

The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

School of Electrical and Electronic Engineering

Energy-Efficient Routing Algorithms Based on Swarm Intelligence for Wireless Sensor Networks

ADAMU MURTALA ZUNGERU, B.Eng., M.Sc.

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy in Engineering

February 2013

I would like to dedicate this thesis to The Almighty God (ALLAH).

Acknowledgements

Firstly, I would like to express my profound and sincere gratitude to my supervisor, Associate Professor Dr. Kenneth Ang Li-Minn, his wife, Professor Jasmine Kah Phooi Seng and my co-supervisor, Professor Prabaharan Savari Rathinam, without whom this thesis would have not been completed. More specifically to Associate Professor Dr. Kenneth Ang Li-Minn for identification and refinement of my research ideas and who out of his tight schedule was still able to meet with me for at least once in a week and provide me with valuable guidance, unfailing support and constant encouragement, hence making the thesis a fruitful one, I thank you.

I also owe sincere thanks to the Tertiary Education Trust Fund (TETFund) of Nigeria for providing financial support during this Ph.D studies. I am grateful to the Management and staffs of the Federal University of Technology, Minna, Nigeria for the support and encouragement. I also thank all the staffs of Engineering of the University of Nottingham for their support and encouragement. My research colleagues at BB41, NB11, my uncle, Mallam Ibrahim Wushishi, my friends, Ambafi James Garba, Jibrin Salihu Marafa, Aliyu Abdul, Hussaini Shehu (Saddam), Danlami Ayishi, Laleh Boroumand, Goodness Obiageli, Dione Ulunma Utah and others, numerous to be mentioned, I thank you all for your prayers, financial, moral and friendly supports and encouragements.

This thesis will not be complete if I fail to acknowledge my guardian, Col. Muhammed Sani Bello (RTD), OON, Vice Chairman of MTN Nigeria Communications Limited for supporting this research. I will always remember his words to me before starting and when I was in the process of the research, "Do not forget when the going is tough, I will always be

there for you Inshallah". By this word, I am always encouraged and motivated, and along the way, he has never failed to fulfil the content of the words. I will always remain grateful to you, you are a true father.

Last but not least, my family (Mallam Adamu Bagudu, Hajiya Maryam Adamu, Sadiq, Lanti, Zuwaira (R.I.P), Aisha, Jemira, Semira, Rabi, and my Step ones), my wife (Aishatu Murtala) and the kids (Noor-Faeezah, Nur Faakhir) were always there to let me know that the world has several other attractions that must be cherished other than the Ph.D. research. I appreciate the devotion and support of my wife during all the ups and downs of this Ph.D. journey. I also pay thanks to other members of research monitoring committee, the internal (Dr. Lim Wee Gin), S. Anandan Shanmugam and the external (Prof. Farhat Anwar), for evaluating the quality of my work. Finally, I will like to thank Ms Hasni Haldi, Ms Amaravathany Ratha Krishnan, and other faculty staffs for their support.

Abstract

High efficient routing is an important factor to be considered in the design of limited energy resource Wireless Sensor Networks (WSNs). WSN environment has limited resources in terms of on-board energy, transmission power, processing, and storage, and this prompt for careful resource management and new routing protocol so as to counteract the challenges. This work first introduces the concept of wireless sensor networks, routing in WSNs, and its design factors as they affect routing protocols. Next, a comprehensive review of the most prominent routing protocols in WSN, from the classical routing protocols to swarm intelligence based protocols is presented. From the literature study, it was found that comparing routing protocols in WSNs is currently a very challenging task for protocol designers. Often, much time is required to re-create and re-simulate algorithms from descriptions in published papers to perform the comparison. Compounding the difficulty is that some simulation parameters and performance metrics may not be mentioned. We then see a need in the research community to have standard simulation and performance metrics for comparing different protocols. To this end, we re-simulate different protocols using a Matlab based simulator; Routing Modeling Application Simulation Environment (RMASE), and gives simulation results for standard simulation and performance metrics which we hope will serve as a benchmark for future comparisons for the research community.

Also, from the literature study, Energy Efficient Ant-Based Routing (EEABR) protocol was found to be the most efficient protocol due to its low energy consumption and low memory usage in WSNs nodes. Following this efficient protocol, an Improved Energy Efficient Ant-Based Routing (*IEEABR*) Protocol was proposed. Simulation were performed using Network Simulator-2 (NS-2), and from the results, our proposed algorithm

performs better in terms of energy utilization efficiency, average energy of network nodes, and minimum energy of nodes. We further improved on the proposed protocol and simulation performed in another well-known WSNs MATLAB-based simulator; Routing Modeling Application Simulation Environment (RMASE), using static, mobile and dynamic scenario. Simulation results show that the proposed algorithm increases energy efficiency by up to 9% and 64% in converge-cast and target-tracking scenarios, respectively, over the original EEABR and also found to out-perform other four Ant-based routing protocols. We further show how this algorithm could be used for energy management in sensor network in the presence of energy harvesters.

However, high number of control packets is generated by the IEEABR due to the proactive nature of its path establishment. As such, a new routing protocol for WSNs that has less control packets due to its on-demand (reactive) nature is proposed. This new routing protocol termed Termite-hill is borrowed from the principles behind the termite's mode of communication. We first study the foraging principles of a termite colony and utilize the inspirational concepts to develop a distributed, simple and energy-efficient routing protocol for WSNs. We perform simulation studies to compare the behavior and performance of the *Termite-hill* design with an existing classical and on-demand protocol (AODV) and other Swarm Intelligence (SI) based WSN protocols in both static, dynamic and mobility scenarios of WSN. The simulation results demonstrate that *Termite-hill* outperforms its competitors in most of the assumed scenarios and metrics with less latency.

Further studies show that the current practice in modeling and simulation of wireless sensor network (WSN) environments has been towards the development of functional WSN systems for event gathering, and optimization of the necessary performance metrics using heuristics and

intuition. The evaluation and validation are mostly done using simulation approaches and practical implementations. Simulation studies, despite their wide use and merits of network systems and algorithm validation, have some drawbacks like long simulation times, and practical implementation might be cost ineffective if the system is not properly studied before the design. We therefore argue that simulation based validation and practical implementation of WSN systems and environments should be further strengthened through mathematical analysis. To conclude this work and to gain more insight on the behavior of the termite-hill routing algorithm, we developed our modeling framework for WSN topology and information extraction in a grid based and line based randomly distributed sensor network. We strengthen the work with a model of the effect of node mobility on energy consumption of Termite-hill routing algorithm as a function of event success rate and occasional change in topology. The results of our mathematical analysis were also compared with the simulation results.

List of Publications

To date, research outputs from this work had been published in international journals and books, and others presented in international conferences. Publications arising from this work are listed below:

Peer-reviewed Journal Publications

1. A.M. Zungeru, L.-M. Ang, K.P. Seng, "Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison", *Journal of Network and Computer Applications, ELSEVIER*, vol. 35, Issue 5, pp. 1508-1536, September 2012.

2. A.M. Zungeru, L.-M. Ang, K.P. Seng, "Termite-hill: Performance optimized swarm intelligence based routing algorithm for wireless sensor networks", *Journal of Network and Computer Applications, ELSEVIER*, vol. 35, Issue 6, pp. 1901-1917, November, 2012.

3. A.M. Zungeru, L.-M. Ang, K.P. Seng, "A Formal Mathematical Framework for Modeling and Simulation of Energy Consumption in Wireless Sensor Network System Environments utilizing the Hill Building Behaviors of Termites", Special issue, SIMULATION SI: Agent-based Modeling and Simulation of Complex Adaptive Communication Networks and Environments (CACOONS). In: *Simulation: Transactions of the Society for modeling and simulation international 2012***.**
doi:10.1177/0037549712466706

4. A.M. Zungeru, L.-M. Ang, K.P. Seng, "Performance Evaluation of Ant Based Routing Protocols for Wireless Sensor Networks", *International Journal of Computer Science Issues (IJCSI)*, vol. 9, issue 3(2), pp. 388-397, 2012.

5. A.M. Zungeru, L.-M. Ang, K.P. Seng, "Termite-hill: From Natural to Artificial Termites in Sensor Networks", International Journal of Swarm Intelligence Research (IGI-Publishers), vol. 3(4), pp. 1-23, 2012.

6. A.M. Zungeru, K.P. Seng, L.-M. Ang, W.C. Chia, "Energy Efficiency Performance Improvements for Ant-Based Routing Algorithm in Wireless Sensor Networks", *Journal of Sensors, vol. 2013, Article ID 759654, 17 pages, 2013. doi:10.1155/2013/759654*

Peer-reviewed Book Chapters

1. A.M. Zungeru, Li-Minn Ang, and Kah Phooi Seng, "Performance of Termite-hill Routing Algorithm on Sink Mobility in Wireless Sensor Networks", *In: Advances in Swarm Intelligence*, Lecture notes in computer science (LNCS) 7332(2), pp. 334-343, Springer 2012.

2. A.M. Zungeru, L.-M. Ang, SRS. Prabaharan, K.P. Seng, "Ant Based Routing Protocol for Visual Sensors", *In: Communications in Computer and Information Science (CCIS) series of Lecture Notes in Computer Sci ence (LNCS), vol. 252(3), pp. 250-264, Springer 2011*.

3. A.M. Zungeru, L.-M. Ang, SRS. Prabaharan, K.P. Seng, "Improved Energy Efficient Ant-Based Routing Algorithm in Wireless Sensor Networks", *In: Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management. K. Ragab, A.B. Abdullah, and N. Zaman (Eds.)*: *Chapter 20, pp. 420-444. Hershey, PA 1 7033, USA: IGI- Global, 2011*.

4. A.M. Zungeru, L.-M. Ang, SRS. Prabaharan, K.P. Seng, "Radio Frequency Energy Harvesting and Management for Wireless Sensor Networks"*, In: Energy Scavenging and Optimization Techniques for Mobile Devices. V. Hrishikesh, and G -M. Mountean (Eds.) USA: Chapter 13, pp. 341-367, CRC Press, Taylor and Francis Group, 2011.*

Referred Conference Publication

1. A.M. Zungeru, L.-M. Ang, K.P. Seng, "Termite-hill: Routing towards a Mobile Sink for Improving Network Lifetime in Wireless Sensor Network" , *in: IEEE- 3rd International conference on Intelligent Systems, Modeling and Simulation (ISMS*), pp. 622-627, 2012. (ISBN:978-0-7695-4668-1)

Table of Contents

Abstract V

List of Publications VIII

Table of Figures XVI

List of Tables XVIVIII

List of Pseudocodes XX

List of Abbreviations XXI

1 Introduction 1

1.1 Introduction 1

1.2 Swarm intelligence 7

1.2.1 The principles of swarm intelligence 8

1.3 Motivation 10

1.4 Research objectives 10

1.5 Problem statement 14

1.6 Research methodology 16

1.7 Major contributions of the research 17

1.7.1 A model of comparison among routing protocols in WSNs 18

1.7.2 An improvement on an existing state-of-the-art routing protocol in WSN 18

1.7.3 A new Termite-inspired routing algorithm for WSN 19

1.7.4 Formal mathematical modeling of wireless sensor network environment 19

1.8 Organization of the Thesis 20

1.8.1 Chapter 2: Survey and comparison of routing protocols in wireless sensor networks 20

1.8.2 Chapter 3: Improved energy efficient ant based routing algorithm 21

1.8.3 Chapter 4: Termite-hill: Termite inspired routing algorithm in WSNs 22

1.8.4 Chapter 5: A Formal Mathematical Framework for Modeling and Simulation of Wireless Sensor Network Environments 23

1.8.5 Chapter 6: Radio frequency energy harvesting and management for wireless sensor networks 24

1.8.6 Chapter 7: Conclusions and future works 25

2 Survey and Comparison of Routing Protocols in WSNs 26

2.1 Introduction.....26

2.2 Previous surveys of WSN routing.....27

2.3 WSNs design and routing factors 29

2.4 Taxonomy of routing protocols in WSNs 34

2.5 Classical based routing protocols 38

2.5.1 Classical based data-centric routing protocols 39

2.5.2 Location-based protocols..... 49

2.5.3 Hierarchical protocols..... 52

2.5.4 Network flow and QoS-aware protocols..... 57

2.6 Swarm intelligence based routing protocols 63

2.6.1 Ant Colony Optimization (ACO)..... 65

2.6.2 Swarm based data-centric routing protocols 66

2.6.3 Swarm Based Location-Based Protocols 67

2.6.4 Swarm based hierarchical protocols 68

2.6.5 Swarm based network flow and QoS-aware protocols 75

2.7 Analytical comparison of classical and swarm intelligence routing protocols.....85

2.7.1 Comparison of data-centric classical and swarm intelligence routing protocols..... 86

2.7.2 Discussion on Analytical Comparison of Data-Centric Routing Protocols..... 86

2.7.3 Discussion on Analytical Comparison of Location Based Routing Protocols 88

2.7.4 Discussion on Analytical Comparison of Network flow and QoS-Aware Routing Protocols 93

2.8 Experimental Comparison of Classical and Swarm Intelligence Routing Protocols 93

2.8.1 Experimental Parameters 93

2.8.2 Performance metrics..... 96

2.8.3 Simulation results 96

2.9 General discussion on the reviewed routing protocols in WSN ...104

2.10 Conclusions and future direction.....106

3 Improved Energy-Efficient Ant Based Routing Algorithm.....108

3.1 Introduction.....108

3.2 Related work 109
3.3 Application scenarios as it affect routing in WSNs 111
3.4 The social insect analogy 115
3.4.1 Positive Feedback 116
3.4.2 Negative Feedback 116
3.4.3 Randomness 117
3.4.4 Multiple Interactions 117
3.4.5 Stigmergy 117
3.5 Improved Energy Efficient Ant-Based Routing algorithm (IEEABR) 118
3.5.1 Energy-Efficient Ant-Based Routing Algorithm (EEABR) 118
3.5.2 Improvements on Energy-Efficient Ant-Based Routing Algorithm 121
3.6 Experimental comparisons using NS-2 simulation environment. 132
3.6.1 The Trace File 133
3.6.2 Simulations Environment 134
3.6.3 Simulation Results 136
3.6.4 Discussion 138
3.7 Experimental comparison using Matlab based simulation environment 138
3.7.1 Performance evaluation metrics 140
3.7.2 Comparison of experimental results for IEEABR, SC, EEABR and Beesensor 140
3.7.3 Discussion 153
3.7.4 Performance Evaluation of Ant-Based Routing Protocols for Wireless Sensor Networks 154
3.8 Conclusions 164

4 Termite-hill: Termite inspired routing algorithm in WSNs 165

4.1 Introduction 165
4.2 The Termite Analogy 169
4.2.1 Positive Feedback 171
4.2.2 Negative Feedback 172
4.2.3 Randomness 172
4.2.4 Multiple Interactions 172
4.2.5 Stigmergy 173
4.3 Simulating the behavior of termites 173

4.4 The Termite-hill Routing Algorithm 183
4.4.1 The Pheromone Table 185
4.4.2 Route Selection 189
4.4.3 Termite-hill Agent Model 190
4.4.4 The Termite-hill Routing Algorithm Modules Design 193
4.4.2 Data Packet Module 201
4.4.3 Route Maintenance Module 202
4.5 Performance Evaluation 203
4.5.1 Performance of Termite-hill with Static Sink 206
4.5.2 Performance of Termite-hill using Dynamic Sink (Target Tracking) 210
4.5.3 Performance of Termite-hill with a Mobile Sink 215
4.5.4 Performance of Termite-hill in terms of Network Reliability 218
4.6 Conclusions 220

5 A Formal Mathematical Framework for Modeling and Simulation of WSN Environments 222

5.1 Introduction 222
5.1.1 Organization of the chapter 224
5.2 Related Work 224
5.3 Network System Description and Definitions 226
5.3.1 Network description 226
5.3.2 Network topology 228
5.3.3 Energy model 234
5.4 Event Extraction Analysis in Wireless Sensor Network 243
5.5 Computational and Simulation Experiments 246
5.5.1 Average Transmission Cost of a Grid Based Sensor Network 247
5.5.2 Energy Consumption of a Direct and Multihop Communication in WSN 250
5.5.3 Energy Consumption of a Termite-hill Routing Protocol in WSN 252
5.6 Conclusions 257

6 Radio Frequency Energy Harvesting and Management for WSN . 259

6.1 Introduction 259
6.2 Review of Energy Harvesting Systems and Power Consumption in WSNs 262

6.2.1 Ambient RF Sources and Available Power 264
6.3 Radio frequency energy harvesting and the use of powercast harvester 265
6.3.1 Intentional Energy Harvesting 266
6.3.2 The Powercast TX91501 Powercaster Transmitter 267
6.3.3 The Powercast Power Harvester Receiver 268
6.4 Measurement of RF Power Received and Gains 269
6.4.1 Friis Transmission Equation 270
6.4.2 Power Density 270
6.5 Energy storage 271
6.6 Energy Management in WSNs 273
6.7 Experiment and Simulation Results 275
6.7.1 Experimental Results 275
6.7.2 Simulation Results 279
6.7.3 Real-time Implementation of RF Powercast Energy Harvester 282
6.8 Conclusions 285

7 Conclusions and Future Work 286

7.1 Conclusions 286
7.2 Future works 289
REFERENCES 290

List of Figures

Fig. 1.1: WSN node architecture and a real example. 3

Fig. 1.2: Application view of sensor network. 13

Fig. 1.3: A multi-hop physical topology sensor network. 15

Fig. 2.1: (a) Implosion problem and (b) The overlap problem. 40

Fig. 2.2: Directed Diffusion routing scheme. 43

Fig. 2.3: SPIN protocol phases.....44

Fig. 2.4: plan at a leader node.....48

Fig. 2.5: Hierarchical network architecture. 53

Fig. 2.6: Description of pheromone table of node C... 64

Fig. 2.7: signaling process phases of multi-sink swarm based routing.....74

Fig. 3.1: Application scenario of WSN in static and mobility of sink 113

Fig. 3.2: Application scenario of WSN in Dynamic (Target Tracking) 114

Fig. 3.3: Food Search in Ants... 116

Fig. 3.4: An IEEABR forward ant flow chart. 126

Fig. 3.5: Data packet content... 127

Fig. 3.6: Description of pheromone table of node G.....129

Fig. 3.7(a): Graphical Representation of NS-2 Environment (120 Nodes)135

Fig. 3.7(b): Graphical Representation of NS-2 Environment (10 Nodes). 135

Fig. 3.8: Performance analysis of IABR, EEABR and IEEABR Protocols....137

Fig. 3.9: Performance evaluation in Converge-cast scenario. 150

Fig. 3.10: Performance evaluation in Target-tracking scenario.....152

Fig. 3.11: Matlab Simulation Environment shows. 155

Fig. 3.12: Performance evaluation in static scenario. 159

Fig. 3.13: Performance evaluation in dynamic scenario. 163

Fig. 4.1: Food Search in Termites. 171

Fig. 4.2: Simulation of Termites with 100 termites and 200 wood. 177

Fig. 4.3: Behavioral pattern of termites gathering wood samples. 181

Fig. 4.4: Description of pheromone table of node G. 186

Fig. 4.5: Forward Soldier from node S to destination D. 194

Fig. 4.6: Backward Soldier from Node D to source node S. 200

Fig. 4.7: Performance evaluation in static scenario. 209

Fig. 4.8: Performance evaluation in dynamic scenario. 213

Fig. 4.9: Performance evaluation in dynamic with varying speed. 214

Fig. 4.10: Performance evaluation in mobility scenario (varying speed). 217

Fig. 4.11: Evaluation of impact of sink mobility on network reliability. .. 219

Fig. 5.1: Modeling the topology of a wireless sensor network. 228

Fig. 5.2: Illustration of graph connectivity. 230

Fig. 5.3: A rectangular x-y grid topology in wireless sensor network. 231

Fig. 5.4: First order radio model (single hop transmission). 233

Fig. 5.5: A Linear Network. 239

Fig. 5.6: Effect of (a) Transmission distance (b) Packet Size, and (c) Number of Nodes on the Transmission cost of a grid based sensor network. 249

Fig. 5.7: Energy Consumption of a Direct and Multihop Communication . 251

Fig. 5.8: Effect of high number of retransmission on termite-hill. 254

Fig. 5.9: Effect of varying number of retransmission termite-hill. 255

Fig. 5.10: Analytical and Simulation Energy Consumption of termite-hill 256

Fig. 6.1: View of intentional, anticipated, and unknown energy sources. 266

Fig. 6.2: (a), a side view, and (b) a real view of a TX91501 Powercaster 267

Fig. 6.3: Time difference between Packets and Distance of harvesting. .. 278

Fig. 6.4: Power harvesting using Dipole and Patch Antenna with P2110. 279

Fig. 6.5: Performance analysis of AODV, EEABR and IEEABR Protocols. . 281

Fig. 6.6: Hardware setup of the real-time implementation. 284

List of Tables

Table 1.1: Classifying applications of WSNs based on areas of application 2

Table 2.1: Summary of previous survey of routing protocols in WSN 28

Table 2.2: Taxonomy of routing protocol's classification in WSNs..... 37

Table 2.3: Comparison of data-centric routing protocols in WSNs 87

Table 2.4: Comparison of Location Based Routing Protocols in WSNs 88

Table 2.5: Comparison of hierarchical routing protocols in WSNs..... 89

Table 2.6: Comparison of network flow and QoS-aware routing protocols in WSNs 91

Table 2.7: Simulation parameters..... 95

Table 2.8: Simulation results for Energy-Efficient Ant-Based Routing Protocol 97

Table 2.9: Simulation results for Sensor-driven and Cost-aware ant Routing (SC) ... 97

Table 2.10: Simulation results for Flooded Forward ant Routing (FF)..... 98

Table 2.11: Simulation results for Flooded Piggybacked ant routing (FP)..... 98

Table 2.12: Simulation results for Beesensor routing 99

Table 2.13: Simulation results for Basic Ant-Based Routing (BABR) 99

Table 2.14: Simulation results for Ad-hoc On-demand Distance Vector protocol.... 100

Table 2.15: Simulation results for Adaptive Spanning Tree (MCBR-AST)..... 100

Table 2.16: Simulation results for Real-Time Search Routing (MCBR-RTSR)..... 101

Table 2.17 Simulation results for Constrained Flooding Routing (MCBR-CFR) 101

Table 2.18: Comparison of the routing protocols based on different metrics..... 102

Table 2.19: Proposed models for routing protocol comparison in WSNs..... 103

Table 3.1: Comparison of Routing Protocols in Converge-cast Scenario 147

Table 3.2: Comparison of Routing Protocols in Target-tracking Scenario 148

Table 4.1: Simulation parameters for SC, FF, AODV and Termite-hill 205

Table 4.2: Impact of sink mobility on network lifetime for routing protocols 216

Table 4.3: Comparison of the Routing Protocols in Dynamic Scenario 216

Table 5.1: Analytical and Simulation Parameters 247

Table 6.1: Comparison of Energy Harvesting Sources for WSNs 263

Table 6.2: Comparison of Power consumption of sensor network nodes 264

Table 6.3: Comparison of the Two RF Energy Powercast receivers 269

Table 6.4: (a) Amount of Power harvested by P2110 using Dipole Antenna 276

Table 6.4: (b) Amount of Power harvested by P2110 using Patch Antenna 276

Table 6.4: (c) Amount of Power harvested by P1110 using Dipole Antenna 277

Table 6.4: (d) Amount of Power harvested by P1110 using Patch Antenna 277

Table 6.5: Simulation Parameters for AODV, EEABR and IEEABR 280

List of Pseudocodes

Pseudocode 3.1 IEEABR Algorithm Pseudocode 129

Pseudocode 4.1 Simulation of artificial termites in a real-world behavior Algorithm 177

Pseudocode 4.2 Route Discovery in Termite-hill Routing Algorithm 197

Pseudocode 4.3 Route Update in Termite-hill Routing Algorithm 200

Pseudocode 4.4 Working Group in Termite-hill Routing Algorithm 202

Pseudocode 4.5 Working Group at intermediate nodes in Termite-hill Algorithm 202

List of Abbreviations

ACALEACH:	Ant Colony Clustering Algorithm
ACLR:	Ant Colony Optimization Based Location Aware Routing
ACMRA:	Ant Colony Based Multipath Routing Algorithm
ACMT:	Ant Colony Multicast Trees
ACO:	Ant Colony Optimization
ACO-C:	Adaptive Clustering for energy efficient WSN based on ACO
ACO-EAMRA:	Ant Colony-based Energy-Aware Multipath Routing Algorithm
ACOLBR:	An Ant Colony Optimization-based Load Balancing Routing Algorithm
ACO-QoSR:	ACO-based Quality-of-Service Routing
ACQUIRE:	ACtive Query forwarding In sensor nEtworks
AI:	Artificial Intelligence
AL:	Artificial Life
AntChain:	Data gathering communication
AntSensNet:	Ant-Based Routing for Wireless Multimedia Sensor Networks
AODV:	Ad-hoc On-demand Distance Vector
APTEEN:	Adaptive Threshold Sensitive Energy Efficient Sensor Network Protocol
AR and IAR:	Ant Colony Based Reinforcement Learning Algorithm
AR:	Adaptive Routing
ASAR:	Ant Based Service-Aware Routing Algorithm
ASK:	Amplitude Shift Keying
AWGN:	Additive White Gaussian Noise
BA:	Backward Ant
BABR:	Basic Ant Based Routing

BCEE:	A Balanced-Clustering Energy-Efficient Hierarchical Routing Protocol
BCO:	Bee Colony Optimization
Beesensor:	A Bee-inspired power aware routing
BS:	Backward Soldier
BWACA:	Best-Worse Ant Algorithm
CADR:	Constrained Anisotropic Diffusion Routing
CBR:	Constant Bit Rate
CDMA:	Code Division Multiple Access
CMLDA:	Cluster-based Maximum Lifetime Data Aggregation
CRP:	Comprehensive Routing Protocol
CSMA:	Carrier Sense Multiple Access
DD:	Directed Diffusion
DPM:	Dynamic Power Management
DSR:	Dynamic Source Routing protocol
DSSS:	Direct Sequence Spread Spectrum
EAD:	Energy-Aware Data-centric routing
EAGRP:	An Energy-Aware Geographic Routing Protocol
EAODV:	Enhanced Ad-hoc On-demand Distance Vector
EAQR:	Energy Efficient ACO based QoS Routing
EAQSR:	Energy-aware QoS Routing Protocol
EAR:	Energy-Aware Routing
EAR-CSN:	Energy-Aware Routing for Cluster-Based Sensor Networks
EBAB:	Energy Balanced Ant Based routing protocol
E-D ANTS:	Energy-Delay Ant-Based Routing
EEABR:	Energy Efficient Ant-Based Routing Algorithm
EIRP:	Equivalent or Effective Isotropic Radiated Power
EM:	Electromagnetic
FA:	Forward Ant

FF:	Flooded Forward Ant Routing
FP:	Flooded Piggybacked
FS:	Forward Soldier
GAF:	Geographic Adaptive Fidelity
GBR:	Gradient Based Routing
GEAR:	Geographical and Energy-Aware Routing
GPS:	Global Positioning System
HEED:	Hybrid Energy-Efficient Distributed Clustering
Iaco:	A Bio-inspired Power Efficient Routing Scheme
IACR:	An Adaptive QoS and Energy aware routing algorithm
IAR:	Improved Adaptive Routing
ID:	Identification
IEEABR:	Improved Energy Efficient Ant-Based Routing Algorithm
IEEE:	Institute of Electrical and Electronics Engineers
JARA:	Jumping Ant Routing Algorithm
LEACH:	Low-Energy Adaptive Clustering Hierarchy
LED:	Light Emitting Diode
MAC:	Medium Access Control
MACS:	Multipath Routing Based on Ant Colony System
MANET:	Mobile Ad Hoc Networks
MCBR:	Message-initiated Constraint-Based Routing
MCBR-AST:	Message-initiated Constraint-Based Routing -Adaptive Spanning Tree
MCBR-CFR:	Message-initiated Constraint-Based Routing- Constrained Flooding Routing
MCBR-STSR:	Message-initiated Constraint-Based Routing- Real-Time Search Routing
MECN:	Minimum Energy Communication Network
MEMS:	Micro-Electromechanical Systems

MLDA:	Maximum Lifetime Data Aggregation
MLDG:	Maximum Lifetime Data Gathering
MLDR:	Maximum Lifetime Data Routing
MLER:	Maximum Lifetime Energy Routing
MO-IAR:	Many-to-one improved adaptive routing
MSRP:	Multi-Sink Swarm-Based Routing Protocol
NAM:	Network Animator
NS-2:	Network Simulator-2
PEADD:	Pheromone Based Energy Aware Directed Diffusion
PEGASIS:	Power-Efficient Gathering in Sensor Information Systems
PROWLER:	probabilistic wireless network simulator
PZSWiD:	A Probabilistic Zonal approach for Swarm-inspired Wildfire Detection using sensor networks
QDV:	Quality of Service Based Distance Vector Routing Protocol
QoS:	Quality of Service
RERR:	Route Error
RF:	Radio Frequency
RFID:	Radio Frequency Identification
RMASE:	Routing Modeling Application Simulation Environment
ROI:	Region of Interest
RR:	Rumor Routing
RREQ:	Route Request
RSSI:	Received Signal Strength Indicator
SAR:	Sequential Assignment Routing
SC:	Sensor Driven and Cost-Aware Ant Routing
SDG:	Self Organizing Data Gathering Scheme
SI:	Swarm Intelligence
SMECN:	Small Minimum Energy Communication Network
SOP:	Self-Organizing Protocol

SPEED:	A Stateless Protocol for Real-Time Communication in Sensor Networks
SPIN:	Sensor Protocol for Information via Negotiation
TBF:	Trajectory-Based Forwarding
TCO:	Termite Colony Optimization
TEEN:	Threshold Sensitive Energy Efficient Sensor Network Protocol
Termite-hill:	Termite inspired routing algorithm
TTL:	Time-To-Live
WSN:	Wireless Sensor Network
ZRP:	Zone Routing Protocol

Chapter 1

Introduction

1.1 Introduction

Wireless sensor networks (WSNs) are collections of compact-size, relatively inexpensive computational nodes that measure local environmental conditions, or other parameters and forward such information to a central point for appropriate processing. WSN nodes can sense the environment, communicate with neighboring nodes, and in many cases perform basic computations on the data being collected. The advancement in technology has produced the availability of small and low cost sensor nodes with the capability of sensing types of physical, environmental conditions, data processing, and wireless communication. The sensors collect information within their vicinity and send such information via a radio transmitter to a sink node (command center). Though, with the distinguishing features, WSN needs more effective methods of information forwarding and processing. Due to the depreciation in size and cost of sensors as a result of the advances in technologies, there exists much interest in some of the applications requiring large sets of disposable and unattended sensors. Also, researchers are then motivated in addressing the potentials of cooperation among the sensors for information gathering and processing.

The main purpose of WSNs lies in the ability to deploy large number of coins like nodes in an ad hoc manner for the purpose of achieving a common goal. WSN is used in many applications such as: radiation and nuclear-threat detection systems, weapon sensors for ships, toxins and to trace the source of the contamination in public-assembly locations, structural faults (e.g., fatigue-induced cracks) in ships, volcanic eruption, earthquake detection, aircraft, and buildings, biomedical applications,

habitat sensing, and seismic monitoring. A brief summary of its area of application is shown in Table 1.1 and classified into five categories based on their area of applications.

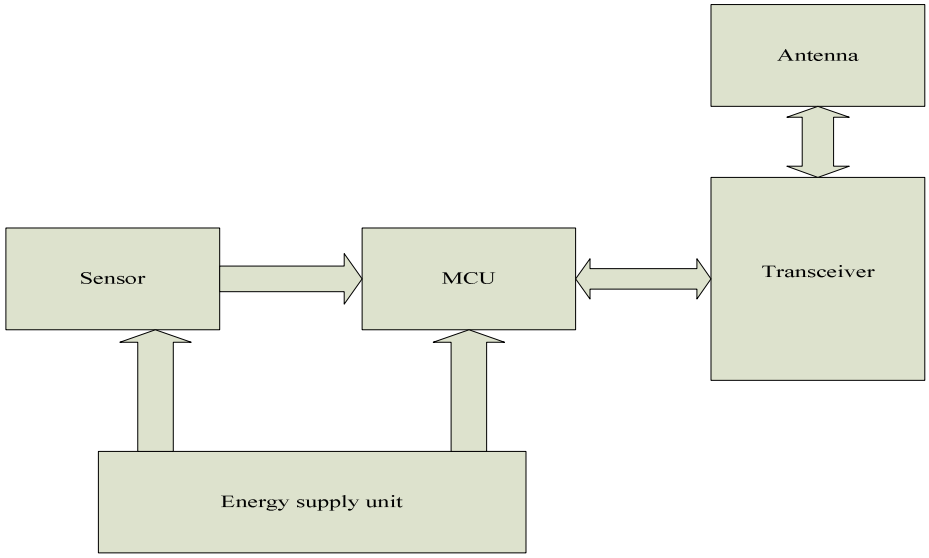
Table 1.1: Classifying applications of WSNs based on areas of application

APPLICATIONS OF WSNs	
Areas of Applications	**Applications**
Environmental	Forest fire detection, Air Pollution, Landslide and Avalanche prevention, Earthquake early detection
Home	Remote control appliances, Energy and water use, Intrusion detection systems, Automated meter reading, Arts and goods preservation
Health	Fall detection, Remote monitoring of physiological data, Medical fridges, Tracking and monitoring doctors, Patients surveillance, Drug administration, Elderly assistance
Commercial	Environmental control of industrial and office buildings, Inventory control, Vehicle tracking and detection, Traffic flow surveillance
Military	Targeting tracking, Inimical force monitoring, battlefield surveillance, Battle damage assessment, Monitoring friendly forces and equipment, Nuclear, biological, and chemical attack detection

More recently, interest has been shifted towards networking chemical as well as biological sensors for different applications ranging from industrial automation to environmental monitoring. Not like the traditional wired networks, deployment cost for WSNs is set to minima. In additional to cost reduction during the installation process, WSNs also have the capacity to adapt dynamically to changes in their environments, which basically solve the issue of network topology change rate. A wireless sensor network node mainly consists of four major parts such as:

1. Sensor unit.
2. Processing unit.
3. Energy source unit.
4. Transceiver.

These major parts are shown in Fig. 1.1. Though, depending on the area and purpose of use, additional components might be required such as localization unit, energy harvesters, position changers and monitors.



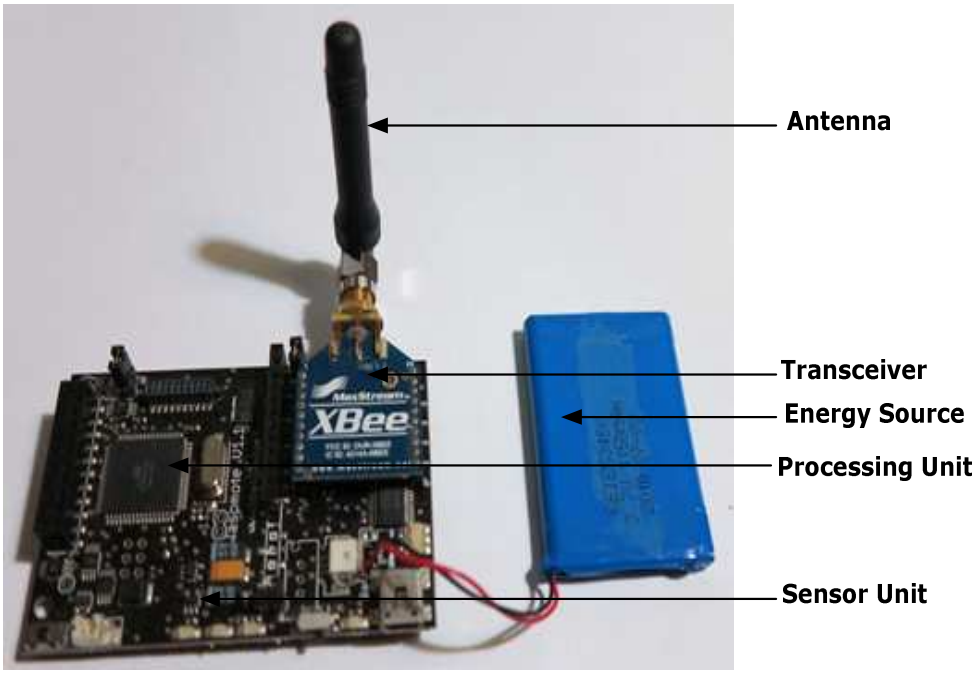


Fig. 1.1: WSN node architecture and a real example.

Ideally, a WSN should be scalable and energy-efficient, smart and configurable, reliable and long-lived, highly responsive and incur small installation and maintenance cost. However, the goals are optimistic taking into view the available limited resources. WSNs are distinguished from other types of ad-hoc networks primarily as a result of the limited capabilities of the sensor nodes. For instance, a typical processor available on a sensor node may have a clock rate of 8 – 400 MHz (Saleem et al., 2011). On-board memory of a sensor node may lie in the range of 32 – 512 KBytes, which may be further extended to few Mbytes by external means. Battery of a sensor node is one of the most critical components. In most cases, it is non-rechargeable or is infeasible to recharge them when nodes are deployed in huge number e.g. few thousands. Consequently, energy-efficient operation of every single component of a sensor node is the prime requirement in a WSN. With such limited hardware resources, accomplishing the ideal characteristics of a WSN is an extremely challenging task if not impossible (Saleem et al., 2011).

A routing protocol is the *nervous* system of any computer network. In a network where hundreds or thousands of nodes are working simultaneously, the job of a routing protocol is to identify/discover one or more path connecting a pair of nodes under a given set of constraints. The discovered paths are then used for information exchange. The routing constraints may vary. For instance, an application might require delivering information with a minimum delay. Furthermore, the type of network also plays an important role in the decision of routing constraints. Take for example a fixed network where the nodes are resource-sufficient and are wired to each other. Therefore, the prime requirement for a routing protocol is to optimize the network performance. On the other hand, ad hoc networks form a distinct category of networks. Nodes are wirelessly connected to each other and may be in constant random motion (e.g.

Mobile Ad Hoc Networks (MANETs). Now, in such a network, wireless medium and mobility poses another set of constraints. A MANET routing protocol's prime function is to maintain the connectivity between a pair of nodes in the face of a frequently changing network topology.

WSNs are a specialized class of ad hoc networks and routing in such networks has even a different set of requirements. Despite the different objectives of sensor network applications, the main function of WSN nodes is to sense and collect information (event) from a target area, process, and transmit the information via a radio transmitter back to a command center where the underlying application resides (sink) (Sohraby et at., 2007; Akkaya and Younis, 2005). In order to achieve this task efficiently, an efficient routing protocol is needed to set up paths of communication between the sensor nodes (sources), and the command center (sink). The path selection must be in such a way that the lifetime of the network as against the throughput is maximized. Due to the characteristics of the environment in which the sensor node is to operate, coupled with severe resource constraints in on-board energy, transmission power, processing capability, and storage limitations, this prompts for careful resource management and new routing protocols so as to counteract the differences and challenges.

Social insect communities have lots of useful qualities from the WSN point of view. The communities of the social insects are formed from the cooperation of the species which are simple and autonomous and are able to organize themselves. Though they lack centralize planning, but they are able to coordinate and effectively organize themselves. Due to their behavior which is adaptive and flexible, they are able to solve most of their environmental objectives. The complexity of the solutions generated by these social insects with their simple behaviors indicates that the cooperation of many will yield many more results and achieve global

objectives. With these behaviors, we can map the characteristics into sensor networks. If we consider sensor network having simple nodes working together to deliver sensed information to the end point (command center), the sensor network environment is a constituent of the network topology, communications links, and traffic patterns in the network. The difference between these social insects and the sensor networks is that social insects have an evolutionary incentive to cooperate, while the sensor networks may require alternative solutions to force nodes to cooperate (Buttyan and Hubaux, 2000). The ability of these biological species to self organizes is as a result of some behavioral principles, which includes positive feedback, negative feedback, randomness, and multiple interactions and stigmergy (Roth and Wicker, 2003). The highlighted behavioral principles, which leads to self-organization of the social insects is called swarm intelligence. Research in this field of swarm intelligence is based on working principles of ant colonies as adopted by (Bonabeau et al., 1999; Dorigo and Caro, 1998), slime mold (Li et al., 2011), Particle swarm optimization (Liu et al., 2012) and honeybees (Saleem and Farooq, 2012).

The process by which data and queries are forwarded efficiently between the source and the sink is an important aspect and basic feature of wireless sensor networks. Due to low cost sensor nodes with smaller in size as a result of technological advancement, has encouraged researchers in the past years to engage in a lot of research in addressing the potential of cooperation within sensor nodes for information processing, sensing and gathering at the destination end (sink node). A simple approach to accomplish this task is for an individual sensor node to exchange information directly with the sink node (a single-hop-based approach), or allowing intermediate nodes to participate in forwarding data packets from source nodes to the destination node and vice-versa, which is basically a multi-hop approach (Sohraby et al., 2007). Determining which set of

intermediate node is to be used for forwarding the information between the source nodes and the sink node is the principal task of the routing algorithm.

1.2 Swarm intelligence

The major contribution of this research is the use of biological behavioral patterns in solving engineering problems. Swarm intelligence is an idea of designing and analyzing large scale systems composed of many simple locally interacting individuals (Roth and Wicker, 2003; Bonabeau et al., 1999). The net result of their actions is an emergent property of the system. The emergent behaviors may be desirable or undesirable to the system designers. However they are the result of the interactions between individuals, and not the result of a pre-programmed or centrally controlled choreography. The methodology is inspired and based upon observations of the behavior of social insects such as ants, termites, social bees, or social wasps etc. These biological systems embody many of the principles that man-made systems should have. They are composed of a large number of simple and cheap components (workers), cooperating locally and independently to accomplish a global task that any individual could not do alone. Colony behavior is often robust against a large number of parameters such as individual misbehavior or loss. Such systems are also able to adapt to the environment as is necessary. Swarm intelligence (SI) is considered as a subfield to Artificial Life (AL). Artificial life is the study of designing machines with traits that mimic those of biological counterparts. Where SI is primarily concerned with studying large systems of interacting individuals, AL is broader and also looks at questions of self-replication, evolution, natural system modeling, or borrowing ideas from nature in order to enhance engineering design. Both SI and AL are generally thought to be subsumed by the even broader field of Artificial Intelligence (AI),

which is ultimately concerned with understanding "intelligent" behavior on all levels (Roth and Wicker, 2003).

1.2.1 The principles of swarm intelligence

There are four basic principles behind the swarm intelligence design framework (Roth and Wicker, 2005). They describe how systems can be created to exhibit emergent properties by taking advantage of the interactions of many individuals. These principles are basically the positive feedback, negative feedback, randomness, and multiple interactions (Roth and Wicker, 2005). Each of these principles describes a critical component of a swarm intelligent system, required to elicit emergent behavior from a large group of simple individuals. An additional principle, stigmergy, is also mentioned here due to its importance to the process, however it is not explicitly required in all SI systems. The following paragraphs introduce a qualitative explanation of each principle. Afterwards some examples will be introduced in order to clarify the role of each.

Positive feedback is used to reinforce good solutions present in the system. When a particular solution is found to be better than others, at least locally, a positive feedback approach is required so as to encourage the use of that solution over all others. Negative feedback is responsible for removing old or poor solutions from the system. Coupled with the use of positive feedback, the system tends to use only the best solutions available at any given time. It is important that positive and negative feedback be carefully balanced. If the former outweighs the latter, too many solutions will exist in the system and there will not be a clear indication as to how a problem should be solved. If the latter outweighs the former, then solutions will die out quickly and the problem cannot be solved at all. There is a need to test different solutions as they become available or change in quality. The randomness principle enables this by allowing a swarm intelligent algorithm to explore options at random. This is an effective

strategy when the number of possible solutions is large or the location and development of a good solution is unpredictable.

Swarm intelligent algorithms rely on the multiple interactions of the individuals in the system. Some amount of information that has been gathered locally may be communicated in each interaction. Consequently, information from one portion of the system may be transmitted to another. Multiple local interactions also allow for events to be coordinated on a local scale. In an ad-hoc network, each node may coordinate local routing information with neighbors during each communication. Another perspective is to insist that swarm intelligence requires large populations of participating individuals. Because individual behaviors are often randomly chosen, it is necessary to have many interactions in order to reliably determine the information necessary to decide on a proper course of action. Having many individuals will increase the interaction rate and make locally sensing system parameters more reliable. Natural SI systems, such as social insects, often contain anywhere from tens to millions of individuals.

Stigmergy is a method whereby individuals communicate with their environment of which the effect of the behavior will be used by other individuals in that environment in deciding what to do; it is the use of an outcome of previous work to guide current work. This term was introduced by Grass's in 1959 in order to describe a communication mechanism of termites engaged in nest construction (Roth and Wicker, 2003). For instance, while ants can communicate directly by feeling each other with their antennae, they also share information via effects on their shared environment. This is often done through the use of an excreted volatile chemical compound called pheromone. Different types of pheromone are used, varying concentrations of which will elicit different behaviors. Another example is with bees, who may decide to gather nectar or store

food based on statistical information on the availability of food or food storers. Stigmergy can become a useful communications channel as it offers a means to eliminate possibly expensive or complex explicit and direct communication between individuals. Moving communications to the environment can oftentimes maintain state for free and even increase the communications robustness; information transfer can be made asynchronous between individuals. The environment acts as a convenient broadcast medium for information, since all individuals exist and interact with it. There are two types of stigmergy, passive and active. The former is often associated with the physical environment which will force a particular action despite the agent's intentions. The latter is the type of stigmergy that is considered in this thesis; the environment simply influences an agent's choice of behavior which it then carries out. Emergent behavior is often the intended product of a swarm intelligent system. Individuals are intentionally simple, with the expectation that the net result of their many interactions will produce the desired behavior of the system. A prime example of this principle is the behavioral differences in biological examples such as between ants and ant colonies or bees and bee colonies. The concept of emergence has also captured the attention of other fields, including philosophy, mathematics, and music (Roth and Wicker, 2003).

1.3 Motivation

The growing number of applications for WSNs and especially their heterogeneous requirements and characteristics demand new communication protocols. The WSN community has put a lot of effort into developing reliable and fast communication services for various applications and network scenarios. However, due to physical limitations of WSN nodes as regards low supply of battery energy, communication bandwidth, processing performance, and storage become main challenges

in data dissemination and designing WSNs. Also this wireless communication technology is an advanced form of a basic concept which has existed for several decades. The basic model includes a single transceiver pair communicating over some distance. For example, the current cellular communications systems which are extensively deployed have a number of wire networked base stations, each of which services a specific area of wireless terminal (cellular telephone) users. If users should happen to move outside of the network's coverage area then all service is lost. Ad-hoc technology seeks to remove this constraint and extend the range of communications far beyond the range of a single access point. This will be accomplished by allowing nodes to act as routers in addition to their usual role as terminals. However no standard exists to facilitate the deployment of this network. One of the greatest difficulties in reaching this goal is that the wireless sensor network environment offers a number of challenging situations which network designers have not yet been able to overcome or even properly characterize. The most serious of these is dynamic nature of the network in some of the applications coupled with limited energy resource. Also due to the dynamic nature of the environment in some application, oftentimes so much control traffic is generated to determine the state of the network that it is most times unable to successfully deliver any user data.

