding lines from the four rings where the intersection point of the lines determine the position of the sensor node. Under some

conditions, it might be possible that all the nodes do not hear the transmission of all the four actors and are unable to determine their location. In such case, we apply the nodes collaborative localization procedure and relax the condition of four actors by considering the localized sensors in place of missing actors to estimate the position of unlocalized nodes and thereby making it range-free solution.

### 2.3.2 Pulse Transmission in TSL

TSL divides the localization time period ( $\mathfrak{J}$ ) into discrete intervals as described in Section 2.3.2 and actors transmit *reference pulses* in each interval which are synchronized with each other to start the intervals at the same time. However, the synchronization error of actors does not effect the measurement of propagation time of the pulses at the sensor nodes. Because sensor nodes wait for the pulses of four actors and then apply the localization algorithm. As a result, a slight delay in pulse transmission of any of the actor does not produce any distance error. If a node receives two pulses from the same actor before receiving the pulses from all the actors, it infers that it is out of the transmission range of some actor(s) and can apply nodes collaborative localization.

#### Pulse Packet Format

The format of the pulse packet containing the required parameters is given in Fig 2.2. The pulse carries the following information which are mandatory for the sensors to calculate the propagation time locally.

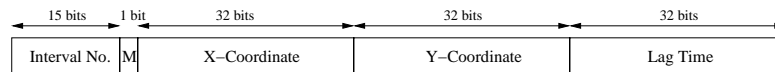


Figure 2.2: Packet format of *reference pulse*

- *Interval No.* ( $\mathfrak{N}$ ): TSL is adaptive to the mobility of nodes that sets the pulse frequency accordingly. To achieve the varying frequency for different applications,

either  $\mathfrak{I}$  can be fixed and divided into different time units (intervals) or the number of intervals are fixed but  $\mathfrak{I}$  is varied i.e. length of intervals is varied. Since the interval numbers are relayed by the actors in pulses and used by the sensors to measure the propagation time. Therefore, we need to fix the number of intervals in localization to reserve space in the packet because it is not possible to change the packet format for each application. Hence, we reserve 15 bits to keep track of the running interval  $\aleph$  of pulses. If we reserve lesser bits in packet then it results in short interval space and require to initialize the intervals sooner. On the other hand, reserving more bits in packet results in extra overheads.

- *M flag*: In order to isolate the errors in timing measurements caused by the external environment, we compute mean of the estimated propagation time of pulses over multiple pulse intervals (sampling pulses) as described in Section 2.3.3. However, the values are considered in the sample until the actors remain stationary. As an actor moves, it propagates the movement information by setting the flag  $M$  in the next scheduled pulse. Sensor nodes ignore the previously measured values in computing mean of the propagation time if the flag  $M$  in the pulse is reported *true*.
- *Lag time*: Although the pulse is scheduled to broadcast at predetermined time, however, there are some factors delaying the start of transmission of the pulse e.g. medium access time and processing time. These factors are accumulated as lag time and actors report this value in the pulse for precisely measuring the time-of-flight of the bacon.
- *Actor Position*: X and Y coordinates of the actor are stamped in the pulse to forms the overlapping rings which are used in position estimation of unknown nodes.

## Timing of Intervals

This section describes the time division approach to define intervals used in TSL. Since the start time of the pulse is estimated by using interval numbers sent by the actors, therefore, synchronization of sensor nodes is not required in this approach. In synchronization-based approaches, a small error of  $1\mu s$  either due to synchronization algorithm or relative clock drifts may produce inaccuracy of approximately 300 meters which is unacceptable in WSANs. Hence, the propagation time is measured by taking the time difference of the start time measured locally at the receiving node and pulse received time.

By reserving 15 bits for intervals in the pulse, the localization period  $\mathfrak{I}$  can be divided into  $2^{15}$  intervals. Therefore, the frequency  $f$  of pulse transmission can be calculated as

$$f = 2^{15}/\mathfrak{I} \quad (2.1)$$

Furthermore, each interval is divided into  $2^{15}$  subintervals to increase the precision of timing information as illustrated in Fig 2.3. Although a pulse is transmitted in an interval, however, it is delayed one subinterval ( $\mathfrak{I}/2^{30}$ ) further to enhance the precision. That is, the transmission does not start at start of each interval rather the subinterval is also shifted linearly with the interval. As a result, the transmission of  $i^{th}$  pulse takes place in the  $i^{th}$  subinterval of  $i^{th}$  interval and  $(i+1)^{th}$  pulse in  $(i+1)^{th}$  subinterval of  $(i+1)^{th}$  interval rather than  $i^{th}$  subinterval. Thus the time period of pulse is  $\mathfrak{I}/(2^{15} + 2^{30})$ . Hence, the timing function  $T(i)$  to determine the start time of  $i^{th}$  pulse is

$$T(i) = (i - 1) \times (\mathfrak{I}/2^{15} + \mathfrak{I}/2^{30}) \quad (2.2)$$

From the above equation,  $1^{st}$  pulse is transmitted in  $0^{th}$  interval at  $T = 0$ ,  $2^{nd}$  pulse in first interval at  $T = \mathfrak{I}/2^{15} + \mathfrak{I}/2^{30}$ , 3rd pulse at  $T = 2 \times (\mathfrak{I}/2^{15} + \mathfrak{I}/2^{30})$  and so on.

Although the pulses are scheduled to be transmitted by each actor at the time calculated above by using (2.2) but it is not possible that all the actors capture the shared

channel at same time for pulse transmission. If the synchronized actors attempt to access the medium, it is high likely that the transmission will collide and result in contention delay which has more severe impacts when the length of interval is short. Consequently, if the nodes could not receive the pulse within the time duration of interval then they either remain unlocalized until next interval or keep incorrect location if moved to new location. Even more specifically, they may start the collaborative node localization procedure unnecessarily and energy consumption in localization is increased.

Therefore, MAC contention delay is an important factor when exploiting the propagation delay that causes inaccuracy in distance calculations. To avoid from the possible collision, we apply time division multiple access (TDMA) schedule to actor nodes for transmission of *reference pulse*. If the sink node is present in the network then it broadcasts the schedule to actor nodes otherwise they decide by their IDs (in the increasing order of their IDs). Therefore, the subinterval is shared among actors by dividing the subinterval into the number of slots equal to the number of actors ( $m$ ). The length  $\zeta$  of the slot can be computed as

$$\zeta = 2^{30}/m$$

Each actor wait for its turn in the subinterval and utilize its allocated slot for transmission. Hence, the transmission pattern for  $m$  actors in  $i$ th interval is  $[i \times T, i \times T + \zeta, i \times T + 2\zeta, \dots, i \times T + (m - 1)\zeta]$ .

### 2.3.3 Node Localization

We describe the node localization procedure in this section. This procedure is composed of two phases; *distance measurement* and *node positioning*. In first phase, sensor nodes measure distance from all the reference stations that is used for estimating the spatial location of nodes in second phase.



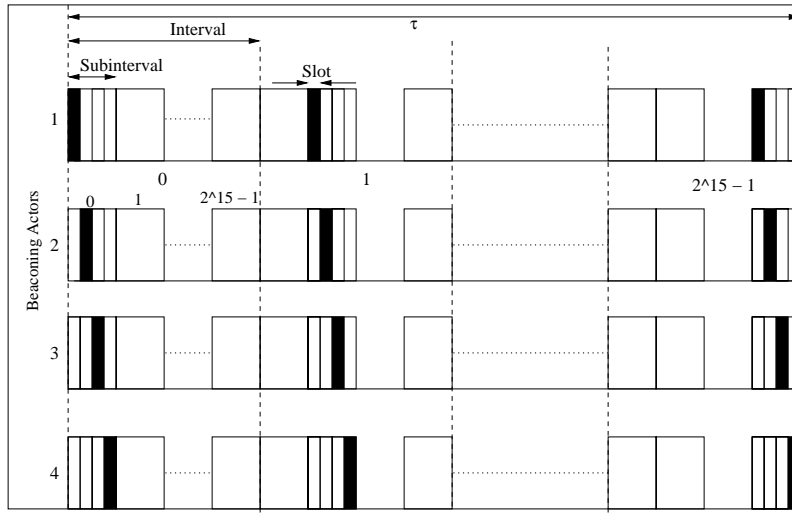


Figure 2.3: Transmission pattern of *reference pulse* by 4 actors.

### Distance Measurement

Given the propagation time of a pulse from sender to receiver, the distance between the two nodes can be measured by using propagation speed of the RF signal which is the speed of light. In TSL, sensor nodes follow the same approach to measure the distance  $D$  from actors as

$$D = c \times E[t_p] \quad (2.3)$$

where  $c$  is the speed of light that is approximately  $3 \times 10^8$  m/s.  $E[t_p]$  is the mean of propagation delay of the pulse over some intervals and is calculated as explained in the following.

Actor node inserts the running interval  $\aleph$  in its pulse that helps the sensor nodes in determining the start time of nodes locally. The running interval represents the current interval of actors in which they transmit the pulses. When a sensor nodes receives the pulse, it can compute the start time  $t_s = T(\aleph)$  of each pulse by using (2.2). Hence, every node is able to estimate  $t_s$  of the pulse without employing synchronization of the nodes. Therefore, a node calculates the propagation time  $t_p$  of the pulse as

$$t_p = t_r - t_s$$

where  $t_r$  is the received time of the pulse. Usually the transmission of pulse does not start immediately as it is scheduled when the wireless channel is contended among large number of nodes that is very common in WSNs. There occurs some lag in the actual transmission of pulse. Various factors involve in lagging transmission. Actors report this lagging time  $t_{lag}$  in pulses to accurately measure  $t_p$  because sensor nodes are unaware of the TDMA schedule of actor nodes. Therefore, they measure incorrect  $t_p$  if the scheduling delay is not informed. The pulse carries this scheduling delay in its *Lag Time* field and is calculated as.

$$t_{lag} = \zeta \times slot_{no}$$

where  $slot_{no}$  is the slot number allocated to each actor in subintervals to start its pulse transmission. Although the actor nodes follow the TDMA schedule in pulses, it is highly likely that the medium is still not available due to the transmission of surrounding sensor nodes or captured by some actor node and pulse transmission is delayed. Therefore, we need to include the medium access time  $t_{mac}$  to get more accurate measurement of  $t_p$ . The more accurate measurement of  $t_{lag}$  is obtained by including the factor  $t_{mac}$ .

$$t_{lag} = \zeta \times slot_{no} + t_{mac} \quad (2.4)$$

The value of  $t_{lag}$  reported in pulse is subtracted from the estimated value of  $t_p$  to obtain the accurate time-of-flight.

$$t_p = t_r - t_s - t_{lag} \quad (2.5)$$

In order to obtain the mean of the propagation time, we collect the the values of  $t_p$  until either the actor moves or node itself. The movement of actor is reported in pulse which sets the flag  $M$  and nodes are aware about their mobility themselves. Let we start

collecting the samples of  $t_p$  at time  $t$  and suppose that at time  $\hat{t}$ , either actor moves or node itself. Now suppose that a node receives  $k$  pulses from a particular actor during the time period  $\hat{t} - t$  and measures the propagation times  $t_{p1}, t_{p2}, \dots, t_{pk}$  of each pulse. Hence, the sample mean over the  $k$  intervals can be found as

$$E[t_p] = \frac{1}{k} \sum_{i=1}^k t_{pi} \quad (2.6)$$

This implies that the accuracy of the sample mean as an estimator of the  $t_p$  mean increases with the sample size  $k$ . Hence, an unbiased estimator of  $t_p$  variance becomes

$$\hat{t}_p = \frac{1}{k-1} \sum_{i=1}^k (t_{pi} - E[t_p])^2 \quad (2.7)$$

The estimator  $\hat{t}_p$  allows us to determine the channel condition that effect the measurement of  $t_p$ . Sensor nodes determine their distance from the actors by first measuring the mean value  $E[t_p]$  given in (2.6) and then multiplying it with the speed of light (2.3). The deviation  $\hat{t}_p$  is accounted for the first few observations after either the actor moves or node moves itself to minimize the possible errors in measurements.

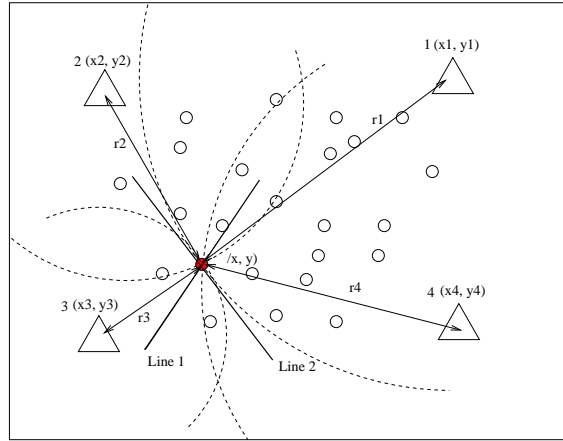


Figure 2.4: Location estimation of a sensor node.

## Spatial Location of Nodes

Nodes determine their position based on the distance estimates to the actors provided in the last section. We apply much simpler method and do not involve additional communication. We obtain two lines from the distance measurement of 4 actor nodes and the intersection point of these lines gives the exact location of the node.

Let  $A_1, A_2, A_3$  and  $A_4$  be the four actor nodes broadcasting *reference* pulse and  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$  and  $(x_4, y_4)$  be their positions respectively. Nodes calculate the distance  $r_1, r_2, r_3$  and  $r_4$  from the actors  $A_1, A_2, A_3$  and  $A_4$ , respectively, by using (2.3). Let  $(x_i, y_i)$  be the position of the node  $i$ . By using the Euclidean distance formula, we can derive the following second order distance equations for the four actors.

$$(x_i - x_1)^2 + (y_i - y_1)^2 = r_1^2$$

$$(x_i - x_2)^2 + (y_i - y_2)^2 = r_2^2$$

$$(x_i - x_3)^2 + (y_i - y_3)^2 = r_3^2$$

$$(x_i - x_4)^2 + (y_i - y_4)^2 = r_4^2$$

As shown in Fig 2.4, four circles are drawn at  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$  and  $(x_4, y_4)$  of radius  $r_1, r_2, r_3$  and  $r_4$  respectively, which pass through the unknown point  $(x_i, y_i)$ . Then, two lines  $L_1$  and  $L_2$  are drawn which pass through  $(x_i, y_i)$  and coincides a pair of circle as shown in Fig 2.4. We assume the actors to be placed in four different quadrants I, II, III and IV of x-y scale and make the pairs of actors lying in diagonally opposite quadrants i.e. pairs of I & III and II & IV are made. Therefore, actors in a pair are selected such that

$$(x_1 - x_2) > r_1 \quad \text{and} \quad (y_1 - y_2) > r_2$$

or

$$(x_1 - x_2) > r_2 \quad \text{and} \quad (y_1 - y_2) > r_1$$

for a pair  $A_1A_2$  of actors  $A_1$  and  $A_2$ . The above condition ensures the right pairs as stated above and consequently we obtain the lines  $L_1$  and  $L_2$  passing through the overlapping point  $(x_i, y_i)$ .

Let  $A_1A_3$  and  $A_2A_4$  be two such pairs then the equations of lines  $L_1$  and  $L_2$  are derived as:

$$2(x_1 - x_3)x_i + 2(y_1 - y_3)y_i = K_1 \quad (2.8)$$

$$2(x_2 - x_4)x_i + 2(y_2 - y_4)y_i = K_2 \quad (2.9)$$

where  $K_1 = r_3^2 - r_1^2 + x_1^2 - x_3^2 + y_1^2 - y_3^2$  and  $K_2 = r_4^2 - r_2^2 + x_2^2 - x_4^2 + y_2^2 - y_4^2$ .

By solving the (2.8) and (2.9), we can obtain the values of unknown coordinates  $x_i$  and  $y_i$ .

$$x_i = \frac{C_1K_1 - C_2K_2}{C_1C_4 - C_2C_3} \quad (2.10)$$

$$y_i = (K_2 - C_3 \frac{C_1K_1 - C_2K_2}{C_1C_4 - C_2C_3})/C_4 \quad (2.11)$$

where  $C_1 = 2(x_1 - x_3)$ ,  $C_2 = 2(y_1 - y_3)$ ,  $C_3 = 2(x_2 - x_4)$  and  $C_4 = 2(y_2 - y_4)$ .

The procedure executed by each node to find its position is given in Fig. 2.5.

### 2.3.4 Collaborative Localization

We have mentioned earlier that if some of the nodes do not hear the transmission of four actors then the collaborative localization is adapted where a localized neighboring nodes can be selected in place of missing actors. Since we restrict the four actors to lie in different quadrants of x-y scale, if a node does not hear the transmission of some actor then it selects one of the localized nodes among its neighboring nodes in the quadrant of missing actor as shown in Fig 2.6.

Due to limited transmission of actor 4, the  $i^{th}$  node could not hear the *reference pulse* of the actor from quadrant IV. Therefore, it looks for its neighboring nodes and selects the

node  $x$  that lies in IV with respect to  $i^{th}$  node. Hence, actor 4 is substituted by sensor node  $x$  and  $i^{th}$  node is localized that makes the approach range-free.

### 2.3.5 Accuracy in TSL and Parameter Selection

TSL aims to work accurately at required level with both the stationary as well as mobile nodes. Accuracy of the algorithm depends on the selection of localization period  $\mathfrak{J}$ . This value is used to control the frequency of pulses transmission that should be adaptive to the velocity of nodes and network dynamics. If the nodes are mobile and move with high velocity then  $\mathfrak{J}$  should be set to small value so that the frequency  $f$  of pulse transmission is increased. On the other hand, we set  $\mathfrak{J}$  to large value for stationary nodes because once the nodes are localized then their position remain valid throughout the life of network. However, the pulse frequency is also adjusted according to network dynamics due to the redeployment of nodes apart from the mobility. Therefore, this factor should be considered as well in deciding the value of  $\mathfrak{J}$ . For example, if we take  $\mathfrak{J} = 10h$  then the value of  $f$  is approximately 0.91 beacons/sec and nodes can renew their spatial location after 1.098 seconds.

In case of incremental deployment of stationary nodes, if nodes are redeployed after every 10 seconds then the *reference beacon* must be received by the nodes at least once in 10 seconds so that the redeployed nodes can be localized. In this case, the required frequency might be 0.1 beacons/sec, ignoring the mobility of nodes, and therefore we can set  $\mathfrak{J} \approx 91$  hours.

Similarly, wrong selection of  $\mathfrak{J}$  causes inaccuracy in location determination. For instance, when the velocity of a mobile node is 10 m/s. It changes its location by 10 meters in a second. If the frequency of *reference beacon* is 1 beacon/sec then the maximum location inaccuracy is 10 meters. However, by increasing  $f$  to 10 beacons/sec, we reduce the inaccuracy to 1 meter. The high value of  $f$  results in high cost in terms of energy and bandwidth. Therefore, the location accuracy increases with large value of  $f$  but decreases

with small value of  $f$  relative to  $v$ . There is trade off between accuracy and cost. High level of accuracy requires more resources but low accuracy is cost efficient. The accuracy can be set according to the application requirements.

We can establish a relation between the accuracy and  $\mathfrak{J}$  that helps in evaluating the performance of the algorithm. Let  $\eta$  be the required accuracy level with velocity  $v$  of mobile nodes, we can derive a relationship between  $f$ ,  $\eta$  and  $v$  as:

$$f = \eta \times v \quad (2.12)$$

$$\mathfrak{J} = 2^{15}/f \quad (2.13)$$

Hence, the TSL protocol parameters can be set to achieve the desired accuracy and a proper choice of the beaconing frequency is essential in order to minimize the overheads as well as location errors. Therefore, the value of  $\mathfrak{J}$  is adjusted relative to the value of  $f$  rather than setting some fix value.

## 2.4 Performance Evaluation

The performance of *TSL* is evaluated by using the network simulator *ns-2* [106]. The example scenario of wireless sensor and actor networks consists of 200 sensors deployed randomly in a field of  $150 \times 150$  and 4 actors initially deployed at the corner of the field such that they lie in four different quadrants of the field with respect to the sensor nodes. The transmission power of actors is adjusted so that all the nodes in field can hear *reference pulses*. However, the actors can move anywhere in the field after some initial localization and we can apply collaborative nodes localization later on if the condition of four actors is not meet.

Applications define their desired level of accuracy  $\eta$ . Since, the selection of  $\mathfrak{J}$  plays an important role in accuracy, we have evaluated the performance with fixed values of  $\mathfrak{J}$  as well as adjusted according to  $\eta$  and nodes velocity. Small value of  $\mathfrak{J}$  gives high

reliability however increases the pulse frequency or communication cost. We also consider the achieved accuracy with associated cost in our performance metrics.

Fig. 2.7 illustrates the accuracy of nodes locality for some fixed values of  $\mathfrak{J}$  irrespective of nodes velocity  $v$  and accuracy  $\eta$ . It can be observed that the accuracy decreases by increasing the velocity of nodes and distance error is more as compared to increase in  $v$ . By increasing  $v$  to two times, the inaccuracy approaches to approximately 3 times. On the other hand, the effect of  $\mathfrak{J}$  is opposite to  $v$ . For small value of  $\mathfrak{J}$ , accuracy is high and is low for large value of  $\mathfrak{J}$ . By decreasing  $\mathfrak{J}$  from 40 hours to 30 hours i.e. by 33% , the location errors are reduced approximately to 100%. Hence, a suitable selection of  $\mathfrak{J}$  (30 hours) keeps the localization errors below 1 meter as shown in Fig. 2.7.

Under high traffic load, it is difficult to achieve the high accuracy at low value of  $\mathfrak{J}$ . Small value of  $\mathfrak{J}$  increases the pulse frequency and decreases the time period of interval. In this case, actors keep contending the busy channel for pulses transmission and might not transmit pulses in the short period of interval. Intuitively, the nodes moving in that interval have incorrect location and therefore the small value of  $\mathfrak{J}$  does not significantly improve the accuracy as shown in Fig. 2.8. However, TSL has still restricted the location inaccuracy below 1 meter until the velocity of nodes is 6 m/s.

Unlike fixed value,  $\mathfrak{J}$  can be adjusted according to the required accuracy level of applications with different predictions about the velocity of nodes. We calculate the frequency of pulses by using (2.12) and, consequently, obtain  $\mathfrak{J}$  to achieve the desired  $\eta$ . The maximum velocity of nodes is set to 50 m/s and we measure the distance after 200 ms to identify the locality error. Clearly, the errors decrease by increasing the value of  $v$  that increases the frequency  $f$  and, eventually,  $\mathfrak{J}$  is reduced. The reduced value of  $\mathfrak{J}$  results in high accuracy by keeping errors less than half a meter as shown in Fig. 2.9. As the value of  $v$  approaches to its maximum, inaccuracy goes down to negligible value. Since  $f$  increases with increase in  $v$  that leads to high communication cost or localization overheads, the value of  $\eta$  can be set according to the required accuracy that limits the overheads relative



to  $\nu$ .

Fig. 2.10 represents the relation of three factors energy consumption,  $\nu$  and mean location errors. The energy consumption reported is the total cost of all the actors and sensors for a localization duration of 100 seconds. In our model, actor node consumes  $0.23 \text{ mJ}$  in transmitting a *reference pulse* over the whole field and nodes consume  $0.56 \mu\text{J}$  in receiving the pulse. We achieve high accuracy by setting large value of  $\nu$  but it increases the energy cost due to high beaconing frequency  $f$ .

We measure the scalability of protocol for different number of nodes and evaluate its performance for fixed values of  $\mathfrak{S}$  as well as adjusted according to  $\nu$  and  $\eta$ . Fig. 2.11 shows that the errors are less than half a meter and are not much affected by increasing the number of nodes. Increasing the number of nodes to 5 times, the errors increase only 10% which is very trivial impact. Similarly, for varying  $\mathfrak{S}$  we achieve high accuracy for large value of  $\nu$  and the increase in number of nodes does not prevent the accuracy achieved by limiting errors less than a meter for higher value of  $\nu$  as shown in Fig. 2.12 that the accuracy is 200% by increasing  $\nu$  to two times for almost all the values of nodes. Hence, we achieve high accuracy by selecting suitable values of  $\nu$  &  $\eta$  regardless of the number of nodes.

## 2.5 Summary

Location information in WSAN is, inherently, as important as the event readings. Actor nodes require the exact location of the sources in order to identify the location where an event originates and take certain action. There has been proposed numerous localization algorithms in WSNs. However, the localization of sensors and actors independent to each other does not suit in WSANs that requires a unified solution for sensors and actors. We propose a precise Timing-based Sensor Localization (TSL) algorithm that follows the discrete time intervals to estimate the location of sensor nodes. This allows us to measure

the propagation time of the pulses locally at the receiving nodes and thereby making it independent of the clocks of sending node. In this paradigm, actors are considered as reference nodes and broadcast *reference pulses* according to the defined pattern of intervals. In particular, our solution works efficiently in mobility scenario according to the application defined accuracy level. Simulation results prove that TSL achieves high accuracy and restricts the localization errors less than 1 meter by tuning it according to the expected velocity of nodes and required accuracy.