There exists an interesting analogy to the world of biology which may shed some light on more applicable solutions to the network routing problem. Social insect colonies exhibit many of the characteristics that networks should ideally have. For instance, they are completely distributed in their operation, they are robust against interference and component malfunctions, they are adaptable, and they are composed of simple individuals. One might think of the relation between a termite and a termite colony, as well as how termite colony is able to adapt to its

surroundings in order to find food, build a hill or to protect against predators. By making simple analogies between termites and nodes, and termite colonies and sensor networks, formalism can be created in order to study how the characteristics of the former may be transferred to the latter. The work presented in this thesis develops the formalism of the biological analogy beyond previous work. The results progress engineering efforts in the field of wireless sensor networks.

1.4 Research Objectives

Within the context of the above mentioned wireless sensor network scenarios and application areas, there are numbers of problems to be addressed. The main goal of this work is to provide an energy efficient solution to support cooperation among the different application scenarios ranging from static, mobile and dynamic sensor nodes. The energy efficiency cooperation of these nodes in the application areas of the sensor network such as environmental monitoring, smart homes, health care to industrial and military application is needed. Taking into account the considered differences in relation to the different application scenarios of the sensor network, and the resource constraints of the network nodes, most especially the limited energy supply, these general aims of this research are divided into six more specific goals:

1. For the static nodes cooperation, the goal is to provide a strategy to distribute the sensing mission to the network nodes and select the most appropriate route to send the mission through until it gets to the end device (sink node) while reducing the communication among the nodes (using less control packets) in order to save energy resources;
2. For the cooperation among mobile sensor nodes, the goal is to provide a solution that helps the missions to reach their respective

collection points (sink node) and to maximize the time that they will stay in these application areas, while observing communication and energy constraints;

3. For the cooperation among static and mobile sensor nodes, the goal is to provide a strategy that will allow them to communicate and select an appropriate route linking to the mobile node, and responding appropriately to the event to be sensed, while minimizing communication to save energy and also avoiding network holes problem so as to maximize network lifetime.
4. From the above mentioned goals, we then intend to propose and develop an agent based techniques to handle the cooperation among the sensor nodes due to the fact that, agent based algorithm have been found to show adaptive, robust and scalable behaviors (Saleem et al., 2011). It is our target to have a new routing technique which will be one of its kind in the sensor network environment;
5. Also, after achieving the four goals listed above, it is also our aim to analyze our proposed techniques and compare them with other existing techniques through simulation and mathematical analysis;
6. Since a sensor node carries limited energy supply, and the fact that we will have to force them to cooperate to reach their mission of sensing, in addition to helping their neighbours to distribute the sensed information until it gets to the destination node (sink), the nodes near to the sink will run out of energy resource much faster than the nodes far away from the sink. We therefore aim at proposing energy harvesting technique to help in replenish the drain energy with the mission of seeing the possibility of the network to operate without human intervention (perpetual operation).

1.5 Problem statement

The main problem to be addressed in this work can be abstracted as the classical data dissemination problem in wireless sensor networks. The challenge does not only stop on the data dissemination, but the efficient data dissemination in an energy efficient manner. The sensor network application consists of collecting data from every sensor in the network and transferring this data to the closest collection point/node called the sink node. Taking for instance, Fig. 1.2 shows structural diagrams of the network application where the nodes are shown with a logical view to the end devise at the middle of the network. Based on the type of sensors built into each node, data are sourced whenever a sensor value changes above a threshold value. The area covered by the network is typically much wider than the radio of a single node can cover. Therefore, a multi-hop ad hoc network must be established in order to maintain network connectivity and to collect sensor information. The physical view of the network may look very different from the logical view of Fig. 1.2, and is mainly dependent on the protocols used for the network establishment and routing.

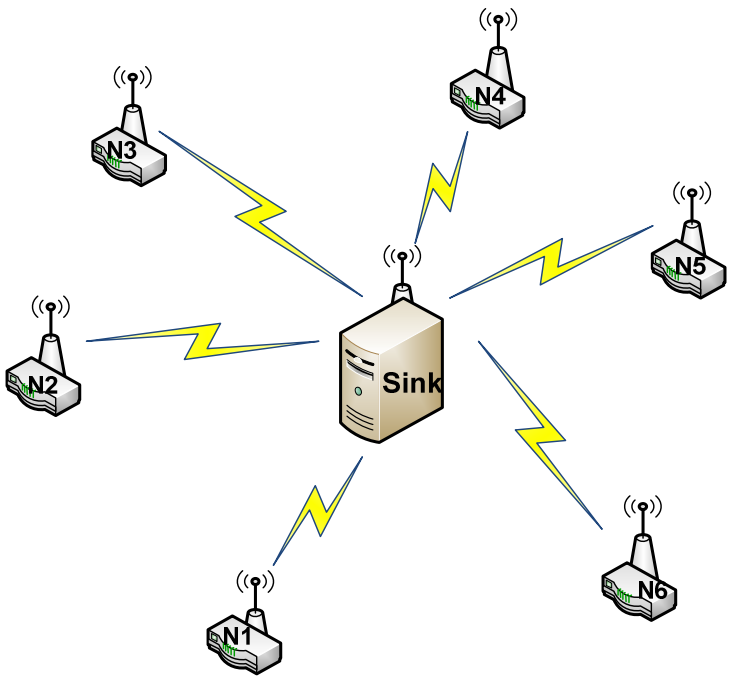


Fig. 1.2: Application view of sensor network.

Fig. 1.3 shows a sensor network with more nodes. This topology reveals an additional task of each node not seen in the logical view of the network of Fig. 1.2. To maintain network connectivity, every node must act as a router and the same time handle packet forwarding. Besides the problem of maintaining network connectivity and routing which is inherent to every network, the main challenge is the energy utilization efficiency which is set as the main goal of this work to prolong the network lifetime. Besides also the energy utilization efficiency to prolong the network lifetime, the network should be reliable in the sense that, it should deliver the packets sensed in the network adequately within the time frame the data is required at the sink node. On the other hand, identification of optimal route for data communication, efficient utilization of energy, providing congestion free communication, offering scalability, maintaining the Quality of Service (QoS) are few research issues in wireless sensor networks.

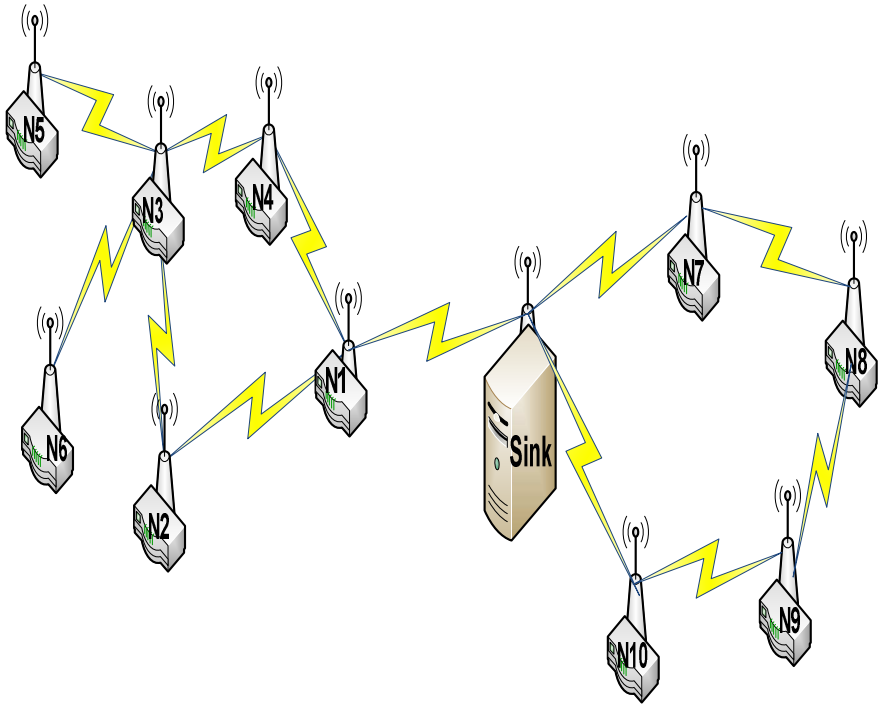


Fig. 1.3: A multi-hop physical topology sensor network.

1.6 Research methodology

In this section, we outline the research methodology that is followed in the design of energy efficient routing algorithms based on swarm intelligence for WSNs. Both simulation, experimental and mathematical analysis were conducted following the following methodology listed below:

1. In the course of our study, we first investigate the constraints and requirements of WSNs by studying and analyzing the reports of other researchers in the area of wireless sensor networks. This helps in understanding the environment in which the newly proposed routing protocols were expected to operate.
2. In the second stage of the work, we reviewed the existing routing protocols designed for WSNs from classical to swarm intelligence based, and make a comparison among them to find the best for WSNs. We further classified this algorithms based on their computational complexity, network structure, path establishment, application scenario, and energy consumption. Based on that, we proposed a model of comparison of routing protocols in WSNs.
3. Having done that, we made several improvements on the most efficient protocol between the ones reviewed (Energy efficient ant based routing algorithm, "EEABR"). The improvements were made after observing the way the algorithm was initialized, the loss of information by the algorithm, control packets generated by the algorithm, and how the algorithm handles link failure. We then proposed methods to solve the drawbacks of the algorithm, and comparison was done with other protocols using simulation approach.
4. In the fourth stage of the work, we then tested the efficiency of the improved version of the most efficient protocol (EEABR) in the presence of energy harvesting techniques in order to see the possibility of the network to operate perpetually.

5. In the fifth stage of the work, we study the foraging principles of a termite colony in-order to take inspiration from the relevant concepts for our new WSN routing protocol. During the study, we programmed artificial termites, and results obtained from the analysis help in the design of the algorithms at a later stage.
6. We then map the principles of a termite colony to solve/address a real world problem. This stage is basically the algorithmic transformation. Having done the above steps, we transform the natural concepts into the algorithmic details of our new routing protocol for WSNs.
7. We verify the performance of the termite-hill routing protocol through simulation and formal techniques to gain insights into the parameters governing its behavior in a wireless sensor network in the last stage, using routing modeling application simulation environment and Matlab software.

1.7 Major contributions of the research

Efficient routing is an important issue for the design of WSN to meet the severe hardware and resource constraints. Generally, protocol engineering is a type of multi-objective optimization problem, and is therefore, complex. We have followed rather novel approaches in the design of new algorithms for WSNs. The novelty of our approach stems from the fact that we utilize the behavioral pattern of biological agents to solve the routing problem in WSN. As part of the novelty, we also utilize a mathematical modeling not only as part of Termite-hill's evaluation process, but also use it in gaining more insight on the behavior of the sensor network environment. Few attempts for performance modeling of WSNs have been reviewed in this thesis, however, to the best of our knowledge, none of them has actually model the sensor network in terms of information extraction with limited energy resources. We also model the two well-

known topology of sensor network, that is, the grid based and line based. Also, due to the no comparative platform of routing protocols in WSN, we also proposed a model for comparison for routing in WSN during the course of this research. In this section, we have highlighted all the major contributions of our years of research on this work.

1.7.1 A model of comparison among routing protocols in WSNs

Comparing routing protocols in WSN is a very challenging task for protocol designers. However, a survey of these works conducted in this thesis shows that, often much time is required to re-create and re-simulate algorithms from descriptions in published papers to perform the comparison. Compounding the difficulty is that some simulation parameters and performance metrics may not be mentioned. With this motivation, we then see a need in the research community to have standard simulation and performance metrics for comparing different protocols. As such, we re-create and re-simulate different protocols using a Matlab based simulator, and we give the simulation results for standard simulation and performance metrics which form the basis of the model of comparison of routing protocols in WSN as a major contribution in this thesis, which we hope will be useful for the research community.

1.7.2 An improvement on an existing state-of-the-art routing protocol in WSN

Having carefully studied different routing protocols in WSN, we showed in this thesis that, despite the fact that the Energy Efficient Ant Based Routing algorithm (EEABR) (Camilo et al. 2006) is one of the-state-of-the-art energy efficient routing algorithm in WSN at the time of this study, there were still room for improvements on it so as to increase on its energy efficiency and reliability. The improvements on routing updates, limitation of network overhead, and the reduction of information loss in the system

after extensive simulation, clearly demonstrate that the improved version of EEABR termed IEEABR is simple, scalable, energy efficient and reliable as compared to some existing swarm intelligence based and classical routing protocols in WSNs. To elaborate on the adaptive nature of the IEEABR, we evaluate its performance in the presence of energy harvesters, and comparison was done with EEABR and other algorithms. Its performance in this case for sensor network utilizing the Waspmote power consumption parameters, clearly indicates that it can be used for energy management in WSN which can lead towards the perpetual operation of the network.

1.7.3 A new Termite-inspired routing algorithm for WSN

After careful study of the improved version of EEABR termed IEEABR as pointed out in sub-section 1.6.2, we find it necessary to design a new routing algorithm which counteracts some of the constraints of the IEEABR. Due to the excessive overhead generated by the EEABR as a proactive protocol, we then design a new routing algorithm termed Termite-hill which is an on-demand and reactive in path establishment. The Termite-hill routing algorithm is an agent based algorithm which takes an inspiration from the natural system of termites' behaviors due to their adaptability, robustness, and scalability. In this context, we demonstrate through extensive simulation that the proposed routing algorithm for WSN termed Termite-hill, is simple, scalable, robust, energy efficient, and reliable in information routing in WSN as compared to the existing state-of-the-art routing protocols for WSNs.

1.7.4 Formal mathematical modeling of wireless sensor network environment

The current practice in modeling and simulation of wireless sensor network (WSN) environments is to develop functional WSN systems for event gathering, and optimize the necessary performance metrics using heuristics

and intuition. The evaluation and validation of the WSN system are mostly done using simulation approaches. Simulation studies despite their wide use and merits used for network systems and algorithms validations has some drawbacks such as long simulation time, and some results reported by some authors are inconsistent. As such, we therefore argue that simulation based validation of WSN systems and environments should be further strengthened through mathematical analysis. Along with this, we were motivated to develop a formal mathematical framework for modeling and simulating WSN environments utilizing the hill building behavior of termites. The mathematical analysis covers (1) a sound model for WSN topology and information extraction including the grid based and line based topology, (2) the derivation of the expected energy consumption of our agent based routing algorithm for WSN: Termite-hill, as a function of event success rate and occasional change in topology. The results of our mathematical analysis were also compared with the simulation results.

1.8 Organization of the Thesis

The rest of the thesis is organized into six chapters as described below. The following subsections described a brief summary of each of the chapters.

1.8.1 Chapter 2: Survey and comparison of routing protocols in wireless sensor networks

This chapter forms the essential learning stage of the work. Its main purpose was to understand the steps taken in the design of the already existing routing protocols in WSN. This also helps in studying the strength and weakness of the routing protocols, which is the requirement engineering phase. To this end, we first briefly summarized some existing survey on the area of routing in WSN, we then follows it with a description of factors that are associated with routing algorithm design in WSN. Afterwards, a taxonomy of routing protocols designed for WSNs was

introduced, this helps in identifying the major features of the existing classical based as well as swarm intelligence based routing protocols surveyed in the chapter. Consequently, we then discussed each of the routing protocols based on the classifications in the taxonomy. After the discussion of each routing protocol based on their classification, we then compare the routing algorithms based on their classes analytically; this then follows by experimental comparison between them after carefully re-creating and re-simulating ten different algorithms from the classes. The results of the simulation results were also included in this thesis. Towards the end, we then proposed a model of comparison between routing protocols in WSN, which is a combination of analytical and experimental performance comparison. Finally, we conclude the chapter with recommendations based on the state-of-the-art routing algorithms proposed by different researchers. This helps us in the design of our newly algorithm. This chapter is based on the following publication:

1.8.2 Chapter 3: Improved energy efficient ant based routing algorithm

This chapter forms the core part of this thesis. We start our work with a learning and understanding of an existing routing protocol designed for wireless sensor network. We begin the work with several improvements on the existing routing algorithm termed Energy efficient ant based routing protocol (EEABR) (Camilo et al., 2006) designed using the ant colony optimization metaheuristic. The choice of the EEABR was based on the insights inferred through the critical comparison in Chapter 2. To this end, we started the chapter with a brief introduction of the energy efficient ant-based routing protocol, followed a summary of work related to our proposed algorithm. Afterwards, we briefly explained the three main application scenarios of WSNs as they affect routing algorithms. We then discussed in brief the concept of social insect that help us understand the

EEABR, and how we could improve on its energy efficiency. After the learning stage, the general concept of EEABR follows, and we then further itemized the areas of which the improvements were made. We conclude the chapter with a series of experimental results using two well-known simulation environments, and comparison was made with several state-of-the-art routing algorithms. In terms of energy efficiency, IEEABR performance was the best, but it is also comparable with the best performing Flooded Piggybacked (FP) algorithm with respect to packet success rate and throughput.

1.8.3 Chapter 4: Termite-hill: Termite inspired routing algorithm in WSNs

This chapter forms the major contribution of our research. In this thesis, we describe the natural engineering process using the principles of a natural system to solve real world problem i.e. routing in sensor network. To this end, we started by introducing the general concepts of biological species that inspired us to use their principles in solving the routing problem. We then explain how their environment can be mapped into sensor network. Furthermore, we highlighted the design guidelines we followed in achieving the Termite-hill routing algorithm. Subsequently, we reviewed work related to ours and further described the termite mode of behavior. This is important for the readers as it enables them trace the features of Termite-hill to the working of the termite colony. We start with the description of the principles of swarm intelligence as it interplays in the natural behavior of termites in their environment. Furthermore, we give a detail description of the design of our routing algorithm as derived from the termite's mode of communication. In this, we started with the description of the pheromone table, and explained pheromone update, pheromone evaporation and pheromone limits. This is followed by the description of route selection and the principle of termite behavior that inspired us to

transform them to an agent model. We then described the modules in which the algorithm is designed to function. Following the trends, we then used relevant metrics and carefully engineer the experiments to reflect the traffic patterns of a real world WSN. We compare the performance of Termite-hill as against two ant-based algorithms for WSN, and a classical routing protocol. The simulation results demonstrated that Termite-hill has a significantly higher throughput, most energy efficient, and low standard deviation in all the three WSN application scenarios.

1.8.4 Chapter 5: A Formal Mathematical Framework for Modeling and Simulation of Wireless Sensor Network Environments

This chapter describes our mathematical framework for modelling and simulation of wireless sensor network environments. We started the chapter by defining the problem that leads to the need for mathematical analysis of any system. This is followed by introducing some work related to ours, which is necessary so that readers can have a feel of other approaches and contribution of other researchers. We then describe and define some important metrics and terms that are used in analysis of the wireless sensor network environment in the next section. The description includes a detailed analysis of the network and its topology. In the subsection, we describe and analysed the energy model of a direct and multi-hop routing scheme. This is then followed by a detailed analysis of energy consumption of Termite-hill routing algorithm described in Chapter 4, with respect to topology change rate. In the next section, we describe our modelling framework for the sensor network such that maximum information is expected to be extracted from the limited energy resource wireless sensor network nodes. Afterwards, we show our computation and simulation results for (1) average transmission cost of a grid based sensor network, (2) energy consumption of a direct and multi-hop communication

schemes in WSNs, (3) energy consumption of a Termite-hill routing algorithm considering the effect of the high number of retransmission and variation of retransmission with respect to network density, packet arrival rate, and mobility. Finally, we show the comparison of our analytical results with simulation results of energy consumption of the Termite-hill. Overall, the chapter addresses the need for systematic modelling methodology by developing an analytical framework for WSN topology and information extraction in a randomly distributed grid and line based sensor network.

1.8.5 Chapter 6: Radio frequency energy harvesting and management for wireless sensor networks

This chapter describes a practical approach for Radio Frequency (RF) Energy harvesting, and management of the harvested and available energy for wireless sensor networks using the Improved Energy Efficient Ant Based Routing Algorithm (IEEABR) as presented in Chapter 3. The chapter forms the backbone for testing of the efficiency of our proposed algorithm as described in Chapter 3, regarding its efficiency as a management algorithm. The chapter begins with a general introduction of energy harvesting. We follow that with a review of energy harvesting systems and power consumption for different sensor nodes. Afterwards, we show how energy could be harvested for wireless sensor networks using Powercast harvesters, and this then followed by looking at the measurement of the RF power density, calculation of the received power, storage of the harvested power, and management of the power in wireless sensor networks. Following the trends, the practical and real-time implementations of the RF Energy using Powercast harvesters, and simulations using the energy model of a Libelium Waspmote to verify the approach were performed. Finally, the chapter concludes with performance analysis of the harvested energy, comparison of IEEABR as described in Chapter 3 and other traditional energy management techniques for WSNs.

1.8.6 Chapter 7: Conclusions and future works

This chapter outlines the main findings of this thesis. The original contributions to the field of swarm intelligence to the routing problem in WSN were also presented. We conclude that ants/termite colonies provide an ideal metaphor for developing a WSN routing protocol. The resulting protocols like their natural counterpart are adaptable, scalable and robust to loss of individual units. Towards the end, we identify some important areas that may be targeted in the future. The first and the foremost is an extension of our formal evaluation framework based on modeling of the pheromone update and evaporation of the routing protocols designed in this thesis. We believe that the framework in its existing form has tremendous potential to evolve into a de-facto formal evaluation framework which can be used in future research. We also conclude that there is a need for additional metrics and results of other protocols to be added to our proposed model of comparison in WSNs routing protocols. Last but not the least, Termite Colony Optimization (TCO) metaheuristic is also emerging as a new field in swarm intelligence that can be a focus of future research.

Chapter 2

Survey and Comparison of Routing Protocols in Wireless Sensor Networks

2.1 Introduction

High efficient routing is an important issue for the design of WSN protocols to meet the severe hardware and resource constraints. Routing in WSNs has been the focus of research in the network community due to their immense potentials for various real world applications ranging from target tracking, disaster management, smart metering, smart parking, structural health monitoring and intelligent transportation systems (Sohraby et al., 2007; Akyildiz et al., 2002). This chapter presents a comprehensive survey and comparison of routing protocols in WSNs. The first part of the chapter surveys state-of-the-art routing protocols in WSNs from classical to swarm intelligence based routing protocols. The routing protocols are categorized based on their computational complexity, network structure, energy efficiency and path establishment. The second part of the chapter presents a comparison of a representative number of classical and swarm based protocols. Comparing routing protocols in WSNs is currently a very challenging task for protocol designers. Often, much time is required to re-create and re-simulate algorithms from descriptions in published papers to perform the comparison. Compounding the difficulty is that some simulation parameters and performance metrics may not be mentioned. We see a need in the research community to have standard simulation and

performance metrics for comparing different protocols. To this end, the final part of the section re-simulates different protocols using a Matlab based simulator; Routing Modeling Application Simulator Environment (RMASE), and gives simulation results for standard simulation and performance metrics which we hope will serve as a benchmark for future comparisons for the research community.

2.2 Previous surveys of WSN routing

Routing is a very important function of the design of WSN. There have been some survey papers on routing. Akyildiz et al. (2002) surveyed protocols on wireless sensor networks, while dealing with a few of the classical routing protocols and their methods of forwarding information. Their work was based on a short period of review (1999-2000). Karaki and Kamal (2004) reviewed different routing techniques in WSNs. The authors surveyed quite a number of routing protocols, but they were limited to classical routing, and were back as old as 2004 protocols. In a similar survey, Akkaya and Younis (2005) work were still based on classical routing protocols, and not much different from that of Karaki and Kamal (2004), even as it was more recent, did not capture in their work protocols that were designed as of 2004. In the same year, Yang and Mohammed (2005) work, was also based on classical routing, and their survey was also based on selected protocols.

Recently, Singh et al. (2010) surveyed articles, only surveyed few among the classical protocols without much comparison among the few surveyed by them. Saleem et al. (2011) was the first survey on swarm intelligence based routing protocols. The authors did quite a good survey, while also considering their area of applications and simulation environments. They concentrated on swarm based protocols and left behind some of the promising protocols among the swarm based that were

present as at the time of their work. Following the trends, Çelik et al. (2010) among the most recent on swarm based protocol survey, concentrated on just few protocols in swarm based even as their work was based on swarm based protocol survey. They also did not compare the protocols based on their performance, area of application and environment of simulations. Villalba et al. (2010), also a most recent article, in their work, the comparison was based on two selected protocols, and also on classical routing protocols. Baranidharam and Shanti (2010) survey work was based on energy efficient protocols and on classical routing. Though, they call it a survey of energy efficient protocols for wireless sensor network, their work did not cover all the energy efficient classical based routing protocols. To this end, we summarize the previous work related to ours in Table 2.1. The table highlights the contribution of each author along with the year of survey and the survey characteristics.

Table 2.1: Summary of previous survey of routing protocols in WSN

S/no.	Year of survey	Authors	Survey characteristic
1	2002	Akyildiz et al.	Classical Routing
2	2004	Karaki and Kamal	Classical Routing
3	2005	Akkaya and Younis	Classical Routing
4	2005	Yang and Mohammed	Classical Routing
5	2010	Singh et al.	Classical Routing
6	2011	Saleem et al.	Swarm Intelligence Routing
7	2010	Çelik et al.	Swarm Intelligence Routing
8	2010	Villalba et al.	Classical Routing
9	2010	Baranidharam and Shanthi	Classical Routing

2.3 WSNs design and routing factors

A large number of research have been carried out to overcome the constraints of WSNs and also to solve the design and application issues. The characteristics of sensor networks and application requirements have direct impact on the network design issues in terms of network performance and capabilities (Akyildiz et al., 2002). Due to the large number of sensor nodes and the dynamics of their operating environment, these then pose unique challenges on the architectural design of sensor networks. New platforms are needed to overcome some of the challenges and cover the following issues: power consumption, fault tolerance, scalability, productive cost, quality of service, data aggregation and fusion, node mobility, connectivity, security, congestion, latency, etc. Routing design is closely related to the network system architecture mode and its design is affected by lot of factors which posse challenges during its design which needed to be addressed as outlined and discussed below.

1. Limited energy capacity: the process of setting up data dissemination paths (routes) in a network is greatly affected by energy considerations due to the fact that sensor nodes are battery powered, and having the limited energy capacity. Energy poses a great challenge in many applications of sensor networks. Since radio transmission is more affected by distance as compared to transmission in free space, it then implies that communication distance and energy consumption should be well taken care of. It will be advisable to employ multi-hop routing scheme in information dissemination as against directed routing to save energy consumption, but directed routing would be preferred if all sensor nodes are close to the sink, in the other hand, multi-hop routing consumes less power than directed routing due to the fact that, sensors are usually randomly scattered in the area of deployment, though it may introduce significant overheads for topology management and MAC

protocols. For applications on the battlefield where it is virtually impossible to access the sensors and recharge their batteries (Chong and Kumar, 2003), routing protocol design for sensor networks should be as energy efficient as possible to extend their lifetime without performance degradation.

2. Node deployment: sensor node deployments in WSNs are application dependent and they are affected by the network topology, which in turns affects the performance of the routing protocol. If nodes are randomly deployed, they need to create an infrastructure and organize themselves in an ad hoc manner to establish paths to route the events using route discovery so as to make connectivity operate in an energy efficient mode.
3. Sensor location: sensor location on the early stage of route discovery is a great challenge in the design of routing protocols. As a lot of the already proposed algorithms assumes that the sensor nodes either are equipped with global positioning system (GPS) receivers or other forms of sensing the destination or sink as in Chong and Kumar (2003) and Zhang et al. (2004), to learn about their locations, another challenge which has to be managed is the location of the sensors.
4. Dynamic network: sensor networks consist of three main components; sensor nodes, event, and sink. Since the sensor node and sink are always assumed to be fixed or mobile, though, nodes are fixed in most of the applications, thus have to support the mobility of sinks or gateways in the network. Hence, the stability of the routing data is an important design issue in addition to energy consumptions and bandwidth utilization (Singh et al., 2010).
5. Hardware resource constraints: sensor nodes also have limited storage and processing capacities, and hence, low computational capabilities. The hardware constraints present many challenges in the network and

software protocol design for sensor networks, which have to be considered alongside with the limited energy.

6. Data aggregation and gathering: data gathering or reporting is concerned with any physical event of the sensor network. This could be event driven, query driven, or automated time driven, or both combined. Data gathering methods are highly important with respect to sensor network routing, as after receiving the signal or data, the node has to transfer or route the data or information to the sink (Yang and Mohammed, 2005). Most times, duplicated information is generated by sensor nodes which could lead to energy consumption in the network. As such it is advisable for packets that are alike to be combined and gathered before transmission in order to reduce the number of transmissions, which will help in energy minimization.
7. Latency: latency or end-to-end delay in WSNs is an expression of how much time it takes for a data packet to move from one node to the sink or vice versa. This is the measure of either one-way (the time it takes for the source to send a packet to the sink), or round-trip (the one-way latency from source to sink and from sink back to the source). Data aggregation and multi-hop relays can affect latency (Zaman and Abdullah, 2010; Karaki and Kamal, 2004).
8. Scalability: since sensor applications may have many sensor nodes, and the fact that nodes in the network are usually deployed in hundreds, it then means that for an algorithm to be scalable, it must be able to handle and respond to changes in the network size and the events around it.
9. Fault tolerance: the failure of a particular sensor node as a result of unavailability of power supply, environmental interference or physical damage in a network (Sendra et al., 2011), should not in any way affect the overall network performance or task handling. In case of the

failures, routing protocols should be able to generate new routes to the data collection point or sink (Krishnamachari et al., 2002).

10. Traffic pattern structure: this is the architecture that denotes the behavioral pattern of network regarding the traffic load and its response to changes in events. Traffic pattern structure is a major factor that makes WSNs entirely different from network types such as wired networks and MANETs. The traffic patterns include: query driven, event driven, continuous monitoring and some hybrid combination of these. In a query driven, the end device (sink node) sends a request to some sets of nodes in the network regarding its interest for an information or event with the aim of getting a response with the desired information. An event-driven traffic is normally traffic due to the occurrence of events or information on some parts of the network, for example if we were interested in the opening of a door, as the door opening exceeded a certain angle which is the threshold, it will be detected by the sensor that is responsible for monitoring that particular location. In continuous monitoring event or information is sent at a regular interval to the sink node. For example the monitoring of temperature without threshold value, the sensor node will keep sending the sensed temperature as far as it has the capability to monitor. The choice on how information is gathered or sensed, have an impact on the choice of routing algorithm. For example, a proactive approach whose main aim is to gather all the information generated in the network at the sink node, is more suitable for continuous traffic, but its drawback might be excessive energy consumption for applications which are based on a query or event driven. But in the case of periodic transmission (event-driven), a reactive or an on-demand approach will be more appropriate. Though, the reactive approach has its drawback

due to high latency and as such might not be good for applications which are sensitive to timing (e.g. Security based applications).

Due to the factors mentioned above, researchers have designed and developed various routing protocols specifically for WSNs due to the differences between routing in WSNs and other wireless networks. Routing in sensor networks involves a lot of challenges due to differences in some properties between them and other wireless networks. For example, other communication systems require general/global address systems for addressing the large deployed network nodes and this is contrary to sensor network which does not use normal internet protocol addressing system. Also, it should be noted that sensor network requires sensing of sensed event from different sensor nodes to a central point (sink node), which is different from a typical communication system, and redundancy is checkmated in sensor network to improve on its energy consumption and bandwidth requirements which is quite different from other networks. Finally, sensor network has stringent requirement in terms of processing capability, on-board and transmission power, and memory available for it along with limited bandwidth, as such need careful management of its resources. Due to the differences and requirements of sensor networks highlighted above, a lot of research has been on-going by different researchers so as to solve the routing or information gathering problem in WSNs by considering the natural behaviors of sensor nodes with their application and architectural needs. We will summarize the routing protocols categories and their differences according to some metrics as shown in Table 2.2.

2.4 Taxonomy of routing protocols in WSNs

Determining which set of intermediate nodes is to be selected to form a data forwarding path between the source nodes and the sink node is the principal task of the routing algorithm. The computational complexity and the differences in the way data are forwarded from the nodes to the sink, leads to classifying the routing protocols as either classical or swarm intelligence based, and/ or data-centric, hierarchical, location based, network flow and quality of service (QoS) awareness (Akkaya and Younis, 2005). Shown in Table 2.2, is the taxonomy of the routing protocol classification in wireless sensor networks. The numbers in parentheses indicate the section numbers for easy and quick referencing. Routing protocols could also be classified based on path establishment. Using the path establishment classification, routing path can be established in one of the three ways: proactive, reactive or hybrid.

1 **Swarm intelligence routing protocols**: these are protocols that depend on the collective behavior of decentralized self-organized systems to provide a natural way of solving distributed system problems without the use of administration or coordination. The basic concepts of the protocols under these categories is self-organization which mainly involves positive feedback, negative feedback, multiple interaction and the indirect communication effect between them (Stigmergy). Most of the algorithms designed for this group utilize either the behavior of ants, bees, termites, birds or lizards and so on for problem solving in sensor networks.

2 **Classical routing protocols**: classical routing protocols are those protocols which were primarily designed for Mobile ad hoc Network (MANET), but have now been used in WSN. Though suited for WSN applications, they still have lots of challenges like scalability and

robustness. These routing schemes are independently employed by a sensor node or a sink node.

3. **Proactive routing protocols**: proactive protocols compute all the routes before they are actually needed, and the routes are stored in a table format called a routing table in each node. Each node stores information on routes to every other node in the network. The settling time for a network using this kind of algorithm is extremely high, and the number of control information/messages exchanged in order to maintain route information does grow at an alarming rate, hence, limiting the scalability of the algorithm.
4 **Reactive routing protocols**: reactive protocols compute routes only when they are needed. In this class, each node store routes only to its immediate neighbors, and determine multi-hop routes as required. In reactive protocols, routing table maintenance overhead is drastically reduced in lieu of the time required to send a message, as the path has to be determined each time a packet has to be transmitted across multiple hops to the sink.
5 **Hybrid routing protocols**: hybrid protocols use the combination of reactive and proactive strength, and use a proactive system within a given radius, while using a reactive system in the determination of routes to nodes outside the radius. The radius is always a function of some metric like the number of hops.
6 **Energy efficiency:** it is a measure of the ratio of total packet delivered at the sink node (base station) to the total energy consumed by the network's sensor nodes (Kbits/J). In most cases, sensor nodes carry low power and non-rechargeable batteries, usually of fewer ampere-hours. Therefore, there is a need for careful resource management for the sensor nodes so as to extend their lifetime, which will in turn extend the network lifetime. In the comparison section of this chapter,

we assigned very strong, strong, moderate, and weak energy utilization efficiency of respective routing protocols depending on the total packets successfully delivered at the destination with respect to the energy consumed during the experiments. Especially for the algorithms that were purposely designed for QoS tend to consume high energy as compared to the successful packet delivered at the sink node. Such algorithms are mostly weak in energy utilization efficiency, while some are moderate depending on the packets delivered in relation to the energy consumption. Algorithms that give more emphases on path selection based on energy metric (energy aware) with less packet delivery will fall under the categories of moderate, while those with average packet delivery are strong, and others with high packet delivery and same time energy-aware, are very strong energy efficient algorithms.

But in general, routing in WSNs can be divided into four main categories as Data-centric, Location-based, Hierarchical, or Network flow and QoS-aware protocols. In this section, we looked at state-of-the-art routing protocols for WSNs, from the classical routing to the swarm intelligence based. The white, light gray and dark gray colors as shown in Table 2.2 indicate another classification based on path selection as proactive, reactive and hybrid respectively.

Table 2.2: Taxonomy of routing protocol's classification in wireless sensor networks

	Classical based routing protocols	Swarm intelligence based routing protocols
Hierarchical	LEACH (2.5.3.1), SOP (2.5.3.3).	SDG (2.6.4.1), EBAB (2.6.4.2), ACO-C (2.6.4.3), ACALEACH (2.6.4.4), MACS (2.6.4.5)
	TEEN (2.5.3.4).	AntChain (2.6.4.6), PZSWiD (2.6.4.7), ACMRA (2.6.4.8), ACMT (2.6.4.9).
	PEGASIS (2.5.3.2), APTEEN (2.5.3.5), HEED (2.5.3.6), EAR-CSN (2.5.3.7), BCEE (2.5.3.8).	ACLR (2.6.4.10), MSRP (2.6.4.11), JARA (2.6.4.12), ACOLBR (2.6.4.13), ACO-RC (2.6.4.14).
Network flow & QoS aware	MLDG (2.5.4.1)	EEABR (2.6.5.1), AR & IAR (2.6.5.5), iACO (2.6.5.7), MO-IAR (2.6.5.9), Ant-aggregation (2.6.5.10), ASAR (2.6.5.11), BABR (2.6.5.12), ACO-EAMRA (2.6.5.13), EAQR (2.6.5.14), IACR (2.6.5.15).
	AODV (2.5.4.8)	E-D ANTS (2.6.5.4), Beesensor (2.6.5.6), ACO-QoSR (2.6.5.8), QDV (2.6.5.16).
	SAR (2.5.4.2), MLER (2.5.4.3), SPEED (2.5.4.4), EAQSR (2.5.4.5), MCBR (2.5.4.6)	FF (2.6.5.2), FP (2.6.5.3), ANTSENSNET (2.6.5.17).

Table 2.2 Taxonomy of routing protocol classification in WSNs (Cont'd)

	Classical based routing protocols	Swarm intelligence based routing protocols
Data-centric	SPIN (2.5.1.3).	CRP (2.6.2.2).
	F&G (5.1.1), DD (2.5.1.2), EAR (2.5.1.5), RR (2.5.1.6), CADR (2.5.1.7), COUGAR (2.5.1.9), EAD (2.5.1.10).	PEADD (2.6.2.1).
	GBR (5.1.4), ACQUIRE (2.5.1.8).	
Location based		
	GEAR (2.5.2.2), TBF (2.5.2.5), EAGRP (2.5.2.6).	
	GAF (2.5.2.1), MECN (2.5.2.3), SMECN (2.5.2.4).	SC (2.6.3.1).

2.5 Classical based routing protocols

Classical routing protocols are those protocols which were primarily designed for Mobile ad hoc Network (MANET), but have now been used in WSN. Like other packet switch networks, most of the research work in the area has also been done by the networking community. The protocols were designed based on different design objectives. However, they can also be applied to the routing problem in WSN. Though, the drawback of the algorithms in terms of their applications to WSNs, are challenges such as scalability and robustness. These methods of routing are used by a sensor node or a base station independently. The following subsections described each of the classes of classical routing protocols, and comparison among them was made and presented in different tables.

2.5.1 Classical based data-centric routing protocols

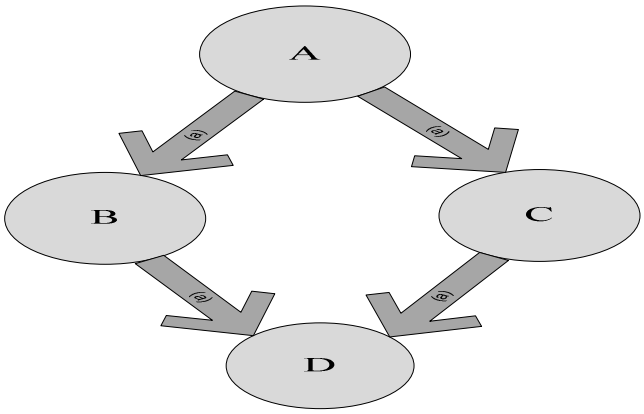
Broadcast and unicast are two operations that sensor nodes use to communicate with each other. In data centric routing, the sink node forwards an interest message to certain parts of the network and waits for information about the query from the sensors located in that area. Data centric utilizes data aggregation in relaying of data, when an event or information sensed by any of the source node arrived at neighboring node, the neighboring node which serves as router now will have to consider the importance of the event received so as to decide either to forward it to another neighbor or not. SPIN (Heinzelman et al., 1999; Kulik et al., 2002), which happen to be the first data-centric protocol, utilizes negotiation between nodes in the sensor networks so as to eliminate information that are redundant, and as such save energy. We described each of the algorithms in the following sub-sections.

2.5.1.1 Flooding and Gossiping (F&G)

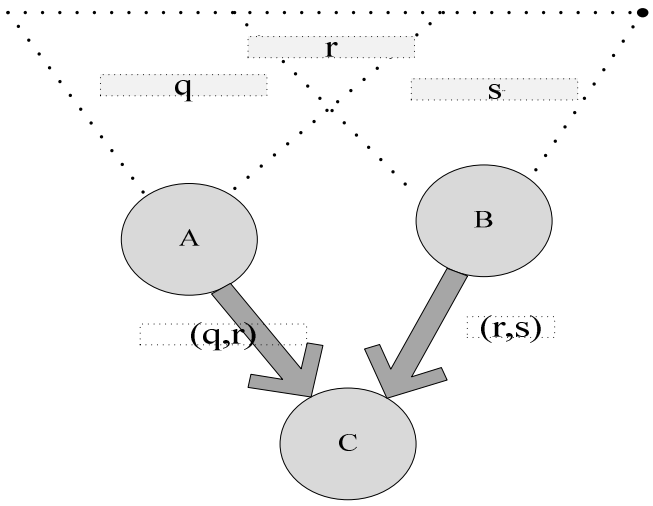
These are protocols which do not utilize routing tables and topology maintenance of data transmission in sensor networks (Heinzelman et al., 1999; Kulik et al., 2002; Hedetniemi et al., 1988). In flooding, sensor nodes flood or broadcast data to its entire neighbor any time it receives data from other neighbors. Gossiping is an improved version of flooding in which the receiving nodes broadcast data to only selected neighbors. Flooding finds its drawback in data overlapping when two or more nodes sensing the same geographical area send almost the same data to neighbor common to them. Implosion problem is mainly caused as a result of sending duplicated information to the same node and increase the amount of energy consumption without preference to the energy constraint of sensor nodes as illustrated in Fig. 2.1 (a) and (b). In gossiping, the problem of implosion incurred in flooding is avoided by randomly selecting

nodes to send the required information instead of the broadcasting scheme. however, the method of random selection might cause delay in the information dissemination through the nodes.

Fig. 2.1 (a) shows implosion which is the major problem incurred by flooding scheme, and node at point A floods its information to all the nodes within its vicinity (neighbors), and accidentally, the node in location D eventually gets two same copies of the information which is not necessary. While in (b), the overlap problem, in this case, two nodes are bounded by an overlapping geographic region, whereby the node at point C gets the same copy of information from these two nodes.



(a)



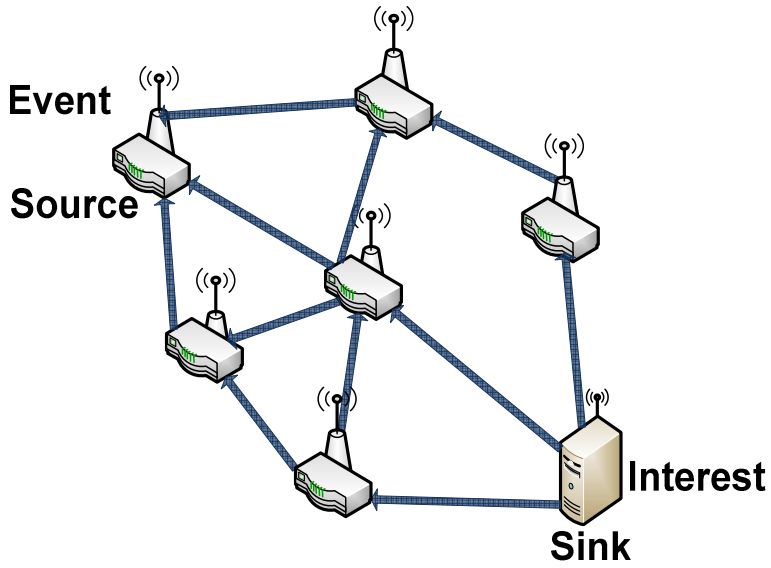
(b)

Fig. 2.1: (a) Implosion problem and (b) The overlap problem.

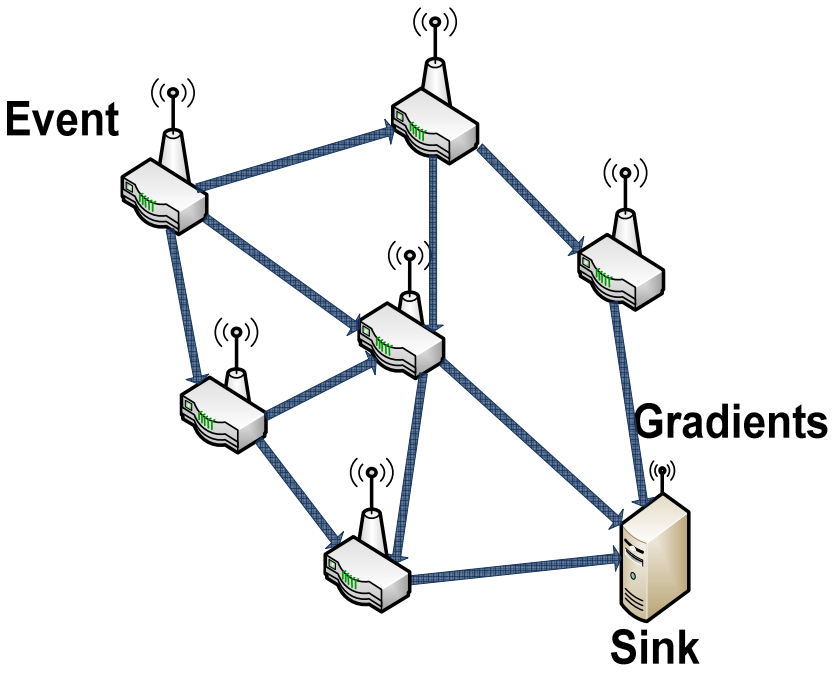
2.5.1.2 Directed Diffusion (DD)

In DD (Intanagonwiwat et al., 2000, 2003), events are diffused towards the destination node with the help of sensor nodes by means of the use of naming scheme for the event. Attribute value pairs for the event is adopted while querying the sensors in an on demand basis. It is a popular information aggregation scheme for WSNs. It is a data-centric and application aware scheme in which event sensed by each source node is named by attribute-value pairs. Creation of query is achieved by defining an interest using any of the attribute value pairs such as name of event, duration of the event, and geographical location etc. The algorithm makes use of three basic steps to achieve its aims in terms of event transfer towards the sink node, which involves periodically flooding of the query message throughout the network by the sink node as shown in Fig. 2.2 (a); in this case, the query is broadcast towards a specific direction. Whenever an intermediate node receives this query information, it immediately transfers the message received to other neighbors, but before that, it create and save the information about the previous hop gotten from the query message as their routing information called gradients as shown in Fig. 2.2 (b). The gradient created will then be used as the reference state to send the sensed event to the sink node. The gradient direction is set to the neighboring node from which the query is received. Through this process, queries are propagated towards the source of the event so as to start sending the sensed event to the sink node as shown in Fig. 2.2 (c). When a sensor node sensed an event, it looks in its query cache for a possible matching entry. If there is an existence of a match, it computes to get the highest among the requested information among all its outgoing gradients. When this process is completed, the intermediate node now becomes the source of the event and as such, it checks all neighbors to which it has gradients to forward the sensed event, and the process

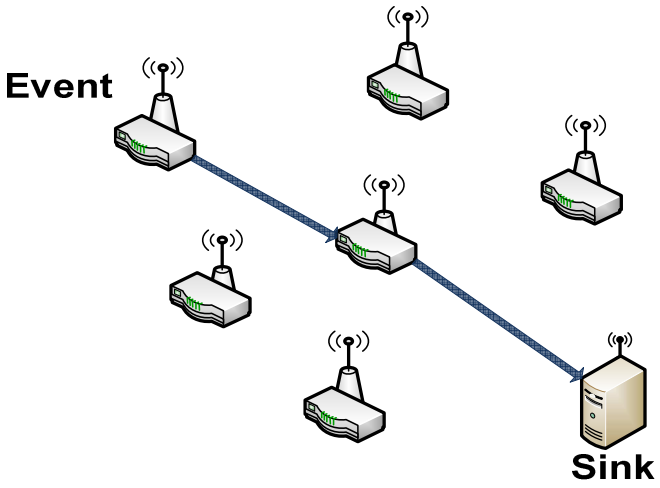
continues till the information gets to the sink node. Directed Diffusion differs from SPIN due to its on demand data querying scheme. DD cannot be applied to some applications of WSNs due to its query-driven data delivery model, since those requiring a continuous flow of information to the destination node will not perform efficiently.