---

```

/* Timing-based Sensor Localization Algorithm. */
/* where node  $i$  estimates location  $(x_i, y_i)$  with reference to four actors 1, 2, 3 and
4. */
TSL()
/* Find mean propagation time  $\bar{t}_p$  of the pulses from each of the four actors over
 $m$  samples. */
 $\bar{t}_{p1}$  = mean_prop_time ( $t_{p1}[\ ]$ ,  $m$ );
 $\bar{t}_{p2}$  = mean_prop_time ( $t_{p2}[\ ]$ ,  $m$ );
 $\bar{t}_{p3}$  = mean_prop_time ( $t_{p3}[\ ]$ ,  $m$ );
 $\bar{t}_{p4}$  = mean_prop_time ( $t_{p4}[\ ]$ ,  $m$ );
/* distance of node  $i$  from actor 1 */
 $r_1 = 3.10^8 \times \bar{t}_{p1}$ ;
/* distance of node  $i$  from actor 2 */
 $r_2 = 3.10^8 \times \bar{t}_{p2}$ ;
/* distance of node  $i$  from actor 3 */
 $r_3 = 3.10^8 \times \bar{t}_{p3}$ ;
/* distance of node  $i$  from actor 4 */
 $r_4 = 3.10^8 \times \bar{t}_{p4}$ ;
/* Select pair of actors so that L1 is drawn by two actors either in quadrants I &
III or II & IV and similarly L2 for the rest two actors. */
if (( $x_1 - x_2$ ) >  $r_1$  && ( $y_1 - y_2$ ) >  $r_2$ ) ||
    (( $x_1 - x_2$ ) >  $r_2$  && ( $y_1 - y_2$ ) >  $r_1$ )
     $x_a = x_1$ ;  $x_b = x_2$ ;  $y_a = y_1$ ;
     $y_b = y_2$ ;  $r_a = r_1$ ;  $r_b = r_2$ ;
     $x_c = x_3$ ;  $x_d = x_4$ ;  $y_c = y_3$ ;
     $y_d = y_4$ ;  $r_c = r_3$ ;  $r_d = r_4$ ;
else if (( $x_1 - x_3$ ) >  $r_1$  && ( $y_1 - y_3$ ) >  $r_3$ ) ||
    (( $x_1 - x_3$ ) >  $r_3$  && ( $y_1 - y_3$ ) >  $r_1$ )
     $x_a = x_1$ ;  $x_b = x_3$ ;  $y_a = y_1$ ;
     $y_b = y_3$ ;  $r_a = r_1$ ;  $r_b = r_3$ ;
     $x_c = x_2$ ;  $x_d = x_4$ ;  $y_c = y_2$ ;
     $y_d = y_4$ ;  $r_c = r_2$ ;  $r_d = r_4$ ;
else if (( $x_1 - x_4$ ) >  $r_1$  && ( $y_1 - y_4$ ) >  $r_4$ ) ||
    (( $x_1 - x_4$ ) >  $r_4$  && ( $y_1 - y_4$ ) >  $r_1$ )
     $x_a = x_1$ ;  $x_b = x_4$ ;  $y_a = y_1$ ;
     $y_b = y_4$ ;  $r_a = r_1$ ;  $r_b = r_4$ ;
     $x_c = x_2$ ;  $x_d = x_3$ ;  $y_c = y_2$ ;
     $y_d = y_3$ ;  $r_c = r_2$ ;  $r_d = r_3$ ;
end;
/* Find the intersection point of L1 & L2 as in (2.10) */
/* L1:  $2(x_a - x_b)x_i + 2(y_a - y_b)y_i = K_1$  */
/* L2:  $2(x_c - x_d)x_i + 2(y_c - y_d)y_i = K_2$  */
 $K_1 = r_b^2 - r_a^2 + x_a^2 - x_b^2 + y_a^2 - y_b^2$ ;
 $K_2 = r_d^2 - r_c^2 + x_c^2 - x_d^2 + y_c^2 - y_d^2$ ;
/* implement Eq 2.10 */
 $x_i = \mathbf{FindX}(x_a, x_b, x_c, x_d, y_a, y_b, y_c, y_d, K_1, K_2)$ ;
 $y_i = \mathbf{FindY}(x_a, x_b, x_c, x_d, y_a, y_b, y_c, y_d, K_1, K_2)$ ;
return( $x_i, y_i$ )

```

---

Figure 2.5: Algorithm of Node Localization.

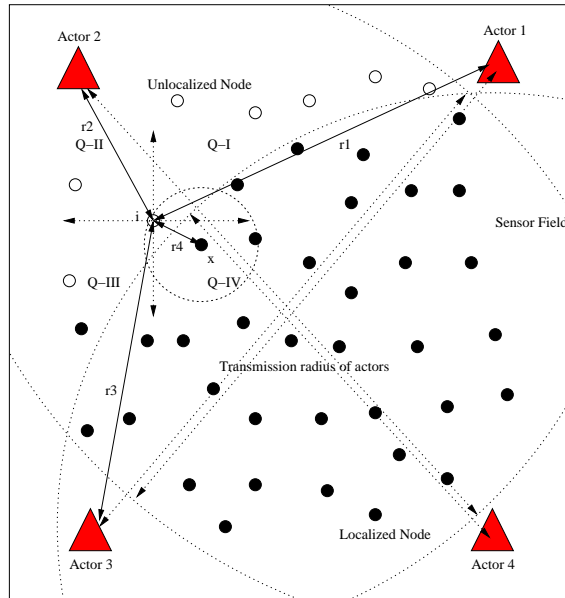


Figure 2.6: Collaborative location estimation of a sensor node receiving pulses from 3 actors.

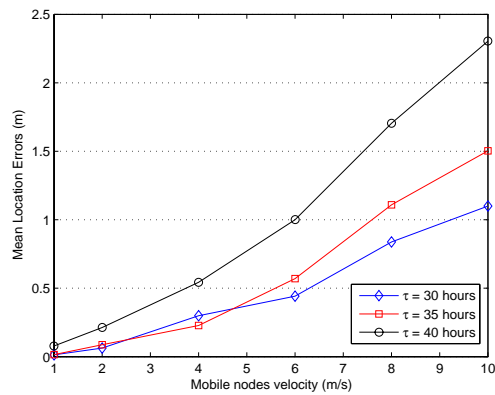


Figure 2.7: Location errors with fixed values of  $\mathfrak{J}$  when the nodes are silent.

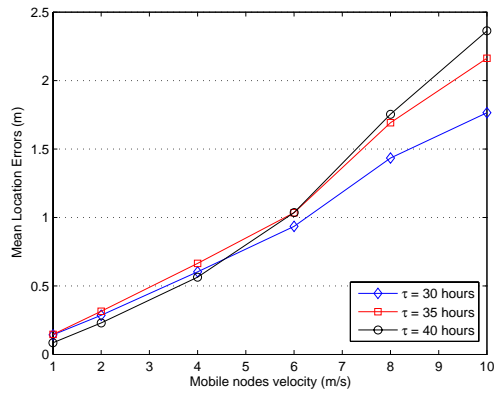


Figure 2.8: Location errors with fixed values of  $\mathfrak{J}$  when nodes are generating traffic at 2kbps.

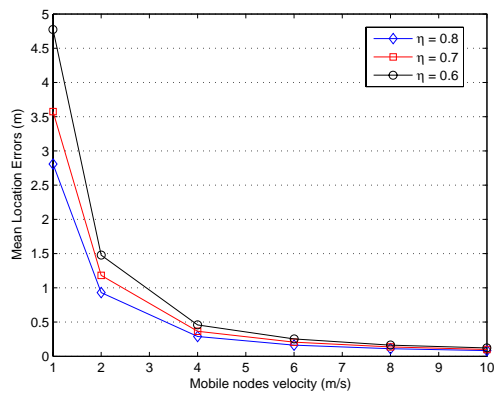


Figure 2.9: Location errors with different values of  $\eta$ . Maximum nodes velocity is 50 m/s and  $\mathfrak{J}$  is adjusted according to  $\eta$  and  $v$

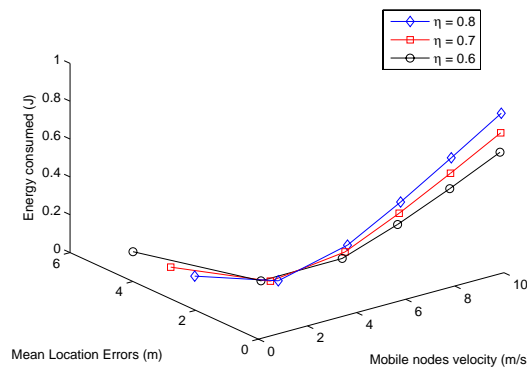


Figure 2.10: Location errors and communication cost with different values  $f$  set according to  $\eta$  and  $v$

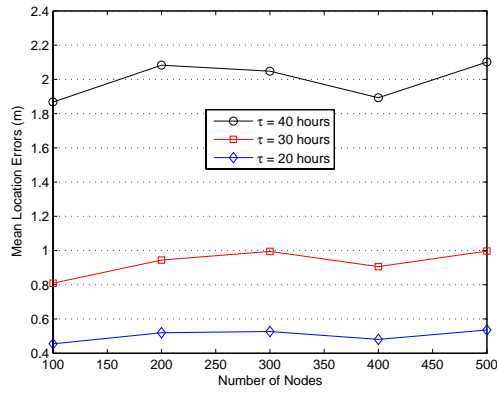
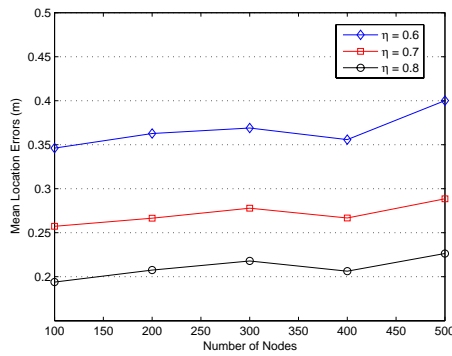
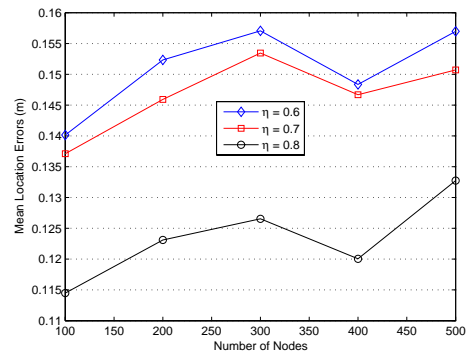


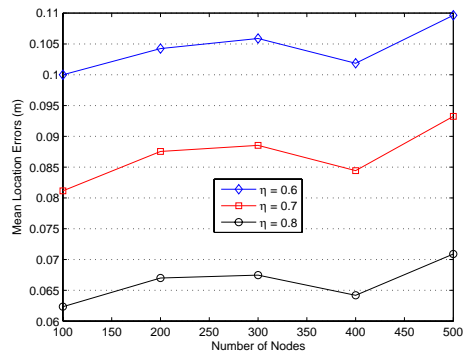
Figure 2.11: Location errors with fixed values of  $\mathfrak{I}$  for different number of nodes.



(a)



(b)



(c)

Figure 2.12: Location errors of nodes with maximum velocity of 10 m/s and varying  $\mathfrak{I}$  set according to  $\eta$  &  $v$  where Fig 2.12(a), 2.12(b) and 2.12(c) show errors for  $v = 1m/s$ ,  $v = 2m/s$  and  $v = 3m/s$  respectively.

## CHAPTER 3

### MULTI-EVENT ADAPTIVE CLUSTERING PROTOCOL (MEAC)

#### 3.1 Introduction

Sensor networks are characterized by a highly dynamic topology, due to a significant level of node failures (e.g. because of energy depletion) or re-energizing caused by deploying new nodes. Therefore, the network must be able to periodically reconfigure itself so that it can continue to function. The implementation of self-configuration then become a requirement in order to guarantee efficient network operation [1]. A number of protocols [32, 34, 36, 37, 39, 40, 41, 42] have been proposed for WSN. In most of these studies, sensor nodes are assumed to be homogeneous. However, depending on the application, sensor nodes can have different role or capability making the network heterogeneous. These special sensors can be either deployed independently or the different functionalities can be included in the same sensor nodes. For example, some applications might require a diverse mixture of sensors for monitoring temperature, pressure and humidity of the surrounding environment and capturing the image or video tracking of objects. Even data reading and reporting can be generated from these sensors at different rates and can also follow multiple data reporting models.

In this study, we present Multi-Event Adaptive Clustering (MEAC) protocol for heterogeneous wireless sensor networks. MEAC constructs clusters which is adaptive to the uniform as well as non-uniform deployment of nodes in heterogeneous wireless sensor

network. MEAC conserves the energy of sensor nodes in three ways. First, it computes the optimal number of clusters that should be formed according to the given topology information i.e. number of sensor nodes, dimensions of the field and the transmission radius of nodes. It loosely restricts the network to avoid from the unnecessary clusters formation since there are some overheads involved in the clustering process. By loose restriction, we mean that the number of formed clusters may be different than the optimal clusters due to the non-uniform deployment of nodes in some scenario.

Second, it uses an application-oriented weight-based clustering algorithm to select optimal number of cluster-heads. Generally, the clustering protocols [34, 36] focus on the currently available energy of the nodes and periodically reorganize clusters to do energy balancing. However, this strategy is not practical when the nodes are sending traffic at different rates due to different events characteristics. If all the nodes have the same probability to become cluster heads then the nodes reporting events at higher rate will eventually lose their energy earlier than the others. Therefore, the load on a node, that is, its data rate is used as a key factor for cluster-head selection. Hence, MEAC distributes the energy usage of nodes by adapting to the multiple events in the field in order to increase the stable period of the network. Finally, it defines threshold to the accumulated weight and does not dissolve the cluster until the weight goes down to the threshold as opposed to the periodical cluster reformation. Simulation results have shown that MEAC is more energy efficient than existing clustering protocols and is capable of handling the network dynamics.

## **3.2 Heterogeneity Model**

This section describes the model of heterogeneous wireless sensor actor network where the heterogeneity of sensor nodes is considered regarding their different initial energy levels and observation of multiple events. Sensor nodes may be embedded either with



multiple sensing units or they have different sensing modules to detect multiple events. Multi-events can generate different reporting rates due to the different characteristics of events and requirements. Hence, it greatly effects the energy consumption of nodes.

We assume that the network is stable \* if all the nodes are alive and they have enough energy to detect and relay packets. This period is known as *stable period* of the network. In heterogeneous wireless sensor actor networks, sensor nodes may have different energy levels and might report events at different data rate. The nodes having the initial least energy  $E_o$  are termed as energy-constrained (EC) nodes and all other nodes having energy higher than  $E_o$  i.e.  $E_o + \delta$  are energy-rich (ER) nodes. The degree of heterogeneity is also affected by multiple data rates in the network. Other factors like computational or memory capabilities also contribute to network's heterogeneity which are not considered in this work.

The degree of the heterogeneity ( $\lambda$ ) is change in energy level and data rate that can be measured as:

$$\lambda = \lambda_{energy} + \lambda_{rate} \quad (3.1)$$

where  $\lambda_{energy}$  is the contribution of energy to  $\lambda$  and  $\lambda_{rate}$  is contribution in  $\lambda$  due to different data rates.

In next subsections, we show how individually  $\lambda_{energy}$  and  $\lambda_{rate}$  contribute to the overall degree of heterogeneity.

### **Contribution of $\lambda_{energy}$ to $\lambda$**

Let us assume that there exist  $w$  number of ER nodes among total of  $n$  nodes in the network. Let  $\delta_1, \delta_2, \dots, \delta_i, \delta_j, \dots, \delta_w$ , be the residual energies of  $w$  nodes. When  $\delta$  is constant for all  $w$  nodes, then  $\delta_i = \delta_j$ ; where  $i \neq j$ . In other words, there are only two different

---

\*Network is considered stable as long as all the nodes are alive. However, if any of the nodes loses its battery power then the network is assumed to be unstable.

kind of nodes having energy levels  $E_o$  or  $E_o + \delta$ . In this case,  $\lambda$  is large when  $w \approx \frac{n}{2}$ , but  $\lambda$  is small when  $w < \frac{n}{2}$  or  $w > \frac{n}{2}$  for  $w \leq n$ . Then the fraction of nodes ( $w_\lambda$ ) making the network heterogeneous due to energy is:

$$w_\lambda = 1 - \left| \frac{n - 2w}{n} \right| \quad (3.2)$$

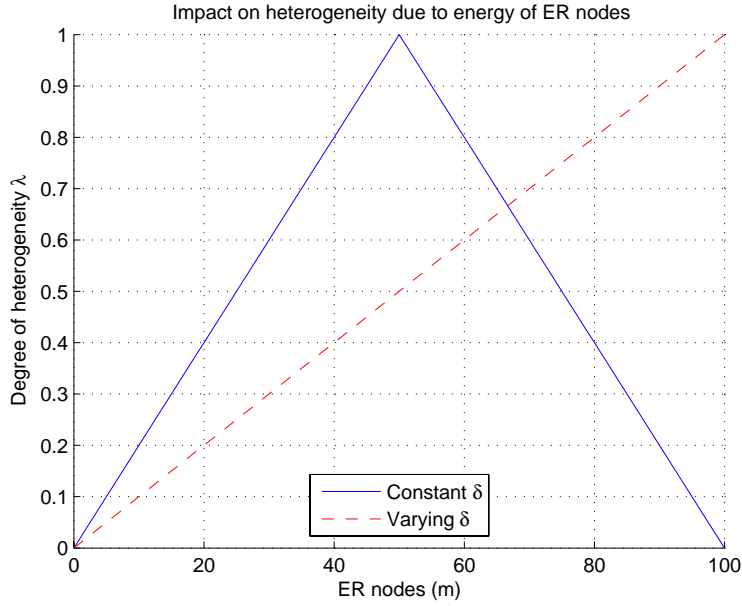


Figure 3.1: Heterogeneity due to number of  $m$  nodes.

On the other hand, when  $\delta_i \neq \delta_j$ , where  $i \neq j$ ; ER nodes have different energy levels above  $E_o$ . Hence, energy levels of ER nodes will be  $E_o + \delta_1, E_o + \delta_2, \dots, E_o + \delta_i, E_o + \delta_j, \dots, E_o + \delta_w$ . This behavior of  $\delta$  is depicted in Fig 3.1,  $\lambda$  increases continuously by increasing  $w$ . Therefore, when  $\delta$  is variable, the fraction  $w_\lambda$  is simply  $w/n$ . Let  $\alpha$  be the energy factor that ER nodes have higher than EC nodes. We can calculate  $\alpha$  as

$$\alpha = \frac{1}{wE_o} \sum_{i=1}^w (E_i - E_o) \quad (3.3)$$

The above equation can be simplified for constant  $\delta$  as

$$\alpha = \frac{w(E_i - E_o)}{wE_o} = \frac{E_i}{E_o} - 1 \quad (3.4)$$

Therefore, heterogeneity due to energy  $\lambda_{energy}$  or the energy gain in the network due to ER nodes is  $\alpha$  times  $w_\lambda$  i.e.  $\lambda_{energy} = \alpha \times w_\lambda$

### Contribution of $\lambda_{rate}$ to $\lambda$

Let  $\rho_o$  be the lowest initial data rate of a node(s) in the network and  $q$  be the number of nodes among the total of  $n$  nodes which have data rate higher than  $\rho_o$  in the network. Similar to (3.2) for constant  $\delta$ , the fraction of nodes ( $q_\lambda$ ) making the network heterogeneous due to diversity in data rates can be calculated as

$$q_\lambda = 1 - \left| \frac{n - 2q}{n} \right| \quad (3.5)$$

Let  $\rho_i = \rho_o + \delta$  be the data rate of node  $i$ , then the data rate fraction  $\varphi$  that  $k$  nodes produce more than  $\rho_o$  can be defined similar to  $\alpha$  as:

$$\varphi = \frac{1}{q\rho_o} \sum_{i=1}^q \rho_i - \rho_o \quad (3.6)$$

The simplified equation of  $\varphi$  for constant  $\delta$  is:

$$\varphi = \frac{q(\rho_i - \rho_o)}{q\rho_o} = \frac{\rho_i}{\rho_o} - 1 \quad (3.7)$$

Hence, the fraction of heterogeneity due to different data rates ( $\lambda_{rate}$ ) can be given as  $q_\lambda \times \varphi$ .

For example, assume that the network contains 30% ER nodes having  $E_o = 1.5j$  and EC nodes with  $E_o = 0.5j$ . 10% nodes report readings at 40 packets/sec while the other nodes at 10 packets/sec. We find out the value of  $\lambda$ . We get  $\alpha = 0.2$  by using Eq 3.3 and

$w_\lambda = 0.6$  due to constant change  $\delta$  in all the  $w$  nodes. Similarly, for  $\rho_o = 10$ ,  $q = 10$ , the higher data rate factor  $\varphi = 3.0$  making  $q_\lambda = 0.2$ . Hence,  $\lambda$  is 1.8 by using (3.1).

### 3.3 MEAC Protocol

To provide a real-time coordination among sensors and actors in WSN, *RCR* configures the sensor nodes hierarchically. The operations of the routing protocols are discussed in Section 4.2. The configuration of sensor nodes to achieve real-time coordination is discussed in this section. To achieve hierarchical configuration, *ERCR* framework incorporates Multi-Event Adaptive Clustering Protocol (*MEAC*). The operations of *MEAC* consist of cluster formation of sensor nodes, delay budget estimation for forwarding a packet from the cluster-heads and to guarantee the packet delivery within the given delay bound  $\tau$ .

*MEAC* does not reform clusters periodically and also the cluster size is not fixed in terms of hops. It adapts according to the dynamic topology of the sensor and actor networks. This work is motivated from the previous work “A Weight Based Distributed Clustering” [38]. However, unlike [38], *MEAC* provides the cluster formation procedure to cope with the dynamic number of hops in a cluster and provides support for real-time routing. Cluster formation is based on the weighting equation formulated in the Section 3.3.3. Once the cluster has been formed, cluster-heads get estimates of *delay budget*<sup>†</sup> of their member nodes. The delay budgets of member nodes help to build the delay-constrained energy efficient path.

When the sensor nodes are not uniformly deployed in the sensor field, the density of nodes could be different in different zones of the field. Choosing an optimal number of clusters  $k_{opt}$ , which yield high throughput but incur latency as low as possible, is an important design goal of *MEAC*.

---

<sup>†</sup>*delay budget* is the time to deliver the data packet from the cluster-head to the member node.

### 3.3.1 Optimal Number of Clusters ( $k_{opt}$ )

MEAC computes optimal number of clusters ( $k_{opt}$ ) such that it decreases the energy consumption, while providing high degree of connectivity. We devise a formula to find  $k_{opt}$  that is directly proportional to total number of nodes and the area of the network, while it is inversely proportional to the transmission range.

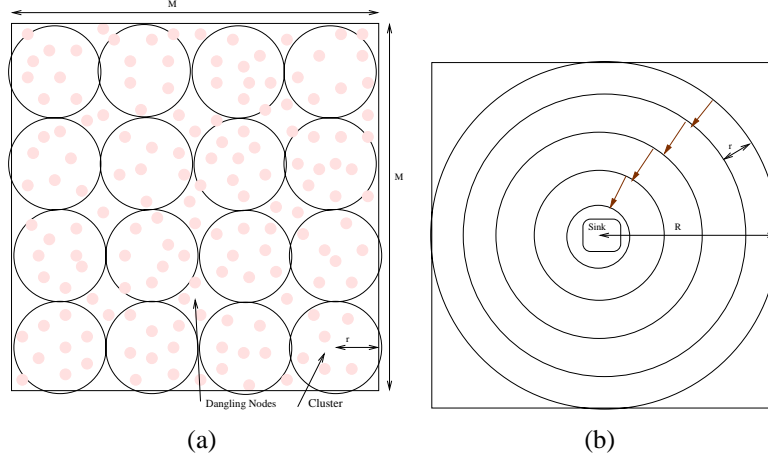


Figure 3.2: Network model to formulate the optimal clusters. Fig 3.2(a) represents the model to find the probability of DN nodes. Fig 3.2(b) illustrates the model of routing packets from cluster-heads to the sink node.

Let  $r$  be the transmission radius of each node regardless of its functioning. In the clustering process, there is some probability that a number of dangling nodes  $\ddagger$  (DN) may exist due to the deployment of nodes or coverage of the elected cluster-head. To find out the probability of such nodes, we map the sensor field ( $M \times M$ ) to non-overlapping circles of radius  $r$  as shown in Fig 3.2(a) and assume that the nodes lying outside the boundary of the circle are DN nodes and the others are member nodes. These DN nodes affiliate to cluster-heads through the nodes inside the circle (member nodes) and become the multi-hop members of cluster-heads. The square field  $M^2$  can be packed by  $M^2/(2r)^2$  non-overlapping circles of radius  $r$ . Thus, the probability  $P_{DN}$  of a multi-hop member is

$$P_{DN} = \frac{M^2}{(2r)^2} \times \frac{(2r)^2 - \pi r^2}{M^2} \approx 0.214$$

$\ddagger$ The nodes which have not joined any cluster are referred as *dangling nodes*.

Let  $E_{elec}$  be the energy consumed by the electronic circuitry in coding, modulation, filtration and spreading of the signal. Whereas,  $\epsilon_{amp}r^2$  is the energy consumed for signal amplification over a short distance  $r$ . Thus, the energy consumed by each member node is

$$E_{Member} = l(E_{elec} + \epsilon_{amp}r^2(1 + P_{DN}))$$

where  $l$  is the size of data packet. The above equation can be simplified by taking the area as circle given in (16) of [35].

$$E_{Member} = l(E_{elec} + \epsilon_{amp} \frac{M^2(1 + P_{DN})}{2\pi k})$$

Let us assume that the sensory field is covered by a circle of radius  $R$ , where the sink node lies at the center of this circle as shown in Fig 3.2(b). This assumption is made for sending packets from cluster-heads to the sink. Cluster-heads do not extend their transmission range and, therefore, has the same radius  $r$  as member nodes. This adapts the multi-hop model proposed by [33] to route packets from cluster-head to the sink.

In the model, a circle is divided into concentric rings with the distance of  $r$ . The energy spent to relay the packet from outside ring toward inside ring is  $l(2E_{elec} + \epsilon_{amp}r^2)$ . The number of hops  $H_{CH-s}$  required to route packet from cluster-head to sink node can be calculated by  $\frac{R}{r}(1 - P_{hops})$ , where  $P_{hops}$  is the probability indicating the distance in terms of hops to the sink. This probability can be calculated by using the nodes distribution in the rings given in [33].

$$P_{hops} = \frac{r}{R} \sum_{i=1}^{R/r} \frac{R^2 - (ir)^2}{M^2}$$

Packets from cluster-heads that are far from the sink are relayed through intermediate nodes. Therefore, if  $N_s$  is the number of neighbors of the source node  $s$  then  $N_s \times E_{elec}$  is

the energy consumed by the electronic circuitry of the neighbors during the transmission of a data packet by  $s$ . The number of neighbors  $N_s$  of a node can be computed by  $n\frac{\pi r^2}{M^2}$ . Hence, the energy consumed in forwarding data from cluster-head to sink is measured as:

$$E_{CH-S} = l(N_s E_{elec} + E_{elec} + (2E_{elec} + \epsilon_{amp}r^2 + N_s E_{elec})H_{CH-S})$$

The total energy dissipated by the network is

$$E_{total} = l((n + nN_s)E_{elec} + k(2E_{elec} + \epsilon_{amp}r^2 + N_s E_{elec})H_{CH-S} + n\epsilon_{amp}\frac{M^2(1 + P_{DN})}{2\pi k})$$

For  $r < R$ , the optimal value of  $k$  can be found by taking the derivative of above equation with respect to  $k$  and equating to zero

$$k_{opt} \approx \sqrt{\frac{n(1 + P_{DN})}{(2\pi(1 + \frac{2E_{elec}}{\epsilon_{amp}r^2} + \frac{N_s E_{elec}}{\epsilon_{amp}r^2}))H_{CH-S}}} \times \frac{M}{r} \quad (3.8)$$

The optimal value depends on the transmission range  $r$ . For long range transmission, the optimal number of clusters  $k_{opt}$  is small. For example, let  $n = 100$ ,  $M = 100$  and the sink is at the center of the field ( $x = 50, y = 50$ ). Then, the value of radius  $R$  is obtained by drawing a circle at  $x = 50, y = 50$  to cover the field. The estimated value is  $R = 60$  and let set the range  $r$  of individual nodes to 25. In this scenario, we obtain the value of  $k_{opt} \approx 10$ . By increasing the range of nodes to 40 meters, we obtain  $k_{opt} \approx 7$ . Whereas, the value of  $k_{opt}$  in SEP [36] is 10 regardless of the transmission coverage of individual nodes.