(a)



(b)



(c)

Fig. 2.2: Directed Diffusion routing scheme (a) Propagate interest (b) Setup initial gradients (c) Send data.

2.5.1.3 Sensor Protocol for Information via Negotiation (SPIN)

SPIN (Heinzelman et al., 1999; Kulik et al., 2002) is the first data-centric routing protocol. In SPIN, data are named using meta-data. The protocol meta-data negotiation helps in elimination of overlapping, redundant information and resource blindness as drawbacks of flooding and as such saves a lot of energy. In SPIN, three messages are defined to aid in data dissemination: ADV message for advertisement of data, REQ message for data requests, and DATA message that carries the actual information. SPIN applications are resource-aware and resource-adaptive. Fig. 2.3 demonstrates the exchange of information in SPIN protocol. The advertisement method of SPIN does not guarantee the delivery of data since nodes that might be interested in the information may be far off from the source nodes, and intermediate nodes may not be interested. In regard to that, the information may not get to the sink, hence, it is not a good protocol for security measures e.g. intrusion detection and car tracking.

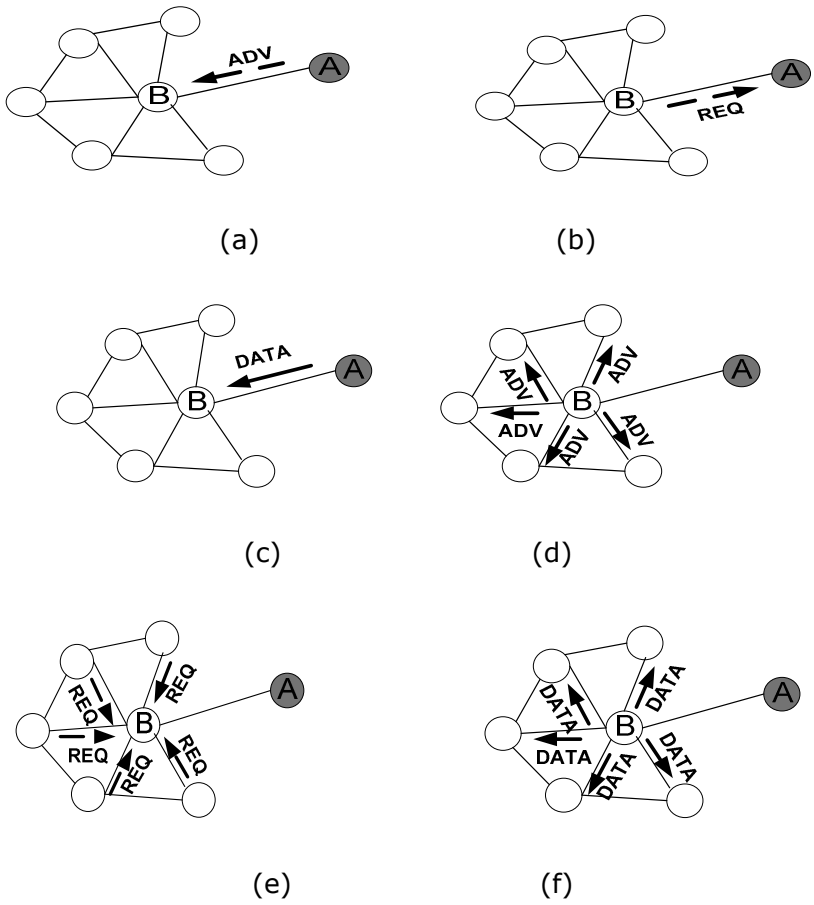


Fig. 2.3: SPIN protocol phases (a) Advertisement phase (b) Response phase (c) Acknowledgement phase (d) Reception and advertisement. (e) Request phase (f) Sending phase.

2.5.1.4 Gradient Based Routing (GBR)

This is an improved version of Directed Diffusion. Its target is to maintain the number of hops when there is a query sent out by the sink node in a request for information about an event in the network (Akkaya and Younis, 2005). This then implies that each node participating in the sensing can discover less hops count to the destination node, which represents the node's height. The drift in height between any node and its neighbor is termed gradient on the link. In GBR, data are aggregated and traffic spread over the entire network so as to distribute the traffic equally and helps in balancing the traffic in the network. The spreading technique of this scheme assists in load balancing among the nodes, as such increases

network lifetime. The protocol outperforms DD in the metrics used for simulation analysis.

2.5.1.5 Energy-Aware Routing (EAR)

The EAR protocol (Shah and Rabaey, 2002) considers the utilization of less optimal routes occasionally so as to prolong the lifetime of sensor networks. The point of the argument of the authors is the fact that, frequent use of minimum energy route depletes the energy of nodes on that route. Hence, it encourages the use of multi-paths with a certain probability. The approach assumes the possibility of addressing each sensor node using class-based addressing scheme that involves the use of the node position and its type. The scheme has three phases (Shah and Rabaey, 2002): (1) Setup phase which is calculated using:

$$\mathrm{C_{Nj,Ni}} = \mathrm{Cost(N_i)} + \mathrm{Metric(N_j, N_i)} \tag{2.1}$$

In an information request sent from node $\mathrm{N_i}$ to node $\mathrm{N_j}$. Energy of nodes is used for decision making on the path of propagation. Paths having a high cost are discarded. For this phase, the probability of selection in each routing table of a node for its neighbors is calculated as:

$$P_{Nj,Ni} = \frac{{}^{1}/_{C_{Nj,Ni}}}{\sum_{k \in FT_j} {}^{1}/_{C_{Nj,Ni}}}; \tag{2.2}$$

And the average cost of arriving at the sink node through the neighbors is computed as:

$$\mathrm{Cost(N_j)} = \sum_{i \in FT_j} P_{Nj,Ni}\, \mathrm{C_{Nj,Ni}}. \tag{2.3}$$

(2) Data communication phase, and (3) route maintenance phase. EAR has similarity with Directed Diffusion as possible available paths from sensor nodes to sink nodes are discovered, but as compared to DD in their experiment, there is an improvement in performance of 21.5% energy savings of nodes, and 44% increase in network lifetime.

2.5.1.6 Rumor Routing (RR)

RR (Braginsky and Estrin, 2002; Patra et al., 2010) is another improved version of Directed Diffusion. The main aim of the algorithm is to fill the gap between query flooding and event flooding. That is to say that, it floods information messages whenever the number of messages is small as compared to the query, and hence deals with event flooding and query flooding. It does that so as to send queries to the nodes that have sensed needed information instead of flooding the entire network for it to access particular information about an event. The protocol ensures that only one path is utilized between source and sink as against DD, which utilizes multiple paths at low rates for information dissemination. It performs best when the number of data is small.

2.5.1.7 Constrained Anisotropic Diffusion Routing (CADR)

This is another version of Directed Diffusion (Chu et al., 2002). The techniques' main goal is to query sensors nodes so as to route events sensed by them throughout network to the destination node for maximizing information gain, in addition to minimizing bandwidth and latency of the network. The protocol implores the activation of sensors closer to an interested event and adjusts the data routes to the sink node. It differs from DD in its consideration of energy consumption and information success rate in the network. IDSQ is a complementary optimization procedure for CADR. The protocol is more energy efficient than DD, since in DD, interest about an event is diffused isotropically, arriving at the closest neighbor before the others.

2.5.1.8 ACtive Query forwarding In sensor nEtworks (ACQUIRE)

ACQUIRE, is a novel mechanism for data extraction in an energy constraint sensor network (Sadagopan et al., 2003). It was proposed so as to counteract the effect of complex queries for information about an event

where response for the query could be provided by several sensor nodes. The protocol is efficient in terms of the interest propagation of querying and modifying the value of d parameter, where d is the look-ahead parameter meaning each sensor can request information from sensors d hops away from it. In a case where d tends to the network size, the algorithm behaves as flooding, else, a larger number of hops are required if *d* is small (Sadagopan et al., 2003). The protocol decides on the next node to forward query based on random selection or maximum potential of query satisfaction. ACQUIRE is similar to Rumor and CADR in terms of query forwarding.

2.5.1.9 COUGAR

The COUGAR approach to In-Network Query Processing in Sensor Networks (Yao and Gehrke, 2002) is a data-centric protocol which tasks sensor networks through declarative queries. A query optimizer generates an efficient query plan for in-network query processing so as to save energy. The protocol utilizes sensor data base system whereby each sensor randomly chooses a leading node for aggregation function and send the information to the destination (sink) node. This function is as a result of responding to the interest message layout by the sink node for an event. The query layout also gives the procedure in the selection of a query leader as shown in Fig. 2.4. The algorithm gives a freedom of independent querying of each sensor node to the network layer. Its main drawback is the extra query layer on each sensor node which in turn brings extra overhead to the sensor nodes, and as such increases energy consumption and storage.

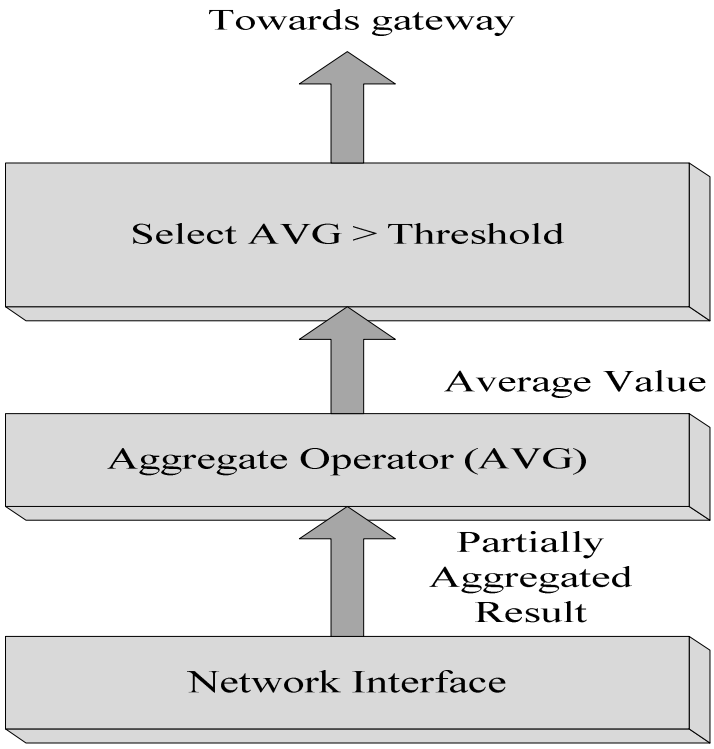


Fig. 2.4: plan at a leader node.

2.5.1.10 Energy-Aware Data-centric routing (EAD)

EAD (Singh et al., 2010; Boukerche et al., 2003) is a distributed routing protocol with the aim of building a virtual backbone consisting of active sensors for in-network data processing and traffic relaying. It tends to build a broadcast tree that approximates an optimal spanning tree with a minimum amount of leaves, hence reducing the size of the backbone formed by the active sensors. The protocol introduces two concepts; neighboring broadcast scheduling and distributed competition among neighbors. While its concept fits quite well in the data-centric network, as the non-leaf nodes are also responsible for data aggregation, it does not address mobility in the sensor network. The problem is that, data-transmit phase is substantially longer than the initialization. In a highly mobile environment, this means in practice that before a round is completed, the tree may already be dissolved.

2.5.2 Location-based protocols

In routing, some of the algorithms designed for WSNs usually require knowledge of locations for the nodes in the network. In that case, the nodes are mainly identifiable by means of their locations. The information of their respective location is needed so as to aid in the calculation of distance between two nodes, and be able to diffuse a query to a particular region, hence a reduction of the number of transmissions (Roychowdhury and Patra, 2010). This in turn helps in the estimation of the energy consumption. We describe each of the algorithms in the following sub-sections.

2.5.2.1 Geographic Adaptive Fidelity (GAF)

GAF (Xu et al., 2001) was mainly designed for mobile ad-hoc networks, and also applicable to WSNs. The protocol ensures low consumption of network energy by disabling nodes that are not active in the routing process without necessarily affecting the routing performance. Each node has a GPS as an attachment to its sensor, indicating its location so as to associate itself with a point on the virtual grid (Akkaya and Younis, 2005). In GAF, each node adjusts its status from sleeping mode to active mode in order to balance its load. The three states that exist in GAF are discovery, active and sleeping states. GAF differs from that of hierarchical protocol since the leader does not do any aggregation or fusion.

2.5.2.2 Geographical and Energy-Aware Routing (GEAR)

GEAR (Yu et al., 2001) is an algorithm for routing queries to target regions in a sensor network in an energy efficient manner. In GEAR, each sensor node is equipped with a GPS sensor for location identification. The protocol utilizes energy aware heuristics which are based on geographic information for the selection of nodes to route data to the sink, and uses a geographically recursive forwarding algorithm for data dissemination within

the target area. The main target of the scheme is to minimize the requested information in DD by considering parts of the network of interest request instead of sending the interest to all nodes in the network. This method helps GEAR in the minimization of energy consumption in the network as against that of DD. There are two phases in the algorithm: (1) Information forwarding towards the target region and (2) Information forwarding within the region in the network. GEAR does not only reduce energy utilization, but also outperforms GPSR (Karp and Kung, 2000) in terms of packet success rate.

2.5.2.3 Minimum Energy Communication Network (MECN)

MECN (Li and Halpern, 2001) is a protocol designed for achieving minimum energy in a randomly deployed ad hoc network using a GPS. The protocol is most suitable for application of sensor network whereby the mobility sensor nodes or sink node is not involved. In order to efficiently utilize the protocol, a minimum power topology involving the use of a master node is required. Its main idea is to source for a sub-network, that will have fewer number of hops to the end device with minimum energy requirements for information transfer between nodes in the network. This helps in achieving paths with minimum energy without looking through the entire nodes in the network. The scheme has the ability to reconfigure itself should there be a failure of nodes in the network or increase in the size of networks due to deployment of new sensor nodes. It is also based on the position of sensors on the plane and consists of two main phases: (1) the enclosure graph construction and (2) cost distribution. MECN assumes that at every point in time, a node in the network can transmit to any node in the same network, which is not always possible.

2.5.2.4 Small Minimum Energy Communication Network (SMECN)

SMECN (Li and Halpern, 2001) is an improved version of MECN. SMECN assumes that in the network of distributed nodes, it is possible to consider any obstacle between any pair of the nodes as against MECN which assumes that a node in a network can transmit to any node in the same network, which is practically not realistic. The network of SMECN is a fully connected network as that of MECN. The sub-network produced by SMECN for energy efficient transmission is lower in terms of number of edges. The results presented in their work show that SMECN has lower energy consumption and lesser links maintenance cost as compared to MECN. Though, with the introduction of its sub-network which gave it smaller edges increases its network overhead.

2.5.2.5 Trajectory-Based Forwarding (TBF)

TBF (Nat and Niculescu, 2003) is a routing algorithm whose main idea is the use of a dense network in the presence of a coordinate system for the sensor nodes to be able to position themselves so as to estimate distance to their neighbors. The source of an event specifies the trajectory in a packet, but does not give the path on a hop-by-hop basis (Singh et al., 2010). Based on the location information of the neighbors, a sensor node forwarding data make a greedy decision to determine the closest trajectory fixed by the source node. In TBF, multipath is also possible, and route maintenance is unaffected by sensor mobility.

2.5.2.6 An Energy-Aware Geographic Routing Protocol (EAGRP)

EAGRP (Elrahim et al., 2010) is an energy aware forwarding protocol for multi-hop WSNs. The algorithm deals with two parameters, location and energy level of each node. Each node knows the location and energy level of its neighbor in order to forward packets. The approach basically involves

calculating the average distance of all the neighbors of transmitting node and consideration of their energy level as a determining factor for selecting the neighbor who is alive, and selection based on the energy level. That is, the neighbor whose remaining energy is above a predefined value and of distance less than other nodes among its neighbors. The simulation result as compared to DSR and AODV, perform better in terms of the packet delivery ratio, delay, energy consumption and throughput.

2.5.3 Hierarchical protocols

A hierarchical protocol is an approach to the balance between scalability and performance. In hierarchical routing, energy consumption of sensor nodes is drastically minimized when the sensor nodes are involved in multi-hop communication in an area of clusters, and engaged in event aggregation and fusion so as to reduce the rate of information transmission to the sink node. The clustering of information is dependent on the remaining energy of sensor node and its distance to the cluster head (Lin and Gerla, 1997). In hierarchical routing, data moves from a lower clustered layer to a higher region, hopping from one node to another which covers larger distances, hence moving the data faster to the sink. Clustering provides the inherent optimization capability at the cluster heads. A view of the architecture of hierarchical network, is as in Fig. 2.5.

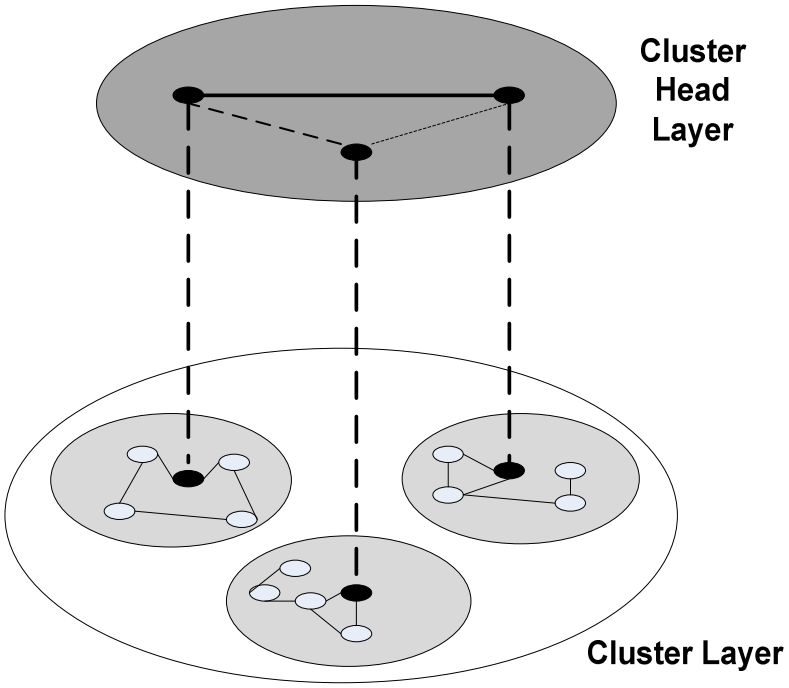


Fig. 2.5: Hierarchical network architecture.

2.5.3.1 Low-Energy Adaptive Clustering Hierarchy (LEACH)

LEACH (Heinzelman et al., 2000, 2002) became the most popular and the first energy-efficient hierarchical algorithm proposed for power consumption reduction in sensor networks. LEACH rotates the clustering task among the participating nodes based on duration. Each cluster head communicates directly to the sink. The algorithm is also based on data aggregation or fusion techniques as the original data is combined and aggregated into a smaller size of data that carry only required information for all individual nodes. The leader of a cluster (head) varies over time so as to help in energy balance within the network nodes. The protocol is balanced and fully distributed scheme which requires no prior knowledge of the network. As it adapts formation of cluster heads, or dynamic clustering, it brings extra overheads, hence diminishing the gain in energy saving. It is also not friendly in a large network deployment.

2.5.3.2 Power-Efficient Gathering in Sensor Information Systems (PEGASIS)

PEGASIS (Lindsey and Raghavendra, 2002) is an improved version of LEACH. It avoids the formation of multiple clusters. Each node can transmit and receive data from its neighbor, but just a single node is made to communicate with the destination node at a time from a chain, and information is combined to form a single message (aggregated) before sending to the sink node. Unlike LEACH, it avoids the formation of clusters, but information transfer to the sink node is through the use of a node in a chain. This scheme avoids the use of multiple nodes, and in turn helps in saving more energy. However, the protocol encounters much delay (high latency) in transmitting information from nodes in a chain that is much farther away from the sink node. In addition, the single leader exhausts its energy as it involves regular transmission. In (Lindsey et al., 2001) comes an improved version of the PEGASIS, known as Hierarchical-PEGASIS. Its aim is to find a solution to the delay incurred during the transmission of packets to the sink, and as such proposed an alternate way of counteracting the effect of information gathering by adopting the energy x delay metrics. It involves code division multiple access (CDMA) in its approach to deal with the problem of signal interference among the sensor nodes, and also allow nodes that are not fully connected to transmit information at the same time.

2.5.3.3 Self-Organizing Protocol (SOP)

The SOP (Subramanian and Katz, 2000) involves basically the self organization of the sensor nodes in addition to the creation of routing tables which is based on four phases as: (1) discovery phase, where node in the neighborhood are discovered and further updated in the routing table; (2) organizing phase, where set of nodes are formed to form a group and further combined by forming a hierarchy of which nodes are addressed

with regards to their position in the hierarchy; (3) self-reorganizing phase, dealing with the situation when partition of nodes fails, and group reorganization takes place; and (4) maintenance phase, in this, the routing tables are updated and the energy levels of nodes is considered before route selection. SOP adopts a local Markov loop scheme which helps in dealing with fault tolerance, and use for broadcasting. The algorithm is cost effective in routing table maintenance, and consumes less energy in broadcasting messages than SPIN scheme which is as a result of broadcast trees used in the scheme. Though, the algorithm introduces extra overhead due to the organizational phase of the algorithm as it is not a reactive path selection scheme.

2.5.3.4 Threshold Sensitive Energy Efficient Sensor Network Protocol (TEEN)

TEEN (Manjeshwar and Agrawal, 2002) is a hierarchical protocol whose main aim is to respond to any change in value of a sensed event with reference to the threshold value set for the event. The protocol combines the hierarchical technique in line with a data-centric approach. It involves the formation of clusters along with cluster leaders which broadcast two pre-set values of the nodes, and are basically the hard and soft thresholds. The first has the minimum values of the sensed event of a sensor node to trigger it for powering its transmitter so as to send the event or information to the cluster head, while the second has the maximum values. The scheme is normally not suited in applications where continuous data is needed, since it is threshold dependant.

2.5.3.5 Adaptive Threshold Sensitive Energy Efficient Sensor Network Protocol (APTEEN)

APTEEN (Manjeshwar and Agrawal, 2002) is an improved version of TEEN, whose main function is not limited to the formation of clusters, but also aimed at capturing recurrent event and proper reacting to time dependant

information or data. In APTEEN, cluster leaders perform aggregation as well as conserve energy. Three queries are supported in the protocol; historical for analysis of past information values, persistent for monitoring of events for some time duration, and one-time for a snapshot view of the sensor network. Simulation results show that it outperforms LEACH, having the problem of overheads and complexity in cluster formation in multiple levels, and implementation of the threshold based functions.

2.5.3.6 Hybrid Energy-Efficient Distributed Clustering (HEED)

HEED (Younis and Fahmy, 2004) is an extension of LEACH which uses node density and residual energy as a metric for cluster selection so as to balance the network energy. The HEED execution process takes three phases; 1. The initialization phase, where cluster heads are selected based on their residual energy and intra-cluster communication cost. 2. Repetition phase, where the probability of selection of the cluster head is repeated due to some parameters if at the first stage it was not selected. 3. Finalization phase, where the selection of cluster head is finalized. However, the cluster selections consider some parameters, which may impose some drawback on the network, though it is suitable for prolonging the network lifetime.

2.5.3.7 Energy-Aware Routing for Cluster-Based Sensor Networks (EAR-CSN)

The algorithm (Younis et al., 2002) was proposed based on three tier architecture, and grouping of sensor nodes known as clusters. Cluster heads formation is less energy constraint and the head tends to know the other nodes location, and the maintenance of the status of the sensor nodes while also setting up a multiple path selection for event collection is the main achievement of the scheme. It uses time division multiple access based medium access control in communicating with the gateway or cluster

heads. In the algorithm, nodes within the cluster could form part of any of the source of event, intermediate nodes in charge of event forwarding, or the combination of the two functions above, or none of them. There is a cost which is a function of delay optimization, energy utilization, throughput, and success rate, and other performance parameters. The algorithm suffers in terms of transmission range, and since the algorithm uses many cluster heads, it introduces more overheads and hence consumes much energy.

2.5.3.8 A Balanced-Clustering Energy-Efficient Hierarchical Routing Protocol (BCEE)

BCEE (Cui and Liu, 2009) aims at equalizing the consumption of energy in sensor network nodes and to prolong the network lifetime by adopting mixed hierarchical and ant based routing method. The protocol operates in two phases similar to other hierarchical protocols where routing is done in two phases. In phase one, cluster heads are selected adopting K-means algorithm where the actual location of nodes is unknown, but uses the idea of received signal strength indicator (RSSI) for cluster formation. And in the second phase called the steady state, it utilizes the techniques of ant colony optimization to establish an optimal multi-hop route from cluster heads to the sink using rational and hop-selecting technique. The protocol was compared with LEACH in simulation and performs better in terms of energy consumption of nodes. BCEE has some drawbacks in terms of network overhead as cluster formation also requires more energy in the network, and delay of data transmission to the sink node.

2.5.4 Network flow and QoS-aware protocols

Some of the algorithms which do not belong to data-centric, hierarchical or location based tends to fit into network flow and QoS-aware approach. In some protocols, routing setup is treated as a factor of network flow, while

in QoS-aware protocols, latency is the major metric considered when setting up paths or routing in the sensor network. We describe each of the algorithms in the following sub-sections.

2.5.4.1 Maximum Lifetime Data Gathering (MLDG)

Maximum Lifetime Data Aggregation (MLDA) (Kalpakis et al., 2002) is a polynomial algorithm in which the system lifetime is based on the number of recurrent information readings from the sensor till the initial sensor node ceased to exist. The event routing technique represents or tells the period for which and how information can be obtained and routed to the sink node. As the network is a function of the period of time for which the scheme remains active, it is therefore of importance to the algorithm to maximize its network lifetime. The protocol main consideration is the network lifetime path, alongside with event aggregation. Though, where there exists some special cases where data aggregation is not achievable as in streams from video sensors, Maximum Lifetime Data Routing (MLDR) comes into play and is a function of network flow problem which considers sensors in terms of the energy constraints. Both MLDA and MLDR as compared to Hierarchical-PEGASIS, outperforms the protocol in terms of the network lifetime. Hierarchical-PEGASIS is better than MLDA in terms of delay for data packets. Cluster-based MLDA (CMLDA) (Dasgupta et al., 2003) which is an improvement on MLDA and MLDR was developed so as to solve the problem of large networks and hence reduce the delay problem of MLDA.

2.5.4.2 Sequential Assignment Routing (SAR)

SAR (Sohrabi et al., 2000) is the first routing algorithm that involves the principle of QoS for path selection in its routing decision, and also a table-driven multipath algorithm whose aim is to achieve network fault tolerance while saving as much energy for the network. In SAR, nodes and link

failures avoidance and maintenance are achieved by using proactive schemes of routing table update. The scheme covers both downstream and upstream nodes on each link. The algorithm involves the creation of rooted trees at one-hop neighbors of the destination node by considering the necessary metrics, through the creation of branches, it builds multi-hop route from the destination node to the source nodes. It encounters a difficulty in the maintenance of the table created due to proactive scheme of updating, and also the overhead created during the process.

2.5.4.3 Maximum Lifetime Energy Routing (MLER)

MLER (Chang and Tassiulas, 2000) is proposed to assist in solving the constraints of WSNs. The protocol is based on a network flow. Its idea is to prolong the network lifetime with the aid of defining path cost as a function of the residual energy of a node, in addition to using the link while requiring energy to transfer information within the network. By maximizing the lifetime of the network, the protocol leads to establishing load distribution, which will help in the reduction of energy consumption in some parts of the network and prolonging network lifetime. When compared to the MTE algorithm, it outperforms MTE only in terms of energy consumption value during information transmission as the link cost. The performance is due to the relative remaining energy which is due to the reflection of the foreseen energy consumption of the nodes used by the algorithm.

2.5.4.4 A Stateless Protocol for Real-Time Communication in Sensor Networks (SPEED)

SPEED (He et al., 2003) is a QoS routing protocol for WSNs. The scheme involves three types of communication techniques: real-time unicast, real-time area-multicast and real-time area-anycast. In the protocol, each node in the network keeps information about its neighbors in their routing tables

in addition to the use of geographic forwarding scheme in order to locate the route to the destination node. The protocol is aimed at minimizing overhead due to control messages. The protocol provides end-to-end real-time communication scheme, and through the use of non-deterministic geographic forwarding and feedback control, it maintained a desired speed of communication across the sensor network. It is an efficient and scalable protocol designed mainly for WSNs for which the nodes are having scarce resources. As compared to DSR (Johnson et al., 1996) and AODV (Perkins and Royer, 1999), the protocol attained lower latency and lesser consumption of energy of nodes.

2.5.4.5 Energy-aware QoS Routing Protocol (EAQSR)

Energy aware QoS routing (Akkaya and Younis, 2003, 2005) is a protocol that uses tables to classify the different kinds of entities (table driven) and as well as uses multiple path in the routing process with embedded QoS in its routing decision. Its aim is to find an optimal route to the sink node in terms of energy consumption, and avoidance of error path while meeting the low latency requirements. It tries as much as possible to consider both paths that meet the requirements for real-time traffic, as well as maximize the throughput for non-real time traffic. This is due to the fact that critical applications such as battlefield surveillance have to receive for instance acoustic data regularly in order not to miss targets. The bandwidth ratio which is evenly distributed over the network nodes makes the protocol not flexible for adjustment of bandwidth sharing for different links. The author did not mention the simulation environment used and the results obtained were not compared to any standard algorithm to compare performance.

2.5.4.6 Message-initiated Constraint-Based Routing (MCBR)

MCBR (Zhang and Fromherz, 2004), is a message-initiated constraint based routing protocol for wireless ad-hoc sensor network. The authors

proposed MCBR which is a general message specification mechanism, and the aim is to explicitly encode the routing destinations, constraints and objectives in messages, so that generic-purpose instead of objective-specific or destination-specific routing strategies can be applied. In their work, they tried to separate routing specifications and routing strategies, hence making it possible for Meta routing exploration, and allowing quality-of-service (QoS) requirements at the application layer for individual messages. Two types of Meta routing were proposed for the MCBR; the search based and constraint-flooding. In MCBR, each node in the network has a list of attributes whose types are predefined. An MCBR protocol specification for a message m is; (U_m^o, C_m^d, C_m^r, O_m) where U_m^o is the source of the message, C_m^d is the set of destination, C_m^r is the set of route constraints and O_m is the objective. The goal of the routing is to deliver the message from U_m^o to one (unicast) or all of the destination nodes (multicast) while satisfying C_m^d via intermediate nodes p: $U_m^i, \ldots, U_m^{n-1}$ such that C_m^r is satisfied at U_m^i and $\min_p \sum_i O_m(U_m^i)$. The algorithm was implemented in NesC on TinyOS-1.x, and simulations were performed in the TOSSIM/TinyViz simulation environment. No comparisons were made with any existing routing protocols.

2.5.4.7 Smart Routing with Learning-based QoS-aware Meta-strategies

Smart Routing with Learning-based QoS-aware Meta-strategies is a framework of MCBR (Zhang et al., 2004a). The protocol consists of QoS specifications and sets of QoS-aware meta-strategies. The meta-strategies are; real-time search, constrained flooding and adaptive spanning tree. Its main focus is on learning based meta-strategies of which it does not create and maintain explicit routes. Instead, packets are discovered and routes improved during the search for a destination. The authors assume that, if a routing specification is given for a message, including the destination and

QoS requirements, it is then easy to define the cost function on each node, called Q-value indicating the minimum cost-to-go from the present node to the destination. The Q-value is computed using the equation below as:

$$Q_m = (1 - \alpha)Q_m + \alpha(O_m + \min_n NQ_m(n)) \quad (2.4)$$

Where α represents the learning rate and O_m the present value of the local objective function, whereby *n* is a neighbor of the node. With the use of Q-value, real-time search passes the packet to the "best" neighbor according to the estimates. The protocols were simulated in Prowler Rmase, compared with AODV was found to have better performance.

2.5.4.8 Ad-hoc On-demand Distance Vector (AODV) Routing Protocol

Ad-hoc On-demand Distance Vector (AODV) (Perkins and Royer, 1999) is a popular classical routing protocol for mobile ad-hoc networks. It is a variant of a distance vector routing algorithm. It has the characteristics of dynamic source routing in terms of on-demand, and discovers routes only as needed. Whenever a node has information about an event to send to the sink node and it does not have the valid routing table entry for the sink node, it sends a route request (RREQ) message to all its neighbors through the use of broadcasting scheme. The algorithm makes use of the broadcast route discovery scheme in searching for the best path to the destination, and then unicast route reply message when a route to the destination is known by any of the intermediate nodes. The original AODV makes use of periodic HELLO messages in order to ascertain the validity of links with its neighbors, but this version of AODV does not use HELLO message in detecting failure of path or links instead of the hello message, it makes use of the feedback message from the link layer, of which same information is gotten. In the case of link failure in AODV, a route error (RERR) message is sent to the source node using the unicast method of transmission, and the

corresponding routing table entries for the intermediate nodes is flushed so as to avoid the corruption of routing information in the table. If the source node gets duplicates of RREP packet messages, it selects the one with the least number of hops. The specific form of AODV used in this work, is the version that considered energy as a metric for routing decision and it is specifically modified to suit the energy constraint of WSNs which is included in routing modeling application simulation environment (RMASE) package. Though, there exist some other variants of AODV, which dealt with its reliability (Zamree et al., 2010).

2.6 Swarm intelligence based routing protocols

Swarm Intelligence (SI) is the characteristic of a system whereby the constituent agents of the system collectively interact locally with their environment, which in turn cause clear meaningful global patterns to emerge. Swarm based routing protocol (Dorigo, 2001) is a promising research on biological agents' behavior of which many of those agents are blind and communication between them is based on adoption of chemicals like substance known as pheromones, produced by the species and deposited on the paths while walking in search for food. With the aid of sensing pheromone trails, each agent in the system can take the path to food discovered by other ants. A pheromone table is created on each node which keeps the information on the concentration of pheromone of each path. A pheromone table at each node guide through the path selection. The pheromone table keeps the information gathered by the forward ant. Every node in the network has a table which keeps the amount of pheromone on each neighbor path. The node has a distinct pheromone scent, and the table is in the form of a matrix with destination nodes listed along the side and neighbor nodes listed across the top. Rows correspond to destinations and columns to neighbors. An entry in the pheromone table

is referenced by $T_{n,d}$ where *n* is the neighbor index and *d* denotes the destination index. The values in the pheromone table are used to calculate the selecting probabilities of each neighbor. Considering Fig. 2.6, when a packet arrives at *C* from previous hope *S*, i.e. the source, the source pheromone decay, and pheromone is added to the link $\overrightarrow{SC}$. The route is more likely to take through *C,* since it is the shortest path to the destination i.e. $\overrightarrow{SCED}$. The pheromone table of node *C* is shown in Fig. 2.6 with nodes E and S as its neighbor, A, B, E, D and S are the possible destinations.

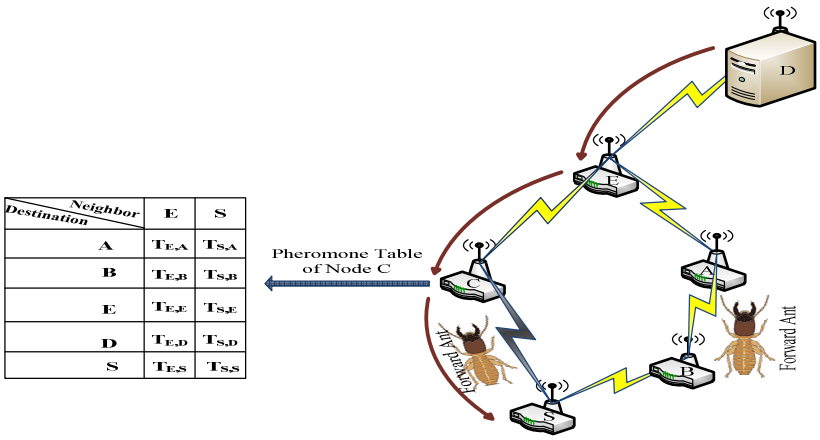


Fig. 2.6: Description of pheromone table of node C.

The ants choosing the closest path to the sink node will get back to the source node faster, and in turn add more pheromone on the path traversed by them, which increases the pheromone concentration on the path, and the path will be fully utilized by the forthcoming ants as the probability of selecting that path will be higher. The probability of selection is given as (Camilo et al., 2006)

$$P_k(r,s) = \begin{cases} \frac{[\tau(r,s)]^{\alpha} * [\eta(r,s)]^{\beta}}{\sum_{u \in j_k(r)} [\tau(r,u)]^{\alpha} * [\eta(r,u)]^{\beta}} & , s \notin j_{k(r)} \\ 0 & , otherwise \end{cases} \quad (2.5)$$

where $J_k(r)$ is the set of nodes that remain to be visited, η the savings of combining two nodes on one tour as opposed to serving them on two

different tours, τ the pheromone level on edge, (r, s, u) the node identifier, P_k represents the probability that an ant *k* chooses to hop from node to node, β evaporation coefficient of local research and α the evaporation coefficient of global research. Swarm based routing are classified into three categories: ant based, bee based, and slim based.

2.6.1 Ant Colony Optimization (ACO)

This section briefly describes the biological behavior of ants and the adoption of their behavior to routing problems in wireless sensor network. ACO is a class of optimization algorithms modeled on the actions of the ant colony, and a subset of Swarm Intelligence. Though, most of the protocols were originally designed for MANET, but are now applied to sensor networks. The optimization of network parameters for the WSN routing process for the enhancement of networks' lifetime might be considered as a combinatorial optimization problem. A lot of research work has been done with the collective attitude of biological species such as ants as a natural model for combinational optimization problems (Bonabeau et al., 1999; Hussain et al., 2008; Iyengar et al., 2007, White et al., 1998a, b). Ant colony optimization (ACO) algorithm simulating the behavior of ant colony has been successfully applied in many optimization problems such as the asymmetric travelling salesman (Liu et al., 2005), vehicle routing (Rodoplu et al., 1999), and WSN routing (Akkaya et al., 2005; Buczak et al., 1998; Camilo et al., 2006). Also, a mobile agent system that is based on Ant Colony Optimization (ACO) method informally, the AntNet algorithm was also proposed by M. Dorigo and G. Di Caro, which is based on packet routing in communication networks (Dorigo et al., 1998).

2.6.2 Swarm based data-centric routing protocols

2.6.2.1 Pheromone Based Energy Aware Directed Diffusion (PEADD)

PEADD (Zhu, 2007) varies slightly from DD (Intanagonwiwat et al., 2003) as it is based on swarm intelligence, and also based on ACO scheme. The protocol is aimed at maximizing the lifetime of sensor networks by involving nodes with higher energy in the information gathering process. In the algorithm, pheromone contents of paths are increased by its ants which are proportional to the remaining energy values of the nodes. The pheromone value of paths with higher remaining energy is increased while decreasing the pheromone of others i.e. the rate of decay of pheromone is proportional to the number of event of information transmitted in the network which is also a dependent of residual energy of each link. The update of pheromone value on each link is also dependent on the information or event transmitted along the link. The algorithm uses the same route selection and updating as that of the general ant based routing as described above. In their experiment, the results obtained showed that its performance was better as compared with Directed Diffusion (DD).

2.6.2.2 Comprehensive Routing Protocol (CRP)

Guo et al. (2010) proposed a comprehensive routing protocol (CRP) which its operation is based on an ant colony algorithm. The algorithm is an improved version of energy aware routing (EAR). But in its routing decision, it uses probability of selection, of which it considers the network lifetime and data packet arrival rate. The author point of argument is that the continuing use of the path which is considered as the best and the optimal path from the point of view might not be the best as it will lead to depletion of the path nodes energy, and instead proposes the use of sub-optimal paths occasionally. The protocol has three phases: routing table setup, data communication, and route maintenance. In the routing table

setup, from the destination node, searching packet is locally flooded until it arrives at the source node in order to find all the possible paths from source to destination and make sure the probability of each one is being chosen according to the square of transmission distance along the path, the strength of pheromone on each path, the remaining energy of the nodes, and the number of occurrences of each node acting as a router. All this is gotten from the probabilistic routing tables during the route discovery and update phases. During the data transmission, when a node is chosen as the next forwarding one, the pheromone strength of the branch between it and the previous node will be updated according to (2.6) and (2.7)

$$I_{ij} = I_{ij} + \Delta I_{ij} \tag{2.6}$$

Where I_{ij} is the strength of pheromone on the branch between nodes *i* and *j*, and ΔI_{ij} is the updated quantity, and is calculated as follows:

$$\Delta I_{ij} = ((1 - D_{j,d}^2 / \sum_{k \in N_i} D_{k,d}^2) \times T) \tag{2.7}$$

Whereby $D_{j,d}$ represents the distance from node *j* and the destination *d*, $D_{k,d}$ represent the distance from node *k* to node *d*, and *T* a constant. The route maintenance is responsible for reflection of the actual condition of the network. The algorithm was compared with EAR in NS-2, and shows promising solution but lacks QoS metrics.

2.6.3 Swarm Based Location-Based Protocols

2.6.3.1 Sensor Driven and Cost-Aware Ant Routing (SC)

In SC (Zhang et al., 2004b), it is assumed that ants have sensors so that they can smell where there is food at the beginning of the routing process so as to increase in sensing the best direction that the ant will go initially. In addition to the sensing ability, each node stores the probability distribution and the estimates of the cost of destination of each of its neighbors. It suffers from misleading data when there is an obstacle which

might cause errors in sensing. Assuming that the cost estimate is Q_n for neighbor n, the cost from the current node to the destination is 0 if it is the destination, otherwise,$C = min_{n\in N}(c_n + Q_n)$ where c_n is the local cost function and β is an evaporation coefficient of local search. The initial probability is calculated according to the expression:

$$P_n = \frac{e^{(C-Q_n)^\beta}}{\sum_{n\in N} e^{(C-Q_n)^\beta}} \quad (2.8)$$

2.6.4 Swarm based hierarchical protocols

2.6.4.1 Self Organizing Data Gathering Scheme (SDG)

SDG protocol (Kiri et al., 2007) aims to achieve scalability and reliability in sensor networks. In the protocol, a node uses another sink in case of sink failure. The scheme queries the fact that with a single sink, a sensor network cannot tolerate energy depletion as once a node near the destination node runs out of energy, the destination node will remains isolated and the sensor network becomes useless as packets can no longer be routed to the sink. In the scheme, routing overhead is minimized by allowing only the destination node to generate backward ants which are broadcasted by the destination nodes continuously. The protocol utilizes the normal ACO techniques in information transfer from sources of the information to the destination node, but the clustering of Nodes in the network is as a result of the grouping behaviors of the infant species observed in ant colonies. The protocol was evaluated in NS-2 with reliability metric. The algorithm was found to consume more energy due to its proactive path establishment and the use of hello message to dictate the validity of links.

2.6.4.2 Energy Balanced Ant Based routing protocol (EBAB)

Wang et al. (2009a) proposed an energy balanced ant based routing protocol (EBAB) which is an adaptive dynamic routing algorithm based on

ant colony optimization. In order to achieve their aims by balancing energy consumption in the network so as prolong network lifetime, the algorithm tends to divide into two separate parts so that energy consumption is brought to minima. This involves (1) intra-cluster and (2) inter-cluster. The first phase is further divided into sub-phases, and in turn creates clusters in each region in the initial routing process. The intra-cluster comprise of the completion for cluster heads where cluster head is competed for based on the areas of which each node belongs due to the strength to the base station. After the cluster head selection, cluster is set up, and the head will then send a message to all the nodes within its transmission range in request to join its cluster. In response to the request message from the head, each of the node reply so as to join the cluster. Any node in the network that receive more than one message as an invitation to join a cluster, will have to take a drastic measure on the cluster to join which is based on the distance from the requestor. The Intra-cluster section used the improved ACO algorithm. When compared with LEACH, it has better performance in terms of the number of nodes that survived at the end of the experiment in addition to its high success rate.

2.6.4.3 Adaptive Clustering for energy efficient WSN based on ACO (ACO-C)

Ziyadi et al. (2009) proposed ACO-C which is an adaptive clustering for energy efficient wireless sensor networks based on ACO. The algorithm proposed a new energy aware clustering protocol by using appropriate cost functions implemented at the base station. It minimizes and distributes the cost of long distance transmission and data aggregation among all sensor nodes evenly. The routing problem was adapted as a clustering problem in which the objective is to select K out of N nodes as cluster heads, which was achieved through agent consideration called software ants. The algorithm simulated in the Matlab platform was evaluated and compared

with LEACH. LEACH-C and PSO-C was found to perform better in terms of data delivery and network lifetime.

2.6.4.4 Ant Colony Clustering Algorithm (ACALEACH)

Wang et al. (2009b) proposed an Ant Colony Clustering Algorithm which is an ant colony based improved version of LEACH. The algorithm not only considers the node residual energy, but also the distance between the cluster heads was considered in selection of cluster heads. It applies the ACA into inter-cluster routing mechanism to lower energy consumption of cluster heads and which will in turn helps in network lifetime maximization. The algorithm as compared with its counterpart LEACH in the Matlab environment outperforms it in terms of average energy consumptions and the number of nodes that survived at the end of the experiment. The protocol did not consider throughput and delay in its routing process, and hence may also be weak in energy efficiency due to network overheads.

2.6.4.5 Multipath Routing Based on Ant Colony System (MACS)

Xiu-li et al. (2008) proposed Multipath Routing Based on Ant Colony System in Wireless Sensor Networks which endues the ants with new characteristic and searching method. The protocol tries to solve the problem of basic ACS being trapped in the solution of global optimization, and also deal with the contingency problem as soon as possible. The protocol was simulated in NS-2, and found to perform better than DD and ACS in terms of average transmission delay.

2.6.4.6 Data gathering communication (AntChain)

AntChain (Liu, 2004) is aimed at energy efficiency, data integrity, and node's life time parameters. It achieves a near optimal chain by using ant colony optimization method running on the base station. The sensor nodes in the network form a bi-directional chain structure which is self-adaptive

to any minor changes. Unlike other sensor routing protocols where sensor nodes have to communicate with each other to set up a transmission route, its sensor nodes only receive useful information through base station broadcast. The discarding of the routing operation preserves a significant amount of energy in the sensor network. In AntChain, sensor nodes do not have to be aware of the prior network knowledge. In the scheme, the sink initially sends a setup enable signal to all the nodes informing them of the starting of the network. The nodes by getting the message from the sink reply by sending their ID and locations or by just simply sending a short message to indicate that they are still alive. The main drawback of the protocol is the fact of being centralized, and the assumption of each node being able to communicate with the destination node without using intermediate nodes, which is not realistic in practical use.

2.6.4.7 A probabilistic zonal approach for swarm-inspired wildfire detection using sensor networks (PZSWiD)

PZSWiD (Ramachandran et al., 2008) aims at covering the speed of information propagation, the accuracy of the information being propagated and the reliability of the network as a whole over a long period of time. The protocol follows a data centric approach whereby the system executes a swarm inspired routing and aggregation algorithm. The algorithm uses a probabilistic model for representing information in a data centric sensor network. In PZSWiD, each node is endowed with the responsibility of responding to different queries sent out by the destination node in addition to forwarding of the sensed information to the destination node. The algorithm has the capability of attending to both queries and event triggered applications. Each sensor node in the network can serve as a source of events, and the path establishment is a proactive based scheme. Being of high complexity, the description of its parameters is difficult, and on selecting a node for information forwarding is based on the closeness of

the detected event with the query message sent from the sink, and the concentration of pheromone on the path. The algorithm was simulated in NS-2 with a variety of different zone radii, while analyzing its average energy dissipated and the average delay. It has not been compared with any existing protocol for performance measurements.

2.6.4.8 Ant Colony Based Multipath Routing Algorithm (ACMRA)

ACMRA (Yang et al., 2008) discovers disjoint multiple paths between the source nodes and destination node. In the multipath routing, multiple paths are established between sensor nodes and sink node. The algorithm utilizes two main types of ants, (1) the search ants which main functions are to search for possible route to the sink node and (2) the reinforcement ants, which takes the responsibility of going around the network so as to collect necessary information from the intermediate nodes and also information about the efficiency of the paths, and also serves the function of the pheromone update along the route back to the source nodes. The algorithm function as an on-demand for information about any event or occurrences on a particular location, as such, it lowers its network overhead. The algorithm was compared with primary and replication mode of multipath routing and found to perform better in terms of energy consumption and energy consumption deviation of nodes in the network. The environment of simulation of the protocol is not stated, and the network nodes not properly distributed, while also not considering quality of service metrics in its design.

2.6.4.9 Ant Colony Multicast Trees (ACMT)

De-min et al. (2008) proposed ACMT based on ant colony multicast trees of wireless sensor network routing. The algorithm tries to prolong the network lifetime by minimizing the communication process to energy consumption.

In the algorithm, the ants found trees, of which the tree that the ant found includes all the destination nodes. There is no single current node for every ants. Every node in the tree that has been found is likely to be the current node. Every step made by each ant has no other meaning of any path than to enable the current tree to grow further. The only principle observed by the new algorithm is the positive feedback mechanism of basic ant colony algorithm. The protocol compared with the YANG model and Flooding performs better in simulation. Also, as the network grows faster, the death rate of nodes becomes higher.

2.6.4.10 Ant Colony Optimization Based Location Aware Routing (ACLR)

ACLR (Wang et al., 2008) is a communication protocol which main logic is the selection of next hop by ants to a subset of the set of the node's neighbors instead of its whole neighbors which guarantees that the data packets are delivered towards the destination while avoiding loops. The protocol proposed a formula for estimating transition probability with which ants select their next hop nodes. For the determination of pheromone deposited by ants, a model was used and a novel scheme to evaporate the pheromone on various paths according to the residual energy and location information of nodes was proposed so as to increase diversity of the best solution by the ants. The protocol was compared with BAR, SC, FF, and IAR for performance.

2.6.4.11 Multi-Sink Swarm-Based Routing Protocol (MSRP)

MSRP (Paone et al., 2009) is a routing protocol for sensor networks which is self-organized, fault tolerant and environmentally adaptable. The protocol is inspired by slime mold organisms. The organism finds their advantage in the ability to organize themselves in clusters using pheromone generation and evaporation. The protocol organizes data traffic

towards the sink by adopting the gradient concept while showing autonomy and fault tolerance. The algorithm uses OMNET++ in the evaluation of its performances, signaling overhead, and adapt to changes in the environment. Fig. 2.7 shows the signaling process phases of the algorithm.

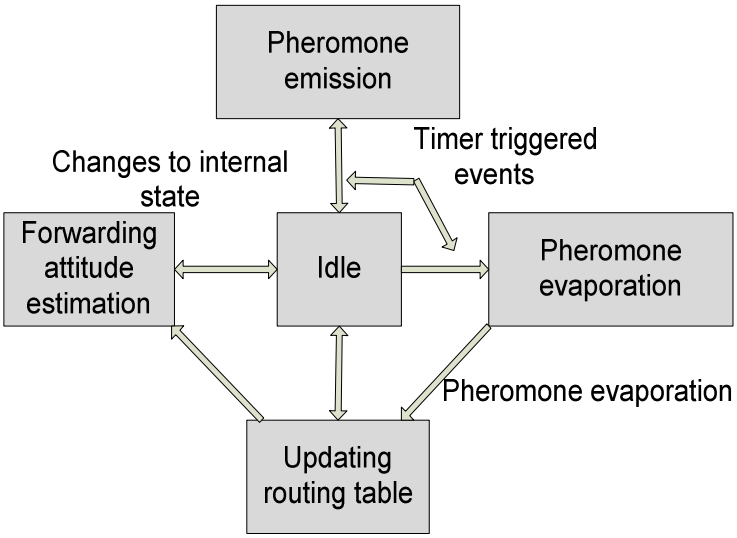


Fig. 2.7: signaling process phases of multi-sink swarm based routing.

2.6.4.12 Jumping Ant Routing Algorithm (JARA)

JARA (Chen et al., 2007) combines the advantages of reactive and proactive routing to speed up the route discovery time and reduce the route discovery overhead in the sensor network. It combines a MANET algorithm based on ant colony (ARAMA), and the zone routing protocol (ZRP), while also employing a jumping mode to reduce the proactive overhead. The algorithm has two parts. The first part deals with the process by which nodes use proactive routing protocol to maintain the topology of number of hops. The second part involves how each node applies ant routing to discover paths outside its zone. This means that every node maintains a zone, and each ant can jump a zone. The algorithm shortens the route discovery time and reduces route discovery overhead especially in dense topologies as compared to ARAMA in its simulation results.

2.6.4.13 An Ant Colony Optimization-based Load Balancing Routing Algorithm (ACOLBR)

Bi et al. (2010) proposed An Ant Colony Optimization-based Load Balancing Routing Algorithm for Wireless Multimedia Sensor Networks to help in solving constraints of WSNs, the protocol first built intra-cluster routing by a minimum spanning tree algorithm with the cluster head as the root. Then the inter-cluster routing is built by ACO to get an optimal path from cluster heads to the sink. It also uses the message's positive feedback to take the node's residual energy, transmission delay and the propagation distance as the heuristic factor which ensure the QoS of the network transmission. The algorithm as compared to M-IAR and AGRA protocols, achieved better performance as compared to them in terms of low latency and energy utilization efficiency.

2.6.5 Swarm based network flow and QoS-aware protocols

2.6.5.1 Energy Efficient Ant Based Routing Algorithm (EEABR)

EEABR proposed by Camilo et al., (2006) is an algorithm that works on the principle of ants' mode of behavior. In the protocol, each node in the network launches a forward ant at a regular interval with the aim of finding a route to the sink node. In the protocol, two ant are used for path determination and update which includes the forward and backward ants, and each ant only carries the address of the last visited nodes which means that router nodes only carries the records of received and forwarded ants in their tables. The table content of each node contains the previous node, forward node, ant identification, and timeout value. In the algorithm, the determinant of path selection is the concentration of path pheromone, and the amount of pheromone trail to be deposited by the backward ant on its return back to the source node is calculated using;

$$\Delta\tau = \frac{1}{C - \left\lceil \frac{EMin_k - Fd_k}{EAvg_k - Fd_k} \right\rceil} \tag{2.9}$$

And the equation used to update the routing tables at each node is:

$$\tau(r,s) = (1-\rho) * \tau(r,s) + \left[\frac{\Delta\tau}{\emptyset.Bd_k}\right] \quad (2.10)$$

Where ρ represents pheromone evaporation factor, C the initial energy level of the nodes, Bd_k the number of visited nodes by the backward ant k, $EMin_k$ is the minimum value of the vector E_k, $EAvg_k$ the average of the vector values, ϕ is a coefficient of which ϕ and $Bd_{k,}$ are two parameters that will force the ant to lose part of the pheromone strength during its way to the source node. The idea behind the behavior is to build better pheromone distributions in order for nodes near the sink to have more pheromone levels and hence forces remote nodes to find better paths. Such behavior is important when the sink node is able to move, since pheromone adaptation will be much quicker (Kalpakis et al., 2002). When compared to basic ant based routing (BABR) and improved ant based routing (IABR), it performs better in terms of energy efficiency, the average energy of nodes and the energy of node with minimum energy. The disadvantages are that it lacks quality of service and increases excessive delay in packet delivery.

2.6.5.2 Flooded Forward Ant Routing (FF)

FF (Zhang et al., 2004b) argues the fact that ants even augmented with sensors, can be misguided due to the obstacles or moving destinations. The protocol is based on the flooding of ants from the source to the sink node. In the case where the specific destination is not known at the beginning by the ants, or cost cannot be estimated, SC protocol then becomes the basic ant routing, and the problem of moving about the network without a definite destination or purpose to find the destination node exist. This is the case where FF exploits the network with the broadcast channel of wireless sensor networks. That is, the protocol simply uses the broadcast method of sensor networks so as to route packets to

the destination. The basic idea is to flood forward ants in the network in order to find the destination. It is basically a variant of the SC protocol with the aim of flooding the network with forward Ant similar to the basic Ant based routing.

2.6.5.3 Flooded Piggyback Ant Routing (FP)

FP (Zhang et al., 2004b) brings a new ant species to forward ants; namely data ants whose function is to carry the forward list. The control of the flooded forward ants is the same as in FF. The protocol succeeded in combining forward ants and data ants using constrained flooding to route data and to discover optimal paths at the same time so as to minimize energy consumption of the network with the data ants carrying the forward list. In the case of control of the flooded forward ant, the algorithm passes the detected event to the sink node, and also keeps in its memory the paths that the backward ants will take back to the source node so as to build a better probability value on the links. As compared to FF, SC, and basic ant routing, in routing modeling application simulation environment (RMASE), it was found to outperform others with high success rate, but incurred relatively high energy consumption.

2.6.5.4 Energy-Delay Ant-Based Routing (E-D ANTS)

E-D ANTS algorithm (Wen et al., 2008) is designed to minimize the time delay in fixed packet transfer for the purpose of energy constrained sensor networks. In the protocol, Energy x Delay model based on ant algorithms was proposed called E and D ants. The protocol aimed at lifetime maximization, and real time data transmission service of sensor networks. The protocol is reactive path establishment scheme. In the protocol, each ant carries a memory stack which stores the remaining energy values of the path traversed by them along with the number of hops taken while hopping from node to node. In the simulation environment of OPNET where

it is being simulated, E and D ANTS converge faster than AntChain and AntNet of which it was compared with. The algorithm is flat in nature would be difficult to scale through large topologies, unless it introduces hierarchical techniques.

2.6.5.5 Ant Colony Based Reinforcement Learning Algorithm (AR and IAR)

Adaptive and improved adaptive AR & IAR (Ghasemaghaei et al., 2007) uses a probability distribution like other ant based routing in decision making on the paths to take in its routing decision. Its main difference is the reinforcement learning algorithm employed by the backward ants in order to get a better and more efficient route than the one taken by the forward ant. In the improved version, IAR, a modified heuristic correction factor $A_{i,d}$ is used which is the cost from the neighbor node *i* to the destination node *d* so as to find a better probability of choosing a better path for hopping. The algorithm as compared with Basic ant routing, SC, FF, and FP performs better in low energy consumption of nodes, high energy efficiency, less latency, and high success rate.

2.6.5.6 A Bee-inspired power aware routing (Beesensor)

Beesensor (Saleem et al., 2012) is a reactive and on demand routing protocol and its mode of operation is based on foraging principles of honey bees. The algorithm has three types of agents which makes it sells well in its mode of operation. The agents are named packers, scouts and foragers. Packers search for foragers for information on the source nodes, whereas scouts takes the responsibility of route discovery to the sink node, but the foragers do the main work whose function is to hop from node to till they get to the sink node. They hop along with data packets carried by them. Its analysis was done and compared with several routing protocols in RMASE simulator. Since it is an on demand based protocol, it suffers in the area of

security application or an application where information needs updating at regular intervals of time. The routing process is reactive, the time use to search for the sink in a dynamic scenario of routing process is high, and as such the algorithm is only good for static scenario. The approach is based on the interactions of scouts and source routing in which small forwarding tables are built during the return of a scout. Its analysis was done and compared with EEABR, FP, and AODV in RMASE simulator. It is an on demand based protocol and may not be fitted for security applications or applications where information needs to be updated at regular intervals of time.

2.6.5.7 A Bio-inspired Power Efficient Routing Scheme (iACO)

iACO (Mahadevan and Chiang, 2010) is based on metaphors of ant food foraging behavior, and partly based on the efficient Max-Min algorithm and quite suitable for flexible structure of wireless sensor networks. In the proposed algorithm, each packet is regarded as individual ant communicating with each other via pheromone values stored in each sensor node routing table. The algorithm constitutes of mainly three steps: generation of local solution based on the paths; pheromone update; and decision making in its update of pheromone table. A mutation parameter is added to the pheromone update rules, where the concept of threshold is also involved so as to increase the convergence speed of the algorithm. The mutation parameter is partly based on best-worse ant algorithm (BWACA) as well as max-min ant system. The algorithm was not compared with other existing algorithms for performance evaluation. The throughput decreases as the network grows and hence has poor QoS.

2.6.5.8 ACO-based Quality-of-Service Routing (ACO-QoSR)

This algorithm was proposed by Cai et al. (2006) which is a reactive protocol whose main aim is to find a solution to delay requirements as well as limited energy and computational limitation of sensor networks. The protocol finds the route from source nodes to the destination node in a way that it achieves low latency as compared to the boundary value *D*, and the remaining energy ratio of node, ERR= $E_{residual}/E_{initial}$, is higher than the reference value. In the protocol, each ant sends its message using the unicast scheme to their next hop using selection probability as in (Cai et al., 2006). The algorithm uses *m* forward ants for each path's probe. It also uses the max-min ant system for smoothing and boundary mechanism. The algorithm utilizes single ant for finding the delay constrained path which is not clear and suspicious in the algorithm description in addition to the algorithm forward ants carrying routing information on their header will cause high energy consumption by the network nodes and it will encounter high latency in packet delivery in a large network.

2.6.5.9 Ant Colony Based Many-to-One sensory data Routing (MO-IAR)

Many-to-one improved adaptive routing (MO-IAR) (Ghasemaghaei et al., 2008) involves two sections of operation. During the first section, forward and backward ants are employed to find the shortest route within multi-hop sensor networks, while in the second phase data ants route the actual sensory data through the shortest path. The protocol is capable of routing upstream information through the shortest path by avoiding congestion, and hence can handle both event-based and periodic many-to-one sensory data flow. The first phase which is concerned with finding the best route using ant colony optimization uses the GPS technology and assumes that each node in addition to knowing its present and past location is also aware of the destination nodes' location. One of the sensor nodes is initially

deployed and each sensor node locally broadcasts a HELLO message to its neighbors to form the neighbor table. In the second phase as soon as the shortest path to the sink is found, the protocol employs the data ants to route the actual data captured by N number of source nodes destined for the sink. The performance of the protocol as compared with SC, FF, and FP protocol in simulation, outweigh the three protocols in terms of low end-to-end delay of packets and having a low collision rate. The algorithm specifies the route for which information should take, and extra information is carried by the forward ants in their header which can cause excessive use of energy and also delay in packet delivery to the sink.

2.6.5.10 Ant-aggregation

This scheme of routing is proposed by Misra and Mandal (2006) and argues the fact that a combination of multi hop data dissemination with aggregation of event in the network before transmitting to the sink node will reduce energy consumption and in turn enhance network lifetime. The algorithm is based on ACO principles, and at every node in the network, a forward ant is sent to the next hop with a certain probability defined in the protocol. In the protocol, the ants either try to find the shortest route to the sink or find the closest aggregation point of the route searched by previous ants. The algorithm converges to local best aggregation tree. As compared with opportunistic aggregation and greedy algorithms, it performs better in terms of energy reduction consumed at the network.

2.6.5.11 Ant Based Service-Aware Routing Algorithm (ASAR)

ASAR (Sun et al., 2008) chooses suitable paths to meet diverse QoS requirements from different kind of services and is mostly suited for multimedia sensor networks. The work aimed at targeting network with QoS requirements. And the basic services include event-driven, data query and stream query. Since the algorithm chooses suitable paths to meet

diverse QoS requirements of different kinds of services, it maximizes network resource utilization and also improves on its network performance. From their simulation result, and compared with DD and Dijkstra's algorithm in NS-2, has better convergence and significantly provides better QoS for multiple types of services in the multimedia sensor networks.

2.6.5.12 Basic Ant Based Routing (BABR) for WSN

The basic ant routing algorithm and its main characteristics (White et al., 1998a, b; Dorigo and Caro, 1998) can be summarized as follows:

1. At regular intervals along with the data traffic, a forward ant is launched from source node to sink node.
2. Each agent (forward ant) tries to locate the destination with equal probability by using neighboring nodes with minimum cost joining its source and sink.
3. Each agent moves step-by-step towards its destination node. At each intermediate node a greedy stochastic policy is applied to choose the next node to move to. The policy makes use of (i) local agent-generated and maintained information, (ii) local problem-dependent heuristic information, and (iii) agent-private information.
4. During the movement, the agents collect information about the time length, the congestion status and the node identifiers of the path followed.
5. Once the destination is reached, a backward ant is created which takes the same path as the forward ant, but in an opposite direction.
6. During this backward travel, local models of the network status and the local routing table of each visited node are modified by the agents as a function of the path they followed and of its goodness.
7. Once they have returned to their source node, the agents die.

With this procedure, the ants are able to build better path and good routing is achieved. The link probability distribution of the algorithm is maintained by;

$$\sum_{i \in N_k} P_{ji} = 1; \; j = 1, \ldots, N. \tag{2.11}$$

2.6.5.13 Ant Colony-based Energy-Aware Multipath Routing Algorithm (ACO-EAMRA)

Xia and Wu (2009) proposed Ant Colony-based Energy-Aware Multipath Routing Algorithm. The algorithm considers the available power of nodes and the energy consumption of each path as the reliance of routing selection. Parameters q and q_o were proposed to improve the state transition rule and the possibility of ants to find a new path to avoid local optimization. The algorithm was compared with DD and performs better in term of energy saving ability. It does not put QoS into consideration in its routing process.

2.6.5.14 Energy Efficient ACO based QoS Routing (EAQR)

Jietai et al. (2009) proposed energy efficient ACO based QoS routing (EAQR). The protocol is based on improved ant colony optimization algorithm. The protocol gives preference to the provision of QoS and balancing of energy consumption over the entire network. With its introduction of minimum path energy, path hop count, and by means of advancing pheromone trail model of the ant colony system, it innovatively provides two heuristic ways based on the length and the comfort of the path to meet the different performance requirements of real time and common traffics. Hence, it provides service differentiation between real time and best effort traffic by introducing the new dual pheromone heuristic model in ant colony system.

2.6.5.15 An Adaptive QoS and Energy aware routing algorithm (IACR)

Peng et al. (2008) proposed an adaptive QoS and energy aware routing algorithm which is an improved version of ant colony routing (IACR). The protocol considers QoS alongside with balancing the nodes energy utilization so as to prolong the network lifetime. The algorithm is composed of two parts, the routing discovery as in the basic ant routing, and routing maintenance where the routing table is maintained and timely response to topology changes. Due to the concern of available bandwidth, the algorithm is more meaningful for real time high bandwidth traffic requirement like voice and video transmission. Simulation result in Omnet++ as compared with DD, show its best performance though delay in packet forwarding to the sink.

2.6.5.16 Quality of Service Based Distance Vector Routing Protocol (QDV)

Dhurandher et al. (2008) proposed a quality of service based distance vector routing protocol using ant colony optimization for wireless sensor networks (QDV). The protocol considers quality of service and reputation of the network in its routing decision. It pointed out that the high value of the reputation of a node signifies that the node is trusted and is more reliable for data communication purposes, and as node shows signs of misbehavior, its reputation decreases which in turn affects its quality of security, thereby disabling the malicious nodes from gaining access to the network. In the algorithm, every sensor node has information about the neighboring nodes in the network, the network has independent nodes and communication between them is distance dependent. In the network, any node can be malicious, so isolating it, facilitating efficient, and secure transfer is the protocol area of consideration. Simulation results show that its performance is better than SNEP, though due to selection of the most

secure node, it causes excessive delay in packet delivery and hence consumes network energy.

2.6.5.17 Ant-Based Routing for Wireless Multimedia Sensor Networks (AntSensNet)

Cobo et al. (2010) proposed an ant-based routing for wireless multimedia sensor networks using multiple QoS metrics known as AntSensNet. The protocol uses an efficient multi-path video packet scheduling in order to achieve minimum video distortion transmission. AntSensNet combines hierarchical structure with ACO-based routing so as to certify the quality of service requirement of sensor networks. Beside its support for power efficient multi-path video packet scheduling scheme for minimum video distortion transmission, it comprises of both reactive and proactive components. It is reactive since routes are set up when needed, and proactive due to the fact that, while a data session is in progress, paths are probed, maintained, and improved proactively using a set of special agents. The algorithm operates in three parts. The cluster network forms nodes into colonies, network route between clusters that meet the requirements of each application using ants, and forwarding of network traffic using the previously discovered route by the ants. In the clustering process, only the channel heads transmit information out of the cluster which helps in preventing collision between sensor nodes in the cluster, hence promoting energy saving and latency. Simulation result of NS-2 shows that the proposed distortion reduction technique used to transport video packets results in better quality video than using TPGF, and ASAR.

2.7 Analytical comparison of classical and swarm intelligence routing protocols

This section describes the comparison of classical and swarm intelligence based routing protocols based on their classification, energy efficiency,

data aggregation, location awareness, path selection, traffic pattern structure, and their simulation environment.

2.7.1 Comparison of data-centric classical and swarm intelligence routing protocols

Tables 2.3-2.6 show the main characteristics of the different routing protocols in both classical and swarm intelligence based. The tables, show the analytical comparison of all the surveyed routing protocols according to their network structure; Data-centric, Location, Hierarchical, Network flow and QoS aware. Each of the routing protocols was described based on the network structure, energy efficiency, data aggregation, location awareness, route selection, query based and simulation environment.

2.7.2 Discussion on Analytical Comparison of Data-Centric Routing Protocols

Many protocols in this category belong to the classical routing protocols. CRP which is a variant of swarm intelligence has the highest energy utilization efficiency, though, it is not a query based protocol and as such has limitations when it comes to query based application. Being a proactive protocol, it is most suited in periodic based applications. This means that, it will generate high overhead in dynamic or mobility scenario of sensor networks. As seen from the table, GBR as a hybrid protocol which should have showed reasonable performance, the environment in which the authors conducted the experiment was not defined, hence it did not give room for comparison. The most efficient among this category is the EAR, which is strong in energy utilization efficiency.

Table 2.3: Comparison of data-centric routing protocols in WSN

Routing Protocols	Classification	Energy Efficiency	Data Aggregation	Location Awareness	Route Selection	Query Based	Simulation Environment
SPIN (2.5.1.3)	Classical	Weak	Yes	No	Proactive	Yes	NS-2
F&G (2.5.1.1)	√	Moderate	Yes	No	Reactive	Yes	NS-2
DD (2.5.1.2)	√	Weak	Yes	No	Reactive	Yes	NS-2
EAR (2.5.1.5)	√	Strong	No	Yes	Reactive	Yes	OPNET
RR (2.5.1.6)	√	Weak	Yes	No	Reactive	Yes	LecsSim
CADR (2.5.1.7)	√	Weak	Yes	No	Reactive	Yes	Math. Model
COUGAR (2.5.1.9)	√	Weak	Yes	No	Reactive	Yes	Unknown
EAD (2.5.1.10)	√	Strong	Yes	No	Reactive	No	NS-2
GBR (2.5.1.4)	√	Moderate	Yes	No	Hybrid	Yes	Unknown
ACQUIRE (2.5.1.8)	√	Moderate	Yes	No	Hybrid	Yes	Math. Model
CRP (2.6.2.2)	Swarm	Very strong	No	No	Proactive	No	NS-2
PEADD (2.6.2.1)	√	Strong	Yes	No	Reactive	Yes	NS-2

Mathematical Model=Mathematical modeling, C-Lang = C-Language, DES=discrete event simulator

Table 2.4: Comparison of Location Based Routing Protocols in WSNs

Routing Protocols	Classification	Energy Efficiency	Data Aggregation	Location Awareness	Route Selection	Query Based	Simulation Environment
GEAR (2.5.2.2)	Classical	Moderate	No	Yes	Reactive	No	Testbed
TBF (2.5.2.5)	√	Moderate	Yes	No	Reactive	Yes	NS-2
EAGRP (2.5.2.6)	√	Strong	Yes	Yes	Reactive	No	OPNET
GAF (2.5.2.1)	√	Weak	No	Yes	Hybrid	No	NS-2
MECN (5.2.3)	√	Weak	No	Yes	Reactive	No	NS-2
SMECN (2.5.2.4)	√	Weak	No	Yes	Reactive	No	NS-2
SC (2.6.3.1)	Swarm	Strong	No	No	Hybrid	No	RMASE

2.7.3 Discussion on Analytical Comparison of Location Based Routing Protocols

In Table 2.4 above, the only protocol that belongs to swarm intelligence is SC. It is strong in energy utilization efficiency and also a hybrid protocol. As a hybrid protocol, it combines the characteristic of both reactive and proactive, hence it is best in event based applications as well as periodic based application to some level, but cannot be applied to query based applications, and its experiment was performed in a well WSN simulation environment. TBF is a query based and good in query based applications and not also bad in energy utilization efficiency. Being also reactive, it will generate fewer control packets in dynamic and mobility scenario of WSN environment.

Table 2.5: Comparison of hierarchical routing protocols in WSNs

Routing Protocols	Classification	Energy Efficiency	Data Aggregation	Location Awareness	Route Selection	Query Based	Simulation Environment
LEACH (2.5.3.1)	Classical	Strong	Yes	No	Proactive	No	Matlab
SOP (2.5.3.3)	√	Weak	No	No	Proactive	No	Unknown
TEEN (2.5.3.4)	√	Strong	Yes	No	Reactive	No	NS-2
PEGASIS (2.5.3.2)	√	Strong	Yes	No	Hybrid	No	Unknown
APTEEN (2.5.3.5)	√	Very Strong	Yes	No	Hybrid	No	NS-2
HEED (2.5.3.6)	√	Strong	Yes	No	Hybrid	No	Unknown
EAR-CSN (2.5.3.7)	√	Strong	Yes	Yes	Hybrid	No	Math. Model
BCEE (2.5.3.8)	√	Strong	Yes	No	Hybrid	No	Matlab
SDG (2.6.4.1)	Swarm	Very Strong	Yes	No	Proactive	No	NS-2
EBAB (2.6.4.2)	√	Strong	Yes	No	Proactive	No	Unknown
ACO-C (2.6.4.3)	√	V.Strong	Yes	No	Proactive	No	Matlab
ACALEACH (2.6.4.4)	√	Very Strong	Yes	No	Proactive	No	Matlab
MACS (2.6.4.5)	√	Moderate	No	No	Proactive	No	NS-2

Table 2.5 Comparison of hierarchical routing protocols in WSNs (Cont'd)

AntChain (2.6.4.6)	√	Strong	Yes	Yes	Reactive	Yes	NS-2
PZSWiD (2.6.4.7)	√	Very Strong	Yes	No	Reactive	Yes	NS-2
ACMRA (2.6.4.8)	√	Moderate	No	No	Reactive	No	Unknown
ACMT (2.6.4.9)	√	Weak	No	No	Reactive	No	Unknown
ACLR (2.6.4.10)	√	Strong	No	Yes	Hybrid	No	OPNET
MSRP (2.6.4.11)	√	Strong	No	No	Hybrid	No	OMNET++
JARA (2.6.4.12)	√	Weak	No	No	Hybrid	No	Unknown
ACOLBR(2.6.4.13)	√	Moderate	Yes	No	Hybrid	No	NS-2
ACO-RC (2.6.4.14)	√	Very Strong	No	No	Hybrid	No	Matlab

Table 2.6: Comparison of network flow and QoS-aware routing protocols in WSNs

Routing Protocols	Classification	Energy Efficiency	Data Aggregation	Location Awareness	Route Selection	Query Based	Simulation Environment
MLDG (2.5.4.1)	Classical	Strong	Yes	Yes	Proactive	No	Unknown
SAR (2.5.4.2)	√	Weak	Yes	No	Hybrid	Yes	Parsec
MLER (2.5.4.3)	√	Strong	No	No	Hybrid	No	C-Lang
SPEED (2.5.4.4)	√	Weak	No	No	Hybrid	Yes	GloMoSim
EAQSR (2.5.4.5)	√	Strong	No	No	Hybrid	Yes	Math. Model
MCBR (2.5.4.6)	√	Strong	No	No	Hybrid	No	RMASE
AODV (2.5.4.8)	√	Weak	No	No	Reactive	Yes	Parsec
EEABR (2.6.5.1)	Swarm	Very Strong	No	No	Proactive	No	NS-2
AR & IAR (2.6.5.5)	√	Strong	No	No	Proactive	No	Java
iACO (2.6.5.7)	√	Strong	No	No	Proactive	No	Castalia
MO-IAR (2.6.5.9)	√	Moderate	No	No	Proactive	No	Java
Ant-Aggregation (2.6.5.10)	√	Strong	Yes	No	Proactive	No	Matlab
ASAR (2.6.5.11)	√	Strong	No	No	Proactive	No	NS-2

Table 2.6 Comparison of network flow and QoS-aware routing protocols in WSNs (Cont'd)

BABR (2.6.5.12)	√	Weak	No	No	Proactive	No	RMASE
ACO-EAMRA (2.6.5.13)	√	Strong	No	No	Proactive	No	Unknown
EAQR (2.6.5.14)	√	Very Strong	No	No	Proactive	No	NS-2
IACR (2.6.5.15)	√	Strong	Yes	No	Proactive	Yes	OMNET++
E-D ANTS (2.6.5.4)	√	Strong	No	No	Proactive	No	OPNET
Beesensor (2.6.5.6)	√	Very Strong	No	No	Reactive	No	RMASE
ACO-QoSR (2.6.5.8)	√	Strong	No	No	Reactive	No	NS-2
QDV (2.6.5.16)	√	Moderate	No	No	Reactive	No	GloMoSim
FF (2.6.5.2)	√	Weak	No	No	Hybrid	No	RMASE
FP (2.6.5.3)	√	Weak	No	No	Hybrid	No	RMASE
AntSensNet (2.6.5.17)	√	Strong	Yes	No	Hybrid	No	NS-2

2.7.4 Discussion on Analytical Comparison of Network flow and QoS-Aware Routing Protocols

Many protocols in this group belong to swarm intelligence, and the classical protocols in this group are hybrid. Hybrid protocols have an advantage in the sense that they combine both the characteristic of reactive and proactive in path establishment. The only proactive algorithm that belongs to classical in this group has no defined environment of the experiment. Even though it is promising in energy utilization efficiency, we cannot guaranty that due to the fact that the results provided in their report cannot be verified. Among this group, Beesensor is more energy efficient and the same time a reactive protocol. Being reactive (on-demand) protocol, it has lots of advantages like low network overhead since fewer control packets will be generated during the routing process. Its performance was conducted in a well know WSN environment. Some promising protocols like MLER, MO-IAR, Ant-Aggregation, perform their experiment in self-designed simulation environment which will not give a fair comparison. EEABR is also very strong in energy efficiency, but it is proactive which means that it will have high overhead since much control packets are used in the path update even though the path is not needed at the time of route discovery. It is best in the area of information is periodically needed (periodic based application).

2.8 Experimental Comparison of Classical and Swarm Intelligence Routing Protocols

2.8.1 Experimental Parameters

We used the Routing Modeling Application Simulation Environment (RMASE) (PARC, 2006; Zhang et al., 2006; Zhang, 2005) which is a framework implemented as an application in the probabilistic wireless network simulator (Prowler) (Sztipanovits, 2004). The simulator is written and runs under Matlab, thus providing a fast and easy way to prototype

applications and having nice visualization capabilities for the experimental and comparison purpose. Prowler is an event-driven simulator that can be set to operate in either deterministic or probabilistic mode. It consists of radio model as well as a MAC-layer model. The MAC layer simulates the Berkeley motes' CSMA protocol, including the random waiting and back-offs. The radio propagation model determines the strength of a transmitted signal at a particular point of the space for all transmitters in the system. Based on this information, the signal reception conditions for the receivers can be evaluated and collisions can be detected. The signal strength from the transmitter to a receiver is determined by a deterministic propagation function, and by random disturbances. The transmission model is given by:

$$P_{rec,ideal}(d) = P_{transmit}\frac{1}{1+d^{\gamma}} \tag{2.12}$$

$$P_{rec}(i,j) = P_{rec,ideal}\left(d_{i,j}\right).\left(1+\alpha(i,j)\right).\left(1+\beta(t)\right) \tag{2.13}$$

Where $P_{rec,ideal}$ is the ideal reception signal strength, $P_{transmit}$, the transmission signal power, d, the distance between the transmitter and the receiver, γ, a decay parameter with typical values of 2≤ γ ≤ 4, α and β, are random variables with normal distributions $N(0,\sigma_{\alpha})$ and $N(0,\sigma_{\beta})$, respectively.

Table 2.7: Simulation parameters

Parameters	Values
Routing Protocols	Beesensor, BABR, AODV, MCBR, EEABR, SC, FF, FP.
X_dist, Y_dist, Number of Nodes	1, 1, 9
Source type, center type, radius, rate, Random rate	Static, random, 1, 4, 0
Destination type, center type, radius, rate, random rate	Static, random, 1, 0.5, 0
Maximum Hops, Data Traffic	Infinity, Constant Bit Rate (CBR)
Data rate	250 Kbps
Simulation Time	100 Sec.
Nodes Energy	30 Joules each
Ant Ratio	2
AntStart, EEABRAntStart	240000, 240000
Window size, C1, Z	10, 0.7, 1
Reward scale	0.3
Data gain	1.2
Bees Learning Rate	1
Bees Resend, ForwardDelta, MaxDelay, FloodTemp	1, Inf, 4000, 5
Rrep_delay, Rreq_Timeout, Rreq_Delay, Aodv_Rtable_size, Aodv_RQCache_size, Rrep_Retries	40000, 40000, 4000, 10, 10, 3

2.8.2 Performance metrics

From several results obtained from our simulation results using the simulation parameters shown in the Table 2.7. We report the following performance metrics for clarity purpose.

1. *Latency:* it is the time delay of an event sent from the source node to the destination node. We reported it in seconds (s).
2. *Success rate:* it is a ratio of total number of events received at the destination to the total number of events generated by the nodes in the sensor network. We reported it in percentage (%).
3. *Energy consumption:* it is the total energy consumed by the nodes in the network during the period of the experiment (Joules).
4. *Energy efficiency*: it is a measure of the ratio of total packet delivered at the destination to the total energy consumed by the network nodes i.e. $\left(\frac{Succes\ rate * total\ size\ of\ packet\ sent\ to\ the\ sink\ in\ bits}{Total\ energy\ consumed}\right)$ (Kbits/J).
5. *Throughput:* it is the average rate of successful packets delivered over the network. It is measured in size of data packets in bits per second (Kbits/s).
6. *Standard Deviation*: this gives the average variation between energy levels of all nodes in the network (Joules).

2.8.3 Simulation results

In our evaluation, we compare the performance of six (6) swarm-based routing protocols in WSNs: EEABR (6.1), SC (6.2), FF (6.3), FP (6.4), Beesensor (6.10), BABR (6.21), and four (4) classical routing protocols; AODV, MCBR-AST, MCBR-RTSR, and MCBR-CFR using the metrics in Section (2.8.2). The results are shown in Tables 2.8 through 2.17.

Table 2.8: Simulation results for Energy-Efficient Ant-Based Routing Protocol (EEABR)

Simulation time(s)	Latency (s)	Throughput (Kbits/s)	Success rate (%)	Energy consumption(J)	Stand. dev.(J)	No. of packets	Energy efficiency(Kbits/J)
10	0.0312	3.5812	88.8890	2.0064	0.2411	36	15.9490
20	0.3227	3.6439	90.7890	3.7024	0.5105	76	18.6365
30	0.0317	3.7322	93.1030	5.2544	0.7910	116	20.5541
40	0.0320	3.6982	92.3080	6.9152	1.0531	156	20.8238
50	0.0317	3.6987	92.3470	8.5936	1.3186	196	21.0622
60	0.0315	3.6988	92.3730	10.1104	1.5923	236	21.5620
70	0.0319	3.699	92.3910	11.8064	1.8616	276	21.5984
80	0.0318	3.6991	92.4050	13.4848	2.1271	316	21.6540
90	0.0316	3.6993	92.4160	15.0016	2.4009	356	21.9311
100	0.0315	3.6994	92.4240	16.6624	2.6624	396	21.9656

Table 2.9: Simulation results for Sensor-driven and Cost-aware ant Routing (SC)

Simulation time(s)	Latency (s)	Throughput (Kbits/s)	Success rate (%)	Energy consumption(J)	Stand. dev.(J)	No. of packets	Energy efficiency(Kbits/J)
10	0.0380	1.2519	30.5560	1.1872	0.1180	36	9.2656
20	0.0327	2.1298	52.6320	2.7424	0.4389	76	14.5859
30	0.0319	2.3625	58.6210	4.2352	0.7493	116	16.0560
40	0.0317	2.5013	62.1790	5.7328	1.0613	156	16.9200
50	0.0315	2.6034	64.7960	7.3056	1.3880	196	17.3839
60	0.0314	2.6369	65.6780	8.8128	1.7016	236	17.5881
70	0.0314	2.6748	66.6670	10.3680	2.0250	276	17.7470
80	0.0314	2.7037	67.4050	11.9232	2.3484	316	17.8643
90	0.0314	2.7258	67.9780	13.4336	2.6626	356	18.0147
100	0.0313	2.7534	68.6870	14.9488	2.9782	396	18.1955

Table 2.10: Simulation results for Flooded Forward ant Routing (FF)

Simulation time(s)	Latency (s)	Throughput (Kbits/s)	Success rate (%)	Energy consumption(J)	Stand. dev.(J)	No. of packets	Energy efficiency(Kbits/J)
10	0.1030	1.0178	25.0000	3.4320	0.1464	36	2.6224
20	0.0683	2.3348	57.8950	6.4320	0.3816	76	6.8408
30	0.0612	2.7044	67.2410	8.4960	0.6488	116	9.1807
40	0.0564	2.8812	71.7950	10.2960	0.9218	156	10.8780
50	0.0551	3.0894	77.0410	11.8800	1.2015	196	12.7105
60	0.0562	3.1423	78.3900	14.1360	1.4750	236	13.0872
70	0.0548	3.2377	80.7970	15.6720	1.7633	276	14.2292
80	0.0547	3.2596	81.3290	17.6640	2.0453	316	14.5493
90	0.0546	3.3081	82.5840	19.2960	2.3283	356	15.2363
100	0.0542	3.3285	83.0810	20.8560	2.6133	396	15.7749

Table 2.11: Simulation results for Flooded Piggybacked ant routing (FP)

Simulation time(s)	Latency (s)	Throughput (Kbits/s)	Success rate (%)	Energy consumption(J)	Stand. dev.(J)	No. of packets	Energy efficiency(Kbits/J)
10	0.0357	4.0124	100.0000	5.3024	0.1704	36	6.7894
20	0.0342	4.0059	100.0000	10.9696	0.3663	76	6.9282
30	0.0338	4.004	100.0000	16.6352	0.5617	116	6.9732
40	0.0340	4.003	100.0000	22.3040	0.7572	156	6.9943
50	0.0335	4.0023	100.0000	27.9632	0.9528	196	7.0092
60	0.0334	4.0019	100.0000	33.6240	1.1486	236	7.0188
70	0.0336	4.0017	100.0000	39.2800	1.3441	276	7.0265
80	0.0335	4.0014	100.0000	44.9504	1.5403	316	7.0300
90	0.0333	4.0012	100.0000	50.6192	1.7358	356	7.0329
100	0.0333	4.0012	100.0000	56.3024	1.9320	396	7.0334

Table 2.12: Simulation results for Beesensor routing

Simulation time(s)	Latency (s)	Throughput (Kbits/s)	Success rate (%)	Energy consumption(J)	Stand. dev.(J)	No. of packets	Energy efficiency(Kbits/J)
10	0.0930	3.5723	88.8900	0.8128	0.1753	36	39.3706
20	0.0587	3.7450	93.0000	1.4688	0.3758	76	48.1209
30	0.0755	3.6819	93.1030	3.0016	0.5535	116	35.9806
40	0.0927	3.7219	92.9490	5.3984	0.7154	156	26.8599
50	0.1103	3.7172	93.3670	8.2272	0.8611	196	22.2432
60	0.1091	3.7483	93.6440	10.0832	1.0401	236	21.9176
70	0.1247	3.5221	88.4060	12.8320	1.1856	276	19.0150
80	0.1256	3.5969	89.8730	15.3440	1.3613	316	18.5088
90	0.1264	3.6415	91.0110	17.5040	1.5267	356	18.5100
100	0.1229	3.6379	90.9090	18.9696	1.7042	396	18.9777

Table 2.13: Simulation results for Basic Ant-Based Routing (BABR)

Simulation time(s)	Latency (s)	Throughput (Kbits/s)	Success rate (%)	Energy consumption(J)	Stand. dev.(J)	No. of packets	Energy efficiency(Kbits/J)
10	0.1533	0.5645	13.8890	2.6880	0.1386	36	1.8601
20	0.1201	1.6413	40.7890	6.2640	0.4636	76	4.9489
30	0.0897	2.3538	58.6210	8.4480	0.8130	116	8.0493
40	0.0822	2.6766	66.6670	10.7040	1.1664	156	9.7160
50	0.0787	2.8229	70.4080	12.7200	1.5383	196	10.8490
60	0.0760	2.9379	73.3050	14.8320	1.9328	236	11.6640
70	0.0739	3.0195	75.3620	16.9440	2.3129	276	12.2757
80	0.0724	3.0308	75.6330	18.7920	2.6882	316	12.7182
90	0.0716	3.0937	77.2470	20.7600	3.0646	356	13.2466
100	0.0704	3.1652	79.0400	22.6560	3.4390	396	13.8153

Table 2.14: Simulation results for Ad-hoc On-demand Distance Vector (AODV) routing

Simulation time(s)	Latency (s)	Throughput (Kbits/s)	Success rate (%)	Energy consumption(J)	Stand. dev.(J)	No. of packets	Energy efficiency(Kbits/J)
10	0.0352	1.6941	41.67	2.7205	0.5033	36	11.7627
20	0.0324	1.8032	44.74	5.3749	1.0558	76	13.1500
30	0.0325	1.7547	41.38	8.3765	1.6799	116	12.8932
40	0.0327	1.7501	43.59	11.3333	2.2951	156	12.7942
50	0.0331	1.6750	41.33	14.4245	2.9375	196	12.6867
60	0.0331	1.5971	39.83	17.3813	3.5525	236	12.7148
70	0.0331	1.5976	39.86	20.3717	4.1746	276	11.9774
80	0.0331	1.5471	38.61	23.8213	4.8918	316	11.9220
90	0.0331	1.5123	37.64	26.9909	5.5507	356	12.0040
100	0.3281	1.5265	38.01	29.4661	6.0657	396	12.2174

Table 2.15: Simulation results for Adaptive Spanning Tree (MCBR-AST)

Simulation time(s)	Latency (s)	Throughput (Kbits/s)	Success rate (%)	Energy consumption(J)	Stand. dev.(J)	No. of packets	Energy efficiency(Kbits/J)
10	0.0307	3.6958	91.67	2.4720	0.3547	36	12.9451
20	0.0308	3.6984	92.11	4.6800	0.7370	76	15.1026
30	0.3102	3.7682	93.97	6.6720	1.1319	116	16.1870
40	0.0309	3.8018	94.87	8.5680	1.5257	156	16.9235
50	0.0308	3.8218	95.41	10.4640	1.9196	196	17.4885
60	0.3082	3.7843	94.49	12.2880	2.2988	236	17.9850
70	0.0307	3.8011	94.93	14.1840	2.6928	276	17.2025
80	0.0308	3.8009	94.94	16.0560	3.0820	316	17.6880
90	0.0308	3.8233	95.51	17.9760	3.4810	356	18.0240
100	0.0308	3.8007	94.95	19.8000	3.8604	396	18.1818

Table 2.16: Simulation results for Real-Time Search Routing (MCBR-RTSR)

Simulation time(s)	Latency (s)	Throughput (Kbits/s)	Success rate (%)	Energy consumption(J)	Stand. dev.(J)	No. of packets	Energy efficiency(Kbits/J)
10	0.0305	3.8590	94.44	2.3760	0.3407	36	13.4682
20	0.0306	3.7747	93.42	4.3200	0.7264	76	16.3611
30	0.0306	3.8183	94.83	6.3360	1.1259	116	17.0454
40	0.0306	3.8392	95.51	8.4280	1.5219	156	17.2046
50	0.0306	3.8106	94.90	10.4160	1.9099	196	17.5691
60	0.0306	3.8430	95.76	12.4320	2.3097	236	17.7767
70	0.0307	3.8074	94.93	14.5440	2.6976	276	16.7767
80	0.0307	3.7939	94.62	16.6320	3.0931	316	17.0754
90	0.0307	3.8059	94.94	18.7440	3.4878	356	17.2855
100	0.0307	3.8255	95.46	20.7840	3.8929	396	17.3210

Table 2.17 Simulation results for Constrained Flooding Routing (MCBR-CFR)

Simulation time(s)	Latency (s)	Throughput (Kbits/s)	Success rate (%)	Energy consumption(J)	Stand. dev.(J)	No. of packets	Energy efficiency(Kbits/J)
10	0.2186	3.6816	91.67	7.2960	0.0695	36	4.3860
20	0.2187	3.8577	96.05	15.4800	0.0862	76	4.5659
30	0.2185	3.8374	95.69	23.1600	0.1147	116	4.6632
40	0.2177	3.8322	95.51	30.7920	0.1616	156	4.7090
50	0.2169	3.7993	94.90	38.2560	0.2130	196	4.7835
60	0.2149	3.8018	94.92	45.8160	0.2659	236	4.8236
70	0.2151	3.8131	95.29	53.6160	0.3145	276	4.5509
80	0.2152	3.8143	95.25	61.4160	0.3640	316	4.6242
90	0.2156	3.8236	95.51	69.3600	0.4031	356	4.6713
100	0.2142	3.8102	95.20	76.9680	0.4487	396	4.6773

Table 2.18: Comparison of the routing protocols based on different metrics

Routing protocols	Latency (S)	Success rate (%)	Energy consumption (J)	Energy efficiency (Kbits/J)	Standard deviation (J)	Lifetime (Days)
EEABR	0.0315	92.4240	16.6624	21.9656	2.6624	265.4862
SC	0.0313	68.6870	14.9488	18.1955	2.9782	265.3608
FF	0.0542	83.0810	20.8560	15.7749	2.6133	265.0694
FP	0.0333	100.0000	56.3024	7.0334	1.9320	261.8122
Beesensor	0.1229	90.9090	18.9696	18.9777	1.7042	266.1881
BABR	0.0704	79.0400	22.6560	13.8153	3.4390	264.0437
AODV	0.3281	38.0100	29.4661	12.2174	6.0657	260.6603
MCBR-AST	0.0308	94.9500	19.8000	18.1818	3.8604	263.9396
MCBR-RTSR	0.0307	95.4600	20.7840	17.3210	3.8929	263.7978
MCBR-CFR	0.2142	95.2000	76.9680	4.6773	0.4487	260.9993

Table 2.18 shows the comparison table summarized from the simulation results in Table 2.8 through 2.17. The result is an average of 10 simulation sets of each of the protocols. As can be seen from the table, EEABR is the most highest in terms of energy efficiency, though the SC consumes lesser energy, but its low performance in terms of energy efficiency is due to the low turn-up in packet delivery. Beesensor has the highest lifetime predication. Applications that require almost 100% packet delivery, without minding the energy consumption will prefer to work with FP which in most cases delivers almost all the packets generated in the network. Only a few of the classical routing protocols perform well in terms of energy efficiency, as it can be seen from the table. MCBR-CFR and AODV perform below expectation, while MCBR-AST and MCBR-RTSR performs quite well while also being QoS aware.