Generally, a cluster is formed among a group of sensor nodes which hear the transmission of each other. The size of a group is large at higher density that results in a small number of clusters. Conversely, the size of cluster is small at lower density because a

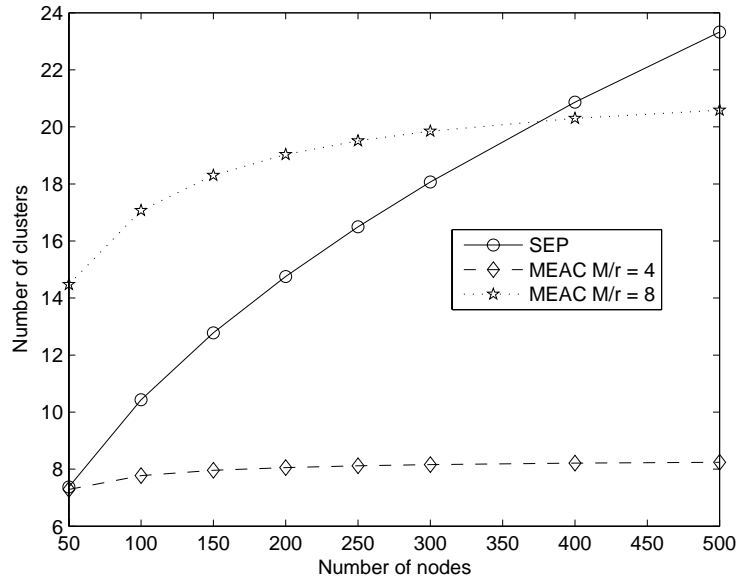


Figure 3.3: Cluster formation in SEP and MEAC for different values of number of nodes.

lesser number of nodes hear each other that results in a large number of clusters. The nodes density ( $\mu$ ) can be computed as

$$\mu(r) = \frac{n\pi r^2}{M^2}$$

For a given number of nodes  $n$ ,  $\mu$  varies by changing either the dimension of the field or the transmission radius. If the nodes are deployed in a small area, then the density increases and decreases if the same number of nodes deployed in relatively large area. On the other hand,  $\mu$  increases by increasing the transmission radius  $r$  and vice versa. However, for a given dimensions of the field and transmission radius, an increase in number of nodes increases only the group size. Eventually, the number of formed clusters remain the same. Fig. 3.3 shows the adaptivity of MEAC to different density of nodes. At higher density for  $100 \times 100 \text{ m}^2$  field and  $r = 25 \text{ m}$ , MEAC forms lesser number of clusters than SEP regardless of the number of nodes deployed. However, the number of clusters increases at lower density by setting the dimensions to  $200 \times 200 \text{ m}^2$ . It is shown in Fig.



3.3 that SEP is not adaptive to dimension of field and transmission radius but the number of clusters varies only with the number of deployed nodes.

### 3.3.2 Optimal Cluster Size ( $N_{opt}$ )

When the deployment is uniform, the optimal value of member nodes  $N_{opt}$  can be easily found by  $n/k_{opt}$ . However, for non-uniform deployment, the number of member nodes depends on the density in a particular zone of the sensor field. Therefore, we put the maximum and minimum limits  $N_{Min}$  and  $N_{Max}$  respectively on the size of cluster, such that, we still achieve  $k_{opt}$  clusters in non-uniform deployment. Let  $N_i$  be the number of neighboring nodes of any  $i$ th node.  $Max(N_i)$  is the maximum number of neighboring nodes that any of the  $i$ th neighbor node have. We measure density of nodes in a particular zone by comparing the neighbor nodes  $N_i$  with  $N_{opt}$ . It can be concluded that the deployment is:

$$\begin{aligned}
 & \textit{dense} \quad \textit{if} \quad N_i/N_{opt} > 1 \\
 & \textit{uniform} \quad \textit{if} \quad N_i/N_{opt} \approx 1 \\
 & \textit{sparse} \quad \textit{if} \quad N_i/N_{opt} < 1
 \end{aligned}$$

Therefore, the number of nodes in a cluster can be constrained by setting the lower bound  $N_{Min}$  and upper bound  $N_{Max}$  as shown in Fig. 3.4 according to the deployment as:

$$N_{Max} = Max(N_{opt}, Max(N_i))$$

That is, the maximum of  $N_{opt}$  and maximum number of neighbors of any cluster-head at the time of cluster formation.

$$N_{Min} = N_{opt} \times \text{Min}(N_{opt}, \text{Max}(N_i)) / N_{Max}$$

These limits allow the configuration to manage the dense as well as sparse deployment of nodes.

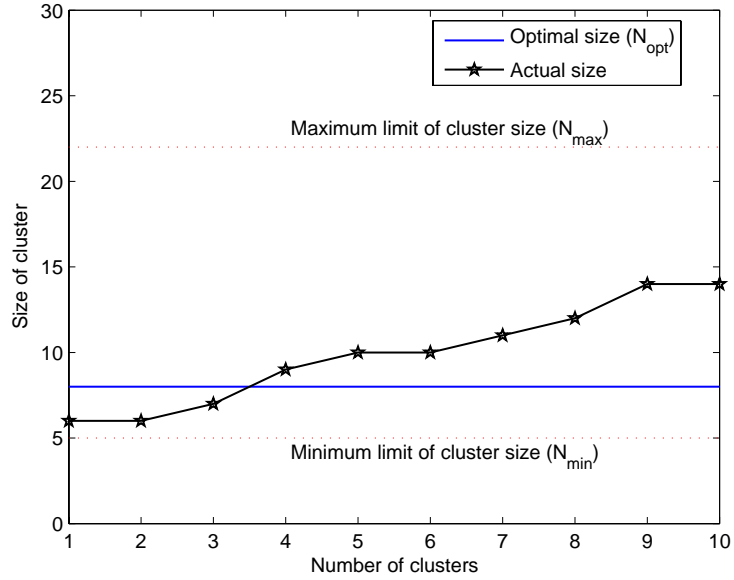


Figure 3.4: Size of clusters in the network of 100 nodes deployed in  $100 \times 100 m^2$  field.

### 3.3.3 Cluster Formation

The first phase of MEAC is to form  $k_{opt}$  number of clusters. During the formation of clusters, each cluster-head gets the *delay budget* of each of its member node. The delay budget is used to identify an appropriate node to send delay-constrained data packet. The cluster-head election procedure is based on calculating weight for each sensor node in the sensor field and ERCR chooses the head that has the maximum weight as explained in Section 3.3.3. We define weight threshold of the cluster-head to rotate the cluster-heads responsibility among all the potential nodes. A cluster is not strictly organized to 1-hop

but it accepts the membership of a node that could not reach any cluster in the first phase of cluster formation. Therefore, a cluster can include d-hop members, for  $d \geq 1$ . Although the operations of the protocol starts after the first phase of cluster formation, there may still exist some DN nodes.

Here, we describe the details of computing the *delay budget* and cluster formation procedure in the following sections.

### Delay Measurement

When nodes are initially deployed in the field, every node  $i$  broadcasts *hello* beacon that contains its ID, which is added in the neighbors list by all the receiving nodes. The receiving nodes compute the delay  $T_{sender}$  of the packet and also records the delay budget  $T_{receiver}$ .  $T_{sender}$  is the delay of the packet experienced and  $T_{receiver}$  is the delay that the sender estimated when a packet was received from the receiver, i.e,  $T_{receiver} : sender \leftarrow receiver, T_{sender} : sender \rightarrow receiver$ .

The total delay in transmitting a packet from one node to a node in its neighbor is measured by the following factors: queue, MAC, propagation and receiving delay represented by  $T_q, T_{Mac}, T_{Prop}$  and  $T_{Rec}$  respectively. The wireless channel is asymmetric that does not imply any synchronization mechanism. Therefore, the delay is measured partially at both the sender and receiver. Sender measures the delay  $L_s$  until the start of transmission that includes  $T_q$  as well as  $T_{Mac}$ . Whereas, the receiver adds the factor  $L_r$  as sum of  $T_{Prop}$  and  $T_{Rec}$  to get the total packet delay  $L_h$ .

$$L_s = T_q + T_{Mac}, \quad L_r = T_{Prop} + T_{Rec}$$

The hop latency  $L_h$  can be computed as sum of these factors:

$$L_h = L_s + L_r$$

Initially, the delay is measured by exchanging the *hello* beacons. Each node maintains this value in its neighborhood table that contains the fields  $\{ID, T_{receiver}, T_{sender}, energy, weight\}$ . To obtain more accurate measurement of packet delay, the delay value is updated when the events flow from member nodes to cluster-head. It is due to the fact that the size of data packet may differ than the *hello* beacon that may experience different delay.

For the  $d$ -hop member of a cluster, packets are forwarded by the intermediate nodes to the cluster-head. Each intermediate node calculates the delay  $L_h$  of the packet and forwards the packet to next hop by adding its  $L_h$  in  $L_s$ . After following through some intermediate nodes, cluster-head gets the packet and adds its factor  $L_r$  as the receiver of the packet. Hence the cumulative delay  $T_{sender}$  of a member node  $d$  hops away from its cluster-head is computed as:

$$T_{sender} = \sum_{i=1}^d L_{h_i} \quad (3.9)$$

Member nodes compute the delay estimate  $T_{sender}$  of *hello* beacon broadcast as cluster-head announcement. The announcement is simply made by setting a flag in the *hello* beacon. In the subsequent *hello* beacons, member nodes put the  $T_{sender}$  of cluster-head as  $T_{receiver}$ . It is important to note that, only cluster-heads require the  $T_{receiver}$  from their member nodes. Therefore, member nodes always put the  $T_{sender}$  of their exclusive cluster-heads in the beacon which represents  $T_{receiver}$  of the cluster-heads. Hence, cluster-head gets the delay budget  $T_{receiver}$  for its members and use this value in routing.

### Cluster-head Election Procedure

Cluster-heads are elected by effectively combining the required system parameters with certain weighting factors. Every node calculates its weight based on its available power, data rate and the density of nodes. Let  $D_i$  be the average distance of node  $i$  to its neighbors,  $N_i$  be the total number of its neighbors,  $E_i$  be its available energy and  $\rho_i$  be its

reporting rate. Node  $i$  computes its weight  $W_i$  as:

$$W_i = c_1 \frac{\rho_o}{\rho_i} \times \frac{E_i}{E_o} + c_2 \frac{D_i}{r} \times \frac{N_i}{N_{opt}} \quad (3.10)$$

where  $E_o$  and  $\rho_o$  represent the lowest energy level and data rate of the nodes respectively due to nodes heterogeneity. The coefficients  $c_1$ ,  $c_2$  are the weighting factors for the energy and data rate parameters. Node  $i$  announces itself a cluster-head if its weight is high among all its neighbors and sets its threshold  $W_{Th} = c_w W_i$ , where  $c_w$  is the threshold adjusting factor. It can be computed as

$$c_w = \frac{E[W]}{W_i} \quad (3.11)$$

where  $E[W]$  is the mean weight of the neighbors of node  $i$ .

It is quite possible that a node receives announcement from multiple cluster-heads. However, it decides its membership according to the closeness and joins the cluster whose cluster-head is closer than the others. The pseudo-code of the operations executed by a sensor node in each round of cluster formation is reported in Fig. 3.5.

Values of the coefficients can be chosen according to the application needs. For example, power control is very important in CDMA-based networks, where the weight of the power factor can be increased. On the other hand, the distance weighting factor can be made larger if the density of nodes is high or the deployment is made in hostile environment. This ensures that a node is elected as a cluster-head that can receive the transmission from farther nodes and the number of clusters formed remain close to the optimal value.

It can be noted in (3.10) that a node having higher energy level than its neighbors is the potential candidate of becoming cluster-head. However, a node reporting events at higher rate is less likely to be elected as cluster-head. The weighting equation includes the ratio of data rate ( $\rho_o/\rho_i$ ) to consider the multiple data rates due to different events. If a

---

```

/*Pseudo-code executed by each node  $i$  in each round.*/
ElectClusterhead( $i$ )
//maximum weight among neighbors.
 $W_{max} = 0$ ;
// $N_i$  is the list of neighbors of  $i$ .
for(each  $j \in N_i$ )
    if ( $W_{max} < W_j$ )     $W_{max} = W_j$ ;
end //for
 $W_i = \text{my\_weight}()$ ;
if ( $status == NONE$ )
    if ( $W_i > W_{max}$ )
        announce_head();
         $W_{th} = W_i \times \text{threshold}_{factor}$ ;
    end //if
else if ( $status == HEAD$ )
    if ( $W_i < W_{th}$ )
        if ( $W_i < W_{max}$ )
            withdraw_head();
        else
             $W_{th} = W_i \times \text{threshold}_{factor}$ 
        end //if
    end //if
end //if

```

---

Figure 3.5: Cluster-head election procedure.

node  $i$  has  $\rho_i > \rho_o$  then its weight is reduced. Therefore, it has lesser chances to become cluster-head than the other nodes. It is due to the fact that nodes sending packets at higher rates exhaust their energy soon and, thereby, the probability of becoming cluster-heads is reduced.

In order to achieve the goal of energy saving, it minimizes the frequency of cluster reformations. It is achieved by encouraging the current cluster-heads to remain cluster-heads as long as possible. MEAC reduces the frequency of clusters reformation by setting weight threshold at the time of cluster formation. In each round, each cluster-head recomputes its weight and compares with its threshold value. If  $W_i$  of cluster-head  $i$  is higher than its  $W_{Th}$  value then it keeps functioning as head. if  $W_i < W_{Th}$  then it checks whether its  $W_i$  is also lower than any of its member node weight. If so, it withdraws itself from being cluster-head and cluster election procedure is initiated.

### 3.3.4 Inter-Clusters Connectivity

The sink or actors can be multi-hop away from the source clusters, packets are then forwarded through intermediate clusters. Clusters are linked with each other to provide multi-hop cluster routing. Some member nodes within a cluster can hear the members of neighboring clusters or heads, such nodes act as *gateways*. It is also possible that there will be multiple *gateways* between two clusters. Cluster-heads keep record of all of these *gateways*.

We build a set of forwarding gateway nodes  $GS$ , for each cluster-head, for routing packets to neighboring clusters. Let  $SM_i$  be the set of members of cluster-head  $H_i$  and  $SM_j$  be the set of members of neighboring cluster-head  $H_j$ .  $H_i$  maintains a set of gateway nodes  $GS_i$  such that

$$GS_i(H_i) = \{x \in SM_i / H_j \in N_x \forall y \in SM_j \wedge y \in N_x \forall i \neq j\}$$

Where  $N_x$  is the set of neighbors of node  $x$ . A member node  $x$  of cluster-head  $H_i$  belongs to the gateway set  $GS_i$  of head  $H_i$  if either  $H_j$  or some member  $y$  of  $H_j$  exists in the neighbors set of  $x$ . The attributes of the elements of  $GS_i$  are  $\{AdjacentHead, Energy, Delay, Hops\}$ . These attributes help the cluster-heads in selecting a particular item from the set  $GS$ . We will describe the selection criteria in detail in Section 4.2.4.

Once the cluster formation is complete, each cluster gets the neighbor clusters list along with the gateways to reach them. The route computation is discussed in the next section.

## 3.4 Performance Evaluation

We evaluate the performance of protocol in terms of energy consumption, network stability with multiple events and throughput metrics. The heterogeneous WSN is composed

of nodes of different energy levels and sensing modules for multiple events detection. The example scenario of wireless sensor networks consists of 100 sensors deployed randomly in a field of  $100 \times 100$ . The sink node is placed at the center of field i.e.  $x = 50, y = 50$ . The initial energy  $E_o$  of EC nodes is set to 0.5 joules. The transmission and reception power is set to 50 nJ/bit and sources produce traffic at 4 kbps.

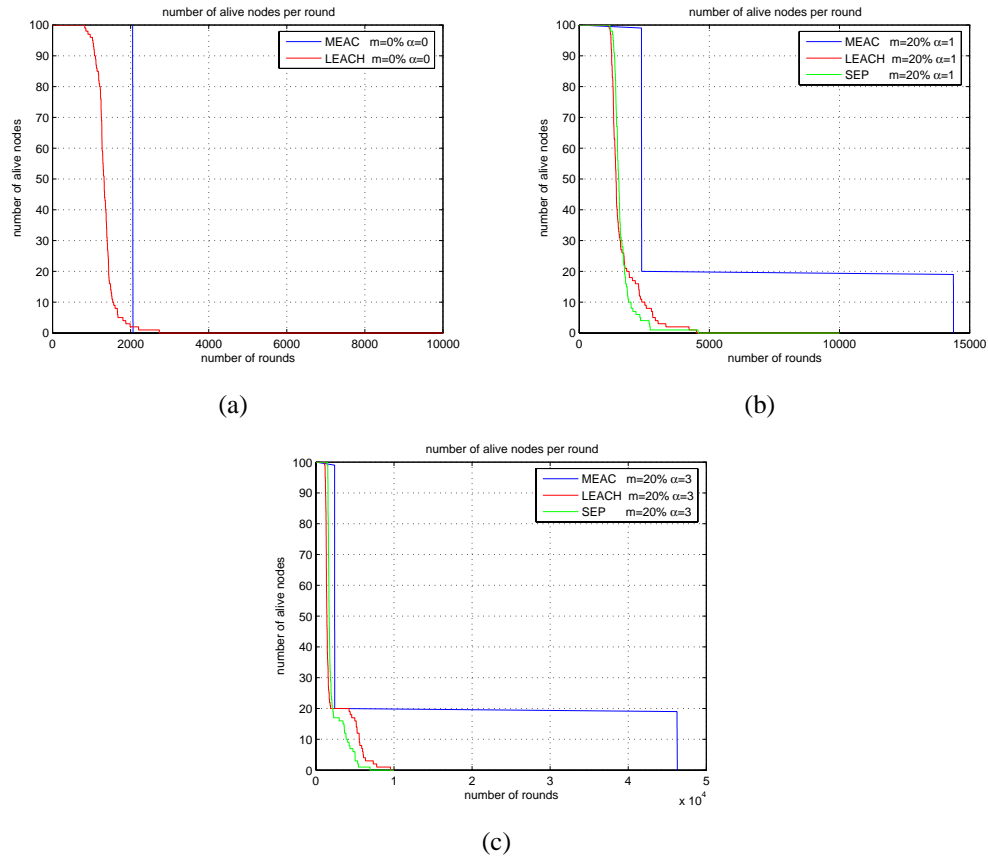


Figure 3.6: Energy consumption comparison among MEAC, LEACH and SEP in the presence of heterogeneity due to energy for  $\alpha = 0$  in 3.6(a),  $m = 20\%, \alpha = 1$  in 3.6(b) and  $m = 20\%, \alpha = 3$  in 3.6(c).

## Energy

The energy efficiency of MEAC is compared with SEP and LEACH. Both SEP and LEACH periodically elect cluster-heads to balance the energy of nodes. Fig 3.6 illustrates a detailed view of the behavior of MEAC, LEACH and SEP for different values of the



parameters. The number of alive nodes are plotted for the scenarios ( $m = 0\%$ ,  $\alpha = 0$ ), ( $m = 20\%$ ,  $\alpha = 1$ ) and ( $m = 20\%$ ,  $\alpha = 3$ ) in Fig 3.6(a), 3.6(b) and 3.6(c) respectively. Unlike SEP and LEACH, MEAC considers the available energy to elect cluster-heads and a node keep working as head as long as its available energy is higher than its threshold value. This approach reduces the frequency of cluster-head election.

It is obvious in Fig 3.6(a) that MEAC extends the stable region compared to LEACH by 55% for homogeneous network. The behavior of SEP is the same for  $m = 0$  and, therefore, the gain in stability is similar to LEACH. Fig 3.6(b) shows the results for  $m = 20\%$  and  $\alpha = 1$  parameters. The stable period is 41% and 33% more than LEACH and SEP respectively. Besides the stable period, the unstable period is also quite large which keep the network alive for 250% more than LEACH and SEP. Fig 3.6(c) illustrates the stability gain of MEAC for  $m = 20\%$  and  $\alpha = 3$ . MEAC achieves the gain of 58% in comparison with LEACH and 35% from SEP. The unstable region is remarkably larger than both these candidate protocols.

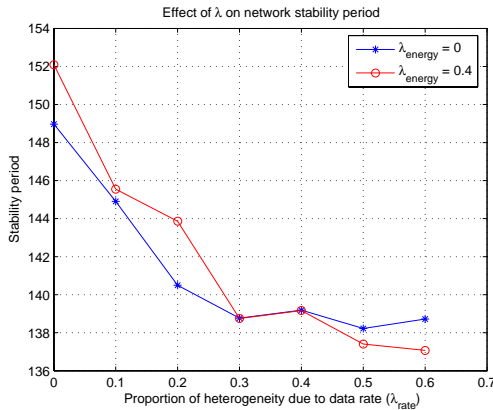


Figure 3.7: MEAC stable period in heterogeneous WSNs for different values of  $\lambda_{rate}$

### Stability with Multiple Events

Fig 3.7 shows the impact of multiple data rates with and without deployment of ER nodes. The stability of network increases by deploying more and more number of ER nodes. Whereas, it decreases when sensor nodes are reporting events at different rates

to sink. It is obvious from Fig 3.7 that the stability is high in the presence of ER nodes ( $\lambda_{energy} > 0$ ). The extra energy of ER nodes is utilized to accommodate the high data rate. If we keep increasing  $\lambda_{rate}$  (by increasing  $\varphi$ ) then the loss in stability is very small as compared to increase in  $\varphi$ .

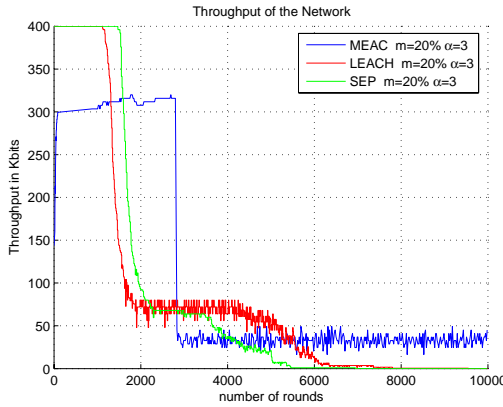


Figure 3.8: Throughput comparison among MEAC, LEACH and SEP.

## Throughput

MEAC does not imply any aggregation technique at cluster-heads because it does not suit for the reliability measure in terms of packet delay for delay-sensitive applications. Fig 3.8 shows the throughput comparison of MEAC with LEACH and SEP. MEAC aims to provide in-time packet delivery and sacrifices some throughput at cost of packet delay that increases due to aggregating data [48]. Although the throughput in MEAC is less than SEP but it continues for the longer time than in LEACH and SEP. Therefore, the low throughput is compensated by longer period. It is observed that when the ER nodes are close to sink then the throughput is high in unstable period but the period is short. When ER nodes are placed far from the sink node then some ER nodes might not reach sink directly or indirectly and, therefore, reduces the throughput but keeps the network alive for longer period. Thus the deployment of ER nodes greatly effect the network performance during unstable period.

## Adaptivity to Nodes Deployment

One of the key issues in WSN is the deployment of mobile sensor nodes in the region of interest (ROI) [44]. Before a sensor can report observation to the monitoring system, it must be deployed in a location that is contextually appropriate. Optimum placement of sensors results in the maximum utilization of the energy of nodes. However, The deployment can not be determined *a priori* when the environment is unknown or hostile in which case the sensors may be air-dropped from an aircraft [45] or deployed by other means. The proper choice for sensor locations based on application requirements is difficult.

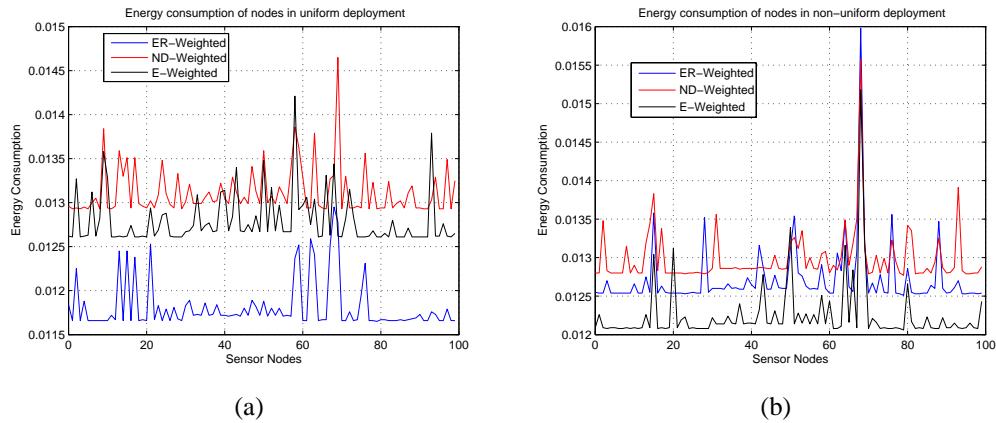


Figure 3.9: Energy consumption in heterogeneous WSN ( $k = 30\%$ ,  $\varphi = 1$ ,  $m = 20\%$ ,  $\alpha = 1$ ) for uniform 3.9(a), and non-uniform 3.9(b) deployment of nodes.

The deployment pattern of sensor nodes greatly affect the performance of the self-configuring clustering protocols. Due to the unpredictable distribution of nodes, MEAC takes into consideration the different system parameters as described in Section 3.3.3 to adapt to the deployment. The performance is evaluated by weighting the parameters according to uniform deployment of nodes as well as non-uniform deployment. We create two different scenarios of deployment; first, 100 nodes are uniformly deployed in  $100 \times 100$  meters area and second, 50 nodes are deployed in  $100 \times 100$  meters area at first and then 50 more are dropped in  $50 \times 50$  meters area of the same region to make it non-uniform.

The performance is measured in both scenarios by adjusting the weighting factors. The experiments are run by; (1) keeping the weighting factors of energy and reporting

rate (ER-Weighted) high, (2) considering only the energy parameter (E-Weighted) and (3) setting the factor of neighbor nodes and distance (ND-Weighted). Fig 3.9 illustrates the energy consumption for both scenarios.

Clearly, the energy consumption is small when the weighting factor of ER parameters is set large in uniform case as shown in Fig 3.9(a). The energy gain in considering R parameter along with E is about 12% as compared to just E-Weighted clustering that the heterogeneity-aware clustering protocol exploit [34], [36]. Fig 3.10(a) shows that the reporting rate (packets/sec) is also 7% higher in ER-Weighted approach than the E-Weighted clustering approach. Hence, by including the data rate due to multiple events detection in the clustering of sensor nodes, it not only achieves the gain in energy but also high data rate.