Table 2.19: Proposed models for routing protocol comparison in WSNs

Protocol characteristics		Routing protocols									
		EEABR	SC	FF	FP	Beesensor	BABR	AODV	MCBR-AST	MCBR-RTSR	MCBR-CFR
Analytical	Network Structure	Qos aware	Location	Qos aware	Qos aware	Qos aware	Qos aware	Qos aware	Qos aware	Qos aware	Qos aware
	Energy Efficiency	V. Strong	Strong	Weak	Weak	Very Strong	Weak	Weak	Strong	Moderate	Weak
	Data Aggregation	No	No	No	No	No	No	No	Yes	No	No
	Location Awareness	No	No	No	No	No	No	No	No	No	No
	Route Selection	Proactive	Hybrid	Hybrid	Hybrid	Reactive	Proactive	Reactive	Proactive	Proactive	Proactive
	Query Based	No	No	No	No	No	No	Yes	No	No	No
	Simulation Environment	NS-2	Rmase	Rmase	Rmase	Rmase	Rmase	Parsec	Rmase	Rmase	Rmase
Performance	Latency (s)	0.032	0.031	0.054	0.033	0.123	0.070	0.033	0.031	0.031	0.214
	Throughput (Kbits/s)	3.699	2.753	3.329	4.001	3.638	3.165	1.527	3.800	3.826	3.810
	Success Rate (%)	92.424	68.68	83.081	100.00	90.909	79.040	61.399	95.000	95.500	95.200
	Energy Consumption (J)	16.662	14.95	20.856	56.302	18.969	22.656	29.466	19.800	20.784	76.968
	Energy Efficiency(Kbits/J)	21.966	18.20	15.775	7.033	18.978	13.815	15.935	18.182	17.321	4.677
	Standard Deviation (J)	2.662	2.978	2.613	1.932	1.704	3.439	1.855	3.860	3.893	0.449

2.9 General discussion on the reviewed routing protocols in WSN

From the reviewed protocols, it is clearly seen that much effort has been made in addressing the methods to design reliable and efficient algorithms for WSNs. In this part of the chapter, we elaborate and commented on the results presented in Table 2.3 to 2.6, and Table 2.18 and 2.19. We also commented on methods related to the way by which the algorithms/protocols were actually evaluated, and the problems found in their presentation, and then build upon the results to provide some indications and steps about designing efficient routing protocols for WSNs.

The results from our analytical detailed presentations are reported in Table 2.3 through Table 2.6. The performances of the protocols for our experimental study are summarized in Table 2.18. The proposed model for routing protocol comparison is reported in Table 2.19. In general terms, from the survey and the report presented in Table 2.3 through Table 2.6, there is an existence on some major drawbacks which are common to most of the algorithms/protocols surveyed in this chapter. Some of the drawbacks include the inability of most of the algorithms to be evaluated over a large network. The simulation environments and the parameters used for simulation experiments are not satisfactorily described, and there is no comparison with some prominent existing algorithms. As such, most of the work reported by some promising algorithms is not sufficient and also lack scientific soundness. The most pronounced drawbacks from the theoretical point of view of most of the routing protocols are:

1. Many protocols end up in self-analysis and comparison, whereby no comparison is made with any standard algorithm design for WSNs.
2. Simulation environment, performance metrics and experimental parameters used for simulation and experimental setup of some promising protocols were not defined which hinders repeatability.

3. Protocols were not evaluated over a large network, and hence, most of the protocols are good on scanty networks were there exists fewer nodes.
4. Some protocols which are promising and very good at energy efficiency, lack quality of service awareness and as such, experience excessive delay in packet delivery which may not be so good in some applications such as; security applications and environmental monitoring.
5. Experimental results reported by some promising algorithms do not have scientific soundness and looks quite weak. And performance comparison with other algorithms, does not present any sound statistical hypothesis tests to carry out comparison so as to test their performance as against other promising algorithms, because it is only a few of the authors that actually does the real comparison.
6. Finally, it is clearly seen from the survey that only one of the simulation environments among the surveyed has some WSNs protocols already patched into it. While some are not even available for others to test their algorithms, and are only available for commercial purpose. We encourage protocols designers to patch their algorithms in whichever environments their experiment was performed to aid in comparison with the new algorithms.

However, from our reports, since energy utilization efficiency and reliability are the major factors for the evaluating an efficient routing protocol for WSN, Table 2.19 presents the combination of the performance metrics and analytical characteristic of the protocols. As can be seen in the table, FP has the highest success rate leading to high Throughput of value 4.001 Kbits/s, but with low energy efficiency due to high energy consumption. EEABR has the highest energy utilization efficiency (21.966 Kbits/J) as against FP (7.033 Kbits/J). But in the real application of target tracking scenario (event based), Beesensor will be preferable since it

is an on-demand (reactive) routing protocol. Moreover, it was the next best in performance in term of energy utilization efficiency. This choice of Beesensor is due to its high data delivery rate per energy consumption and the same time, it will generate fewer overheads in a dynamic network. Besides, its network lifetime is longer as compared to other since the average variation between the energy levels of all the nodes in its network (standard deviation) is lower. The only problem it will encounter is that it will have a slightly higher delay in packets or event delivery as can be seen in Table 2.19. Its end-to-end delay (latency) is 0.123 seconds as against 0.032 of EEABR. But AODV will also do well when it comes to an application that has to do with querying each node for information (query based). Even with its relatively high energy consumption, it has a very low standard deviation and low latency, hence will be better when it comes to deliver events of high importance and at a faster rate as compared to Beesensor. Finally, it then means that, AODV for query based applications, EEABR for periodic based application due to its high energy efficient and proactive routing scheme, while Beesensor will be best in event based applications due to its high throughput and on-demand (reactive) routing scheme.

2.10 Conclusions and future direction

From the review protocols it is clearly seen that, much effort has been made in addressing the methods to design effective, and efficient routing protocols for WSNs. The results of our Analytical comparison are reported in Tables 2.3 to 2.6, while that of the experimental comparison is shown in Table 2.18. There exist some drawbacks in the presentation of most of the routing protocols for comparison purpose for most of the routing in WSNs, apart from the problems mentioned above, it is clearly seen that many surveyed algorithms end up in either mathematical modeling or simulation.

Also, a lot of them do not compare their protocols with any of the standard ones. Simulation based studies should be supported by mathematical analysis, which will then allow for studying a very large system, and also favors fair comparison among protocols. Towards this end, we present Table 2.19 as a proposed standard table for routing protocols comparison. Due to the requirements highlighted in Section 2.9, we then presented our newly designed algorithms in Chapter 3 and Chapter 4 to help in dealing with some of the itemized problems and the constraints that characterize WSNs.

Chapter 3

Improved Energy-Efficient Ant Based Routing Algorithm

3.1 Introduction

The main problem for event gathering in wireless sensor networks (WSNs) is the restricted communication range for each node. Due to the restricted communication range and high network density, event forwarding in WSNs is very challenging, and requires multi-hop data forwarding. Currently, the Energy Efficient Ant Based Routing (EEABR) Algorithm, based on the ant colony optimization (ACO) metaheuristic is one of the state-of-the-art energy aware routing protocols. In this chapter, we propose several improvements to the EEABR algorithm to further improve its energy efficiency and reliability. The improvements to the original EEABR are based on: (1) a new scheme to intelligently initialize the routing tables, giving priority to neighboring nodes that could simultaneously be the destination, (2) intelligent update of routing tables in case of a node or link failure, and (3) reducing the flooding ability of ants for congestion control. Experimental results using the network simulator (NS-2), the original environment of which EEABR was adopted for experimental purpose, show that our proposed method outperforms EEABR and other ant and bee-based routing protocols. Further experiments using RMASE simulation environment show that the proposed method increases energy efficiency. The energy efficiency improvements are significant particularly for dynamic routing environments. The experimental results using the RMASE simulation environment show that the proposed method increases energy

efficiency by up to 9% and 64% in converge-cast and target-tracking scenarios respectively over the original EEABR without incurring a significant increase in complexity. The method is also compared to and found to outperform other swarm-based routing protocols such as Sensor-driven and Cost-aware ant routing (SC), Flooded Forward ant routing (FF), Flooded Piggyback ant routing (FP), Basic Ant-Based Routing (BABR), and Beesensor.

3.2 Related work

The idea of using a swarm paradigm to establish routes in communication networks is not new. In (Dorigo and Caro, 1998), an ant-based algorithm was adopted to calculate the optimal paths among the nodes through an architecture called AntNet. Smaller agents, the virtual ants, migrate from a node to another, building the routing rules in a distributed way.

In SC (Zhang et al., 2004b), it is assumed that ants have sensors so that they can smell where there is food at the beginning of the routing process so as to increase the possibility of sensing the best direction that the ant will go initially. In addition to the sensing ability, each node stores the probability distribution and the estimates of the cost of destination from each of its neighbors. But it suffers from misleading when there is an obstacle which might cause errors in sensing the best direction. In their extended work, Flooded Forward ant routing (FF) Zhang et al. (2004b) argued the fact that ants even augmented with sensors, can be misguided due to the obstacles or moving destinations. The protocol is based on the flooding of ants from source node to the sink node. In the case where the destination is not known at the beginning by the ants, or cost cannot be estimated, the protocol simply uses the broadcast method of sensor networks so as to route packets to the destination. Probabilities are updated in the same way as the basic ant routing, though, FF reduces the

flooding ants when a shorter path is traversed. In a nutshell, the authors were more concern on initial pheromone distribution, which is good at the beginning of the routing process, but bad when the system density is high.

The EEABR algorithm as proposed in (Camilo et al., 2006), is an improved version of the ant based routing in WSN which does not only consider the nodes in terms of distance, but also in terms of energy level of the path traversed by the ants. The authors in their work, pointed out that, in the basic ant algorithm, the forward ants were sent to an unknown location, which is to say that, they must interact and cooperate with each other. So also, their routing tables must contain the ID of each node in their neighborhood and the pheromone contents of the neighbor nodes. In their work, much achievement was recorded in energy savings, but encounter difficulty in the mobility and dynamic scenario since much control traffic are generated hence consuming much energy with less reliability.

Beesensor (Saleem et al., 2012) is a reactive and on demand routing protocol, which mode of operation is based on foraging principles of honey bees. The algorithm has three types of agents which makes it function well in its mode of operation. The agents are named packers, scouts and foragers. Packers search for foragers for information at the source nodes, whereas scouts takes the responsibility of route discovery to the sink node, but the foragers do the main work whose function is to hop from node to node until they get to the sink node. They hop along with data packets carried by them. Its analysis was done and compared with several routing protocols in RMASE simulator. Since it is an on demand based protocol, it suffers in the area of security application or an application where information needs updating at regular intervals of time. The routing process is reactive, the time use to search for the sink in a dynamic scenario of the routing process is high, and as such the algorithm is only

good for static applications. For more explanation of the algorithm operations discussed above, readers are referred to Chapter 2 so that we can avoid the repetition of words.

Besides all the drawback of each of the related protocols, almost all the algorithms tend to sacrifice the network performance as against the improvement of energy consumption of the nodes, and vice-versa for others.

3.3 Application scenarios as it affect routing in WSNs

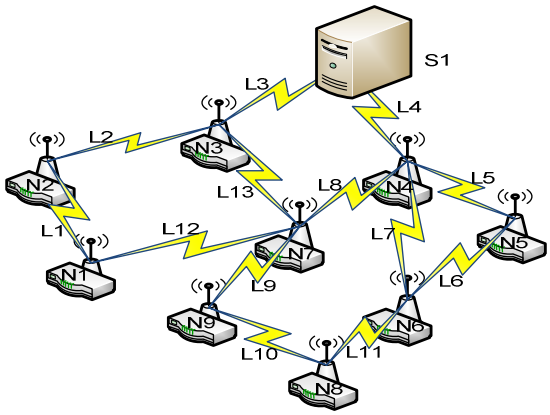
In this section, we try to highlight the problem of topology change rate with respect to routing issues in WSN. In sensor networks, each sensor generates data packets at a fixed data rate. If a sensor node *j* is neither co-located with sink *s* nor directly connected with it, then data packets generated at node *j* will have to be relayed through multiple hops to reach the sink. In a static sink scenario, the sink node is always in a fixed position. All the traffic destined to it must pass through the nodes closer to it, which will make them to deplete in their energy resource faster. As seen in Fig. 3.1(a), node 3 (N3) and node 4 (N4), are the closest and in the transmission range of the sink. Besides transmitting their own information, they also transmit information belonging to other nodes, which will make them to deplete in their energy faster than other nodes. At time t, when their energy will get exhausted, there will be holes created in their spot as such, since other nodes are not in the transmission range of the sink, the information will no longer get to the sink, hence making the network useless. Sink mobility is needed due to this problem of energy-holes and hot spots. The mobile sink will be able to collect information from other nodes even when node 3, and node 4 are no longer active. It will also help in balancing the network energy due to its mobility, as nodes closer to it will keep on changing as can be seen in Fig. 3.1(b), and (c). It should also

be observed that as the sink moves, some of the links to it have to be broken due to communication range as can be seen with the communication links crossed with a bar. With this, sink mobility will not only improve on the network lifetime, but helps in collecting the other useful information that could have been lost when node 3 and node 4 were no longer active. Though a typical routing protocol that involves sink mobility will do the following:

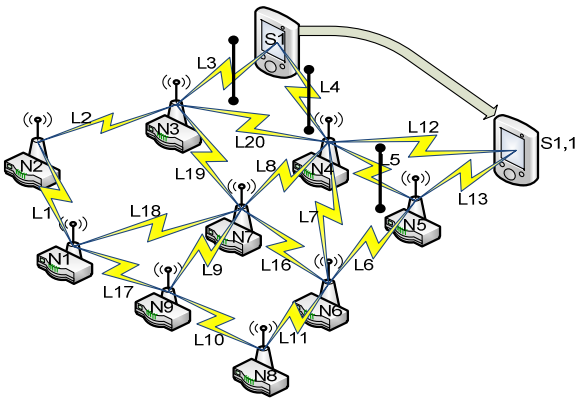
a) Inform the whole network of the topological changes during the sink mobility,

b) Notify a node when its link with the sink gets broken due to sink mobility.

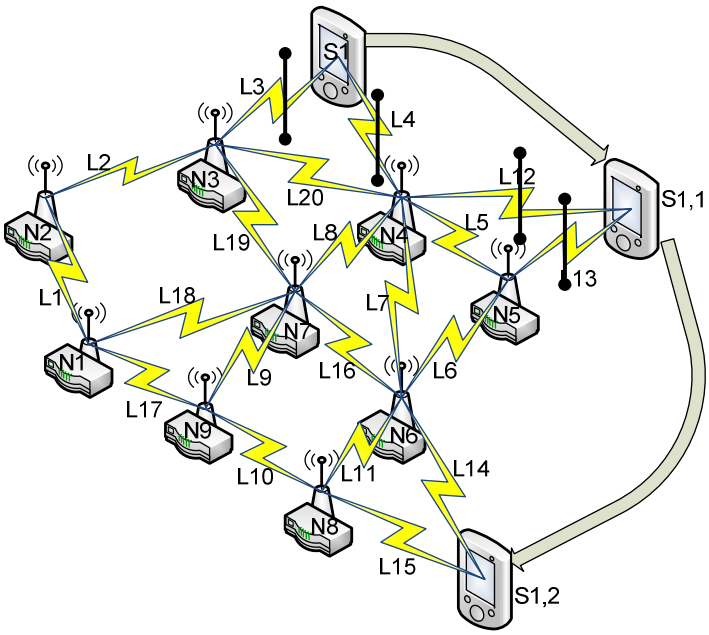
This leads us to the social insect way of solving real world problems which are reviewed in the section 3.4.



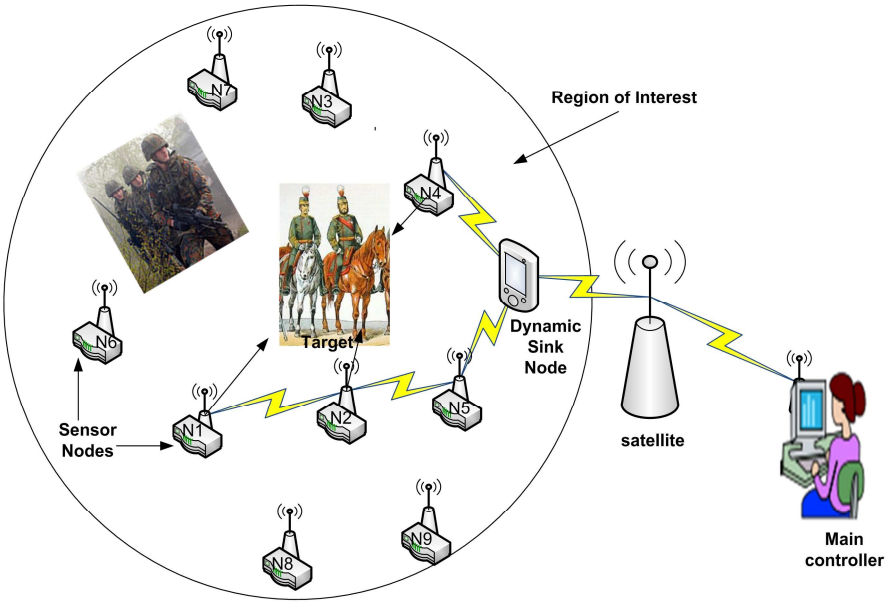
(a)



(b)



(c)



(d)

Fig. 3.1: Application scenario of WSN (a) sink in a static position, (b) mobility of sink (S1→S1, 1), (c) mobility of sink (S1→S1, 1→S1, 2), (d) Dynamic Sink (Target Tracking)

Target tracking in WSNs is a process of estimating the location, trajectory, velocity and/or acceleration of a mobile target. It often needs accurate estimation and prediction of the target state. In a typical target tracking (Dynamic Scenario) application in WSN, the targeted event is mobile. Hence, the source nodes or sink node has to change its location at every interval of time to cope with the dynamic nature of the target event. Fig. 3.1(d) shows a target tracking scenario in a typical WSN.

In this scenario, the sink is responsible for forwarding the desired information from the WSN to the headquarters (i.e., main controller) through the Internet, via satellite or other wireless technology. The target can be a human being, moving vehicle, animal, tank, enemy or any interesting object that needs to be tracked which is usually mobile. Targets can move in an unexpected manner and this causes the loss of target sometimes. Locating the position of the target at any point in time is one of the main challenges for target tracking in WSNs. Each sensor node has a sensor device to sense or detect the presence of the target in the Region of Interest (ROI). Detection of the target is always handicapped when it is out of transmission range of the nodes. Therefore, using a mobile sink or nodes can lessen the burden of the loss of the target, which will also in turn reduce energy consumption by the network nodes since less control traffic will be used to locate the position of the target. Due to the limited battery-supplied energy of the sensor nodes and the difficulty to physically access them, energy-efficiency target tracking is a crucial aim. Additionally, hundreds or thousands of sensor nodes are deployed in the ROI. But with an energy efficient routing algorithm and sink in dynamic scenario, the position of the target can easily be tracked and fewer hops will be needed by the sensor nodes to get the information across the sink node.

3.4 The social insect analogy

The search for food in ants is organized in part by chemical trails laid while searching for food source. During foraging for food, ants communicate with the aid of the pheromone laid on their way back to the nest. When food is discovered, ants return to the nest laying a trail to recruit nest mates to the food source. The difference between foraging and recruitment trails is attributed to different quantities of trail pheromone present on the path. Ants have been found to adapt to their environment and always find the most efficient path to their food source (Dorigo and Caro, 1998). The optimization of path behavior of ant is now widely used to motivate applications such as routing in WSN. Considering Fig. 3.3, the path from the nest to the food source at time $t = 0$, the ants find the food and bring it back efficiently, establishing a pheromone trail to it. At time $t = 1$, when there is an obstacle in their path such that there is one path that is shorter than the other, the ants can choose either path with equal probability, hence having the same number of Ants on both sides. The path that is shorter will allow ants to gather food quickly and strengthen the pheromone trail on the way back faster than the ants on the longer path as seen when $t = 2$, causing the other batch of ants to move with higher probability towards the trail that is stronger. As the process continues till time $t = n$, it will be observed that all the ants will use the shortest path towards the food source. The obstacle in this analogy can be congestion, number of nodes on the path to the sink, latency (due to some other factors) etc.

The following subsections explain the behavioral pattern of foraging of ants toward food sources.

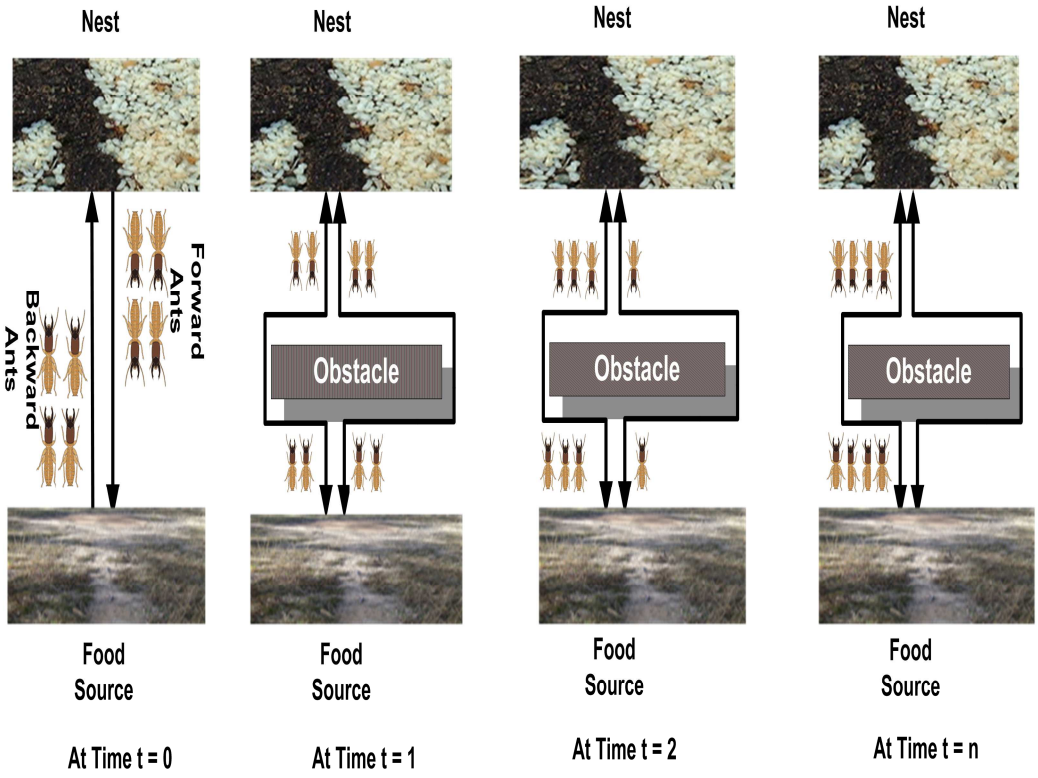


Fig. 3.3: Food Search in Ants.

3.4.1 Positive Feedback

In ants foraging, an ant's attraction along the pheromone site motivate it in adding to where there exist much food piles. If there are many food piles in a place, it is expected that much pheromone will be present in that food pile, that is to say that, more ants will visit the place due to its pheromone concentration, and as such the ants will add more pheromone to the pile on that path. The greater the biases towards the food source, more ants are also likely to take the path to that food source, which means that pheromone content of the path will be increased.

3.4.2 Negative Feedback

Negative feedback is accomplished by pheromone evaporation. This happens so as to avoid premature convergence among ants (stagnation). For good communication among the ant individuals, pheromone must evaporate over the environment. The evaporation helps to weaken the pheromone, which will bring down the amount of pheromone on that path.

The path with lower pheromone concentration will have fewer ants as it will attract fewer ants towards that direction. Though this may seem contrary to the task of collecting all food to the nest, but it is important. Negative feedback is entirely useful in the removal of past or poor solutions for the memory of constituent of any network or system.

3.4.3 Randomness

The location and path taken by ants towards the food source is determined by chance. A little drift in the behavior of ants will lead to significant effect on the future behavior of the system. Randomness is useful so that new solutions can be built, since the network and system under consideration is dynamic.

3.4.4 Multiple Interactions

In the food collection of ants to their nest, it is a necessity that many individuals cooperate and work together to achieve their target. This is in accordance with neighboring nodes of a sensor network acting as routers to other source nodes. If not enough ants exist in a nest, then the pheromone would decay before any more food could be collected at the nest. Also, if we map this to a sensor network, if there are not many nodes on the path to the sink node, more packets will be dropped on the way to the sink. This might also be as a result of the low transmission distance (range) of the nodes too. But if there exist more ants in the environment, the more food will be gathered fast to avoid complete pheromone decay in the shortest path, else ants would continue their random walk without building any strong solution as regards the best paths.

3.4.5 Stigmergy

This is the indirect communications between individuals of the social insect, generally through their environment. Complexity in stigmergic systems is due to the fact that, individuals or constituents of a system do not interact

within themselves rather they do so with their environment. This behavior or actions lead to changes in the environment where they interact. Due to the changes, there is an advance effect on further behaviors, which give rise to a positive feedback effect where events are dependent on other events. This behavior is similar to the interaction between people in their environment based on their response to other people's comments during conversations. Ants are directed to the path with a high pheromone gradient, it is not necessary for ants to directly interact with themselves or to know the whereabouts of other ants. As such, ants are allowed to cooperate and interact with one another, which is the main issue behind stigmergy.

3.5 Improved Energy Efficient Ant-Based Routing algorithm (IEEABR)

WSNs protocols are designed to meet some important constraints like quality of service and improvement of the network lifetime. It is then important for each protocol designed for WSN to pay more attention to the energy utilization and the reliability of the algorithm under consideration. This is necessary, since the network under consideration have limited power supply, constrained memory capacity, low processing capability, and constraint available bandwidth. To this end, we proposed some important improvements on an EEABR algorithm.

3.5.1 Energy-Efficient Ant-Based Routing Algorithm (EEABR)

EEABR algorithm proposed by Camilo et al. (2006), is an improved version of the basic Ant based routing for WSN. The author's main idea is to minimize the communication load in the network that will be as a result of the control packets, and also minimize energy consumption used in the communication process. The basic algorithm of EEABR (Camilo et al., 2006) is outlined as below:

1. At any point in time from every network node, a forward ant is launched which mission is to locate a good route to the end node (sink). The ID of the nodes visited in the process is saved onto a memory *M* which is being carried by the ant. Where *k* is any network node having a routing table will have *N* entries, one for each possible destination, and *d* is one entry of the *k* routing table (a possible destination).
2. For each source node, a forward ant will choose the next hop node with the aid of the probability selection procedure of the basic ACO metaheuristic as:

$$P_k(r,s) = \begin{cases} \frac{[\tau(r,s)]^{\alpha}.[E(s)]^{\beta}}{\sum_{u \notin M}[\tau(r,u)]^{\alpha}.[E(s)]^{\beta}} & ,s \notin M \\ 0 & otherwise \end{cases} \quad (3.1)$$

where $P_k(r, s)$ represents the probability that ant *k* use in the selection process from node *r* to node *s*, τ is the routing table at each node that stores the amount of pheromone trail on connection (r, s), *E(s)* is the visibility function given by $\frac{1}{(C_s - e_s)}$ (C_s is the initial energy level of the node *s* and e_s is the actual energy level of node *s*), and α and β are parameters that control the relative importance of trail versus visibility. The selection probability is a trade-off between visibility (which says that nodes with more energy should be chosen with high probability) and actual trail intensity (that says that if in connection (r, s) there has been a lot of traffic then it is highly desirable to use that connection).
3. At each point in time that a forward ant arrived at the end node (sink), it is converted to a backward ant which mission is now to update the pheromone trail of the path that was used by the forward ant in getting to the sink node, and the information is stored in its memory.

4. Prior to the return of backward ant *k* to the source node using the path traversed by the forward ant, the sink node computes the amount of pheromone trail that the ant will drop during this return journey as:

$$\Delta\tau = \frac{1}{C_{av} - \left[\frac{E_{min} - N_j}{E_{av} - N_j}\right]} \tag{3.2}$$

Where C_{av} is the initial average energy of the nodes, E_{min}, E_{av} are the minimum and average energy respectively of the path traversed by the forward ant as it moves towards the sink. The values of E_{min} and E_{av}, depends on the number of nodes on the path and the energy consumed by the nodes on the path during transmission and reception of packets. The minimum energy of the path (E_{min}) can be less than the number of nodes visited by the forward soldier, but the average energy of the path (E_{av}) can never be less than the number of visited nodes. N_j represent the number of nodes that the forward ant has visited. The idea behind the calculation of $\Delta\tau$ is that, it brings optimized routes, since it is a function of the energy level of the path, as well as the length of the path. For example, a path with 10 nodes can have the same average energy as path with 4 nodes. Therefore, it is important to calculate the pheromone trail as a function of energy and the number of nodes as against the number of nodes as it used in other ACO.

5. Whenever a node *r* receives a backward ant coming from a neighboring node *s*, it updates its routing table in the following order:

$$\tau(r,s) = (1-\rho) * \tau(r,s) + \left[\frac{\Delta\tau}{\emptyset B d_k}\right] \tag{3.3}$$

Where ϕ a coefficient and B_{dk} is the distance travelled (the number of visited nodes) by the backward ant *k* until node *r*, which the two parameters will force the ant to lose part of the pheromone strength during its way to the source node. ρ, is a coefficient and (1- ρ) denotes the evaporation of pheromone contents since the last time $\tau(r,s)$ was

updated. The idea behind the behavior is to build a better pheromone distribution (nodes near the sink node will have more pheromone levels) and will force remote nodes to find better paths. Such behavior is important when the sink node is able to move, since pheromone adaptation will be much quicker.

6. When the backward ant reaches the node where it was created, its mission is complete and the ant is eliminated. This action is performed several times to arrive at optimal solutions.

3.5.2 Improvements on Energy-Efficient Ant-Based Routing Algorithm

The improved version of EEABR algorithm considers the available power of nodes and the energy consumption of each path as the reliance of routing selection. It improves on memory usage, utilizes the self-organization, self-adaptive and dynamic optimization capability of an ant colony system to find the optimal path and multiple candidate paths from source nodes to sink nodes. The algorithm avoids using up the energy of nodes on the optimal path and prolongs the network lifetime while preserving network connectivity. This is necessary since for any WSN protocol design, the important issue is the energy efficiency of the underlying algorithm due to the fact that the network under investigation has strict power requirements. As proposed in (Kalpakis et al., 2003) and adopted in (Camilo et al., 2006), for forward ants sent directly to the sink-node, the routing tables only need to save the neighbor nodes that are in the direction of the sink-node, which considerably reduces the size of the routing tables and, in consequence, the memory needed by the nodes. The memory M of each ant is reduced to just two records, the last two visited nodes. Since the path followed by the ants is no more in their memories, a memory must be created at each node that keeps record of each ant that was received and sent. Each memory record saves the previous node, the

forward node, the ant identification and a timeout value. Whenever a forward ant is received at any node, it searches for any possible loop with the aid of its identification (ID). For the situation where no record is found, the necessary information is retrieved and the timer is restarted, hence forwarding the ant to the next node, else, the ant is eliminated if a record containing the ant identification is found. When a backward ant is received, the source ID is searched so as to know where to send it to. In this section, we proposed some modifications on EEABR to improve the Energy consumption in the nodes of WSNs and also in turn improve the performance. The improvements are based on a new scheme to intelligently initialize the routing tables, giving priority to neighboring nodes that simultaneously could be the destination, intelligent update of routing tables when there is a node or link failure, and reducing the flooding ability of ants for congestion control. The sub-sections below describe the improvements.

3.5.2.1 Intelligent Initialization of Routing Tables

The improved ant based routing termed EEABR has no method of initializing the routing tables. As such, we proposed an initial uniform distribution of probabilities in the routing table. Due to the situation of no a-prior knowledge about the network topology, the proposal leads to the reflection of an initial knowledge about the network topology as the routing process progresses. The initialization of the routing table is done with a uniform probability distribution as;

$$P_{id} = \frac{1}{N_k} \tag{3.4}$$

and P_{id} represents the probability of hopping from node *i* to node *d* (sink), N_k is the set of neighboring nodes of node *k*. This is done to reflect the previous (initial) knowledge about the network topology as the routing process progresses.

At a given time after network topology update, a higher probability value is given to the neighboring nodes which serve simultaneously as the sink node (destination) according to (3.5). With this method, the network energy consumption is saved. This is because it is possible to arrive at the sink node using just one link. For the situation whereby the sink node *d* is also a neighbor of a source node, that is $d \in N_k$, we can then initialize the probability with which the ant *k* chooses to hop to the sink node *d* using the following:

$$P_{dk} = \frac{9N_k - 5}{4N_k^2} \tag{3.5}$$

also, for the other neighboring nodes among the neighbors for which $i \neq d$, and $i \in \mathrm{N_k}$ will then have their probability of selection in the routing table of *k* as:

$$P_{ik} = \begin{cases} \frac{4N_k - 5}{4N_k^2}, & if\ N_k > 1 \\ 0, & Otherwise \end{cases} \tag{3.6}$$

N_k being a set of k which are its neighbors, P_{ik} represents the probability with which an ant at node *k*, hops to a node *i*, $i \in N_k$, whereby the sink node is *d* ($d \neq k$). Then, for each of the *N* entries in the node *k* routing table, it will be N_k values of P_{id} subject to the condition:

$$\sum_{i \in N_k} P_{id} = 1;\ \ d = 1, \ldots, N \tag{3.7}$$

For example, if a source node has five neighbors, the probability of selecting a neighbor node which in turn serves as the destination according to previous knowledge of the route update, is

$P_{dk} = \frac{9(5)-5}{4(5^2)} = \frac{2}{5}$, and the other neighbors will be selected based on the probability $P_{ik} = \frac{4(5)-5}{4(5^2)} = \frac{3}{20}$ each. That is to say that, the sum of probabilities of selecting all the five neighbors of the source node will be $P_{id} = \frac{2}{5} + \frac{3}{20} + \frac{3}{20} + \frac{3}{20} + \frac{3}{20} = 1$.

Of course we can see that equation (3.5) and (3.6) satisfy (3.7), (note: probability distribution table is maintained by the source nodes only). But for a source node whose neighbor is one, and the neighbor falls to be the destination, according to equation (3.5) and (3.6) $P_{dk} = 1, and\ P_{ik} = 0$. This also satisfies equation (3.7).

3.5.2.2 Reduction of the Network Control Packets

In the improved ant based routing termed EEABR, there is no specification on the number of ants hopping in the network, which in some applications on sensor network, could lead to network congestion. In order to control the number of ants due to the dynamic nature of the sensor network, the total number of ants launched at each node was limited to just five times the number of network nodes (*5 x N*), this is because for a given node, the average number of links to it on the networks used as seen in Fig. 3.1 is five (5). With this method, simulation results were improved.

3.5.2.3 Self destruction of Control Packets

There are situation when the forward ant sent from the source nodes do not eventually return back to the source node following the reverse link. In that case, in order to avoid infinite loops when a loop occur, we introduced ant self destruction. This method is introduced due to the fact that when the forward ant $F_{s\rightarrow d}$ has performed more than half of the total number of hops in a cycle, it should be destroyed. This case is also applied to the backward ant $B_{s\leftarrow d}$, and should in case if the backward ant cannot return back to the source nodes where the forward ants was created. The action is employed so that false information about the network carried by the ant which was trapped in a cycle is not used in the updating of the nodes routing tables. This method helps to reduce wrong update of information in the routing table. Hence, better results were obtained from simulation results.

3.5.2.4 Intelligent Update after Network Resources Failures

The original EEABR do not deal with situations of network resource failures. In the case of link failure, an automatic update is made on the routing tables in case of a node *k* loses its link l_{mi} which is the link between it and its neighbor node *i* when it is at node *m*. It is assumed that if an ant is at a node *m*, the probability of hopping of the node through node *i* to get to the sink node *d* labeled P_{id} is uniformly distributed among the remaining N_k-*1* neighbors for the entry *d* in the routing table of node *k*, where $P_{id} = 0$, this occurs during a link l_{mi} failure. Hence, it is impossible to move from *m* to *i* for arrival to *d*. Hence, new probability values after link l_{mi} failure is introduced as P_{id}, and the probability relate to their values just before the failure. This is to avoid the losing of information gathered before the incident of the failure and is updated as:

$$P'_{id} = P_{id} * (1 + z) \qquad i \neq m, and\ i, m \in N_k \tag{3.8}$$

Where,

$$z = \frac{P_{id}}{1 - P_{id}} \tag{3.9}$$

Where $P_{id} \leq 0.5$. The probability of link to fail is normally less than 0.5 due to the fact that, the link fails either due to the hotspot as a result of low energy, or longer path of which the TTL of the ant elapsed. As such, it is not possible to have a probability of selecting that link greater than the summation of probabilities of other links. With this method, the previous knowledge about the network traffic and topology before the failure is kept intact. With these improvements, the network converges faster, and better results were achieved. The step-by-step description of the algorithm is shown in the algorithm pseudo-code in algorithm 3.1. In the algorithm pseudo-code, we break down the prototype of procedures into different stages, and in line 6 to line 24, we describe the procedures of initializing the global variables. In that, we define most of the important variables and

parameters. Line 26 to 43, is a decision making procedure, where decision to calculate pheromone, construction of backward ant, next hop selection, and elimination of backward ant is decided upon. From line 44 to 74, we describe the selection for the next hop selection, which is in the decision of line 29. In this step, the calculation of the probability of selection is done, and the decision for the elimination of forward ant in the event of occurrence of a loop is also taken. Thereafter, the procedure of updating of the source probability distribution table is given in line 75 to 91. From line 92 are the procedures for calculating the pheromone of paths, which is in the first decision of line 26. With this description, all procedures of pheromone calculation, pheromone updating, next-hop selection, is achieved, and better routing procedure is also achieved. The flow chart describing the action of movement of the forward ant for our proposed algorithm is as shown in Fig. 3.4.

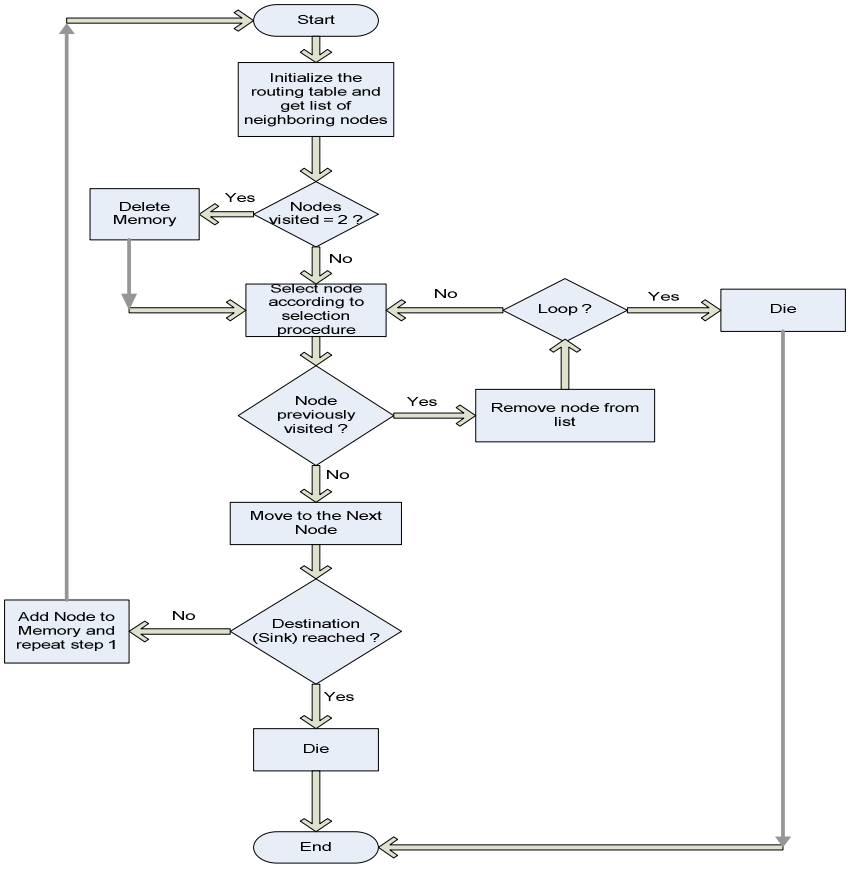


Fig. 3.4: An IEEABR forward ant flow chart.

3.5.2.5 Algorithm Operations

After the initialization of the routing table, and setting up a forward ant for hopping from node to node in the search for the sink, at every point in time, a node becomes a source holding in its stack or memory information about an event around it (neighbors). The information gathered in its memory is transferred or disseminated towards the sink node with the help of neighbor nodes behaving as repeaters. Associated raw data generated at each source (nodes) is divided into M pieces known as data parts. An integer value M also represents the number of ant agents involving in each routing task. This raw data provided by the source node about an event contains information such as source node identification, event identification, time and the data about the event. The data size is chosen based on the sensor nodes deployed and the size of the buffer. After the splitting of the raw data, each part is associated with routing parameters to build a data packet ready to transfer. These parameters are coded identification, describing the code following as data, error or acknowledge; C_{ID}. Next is the node identification to which the packet is transferred; N_{ID}. Packet number also represents the ant agent k; S_N is the sequence number, N_k which contains the number of visited nodes so far, and the k^{th} data part as shown in Fig. 3.5. In this figure, the group of the first four fields is named the data header. When delivery of all data packages is accomplished, the base combines them into raw data.

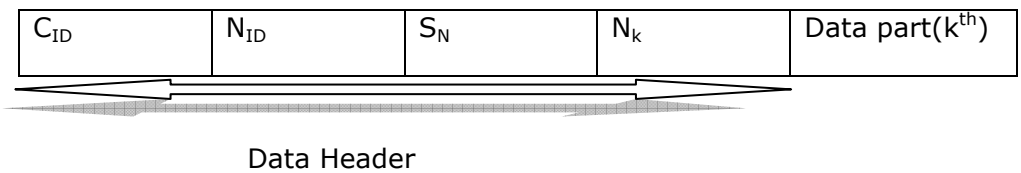


Fig. 3.5: Data packet content.

When a node participating in a routing received a data packet whose agent number is given, it makes decisions about the next destination for that

packet of data. The decision on the next node or destination for which the packet data should be transferred, will depend on the equation (3.1) with the highest $P_k(r,s)$. The pheromone level of the neighbors which is the first determining factor follows by the Energy levels of the neighbor nodes which are most important on the decision rule. For any of the neighbor chosen, the N_{ID} field of the node is updated and the packet is then broadcasted. The remaining neighbors among the chosen node also hear the broadcast, they check the N_{ID} field and understand that the message is not made for them; then as such quickly discard the packet immediately after only listening to the N_{ID} field of the packet. The N_k is updated with an increment by one after ensuring that S_N is not in the list of the routing table of the chosen node. The next node is determined to update the N_{ID} field by performing the same operation as perform earlier by the first node and the sequence continue till the packet gets to the sink node. The reversed operation is done for the backward ant as for the acknowledgement, which get to the source now the last bus stop for the backward ant and die off after reaching the source.

As an extension of this work, we also describe the pheromone table contents of a node in the network during the routing process. Below is a description of the pheromone table.

3.5.2.6 The Pheromone Table

This table keeps track of the information gathered by the forward ant. A node maintains this table which keeps the amount of pheromone on each path leading to their neighbors. The node has a peculiar pheromone scent different from each other, and the table is in the form of a matrix with destination nodes listed along the side and neighbor nodes listed across the top. Rows correspond to destinations and columns to neighbors. An entry in the pheromone table is referenced by $T_{n,d}$ where *n* is the neighbor index

and *d* denotes the destination index. The pheromone table has a value, which are used to calculate the probabilities of selecting each neighbor. When a packet arrives at node *G* from previous hop *S*, i.e. the source, the source pheromone decay, and pheromone is added to the link $\overrightarrow{SG}$. Backward ant on their way back from the sink node is more likely to take through *G,* since it is the shortest path towards the sink node i.e. $\overrightarrow{SGED}$. The pheromone table of node *G* is shown in Fig. 3.6 with nodes A, S, F, and E as its neighbor, while node A, node B, node C, node D, node E, node F and S are the possible destinations. It is worth noting that all neighbors are potential destinations in the route selection process of routing. At node G, the total probability of selecting links $\overrightarrow{ED}$, $\overrightarrow{FE}$, $\overrightarrow{AC}$ or $\overrightarrow{SB}$ to the destination node is equal to unity (1) i.e. $\sum T_{ED} + T_{SD} + T_{AD} + T_{FD} = 1$. It will then be observed that, since link ED is shorter, more pheromone will be present on it and hence, ants are more likely to take that path.

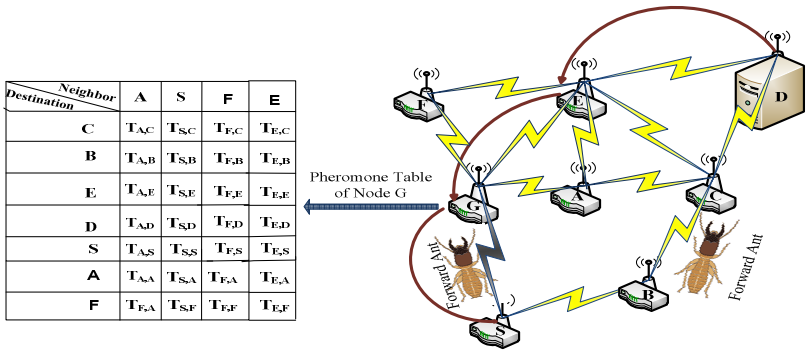


Fig. 3.6: Description of pheromone table of node G.

The pseudo code describing the algorithm description of our proposed methods is given in Algorithm Pseudo Code 3.1.