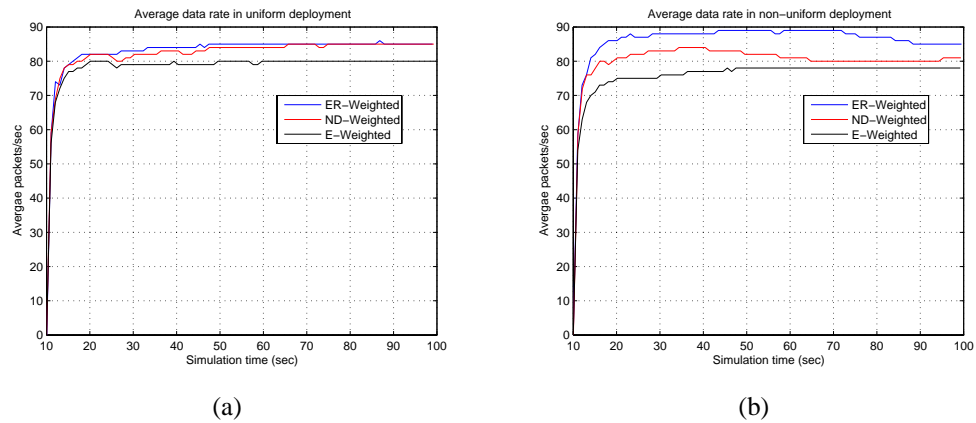


Figure 3.10: Average reporting rate in heterogeneous WSN ( $q = 30\%$ ,  $\varphi = 1$ ,  $m = 20\%$ ,  $\alpha = 1$ ) for uniform 3.10(a), and non-uniform 3.10(b) deployment of nodes.

For non-uniform case, Fig 3.9(b) illustrates that although the energy consumption is 4% lower in E-Weighted than ER-Weighted, the data rate is also lower by 15%. Therefore, the 4% gain is not actually due to the efficiency of E-Weighted approach but due to the fact of low data rate. Even if all the sensor nodes have same data rate, ND-Weighted approach delivers events at higher rate than E-Weighted as shown in Fig 3.10(b), with some extra cost of energy. Hence, there is a tradeoff between high data rate and energy consumption.

### 3.5 Summary

A number of clustering protocols have been proposed for heterogeneous wireless sensor networks. However, they do not consider the presence of multiple phenomenon in the sensor field. When a sensor node detects either multiple events or an event whose required reporting rate is higher than the other nodes, it consumes relatively higher energy. *MEAC* is a cluster-based routing protocol that considers the heterogeneity of nodes due to energy as well as multiple events. *MEAC* makes use of heterogeneity factors in such a way that energy consumption is reduced and stability period is extended compensating for reduced throughput of non-aggregated data. This conclusion was verified by the simulation experiments compared with *SEP* and *LEACH*.

## CHAPTER 4

### REAL-TIME COORDINATION AND ROUTING (RCR)

#### 4.1 Introduction

The main aim of the ERCR framework is to provide real-time coordination and routing in WSN with least energy consumption. This is achieved in RCR by incorporating a delay-constrained energy aware routing (DEAR) protocol . DEAR protocol ensures end-to-end (E2E) deadline ( $\tau$ ) for data delivery in the *responsibility cluster* with balanced energy consumption of forwarding nodes. In RCR, cluster-heads are responsible of determining their possible destination actors which is explained in Section 4.3. To conserve energy, RCR constrains the traffic volume by implementing a novel real-time data aggregation (RDA) approach presented in Section 4.4 in which cluster-heads perform aggregation such that the packets deadlines are not affected. In S-A coordination, it is not always possible that the cluster-heads have selected right actors to receive the event reports. Therefore, we provide A-A coordination in order to optimally react to the reported event. Actors coordinate with each other to respond to the event such that the whole field is targeted by the limited number of mobile actors. However, if an event region does not lie within the action coverage of any actor then RCR ensures the action by implementing event targeting procedure explained in Section 4.5.

## 4.2 Delay-constrained Energy Aware Routing (DEAR) Protocol

DEAR works on top of the clustering protocol MEAC and delivers packets to the target nodes (Sink/Actors) respecting the end-to-end (E2E) deadline ( $\tau$ ) and balanced energy consumption of the relaying nodes. Routes between the cluster-heads and actors are established through a backbone network, which is obtained by integrating the forward tracking and backtracking mechanism. Path from single/multiple hop members to cluster-heads is established during cluster formation. While the route from cluster-heads to destination sink/actor is initiated by the destination nodes in backtracking manner as described in Section 4.2.2. DEAR supports both the *semi-automated* as well as *automated architecture* of WSANs. In the presence of the sink, it adapts to the centralized version of DEAR (C-DEAR) to coordinate with the actors through the sink if required. On the other hand, when there is no sink or ignoring its presence, it provides the distributed version of DEAR (D-DEAR) for coordination among sensors and actors.

### 4.2.1 Network Model

Before going into the details of the algorithm, we model the network as a connected directed graph  $G = (V, E)$ . The set of vertices  $V$  represents the sensor nodes, where  $|V| = n$ .  $E$  is the set of directed edges such that an edge  $e(u \rightarrow v) \in E$  if  $(u, v) \in V$ . Two non-negative real value functions  $R(e)$ , the available energy resource of node  $v \in V$  on the outgoing link  $e(u \rightarrow v) \in E$ , and  $\Delta(e)$ , the delay experienced by the data packet on the corresponding link, are associated with the edges. These real values are used to compute the weight  $W(u,v)$  of the link  $e(u \rightarrow v) \in E \vee (u, v) \in V$ . The weight of an edge  $e(u \rightarrow v) \in E$  can be defined as follows:

$$W(u, v) = R(e)/\Delta(e), \quad \text{where } u, v \in V$$

Links are presumably asymmetrical because the  $R(e)$  and  $\Delta(e)$  for the link  $e(u \rightarrow v)$  may not be same while going in the opposite direction of this link  $e(v \rightarrow u)$ . The existence of alternative paths between a pair of vertices  $u, v \in V$  provides the possibility of some paths being shorter than others in terms of their associated cost. We need to find out a minimum spanning acyclic subgraph of  $G$  having high total weight.

Let  $s$  be a source node and  $d$  be a destination node, a set of links  $e_1 = (s, v_2), e_2 = (v_2, v_3), \dots, e_j = (v_j, d)$  constitutes a directed path  $P(s,d)$  from  $s \rightarrow d$ . The weight of this path is given as follows:

$$W[P(s, d)] = \sum_{e \in P(s,d)} W(e)$$

Likewise, the E2E delay experienced by following the path  $P(s,d)$  is measured as:

$$\Delta[P(s, d)] = \sum_{e \in P(s,d)} \Delta(e)$$

After the formation of clusters, we can have a vertices subset  $H$  of the set  $V$  such that the elements in  $H$  are only the cluster-heads and has an associated integral function  $hops[P(h \rightarrow target)], h \in H$ . Similarly, we obtain the set  $GS_h \quad \forall h \in H$  as the result of linking the clusters described in Section 3.3.4. Each element  $h$  of set  $H$  maintains a set of outgoing links  $OUT_h$  subset of  $GS_h$  to the single destination node either sink or actor. In the next section, we describe the way of building the set  $OUT_h \quad \forall h \in H$ .

## 4.2.2 Establishing Routes

In order to compute the delay-constrained paths efficiently, we decompose  $G$  into a minimized acyclic subgraph  $\tilde{G} = (\tilde{V}, \tilde{E})$  constituting a large acyclic region within  $G$ .  $\tilde{V}$  is the set of nodes either in  $H$  or belong to the  $GS$  sets of cluster-heads i.e.  $\tilde{V} = H \cup GS_1 \cup GS_2 \dots \cup GS_k$  for  $k$  number of clusters.  $\tilde{E}$  is the set of directed edges such that an



edge  $\bar{e}(u \rightarrow v) \in \bar{E}$  if  $u, v \in \bar{V}$ . The length of an edge  $\bar{e}(u \rightarrow v) \in \bar{E}$  may be greater than one because the members in  $GS$  may be multi-hop far from heads. For instance, an edge  $\bar{e}(u \rightarrow v) \in \bar{E}$  might exist due to some member node  $w$  such that  $u \rightarrow w \rightarrow v$ ,  $w \notin \bar{V}$ ,  $(u, v) \in \bar{V}$ . Here,  $R(\bar{e})$  is the least available energy of any node visited while traversing the link  $\bar{e}(u \rightarrow v)$  and  $\Delta(\bar{e})$  is the cumulative delay experienced by the data packet on the corresponding link.

The decomposed minimized graph  $\bar{G}$  is the backbone to establish the route from the source nodes to either the sink (*semi-automated architecture*) or the actor (*automated architecture*). In the next section, we look into the formation of the graph  $\bar{G}$ .

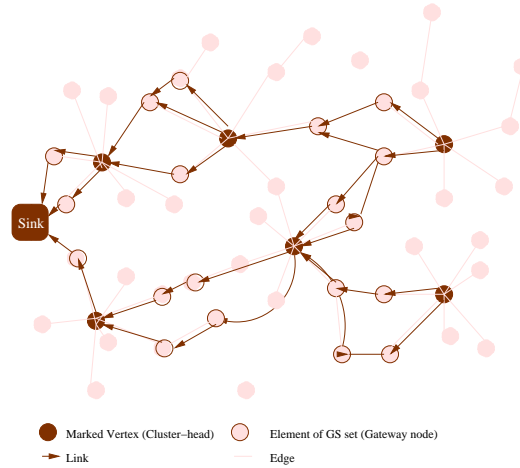


Figure 4.1: Decomposition of graph  $G$  into the minimized acyclic subgraph  $\bar{G}$  within the region  $G$ .

### 4.2.3 Sink-based DEAR (S-DEAR)

In this section, we deal with the *semi-automated architecture* and its modification for *automated architecture* is provided in the following section. We assume that the sink node is stationary like sensor nodes and the path from cluster-heads to sink is built in proactive way. Sink is the destination for all the source nodes in *semi-automated architecture*. Source to sink path is divided into two phases; source to cluster head and cluster-head to sink. The first phase builds the path from source nodes to cluster-head that is done during

the cluster formation in a forward tracking manner. The next phase deals with finding the path from cluster-heads to the sink using backtracking. It is activated initially by the sink during the network configuration phase and is updated periodically. To achieve this, the algorithm visits the graph  $G$  and marks all the vertices  $h \in H$ . A mark is associated with the life of the node, which is deleted as that vertice(node) expires. A vertex can be marked if  $h \in H$  has not been already marked or the current path delay  $\Delta[P(sink, h)]$  is less than the previously observed path delay. Once all the elements  $h$  of set  $H$  are marked, we build a path  $P(sink, h) \quad \forall h \in H$  in proactive fashion and each element  $h \in H$  set its  $hops[P(sink, h)] = |P(sink, h)|$ .

When  $h$  is marked,  $h$  adds the incoming link  $in(x \rightarrow h), x \in V$  to the set  $OUT_h$  in reverse-topological order  $out(x \rightarrow sink)$ . The incoming link  $in$  may be associated with the last marked element  $g \in H$  in the marking process or *null* if  $h$  is the first marked item and represents link to the root (sink node). This helps  $h$  to extend the set  $OUT_h$  by using the pre-determined set  $GS_h$ . The attributes of the elements of  $GS$  set contains the *AdjacentHead* ID that corresponds to  $g$ . For each element  $o(m \rightarrow g) \in GS_h$ , it searches for the match of  $g$  with the attribute *AdjacentHead* of  $o$ . If there exists such element  $o(m \rightarrow g)$  then  $h$  adds the link as  $o(m \rightarrow g)$  to  $OUT_h$  and associate an integral value  $H(o)$  apart from the other two real value functions  $R(o)$  and  $\Delta(o)$ . Hence, the edges set  $\bar{E}$  of  $\bar{G}$  can be obtained as  $OUT_1 \cup OUT_2, \dots, \cup OUT_k$  for  $k$  number of clusters.

Fig 4.1 illustrates the decomposed subgraph  $\bar{G}$  with all the possible links to the sink node. We use the term link for set  $\bar{E}$  rather than edge because vertices of set  $\bar{V}$  may be connected by some intermediate vertices in  $V$ . The set  $OUT_h$  provides all the possible routes to the sink node and we exploit the multiple entries in  $OUT_h$  to provide delay-constrained energy aware routes and implicit congestion control. We describe the criteria of selecting the outgoing link in Section 4.2.4. The cost of marking process is  $O(n)$  and, in fact, it is the actual cost of building route from source nodes to the sink node.

**Implementation:** The marking process is implemented by broadcasting *SACInfo* bea-

con in the network. That is, sink initiates the connection with the sensor nodes by broadcasting its *SACInfo* beacon periodically, where the length of period (life of mark) is larger than the *hello* beacon. This periodic beacon helps to refresh the path because the topology of the sensor nodes is dynamic. A receiving node accepts this beacon if it meets one of the following conditions:

1. It has not already received this beacon or beacon has expired.
2. Delay of this beacon is smaller than the last received beacon.
3. The number of hops traversed by this beacon are small.

When a node receives a packet it calculates the delay and forwards the request in the direction of cluster-head. While the cluster-head forwards it to its neighboring cluster-head. Hence, each cluster-head learns the loop free path to the sink node and gets the delay value and number of hops so far.

#### 4.2.4 Alternative Path Selection

In addition to energy, E2E deadline  $\tau$  is another constraint for real-time applications in wireless sensor and actor networks. Real-time event delivery is the first and foremost goal of DEAR. We have described the process of building the set of outgoing links *OUT* in the last section. The selection of a particular link  $o(m \rightarrow g) \in OUT_h, g \in H$  originating from the member  $m$  by the cluster-head  $h \in H$  is based on the criteria to balance the load in terms of delay and energy of its member nodes. Therefore, the cluster-head  $h$  selects an outgoing link  $o(m \rightarrow g) \in OUT_h, g \in H$  for which  $W[P(h, o)]$  is maximum.

Cluster-head adds the *time<sub>left</sub>* field to its data packet that is set to  $\tau$  by the source cluster-head. Each intermediate cluster-head looks for this *time<sub>left</sub>* field and selects the outgoing link accordingly by executing the above procedure. If the delay constraint can be meet through multiple links then it selects the one according to the criteria as described below:

“The link, along which the minimum power available (PA) of any node is larger than the minimum PA of a node in any other links, is preferred”.

Every receiving node then updates the  $time_{left}$  field as  $time_{left} = time_{left} - delay_s$ . It can be seen that the link selection criteria implicitly eliminates the congestion by alternating the links toward destination. Whenever a link is congested, the packet delay is increased and this delay is reported to the cluster-head in successive *hello* beacon. The weight of this link is reduced and, eventually, the cluster-head reacts to it by selecting the other available link. Hence, the congestion is avoided in addition to the energy efficiency.

### 4.3 Sensor-Actor (S-A) Coordination

An important communication paradigm in WSANs is based on the effective sensor-actor coordination. Right actions against the detected events cannot be performed unless event information is transmitted from sensors to actors. Therefore, the ultimate goal of any routing protocol in WSANs is to relay the event readings to the actors within a certain delay limit. In the classical *semi-automated architecture*, there is a central node that is responsible to collect the readings and issue action commands to the actors responsible for the action. Unlike this approach, *automated architecture* has also been realized due to the need of immediate action on the phenomena observed in the sensory field. In the former approach, sink is the destination of events reported by all the sources and is responsible to coordinate with actors. In the latter case, the mobile actors in *automated architecture* are the targets of the event readings observed by the sensor nodes and, hence, the coordination is local.

The distributed data routing approach is imperative due to the non-existence of central controller. Events detected by the sensor nodes are directly routed to the actor nodes without the intervention of the sink node. To provide the distributed routing in the *automated architecture*, ERCR incorporates the distributed version of DEAR. In A-DEAR, we de-

compose the graph  $G$  into the  $m$  number of  $\bar{G}$  subgraphs for each of  $m$  mobile actors. The idea is similar to S-DEAR described in detail in Section 4.2.3 except that we have  $m$  possible destinations. The marking process is triggered independently by all the  $m$  actors to construct  $m$  number of  $\bar{G}$  representing the paths  $P(h, actor_1), (h, actor_2), \dots, (h, actor_m) \quad \forall h \in H$ . The cost of A-DEAR is  $O(mn)$ .

In order to optimize the sensor-actor coordination in the distributed environment, the marking process also propagates the current *load* factor of the actor. The *load* represents the number of sources the actor is serving at the moment. The marking criteria in A-DEAR is modified such that  $h$  accepts the mark of an actor on the basis of its Euclidean distance. The nearest one is the best candidate for marking the element  $h$  of the set  $H$ . There might be the possibility that  $h$  lies within the action coverage of two or more actors. In such case, *load* factor breaks the ties among such candidates and less-loaded actor is selected for coordination.

Actors are location aware mobile nodes. Whenever an actor moves, it triggers the construction of graph  $\bar{G}$  in addition to the periodic reconstruction of graphs. The periodic update of graphs is required due to the highly dynamic topology of the wireless sensor and actor networks since sensor nodes may be deployed at any time or their energy deplete. Hence, the algorithm updates the path proactively to reduce the chances of path failure like the path establishment.

## 4.4 Real-time Data Aggregation (RDA)

Wireless sensor and actor networks are mostly designed to monitor and respond in various hostile environments. In order to increase the reliability of applications in such environments, sensors are deployed densely in the field. However, the dense deployment results in huge volume of traffic and create hot spots in the network because the event data is synchronous by nature triggering all the nodes at once. The raw sensed data is typically

forwarded to a sink or actors for processing which contains redundant event reports and unnecessarily consumes the scarce energy resource of the sensor nodes.

An important energy saving mechanism for sensor nodes is to exploit in-network data aggregation [46, 47, 48]. The main idea of in-network data aggregation is to eliminate unnecessary packet transmission by filtering out redundant sensor data and/or by performing an incremental assessment of the semantic of the data, e.g. picking the maximum temperature reading. It is argued that aggregation extends the queuing delay at the relay nodes and can thus complicate the handling of latency-constrained data. When we consider mixed-type traffic due to concurrent events where real-time and non-real-time traffic coexist, data aggregation becomes more challenging. In that case, we need to consider delay requirements for real-time data along with energy consumption of both real-time and non-real-time traffic.

ERCR incorporates a very simple yet practical aggregation method in which only cluster-heads involve to aggregate data from their member nodes and make sure that the end-to-end delay constraint is not violated. Hence, it does not require to build aggregation trees [46, 47, 48] because the nodes have already been configured in the form of clusters. To meet delay-constraint, cluster-head considers the event reports from its member node for aggregation as long as the deadline of any of the packet is not violated. For example at time  $t = t_1$ , a cluster-head  $h$  either detects an event itself or receives an event report from some of its member for the first time and has to send it to destination actor  $a$ . As it receives an event report, it checks for the condition

$$\Delta[P(h, a)] + \Delta(o) + T < \tau \quad (4.1)$$

where  $o \in OUT_h$  is a potential *gateway* for  $a$ ,  $T$  is the received packet delay and  $\Delta[P(h, a)]$  is the expected route delay measured during path establishment in DEAR. Let us assume that at  $t = t_2$ , it receives a packet for which the above condition is not satisfied. It

immediately transmits the aggregated packet. It is important to note that the head does not queue all the packets received during time  $t_1$  to  $t_2$  rather it aggregates the data as the packet is received. Consequently, it keeps the single aggregated packet for each different event and does not occupy the memory for aggregation. We apply three aggregation functions; maximum, minimum and average in order to better understand the event nature and thus the aggregated packets has the format  $\langle \#reports, least_{time}, max_{value}, min_{value}, mean_{value} \rangle$ . where the number of reports represents the number of packet aggregated and  $least_{time}$  is the least timestamps of all the aggregated packets.

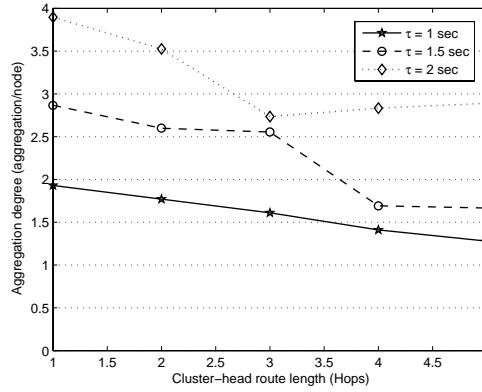


Figure 4.2: Degree of aggregation applied by cluster-heads at different route lengths from the sink node at  $\rho_o = 3 \text{ samples/sec}$ .

It is worthwhile to note that in this approach the closer cluster-heads aggregate more packets than the farther cluster-heads as shown in Fig. 4.2. This achieves the fairness among them because the closing heads have more time for aggregation due to short path delay and thereby reduce the traffic significantly near the destination.

## 4.5 Actor-Actor (A-A) Coordination

Actor-actor coordination is usually required in two scenarios; *overlapping region* and *inaccessible region*. In case of *overlapping region*, actors may have overlapping action regions due to random deployment/movement of actors. In this case, cluster-heads select

optimal actors as described in Section 4.3 by considering the distance and load on these actors. Since the load on actor changes dynamically, therefore, it is possible that an actor receiving the request might be busy in responding to other requests or have shortage of resources and can not respond to the source node. This requires the source head to update its coordinating actor. In order to update the coordinating actor, the receiving actor initiates actor-actor coordination. This is done by simply broadcasting the request to the nearby actors. An actor replies back to such requests if it is able to respond to the requesting source while attending its current sources as well. The source cluster-head then applies S-A coordination to all the replying actors and update its coordinating actor.

In second case of *inaccessible region*, actors are insufficient in the field to cover the whole region and therefore unattended regions are observed. As a result, an actor receiving request either needs to move itself toward such region if possible or trigger actor-actor coordination to relocate some actor in that region. This is a controlled targeting i.e. actor moves toward the requesting source such that the cluster-head is within the boundary of its action. However, this movement is possible only if the moving actor can attend its current request as well as the request from the inaccessible region. Hence, as an actor receives a request from a sensor node that lies outside of its coverage area, it starts the event targeting procedure.

### 4.5.1 Event Targeting

We assume that all the actors have the same circular action range of radius  $R$ , which is limited and different from the transmission range. When an actor  $a_i$ , residing at the point  $I(x_i, y_i)$ , receives a request from the source  $s$  at position  $J(x_j, y_j)$ , it calculates the distance  $D(I, J) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$  from the source. If  $D(I, J) \leq R$ ,  $a_i$  is able to respond to the request. However, if  $D(I, J) > R$ , there are two possible scenarios. First, actor  $a_i$  needs to move  $D(I, J) - R$  units of distance toward the target location at some new point  $\bar{I}(\bar{x}_i, \bar{y}_i)$  so that the coverage is provided to  $s$  as shown in Fig. 4.3. The new actor location  $\bar{I}$  can be



computed as follows

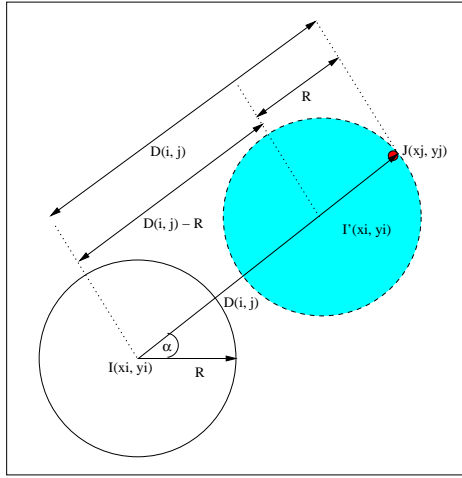


Figure 4.3: Positioning actor toward uncovered source.

Taking  $I(x_i, y_i)$  as the center of axis on the  $XY$  plane, we can find the coordinates  $\bar{x}_i$  and  $\bar{y}_i$  as

$$\bar{x}_i = x_i + (D(I, J) - R) \times \cos(\alpha)$$

$$\bar{y}_i = y_i + (D(I, J) - R) \times \sin(\alpha)$$

where  $\alpha$  is the angle between the vector  $X(R, 0)$  along x-axis and the vector  $A(x_j - x_i, y_j - y_i)$ .

To find out the value of  $\alpha$ , first compute the length  $|X|$  and  $|R|$  of two vectors  $X$  and  $R$ .

$$|X| = R, |A| = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

Normalize each vector (unit vector)

$$X_u = (R, 0)/|X|, A_u = (x_j - x_i, y_j - y_i)/|A|$$

Hence, the angle between  $X$  and  $A$  is:

$$\alpha = \cos^{-1}(X_u \cdot A_u) \tag{4.2}$$

---

```

    /*Pseudo-code executed by the actor  $a$  to determine the coverage of
    source  $i$ .*/
    TargetEvent(i)
         $d(i, a) = \sqrt{(x_a - x_i)^2 + (y_a - y_i)^2}$ ;
         $C_a = coverSource((d(i, a) - R), (x_i, y_i))$ ;
        /* New center position  $C_a$  of  $a$  if it can move  $d(i, a) - R$  toward
         $(x_i, y_i)$ .*/
        MoveOK = true;
        for(each  $s \in sources_a$ )
            MoveOK = canCover( $C_a, R, (x_s, y_s)$ );
            if (MoveOK = false)
                StartAACoordination(i);
                exit;
            end //if
        end //for
        moveTo( $C_a$ );

```

---

Figure 4.4: Algorithm of action coverage of a source.

However, the displacement is only possible if the movement of  $a_i$  does not leave the presently attended sources uncovered. In the second case, if the displacement of  $a_i$  is not valid then it initiates the actor-actor coordination algorithm.

The coordination among actors to move an actor in the uncovered region is triggered by broadcasting the *relocate* message that contains the location of the source  $s$ . Each actor receiving this message runs the procedure outlined in Fig. 4.4 to check for the possibility of attending the source. An actor  $a_j$  sends back the *relocation-ok* message to  $a_i$  if it is able to move toward the source. This message contains the residual energy of actor  $a_j$  as well as the total number of sources being attended. If the field is being operated by sufficient number of actors such that all the sources can get response from the actors then the actor  $a_i$  receives *relocation-ok* message from some of the neighboring actors. Consequently,  $a_i$  forwards the request to the actor covering lesser number of sources and has the highest residual energy.

## 4.5.2 Example Scenario

Fig. 4.5 illustrates the target tracking procedure in two different scenarios. Sensor X subscribed the actor A because it is the nearest one and similarly sensor Y subscribed the actor B. Although node X reports its event to A but its out of the action range of A. Actor A runs the target tracking procedure and finds that node X can be covered with its existing sources. Therefore, actor A moves  $(D(A \rightarrow X) - R)$  units toward the node X. This is depicted in shaded area covering X in Fig. 4.5.

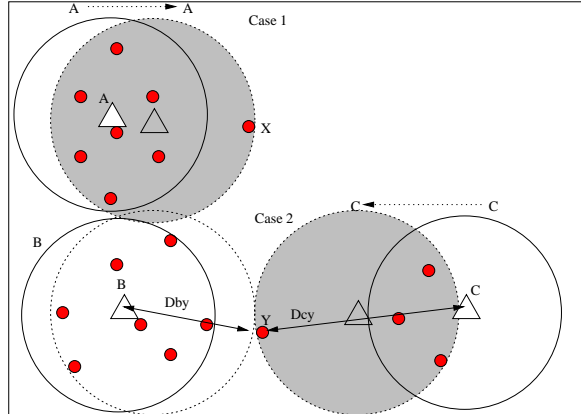


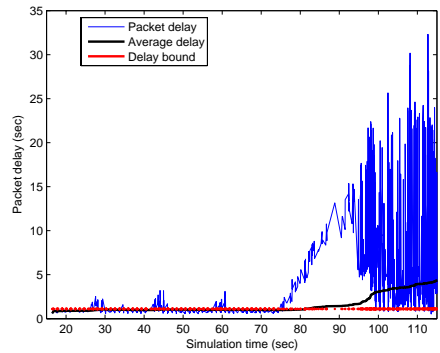
Figure 4.5: A-A coordination to target the event area.

The case 2 requires A-A coordination because actor B is not able to serve node Y. If it moves to cover the event area of Y then it leaves the current sources uncovered. Therefore it coordinates with A and C and finds C as an appropriate actor to cover Y. Although  $D_{by} < D_{cy}$ , C is the right actor because the position of its existing sources permit it to attend.

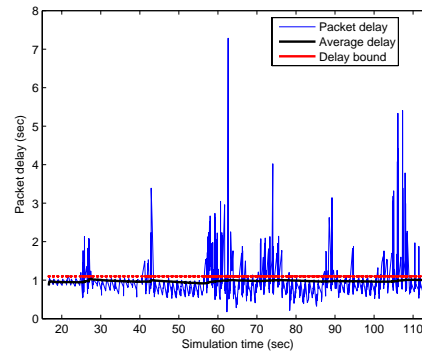
## 4.6 Performance Evaluation

The performance of the framework is evaluated by using the network simulator ns-2 [106]. We report performance results of energy, packet delay and delivery ratio as the three main evaluation parameters of of sensor-actor coordination, actor-actor coordination and aggregation. We create different scenarios to evaluate the performance metrics. In

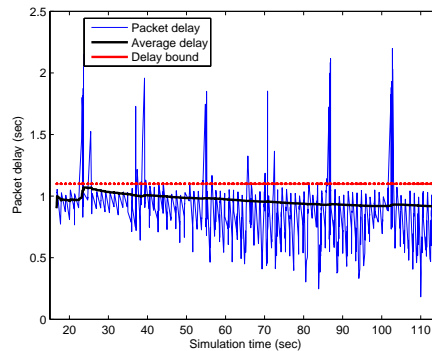
the first scenario, 200 sensor nodes and 4 actors are randomly deployed in  $150 \times 150$  field to evaluate the performance gain in S-A coordination. However, in the second scenario for A-A coordination, we consider two cases. In first case, we keep the number of actors constant and increase the number of sensors gradually. In the second case, actor nodes are increased for the same dimension of field and number of sensor nodes. The first case consists of 4 actor nodes and 100 to 300 sensor nodes. While in the second case, actor nodes vary from 2 to 7 for 200 sensor nodes deployed in  $150 \times 150 m^2$  field.



(a) Delay in semi-automated architecture



(b) Delay in immobile A-A coordination scenario of automated architecture



(c) Delay in mobile A-A coordination scenario of automated architecture

Figure 4.6: 150 sensor nodes and 4 actors randomly deployed in  $100 \times 100$  and action range of actors set to 25 meters.

### 4.6.1 S-A Coordination

To evaluate the S-A coordination performance, we consider three different network configurations; *semi-automated*, *automated* with simply S-A coordination and *automated* with mobile A-A coordination. In *semi-automated* scenario, all the packets are transmitted to the actors through the centralized sink node. Contrarily, *automated* configuration delivers the packets from source nodes to the actors directly without the intervention of sink node. However, we classify the *automated* architecture into two further configurations. S-A coordination in which sensors coordinate with actors to select appropriate destination actors but the actors do not coordinate with each other to ensure response to the source nodes lying out of the action coverage. In the second case of mobile A-A coordination, actors coordinate with each other in addition to S-A coordination so that an actor moves toward the cluster-head if it is out of the action range of any actor.

Fig. 4.6 illustrates the average delay at  $\tau = 1 \text{ sec}$  for the three configurations. We assume 5% tolerance in delay and, therefore, plot the packet delay and average delay for a delay bound of  $1.05 \text{ sec}$ . The packet delay in *semi-automated* architecture hardly achieves the deadline and the network becomes congested after some time that results in extremely large packet delay as shown in Fig. 4.6(a). The average delay in *semi-automated* is 5 times greater than both the scenarios of *automated* architecture. However the two configurations of *automated* architecture differ in miss-deadline ratio. This ratio is about 14% in S-A coordination while just 5% with mobile A-A coordination. Hence, the adaptive mobility of actors in ERCR improves the performance further in addition to S-A coordination.

Similarly, the packet delivery ratio in *automated architecture* is higher than *semi-automated architecture*, which is approximately 30% higher as shown in Fig. 4.7.

We assume that the actors have limited action range and they can not respond to the sensor nodes if they lie out of their action coverage. Such requests may not be fulfilled and therefore assumed as unattended requests. The mobile A-A coordination is used to minimize such requests by moving the actors sufficiently toward these nodes. Intuitively,

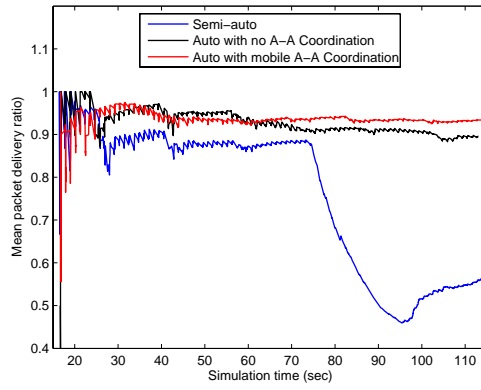
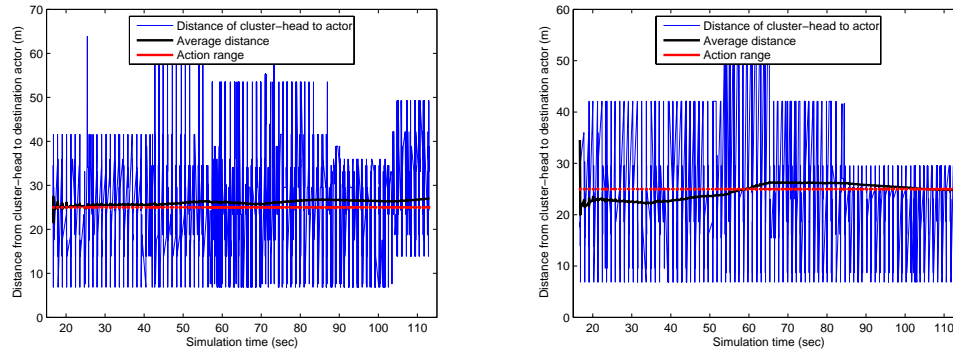


Figure 4.7: Packet delivery ratio in *semi-automated architecture* and *automated architecture* for 2 samples/second triggered by 3 phenomenon nodes of event radius 50 meters.



(a) Distance of cluster-heads from coordinated actors in simple S-A coordination. (b) Distance of cluster-heads from coordinated actors with S-A/A-A coordination.

Figure 4.8: 200 sensor nodes and 4 actors randomly deployed in  $150 \times 150$ .

increasing the responsiveness of the actors. Fig. 4.8 shows the distance of the cluster-heads to the coordinated actors when the action range is 25 meters. It is obvious that the average distance in immobile coordination given in Fig. 4.8(a) is beyond of action range. However, the average distance in mobile actors scenario is below the action range for most of the time. The unattended request ratio is 50% in the former case and 25% in the latter case, which is computed 11% by using Eq 5.1. It happens when the number of actors operating in the field are lesser than the required number of actors to cover the whole field. However, with mobile A-A coordination, we can significantly achieve better performance with smaller number of actors.

## 4.6.2 A-A Coordination

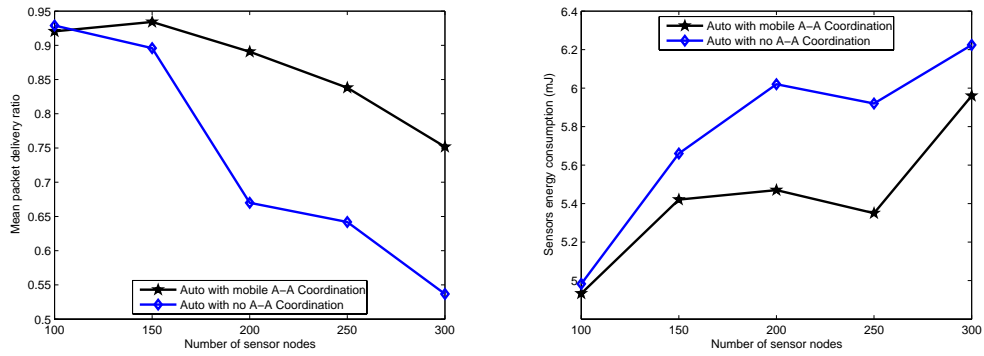
In this section, we show the performance achieved when employing mobile A-A coordination in WSN presented in Section 5.3.2, which is not possible with simple S-A coordination.

Fig. 4.9 shows the delivery ratio in first scenario. It is observed that the scalability is achieved by employing A-A coordination. The delivery ratio drops down to 17% by increasing the sensor nodes 3 times and dimensions of the field to 2 times. However, this value is approximately 50% when there is no A-A coordination and actors stay at their initial locations. Hence the gain in delivery ratio due to mobile A-A coordinations is 3 times which is a worthwhile improvement. Similarly the energy consumption of not only the sensor nodes but also the actor nodes is reduced as shown in Fig. 4.9(b) and Fig. 4.9(c) respectively.

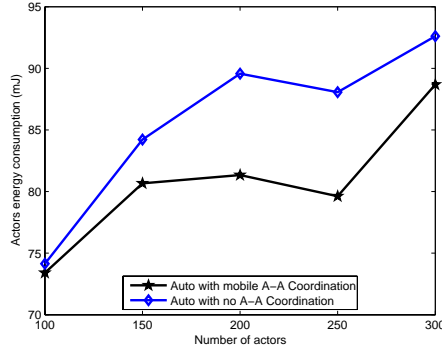
In the second scenario, we show that higher delivery ratio is obtained at smaller number of actors when employed A-A coordination. We achieve maximum delivery ratio of 90% at  $m = 4$ . The delivery ratio for the same number of actors in simple S-A coordination is approximately 68%, which is significantly lower. Even the delivery ratio for  $m = 7$  is 82% which is still lower than the delivery achieved in A-A coordination for  $m = 4$ . Intuitively, we achieve the delivery in A-A coordination at lower number of actors which reduces the cost of application because the cost of actors is the major contribution in WSNs applications cost. Fig. 4.10(b) and 4.10(c) show the gain in energy of both sensor nodes and actors respectively. It is clear that we save the energy of sensors and actors by approximately 12%.

## 4.6.3 Aggregation

Data aggregation in WSNs is applied to reduce the volume of traffic in network and eventually conserve the energy of sensor nodes. Although, it achieves the energy efficiency but increases the packet delay which is usually not suitable for real-time appli-



(a) Packet delivery ratio with and without mobile A-A coordination. (b) Mean energy consumption of sensor nodes.



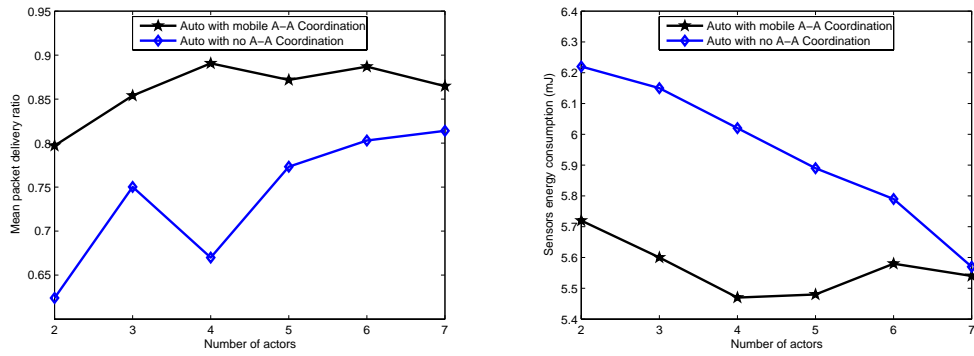
(c) Mean energy consumption of actor nodes.

Figure 4.9: Performance for different number of sensor nodes with 4 actor nodes randomly deployed.

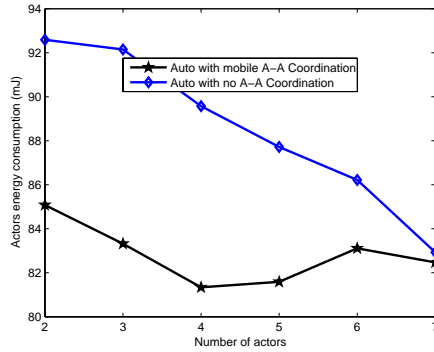
cations. However the data aggregation approach presented in Section 4.4 is adaptive to the application deadlines. Fig. 4.11 reflects the adaptivity of RDA, where the average delay remains lower than  $\tau$ . This is not applicable at shorter deadline and higher traffic rate ( $\tau \leq 0.5 \text{ sec}$ ,  $\rho_o = 4 \text{ samples/sec}$ ) where the average packet delay is much higher than  $\tau$ . We consider the traffic in terms of event sampled by a sensor node in one second, where the size of a sample is kept 100 bytes. It is due to the fact that lesser or no packets are aggregated and larger number of packets are transmitted. Therefore, sensor nodes can not deliver the packets at higher rate within the given short deadline. As a result, aggregation can not be accused for such scenarios. This situation is improved at  $\tau > 0.5 \text{ sec}$ . Fig. 4.13 shows the effect of delay bound on the aggregation achieved and packet delay.

The adaptivity of RDA can be justified in Fig. 4.12 too. It shows that the mean





(a) Packet delivery ratio with and without mobile A-A coordination. (b) Mean energy consumption of sensor nodes.



(c) Mean energy consumption of actor nodes.

Figure 4.10: Performance for different number of actor nodes deployed with 200 sensor nodes in  $150 \times 150$ .

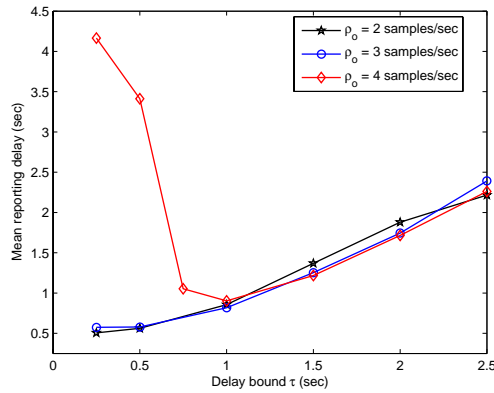


Figure 4.11: Event reporting delay for different values of  $\tau$ .

aggregation is lower at smaller  $\tau$  but higher at larger  $\tau$ . Since the higher volume of traffic introduces larger packet delay, it must be considered in aggregating and routing real-time data. This effect is also supported in RDA and it is witnessed in Fig. 4.12 that by increasing

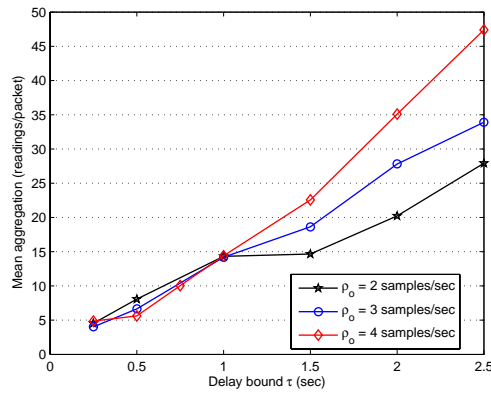


Figure 4.12: Mean aggregation achieved at different values of  $\tau$ .

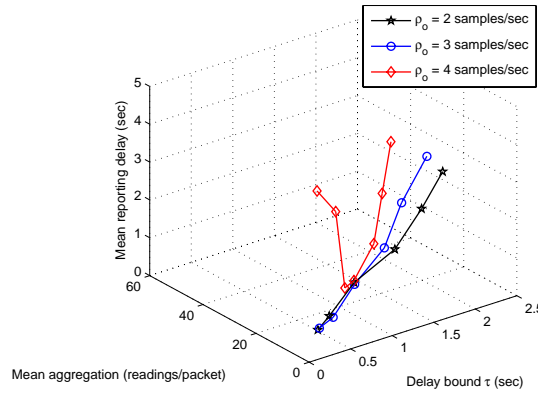


Figure 4.13: Event reporting delay with different values of aggregation achieved applied for different values of  $\tau$ .

traffic 2 times i.e. from  $\rho_o = 2 \text{ samples/sec}$  to  $\rho_o = 4 \text{ samples/sec}$ , the aggregation is increased 60% because RDA gets less time for aggregation due to large communication delay. Hence, the integrated RDA approach of ERCR is not only adaptive to different deadlines of the packets but to the different data rates in the network as well.

At shorter deadlines, lesser number of packets are aggregated. Eventually the number of packets transmissions are increased that result in lower delivery ratio due to high volume of traffic. This fact is shown in Fig. 4.14. At  $\rho_o = 2 \text{ samples/sec}$ , the delivery ratio is initially  $\approx 10\%$  higher than at  $\rho_o = 4 \text{ samples/sec}$ . However, as  $\tau$  reaches to 0.75 sec, the delivery ratio is improved for  $\rho_o = 4 \text{ samples/sec}$  and the difference of delivery ratio is reduced to 5%. By the suitable selection of  $\tau$ , we can achieve the required packet delivery.

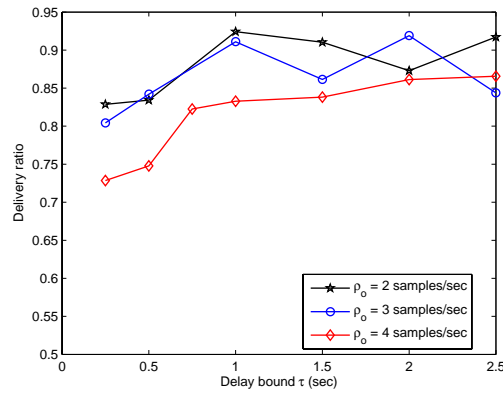


Figure 4.14: Event delivery ratio for different values of  $\tau$ .

Similarly the energy consumption is reduced by applying aggregation as shown in Fig. 4.15.

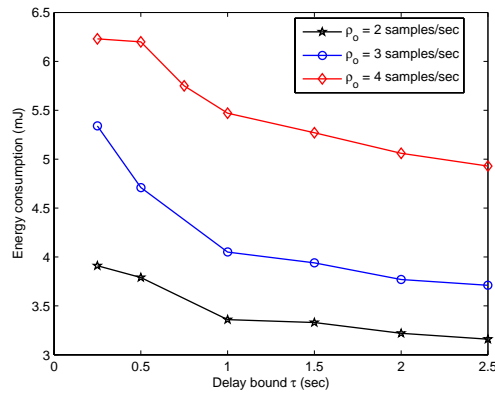


Figure 4.15: Mean energy consumption for different values of  $\tau$ .

## 4.7 Summary

In this chapter, we propose a cluster-based real-time coordination and routing framework. The framework achieves energy efficiency and real-time data routing support at different levels. To achieve energy efficiency, sensor nodes coordinate with each other to form clusters and a node is elected as a cluster-head which has higher energy level but lower traffic rate. For real-time data routing, cluster-heads select the next forwarding node

such that the delay-bound of packet can be met efficiently. At second level, it presents a real-time data aggregation model which is adaptive to the packet deadlines. It helps in routing data within the given delay bound ( $\tau$ ) and conserving the nodes energy by reducing the traffic volume. Thirdly, it provides sensor-actor coordination so that the event reports are routed directly to the actors without involving the centralized sink node. This coordination not only saves the energy of sensor nodes but also reduces the packet delay for real-time event routing. At higher level when the data is received by the actors, they coordinate with each other to react in the event area optimally. In A-A coordination, an actor moves toward the event area if it does not come in the action coverage of any actor due to their limited action range. As a result, future event reports are sent to the actor which can cover the event area. Hence, the responsiveness of the actors is improved that improves the performance of the entire application. Simulation experiments show that RCR achieves its goal of real-time event delivery to honor the realistic application-specific delay bound and energy efficiency.

## CHAPTER 5

### RAT: ROUTING BY ADAPTIVE TARGETING

#### 5.1 Introduction

Generally, when sensors detect an event (a change in the environment), they communicate with each other to confirm the event or generate new data based on the event, which is reported to the sink/actuator. There are two ways to disseminate the event data, namely push-based and pull-based. In push policy, event information is sent out to the sink without explicit requests for it. In contrast, with pull based approach events are reported in response to explicit requests received from sink/actor. In this study, we refer to the communication initiated by the actors to disseminate their location and their acting capabilities as pull communication. Pull-based approach can be used either for unicast or for broadcast. When used for broadcast, pull is also referred to as interactive or on-demand broadcast. It is more efficient when the occurrence of event is very rare and applications are interested in a particular event scenario.

It has been long argued that push is more beneficial than pull since it provides scalability and well suits to monitoring application, which demands continuous event reports [70]. It is unnecessary to report a particular event scenario which is not of interest for the applications. One focus of this study is to control the level of data communications through pull/push coordination. There have been a number of studies for developing push (data collection) [53, 55, 58, 68] based routing in WSN as well as pull based (querying)

approach [56]. Some hybrid push-pull techniques [70, 71] are suggested to overcome the drawbacks of pure push or pure pull based approaches in WSN. However, there exist no integrated push/pull [67] solution for distributed processing of a query in wireless sensor and actuator networks.

We propose a solution where the network can respond to emergencies without coordinating with a centralized control. In particular, we propose routing by an adaptive targeting (RAT) protocol that allows sensor-to-actor communication and dynamic coordination of actors in response to emergencies. RAT comprises of two component; Delay-constrained geographical-based routing (DC-GEO) and Integrated Pull/Push (IPP) coordination. DC-GEO relays the packets in greedy mode such that delay constraint can be meet as well as energy consumption of forwarding nodes is balanced. In IPP, actor nodes subscribe to specific events of their interest in the field and sensor nodes disseminate the event readings to subscribed actor for a time period of subscription life. As a result, actor nodes do not require to send a query every time they need event readings. Similarly, sensor nodes pushes the data as long as there is a subscribed actor interested for the observed event.

RAT exploits the mobility of actor nodes to form dynamic *responsibility clusters* that ensures a specific response time to emergencies. A *responsibility cluster* in the field consists of a group of sensors observing events and at least an actor responding to the events. The whole field is divided into such *responsibility clusters* to collectively achieve the goal of applications. To achieve such responsibility zones in the field, the actor position may need to be changed to keep them close to emergency areas and reduce the response time.

The remainder of the chapter is organized as follows. A delay constrained geographical-based routing mechanism is proposed in Section 5.2. Section 5.3 describes the Pull/Push Coordination in WSN. Performance evaluation and results are considered in Section 5.4. Finally, the chapter is concluded in Section 5.5.

## 5.2 Delay-constrained Geographical (DC-GEO) Routing

### Mechanism

Geographic forwarding is so called a stateless routing algorithm based on only the location information of the neighboring nodes, which is more attractive for resource constrained nodes. Sensor nodes are localized in the framework using TSL and is not a concern in greedy mode routing. More specifically, a forwarding node relays an incoming data packet to the neighbor who is the nearest to the destination among its one-hop neighbors. The forwarding set (FS) of a given node is a set of the neighbor nodes, which are more close to the destination. Generally, the FS is a subset of neighboring nodes, according to the given location of the single sink in WSNs. However, in WSAN, multiple actors, placed or moved at different geographical positions in the field, are the destinations of the sensors readings and, therefore, a relaying node keeps all the neighbors in FS initially.

#### 5.2.1 Delay Measurement

When nodes are initially deployed in the field, every node  $i$  broadcasts *presence beacon*, which is added in FS by all the nodes that receive it. Consequently, FS is maintained by all the nodes. A node that receives the beacon, computes the delay  $T$  of the packet and records the value in FS along with the sending node ID.

The delay  $T$  in transmitting a packet from one node to another in its neighbor is measured by the following factors; queue waiting time, MAC delay, propagation time and receiving delay represented by  $T_q$ ,  $T_{Mac}$ ,  $T_{Prop}$  and  $T_{Rec}$  respectively. The wireless channel is asymmetric. Therefore, the delay is measured at both the sender and receiver. Sender measures the delay  $T_s$  until the start of transmission that includes the queue delay as well as the MAC contention delay. Whereas, the receiver adds the factor  $T_r$  as sum of propaga-

tion delay and receiving delay to get the total packet delay  $T$ :

$$T_s = T_q + T_{Mac}, \quad T_r = T_{Prop} + T_{Rec}$$

The hop latency  $T$  can be computed as sum of  $T_s$  and  $T_r$ :

$$T = T_s + T_r$$

The delay is measured by exchanging the *presence beacon*, which represents the load on that node. The elements of FS has the attributes  $\{ID, T, energy, distance\}$ , where *energy* and *location* are reported in *presence beacon*. However, this delay value for a particular neighbor node  $j$  varies under different traffic load. A large delay value reflects a high traffic volume a node is experiencing. As a result, a periodic beacon keeps the neighbors informed about the current state of the node: energy, location and traffic load in terms of delay.

### 5.2.2 Delay-constrained Forwarding Set (DFS)

RAT focuses on applications that require time critical actions. Such as monitoring a strategic area where emergencies should be responded as quickly as possible. In this regard it is more efficient to report the abnormal event readings in WSN to nearby actors, rather than a stationary sink node that further needs to communicate with the actor. We define the delay tolerance of events as  $\tau$  and try to minimize  $\tau - \epsilon$  where  $\epsilon$  represents the time that is required by the actor to respond to sensor input. The value of  $\epsilon$  depends on two factors; first, the time required by an actor to get ready and take certain action, second, how many action requests are in the waiting queue. The process of forwarding the data packet toward actor consists of two phases:

- Construct a delay-constrained forwarding subset (DFS) from the set FS such that



the given delay constraint  $\tau$  can be met. The source node set the time to live (TTL) field in the data packet to  $\tau - \epsilon$  and each forwarding node updates the TTL field by deducting the traversed hop delay. The packet is considered obsolete and dropped whenever TTL reduces to 0.

- Balance the load on the nodes in the DFS in terms of energy consumption by selecting the forwarding node from the set DFS that has the higher energy level.

Initially, all the neighbors nodes are included in FS. More specifically, multiple mobile actors in the field are the possible destinations of the packets generated by the sensor nodes. However, this approach is modified in order to consider the response time demanded by the application and efficient energy consumption of nodes. Hence, a forwarding node is selected such that the response time can be met as well as energy consumption is balanced. For each possible destination actor, a node  $i$  decomposes its set FS into subsets  $DFS_i^a$  for each destination actor  $a$  as:

$$DFS_i^a = \{j | j \in FS_i \wedge D(i, a) > D(j, a) \wedge \frac{TTL}{T(j)} > \frac{D(i, a)}{D(i, j)}\}$$

That is, a node  $i$  builds a subset  $DFS_i^a \subseteq FS_i$  taking delay and distance into consideration. Where a node  $j \in FS_i$  is closer than  $i$  to a destination actor  $a$  and also the expected number of hops with respect to delay from  $i$  to  $j$  is larger than the expected number of hops with respect to the distance between  $i$  and  $j$ . Where,  $T(j)$  is the time required to relay the packet to node  $j$ ,  $D(i, a)$  is the distance between the node  $i$  and actor  $a$  and  $D(i, j)$  is the distance between the node  $i$  and node  $j$ . From  $DFS_i^a$ ,  $i$  selects the next node which has the highest residual energy.