Pseudo Code 3.1: IEEABR Algorithm Pseudocode.

```
1. // IEEABR Algorithm Pseudocode
2. // Prototype of Procedures
3. Next-Hop-Selection ;
```

4. Calculate-Pheromone;
5. //Initialize the global variable
6. S = Source ID;
7. //Initialize probability distribution table
8. N= Number of nodes in network;
9. N_k = the set of neighboring nodes of node *k*;
10. n = N_k;
11. dp = probability distribution;
12. S_{PDT} [n][dp] = Source probability distribution table;
13. $P_{id}= \frac{1}{N_k}$; // P_{id} is the probability of jumping from node *i* to node *d*
14. // Initiate the routing table
15. For (i=0 ; i=n ; i++){
16. [dp].S_{PDT} = P_{id} ;
17. }
18. D = Destination ID (sink ID);
19. FA [S, M[2], D] = forward ant [source ID , memory of forward ant, Destination ID];
20. Ph = amount of Pheromone;
21. C = initial energy;
22. BA [S, M[2], Ph , D] = Backward Ant [Source ID, Memory of backward ant, Pheromone value, Destination ID];
23. E[] = visibility array;
24. r = Intermediate node ID;
25. // If the intermediate node is equal to the destination node, then calculate the pheromone and construct the backward ant
26. L1: If (r = d)
27. Calculate-Pheromone;
28. Construct the BA [S, M[2], Ph , d];
29. L2: Next-Hop-Selection;
30. If (r = s)
31. Eliminate BA [s, M[2], Ph , d] ;
32. // Update the routing table
33. ρ = 0.8
34. $\emptyset$ = 0.3
35. Bd_k= number of visited nodes by the backward ant
36. $$\tau'(r,s) = (1-\rho) * \tau(r,s)\left[\frac{\Delta\tau}{\emptyset Bd_k}\right]$$
37. Goto L3;
38. Else
39. Goto L2;
40. Else
41. Next-Hop-Selection;
42. Goto L1;

43. L3: End
44. // Procedure Next-Hop-Selection
45. Proc Next-Hop-Selection {
46. s = next Intermediate node
47. e_s = actual energy;
48. N = Number of nodes;
49. E(s) = visibility of *s*;
50. E(s) = $\frac{1}{C_s - e_s}$;
51. E[] ← E(s);
52. τ = pheromone routing table;
53. P(r, s) = probability of *r* jump to *s* as a next hop;
54. α = 0.9
55. β = 0.2
56. P(r, s) = $\frac{[\tau(r,s)]^{\alpha} * [E(s)]^{\beta}}{\sum[\tau(r,u)]^{\alpha} * [E(s)]^{\beta}}$;
57. X = number of neighbors which are located in the destination direction;
58. P[X] = array for storing probability amount of neighbors;
59. P[X] ← P(r, s);
60. P_{max} = 0;
61. For (i=0 ; i=X ; i++){
62. If (P [i] > P_{max})
63. P_{max} = P[i];
64. }
65. r= s.P_{max} ;
66. If (r ∈ M[].FA)
67. Loop happens then eliminate FA;
68. Else
69. { LastV= Count the member of M[];
70. If (LastV = 2)
71. Delete M[i];
72. M[i] ← M[i+1] ;
73. M[i+1] ← r ;
74. }
75. //Update the source probability distribution table
76. For (i=0; i=n; i++){
77. Check the RoutingTable.S ;
78. Find nodes which could be D simultaneously Then
79. P_{dk} = $\frac{9N_k - 5}{4{N_k}^2}$;
80. Update [dp].S_{PDT} ← P_{dk} ;
81. For other neighbor
82. P_{ik} = $\frac{4N_k - 5}{4{N_k}^2}$;

```
83.          Update [dp].S_PDT ← P_ik ;
84.              //check the link failure
85.           If ([dp].S_PDT = 0)       // it means the link is lost
86.           {
87.                Z= P_id / (1−P_id) ;
88.          P'_id = P_id * (1 + Z);
89.           Update [dp].S_PDT ← P_id ;
90.           }
91. }
92. // Procedure Calculate-Pheromone
93. Proc Calculate-Pheromone{
94.      E_min ← Min.E[];
95.      E_avg ← Avg.E[];
96.      N_j = number of visited nodes by forward ant
97.      Δτ = 1 / (C − ⌈(E_min − N_j)/(E_avg − N_j)⌉) ;
98.      BA.Ph ← Δτ }
```

Several experiments were conducted to verify the performance of the improved EEABR protocol for WSN. In the first case, we verify the performance with the protocols in which the original EEABR was compared, and we also used the same environment. In the second part of our experiment, we used a well known simulation environment mainly designed for wireless sensor network (RMASE), for both comparisons with other ant based routing protocols, and finally compared to other swarm intelligence routing protocols.

3.6 Experimental comparisons using NS-2 simulation environment

This section highlighted on the main contribution of IEEABR algorithm, and evaluates its performance on the wireless sensor network platform, while comparing it with some standard routing algorithms using NS-2 simulation environment. The simulation environment in visual form normally seen with the Network Animator (NAM) of which its file ends with '.nam' and sample of the trace files also ends with the '.tr' for analysis are presented below.

3.6.1 The Trace File

The trace files created by a simulation in NS-2 are always incredibly large. They are so detailed that it is almost impossible to find out any high level statistical instrument from the raw data without extra effort on developing software tools (awk scripts) or carefully extracting the required information. Though, one could extract the information needed for statistical analysis using the format used below with the sample of a trace file (raw data) as below.

(1) s 5.000000000 _1_ AGT --- 0 cbr 1 [0 0 0 0] [energy 50.000000 ei 0.000 es 0.000 et 0.000 er 0.000]

We can interpret the above raw date (trace file data) gotten from the NS-2.34 simulation as; node 1 sent a cbr packet whose id is 0, and size 1Mb at time 5.0 second to application 0 on node 0. The initial energy is 50.0, energy consumption in idle state (ei) = 0, energy consumption in sleep (es) = 0, energy consumption in transmitting packet (et) = 0, energy consumption in receiving a packet (er) = 0.

(2) r 98.026409126 _6_ RTR --- 0 IEEABR 48 [0 ffffffff 3 800] [energy 49.927119 ei 0.070 es 0.000 et 0.001 er 0.002]

In the same way, the routing agent on node 6 received an IEEABR packet with mac address 0xff; routing packet whose id is 0 and size 48 at time 98.0264 seconds from node 3 and mac address 8. The total remaining energy on the path (residual energy) is 49.927J, energy consumed at the idle state = 0.07, at the sleeping state = 0, transmitting state = 0.001, and receiving = 0.002J.

This was used for the whole trace file gotten from the simulation and hence further processed (analyzed) for further used.

3.6.2 Simulations Environment

We use an event driven network simulator-2 (NS-2) based on the network topology so as to evaluate the performance of IEEABR algorithm. The NS-2 provides a reliable simulation environment for wireless communication and it is provided with detailed propagation, MAC and radio layers. AntSense (an NS-2 module for Ant Colony Optimization) (Camilo, 2007) was used for the EEABR. The simulation environment was set to the same settings as that of the EEABR. Deployment of sensor nodes is randomly distributed over 200 x 200 square meters (10 nodes), 300 x 300 square meters (20 nodes), 400 x 400 square meters (30 nodes), 500 x 500 square meters (40 nodes) and 600 x 600 square meters when 50 to 100 nodes for a step of 10 nodes are used to monitor a static environment. Though, we also extend the number of deployed nodes to 120 to ascertain the behaviors of the network. We used a constant bit rate traffic model whereby the packet size was set to 1Mb, and each node had an initial energy of 5 Joules. We set the propagation model of wireless sensor network as a two-ray ground reflection model. The MAC (medium access control) protocol for IEEE 802.11 and the bandwidth of the channel is set to 1Mbps. We assume that all nodes have no mobility since the nodes are fixed in the application of most wireless sensor networks. Simulations were run for 5 minutes (300 seconds) for each set of experiment, and the remaining energy of all nodes were taken and recorded at the end of each simulation. The average energy calculated while also noting the minimum energy of the nodes. Fig. 3.7(a) and (b) shows a screenshot of a NAM window of the simulation environment for 120 nodes randomly deployed and 10 nodes respectively.

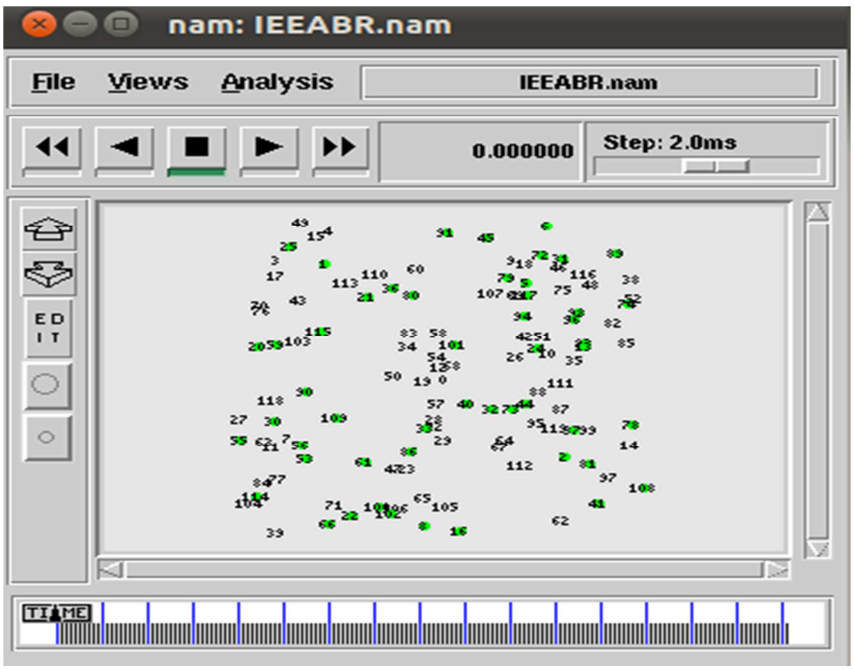


Fig. 3.7 (a): Graphical Representation of the Simulation Environment in NS-2.34 with 120 Nodes.

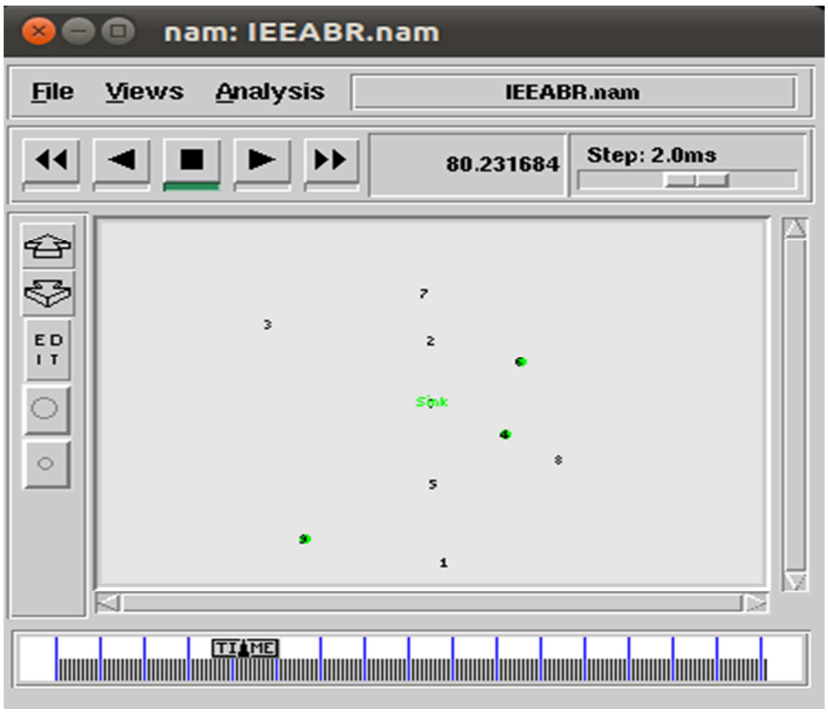


Fig. 3.7 (b): Graphical Representation of the Simulation Environment in NS-2.34 with 10 Nodes.

3.6.3 Simulation Results

The results obtained for IABR, EEABR, and IEEABR algorithm is presented and commented on in this section. Detailed description of the algorithms used for the purpose of comparison is presented in Chapter 2 of this thesis. In order to obtain greater understanding of performance and differences between the algorithms used for the comparison purpose, we employ three performance parameters for the analysis, which includes Minimum Energy of nodes, Average Energy of nodes, and the Energy Efficiency of the algorithm. The simulation was done on a static WSN, and the actual position where the event to be monitored and the destination node resides are unknown. Each participating node was given the responsibility of monitoring its immediate environment, and the sensed event had to be sent to the destination node. That is to say sensor nodes within the vicinity of the target event will depreciate easily in energy as they must assist in routing the sensed event from other nodes that are far away from the sink node. Simulations were run for 5 minutes (300 seconds) each time the simulation starts, and the remaining energy of all nodes in the network was recorded at the end of each simulation experiment. Fig. 3.8 provides the results of the experiments for the performance metrics used for the comparison purpose: The Average Energy, Minimum Energy, and Energy Efficiency of IABR, EEABR and IEEABR. As can be seen from the results presented in the figures (Fig. 3.8), the IEEABR protocol achieved better results in both Average energies of the nodes and the Minimum energy of node experienced at the end of the simulation. The IABR as compared to the EEABR perform worst in all cases. However, both had a very good result in terms of Energy Efficiency since both Algorithms are Energy aware. Both Algorithms had a dead node at the end of the simulation, since it is not practically avoidable as the nodes were randomly distributed and some nodes were responsible for forwarding the sensed events on-behalf of

others (intermediate nodes) to the sink node. The results of the simulations are as shown in Fig. 3.8.

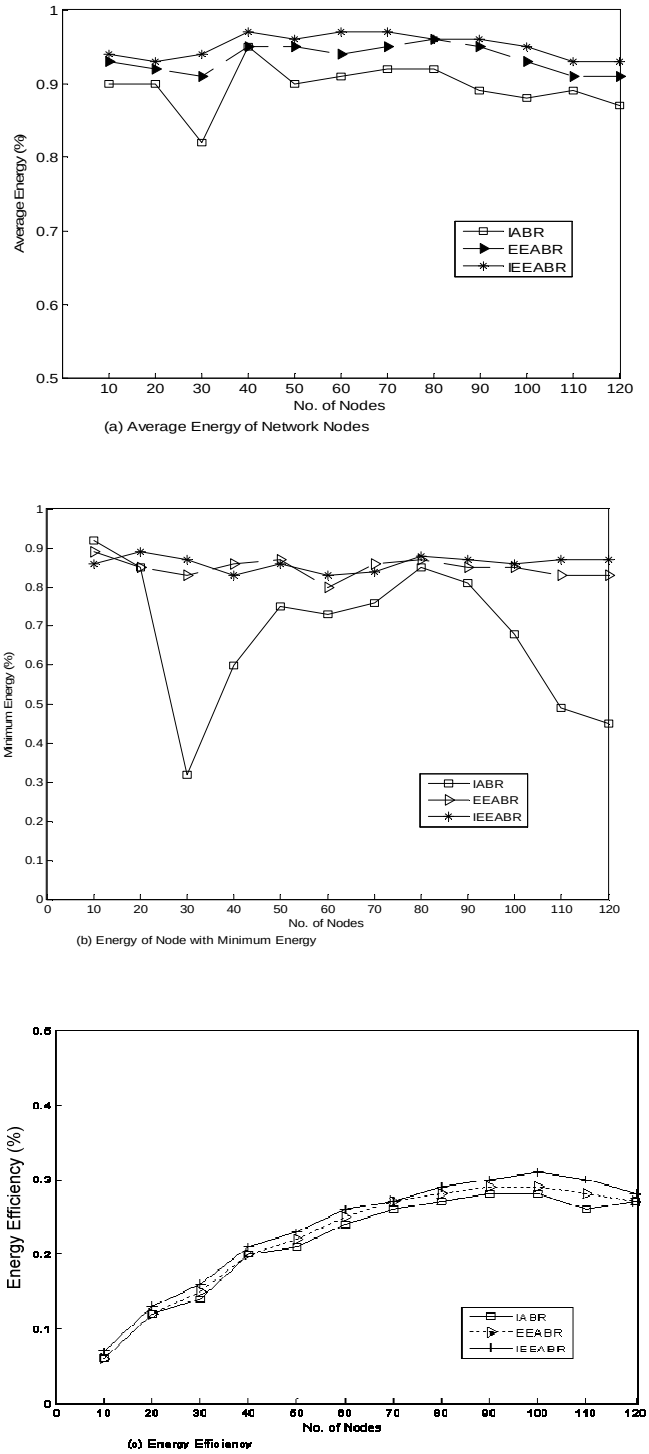


Fig. 3.8: Performance analysis of IABR, EEABR and IEEABR Energy Efficient Protocols.

3.6.4 Discussion

In this section of the chapter, research based on the application of ACO metaheuristic for solving the routing problem in WSN was adopted. We proposed an Improved Energy Efficient Ant Based routing Algorithm, which improves the lifetime of sensor networks. The improved version of the EEABR called the Improved Energy Efficient Ant-Based Routing (IEEABR), utilizes initialization of uniform probability distribution in the routing table, while giving special consideration to neighboring nodes which falls to be the destination (sinks) in order to save time in searching for the sink leading to reduced energy consumption by the nodes. The IEEABR approach has reduced energy usage leading to an effective way of balancing the WSN node power consumption and in turn helps to increase the network lifetime. From the results received from our experiments it was observed that IEEABR performance was good as compared to others in different WSNs. The protocol considers the residual energy of nodes in the network after each simulation period. Based upon the NS-2 simulation, the algorithm has been verified with very good performance in low energy consumption and high Energy efficiency. Consequently, our proposed algorithm can efficiently extend the network lifetime without performance degradation.

3.7 Experimental comparison using Matlab based simulation environment

In this experiment, we use a Routing Modeling Application Simulation Environment (RMASE) (PARC, 2006; Zhang et al., 2006; Zhang, 2005) specifically designed for wireless sensor networks, which is a framework implemented as an application in the Probabilistic wireless network simulator (Prowler) (Sztipanovits, 2004) written and runs under Matlab. Prowler is an event-driven simulator that can be set to operate in either deterministic or probabilistic mode, and it provides an easier way to

prototype applications and having better visualization capabilities. Prowler consists of radio model as well as a Medium Access Control (MAC) layer model. The MAC layer simulates the Berkeley motes' CSMA protocol, including the random waiting and back-offs. Also, it has a support for event-based structure which is similar to that of TinyOS and as such, facilitates the implementation of algorithms on real sensor nodes. The Radio propagation model determines the strength of a transmitted signal at a particular point of the space for all transmitters in the system. Based on this information, the signal reception conditions for the receivers can be evaluated and collisions can be detected. The signal strength from the transmitter to a receiver is determined by a deterministic propagation function, and by random disturbances.

We evaluated Improved Energy Efficient Ant Based Routing (IEEABR) with three candidate algorithms: SC, Beesensor, and EEABR algorithm using the metrics defined in Section 3.7.1 based on the experimental results obtained. The evaluation of the protocols was performed for two different types of application scenarios, that is the static scenario, and dynamic scenario. In both scenarios, the event has a length of 512-bits and this is generated at a rate of four events per second at each source node. In our experiment, the initial network topology was a 9 sensor nodes (3x3) grid with small random offsets. We later test both applications on other topologies consisting of 12, 36, 49, 64, and 100 sensor nodes. The network of 36 nodes is in a square of $120m\ x\ 120m$, and the maximum allowable transmission radius of a node was 70m, and the initial energy of each node is set to 5J each for both application types. Other topologies are generated based on the number of nodes required for the area. Each experiment was performed for duration of 100 seconds. The set of results recorded were averaged over ten different simulation results.

3.7.1 Performance evaluation metrics

From several results obtained from our simulation experiments, we report the following performance metrics for clarity purpose.

Success rate: it is defined as the ratio of total number of events received at the sink node to the total number of events generated by the nodes in the sensor network. We report it in percentage (%).

Energy consumption: it is defined as the total energy consumed by the nodes in the network during the period of the experiment in Joules (J).

Energy efficiency: it is a measure of the ratio of total packet delivered at the destination to the total energy consumed by the network's sensor nodes i.e. $\left(\frac{Succes\ rate * total\ packet\ sent\ to\ the\ sink\ in\ bits}{Total\ energy\ consumed}\right)$ (Kbits/Joules).

Standard Deviation: this shows how much variation it is between the energy values of all nodes in the network during the experiment. It is measured in Joules (J).

Network Lifetime: it is defined as the difference of total energy of the network and the summation of average used energy of nodes and the standard deviation of their energy levels i.e. $Lifetime =$ $\left(total\ network\ energy - (\frac{used\ energy}{total\ nodes} + energy\ deviation)\right)$ (Joules). The measurement was taken as a percentage (%) of the obtained values and converted to days.

Latency: it is defined as the time delay of an event sent from the source node to the destination node (seconds). That is, it is the actual time difference between when information is transferred to the destination node by the source node, and when it eventually reaches the destination node.

3.7.2 Comparison of experimental results for IEEABR, SC, EEABR and Beesensor

To better understand the differences between these four algorithms, we tested both algorithms using two well known applications of WSN. In all the application, each one tries as much as possible to denote the application to

real WSN deployment environments, and all nodes in both scenarios were randomly distributed in the experimental environment. This is because, in real wireless sensor network applications, node distribution cannot be controlled by any one due to the dynamic nature of the environment and its characteristics. In the converge-cast scenario, where sensor nodes were distributed in a random fashion with the main aims of monitoring a static event, all sources and sink are fixed, while the center of the circle is randomly selected at the start of the experiment. In this scenario, the location of the event and the position of the end node are assumed unknown, and each node is responsible to monitor its own vicinity and send the relevant sensed information to the destination node. The results of our experiments are shown in Table 3.1, and Fig. 3.9 (a-g). In this scenario of fixed sensor nodes, nodes near the sink are more likely to use up their energy as they serve as sources and the same time as routers for the other source nodes, hence, they will be forced to periodically transmit information on behalf of other nodes. In Table 3.2, and Fig. 3.10, we show the simulation results received from our extensive experiment using the target-tracking scenario. In target-tracking scenario, a sensor node within the coverage of a moving target generates random sequence of information to be able to track the target, and as the target drifts away from the coverage or transmission range of that particular node, it stops generating the random information, and this is taken up by another node within the coverage of the target/event, and in this, the sink node move randomly in the monitored area, hence some broken path are taken up by other paths so that the event can be delivered to the sink node via other alternate available path. We then analyzed our results based on the metrics used for the performance comparison as highlighted below.

3.7.2.1 Energy consumption

Limited available energy which is the major problem of wireless sensor network needs more attention when designing an efficient protocol. Fig. 3.9(a) shows the energy consumption of protocols for 9 nodes in the static network scenario, while Fig. 3.9(c) is the energy consumption of protocols for different densities of the network for the variation from 9, 16, 36, 64, and 100 nodes. SC performs better in the lower dense network of 9 nodes with 3% difference in performance as against the proposed algorithm, while the proposed algorithm performs better when the network grows higher. The percentage difference between IEEABR algorithm and SC when the network grows to 49 nodes is 25%, hence, much performance difference. EEABR and Beesensor consumed 31%, and 29.3% of networks energy than the proposed algorithm respectively. Lower energy consumption of SC protocol in the low density network is due to the assumption that each node in the network carries a location sensor so as to sense the location of the sink node at the beginning of the routing process, in this case, a GPS. This is not advantageous when it comes to the cost of purchasing the extra accessories for each node before implementation. It was also observed that in the densely populated network, the performance of SC protocol drops below that of IEEABR due to more nodes participating in the routing process, and less flooding of ants by IEEABR. As also observed in the target-tracking scenario in Fig. 3.10(a) and 3.10(c) for both fewer nodes and highly dense network of varying density, in Fig. 3.10(a), SC protocol consumes 72.65% energy more than IEEABR algorithm, which shows a high performance in the converge-cast scenario. The poor performance of the SC protocol as compared to IEEABR is due to the dynamic network, and more control packets are flooded in the network to locate the sink position. SC finds it difficult to re-calculate the location of the sink any time it changes position. While also the proposed algorithm

shows a great improvement over the original EEABR with the percentage difference of 10.6%, though Beesensor has the poorest performance as it consumes 83% higher than the IEEABR. As can be seen in Fig. 3.10(c), the percentage difference between IEEABR algorithm and SC when the network grows to 49 nodes is 60% which is a high performance difference. The proposed algorithm with its predecessor at that point is 29.66%, and Beesensor is lower at that point but higher when the network grows to 100, with a difference of 88% as against IEEABR algorithm. Hence the proposed algorithm outperforms all the protocols in term of low energy consumption. The high improvement is due to the reduced flooding of ants in the network, and proper initialization of the routing table, while giving preference to the sink selection among the neighbors.

3.7.2.2 Energy efficiency

Fig. 3.9(b) shows the energy efficiency of the protocols in static scenario. As energy consumption is an important metric to be considered when designing an efficient protocol. IEEABR and EEABR performance was higher than others in terms of energy utilization efficiency in this scenario. IEEABR better performance was with respect to its lower energy consumption and high success rate. For example, if the packet loss rate is high as compared to a success rate as in the case of SC, it will lead to excessive route discovery messages, which will contribute to more consumption of energy by the participating nodes. It will be observed that Beesensor consumes more energy than SC protocol. Though in comparison with their energy utilization efficiency, SC is close to Beesensor which is clearly due to low success rate of SC. In this scenario, energy-utilization efficiency bars of IEEABR and EEABR approaches each other. But in the dynamic scenario, the performance of EEABR was not comparable to IEEABR. This is due to the fact that IEEABR converges faster in this application and as such attained high success rate. Though in the static application, their route

discovery messages were low and as such, the total energy consumption of both algorithms approaches each other. That is to say that as more route discovery message are utilized, more control packets are released leading to high overhead and hence, affect negatively on the energy utilization efficiency of the EEABR algorithm. In most of the nodes ran out of energy in Beesensor, which is the overshoot as seen in the Fig. 3.9(c). Also, if we compare the protocols in the target-tracking application, Fig. 3.10(b) shows the energy efficiency of the protocols. It is clearly seen that, IEEABR not only having a high success rate, low energy consumption, is the most energy efficient among the protocols under consideration. In the converge-cast scenario, the energy-efficiency bars of IEEABR and EEABR are close to each other. On the other hand, in this Dynamic scenario, IEEABR performance was higher as compared to EEABR. The better performance was the ability of IEEABR to converge faster in this application and hence achieve a high success rate. IEEABR also outperform all the routing protocols in term of Energy efficiency. The percentage difference in the target-tracking scenario between the proposed algorithm and its predecessor is 64.22%, 84.7% as against Beesensor and 93.2% for SC which is most costly in its algorithm implementation. Though, all the protocols are energy aware protocols, the proposed algorithm still has a high success rate and the lowest end-to-end delay.

3.7.2.3 Success rate

The success rate of any protocol is the ability of the protocols to successfully deliver to the sink, the packets generated at each node in the network. The success rate of the protocols in both applications is shown in Fig. 3.9 (e), and Fig. 3.10 (d) respectively for static and dynamic network. EEABR shows a high performance as it delivered 92.4% of all the packets generated in the network to the sink during the period of observation without much loss, whereas IEEABR algorithm having an average of 96%

was the best among the protocols. SC has the poorest delivering rate in the experiment. The poor performance of the SC routing protocol is due to the path selection based on distance only without consideration of the energy of the path, in which some nodes of the paths might not be able to deliver the packets given to them for onward delivery. The high success rate of IEEABR shows that event gathering through multipath and link failure management is robust in both applications of sensor networks. Looking at both static and dynamic scenarios, the success rate of EEABR is much higher when compared with SC and Beesensor algorithms and this is mostly observed in high density networks. It will also be noticed that SC performance was poor. The poor performance of the SC is due to the flooding of ants without consideration of energy of paths, and path selection is based on distance only, in which some nodes of the paths might not be able to deliver the packets given to them for onward forwarding.

3.7.2.4 Latency

Fig. 3.9(d) shows the end-to-end delay of the protocols under evaluation. As seen from the figure though not our priority in this work, the proposed algorithm has the lowest end-to-end delay (latency) followed by SC. Beesensor performance was worst throughout the period of observation as can be seen in the figure. The poor performance of Beesensor is due to the reactive method of route discovery of which routes have to be recalculated any time there exist an event to be sent to the sink. As IEEABR algorithm limits the number of ants flooding in the network to a fraction of 5 times the number of network nodes, while also assigning greater probability to a neighbor that falls at the same time as the sink, it performs better than all the protocols. It is also worth noting that Beesensor routes foragers throughout high energy nodes and on demand, couple with small event cache maintained at each node, and each of the detected information is

stored in these caches when the routing process is in progress which contributes to Beesensor having higher latency.

3.7.2.5 Standard deviation

The standard deviation of node shows how much variation it is between the energy values of all nodes in the network during the experiment. A routing protocol is expected to achieve a high average leftover energy level of nodes during the experiment using small standard deviation so as to prolong the network lifetime. Fig. 3.9(f) and Fig. 3.10(e) gives the plots of results obtained for the four different algorithms in static and dynamic scenarios respectively. It will be observed that Beesensor has the lowest standard deviation in both scenarios, but the performance of IEEABR as compared to SC and EEABR protocol is better in both scenarios. The lowest standard deviation attained by Beesensor is due to the fact that, it is designed to operate as a reactive protocol, as such average variation between energy levels of all its nodes is significantly lower. The low standard deviation of Beesensor, help the protocol in the improvement of its lifetime even though it tends to consume high energy in the target tracking scenario, which significantly drops it lifetime below that of IEEABR.

3.7.2.6 Lifetime prediction

Fig. 3.9(g) and Fig. 3.10(f) shows the network lifetime prediction of the algorithms in converge-cast and target-tracking applications respectively. The result in Fig. 3.9(g) shows that Beesensor has better lifetime than IEEABR in a static scenario but degrade in its lifetime in the dynamic scenario. This drop in performance in the target-tracking scenario is expected of Beesensor since it generates significantly higher control overhead due to its reactive path establishment and the dynamic nature of the topology in that scenario. As a result of the high control packets, much energy is consumed by the protocol and in turn decreases its lifetime, even

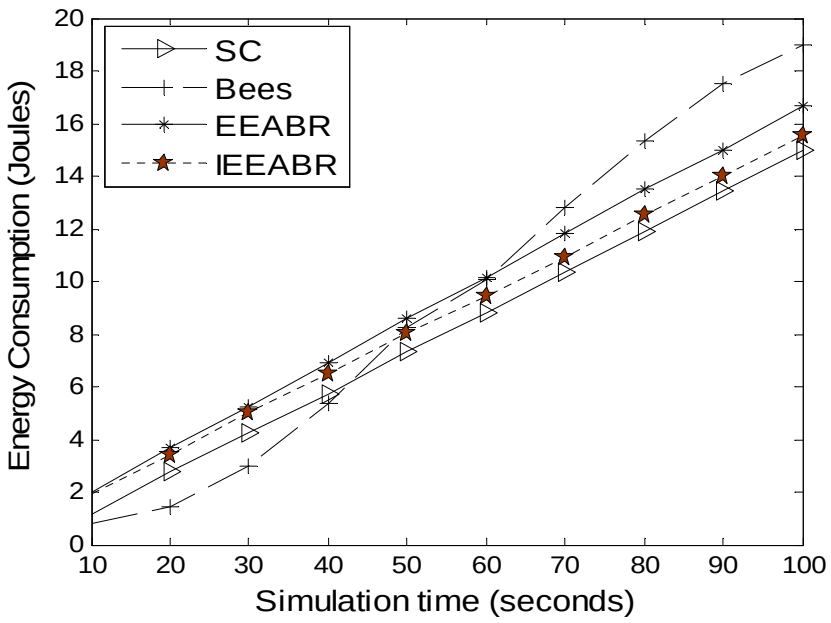
as it has a lower standard deviation. Though, EEABR protocol has a relatively better network lifetime as compared to the SC protocol in both scenarios. The reason is because, EEABR route packets through the path with higher average energy and avoid using the shortest path regularly. Also, it should be noted that SC protocol unicasts the forward ants using the shortest path, which could lead to high energy consumption on the path as it is often used and in turn will eventually lead to lower network lifetime for the SC protocol. Generally, IEEABR has the highest lifetime in the target tracking scenario and its performance is also comparable with Beesensor in the converge-cast application.

Table 3.1: Comparison of Routing Protocols in Converge-cast Scenario based on different Metrics

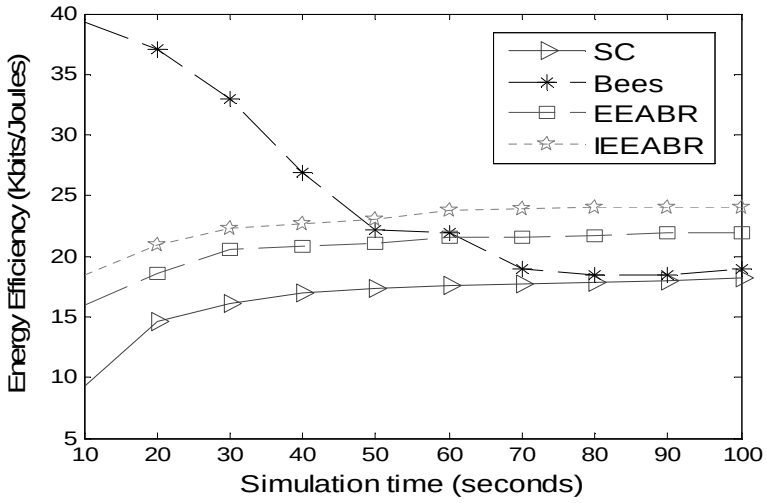
Comparison of the Routing Protocols based on different Metrics					
Routing Protocols	Latency (s)	Success Rate (%)	Energy Consumption (J)	Energy Efficiency (Kb/J)	Standard Deviation (J)
EEABR	0.0315	92.4240	16.6624	21.9656	2.6624
SC	0.0313	68.6870	14.9488	18.1955	2.9782
The proposed algorithm	0.0309	94.1920	15.5104	24.0484	2.7565
Beesensor	0.1229	90.9090	18.9696	18.9777	1.7042

Table 3.2: Comparison of Routing Protocols in Target-tracking Scenario for different Metrics

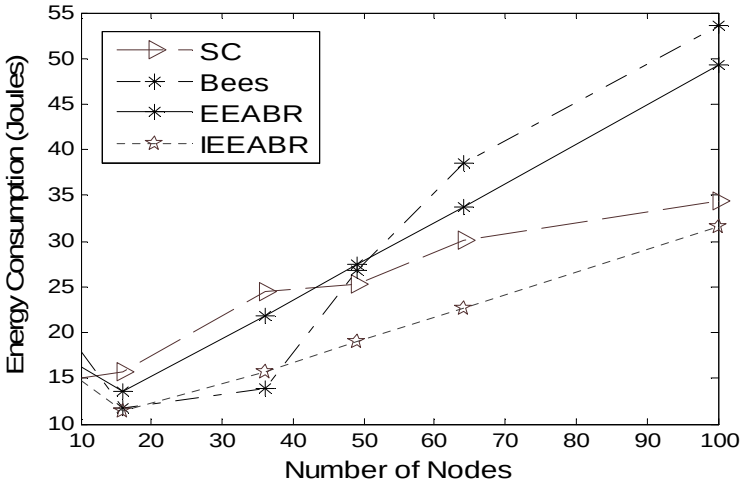
Comparison of the Routing Protocols based on different Metrics					
Routing Protocols	Latency (S)	Success Rate (%)	Energy Consumption (J)	Energy Efficiency (Kb/J)	Standard Deviation (J)
EEABR	0.0322	20.21	10.8032	0.7405	2.1284
SC	0.1941	12.63	40.3936	0.1238	3.0880
The proposed algorithm	0.0302	50.54	9.4064	2.1262	2.1230
Beesensor	0.0629	45.49	55.4075	0.3249	0.3463



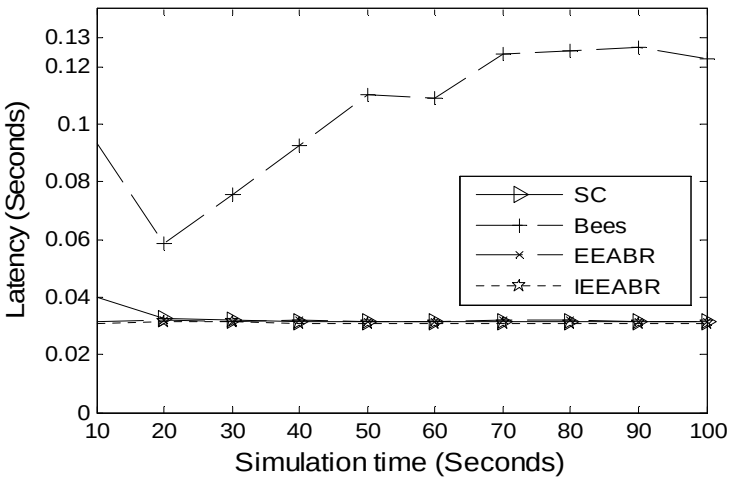
(a)


Energy Efficiency (Kbits/Joules)
Simulation time (seconds)
SC
Bees
EEABR
IEEABR

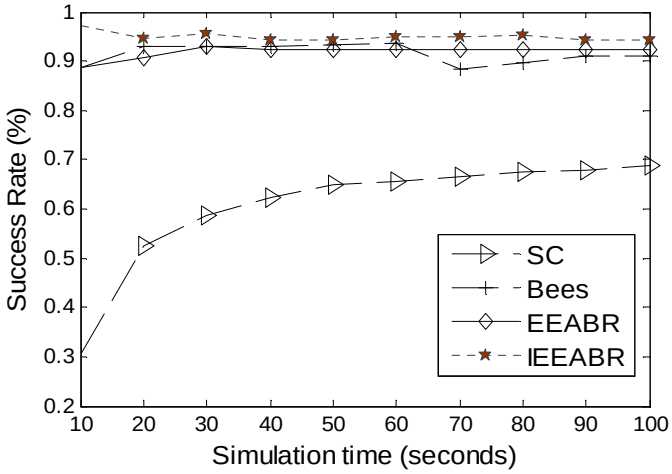
(b)


Energy Consumption (Joules)
Number of Nodes
SC
Bees
EEABR
IEEABR

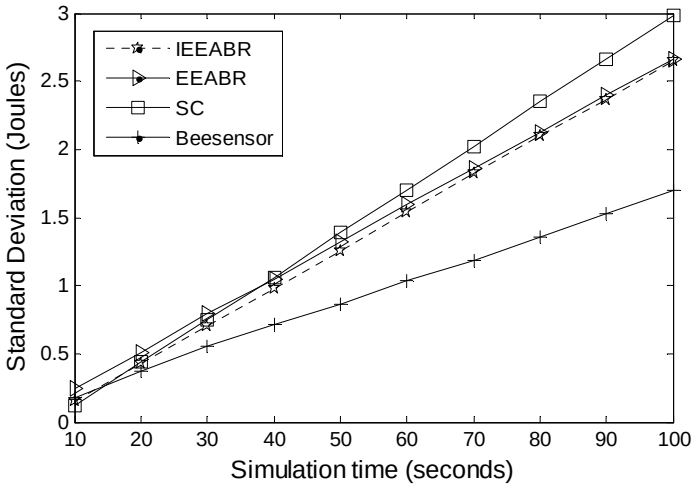
(c)


Latency (Seconds)
Simulation time (Seconds)
SC
Bees
EEABR
IEEABR

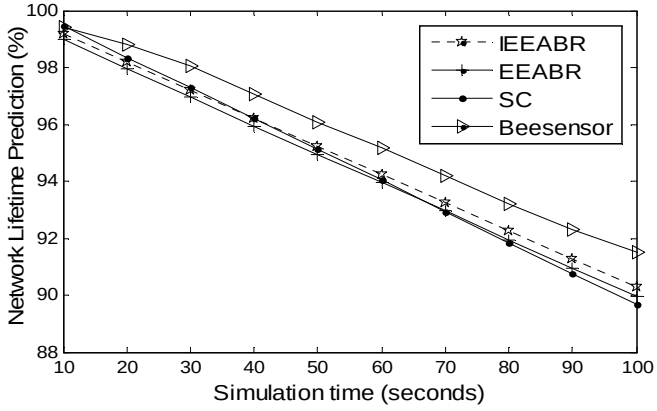
(d)



(e)

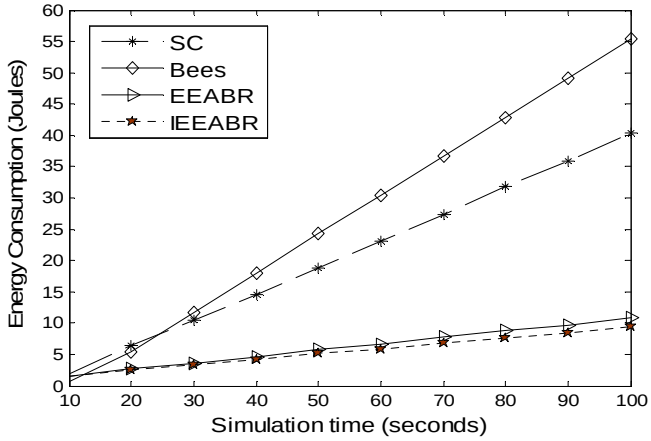


(f)

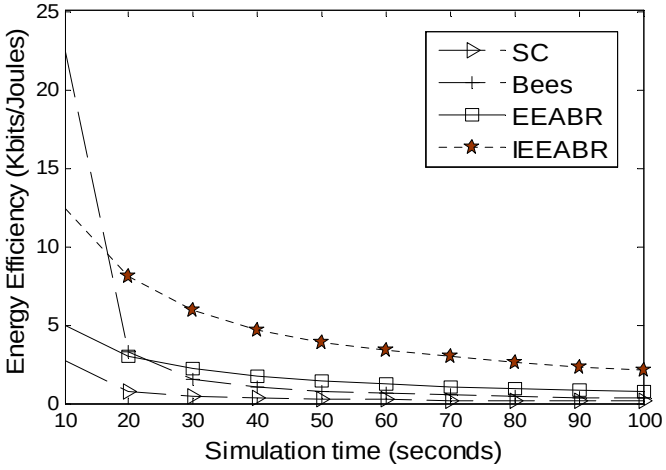


(g)

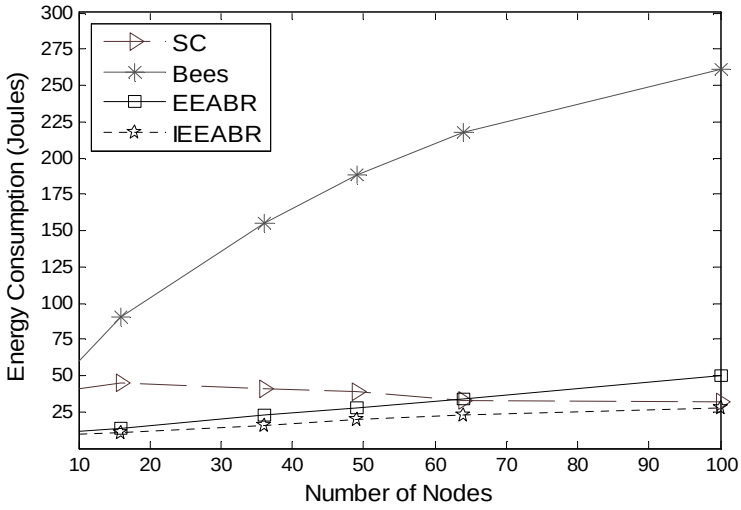
Fig. 3.9: Performance evaluation in Converge-cast scenario among four protocols: (a) Energy consumption (b) Energy efficiency (c) Energy consumption for different network's densities (d) Latency (e) Success rates (f) Standard deviation (g) Network lifetime prediction.


SC
Bees
EEABR
IEEABR
Energy Consumption (Joules)
Simulation time (seconds)

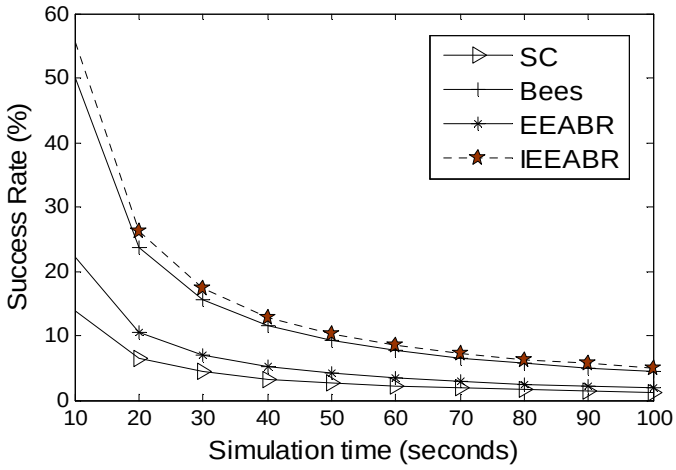
(a)


SC
Bees
EEABR
IEEABR
Energy Efficiency (Kbits/Joules)
Simulation time (seconds)

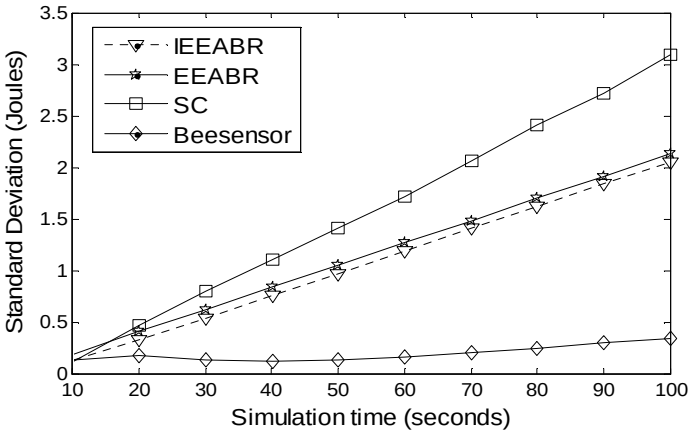
(b)


SC
Bees
EEABR
IEEABR
Energy Consumption (Joules)
Number of Nodes

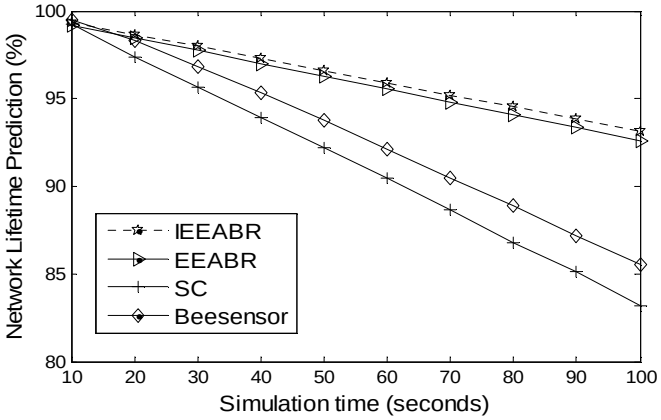
(c)



(d)



(e)



(f)

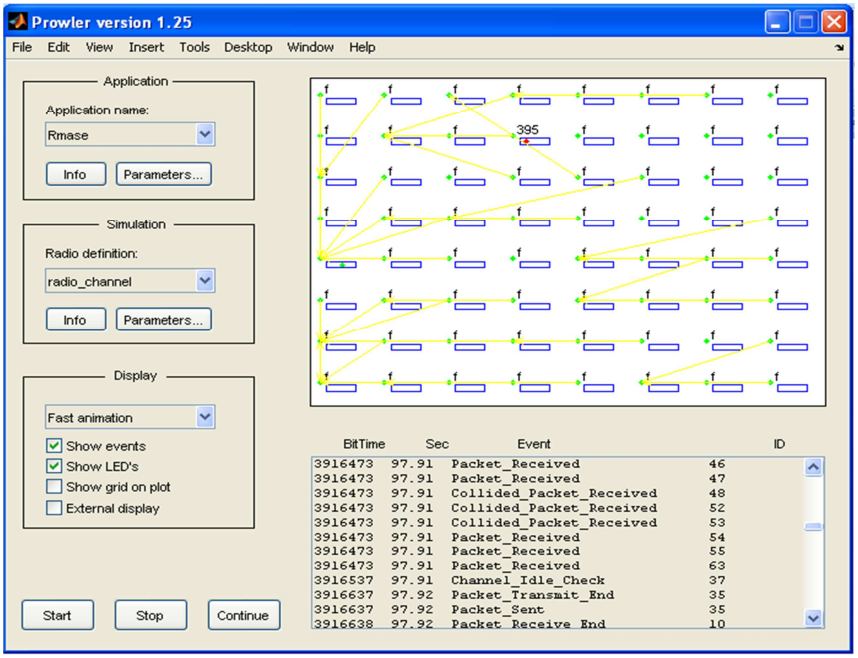
Fig. 3.10: Performance evaluation in Target-tracking scenario among four protocols: (a) Energy consumption (b) Energy efficiency (c) Energy consumption for different network's densities (d) Success rates (e) Standard deviation (f) Network lifetime prediction.