At larger distance between the node  $i$  and  $j$ , it is highly likely that the delay is also larger. It is due to the long propagation delay and the intermediate nodes interfering the transmission between them. At larger distance between sender and receiver, more nodes

share the same channel and thereby delaying the transmission. However, it may not be true when the traffic is lower between two distant nodes and intuitively lesser medium access delay is experienced. In such case, packets mostly experience the propagation delay which is not dominant at different distance. Hence, such a long distant path is better than the shorter path having larger medium access delay since propagation delay is much lesser than the medium access. Consequently, traffic between two nodes affects the communication delay more than the distance between them. Therefore, DC-GEO modifies the greedy routing to adapt to different circumstances in the network. It selects the forwarding node which is closer to the destination but guarantee the in-time packet delivery. Intuitively, its rationale is to relate the delay with distance in terms of packet relaying speed but adaptive to the current traffic conditions. Fig. 5.1 justifies our delay-constrained routing. It is quite often that the packets follow long routes but they are selected according to load of traffic. It is not wise to always select the shortest route that becomes bottleneck and is sooner congested incurring larger delay.

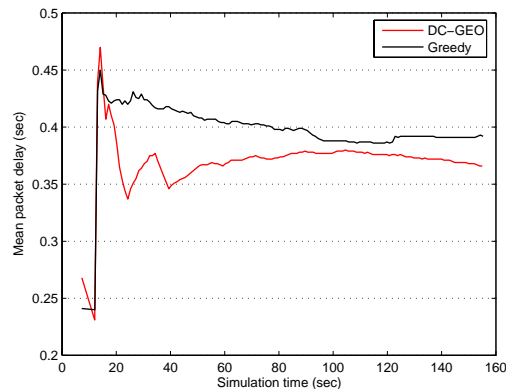


Figure 5.1: Delay in DC-GEO vs pure greedy routing.

A node  $i$  forwards the request progressively to its neighbor node  $j \in DFS_i^a$ , which has the highest energy level among the other elements in  $DFS_i^a$ . However, if  $DFS_i^a = \phi$  then it cannot forward the incoming data packet further; the packet is stuck in a local minimum where FS contains no element for the destination target as shown in Fig. 5.3 that node 1

and 6 are stuck and find no element in FS. In such a case, there exists a void region \* and greedy geographical approach fails.

### 5.2.3 Dual-Forwarding Recovery

The routing protocol has to recover when it ends or stuck in a void region. Different recovery methods from the void region have been proposed in [66, 55, 69]. HGR [66] is a hybrid geographic and virtual coordinate (VC) routing. VC of the nodes are computed based on their distance from fixed reference points. It is argued that virtual coordinates system (VCS) is not susceptible to conventional *voids* because the coordinates are based on connectivity and not physical distance [66]. It exploits virtual coordinates in recovery mode in place of geographical location. However this approach requires additional anchor nodes as reference stations to compute the virtual coordinates, which is cost inefficient. GPSR [69] follows the single route along the perimeter of the void region. For a highly non-uniform or terrain deployed nodes, it is highly likely that successive void regions exist as shown in Fig. 5.3, which introduce significant delay.

#### Routing in Recovery mode

We overcome the existence of a void region by using *dual-forwarding* geographical approach. In this approach, a packet is relayed by forwarding dual copies toward a node on left and a node on right along the perimeter of the void region. This increases the reliability and improves the response time in case of highly irregular topology. Hence, node  $i$  sets a *recovery flag* in the packet header and forwards the request to its left node  $j$  such that  $j$  is the closest to destination among the nodes on the left and similarly, a copy of the packet toward the node  $k$  on its right side. The nodes  $j$  and  $k$ , receiving a request from  $i$ , transmit the packets individually and selects the forwarding node other than  $i$  by

---

\*The region is called void if a node fails to find a forwarding node in that region which is more close to the target than itself, by using the greedy approach.

looking at *recovery flag*. These transmissions might reach to a single node  $x$  after crossing the void region or they may follow disjoint paths. If a node receives the same packet, it drops the duplicate.

The proposed recovery mode shows that the routing is highly resilient since at some point the packet is almost flooded in an extremely hostile environment. For example, at first failure, two identical copies of a packet are relayed independently. For another successive failure of both packets, four identical copies try to reach the destination and so on, which becomes almost controlled flooding at some point. However, nodes keep track of the serials of recovery packets and do not retransmit it if already transmitted in order to avoid from loops. The maximum cost of this approach is  $n - 1$  transmissions if the destination is unreachable. It is important to note that this cost is only for the first packet since the source keeps track of the unsuccessful forwarding nodes which are not considered in next transmissions. This is explained in next section. It might be a good idea to send a dummy packet as part of network configuration to know about the possible void regions which can be prevented preemptively.

### **Escaping from Void Regions**

We maintain a set  $VS$  of *void nodes* which have detected void regions in order to keep track of the nodes failing to relay the packet in greedy mode for a particular destination actor. The purpose of maintaining this set is to escape from the void region for future transmissions since recovery in void region generates extra traffic and incur some overheads. However, it is possible that a *void node* for a particular destination actor is a forwarding node for some other actor. Therefore, a member of  $VS$  have two attributes; the void node ID (NID) and destination actor ID (AID). On detection of void node, an entry of that node associated to the destined actor is made in  $VS$ . Before picking an element  $i$  from DFS, the relaying node lookups  $VS$  for  $i$  and if it is found then it is not selected as forwarding node as outlined in Fig. 5.2.

In order to form VS, we assume symmetric links between the neighbors and, therefore, the source node is able to hear the transmission of  $i$  when destined for  $j$  and  $k$ . By reading the *recovery flag* set in the packet header and the same destination actor  $a$  of the transmitted packets to  $j$  and  $k$ , source node infers  $i$  as a *void forwarding* node for  $a$ . It puts an entry  $\langle i, a \rangle$  in VS and do not consider it for further transmissions.

Nodes  $j$  and  $k$ , knowing that they are relaying the recovery packet received from  $i$ , keep track of such packets in the form of  $\langle i, a \rangle$  in *recovery table*. The process of relaying the packet continues until it is either successfully delivered to  $a$  or no forwarding node is found even in recovery mode. The former case represents successful route establishment while the latter case shows that there exist no route for  $a$ .

Let us assume that the packet has arrived at the node  $z$  from the previous node  $y$ , which could not find any forwarding node. It might be due to existence of its neighbors either in VS or *recovery table*. It infers that there exist no route for  $a$  and broadcast the *RFailure beacon*. All the nodes hearing this beacon adds an entry  $\langle z, a \rangle$  in VS. Node  $z$  also removes the entry  $\langle y, a \rangle$  from *recovery table* which had sent it the packet. In addition to  $y$ , all the nodes hearing *RFailure beacon* will add  $z$  in their void sets (VS). Now if the receiving nodes have entries in their *recovery table* for  $a$ , they also broadcast the *RFailure beacon*. The process continues until a beacon is received by the source node. Source node marks the nodes as void nodes from which it receives *RFailure beacon*. However, the nodes marked as void for destination  $a$  are removed if the mobility of  $a$  is observed.

To terminate a packet that can never reaches its destination, it contains time to live (TTL) and maximum number of hops ( $H_{max}$ ). As described earlier, TTL is set to  $\tau - \epsilon$  and is decremented at each intermediate hop. The value of  $H_{max}$  is computed on the basis of distance between source node and destination and the assumption that each forwarding node is half of the transmission radius  $r$  closer to the destination  $a$ , on the average, than the source node  $i$ . Therefore, we can compute the limit on number of hops as  $H_{max} = 2 \times D(i, a) / r$ . The packet is dropped whenever one of the two values TTL or  $H_{max}$  becomes

---

```

/*Relaying node executes this procedure to select a forwarding node
from the set DFS for a destination actor a.*/
SelectForwarding()
  f = null;
   $E_{max} = 0;$  //Maximum energy
  for (each i  $\in$  DFS)
    /*  $E_i$  is the energy of node i */
    if ( $E_{max} < E_i$  &&
        FindVN(i, a, VS) == false)
      f = i;
       $E_{max} = E_i;$ 
    end //if
  end //for
return f

```

---

Figure 5.2: Algorithm of the DC-GEO protocol for selecting forwarding node.

0.

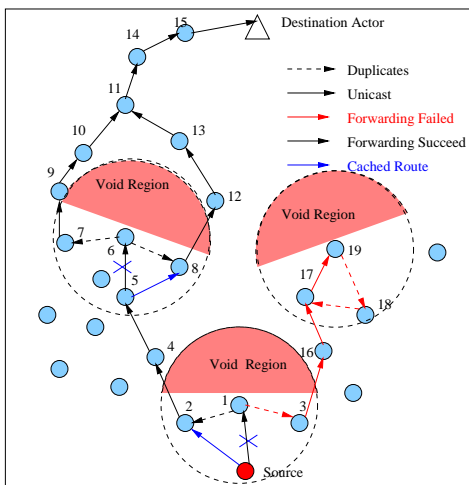


Figure 5.3: Packet forwarding from source to destination actor in the presence of void region in a highly non-uniform deployment of nodes.

### Example Scenario

Fig. 5.3 demonstrates the DC-GEO data routing approach with void regions. Source node selects node 1 as the forwarding node initially when the set *VS* is empty. However, node 1 detects a void region due to non-existence of any forwarding node in set *DFS*. Node

1 transmits the request to the node 2 on its left and node 3 on its right and is therefore added by the source in VS. Node 2 and 3 forward the request to 4 and 16 respectively, which are closer to the destination actor D. However, the path is established through node 2 but not node 3 as shown in Fig. 5.3. The node 3 is not considered because source node has received *RFailure beacon* from 3 which was initiated by 19 as described in Section 5.2.3 and is also added in VS. Hence, the future requests are forwarded through the node 2.

#### 5.2.4 DC-GEO Overheads

Each node  $i$  broadcasts *presence beacon* periodically to maintain the neighbors list that contains the location information  $(x_i, y_i)$  and residual energy  $E_i$ . The beacon is 16 bytes long. The frequency  $\phi$  of the beacon is small because we assume that the sensor nodes are stationary and the neighbors list remains valid as long as the nodes are alive. Therefore, the overheads in terms of bandwidth for maintaining the list of  $n$  neighboring nodes sharing the same channel is  $(128 \times \phi \times n \text{ bits/sec})$ . At  $n = 20$ ,  $\phi = 0.2$  i.e., a beacon is transmitted after 5 sec, the overhead is only 512 bps which is very small over 1 Mbps channel capacity. Similarly, the overhead in terms of energy consumption is  $(512 \times E_{elec}) + (128 \times 0.2) \times E_{amp}$  which is negligible. where  $E_{elec}$  is the energy consumed per bit by the transceiver circuitry of the nodes and  $E_{amp}$  is the energy consumed per bit by the sender for signal amplification. If there is no incremental deployment then we further decrease the frequency  $\phi$ .

### 5.3 Location-aided Integrated Pull/Push (IPP) Coordination

In WSAN, appropriate actions corresponding to the sensed phenomenon cannot be performed unless event information is transmitted from sensors to the actor capable of taking action in the event area. Unlike sensors with limited resources, however, actors have abundant resources and are mobile in the area. Actors can have long range transmission

as opposed to sensors and they can transmit at larger distance possibly ranging the whole field. Moreover, it is assumed that each actor has a limited action range ( $R$ ) which is different from the transmission range and let us not confuse the two different values. It means that an actor can perform an action within  $R$  units of distance in the field. A *sensor node is said to be covered by an actor if its distance to that actor lies within the limit  $R$  otherwise it is assumed to be uncovered*. If an actor does not cover an event area, that is it can not cover the sensor node reporting event from that area, it should coordinate with the other actors so that the uncovered field is targeted by some actor.

As actors move within a geographical delta they report their location, *event type*, *interest duration* ( $\gamma$ ), event response time  $\epsilon$  and action response capabilities using a long-range in-network *subscription* message. Actors explicitly mention the type of event they are interested in and also the *interest duration* after which an actor is not interested in the readings. A small value of  $\gamma$  refers to a pull-based dissemination because the frequency of pull messages is very high, which is further increased when the actors are moving frequently. Due to actors movement, coverage or targeting area also changes and there needs an updated interest *subscription* message. On the other hand, the frequency of the interest message decreases when an actor moves infrequently and  $\gamma$  is large. When sensor nodes detect some event, they push the information to the interested actor for a time period of  $\gamma$ . Consequently, the event readings are not necessarily pulled by the actors every time they occur but are pushed by the sensor nodes once they know about the actors capable of taking action. Hence, an integrated Pull/Push coordination mechanism is used to efficiently operate actors in WSANs.

### **5.3.1 Sensor-Actor (S-A) Coordination**

In WSANs, multiple destinations (actors) of sensor readings are possible as opposed to single destination (sink) in WSNs. In other words, all sensor nodes may not transmit data to the single actor. Hence, the problems in S-A coordination are; which sensors com-



municate with which actors and how to establish a communication path between sensors and actors [1]. Additionally, the given response time must be considered to achieve the right action. We propose a S-A coordination scheme by employing an integrated Pull/Push (IPP) approach, which forms *responsibility clusters* in the field.

### **Pull/Push in S-A Coordination**

Applications in WSN involve dissemination of observed event readings to the interested clients. There are two ways to disseminate the information. The first way is to reactively send the queries in the field and pull the relevant information out of the field. The second way is to proactively push all the relevant information out of the network regardless of the current queries. Both approaches have some benefits as well as drawbacks.

Sensor networks are proposed to play role in detecting and characterizing Chemical, Biological, Radiological, Nuclear, and Explosive (CBRNE) material to prevent disastrous conditions. Obviously, an immediate action is imperative to encounter the threats in such applications. In such circumstances, pure-pull policy is not applicable because the applications need to pay immediate attention to certain events, which is not possible unless the queries are sent frequently. However, frequent queries introduce significant overheads in terms of resource utilization and delay. On the other hand, pure-push policy is cost inefficient when the events are received from the regions where actions can not be executed due to the restricted movement of actors or lack of resources to execute action or, even more specifically, the events are out of the region of interest.

We deal with the above mentioned problems by using integrated Pull/Push Coordination to consider the disadvantages of sole pull or push policy. Hybrid Push/Pull coordination is studied in [64, 70, 71]. These studies are proposed for monitoring and control purposes in WSN that has a single sink. However, in WSN, an efficient coordination mechanism is required to overcome the mobility of multiple destination actors. Unlike these studies, we exploit the mobility of actors. Actor nodes show their interest about a

particular event by broadcasting a *subscribe* message request. Keep in mind that the actors use long range transmission so that it is received by all the sensor nodes which do not involve in propagating such message. The request is announced to let nodes decide suitable actor that can react when the event is reported. Sensor nodes may receive the request from a subset of actor nodes. However, we assume that the event is reported to only a single actor. Furthermore, we assume that the subscription is made on the basis of geographical locations. This criterion can be extended to include the load factor and energy of the actor to decide the subscription in an overlapping region. There is an associated life  $\gamma$  of the actor subscription and it expires if no further *subscribe* message is received during that time or the subscribed actor has moved.

An actor broadcasts a *subscribe* message at two occasions; first, when it changes its location by some threshold ( $M_{th}$ ) value or leaves the event area it is currently targeting. This is done to handle the actor mobility and keep the coordination effective. Second, the interest is refreshed after each subscription period. Hence, the control message cost is directly related to the mobility of actors and integrated pull mechanism. Short subscription life leads to more frequent queries and intuitively more overheads. The mobility can be restrained by deploying sufficient number of actors to cover the whole region. The communication path between sensor and target actor is established by DC-GEO approach described in Section 5.2.

### **Subscribe (Pull) Message Frequency ( $f$ )**

The frequency of *subscribe* message ( $f$ ) is computed by  $1/\gamma$  initially which is the minimum bound on  $f$ . However, the mobility of actors dynamically alters this frequency. In such case, we define maximum bound on  $f$  in order to avoid the frequent queries sent as in pure pull-based approach. This is achieved by setting the value of  $M_{th}$  according to the movement velocity of actors. For such changes, we restrict  $f$  not to go beyond 1 message per unit time. As a result, the value of  $M_{th}$  is set according to the movement

speed and the density of sensor nodes. It is calculated such that the coordination remains effective. By effectiveness we mean that whenever an event occurs, it should be propagated to some actor. If the movement speed is high, then the large value of  $f$  (resulted from low threshold value relative to the speed) would result in wrong subscription of actors and loss of event reports. Clearly, it seems that the threshold value should be relative to the speed of actors, somehow. That is, if the speed is high,  $M_{th}$  should be relatively large to reduce  $f$ . At lower velocity, the value of  $M_{th}$  will also be lower by relating threshold with velocity. Intuitively, there would be unnecessary *subscribe* messages that limit the benefits of integrated Pull/Push coordination. Hence, we define a relation between  $M_{th}$  and the movement velocity to obtain the suitable value of threshold.

Let  $s$ ,  $n$  and  $M$ , respectively, be the mobility speed of an actor, total number of nodes and dimension of the field. Then the unit  $\theta$  occupied per node (node occupancy) can be computed by  $\sqrt{M^2/n}$ . By using the node occupancy value  $\theta$ , we can find the value of  $M_{th}$  as:

$$M_{th} = \begin{cases} s & \text{if } s > \theta \\ \theta & \text{if } s \leq \theta \end{cases}$$

Hence, the value of  $f$  is computed dynamically to minimize the overheads experienced due to subscription requests.

$$f = \begin{cases} \frac{1}{\gamma} & \text{if } s \approx 0 \\ \frac{s}{M_{th}} & \text{if } s > 0 \end{cases}$$

### 5.3.2 Actor-Actor (A-A) Coordination

Actors are assumed to attend an emergency anywhere in the field. An actor is moved toward emergency zone if the source node is not within its action range. This is a controlled targeting i.e. actor moves toward the requesting source such that the source node is within the boundary of its action. In S-A coordination, sensor nodes send their readings

to the nearest actor. However, an actor receiving sensor data may not act on the event area due to small action range or low energy. This requires exploiting the coordination between actor nodes to track the target area. Each actor node maintains a neighborhood list of nearby actors in order to trigger coordination. A *subscribe* message also contains information about the number of requests (sensors) currently the actor is serving. Thus, every actor maintains a list of actors with the location information as well as the load. We assume that there are sufficient number of actors to cover the whole sensor field. Sensor nodes send event readings with their location information. As an actor receives a request from a sensor node that lies outside its coverage area, it starts the event targeting procedure as described in Section 4.5.1.

When the actors have either limited action range or they are small in number to cover the whole field, it is quite possible that some action requests remain unsatisfied. This situation arises when the intensity of events is too high to affect the entire field. We can compute the number of unattended sources  $\mu$  in ideal case for  $M \times M$  field as

$$\mu = \frac{M^2 - m \times \pi R^2}{M^2} \times n \quad (5.1)$$

where  $m$  is the number of mobile actors,  $n$  is the total number of deployed sensors. However, the value of  $\mu$  depends on the density of nodes in the field. In case of non-uniform deployment when the action range is limited, actors might be stuck in sparse zone of the field that can not attend the nodes in dense part as described above, which greatly effects the optimal value of  $\mu$ .

## 5.4 Performance Evaluation

To evaluate the performance of RAT, the experiment model consists of 150 sensor nodes, having 20 meters of transmission radius each, deployed randomly in  $100 \times 100 m^2$  area. Additionally, 5 actors are also placed in the field at appropriate locations. Experi-

ments are run with different event radius  $\mathfrak{R}$  and action range  $R$ . Table 5.1 illustrates the number of nodes detecting events at different event radii. In the example runs, sources generate 3 packets per second, each 100 bytes long, for 100 seconds.

Table 5.1: Number of Nodes vs Event Radius

Event Radius $\mathfrak{R}$	20	22.5	25	30	32.5	35	40	50
Nodes $\bar{n}$ ( $\approx$ )	19	24	30	43	50	56	76	118

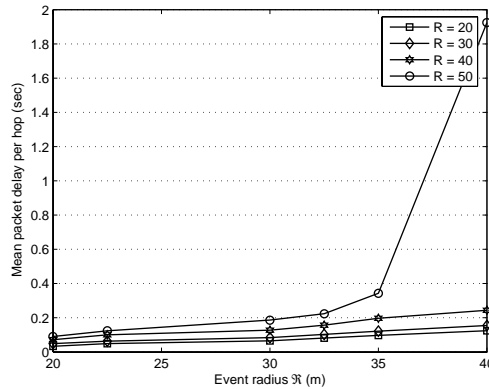


Figure 5.4: Average response time in RAT.

Fig. 5.4 illustrates the average response time, in terms of delay per hop, as a function of event radius. It is obvious from the graph that the delay per hop increases as we increase the number of sources by expanding the event radius. By increasing the event radius, the number of sensors detecting the events also increase producing more traffic. Whereas, a large value of action range covers wide area and results in attending requests from farther nodes, increasing the average number of hops. Intuitively, the impact of large event radius and action range causes more traffic effecting the performance negatively, in terms of delivery ratio, average hops and packet delay. Obviously, the requests per actor are increased as well and the average number of hops is approximately 2.2 per request at  $R = 50$  as shown in Fig. 5.5 as compared to 1.4 at  $R = 30$ . The increase in number of hops introduces more delay and poor response time. We can conclude from the results that the average hops are approximately  $R/r$ .

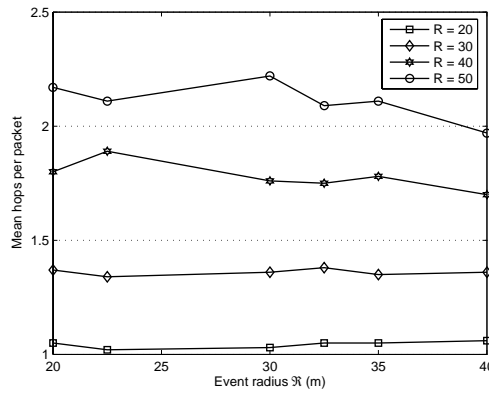


Figure 5.5: Average number of hops for different action range of actors in RAT.

By employing movement model in A-A coordination, we can improve the response time and the packet delivery ratio as shown in Fig. 5.6. By keeping the action range ( $R = 20$ ) too small results in lower delay. However, it reduces the delivery ratio. A large fraction of requests remain unsatisfied as shown in Fig. 5.7. Hence, for  $R/r \approx 1.5$  (where  $R = 30$  and  $r = 20$ ), the response time is improved as well as the overall packet delivery ratio.

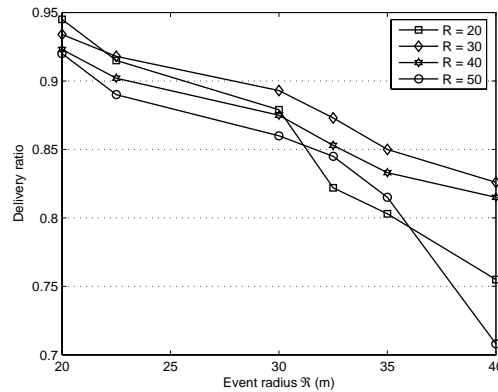


Figure 5.6: Delivery ratio in RAT.

The theoretical value of unattended requests computed by using Eq 5.1 for  $R = 20$  with 5 actors is also plotted in Fig. 5.7. It can be seen that when event radius goes beyond 30, the actual unsatisfied request ratio found by simulation becomes worst. This is due to the random distribution of nodes that leaves some area sparse and some dense. The effect

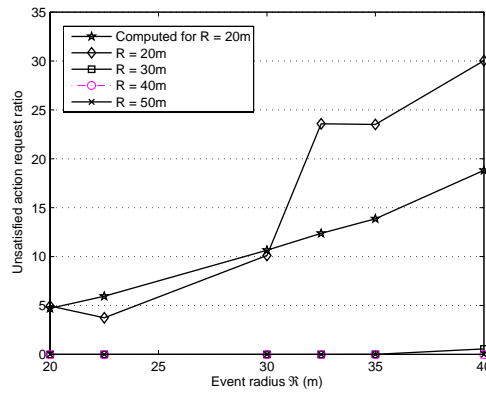


Figure 5.7: Unattended requests in RAT.

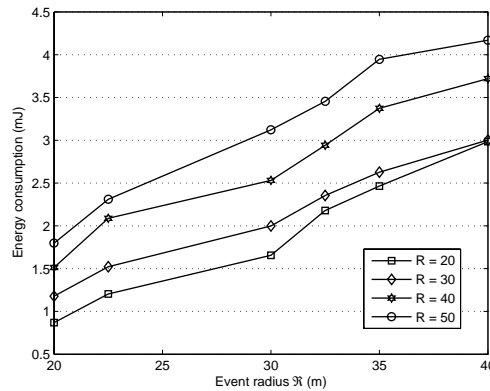


Figure 5.8: Average energy consumption in RAT.

seems more significant as we increase the event radius with small action range. However, for large action range  $R \geq 30$ , the field is mostly covered by the actors, leaving unattended requests at minimum.

Fig. 5.8 refers to overall energy consumption with different values of  $R$ . Although the energy consumption for  $R = 20$  is very low, this value offers lower delivery rate that might not be acceptable. On the other hand, the energy consumption is slightly higher at  $R = 30$  with higher delivery rate and better response time. Hence, the results suggest that response time and delivery ratio can be improved with efficient energy consumption when the route length is approximately 1.4.

Although we have reported that the performance is better at  $R/r \approx 1.5$ , it is just ac-

According to the given scenario and applications can choose  $R$  and  $r$  according to their own delay constraint that they can tolerate or other performance parameters. This result can be used as a base to design their network configuration.

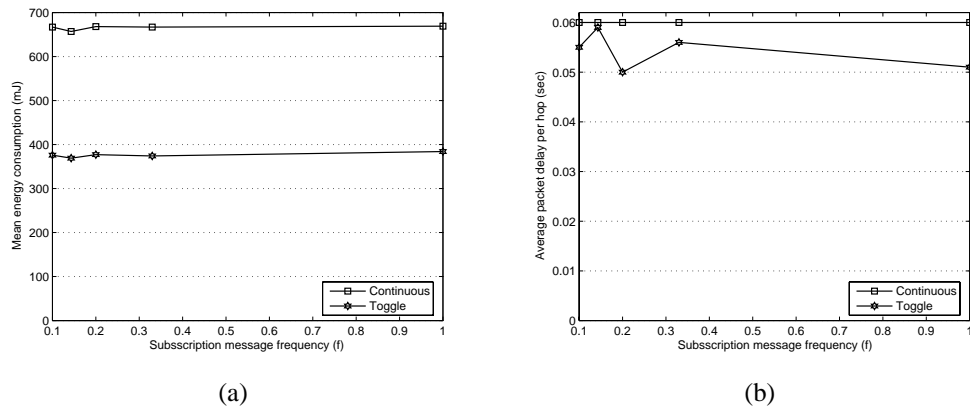


Figure 5.9: Energy and delay comparison at different subscription period.

In the tests, due to long range transmission of actor nodes, all the nodes receive the *subscribe message* directly. As a result, the large value of  $f$  does not make any significant difference in energy consumption as shown in Fig. 5.9(a). The energy consumption is almost the same for all the values of  $f$ . It is due to the fact that the traffic is huge as compared to the *subscribe message* frequency which does not seemingly change the energy consumption by increasing  $f$ . However, the energy is saved by pushing the event observations only when there is an interested actor. The results plotted in Fig. 5.9(a) illustrate the energy consumption of two scenarios. In the first scenario, nodes continuously push data to the subscribed actors. In the second case, actors are interested in event readings only for the time period of  $\gamma$  and remain silent for an equal time period i.e., they toggle their interest. It is apparent that the energy can be saved in *toggle* mode for approximately 60%. Eventually, pushing data only when it is required by the interested clients saves energy significantly. Fig. 5.9(b) shows the average packet delay in aforementioned scenarios. It is obvious that the delay is not reduced much in *toggle* scenario because the data rate remains same whenever it is reported. However, if  $\gamma$  is small i.e.  $f$  is large, then event flows for shorter time that results in small delay.



## 5.5 Summary

We have presented a unique integrated push/pull coordination mechanism in wireless sensor and actor networks. The information dissemination is triggered by a *subscribe* messages from actors showing their interest and range of action. A pure pull-based approach is not applicable, especially when the frequency of *subscribe* message is small as compared to the events reported by the sensor nodes. In this study, once the subscription of an actor is performed by the sensor nodes, we adapt to push policy and continue to push the event data as long as the subscription is valid. The subscription is refreshed whenever it expires or actor moves.

Moreover, it is not possible to predetermine the exact location of a limited number of actors to provide full coverage in the field. We exploit the mobility of actors in order to target the uncovered event area. Simulation results reveal that with the proper choice of actor action range and A-A coordination, we achieve higher delivery rate, better response time and better energy consumption. The results prove that IPP shows better performance under varying traffic load.

## CHAPTER 6

### EXPLOITING ENERGY-EFFICIENT SPATIAL CORRELATION

#### 6.1 Introduction

Wireless sensor networks promise fine-grain monitoring in a wide variety of applications. The environments in some of these applications e.g. indoor environments, battle-field or habitats, can be harsh for wireless communication. Due to hostile environments and limited energy or transmission range, sensor nodes are highly prone to failures. Sensor nodes may fail or be blocked due to lack of power, have physical damage or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network. This is the reliability or fault tolerance issue. Fault tolerance is the ability to sustain sensor network functionalities without any interruption due to sensor node failures [72].

In order to increase the reliability, hundreds to several thousands of nodes are deployed within tens of feet of each other throughout the sensor field. The node densities may be as high as 20 nodes/m<sup>3</sup> [73]. As a result of high node density, spatially redundant or correlated information is available in the network. Redundancy increases the reliability level of information delivery while increasing energy consumption of nodes as well. Since energy conservation is a key issue for WSNs, therefore, spatial correlation can be exploited to deactivate some of the nodes generating redundant information due to high node density. This decreases the number of transmissions and increase the life time of the network. Ag-

gregation has also been suggested to reduce the volume of traffic and eventually an energy conserving approach. However, it provides estimated results that may not be appropriate for applications demanding intolerable distortion in event readings. Furthermore, all the nodes in network remain active unnecessarily and participate in observing the phenomenon that may be otherwise avoided by exploiting spatial correlation.

Spatial correlation has been explored in the literature to some extent [75]-[87]. However, none of these studies focus on selecting appropriate representative nodes <sup>\*</sup> in terms of their residual energy. In this paper, we present an energy-aware spatial correlation mechanism based on the cluster-based configuration in WSNs. The underlying clustering protocol MEAC [63] itself is an energy efficient and we further conserve the energy of nodes by exploiting spatial correlation. In this approach, only the cluster-heads are responsible of applying spatial correlation to their member nodes and selecting the appropriate member nodes to remain active for observing the phenomenon. The correlation is based on the distortion tolerance and the energy of member nodes. Applications define the correlation region <sup>†</sup> in which only one reading is sufficient for the event reading precision. However, the correlation factor can be changed dynamically in order to achieve the distortion tolerance. Each cluster-head divides its clustered region into correlation regions and selects a representative node in each correlation region which is closer to the center of correlation region and has the higher residual energy. The non-representative nodes remain passive until the energy of active nodes go down to some threshold value. Hence, the whole field is efficiently represented by a subset of active nodes which perform the task well equal.

## 6.2 Related Work

In this section, we investigate some of the existing approaches exploiting spatial correlation in WSNs. The information theoretical aspects of the correlation are explored in

---

<sup>\*</sup>A node which reports event on behalf of a group of nodes detecting similar readings.

<sup>†</sup>The region in which the sensor nodes report almost the similar readings is called correlation region.

depth in [79, 80]. In other words, these studies aim to find the optimum rate to compress redundant information in the sensor observations. However, no correlation (spatial or temporal) between sensor observations is considered in these studies. The work in [77] exploits the spatial correlation to measure the link quality of wireless sensor nodes. The intuition behind spatial correlation is that sensor nodes geographically close to each other may have correlated link quality. It shows that the spatial correlation in link quality of neighbor sensor nodes can be captured to estimate the link quality with substantially less transmission cost than the link quality estimators based on temporal correlation. The history information of link quality for one node may be used for estimating not only its own link quality but also that of other neighbor sensor nodes geographically close. These approaches do not exploit spatial correlation in communication network protocols to eliminate the redundant event reports in order to conserve the energy of nodes.

In [2], the relation between spatial locations of the sensor nodes in the event area and the event estimation reliability has been formulated. It estimates the minimum number of representative nodes in the field and selects a representative node among a group of spatially correlated sensor nodes. The correlation region is determined by applying Vector Quantization (VQ) statistical approach. However in this approach, the energy has not been considered in the selection of a representative node criterion. It achieves the overall energy gain in the network regardless of the energy of individual node, which is not a load-balancing approach. Moreover, if a representative node, which is the center of correlation region, moves or its energy depletes, the selection of another representative node for that region effects the tolerable distortion. Frequent change in representative nodes produces more distortion and hence, the whole network is required to reconfigure to achieve the distortion limit. On the other hand, nodes working as representatives in their correlation regions for longer period results in unfair load distribution that reduces the network life. CC-MAC [78] uses the above formulation of spatial correlation to work with the medium access control (MAC).

Spatial correlation is also considered as a quality of service (QoS) parameter in some studies [85, 86, 87]. They use the idea of allowing the base station to communicate QoS information to each of the sensors using a broadcast channel and use the mathematical paradigm of the Gur Game to dynamically adjust to the optimum number of sensors. The sink node dynamically adjusts the number of sensors being activated to achieve the required reliability, thereby controlling the resolution of QoS it receives from the sensors. However, this approach controls just the number of active nodes ignoring their location to select appropriate nodes covering non-overlapping correlation regions. In short, all of the above approaches do not consider the energy of nodes in selecting the representative nodes and therefore do not efficiently exploit the spatial correlation.

## **6.3 Cluster-based Spatial Correlation of Sensor Nodes**

In this section, we present a grid-based spatial correlation of sensor nodes that is deterministic and more independent of network dynamics. Generally, the nodes coverage control mechanism is required when the number of sensor nodes deployed in the field are more than the sufficient number of nodes to provide the coverage in the whole field. Intuitively, the high density of nodes produce more traffic and results in congestion. We control the spatial resolution of nodes by deactivating the redundant nodes. By redundant nodes, we mean the nodes which are so close to each other that identical event information is reported. However, we divert the drawback of redundancy into the network favor. An active node is deactivated when its energy goes down below the energy of its nearby inactive node by a certain threshold value and, hence, inactive node is activated.

### **6.3.1 Correlation Model**

We assume that the sensor nodes report almost similar values when they are close to each other. However, this closeness ( $\theta$ ) depends on the application requirements and event

characteristics. Some applications are more critical and are less tolerant to discrepancies in the event readings requiring closer nodes to report event readings. While the others can be more tolerant requiring the farther nodes to report events. *We define the region as correlation region in which the readings reported by the sensor nodes are considered similar by the applications and therefore a single report is sufficient to represent that region.* It varies from application to application and event to event. For intolerable applications, the region is very small and large for tolerable. Hence the region is directly related to the application defined distortion tolerance. Similarly for certain events, the event features vary significantly as the event signal propagates in the field. Such an event requires to be reported by relatively nearby nodes as opposed to the events whose detected features do not change at short distance.

The distortion in readings is observed when the sensor nodes fail to report the event from within the defined correlation region. The correlation region can then be changed dynamically according to the observed reliability. For example, an application may require to get readings at locations 10 meters apart. In that case the dimensions of correlation region is considered  $10 \times 10 \text{ m}^2$ . If a sink node receives readings from the nodes which are more than 10 meters far from each other then the distortion is observed and readings are considered unreliable relative to their closeness. *Hence, the distortion is observed if the reporting nodes are apart more than  $\theta$  and therefore it is measured in terms of closeness rather than the event features. It is up to the application to evaluate the event features and set the value of  $\theta$  accordingly.* Hence,  $\theta$  controls the number of active nodes ( $K$ ).

In order to investigate the distortion achieved when smaller number of nodes sending information, we assume that only  $K$  out of  $n$  packets are received by the sink, where  $n$  is the total number of sensor nodes in the event area. The distortion function  $D(K)$  to find out the relation between the spatial locations of the sensor nodes in the event area and the event estimation precision has been formulated in [78] as

$$\begin{aligned}
D(K) &= \sigma_S^2 - \frac{\sigma_S^4}{K(\sigma_S^2 + \sigma_N^2)} (2 \sum_{i=1}^K K \rho(s, i) - 1) \\
&+ \frac{\sigma_S^6}{K^2(\sigma_S^2 + \sigma_N^2)^2} \sum_{i=1}^K \sum_{j \neq i}^K \rho(i, j)
\end{aligned}$$

where  $\sigma_S^2$  and  $\sigma_N^2$  are the variances of the  $i^{\text{th}}$  event reading  $S_i$  and the observation noise  $N_i$ , respectively.  $D(K)$  shows the event distortion achieved at the sink as a function of the number of sensor nodes  $K$  that send information to the sink and correlation coefficients  $\rho(i, j)$  and  $\rho(s, i)$  between nodes  $n_i$  and  $n_j$ , and the event  $S$  from the sensor field and the sensor node  $n_i$ , respectively. The correlation coefficient between nodes  $n_i$  and  $n_j$  can be computed by  $\frac{E[S_i S_j]}{\sigma_S^2}$ .

### 6.3.2 Gridiron Spatial Correlation (GSC)

The gridiron spatial correlation mechanism is adaptive to achieve the required reliability by dynamically changing the correlation region. The correlation regions are formed as squared rectangles and nodes lying in the rectangle are assumed to be spatially correlated. However, the correlation is fine tuned by resizing the rectangle according to the feedback provided by the sink computed on the basis of redundancy and reliability. The rectangles are considered independent of each other and, hence, switching the representation among nodes in a rectangle does not require the reconfiguration of the entire network. The switching is performed to balance the energy usage of correlated nodes in a heterogeneous network. Hence, the network is highly adaptive to the events of different intensity and heterogeneity.

Cluster-head identifies the redundant and close sources in its vicinity and turns off the activity of nodes by considering their energy level and closeness as criterion, where the closeness  $\theta$  is either defined by the application or evaluated according to the density of

nodes in the zone.

$$\theta = \sqrt{R^2/n} \quad (6.1)$$

where,  $R$  is the dimension of the squared field and  $n$  represents the total number of nodes deployed in  $R \times R$  field.

Furthermore, the number of active sources can also be adjusted according to the required reliability level announced by some central node. However, this is applicable in continuous data thirst applications that would like to monitor the environment continuously. By controlling the spatial resolution, we implicitly avoid the congestion in network to happen.

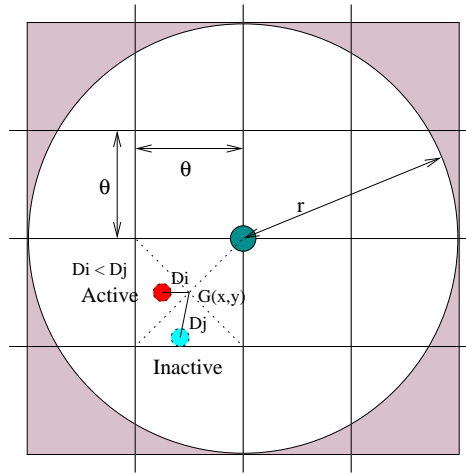


Figure 6.1: Spatial resolution control mechanism employed by the cluster-head.

Fig 6.1 illustrates the mechanism to control the spatial resolution. Cluster zone is decomposed into grid of  $2r/\theta \times 2r/\theta$  rectangles, where the cluster-head lies at the center of grid. The aim of our spatial resolution control mechanism is to select a single node in each rectangle of  $\theta \times \theta$  dimensions. Let  $G(x_i, y_i)$  be the center of an  $i^{th}$  rectangle. We measure the distance of each node lying in the  $i^{th}$  rectangle from the point  $G(x_i, y_i)$ . The closest node is selected as an active member of the cluster and other nodes in that rectangle are deactivated by the cluster-head. Similarly, the inactive nodes do not participate in relaying



packets i.e. their transceivers are also turned off during the current cycle. Fig 6.2 shows the active members in a cluster selected by applying our resolution control procedure. It is important to note that the multi-hop members do not lie in the direct coverage of cluster-head and, therefore, control mechanism is not applied to multi-hop members.

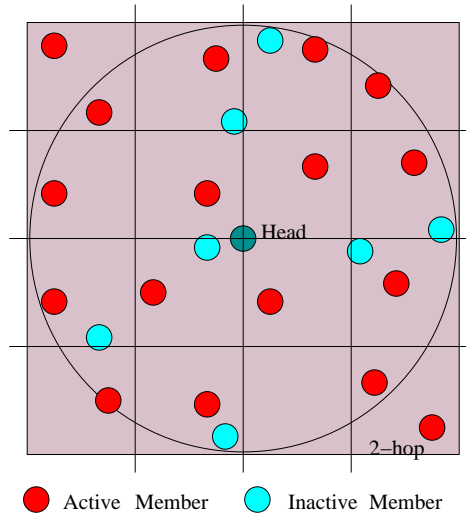


Figure 6.2: Spatial resolution in cluster zone.

Cluster formation procedure in [63] is followed by exploiting gridiron spatial correlation. Cluster-head applies the gridiron procedure given in Algorithm 1 over its member nodes and decides the active nodes. It then announces the list of active nodes in a single *AList* message. The size of *AList* message is dynamic relative to the number of active nodes. Cluster-heads run this procedure periodically and whenever there is a change in active nodes list, they broadcast the updated list. This change might be either due to changed  $\theta$  or the energy of active nodes drop to threshold.

### 6.3.3 Scheduling Nodes

GSC does not apply duty cycle to alternate the activity of nodes. Cluster-heads rather use threshold to decide about the activeness of nodes. However, it follows the duty cycle so that all the member nodes (active or inactive) hear the transmission of their cluster-heads at some time. For instance, the cluster-head announces the initial list (*AList message*) of

---

**Algorithm 1** Gridiron Spatial Correlation

---

```
1: Cluster-head runs the procedure to decide the active nodes.
2:  $nRects = r/\theta$ 
3: for  $x = -nRects; x + +; x < nRects$  do
4:   for  $y = -nRects; y + +; y < nRects$  do
5:      $rect_x = head_x + x \times \theta$ 
6:      $rect_y = head_y + y \times \theta$ 
7:      $C_x = rect_x + \theta/2$ 
8:      $C_y = rect_y + \theta/2$ 
9:      $near_{dist} = \infty$ 
10:     $active_{member} = null$ 
11:    for all  $m \in members$  do
12:       $d = \sqrt{((C_x - m.nbr_x)^2 + (C_y - m.nbr_y)^2)}$ ;
13:      if  $d < near_{dist}$  &&  $m.energy + threshold > MaxMembersEnergy()$  then
14:         $near_{dist} = d$ 
15:         $active_{member} = m$ ;
16:      end if
17:    end for
18:    if  $active_{member} \neq null$  then
19:       $active_{member}.is\_active = true$ 
20:    end if
21:  end for
22: end for
```

---

active nodes at time  $t = t_1$ . All the member nodes obey the decision of cluster-head for time period of  $\Delta$  seconds. During this time i.e.  $t_1 \rightarrow t_1 + \Delta$ , inactive remains inactive and do not overhear any transmission. However, as this time period passes, all the nodes become active for a shorter time period of  $\epsilon$  seconds. Cluster-heads, during this time period, broadcast an updated list of active nodes. Henceforth, the member nodes obey the new decision and unnamed node in this list go into sleep mode for  $\Delta$  seconds. Fig. 6.3 illustrates the state transition of the member nodes.

The list of active nodes is modified if the energy of current active nodes goes down to some threshold value as compared their correlated nodes. Hence, GSC maintains equilibrium of the energy of correlated nodes in order to extend the life of network. Cluster-head also keep information about their neighboring cluster-heads as described in [63] in order to establish route with the sink node. In order to exploit GSC, it makes sure that deacti-

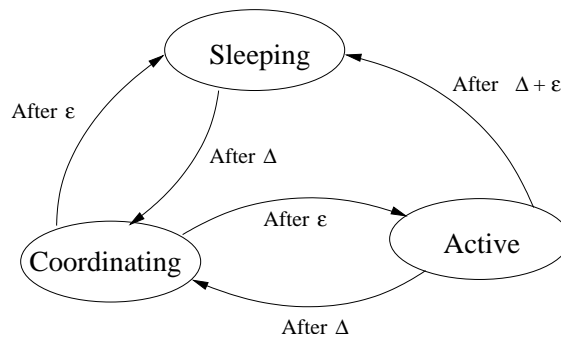


Figure 6.3: State transition of member nodes in exploiting GSC.

vating the member nodes does not break its connection to the cluster-head through which it forwards the packet. If it suffers in such a problem then it does not deactivate the node working as a gateway and let it to remain active. Hence, a steiner tree is maintained in order to ensure the minimum connectivity of the nodes to the sink.

## 6.4 Performance Evaluation

The performance of the spatial correlation mechanism is evaluated by using the network simulator *ns-2* [106]. The example scenario consists of various number of nodes randomly deployed in the field of  $100 \times 100 m^2$ . An event source is also included in the scenario to trigger events by using the NRL phenomenon node extensions [107] for *ns-2*. The experiments are run at different density of nodes ( $\mu = n\pi r^2/M$ ), where the density is changed by varying the number of nodes ( $n$ ) while keeping the area of the field ( $M$ ) and transmission radius of nodes ( $r$ ) constant. The performance metric consists of the event readings distortion, number of active nodes to achieve the required reliability, average packet delay and, of course, energy consumption.

By exploiting spatial correlation, we control the number of active nodes. For larger value of  $\theta$ , lesser number of nodes are active and eventually lower reporting rate is observed. Fig. 6.4 shows the reporting rate of active nodes at different number of deployed nodes. Applications can define the correlation region by setting the value of  $\theta$  accord-

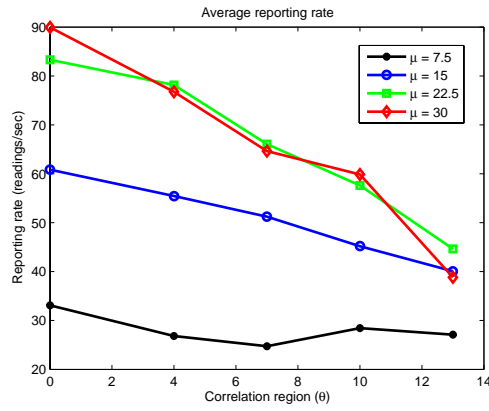


Figure 6.4: Event reporting rate of active nodes.

ing to their required reporting rate for different number of deployed nodes. Although the nodes are densely deployed but GSC keeps the number of nodes as active as required and eventually the reporting report is almost same at higher densities ( $\mu = 22.5$  and  $\mu = 30$ ). However, this is applicable only when the number of deployed nodes are higher than the sufficient number of nodes to cover the field. At  $\theta = 10$  meters, 40% nodes remain passive for  $n = 200$ , which is 20% for  $n = 100$  and just 5% for  $n = 50$ . Ideally, there should not be any inactive nodes at lower density. But it is due to the random deployment in which some nodes may be deployed very close to each other and are made inactive as given for  $n = 50$ . Hence, the results in Fig. 6.4 reveal that the number of nodes are activated relative to the ratio of their deployment or density. Similarly, the number of active nodes are directly related to the value of  $\theta$  at higher density ( $n = 200$ ), which shows a linear relation.

Fig. 6.6 shows the distortion at different correlation regions. It is clear that the distortion is lesser at higher density and larger  $\theta$ . The distortion in readings occurs not only due to low density but also due to the non-uniform deployment. Generally the distortion at lower density should be reduced by the factor of additional deployment of nodes. However this reduction is until certain limit and do not improve further because the nodes are randomly deployed and are placed non-uniformly in some part of the field. At lower density ( $\mu = 7.5$ ), the reduction is linear and is reduced 100% by expanding the correlation region while an increase in  $\theta$  at higher density ( $\mu = 30$ ) reduce distortion upto 60%.

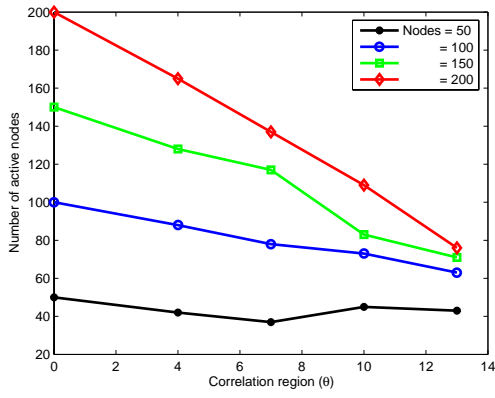


Figure 6.5: Active nodes for different values of correlation region  $\theta$ .

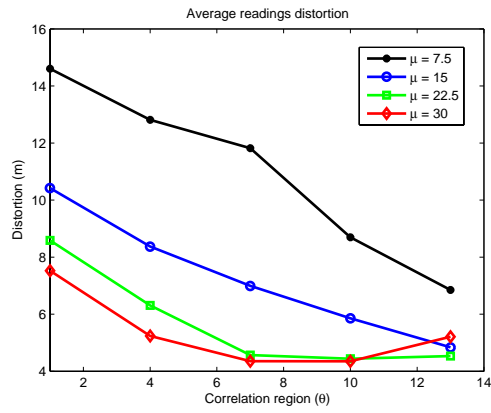


Figure 6.6: Distortion in sensor readings at different values of correlation region  $\theta$

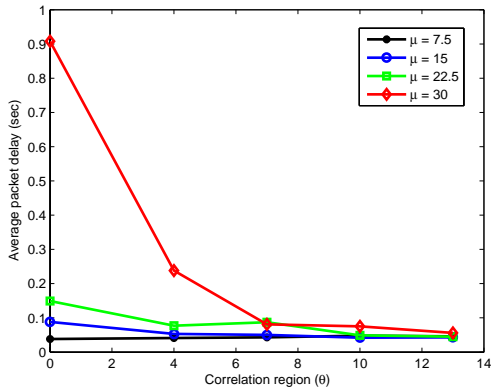


Figure 6.7: Average packet delay observed at different values of correlation region.

When the density of nodes is high, the wireless channel is contended by large number of nodes that result in large communication delay. Fig. 6.7 shows that the average packet

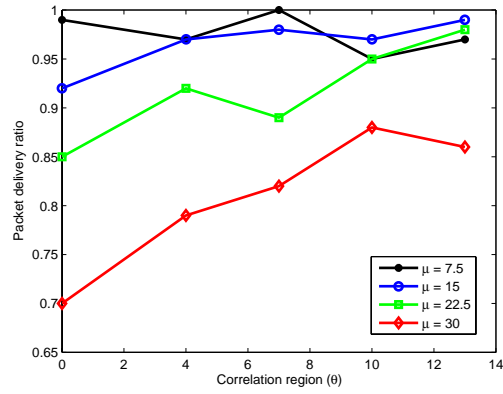


Figure 6.8: Packet delivery ratio of active nodes at different values of correlation region.

delay at  $\mu = 30$  and  $\theta = 0$  is 9 times larger than at lower densities and  $\theta > 0$ . By exploiting spatial correlation at higher density ( $\mu = 30$ ), we reduce the delay significantly by increasing the value of  $\theta$  and hence it approaches to the same value as for lower densities i.e. it is reduced by 9 times. Similarly the delivery ratio is improved by increasing the value of  $\theta$  as shown in Fig. 6.8.

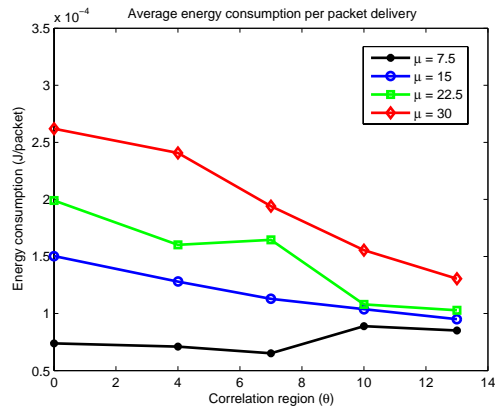


Figure 6.9: Energy consumed per packet delivery at different values of correlation region.

Likewise, the energy consumed per packet delivery is also conserved at higher density by exploiting spatial correlation which is not true for lower density. It is obvious from Fig. 6.9 that for  $\mu = 30$  the energy is saved approximately two times at  $\theta = 13$ . However, this improvement can not be experienced at lower density and eventually there is no energy conservation at  $\mu = 7.5$ .

## 6.5 Summary

Spatial correlation has been exploited in the literature to conserve the energy of sensor nodes. The existing approaches do not consider the residual energy of nodes in selecting the representative nodes and therefore do not efficiently exploit the spatial correlation. In this paper, we present an energy-aware spatial correlation based on the clustering protocol. Cluster-heads apply spatial correlation in their regions independently to keep the member nodes active subject to the information reliability. Each cluster-head divides its clustered region into correlation regions and selects a representative nodes per correlation region which is closer to the center of correlation region and has the higher residual energy. However the correlation regions can be resized according to distortion tolerance which is measured and announced by the sink node. Hence, the whole field is efficiently represented by a subset of active nodes which perform the task well equal to the all deployed nodes. Simulation results prove that the required reporting rate can be achieved with lesser number of nodes by exploiting spatial correlation and eventually conserves the nodes energy.