3.7.3 Discussion

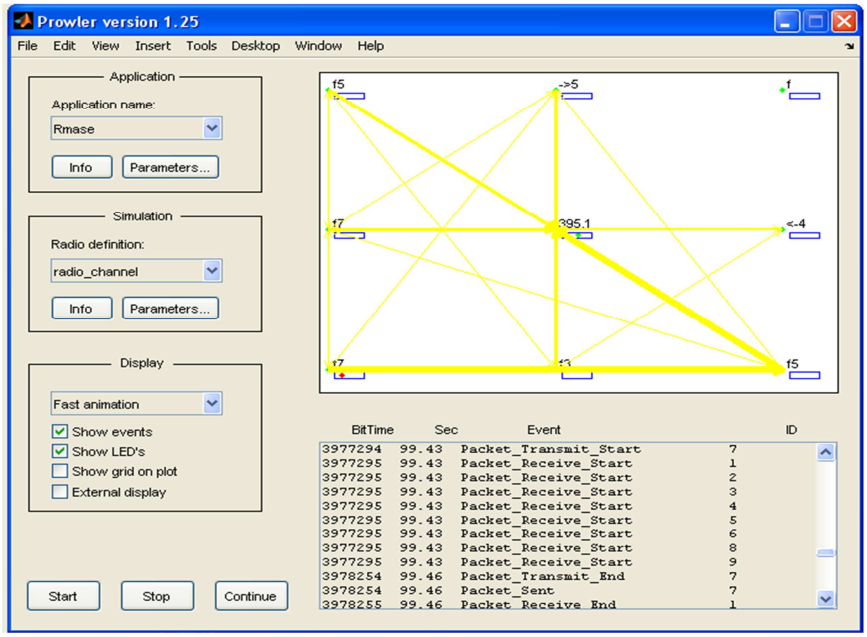
In this section, we have compared the performance of nature inspired state-of-the-art energy-aware routing protocols in WSNs that utilize the behavior of ant and bees in routing decisions. Our proposed routing protocol has shown and demonstrated good performance in terms of energy efficiency. The performance was not only on low energy consumption of its nodes in the network, but has low latency, high throughput and success rate. The algorithm is capable of handling target tracking applications as well as QoS demanding applications. The proposed algorithm performs quite well in all the metrics used for evaluation purpose while also showing reasonable differences between it and its predecessor 'EEABR'. EEABR consumes 31% and 29.66% higher than the proposed algorithm in term of energy consumption of the nodes in the network for converge-cast and target-tracking scenario respectively. SC which assumes that all sensor nodes have sensor to get the location of the sink did not do well as compared to the proposed algorithm. Beesensor consumes higher energy in both converge-cast and target-tracking scenario with 29.3% and 88% respectively due to its on-demand routing. The convergence time is low, and finds it difficult to locate sink whenever there is change in network topology. In comparison with the results gotten from the converge-cast scenario, when the target mobility increases, the overall performance of the algorithms depreciated. We actually expected that behavior due to the fact that, more nodes has to become sources since the target keeps changing position and at a faster rate. But it will still be observed that the IEEABR algorithm performance in that scenario was still better as compare to its counterpart. Though, the performance can easily be compared in the applications where by the environment have static variables (converge-cast scenario).

3.7.4 Performance Evaluation of Ant-Based Routing Protocols for Wireless Sensor Networks

This section presents the performance of our proposed algorithm, the Improved Energy-Efficient Ant-Based Routing (IEEABR) Algorithm in wireless sensor networks with other well-known ant-based protocols. Compared to the state-of-the-art Ant-Based routing protocols; Basic Ant-Based Routing (BABR) Algorithm, Sensor-driven and Cost-aware ant routing (SC), Flooded Forward ant routing (FF), Flooded Piggybacked ant routing (FP), and Energy-Efficient Ant-Based Routing (EEABR), the proposed IEEABR approach has advantages of reduced energy usage which can as well increase the network lifetime. The performance evaluations for the algorithms on a real application are conducted in a well-known WSNs MATLAB-based simulator (RMASE) using both static and dynamic scenarios. The evaluation of all the protocols was done using the metrics defined in Section 3.7.1. In our experiment, the network initially was a 3x3 (9) sensor grid, and later increase to 12, 36, 49, 64, and finally 100 nodes. Each experiment was performed for duration of 100 seconds. The experiment was conducted for two situations; when the sink is static, and when it is dynamic. In our experiment, the initial network topology was a 9 sensor nodes (3x3) grid with small random offsets. We later test both applications on other topologies consisting of 12, 36, 49, 64, and 100 sensor nodes and the nodes in the network were powered with initial energy values of 30J and 60J respectively for static and dynamic scenarios. The network of 36 nodes is in a square of $120m\ x\ 120m$, and the maximum allowable transmission radius of a node was 70m. Other topologies are generated based on the number of nodes required for the area. Each experiment was performed for duration of 100 seconds. The set of results recorded were averaged over ten different simulation results. The graphical user interface while simulating IEEABR and Basic ant routing is as shown in Fig. 3.11.



(a)



(b)

Fig. 3.11: Matlab Simulation Environment shows (a) Traces of forward ants in the IEEABR routing protocol (b) Traces of forward ants SC routing, where lines intensity indicates the probability of link selection.

3.7.4.1 Static Scenario

In the static scenario, all sources and sink are fixed, while the center of the circle is randomly selected at the start of the experiment.

Latency: Fig. 3.12(a) shows the end-to-end delay of the protocols under evaluation. As seen from the figure, IEEABR has the lowest end-to-end delay (latency) followed by its predecessor (EEABR). FF performance was poor, though, the basic ant routing performs worst throughout the period of observation as can be seen in the figure. The poor performance of FF and the basic ant routing is due to the flooding method of ants without control which could cause congestion in the network, hence increasing the latency. IEEABR limits the number of flooding ants in the network to a fraction of 5 times the number of network nodes, while also assigning greater probability to neighbor who falls the same time as the sink, perform better than all the protocols.

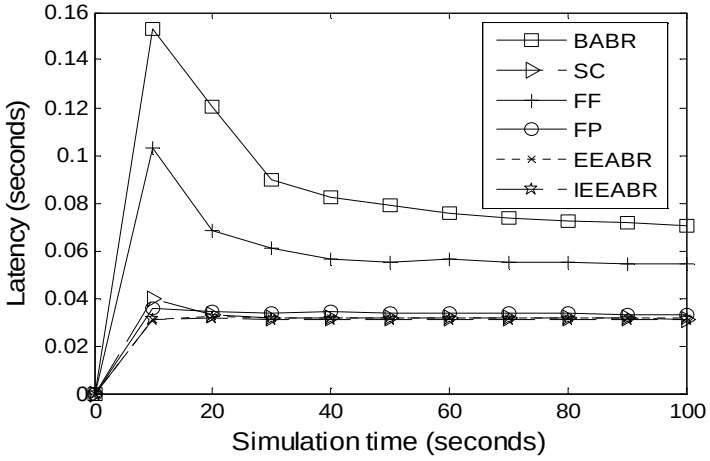
***Success rate*:** Fig. 3.12(b) shows the success rate of the protocols in other words, the ability of the protocols to deliver successfully to the sink the packets generated at each node in the network. Though, FP shows a wonderful performance as it delivered almost all the packets generated in the network to the sink during the period of observation without loss, whereas IEEABR having an average of 96% follows. FP-Ant has the highest packet- delivery ratio followed by IEEABR in this scenario. Though, in this case of the static application, the success rate of IEEABR is higher when compared with AODV, SC, BABR, FF, and EEABR, in a high density network. Another important observation is the poor performance of the SC and the basic ant routing. The poor performance of the basic ant routing and the SC is due to the flooding of ants without consideration of energy of paths, and path selection is based on distance only, in which some nodes of the paths might not be able to deliver the packets given to them for onward forwarding.

***Energy consumption*:** Fig. 3.12(c) shows the energy consumption of the protocols for 9 nodes in the network. While Fig. 3.12(e) is the energy consumption of protocols for different densities of the network, for the

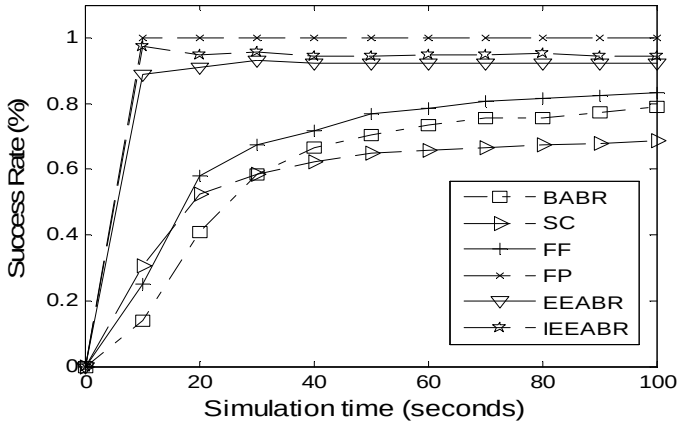
variation from 9, 16, 36, 64, and 100 nodes. SC performs better in the lower dense network of 9 nodes with 3% difference in performance as against IEEABR, while IEEABR perform better when the network grows higher. Lower energy consumption of SC is due to the assumption that each node has sensors to sense the location of the sink node at the beginning of the routing process, in this case GPS. This in turns adds to the cost of purchasing extra GPS to each node for practical implementation. The percentage difference between IEEABR and SC when the network grows to 49 nodes is 25%, hence, much performance difference. At that point of 49 nodes, EEABR consumes more of 31% of energy than IEEABR. Hence outperform all the protocols in term of low energy consumption. The FP performs worse in that case as almost all the nodes went down due to high energy consumption consuming 719.9J in the network of 100 nodes whereas IEEABR consumes 31.6J. The difference in the energy consumption is not comparable, even though it has the highest delivery ratio.

***Energy efficiency*:** Fig. 3.12(d) shows the energy efficiency of the protocols. As energy consumption is an important metric to be considered when designing an efficient protocol. IEEABR and EEABR have the best performance in terms of energy utilization efficiency in this scenario. IEEABR better performance was with respect to its lower energy consumption and high success rate. For example, if the packet loss rate is high as compared to a success rate as in the case of BABR, it will lead to excessive route discovery messages, which will contribute to more consumption of energy by the participating nodes. It will be observed that FP consumes more energy than BABR protocol. Though in comparison with their energy utilization efficiency, BABR is close to FP which is as a result of the low success rate of BABR. In this scenario, energy-utilization efficiency bars of IEEABR and EEABR approaches each other. But in the dynamic

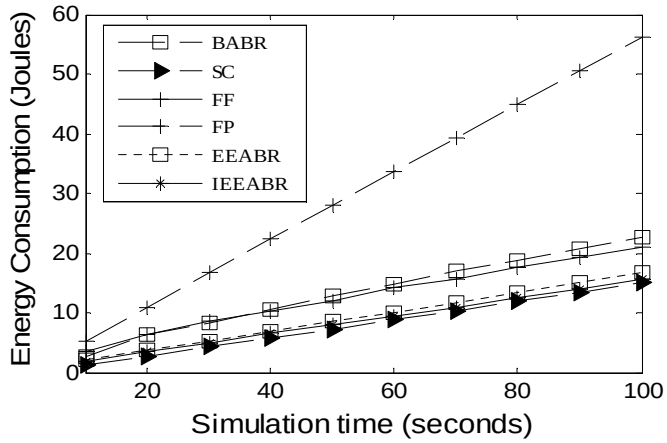
scenario, the performance of EEABR was not comparable to IEEABR. This is due to the fact that IEEABR converges faster in this application and as such attained high success rate. Though in the static application, their route discovery messages were low and as such, the total energy consumption of both algorithms approaches each other. That is to say that as more route discovery message are utilized, more control packets are released leading to high overhead and hence, affect negatively on the energy utilization efficiency of the EEABR algorithm. In fact virtually all the nodes ran out of energy in FF, which is the overshoot as seen in the Fig. 3.12 (e).



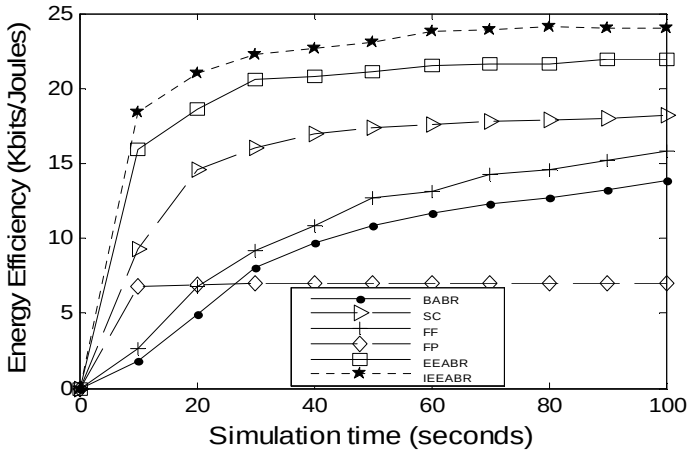
(a)



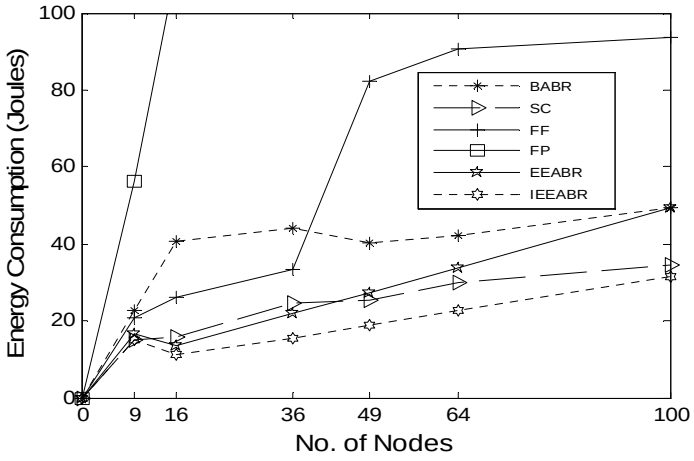
(b)



(c)



(d)



(e)

Fig. 3.12: Performance evaluation in static scenario among six (6) Ant-Based routing protocols: (a) Latency (b) Success rates (c) Energy consumption (d) Energy efficiency (e) Energy consumption for different network's densities.

3.7.4.2 Dynamic Scenario

In the dynamic scenario, all source nodes are fixed while sink dynamic, and the center of the circle is randomly selected at the start of the experiment.

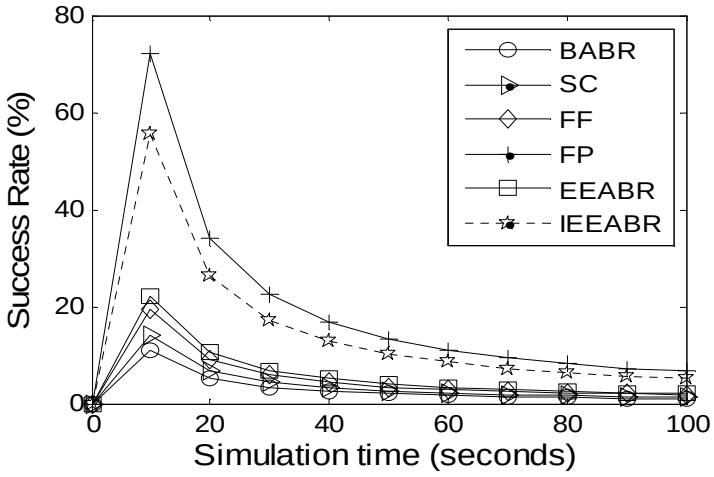
***Success rate*:** Fig. 3.13(a) shows the success rate of the protocols in the dynamic scenario, where the sink keeps on changing position, which is sometimes known as the target tracking. The success rate of any protocol is the ability of the protocols to deliver successfully to the sink the packets generated at each node in the network. FP algorithm has the best performance in terms of success rate followed by IEEABR in this scenario. In this dynamic scenario, the packet-delivery ratio of IEEABR is much more when compared with AODV, SC, BABR, FF, and EEABR, especially in large networks. As also observed is the poor performance of the SC and the basic ant routing. The poor performance of the basic ant routing and the SC is due to the flooding of ants without consideration of energy of paths, and path selection is based on distance only, in which some nodes of the paths might not be able to deliver the packets given to them for onward delivery, this was also noticed in the static scenario. IEEABR not only having a high success rate, but also, have the lowest energy consumption and more energy efficient. It will be noticed in this scenario that IEEABR outperforms its predecessor with 60%, which is quite a large difference in performance in terms of quality of service.

***Energy consumption*:** Limited available energy which is the major problem of wireless sensor networks has to be looked upon critically when designing an efficient protocol. Fig. 3.13(b) shows the energy consumption of protocols for 9 nodes in a grid network. While Fig. 3.13(d), is the energy consumption of protocols for different densities of the network for the variation from 9, 16, 36, 64, and 100 nodes. As it can be seen in Fig. 3.13(b), SC consumes more 72.65% energy as compared to IEEABR, which shows a high performance in the static scenario, where it assumes that it

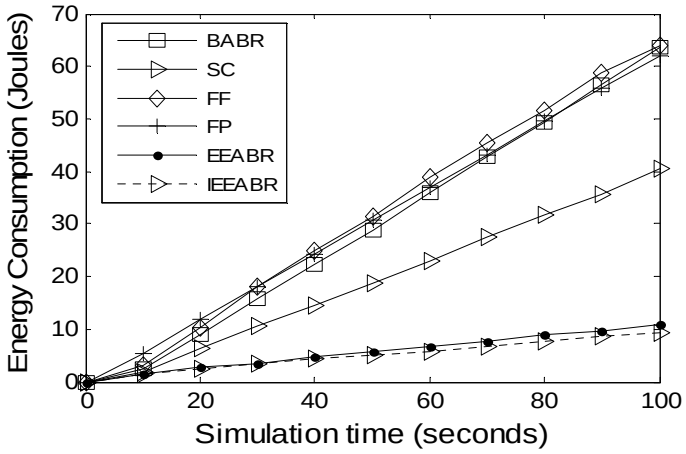
knows the location of the sink using a form of sensing level or otherwise GPS to detect the position of the sink during the initial routing process. While also IEEABR shows a great improvement on EEABR with the percentage difference of 10.6%. As can be seen in Fig. 3.13(d), the percentage difference between IEEABR and SC when the network grows to 49 nodes is 60% which is a high performance difference. IEEABR with its predecessor at that point is 29.66%. Hence outperform all the protocols in term of low energy consumption. The FP still performs worst in the tracking scenario, where almost all the nodes went down due to high energy consumption, consuming 812.7J in the network of 100 nodes whereas IEEABR consumes 27.82J. The difference in the energy consumption is not comparable, even though it has the highest delivery ratio and lowest end-to-end delay in packet delivery. The high improvement is due to the reduced flooding of ants in the network, and proper initialization of the routing table, while giving preference to the sink selection among the neighbors.

***Energy efficiency*:** Energy efficiency which is a function of energy consumption and the success rate, tells how well a protocol performs in both quality of service and network lifetime. As a network is expected to perform optimally while also performing for a long period of time without the performance degradation, Fig. 3.13(c) shows the energy efficiency of the protocols. It is clearly seen that, IEEABR not only having a high success rate, low energy consumption, is the most energy efficient among the protocols under consideration. In the static converge-cast scenario, the energy-utilization efficiency bars of IEEABR and EEABR approaches each other. But in the dynamic scenario, the performance of EEABR was not comparable to IEEABR. This is due to the fact that IEEABR converges faster in this application and as such attained high success rate. Though in the static application, their route discovery messages were low and as such,

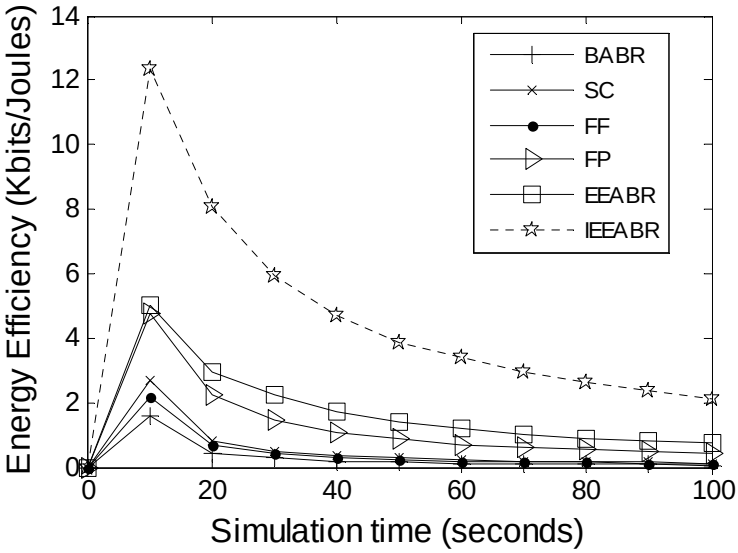
the total energy consumption of both algorithms approaches each other. That is to say that as more route discovery message are utilized, more control packets are released leading to high overhead and hence, affect negatively on the energy utilization efficiency of the EEABR algorithm. IEEABR also outperform all the routing protocols in term of Energy efficiency. The percentage difference in the dynamic scenario between IEEABR and EEABR is 64.22% and 93.2% for SC which is most costly in its algorithm implementation. FP having the highest success rate in the low density network as compared to BABR has the poorest result in terms of energy efficiency. Though, IEEABR and EEABR are energy aware protocols, and IEEABR still having highest success rate and lowest end-to-end delay.



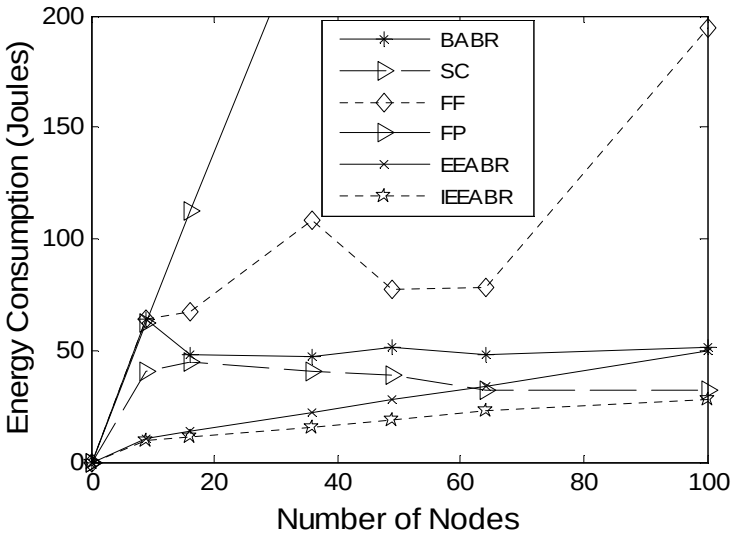
(a)



(b)



(c)



(d)

Fig. 3.13: Performance evaluation of routing protocols in dynamic scenario: (a) Success rates (b) Energy consumption (c) Energy efficiency (d) Energy consumption for different network's densities.

3.8 Conclusions

In this Chapter, we have proposed several improvements to EEABR. Consequently, we have compared the performance of nature inspired state-of-the-art energy-aware routing protocols in wireless sensor networks utilizing the behavior of ant and bees in routing decisions. IEEABR demonstrated good performance in terms of energy efficiency. The performance was not only on low energy consumption of its nodes in the network, but has low latency, high throughput and success rate. In terms of energy consumption, EEABR consumes 31% and 29.66% higher than the proposed algorithm in terms of energy consumption of the nodes in the network in converge-cast and target-tracking scenario respectively. The most interesting part of the results is in the dynamic scenario. The energy efficiency improvements are significant particularly for dynamic routing environments. The experimental results using RMASE simulation environment show that the proposed algorithm increases energy efficiency by up to 9% and 64% in converge-cast and target-tracking scenarios respectively over the original EEABR. SC which assumes that all sensor nodes have sensor to get the location of the sink did not do well as compared to the proposed algorithm. In comparison with the results gotten from the converge-cast scenario, when the target mobility increases, the overall performance of the algorithms depreciated. We actually expected that behavior due to the fact that, more nodes has to become sources since the target keeps changing position and at a faster rate. But it will still be observed that the IEEABR algorithm performance in that scenario was still better as compared to its counterpart as pointed out above. Though, the performance can easily be compared in the applications where by the environment have static variables (converge-cast scenario).

Chapter 4

Termite-hill: Termite inspired routing algorithm in wireless sensor networks

4.1 Introduction

The field of data/information gathering in wireless sensor networks (WSNs) has received much attention in recent years. As such, routing algorithms design and development have come into play with the main aims of overcoming the constraints that characterized WSNs. After the high success of Bee Colony Optimization (BCO) metaheuristic, and Ant Colony Optimization (ACO) metaheuristic, Termite Colony Optimization (TCO) metaheuristic has been proposed very recently as a new potential research direction in the area of swarm intelligence with the applications to engineering problems (Ramin et al., 2010; Roth and Wicker, 2003). Termite inspired routing protocols can be of immense benefit in the maximization of network lifetime without performance degradation. This will only be possible if the algorithm is adaptive and efficient and take into account the main constraints of WSN. Social insect communities have lots of pleasing qualities from the WSN point of view as surveyed in (Zungeru et al., 2012). The communities of the social insects are formed from the cooperation of the species which are simple and autonomous and are able to organize themselves. Though they lack centralize planning, but they are able to coordinate and effectively organize themselves. Due to their behavior which is adaptive and flexible, they are able to meet most of their

environmental objectives. The complexity of the solutions generated by these social insects with their simple behaviors indicates that the cooperation of many will yield many more results and achieve global objectives. With these behaviors, we can map the characteristics into sensor networks. We can consider sensor network having simple nodes working together to deliver sensed information to the end point (command center), with the environment consisting of its own topology and physical layer, communications links, and traffic patterns on the network. The difference between these social insects and the sensor networks is that social insects have a natural means of cooperation, while the sensor networks may need to be governed or force its nodes to cooperate (Buttyan and Hubaux, 2000). The ability of these biological species to self-organize is as a result of some behavioral principles, which includes positive feedback, negative feedback, randomness, multiple interactions and stigmergy (Roth and Wicker, 2003). The highlighted behavioral principles, which leads to self organization of the social insects is called swarm intelligence.

Research in this field of swarm intelligence is based on the working principles of ant colonies as adopted in (Dorigo and Caro, 1998; Bonabeau et al., 1999), slime mold (Li et al., 2011) and honeybees (Saleem et al., 2012). To the best of our knowledge, this chapter happens to be the first of its kind to adopt the behavior patterns of termites in solving routing problems in sensor networks. In comparison, a termite colony also consists of minimalist termite agents that through local interaction perform a system level behavior that is scalable, robust and tolerant to a dynamically changing environment without any centralized control. Second, individual nodes are less important as compared to overall sensor network. A similar characteristic is observed in a termite colony where termites always work to optimize the colony profit. Third, termite colony is extremely adaptive in

terms of its resource management e.g. when a foraging site quality falls below a certain threshold level, the foraging force is diverted towards a better quality food source. This is also a desirable characteristic as sensor networks are dynamic in nature and a routing protocol has to adapt to such environmental changes. Motivated by the similarities of the systems, in this chapter, the foraging principles of a termite colony were utilized to design a simple, scalable and energy efficient routing algorithm for WSNs termed *Termite-hill.* More specifically, we followed the following design guidelines to achieve the final objective.

1. Investigation of the constraints and requirements of WSNs. This helps in understanding the environment in which the newly proposed routing protocol was expected to operate.
2. The study of the foraging principles of a termite colony in order to take inspiration from the relevant concepts for our new WSN routing protocol.
3. The termite agent model must be simple and independent of the size of the network. This requirement primarily targets the scalability aspect of a routing protocol. The intended applications of wireless sensor networks, for instance battlefield surveillance, may contain several thousands of small tiny nodes. If an agent is allowed to have a source routing header of the complete path, its size will grow indefinitely making it impractical for large scale networks.
3. Routing overhead must be minimized in every possible way without affecting the capability of nodes to reach the most distant sink node(s). This is a critical requirement keeping in view of the fact that wireless sensor nodes carry limited and non-rechargeable batteries (Saleem et al., 2012; Sohraby et al., 2007). Pure flooding based protocols generate large routing overhead that can quickly drain the energy resources of individual nodes ultimately resulting in a lower network

lifetime. Stochastic rebroadcasting is one of the options usually used for minimizing the control overhead. However, its performance in large scale networks must be evaluated before making the final design choice.

4. Termite agents having limited knowledge about the overall network topology must cooperate and share the routing information with their fellow agents to make more intelligent routing decisions. This is one of the typical characteristics of Termite-hill routing protocol. Allowing the agents to communicate with each other at intermediate nodes will enable them to select the highest quality routes.
5. Distribution of network traffic across multiple routes is another critical requirement due to two main reasons. First, it naturally results in lower number of route discoveries in a dynamic network because the source and sink nodes are connected through multiple routes. Secondly, the network lifetime is extended as the batteries of individual nodes are drained at an approximately equal rate.
6. Nodes in an ad hoc network perform an additional task of routing the packets on behalf of the other nodes. Therefore, we have to ensure that the nodes lying on a route to the destination node should not be over stretched in terms of their resource usage (memory, energy, processing etc.) because it might affect the overall network performance. This chapter presents a biologically inspired self-organized routing protocol for WSN termed Termite-hill. The performance of the algorithm was studied by simulation on static, mobility and target tracking scenarios in WSN. The results of our extensive experiments clearly show that the proposed algorithm was able to balance the network traffic load, and prolong the network lifetime without performance degradation. Extensive simulations are used to evaluate the performance of the *Termite-hill*. The results clearly demonstrate that *Termite-hill*

outperforms the existing ACO-based routing algorithms; *SC* (Zhang et al., 2004b), *FF-Ant* (Zhang et al., 2004b) and a classical algorithm *AODV* (Perkins and Royer, 1999) in terms of energy efficiency, success rate, throughput, and network lifetime. Each of the algorithms has its advantages in terms of area of applications as well as drawbacks.

Besides all the drawback of each of the related protocols, almost all the algorithms tends to sacrifice the network performance as against the improvement of energy consumption of the nodes, and vice-versa for others. Detailed explanation of the principle of operation of the algorithms used for comparison purpose are outlined and described in Chapter 2. We also described the principles and behavioral pattern of termite species that inspired us in converting them to routing agents as described in Section 4.2.

4.2 The Termite Analogy

The search for food in termite is organized in part by chemical trails laid with the secretion of their abdominal sterna gland. During foraging for food, termites walk slowly at an average speed of 2.3mm/s, and lay a dotted trail by dabbing the abdomen at intervals on the ground (Reinhard and Kaib, 2001; Olugbemi, 2010). When food is discovered they return at a quick pace at an average of 8.9mm/s to the nest, laying a trail to recruit nest mates to the food source. While laying this recruitment trail, the workers drag the abdomen continuously on the ground. The difference between foraging and recruitment trails is attributed to different quantities of trail pheromone present on the path. Soldiers respond only to trails with a high concentration of trail pheromone. Considering Fig. 4.1, the path from the termite hill to the food source at time $t = 0$, the termites find the food and bring it back efficiently, establishing a pheromone trail to it. At time $t = 1$, when there is an obstacle in their path such that there is one

path that is shorter than the other, the termites can choose either path with equal probability, hence having the same number of termites on both sides. The path that is shorter will allow termites to gather food quickly and strengthen the pheromone trail on the way back faster than the termites on the longer path as seen when $t = 2$, making the other batch of termites to move with higher probability towards the trail that is stronger. As the process continues until time $t = n$, it will be observed that all termites will use the shortest path towards the food source. The obstacle in this analogy can be congestion, number of nodes on the path to the sink, latency etc. A simple example of the hill building behavior of termites provides a strong analogy to the mechanisms of the proposed Termite-hill algorithm. This example illustrates the four principles of self-organization (Roth & Wicker, 2003). If we take a flat surface as an example upon which termites and pebbles are distributed, the termites would like to build a hill from the pebbles, i.e. all of the pebbles should be collected into one place. Termites act independently of all other termites, and move only on the basis of an observed local pheromone gradient. Pheromone is a chemical excreted by the insect which evaporates and disperses over time. A termite is bound by the following rules:

1. A termite moves randomly, but is biased towards the locally observed pheromone gradient. If no pheromone exists, a termite moves uniformly randomly in any direction.
2. Each termite may carry only one pebble at a time.
3. If a termite *is not* carrying a pebble and it encounters one, the termite will pick it up.
4. If a termite *is* carrying a pebble and it encounters one, the termite will put the pebble down. The pebble will be infused with a certain amount of pheromone.

With these rules, a group of termites can collect dispersed pebbles into one place. Detail explanation of these rules and the results of the rules are given in the next section (Section 4.3). The following subsections explain the behavioral pattern of foraging of termites towards hill building. We purposely repeat this section on this part to highlight more on the behavioral pattern of foraging of termites towards hill building behavior, and also show the differences between ants and termites.

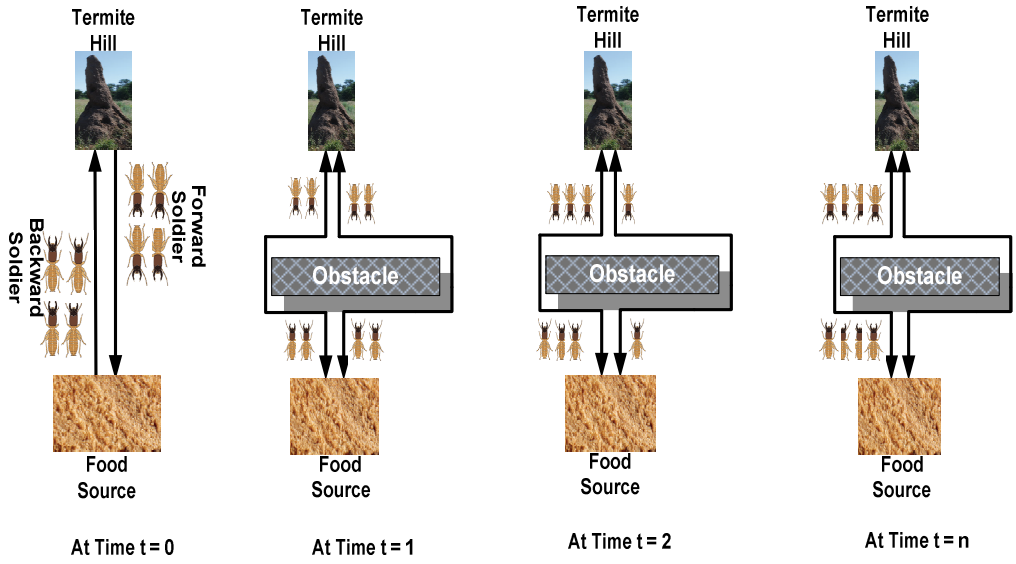


Fig. 4.1: Food Search in Termites.

4.2.1 Positive Feedback

In termites foraging, termites are motivated to add piles of wood to the site where there exist much piles of the woods, as it will have a high concentration of pheromone. If there are many wood piles in a place, it is expected that much pheromone will be present in that wood pile, that is to say that, more termites will visit the place due to its pheromone concentration, and as such the termites will add more pheromone to the pile on that path. If more termites are biased towards the woodchip source due to the concentration of pheromone of the path, it then means that more of them will take the path to that woodchip source, which means that pheromone content of the path towards the hill will be increased, and in turn attract more termites.

4.2.2 Negative Feedback

Negative feedback is accomplished by pheromone evaporation. This happens so as to avoid premature convergence among termites (stagnation). For good communication among the termite individuals, pheromone must evaporate over the environment. The evaporation helps to weaken the pheromone, which in turn lower the resulting hill. The path with lower pheromone concentration (lower hill) will have fewer termites as it will attract fewer termites towards its direction. Though this may seem contrary to the task of collecting all woodchip to the nest, but it is important. Negative feedback is entirely useful in the removal of past or poor solutions for the memory of constituent of any network or system.

4.2.3 Randomness

The location and path taken by termites towards the woodchip source is determined by chance. A little drift in the behavior of termites will lead to advance effect on the future behavior of the system. Randomness is useful so that new solutions can be built as the network and system under consideration is dynamic.

4.2.4 Multiple Interactions

In the woodchip collection of termites to their hill, it is a necessity that many individuals cooperate and work together to achieve their target. This behavior is in accordance with neighboring nodes of a sensor network acting as routers to other source nodes. If there are not many termites in a nest, then the pheromone content on the path to that nest will decay on time and less woodchip will be collected. Also, if we map this into a sensor network, if there are not many nodes on the path to the sink node, more packets will be dropped on the way to the sink. This might also be as a result of the low transmission distance of the nodes too. But if there exist more termites in the environment, the more woodchip will be gathered fast

to avoid complete pheromone decay in the shortest path, else termites agents will have to keep taking successive random steps in the network without any reasonable solution as regards finding the best paths to the destination. Though, many termites in the environment have a side effect as further discussed and explained in Section 4.3.

4.2.5 Stigmergy

This is the indirect communications between individuals of the social insect, generally through their environment. Complexity in stigmergy systems is due to the fact that, individuals or constituents of a system do not interact within themselves rather they do so with their environment. This behavior or actions lead to changes in the environment where they interact. Due to the changes, there is an advance effect on further behaviors, which give rise to a positive feedback effect where events are dependent on other events. This behavior is similar to the interaction between people in their environment based on their response to other people's comments during conversations. Termites are biased towards the hill with high concentration of pheromone, and it is not important that termites communicate directly with themselves or to know where about each other's. As such, termites are allowed to cooperate and interact with one another, and it is the main issue behind stigmergy.

4.3 Simulating the behavior of termites

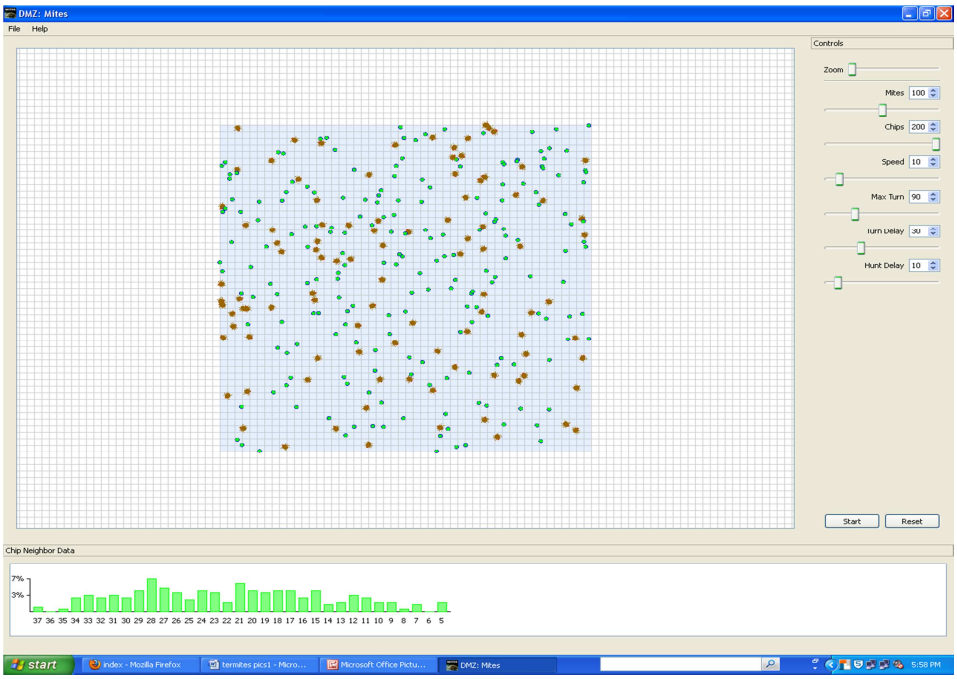
Simulation is the imitation representation of the functioning of one system or process by means of the functioning of another. Many computer simulations try to imitate some real-world systems or processes as accurately as possible. Though, in many cases, computer simulations are used to make predictions about real-world processes. In this section, we program artificial termites so as to investigate their termite-like behaviors. The main target is to simulate the termite world, and this will probe some

challenges that will be helpful in solving the routing problem in wireless sensor networks. Thus with the increasing interest in social insects, lots of people tend to be fascinated with the general behavior of ants. Thus an ant as an individual performs a simple act, but with the collection of many (colony), they perform a rather sophisticated behavior. Thus termites as a subset of social insects have come to be viewed as a prototypical example of how complex group behavior can arise from simple individual behavior. As such, the relationship between the colony and termites can be seen as an illuminating model, or at least an inspiring metaphor for thinking about other groups or individual relationships, such as the relationship between an organ and its cells, a cell and its macromolecules, a corporation and its employees, or a country and its citizens (Resnick, 1994, 1997).

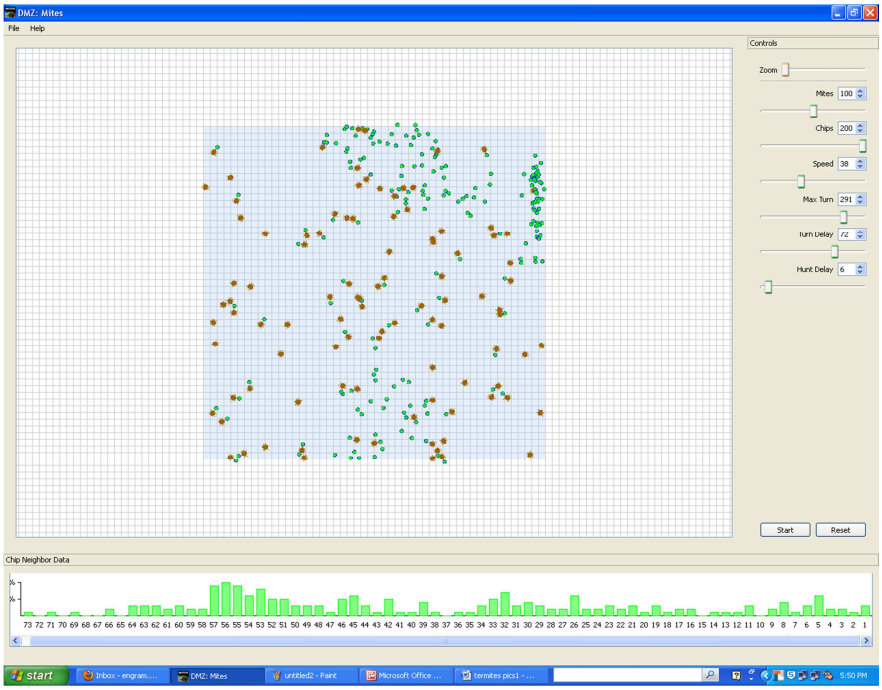
Again, each termite colony has a queen, unlike the ant system whereby the queen serves as leader. The termite queen does not give directives to the workers as in ants. Though, "Queen" seems to imply "Leader", but it is more of a mother than a leader to the colony. Detail explanation of this relationship is given in Section 4.5. It is worth knowing that on the termite hill building site, the termite has no site engineer (leader), that is to say that there is no one to take control of the master plan. Even with that, each of the individual termites carries out a specific simple task. Being practically blind and they must interact, and they do that through their senses of smell and touch. Through their local interactions, an important feature opens up. The principle of hill building, through cooperative behavior without site engineer to give directives, make them suited for solving the routing problem in sensor network where information is expected to be gathered in one place (sink). This means that simulating the construction of an entire termite nest will give more insight on their behavior, and thus can easily be mapped to simulating the sensor network. As such, in this section, we program some artificial termites to collect wood

samples, and the wood samples are expected to be gathered into particular sites (hills). Though, real termites do not actually carry wood samples from place to place, rather, they eat pieces of woods, then build hills with the faces they produce from the digested wood. The main challenge which is the motivating factor in this work is how to figure out a decentralized strategy for adding some order to a disordered collection of wood samples. Initially, the wood samples are randomly distributed throughout the termites' environment, but as the program runs, the termites are expected to organize the wood samples into a few orderly piles. With this model, we could map this to sensor network of which sensor nodes are distributed haphazardly with the aim to sense their environment and to gather the sensed event into one place (sink). Following the four (4) rules bounded by each termite as proposed in Section 4.2, we then program the termites to gather the disordered wood samples into an ordered form, and in fewer number of piles. In this program of termites piling up the disordered wood samples into order, we wrote set of programs for different functions. This set of functions includes: (1) defining sets of variables and initializing the global variables and functions. This includes the number of woods needed in the termites' world to the dimension of the termites' environment. In this, the number of woods equals the number of potential hills in the environment. (2) A function to distribute the wood and termites in the simulation environment. (3) Function definitions. (4) Function to make the termites move in the simulation environment, and (5) function to make termites pick up and put down the woods in a pile. As the termites pick up woods and look for piles, they do so in an orderly manner in which they put down wood samples only at a place where there exist at least a sample of wood. That is to say that, they do not put down the wood samples in an empty space. This process leads to the gathering of wood into fewer piles. If all of the wood samples from a particular pile are by chance removed

completely, due to the fact that all the wood samples are completely removed from that point, it then means that termites will never drop any wood sample on that spot. This means that, that particular hill will not by chance grow again. If there happen to be an existence of a pile or hill, its size will have the probability of increasing or decreasing, though, the existence of a pile once gone, it is gone forever. With this behavior, termites are able to gather the disordered wood into ordered forms. As an example of this behavior, we simulate 50 termites and 100 woods, and later increased it to 100 termites and 200 wood samples in a 200m by 200m DMZ (Demilitarized Zone) application environment. Fig. 4.2 (a) and 4.2 (b) shows the simulation at start and when the piles shrink to minima. In Fig. 4.3, we show the results gotten from the simulation. The graph shows clearly their behavior with respect to simulation time and number of termites in the environment to gather the widely dispersed woods.



(a)



(b)

Fig. 4.2: Simulation of Termites with 100 termites and 200 wood samples (a) start with 200 piles and (b) fewer piles.

As described above, the following algorithm Pseudo Code (Pseudo Code 4.1) explains the process of the program.