## CHAPTER 7

### CONCLUSION AND FUTURE WORK

#### 7.1 Conclusion

Wireless sensor actor networks (WSANs) consists of a group of distributed sensors and actors that communicate through wireless links. Sensors are small and static devices with computation and communication capabilities responsible for observing the physical world, at limited power. On the other hand, actors are lesser in number but are equipped with richer resources and are able to move around to perform appropriate actions. Such systems require efficient communication and coordinations to be attractive for certain applications. The energy conservation always remain as the primary concern during all the operations of such a network. In this study, a novel energy-efficient multi-level communication and coordination framework for wireless sensor and actor networks is presented. The framework incorporates an accurate sensors localization algorithm TSL, two real-time coordination and routing components (RCR and RAT) and an energy conserving mechanism exploiting the spatial correlation of sensor nodes.

A novel timing-based sensor localization (TSL) algorithm is presented in Chapter 2, which localizes the sensor nodes with reference to the actor nodes. TSL is adaptive to different velocity of mobile sensor/actor nodes. The least energy consumption algorithm is aimed at accuracy within tolerable limits. Simulation results prove that TSL achieves high accuracy and restricts the localization errors to less than 1 meter by tuning it according to



the expected velocity of nodes.

Multi-event adaptive clustering (MEAC) protocol described in Chapter 3 provides foundation for the distributed real-time coordination and routing (RCR) protocol. It makes use of heterogeneity of sensor energy and data rates in such a way that energy consumption is reduced with the presence of multiple events. This results in balancing the load on the sensor nodes and extending the life of network as compared to SEP [36] and LEACH [34] with about 33 % energy gain.

RCR, described in Chapter 4, addresses the issues of real-time data aggregation, packet delivery, sensor-actor and actor-actor coordinations. It uses MEAC protocol for sensor network clustering. Cluster-heads performs real-time data aggregation (RDA) reducing the volume of traffic while taking care of the packets deadline. The novel RDA employs fairness among the nodes lying at different distance from the destination actors. The nearer nodes have lesser communication delay and more time for the aggregation leaving the bandwidth for the farther nodes. The response time is improved with actors moving toward the reporting nodes if they do not come in their action range. Hence, the response time in RCR is achieved by applying real-time aggregation, routing and coordination.

The adaptive targeting (RAT) protocol discussed in Chapter 5 is an alternative routing protocol, which performs better than RCR when the nodes are densely deployed. It incorporates; delay-constrained geographical-based routing (DC-GEO) and integrated Pull/Push (IPP) coordination. It is proved that DC-GEO relays the packets more speedy than a classical greedy mode routing. IPP is a unique integrated and distributed push/pull coordination mechanism, which improves data delivery, packet delay and energy consumption.

Spatial correlation is exploited to disable the activity of some of the nodes generating redundant information as described in Chapter 6. Some solutions have been presented in the literature but they do not distinguish among the nodes having different energy levels i.e. they are not heterogeneity aware. Our approach is more adaptive to heterogeneity and

keeps the energy consumption of the nodes balanced. It selects nodes that have higher energy levels and keeps checking their remaining energy. Consequently, the correlation is based on the distortion tolerance and the energy of member nodes. The whole field is efficiently represented by a subset of active nodes as shown by the simulation results.

## 7.2 Future Work

The research work mainly focuses on the real-time applications in WSANs and provides solutions for localization, configuration, routing and coordination in the field. The energy is considered as the primary concern in the design of these approaches. Other important factors in WSANs that needs further investigation are mobility of sensor nodes, co-existence of real-time traffic with best effort traffic, TCP/IP interface, and distributed security.

The mobility of sensor nodes is an important aspect in the design of applications, which severely degrades the performance of routing protocols. Therefore, research might be conducted to identify the impact of different models of mobility of sensor nodes on the routing protocols proposed in this study.

Multi-path routing protocols have been explored to provide improved throughput and route resilience as compared with single-path routing in wireless sensor networks. However, it is unnecessary to always transmit duplicate packets through scarce resource devices to achieve the required reliability. The coexistence of real-time traffic and best effort traffic requires a QoS-based routing protocol that supports the real-time traffic as well as provides certain reliability assurance to best-effort traffic. RCR and RAT can be extended in these directions to support the QoS parameters other than delay and energy. Also as future work, we aim to provide analytical support in GSC and compare its performance with the existing duty cycle control approaches in the literature.

Many wireless sensor networks cannot be operated in isolation; the sensor network

must be connected to an external network through which monitoring and controlling entities can reach the sensor network. Hence, designing an interface in the proposed framework to connect to the rest world through TCP/IP would be an important extension.

Critical applications, which are highly sensitive to task failures, require to implement a distributed security mechanism over the communication protocols. Due to the lack of infrastructure in the deployment area, the nodes in WSNs can be easily compromised to perform their tasks. For example, an intruder can pretend to be a destination actor preventing the event readings to be sent to the right actor and in turn escaping the event region to get proper treatment. Thus employment of some security mechanism is important in order to accomplish the given task.

## BIBLIOGRAPHY

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey" *IEEE Communications Magazine*, pp. 102-114, August 2002.
- [2] I. F. Akyildiz, and I. H. Kasimoglu, "Wireless Sensor and Actor Networks: Research Challenges," *Ad Hoc Networks*, Vol. 2, Issue 4, pp. 351-367, October 2004.
- [3] H. Peng, G. Chuanshan, Y. Min, M. Dilin, and Y. Bo, "ECLS: An Efficient Cooperative Localization Scheme For Wireless Sensor and Actor Networks," *In Proc. of the Fifth International Conference on Computer and Information Technology (CIT'05)* pp. 396-400, 2005.
- [4] C. Liu, K. Wu, and T. He. "Sensor Localization with Ring Overlapping Based on Comparison of Received Signal Strength", *IEEE Mobile Ad hoc and Sensor Systems (MASS)*, pp 516-518, October 2004.
- [5] T. He, C. Huang, B. M. Blum, and J. A. Stankovic, Tarek Abdelzaher, "Range Free Localization Schemes for Large Scale Sensor Networks", *In proceedings of MobiCom*, 2003.
- [6] N.B. Priyantha, H. Balakrishnan, E.D. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks", *IEEE, INFOCOM*, March 2005.
- [7] L. Fang, W. Du, and P. Ning, "A Beacon-Less Location Discovery Scheme for Wireless sensor Networks", *IEEE INFOCOM*, 2005.

- [8] N. Bulusu, D. Estrin, L. Girad, and J. Heidemann, "Scalable Coordination for WSN:Self Configuring Location Systems", *ISCTA'01, Ambleside, Lake District, UK, July 2001*.
- [9] R. L. Moses, R. M. P., and W. Garber, "Self Localization of Acoustic Sensor Networks", *2002 MSS Specialty Group on Battlefield Acoustic and Seismic Sensing, Magnetic and Electric Field Sensors, Laurel, MD, Oct. 7-10, 2003*.
- [10] J.C. Chen, L. Yip, J. Elson, H. Wang, D. Maniezzo, R.E. Hudson, K. Yao, and D. Estrin, "Coherent Acoustic Array Processing and Localization on Wireless Sensor Networks", *Proc. The IEEE, 91(8):1154-1162, August 2003*.
- [11] R. Niu, and P. K. Varshney, "Target Location Estimation in Sensor Networks with Quantized Data", *IEEE Transactions on Signal Processing, October 2004*.
- [12] V. Ramadurai, and M. L. Sichitiu, "Localization in Wireless Sensor Networks: A Probabilistic Approach", *International Conference on Wireless Networks, 2003*.
- [13] K. Langendoen, and N. Reijers, "Distributed Localization in WSN:A Quantitative Comparison", *Elsevier, Computer Networks, 2003*.
- [14] X. Sheng, and Y. H. Hu, "Collaborative Source Localization in Wireless Sensor Network System". *IEEE Globecom 2003*.
- [15] C. Liu, and K. Wu, "Performance Evaluation of Range-Free Localization Methods", *Mobicom 2003, Sept.14-19 2003*.
- [16] Ling Xuan Hu, and David Evans, "Localization for Mobile Sensor Networks", *ACM Mobicom, pages 45-57, 2004*.
- [17] Sameer Tilak, Vinay Kolar, Nael B. Abu-Ghazaleh, and Kyoung-Don Kang, "Dynamic Localization Protocols for Mobile Sensor Networks," *IEEE International Work-*

*shop on Strategies for Energy Efficiency in Ad Hoc and Sensor Networks (IWSEEASN)*  
*April 2005.*

- [18] R. L. Moses, D. Krishnamurthy, and R. Patterson, "A Self-Localization Method for Wireless Sensor Networks", *Eurasip Journal on Applied Signal Processing*, March 2003.
- [19] M. L. Sichitiu, and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon", *IEEE International Conference on MASS*, 2004.
- [20] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-Less Low Cost Outdoor Location for Very Small Devices", *IEEE Comm. Magazine* 7(5), pp 28-34, October 2000.
- [21] A. Nasipuri, and K. Li, "A Directionality Based Location Discovery Scheme for Wireless Sensor Networks", *In Proc. of ACM WSNA'02 September 2002*.
- [22] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad hoc networks of sensors," *In ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, July 2001.
- [23] N. Patwari, and A. O. Hero III, "Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks", *IEEE/ACM 2nd Workshop on Wireless Sensor Networks & Applications*, September 2003.
- [24] X. Cheng, A. Thaeler, G. Xue, and D. Chen, "TPS: a Time-based Positioning Scheme for outdoor wireless sensor networks," *In Proc. of IEEE INFOCOM*, 2004.
- [25] L. Lazos, and R. Poovendran, "SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks," *In Proc. of the ACM workshop on Wireless security (WISE'04)*, October 2004.
- [26] D. Niculescu, and B. Nath, "DV Based Positioning in Ad hoc Networks," *Journal of Telecommunication Systems*, Vol. 1 2003.

- [27] A. Günther, and C. Hoene, “Measuring Round Trip Times to Determine the Distance between WLAN Nodes,” *In Proc. of Networking 2005, Waterloo, Canada, May 2005*.
- [28] T. Ajdler, I. Kozintsev, R. Lienhart, and M. Vetterli, “Acoustic Source Localization in Distributed Sensor Networks”, .
- [29] K. Yao, R. E. Hudson, C. W. Reed, D. Chen, and F. Lorenzelli, “Blind beamforming on a randomly distributed sensor array system”, *IEEE J. Selected Areas in Communications*, 16 (1998) 1555-1567.
- [30] C. W. Reed, R. Hudson, and K. Yao, “Direct joint source localization and propagation speed estimation,” *In Proc. ICASSP’99, Phoenix, AZ, (1999) 1169-1172*.
- [31] I. Chatzigiannakis, A. Kinalis, and S. Nikolettseas, “An Adaptive Power Conservation Scheme for Heterogeneous Wireless Sensor Networks with Node Redeployment,” *Proc. of the 17th annual ACM SPAA’05, July 2005*.
- [32] E. J. Duarte-Melo, and M. Liu, “Analysis of Energy Consumption and Lifetime of Heterogeneous Wireless Sensor Networks,” *GLOBECOM’02. IEEE vol. 1 pp. 21-25, November 2002*.
- [33] V. Mhatre, and C. Rosenberg, “Homogeneous vs heterogeneous clustered sensor networks: A comparative study,” *IEEE International Conference on Communications ICC’04 , vol. 27, no. 1, pp. 3646-3651, June 2004*.
- [34] M. J. Handy, M. Haase, and D. Timmermann, “Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection,” *IEEE International Conference on Mobile and Wireless, 2002*.
- [35] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “Energy efficient communication protocol for wireless microsensor networks,” *Proc. of the 33rd Hawaii International Conference on System Sciences (HICSS-33), January 2000*.

- [36] G. Smaragdakis, I. Matta, and A. Bestavros “SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks,” *2nd International Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004)*.
- [37] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “Minimizing communication costs in hierarchically-clustered networks of wireless sensors,” *Computer Networks*, vol. 44, no. 1, pp. 1-16, January 2004.
- [38] M. Chatterjee, S.K. Das, and D. Turgut, “A Weight Based Distributed Clustering Algorithm for Mobile Ad hoc Networks,” *High Performance Computing (HiPC)*, pp. 511-521, 2000.
- [39] G. Gupta, and M. Younis , “Performance Evaluation of Load-Balanced Clustering of Wireless Sensor Networks,” *10th International Conference on Telecommunications (ICT’03)*, vol. 2, March 2003.
- [40] A.D. Amis, and R. Prakash, “Load-Balancing Clusters in Wireless Ad Hoc Networks”, *Proc. of ASSET 2000, Richardson, Texas, March 2000*.
- [41] A.D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Huynh, “Max-Min-D-Cluster Formation in Wireless Ad hoc Networks,” *Proc. of the IEEE INFOCOM-00, Tel Aviv, Israel, March 2000*.
- [42] S. Bandyopadhyay, and E.J. Coyle, “An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks,” *IEEE INFOCOM 2003*.
- [43] S. Basagni, “Distributed Clustering for Ad Hoc Networks,” *Proc. of the International Symposium on parallel architectures, Algorithms and Networks, 1999*.
- [44] N. Heo, and P.K. Varshney, “Energy-efficient deployment of Intelligent Mobile sensor networks,” *Systems, Man and Cybernetics, Part A, IEEE Transactions on* vol. 35, issue 1, pp. 78-92, January 2005.



- [45] S. S. Dhillon, and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," *Proc. of the IEEE Wireless Communication Networking Conference*, pp. 1609-1614, 2003.
- [46] Y. Hu, N. Yu, and X. Jia, "Energy efficient real-time data aggregation in wireless sensor networks," in *Proc. of the International Wireless Communications and Mobile Computing Conference (IWCMC 2006)*, pp. 803-808, July 2006.
- [47] K. Akkaya, M. Younis, and M. Youssef, "Efficient aggregation for delay-constrained data in wireless sensor networks," in *Proc. of Internet Compatible QoS in Ad Hoc Wireless Networks*, 2005.
- [48] T. He, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "AIDA: Adaptive Application-Independent Data Aggregation in Wireless Sensor Networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 2, pp. 426-457, May 2004.
- [49] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz, "A Distributed Coordination Framework for Wireless Sensor and Actor Networks," *ACM MobiHoc'05*, May 2005.
- [50] E.C.H. Ngai, M.R. Lyu, and J. Liu, "A Real-Time Communication Framework for Wireless Sensor-Actuator Networks," in *Proc. of IEEE Aerospace Conference*, pp. 1-9, March 2006.
- [51] H. Peng, W. Huafeng, M. Dilin, and G. Chuanshan, "ELRS: An Energy-Efficient Layered Routing Scheme for Wireless Sensor and Actor Networks," *In Proc. of the 20th Intl Conference on Advanced Information Networking and Applications*, pp. 452-460, 2006.
- [52] X. Z. Yang, Z. GuangSheng, D. WenHua, and F. Qi, "A Hop-Bounded Single-actor Selection Algorithm for Wireless Sensor and Actor Networks," *In Proc. of the Intl conference on Communications and mobile computing*, pp. 1067-1072, Jul 2006.

- [53] K. Akkaya, and M. Younis, "An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks," *23rd ICDCSW*, pp. 710-715, 2003.
- [54] D. Tian, and N.D. Georganas, "Energy Efficient Routing with Guaranteed Delivery in Wireless Sensor Networks," *Mobile Computing and Communications Review*, 2001.
- [55] Y. Yu, R. Govindan, and D. Estrin, "Geographical and Energy Aware Routing (GEAR): a recursive dissemination protocol for wireless sensor networks," *UCLA/CSD-TR-01-0023, Tech. Rep.*, 2001.
- [56] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. ACM MobiCom*, March 2000.
- [57] J. Broch, D. B. Johnson, D. A. Maltz, Y.-C. Hu, and G. Jetcheva. "The dynamic source routing protocol for mobile ad hoc networks," *Internet- Draft, draft-ietf-manet-dsr-05.txt*, 2001.
- [58] T. He, J.A. Stankovic, C. Lu, and T. Abdelzaher, "A Real-Time Routing Protocol for Sensor Networks," *Proc. IEEE International Conference on Distributed Computing Systems*, May 2003.
- [59] V. Rodoplu, and T.H. meng, "Minimum Energy Mobile Wireless Networks," *IEEE JSAC*, August 1999.
- [60] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocol for Information Dissemination in Wireless Sensor Networks," *Proc. ACM MobiCom*, 1999.
- [61] Q. Sun, and H. Langendörfer, "A new distributed routing algorithm for supporting delay-sensitive applications," *Journal of Computer Communications vol. 9 no. 6 May 1998*.

- [62] R. Gawlick, A Kamath, S. Poltkin, and K. G. ramkrishnan “Routing and admission control in general topology networks,” *Tech. Report STAN-CS-TR-95-1548, Stanford University, CA, 1995.*
- [63] G. A. Shah, M. Bozyigit, Ö. B. Akan, and B. Baykal, “Real-time Coordination and Routing in Wireless Sensor and Actor Networks,” *To appear in Proc. of 6th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN), May 2006.*
- [64] S. Kapadia<sup>1</sup>, and B. Krishnamachari, “Comparative Analysis of Push-Pull Query Strategies for Wireless Sensor Networks,” *To appear in International Conference on Distributed Computing in Sensor Systems (DCOSS), June 2006.*
- [65] A. Durresi, V. Paruchuri, and L. Barolli, “Delay-Energy Aware Routing Protocol for Sensor and Actor Networks,” *Proc. of the 2005 11th International Conference on Parallel and Distributed Systems (ICPADS’05).*
- [66] K. Liu, and N. Abu-Ghazaleh, “HGR: Hybrid Geographic and Virtual Coordinate Routing for WSN,” *<http://arxiv.org/abs/cs.NI/0511059>, Nov 2005.*
- [67] A. Aksoy, and M. Leung, ”Pull vs. Push: A Quantitative Comparison,” *IEEE GLOBECOM, Dec 2004.*
- [68] J. Kulik, W. Heinzelman, and H. Balakrishnan, “Negotiation-based protocols for disseminating information in wireless sensor networks,” *Wireless Networks, Vol. 8, 2002, pp. 169-185.*
- [69] B. Karp, and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” *MobiCom 2000.*
- [70] X. Liu, Q. Huang, and Y. Zhang, “Combs, Needles, Haystacks: Balancing Push and Pull for Discovery in LargeScale Sensor Networks,” *emphACM Sensys 2004, November 3-5, 2004.*

- [71] N. Trigoni, Y. Yao, A. Demers, and J. Gehrke, "Hybrid Push-Pull Querying Processing for Sensor Networks," *In Proc. of the Workshop on Sensor Networks (WSN) 2004*.
- [72] G. Hoblos, M. Staroswiecki, and A. Aitouche, "Optimal Design of Fault Tolerant Sensor Networks," *In Proc. of IEEE International Conference on Control Applications*, pp. 467-472, Anchorage, AK, September 2000.
- [73] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," *In Proc. of ACM MobiCom'01, Rome, Italy, July 2001*, pp. 272-286.
- [74] E. Felemban, C.G. Lee, E. Ekici, R. Boder, and S. Vural, "Probabilistic QoS Guarantee in Reliability and Timeliness Domains in Wireless Sensor Networks," *in Proc. of IEEE INFOCOM 2005, Miami, FL, USA, March 2005*.
- [75] I. F. Akyildiz, M. C. Vuran, and O. B. Akan, "On Exploiting Spatial and Temporal Correlation in Wireless Sensor Networks," *in Proc. IEEE WiOpt 2004, University of Cambridge, UK, March 2004*.
- [76] S. Yoon, and C. Shahabi, "Exploiting Spatial Correlation Towards an Energy Efficient Clustered AGgregation Technique (CAG)," *IEEE International Conference on Communications (ICC)*, vol. 5, pp. 3307-3313, May 2005.
- [77] Y. Xu, and W. Lee, "Exploring Spatial Correlation for Link Quality Estimation in Wireless Sensor Networks," *in Proc. Fourth Annual IEEE International Conference on Pervasive Computing and Communications ( PerCom 2006) March 2006*.
- [78] M. C. Vuran, and I. F. Akyildiz, "Spatial Correlation-based Collaborative Medium Access Control in Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 316-329, April 2006.

- [79] M. Gastpar, and M. Vetterli, "Source-Channel Communication in Sensor Networks," *in Proc. of 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*, vol. 219, pp. 162-177, 2003.
- [80] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed Compression in a Dense Microsensor Network," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 51-60, Mar 2002.
- [81] A. Coman, M. A. Nascimento, and J. Sander, "Exploiting redundancy in sensor networks for energy efficient processing of spatiotemporal region queries," *In Proc. of the 14th ACM international conference on Information and knowledge management*, pp. 187-194, 2005.
- [82] S. Fahmy, and O. Younis, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks", *IEEE Transactions on Mobile Computing*, vol. 3, Issue 4 pp. 366-379, October 2004.
- [83] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and Tian He, "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks," *In Proc. of the 8th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'02)*, 2002.
- [84] C. Liu, K. Wu, and J. Pei, "An Energy Efficient Data Collection Framework for Wireless Sensor Networks by Exploiting Spatiotemporal Correlation," <http://www.cs.uvic.ca/wkui/research/spatiotemporal-wu.pdf>, Jun 2006.
- [85] J. Kay, and J. Frolik, "Quality of Service Analysis and Control for Wireless Sensor Networks," *1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, October 2004.
- [86] I. Ranjit, and L. Kleinrock, "QoS Control for Sensor Networks," *IEEE International Communications Conference (ICC 2003)*, Anchorage, AK, May 2003.

- [87] Frolik, J. "QoS Control for wireless sensor networks," *Wireless Communication and Networking Conference (WCNC), Atlanta, GA, March 2003*.
- [88] D. Chen, and P. K. Varshney, "QoS Support in Wireless Sensor Networks: A Survey," *In Proc. of International Conference on Wireless Networks (ICWN 2004), June 2004*.
- [89] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran, "A Survey of Quality of Service in IEEE 802.11 Networks," *IEEE Wireless Communications, August 2004*.
- [90] M. Younis, K. Akkaya, M. Eltoweissy, and A. Wadaa, "On Handling QoS in Wireless Sensor Networks," *In the Proc. of HAWAII International Conference on System Sciences (HICSS-37), January 2004*.
- [91] S. Bhatnagar, B. Deb, and B. Nath, "Service Differentiation in Sensor Networks," *In the Proc. of Fourth International Symposium on Wireless Personal Multimedia Communications, September 2001*.
- [92] B. Deb, S. Bhatnagar, and B. Nath, "ReInForM: Reliable Information Forwarding using Multiple Paths in Sensor Networks," *In 28th Annual IEEE Conference on Local Computer Networks (LCN 2003), Bonn, Germany, October 2003*.
- [93] B. Deb, S. Bhatnagar, and B. Nath, "Information Assurance in Sensor Networks," *In 2nd International ACM Workshop on Wireless Sensor Networks (WSNA), San Diego, September 2001*.
- [94] Y. Ge, T. Kunz, and L. Lamont, "Quality of Service Routing in Ad-Hoc Networks Using OLSR," *In Proc. of the 36th Hawaii International Conference on System Sciences (HICSS'03), January 2003*.
- [95] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *Mobile Computing and Communications Review (MC2R), 2002*.

- [96] G. J. Pottie, and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, pp. 51-58, May 2000.
- [97] P. P. Pham, and S. Perreau, "Performance Analysis of Reactive Shortest Path and Multi-ath Routing Mechanism with Load Balance," *IEEE INFOCOM 2003*.
- [98] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network communication models," *ACM Mobile Computing and Communication Review(MC2R)*, June 2002.
- [99] Yogesh Sankarasubramaniam, Özgür B. Akan, and Ian F. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks" *MobiHoc'03, Annapolis, Maryland, USA. June 1-3, 2003*.
- [100] F. Stann, and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA'03), 11 May 2003*.
- [101] N. Tezcan, E. Cayirci, and M. U. Caglayan, "End-To-End Reliable Event Transfer in Wireless Sensor Networks," *In Proc. of the IEEE PIMRC, Barcelona, Spain, September 2004*.
- [102] A. Dunkels, J. Alonso, and T. Voigt, "Making TCP/IP Viable for Wireless Sensor Networks," <http://www.sics.se/adam/ewsn2004.pdf>, Dec 2006.
- [103] Crossbow Technology MICAz,  
[http://www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/MICA2\\_Datasheet.pdf/](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf/),  
Oct 2006.
- [104] Atmel Corporation, <http://www.atmel.com>, Jan 2007.
- [105] TinyOS, <http://www.tinyos.net>, Jan 2007.

- [106] UC Berkeley, LBL, USC/ISI and Xerox PARC, “The Network Simulator ns-2,” *The VINT Project*, <http://www.isi.edu/nsnam/ns/>, Jan 2007.
- [107] NRL Sensor Network Extension for ns-2, <http://downloads.pf.itd.nrl.navy.mil/nrlsensorsim/>, Dec 2006.
- [108] I. Haddad, and D. Gordon, “Network Simulator 2: a Simulation Tool for Linux,” <http://www.linuxjournal.com/article/5929>, Dec 2002.



## CURRICULUM VITAE

### PERSONAL INFORMATION

Surname, Name: Shah, Ghalib Asadullah  
Nationality: Pakistan  
Date and Place of Birth: 20 October, 1975, Pakpattan  
Marital Status: Married  
Phone: +90 312 210 5529  
Fax: + 90 312 210 5544  
Email: asad@ceng.metu.edu.tr

### EDUCATION

Degree	Institution	Completion Year
MSc	B. Z. University, Computer Science	2000
BSc	B. Z. University, Mathematics/Physics	1997
Higher Secondary	Govt Tech. College, D.A.E. in Electronics	1994
High School	Govt Islamia High School, Pakpattan	1991

### WORK EXPERIENCE

Year	Place	Enrollment
Jun 2000 - Dec 2001	Elixir Technologies, Islamabad	Software Engineer
Jan 2001 - Dec 2001	COMSATS Institute of IT, Islamabad	Instructor (Part Time)

### FOREIGN LANGUAGES

Advanced English, Fluent Turkish

## PUBLICATIONS

1. G. A. Shah, M. Bozyigit and B. Baykal, "Energy-efficient Real-time Coordination and Routing in Wireless Sensor and Actor Networks," *submitted to ACM WiNet, 2006*.
2. G. A. Shah, M. Bozyigit and Demet Aksoy, "RAT: Routing by Adaptive Targeting in Wireless Sensor/Actor Networks," in Proc. Second IEEE/ACM International Conference on COMMunication System softWARE and middlewaRE (COMSWARE), January 2007.
3. G. A. Shah and M. Bozyigit, "Exploiting Energy-aware Spatial Correlation in Wireless Sensor Networks," in Proc. 2nd International Workshop on Software for Sensor Networks (SensorWare 2007), January 2007.
4. G. A. Shah, Ö. B. Akan and M. Bozyigit, "Multi-Event Adaptive Clustering (MEAC) Protocol for Heterogeneous Wireless Sensor Networks," in Proc. Fifth Annual Mediterranean Ad Hoc Networking Workshop (MedHoc-Net), June 2006.
5. G. A. Shah, M. Bozyigit, Ö. B. Akan and B. Baykal, "Real-time Coordination and Routing in Wireless Sensor and Actor Networks," in Proc. 6th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN), May 2006.

## HOBBIES

Cricket, Gathering with Friends