Pseudocode 4.1: Simulation of artificial termites in a real-world behavior Algorithm

---

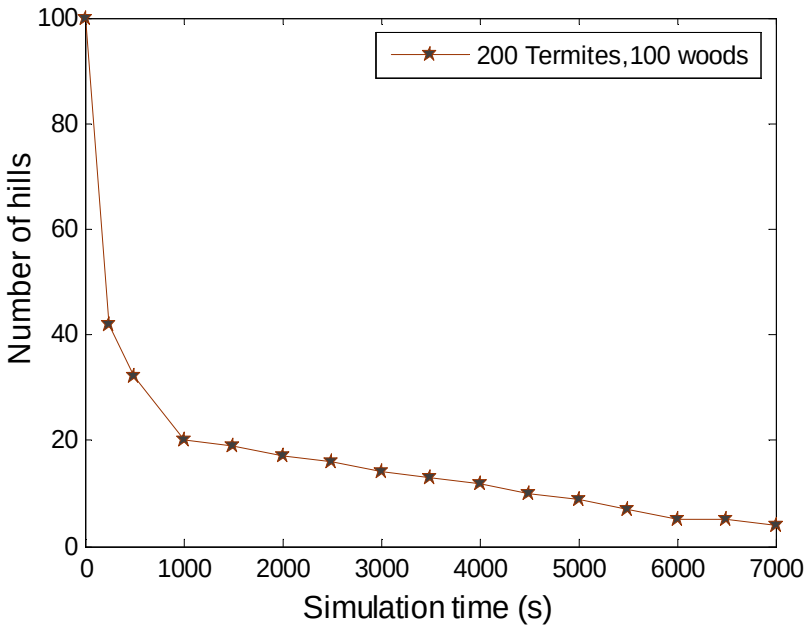
```
1. //Termite's real world behavior:
2. //Define variable and Initialization
3.  int Termite=50;
4. int woodchip=100;
5. pile-name;
6. Pile-wood-count;          //indicates the number of woods each pile is
   constructed with.
7. int Number-of-pile ;
8. Int x=200, y=200;         // the environment in which termite and wood
   chips are distributed.
9.  //functions' prototype
10. Distribute_wood (wood_chip,x,y);
11. View_wood ();
12. Distribute_termite (Termite,Distribute_wood,x,y);
```

```
13. Termite-move();
14. //main
15. Void main (){
16.   Distribute_wood;
17.   Distribute_Termite;
18.   //call Distribute_wood function for distributing woods randomly
19.   Distribute_wood=Distribute_wood (wood_chip,x,y);
20.   //Distribute_ termite function is in charge of distributing termite and
      pick up and put
21.   down the woods
22.   Distribute_termite=Distribute_termite (Termite,Distribute_wood,x,y);
23.   Print ("number of pile = " number-of-pile);
24.   Print (Pile$i,Pile$i.pile-wood-count)
25.   Wood-in-piles=0
26.   For (c=0;c<number-of-pile,c++){
27.           Int Wood-in-piles =Wood-in-piles +Pile$c.pile-wood-count
28.    }
29.   Print Wood-in-piles;
30. }
31. // Functions' definition:
32. Function Distribute_wood (wood_chip,x,y){
33.   int number-of-wood=0;
34.   for(int i = 1; i <= woodchip; i++)
35.   {
36.           L1:     int wood-x  ← choose random number between 0 to x;
37.                   int wood-y  ← choose random number between 0 to y;
38.                     check (x,y) ;
39.                   if  the place is empty{
40.                         put wood there
41.                         number-of-wood=number-of-wood++;
42.                    Number-of-pile =    number-of-wood;
43.                        /*in the initialization step, each wood determines a
      pile, and therefore,
44.                 when we find an empty place for wood,
45.                 we should keep the coordinate in the array for storing the
46.                 pile's location*/
47.                         matrix[wood-x][wood-y]="Pile$i";
48.                           pile-name[i] = "Pile$i";
```

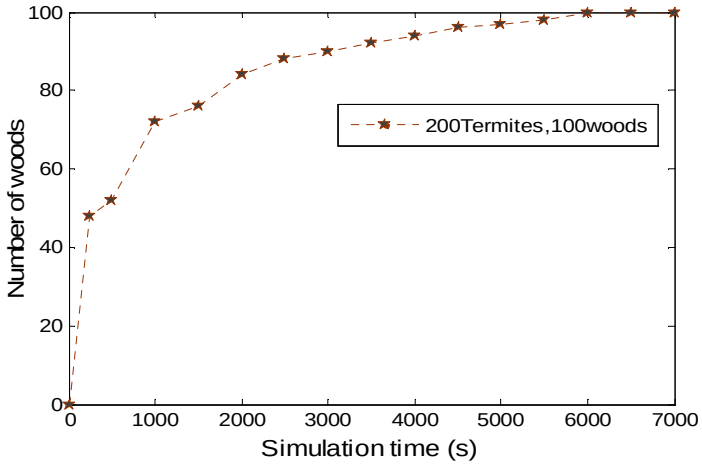
```
49.                pile-wood-count[$i] = 1;
50.                   }
51.                   else
52.                         goto L1;
53.  }
54.  Return   number-of-wood;
55.}
```

```
56. Function Distribute_termite (Termite,Distribute_wood,x,y){
57.  int  pick_up_wood = 0 ; //indicate how many woods are been carried
     by termites
58.   // distributes the termite in the environment
59.  For(i= 0 ; i<Termite;i++){
60.          L2:  int Termite-x  ← choose random number between 0 to x;
61.                int Termite-y  ← choose random number between 0 to y;
62.                   check (x,y) ;
63.                   if the place is empty Put termite in (x,y)
64.                  else goto L2 ;
65.   }
66. While (simulation's time > 0){
67.   //Termites should keep moving until they find a wood
68.   Termite-move();
69.  // termite find a wood
70.  pick_up_wood = pick_up_wood + 1;
71.  Pile$i.pile-wood-count=Pile$i.pile-wood-count - 1;
72.  If (Pile$i.pile-wood-count < 1){
73.     Delete Pile$i;
74.     Number-of-pile = Number-of-pile-1;
75.   }
76.// termite should keep a random movement until they find another
     wood and put down this
77.  one near it.
78.  Termite-move();
79.  Select the nearest empty place
80.  Put the wood;
81.  Pile$i.pile-wood-count=pile-wood-count+1;
82. Termite-move();
83.}
```

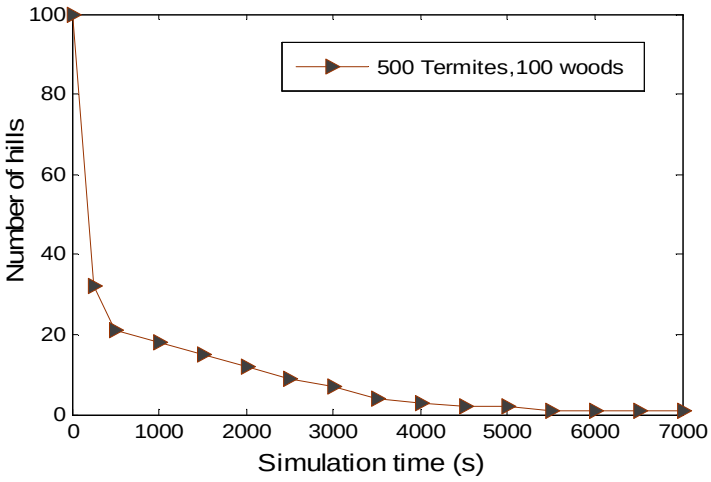
```
84. Function Termite-move(){
85. Int row = Termite-x;
86. Int col  = Termite-y;
87. L4:
88.         For (row; row<x;row++){
89.         For (col;col<y;col++){
90.             If ((row,col ) == (wood-x,wood-y)){
91.                 Return (Pile$i);
92.                  Break;
93.             }
94.         }
95.     }
96. // that means termite did not find wood and it reaches (200,200)
97. Col=0;
98. Row=0;
99. Goto L4;
100.    }
```



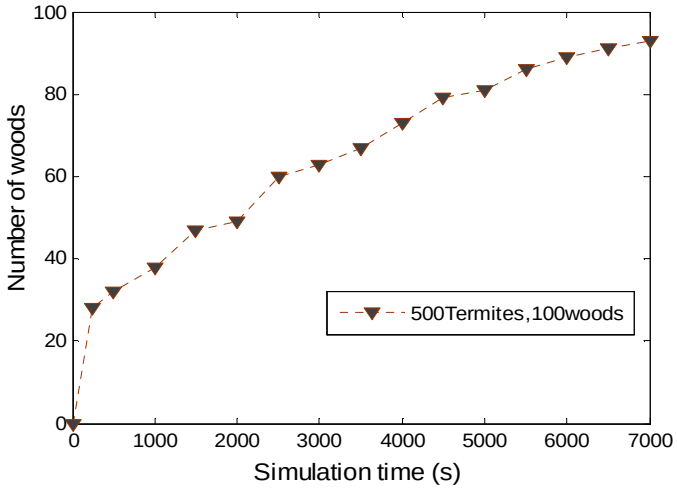
(a)



(b)



(c)



(d)

Fig. 4.3: Behavioral pattern of termites gathering wood samples in the presence of variation in simulation time with respect to (a) fewer termites and number of hills built (b) fewer termites and number of woods gathered (c) more termites and number of hills built and (d) more termites and number of woods gathered.

At the initial runs of the program when we run the simulation, termites put down the wood samples next to each other wood sample (see Fig. 4.2), rather than on top of other samples in the normal hill building. But to us, it is still fine as long as the wood samples are gathered into one or fewer piles. Though, piles are not clearly defined as termites pick up wood samples from the middle of well-established piles such that what are remained could be seen as two or more piles with one pile of higher concentration of woods. Though, as we keep running the simulation, initially, the wood samples are gathered into tenths of samples, but as the simulation proceeds, the number of samples per pile increases while the number of piles in the environment decreases. This action can be seen in Fig. 4.3 (a-d). It was also observed that, after 1000 seconds of simulation time with 200 termites randomly distributed along with 100 wood samples in the environment, there were about 20 piles of the woods out of the initial piles of 100, with a total of 72 woods. After 5000 seconds, 9 piles were recorded with a total of 97 woods, and this continues, and as the simulation time proceeds to 7000 seconds, 4 piles were recorded and a total of 100 woods was also recorded. However, it then means that with more time of simulation, the number of piles shrinks to just a single whole pile. Also, the number of woods gathered (success rate) tends to 100%. This is shown in Fig. 4.3 (a) and (b). But the shrinking of the piles as observed with fewer termites of 200 was not a fast one as expected. We then increased the number of termites to 500 as against the original 200 in the first case. In this, after 1000 seconds, we recorded 18 piles with a total of 38 woods, and after 5000 seconds, we recorded 2 piles with a total of 81 woods, and also at 7000 seconds, a large pile of 1 was recorded with 92 woods, which implies that 7 woods were still carried by some termites. This is expected since in most cases, with a high population of termites in the environment, the piles shrink faster which means that the latency reduces

and the success rate diminishes along. In all, the ability of termites to gather woods in fewer piles is the convergence of the network when we have more termites in the environment. But there is a threshold for the number of termites as observed in the experiment for congestion control and to avoid the low success rate level. It was also observed in the experiment that as we increase the number of termites above 5 times the number of wood samples, the environment gets congested and all woods are carried by the termites of which it becomes difficult for them to form any reasonable pile. With these observations in their behavior, we also testify the assumption we made in Chapter 3 in the reduction of ants in the network for congestion control. With this behavior and observations, we then map our findings into sensor network. Section 4.4 explained in detail the mapping process and the steps adopted in modelling the termites' agents for solving the routing problem in sensor networks.

4.4 The Termite-hill Routing Algorithm

The termite-hill is a routing algorithm for wireless sensor networks (WSNs) that is inspired by the termite behaviors. The principles of swarm intelligence are used to define rules for each packet to follow in the network which results in emergent routing behavior. The algorithm finds its way for better performance in the reduction of control traffic, quick route discovery and repair, utilization of energy as a criterion for route selection, and reduction of memory usage along with other additional benefits. Analogous to the termite ad-hoc networking (Roth and Wicker, 2003), each node serves as a router and a source, and the hill is a specialized node called sink which can be one or more depending on the network size. Depending on the network, each network node can also serve as a termite hill. Packets traverse around the network of paths joining sensor nodes and they do that based on the pheromone concentration on each path. The

information contents of each sensor from the source node are transferred to the sink node, and the movement of these packets to the sink node is more biased towards the link with higher concentration of pheromone. When these packets are dispatched to the sink node, the path taken is also influenced with some additional pheromone which will in turn increase the concentration of pheromone on that path. If we consider the termites' wood chips collection behavior, packets move more to the path with high concentration of pheromone, so also is the selection of the next path to be taken which is randomly decided upon. As more and more termites took a path to the destination point where wood chips are gathered, more pheromone is deposited on that path and more packets are likely to take that path as a reverse direction to the source node, which is a positive feedback. That is, to avoid outdated routing solutions remaining in the memory of the network, time based decay is introduced as a negative feedback which is a percentage reduction of the pheromone on the path, and also depends on the area of application. Pheromone increases as data packets increases, but decreases with time.

Termite-hill discovers routes only when they are required. When a node has some events or data to be relayed to a sink node and it does not have the valid routing table entry, it generates a *forward soldier* and broadcasts it to all its neighbors. When an intermediate node receives this *forward soldier*, it searches its local routing table for a valid route to the requested destination. If the search is successful, the receiving node then generates a *backward soldier* packet, which is then sent as a unicast message back to the source node where the original request was originated using the reverse links. If the node has no valid route to the destination, it sets up a reverse link to the node from which the *forward soldier* was received and further broadcasts the *forward soldier* packet. When the destination node receives the *forward soldier* packet, it generates a *backward soldier* packet

which is also unicast back to the source node. On reception of the backward *soldier* packet, each intermediate node updates its routing table to set up a forward pointer and relays the *backward soldier* message to the next hop using the reverse pointer. The process continues till the *backward soldier* is received by the original source node. For Termite-hill algorithm for WSNs, *HELLO* packets are not used for detecting path failures or validity of paths. But in replacement of that, Termite-hill makes use of feedback from the link layer-medium access control layer for achieving the same objective. It is also worth knowing that intermediate nodes are not responsible for generating *backward soldier* even if they have a valid path to the end node (sink). This process is to make sure that excessive overhead is avoided in the network which might be as a result of multiple replies. Also, Termite-hill uses cross layer method to avoid the use of paths with high data loss rate. As such the termite-hill is designed to function in three modules. In the course of the algorithm design, the following assumptions were also made: 1. each node is linked to one or more nodes in the network (neighbors), 2. A node may act as a source, a destination, or a router for a communication between different pairs of nodes, 3. Neither network configuration nor adjacency information is known beforehand, and 4. The same amount of power is required for sending a message between any pair of adjacent nodes throughout the network.

4.4.1 The Pheromone Table

The pheromone table keeps the information gathered by the forward soldier. Each sensor node keeps a table which contains the quantity of pheromone on each neighbor path. The sensor nodes have a different pheromone scent, and the table is in the form of a matrix with destination nodes listed along the side and neighbor nodes listed across the top. Rows correspond to destinations and columns to neighbors. An entry in the

pheromone table is referenced by $T_{n,d}$ where *n* is the neighbor index and *d* denotes the destination index. The values in the pheromone table are used to calculate the selecting probabilities of each neighbor. When a packet arrives at node *G* from previous hop *S*, i.e. the source, the source pheromone decay, and pheromone is added to the link $\overrightarrow{SG}$. Backward soldier on their way back from the sink node are more likely to take through *G,* since it is the shortest path to the destination i.e. $\overrightarrow{SGED}$. The pheromone table of node *G* is shown in Fig. 4.4 with nodes A, S, F, and E as its neighbor. It is worth noting that all neighbors are potential des tination. At node G, the total probability of selecting links $\overrightarrow{ED}$, $\overrightarrow{FE}$, $\overrightarrow{AC}$ or $\overrightarrow{SB}$ to the destination node is equal to unity (1) i.e. $\sum T_{ED} + T_{SD} + T_{AD} + T_{FD} = 1$. It will then be observed that, since link $\overrightarrow{ED}$ is shorter, more pheromone will be present on it and hence, soldiers are more likely to take that path.

Destination \ Neighbor	A	S	F	E
C	$T_{A,C}$	$T_{S,C}$	$T_{F,C}$	$T_{E,C}$
B	$T_{A,B}$	$T_{S,B}$	$T_{F,B}$	$T_{E,B}$
E	$T_{A,E}$	$T_{S,E}$	$T_{F,E}$	$T_{E,E}$
D	$T_{A,D}$	$T_{S,D}$	$T_{F,D}$	$T_{E,D}$
S	$T_{A,S}$	$T_{S,S}$	$T_{F,S}$	$T_{E,S}$
A	$T_{A,A}$	$T_{S,A}$	$T_{F,A}$	$T_{E,A}$
F	$T_{F,A}$	$T_{S,F}$	$T_{F,F}$	$T_{E,F}$

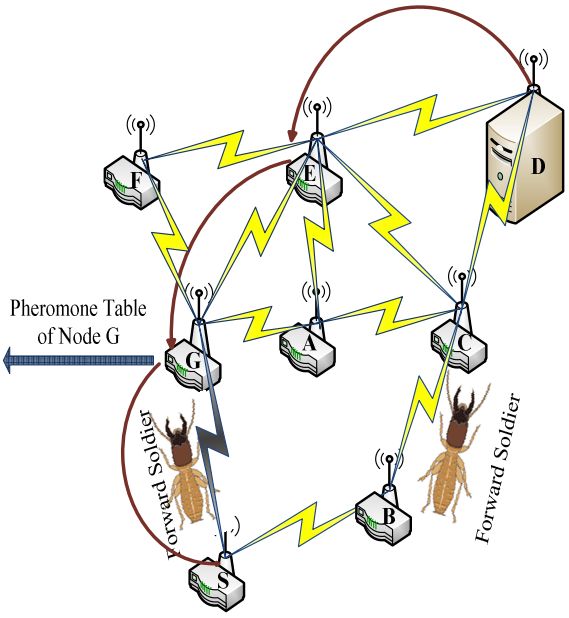


Fig. 4.4: Description of pheromone table of node G.

4.4.1.1 Pheromone Update

When a data packet reaches a sensor node, the pheromone content for the source of the information is increased using γ factor, where γ represent the reward for the source node. Data packets which are not made for a node will not be processed by the node. But a node can be said to be addressed if it happens that it is selected as the next hop node to be visited according to the information contained in the packet. Equation (4.1) describes the pheromone update procedure when a packet from source *s* is delivered from previous hop *r*. A prime indicates the updated value.

$$T'_{r,s} = T_{r,s} + \gamma \tag{4.1}$$

And,

$$\gamma = \frac{N}{E - \left(\frac{E_{min} - N_j}{E_{av} - N_j}\right)} \tag{4.2}$$

Where *E* is the initial average energy of the nodes, E_{min}, E_{av} are the minimum and average energy respectively of the path traversed by the forward soldier as it moves towards the hill. The values of E_{min} and E_{av}, depends on the number of nodes on the path and the energy consumed by the nodes on the path during transmission and reception of packets. The minimum energy of the path (E_{min}) can be less than the number of nodes visited by the forward soldier, but the average energy of the path (E_{av}) can never be less than the number of nodes visited. N_j, represents the number of nodes that the forward soldier has visited, and *N* is the total number of network nodes.

4.4.1.2 Pheromone Evaporation

Pheromone is evaporated so as to build a good solution in the network. Each value in the pheromone table is periodically multiplied by the evaporation factor $e^{-\rho}$. The evaporation rate is $\rho \geq 0$. A high evaporation rate will quickly reduce the amount of remaining pheromone, while a low value will degrade the pheromone slowly. The nominal pheromone

evaporation interval is one second; this is called the decay period. Equation (4.3) describes the pheromone decay.

$$T'_{n,d} = T_{n,d} * e^{-\rho} \tag{4.3}$$

Though for robustness and flexibility some application needs a slow decay rate, and some applications like security and target tracking applications, need fast decay process and will determine the value of the decay period. That is, the value of ρ and x in equation (4.4) depends on the application area. Hence to account for the pheromone decay each value in the pheromone table is periodically subtracted from the percentage of the original value as shown in equation (4.4).

$$T'_{n,d} = (1 - x)T_{n,d} \tag{4.4}$$

Where, $0 \leq x \leq 1$

If by any means all of the pheromone for a specific sensor node evaporated, then the row and or column in which the information for that particular node is contained is completely removed from the table. If the entry for a node on a pheromone table is removed, it indicates that for quite some times, information has not been received from that particular node. That either means that it is not within the area of coverage of that node or it does not exist anymore. In other words, for a column to be considered decayed, it means that all the pheromone in it is equal to the lowest pheromone value. Also, a row is considered the same if all the pheromone content in it is equal to the lowest pheromone. It is worth mentioning that neighbor node of a source node should be given preferential treatment as they are likely to be destination in addition to the task given to them as routers as well as serving as a source of information. If by any means a neighbor is lost due to communication failure, that is to say that it is out of communication range, and the column corresponding to that neighbor is removed from the pheromone table.

4.4.1.3 Pheromone Limits

The limit of the pheromone table is bounded by three values which basically are: (1) the *upper pheromone*, (2) the *lower pheromone*, and (3) the *initial pheromone*. When a data packet is received at a node from the node that is not known to it, an entry for it is created in that receiving node pheromone table. The entries consist of a column and a row. If the information received about the node tells that it is a neighbor, a column in addition to a row is created for it, otherwise, only a row is created in the case that it is not a neighbor. The cells created will be initialized with the initial pheromone values. When pheromone is to be evaporated, the value is never allowed to enter the critical value which is normally the lowest pheromone value. This is done to make sure that nodes that are hardly used are detected. Also, no value is permitted to be more than the upper value. These limits help in safeguarding the pheromone difference from affecting the calculation of the probabilities of the next hop selection. Though, each parameter may be chosen based on the network environments and requirements.

4.4.2 Route Selection

Each of the routing tables of the nodes is initialized with a uniform probability distribution given as:

$$P_{s,d} = \frac{1}{N} \tag{4.5}$$

Where $P_{s,d}$ is the probability of jumping from node *s* to node *d* (destination), N the number of nodes in the network.

Upon arrival at a node *s*, an incoming packet with destination *d* is routed randomly based on the amount of *d's* pheromone present on the neighbor links of *s*. A packet is never forwarded to the same neighbor from whom it was received, *the previous node*. If *s* has only one neighbor, i.e. the node that the packet was just received from, the packet is dropped.

The equation below details the transformation of pheromone for *d* on link *s*, $T_{s,d}$ into the probability $P_{s,d}$ that the packet will be forwarded to *d*.

$$P_{s,d} = \frac{(T_{s,d}+\alpha)^{\beta}}{\sum_{i=1}^{N}(T_{i,d}+\alpha)^{\beta}} \quad (4.6)$$

The parameters α and β are used to fine-tune the routing behavior of Termite-hill. The value of α determines the sensitivity of the probability calculations to small amounts of pheromone, $\alpha \geq 0$ and the real value of α is zero. Similarly, $0 \leq \beta \leq 2$ is used to modulate the differences between pheromone amounts, and the real value of β is two. But for each of the *N* entries in the node *k* routing table, it will be N_k values of $P_{s,d}$ subject to the condition:

$$\sum_{s \in N_k} P_{s,d} = 1; \; d = 1, \ldots, N \quad (4.7)$$

4.4.3 Termite-hill Agent Model

This section describes the principles of termites' behavior that inspired us to transform them to an agent model. The termites evaluate the quality of each discovered path to a hill by the pheromone contents of the pebbles on the path (Mackenzie and Wicker, 2001). This means that, not all the discovered path receives reinforcement. *Termite-hills* work with three types of agents: reproductive, soldiers and workers.

4.4.3.1 Reproductive

In termite colonies, the male reproductive "*king*" and the female reproductive "*queen*" are the primary source of pheromones useful in colony integration, and these are thought to be spread through shared feeds. Analogous to that, the *queen* receives data packets from the upper layer and locates an appropriate worker (route) for them at the source node. When a worker is found, the packet is encapsulated in its payload and the queen starts waiting for the next packet. But in a situation when a worker is not found, it is an indication to the queen that no route exists for

the sink. At the sink node, the *king* recovers data from the payload of workers (routes) and provides it to the upper layer.

4.4.3.2 Soldiers

Soldiers in *Termite-hill* are grouped into two as *forward soldiers* and *backward soldiers*, which depends on their movements in the network. Soldiers leaving the source to the sink node are named *forward soldiers* while those from the destination to the source node are labeled *backward soldiers*. At any point in time when a sensor node in the network sensed an information about an event and is not able to locate a valid worker to encapsulate the information in its payload to deliver it to the destination node, it launches a *forward soldier* to all its neighbors with the mission to find an appropriate worker for the destination node. The soldiers are given different identification, and each goes along with the sensed information on them towards the destination node. In Termite-hill, soldiers have no source routing headers for which information about the visited nodes up to the sink node is saved. That means that, soldiers have got a fixed size that is independent of the path length, that is, the number of hops taken between the source and a sink node. This helps Termite-hill to scale to large networks. As adopted in ant-based routing (Camilo et al., 2006; Zungeru et al., 2011b), the memory M_k of each soldier is reduced to just two records; the last two visited nodes. Since the path followed by the soldiers is no more in their memories, a memory must be created on each node that keeps a record of each soldier that was received and sent. Each memory record saves the previous node, the forward node, the soldier identification and a timeout value. Whenever a *forward soldier* is received at any node, it searches for any possible loop with the aid of its identification (*ID*). For the situation where no record is found, the necessary information is retrieved and the timer restarted, hence forwarding the soldier to the next node, else, the soldier is eliminated if a

record containing its identification is found. Once *forward soldier* reaches the sink node, it then update the pheromone trail of the path it used to reach the destination and that is stored in its memory. It then starts going back to where it was created (source node) as a *backward soldier.* But before the *backward soldier* starts its return journey, the destination node computes the amount of pheromone trail that the *soldier* will drop during its journey. When this *backward soldier* reaches the node where it was created, its mission is accomplished and is eliminated.

4.4.3.3 Workers

In *Termite-hill*, workers undertake the labors of foraging. Analogous to the worker's behavior in real termite organization whereby the workers feed the other members of the colony with substances derived from the digestion of plant material, either from the mouth or anus (Zungeru et al., 2012b), they transport data packets from a source node to the sink node. They receive data packets from *queen* at a source node and deliver them to the *king* at the sink node. The basic motivation in *Termite-hill* is to find multiple route to a destination node where the hill is being built. Each worker in the network has different identification with the help of their identification and path identification. A worker hops from node to node until it arrives at the destination node. A worker is normally piggybacked to the node where it was created in response to the information from the data link layer. That is to say that, the overhead to be incurred in sending workers using a clearly defined network layer transmission is avoided. In the case of arrival of the workers to the node where they were created, the amount of pheromone in association to the hops used by the workers is computed in the same manner as that of the *backward soldier*. As such, better routes are found and utilized. In *Termite-hill* small forwarding tables are created at each nodes in-between the source and the sink node (intermediate

node), which help to simulate the behavior of a sender specifying the route by which information should follow in the network (source routing).

4.4.4 The Termite-hill Routing Algorithm Modules Design

The algorithm is designed to function as three main modules*: route discovery, seed, and data.* The first two modules are control messages. The data packets are used for sending data or events to the other nodes, and the control packets are mainly used for the maintenance of the route as well as increasing the pheromone concentration of the link. Each packet type contains at least six fields, including *message identification, source address*, *destination address*, *previous hop address*, *next hop address*, and *Time-To-Live (TTL)*. Though, the addresses of hop before (previous hop) and hop to be decided upon (next hop) can be extracted from the medium access control header which carries the routing data, and as such can be removed from the packet. Thus, the message ID cell makes it easier for fast identification of different packets of information in the network. The message identification is normally used for defining the type of the packets. These types are numbered from 0 through 3, where each one stands for forward soldier, backward soldier, data and data acknowledgement packet respectively.

4.4.4.1 Route Discovery Module

This module main function is to generate routing path between source nodes and destination node. The *forward soldiers* are normally sent when there is an information/message available at a source node, and a path to the sink node is not known. The *forward soldier* takes successive random steps in the network with the mission to find a node which has information about the sink node based on the pheromone content of the node. In random steps, a packet uniformly randomly chooses its next hop, except for the link it arrived at. In a situation whereby a *forward soldier* cannot be

forwarded, it is discarded. Though, the use of pheromone in path selection is not considered during the successive step movement. Any number of *forward soldiers* are allowed to be forwarded in search for a path (path discovery), and the actual value is dependent on the particular environment and application intended. It is worth knowing that a forward soldier is not looking for an exact path to the sink node, but it is looking for the initial path to the sink node. Though, the path will be stronger as more pheromone is deposited on it due to future communications. Fig. 4.5 shows the path discovery as *forward soldiers* were sent to an unknown destination by node *S.* The *soldiers* walk through the network in order to find a path to the sink node *D.*

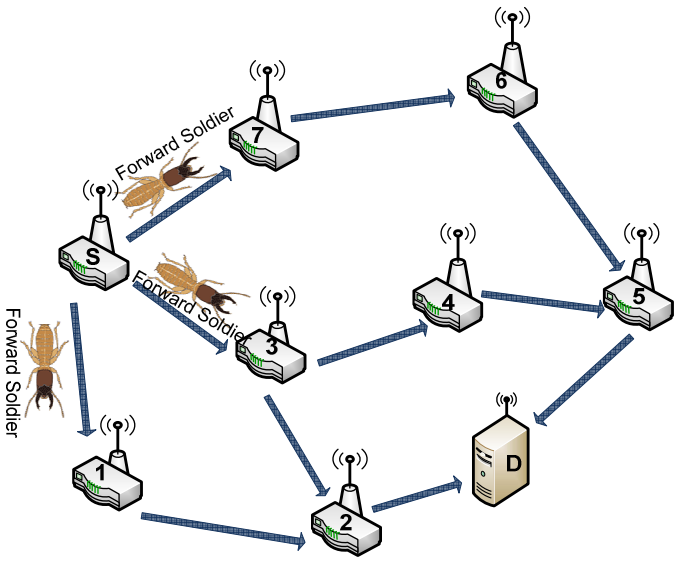


Fig. 4.5: Forward Soldier from node S to destination D.

Whenever information is sensed by a sensor node, the sensed information is given to a queen whose main function is to locate an appropriate worker to move the information to the destination node. But if it happens that the queen has difficulty in locating an appropriate worker, a forward soldier is launched and the information is loaded into its payload. The forward soldier carries along five information fields in addition to the message type field. The field carries by the soldier includes; the soldier identification, minimum residual energy of the path taken by it, the average energy of the path,

source node identification, and the number of hops traversed by the soldier, in which it is initially set to zero. A destination node that has an interest on any information with the soldier will then turn the forward soldier to a *backward soldier* with the mission of sending it back to the source node where the *forward soldier* was created, and along, update the path taken by the forward soldier. If along the network a node *m* who is neither the source of the information nor the sink node receives a forward soldier from a node *n* for the first time, it adds a value of one to the hop field of the soldier. Node *m* further decides whether to transfer the forward soldier to its neighbors as the next hop or not. If node *m* is at $\mathcal{H}_{max}$ or less hops count away from the node where the forward soldier was initiated, it considers to re-transfers the forward soldier without a second thought Else, it re-transfers the forward soldier with a broadcast forwarding probability (P_b). $\mathcal{H}_{max}$ and P_b define an exchange of route discovery probability return for energy efficiency and vice-versa. $\mathcal{H}_{max}$ denotes the present estimate of the hops count between a pair of sensor node and the destination node. Furthermore, node *m* compares the minimum energy of the nodes in its path with the energy remaining in it. The value of the minimum is updated as the minimum energy in the forward soldier, and the average of the total nodes energy of the path is also recorded and carries by the forward soldier. Node *m* finally creates a soldier cache entry in the table which now encompasses information like the source node identification, soldier identification, previous hop, rebroadcasting the signal which is normally equal to one if node *m* has an intention to re-transfer, and otherwise, its value equals to zero. The reward (γ) of the path is calculated with the aid of the minimum energy of the path from node *m* to node s (E_{min}), the average energy of the path (E_{av}) and N_j is the number of hops from node *m* to source s. The intuition behind this (Eqn. 4.2) is to prefer the least hop path. For a situation whereby there exist paths having the same number of

hops, it will be preferable to take the path with the highest amount of minimum energy on its link. When the soldier cache is updated, node *m* rebroadcasts the forward soldier to the chosen next hop node according to the decision of selection if the re-transfer signal is set in the soldier cache. Else, the soldier is destroyed and also its memory contents. The procedure simultaneously repeated within the intermediate nodes until the forward soldier gets to the destination node. The process of forward soldier is shown in Pseudo Code 4.2.

Termite-hill algorithm performs well due to the information transfer between copies of the same agent which might get to a particular node which is neither a source nor sink (i.e. intermediate) node using a different route. Whenever a node receives a copy of the same message it has received earlier on with the same source and soldier identification, it takes some major steps to choose any of the copies which includes checking to make sure whether the first copy of the soldier identification has been forwarded or not. It is worth noting that a signal message (flag) is maintained in the soldier cache which handles this kind of situation. In the case that the initial copy of the message is dropped, the second copy which now serves as a duplicate copy is then eliminated without the need to process it. This action is necessary since it is not beneficial to keep the content of this copy at node *m* for routing decision as there is no backward soldier to make use of the information since they will not visit this particular node. If the reverse is the case meaning that node *m* has earlier transfer the copy to its neighbors, it then updates the hop count and the energy cells of the new copy received unconditionally and hence, the value of γ is calculated. In a situation whereby the value of the new γ is higher than the value contained in the cache of the soldier for the previous hop, the cell carrying γ is updated. That is to say that the route having the highest γ is chosen as the path over the low value ones. Hence, the new

copy is destroyed. These processes of in-house communication and negotiation enables better path discovery by the algorithm without incurring any additional overhead. It is also pointed out here that the soldier cache entries are just for a short period of time, and if by any means the expected backward soldier fails to reach the source node where it was created within this time, the entry is deleted. The route discovery algorithms are described as in Pseudo Code 4.2 and 4.3:

Pseudocode 4.2: Route Discovery in Termite-hill Routing Algorithm

1. Required: A copy of Forward Soldier (FS)
2. **if** (SinkNode) **then**
3. // Upload Payload and pass to the application layer
4. PayloadToApplication (FS);
5. UpdateForwardingTable (FS.From, NodeID, PathID);
6. // Construct a Backward Soldier and forward to FS.From
7. BS ← ConstructBackwardSoldier (BS);
8. Forward (BS, FS.From);
9. **else if** (NotSeenBefore (FS)) **then**
10. $N_j \leftarrow$ FS.Hops ← FS.Hops + 1;
11. **if** (FS.Hops ≤ H_{max}) **then**
12. // Set Broadcast Flag
13. BFlag ← 1;
14. **else**
15. BFlag ← StochasticForwarding ( );
16. **end if**
17. N ← TotalNetworkNodes ← (Node.Total);
18. E ← InitialNodesEnergy ← (Node.Energy);
19. E_{min} ← FS.MinEnergy ← Min (FS.MinEnergy, Node.Energy.Min);
20. E_{av} ← FS.AvEnergy ← Av (FS.AvEnergy, Node.Energy.Av);
21. $\gamma \leftarrow \frac{N}{E-\left(\frac{E_{min}-N_j}{E_{av}-N_j}\right)}$;
22. UpdateSoldierCache (FS.From, FS.SourceID, FS.SoldierID, BFlag, γ);
23. **if** (BFlag) **then**
24. Broadcast (FS);

```
25.    else
26.        DeleteForwardSoldier (FS);
27.    end if
28. else
29.    if  ( Forwarded(FS) )    then
30.        N_j ← FS.Hops ← FS.Hops + 1;
31.    E_min ← Min (FS.MinEnergy, Node.Energy.Min);
32.        E_av ← Av (FS.AvEnergy, Node.Energy.Av);
33.        γ ← N / (E − (E_min − N_j)/(E_av − N_j));
34.        if  (γ > RewardInSoldierCache ( )) then
35.        UpdateSoldierCache (FS.From, FS.SourceID, FS.SoldierID,
     BFlag, γ);
36.        end if
37.    end if
38.    DeleteForwardSoldier (FS);
39. end if
```

Once a *forward soldier* is received at a node that has the pheromone content of the destination node that makes the request, a *backward soldier* is created with a mission to take the reverse path and update the route taken by the forward soldier. The *backward soldier*s walk probabilistically in the network by taking the path to the source node. The nodes serving as intermediate nodes help in forwarding the backward soldier to the source node by following the field contained in the source identification. As can be seen in Fig. 4.6, the shortest and best path discovered by the *forward soldier* is being taken by the *backward soldier* on his return journey. Of course, path D→2→3→S is taken by the *backward soldier* as shown in Fig. 4.6.

Nodes in *Termite-hill* maintain three different types of tables which include (1) the routing table, (2) probability distribution table, and (3) the forwarding table. The first two tables are mostly resided at each source nodes, but the last of the three is carried by the destination node and the

nodes serving as routers within a route to the destination. In the event of the destination node receiving a forward soldier, the destination (sink) node uploads the information carried by the forward soldier, and this information is passed to the application. And a forward table cell is created for the forward soldier that mainly consists of three fields including (1) a special path identification, (2) next hop identification, and (3) previous hop identification. The second and the third fields are set to the destination node identification and identification of the node where the forward soldier is received respectively. The content of the actual data set to the lowest value, while also setting the minimum residual energy cell carried by the forward soldier header to the maximum energy of the network's nodes, and special path identification is added to the header of the forward soldier. After the whole process, the forward soldier is converted to the backward soldier by changing the soldier identification to signify that it is now a returning (backward) soldier. The soldier is then transferred along the path taken by the forward soldier, which is the path from which the forward soldier traversed before getting to the sink node. When an intermediate node *m* now received the backward soldier from a node *n*, it searches to see a matching soldier cache entry. In the case of relevant information found, a forwarding table is created with next hop set to *n*, path identification set to the value which is carried by the backward soldier on its header, and its hop before the present position set to the previous hop identification carried by the soldier in its cache. Hence, the area of the soldier fast memory (cache) is deleted, and the node weighs the difference between its own energy with the one carries by the backward soldier, and an update is made which is based on the minimum of the two values, and also calculates the average energy of the path traversed. It then forwards the backward soldier to the previous hop. Each intermediate node processes the backward soldier almost the same way until the backward

soldier arrives at the source node. This action is described in Pseudo Code 4.3.

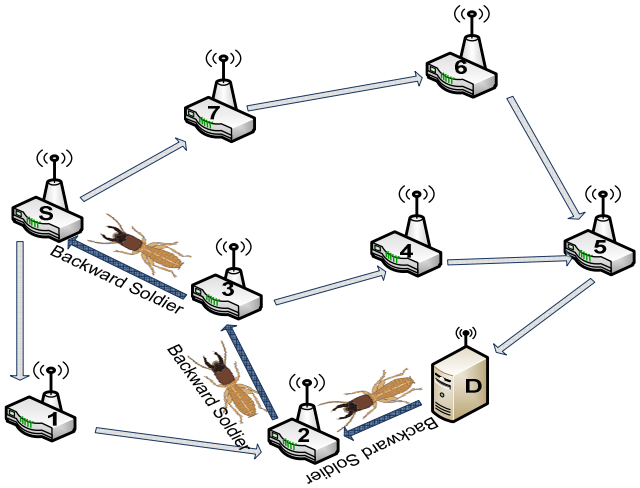


Fig. 4.6: Backward Soldier from Node D to source node S.

Pseudocode 4.3: Route Update in Termite-hill Routing Algorithm

```
1.  Required:  A copy of Backward Soldier (BS)
2.  if  (SourceNode) then
3.      T_rs ← CalculatePheromoneValue (BS.Pheromone);
4.      // Update the pheromone and probability tables
5.      UpdatePheromoneTable (BS.From, BS.SinkID, BS.PathID, T_rs);
6.      UpdateProbabilityTable (P_sd);
7.      DeleteBackwardSoldier (BS);
8.      // announce path to the neighbors
9.      BroadcastBeacon ( );
10. else
11.     // Check for matching BS if earlier forwarded
12.     if  (MatchInSoldierCache(BS))  then
13.         // Update the forwarding table
14.         UpdateForwardingTable (BS.From, SoldierCache, PrevHop,
        BS.PathID);
15.         Forward (BS, SoldierCache.PrevHop);
16.         DeleteSoldierCacheEntry (BS);
17.         BS.Pheromone ← (BS.Pheromone, Path.Pheromone);
18.     else
19.         DeleteBackwardSoldier (BS);
20.     end if
21. end if
```

4.4.2 Data Packet Module

These packets/events are transferred normally towards the destination node using the best path in the network. When a sensor node sensed an event or has information to be forwarded to the sink node and has no valid route to it which does happen when node pheromone content in the table does not carry the information of the destination where it is intended, the event/packet is kept for a short period of time in the cache, and a forward soldier is broadcasted towards the node's neighbors with the mission to find the possible route to the destination. It is expected that a backward soldier takes the return path and update the path traversed by the forward soldier. But in a situation whereby the backward soldier is not received within the time frame set for it, that is the forward soldier timeout, the event is dropped and hence considered lost. Whenever a route is discovered, the detected phenomena are transported via the payload of the workers to the sink node. An event cache is also maintained at the source node so as to store any sensed information during the route discovery. The queen in the network chooses a worker randomly, mounts the sensed event in its payload, and transferred it to the next hop using the path identification. The worker also takes a path which was determined earlier on, which means that intermediate nodes are not in any way to make decisions on the routing process, which is contrary to that of ant based algorithms whereby intermediate nodes make decisions on routing process by maintaining routing tables in addition to a random selection of the next hop. In *Termite-hill* algorithm, the intermediate nodes main function is to provide a link for forwarding the worker to the next hop node which is based on the identification of the path, and in turn helps to decrease of overhead incurred in the processing of information at the intermediate nodes. The table which contains the probabilities of selection is updated once a worker gets back to the source node. Forwarding of a worker at the

source node and intermediate nodes are summarized in Pseudo Code 4.4 and 4.5.

Pseudo Code 4.4: Working Group in Termite-hill Routing Algorithm

```
1.  Required: A Phenomenon for Transportation to Sink Node
2.  for all Phenomenon received from Application layer  do
3.      W = Worker ( );
4.      if  (W = = NULL)  then
5.          if  (RouteDiscoveringInProgress( ))  then
6.              // Route discovery in progress, wait in cache
7.              StorePayloadInCache (P);
8.          else
9.              // Route required, initiate forward soldier
10.             LaunchForwardSoldier (FS);
11.         end if
12.     else
13.         //Worker found, forward to next hop
14.         Forward (W, NextHop);
15.     end if
16. end for
```

Pseudo Code 4.5: Working Group at intermediate nodes in Termite-hill Routing Algorithm

```
1.  Required:  A Worker
2.  if  SinkNode( )  then
3.      PassToApplication (W.P);
4.      AddToWorkersList (W);
5.   else
6.      Next ← GetNextHop (W.PathID);
7.      Forward (W, Next);
8.  end if
```

4.4.3 Route Maintenance Module

This module's main function is to maintain the entire route that was detected in the course of the route discovery process. This constitutes the *Seeding packets:* Seed packets are used to piggyback the workers to the

nodes where they were created in response to events generated by the second layer of the OSI. Due to the resource constraints of sensor nodes and high dynamics of low-power wireless links, paths are highly error prone. Therefore, path reconstruction should be provided to reduce performance degradation. The seed packets are useful in verifying the validity of routes in the network. When a node fails to receive its worker within a time frame set for it from any path in the network, it therefore means that, that particular path to the destination node is not more valid, hence information regarding that path is deleted from the routing table. The route rediscovery process can be initiated in three different situations: (1) when an active path has failed, (2) when all the active paths have failed or, (3) when a certain number of active paths have failed. Since, the frequency of initiating route rediscovery process in the first approach is higher than for the two other approaches, using this strategy imposes a high overhead. Nevertheless, performing a route rediscovery process after the failure of all the active paths may significantly reduce network performance. Therefore, we concentrate on the third approach which represents a trade-off between the advantages and disadvantages of the first two approaches. When certain numbers of active paths are lost, flooding of *forward soldiers* is re-scheduled whenever information is sensed. Seeds making stochastic movement help in the advertisement of the existence of a node in the network. These seeds are needed so that forward soldier flooding in the network is reduced.

4.5 Performance Evaluation

This section evaluates the performance of the proposed routing algorithm (Termite-hill) implemented in Routing Modeling Application Simulation Environment (RMASE) (PARC, 2006; Zhang et al., 2006; Zhang, 2005) which is a framework implemented as an application in the Probabilistic

Wireless Network Simulator (PROWLER) (Sztipanovits, 2004). The simulator is written and runs under Matlab, thus providing a fast and easy way to prototype applications and having nice visualization capabilities for the experimental and comparison purpose.

Prowler is an event-driven simulator that can be set to operate in either deterministic or probabilistic mode. Prowler consists of radio model as well as a MAC-layer model. The MAC layer simulates the Berkeley motes' CSMA protocol, including the random waiting and back-offs. The Radio propagation model determines the strength of a transmitted signal at a particular point of the space for all transmitters in the system. Based on this information, the signal reception conditions for the receivers can be evaluated and collisions can be detected. The signal strength from the transmitter to a receiver is determined by a deterministic propagation function, and by random disturbances. The transmission model is given by:

$$P_{rec,ideal}(d) = P_{transmit}\frac{1}{1+d^{\gamma}} \tag{4.8}$$

$$P_{rec}(i,j) = P_{rec,ideal}(d_{i,j}).(1+\alpha(i,j)).(1+\beta(t)) \tag{4.9}$$

Where $P_{rec,ideal}$ is the ideal reception signal strength, $P_{transmit}$, the transmission signal power, d, the distance between the transmitter and the receiver, γ, a decay parameter with typical values of 2≤ γ ≤ 4, α and β, random variables with normal distributions $N(0,\sigma_{\alpha})$ and $N(0,\sigma_{\beta})$, respectively. In the simulation experiments, sensor nodes are placed in a 200 m x 200 m monitored rectangular area randomly and uniformly. The simulation parameters are as shown in Table 4.1.

Table 4.1: Simulation parameters for SC, FF, AODV and Termite-hill

Parameters	Values
Routing Protocols	SC, FF, AODV, Termite-hill
Number of Nodes, Source type, center type, radius, rate, Random rate	9, Static, random, 1, 4, 0
Destination type, center type, radius, rate, random rate	Static & Mobile, random, 1, 0.5, 0
Data Traffic, Data rate	CBR, 250Kbps
Simulation Time, Nodes Energy	360 Secs., 30J
Ant Ratio, AntStart, Window size, C1, Z, Reward scale, Data gain	2, 240000, 10, 0.7, 1, 0.3, 1.2
Rrep_delay, Rreq_Timeout, Rreq_Delay, Aodv_Rtable_size	40000, 40000, 4000, 10
Aodv_RQCache_size, Rrep_Retries	10, 3
BSoldier_Delay, FSoldier_Timeout, FSoldier_Delay	20000, 28000, 4000,
Termite_FSCache, Termite_RTable_Size, Termite_Ftable_Size,	10, 10, 10
Seed_Table_Size, BSoldier_Retries	5, 3

From several results obtained from our simulation work, we observed the following metrics to evaluate the performance of Termite-hill routing algorithm in WSN.

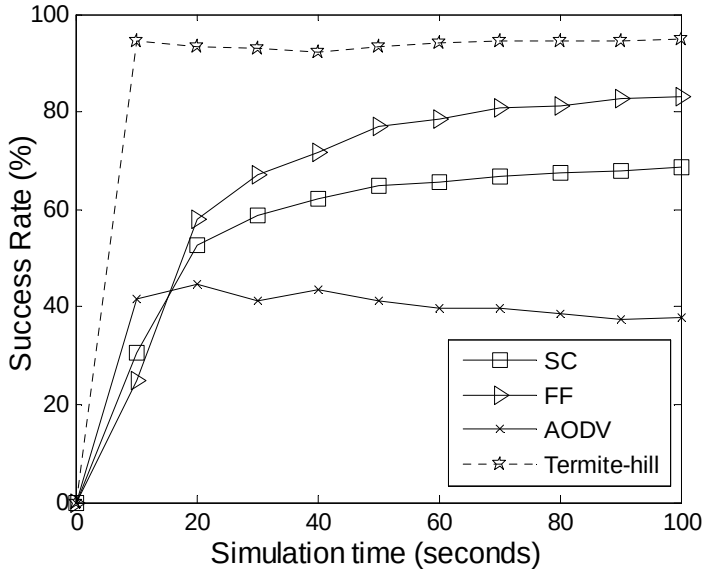
1. ***Throughput:*** it is the average rate of successful packets delivered over the network. It is measured in data packets per second.
2. ***Energy consumption*:** it is the total energy consumed by the nodes in the network during the period of the experiment (Joules).
3. ***Standard Deviation***: this is the average variation between energy levels of all nodes in the network (Joules).
4. **Success Rate:** it is a ratio of total number of events received at the destination to the total number of events generated by the nodes in the sensor network (%).

5. ***The Average Energy:*** it represents the average of energy of all nodes at the end of the experiment. We reported it in percentage (%).
6. ***Energy Utilization Efficiency*****:** it is a measure of the ratio of total packet delivered at the sink node to the total energy consumed by the network's sensor nodes (Kbits/Joules).
7. ***Network Lifetime:*** it is the difference of total energy of the network and the summation of average used energy of nodes and their energy deviation i.e. $Lifetime = \left(total\ network\ energy - (\frac{used\ energy}{total\ nodes} + energy\ deviation)\right)$.

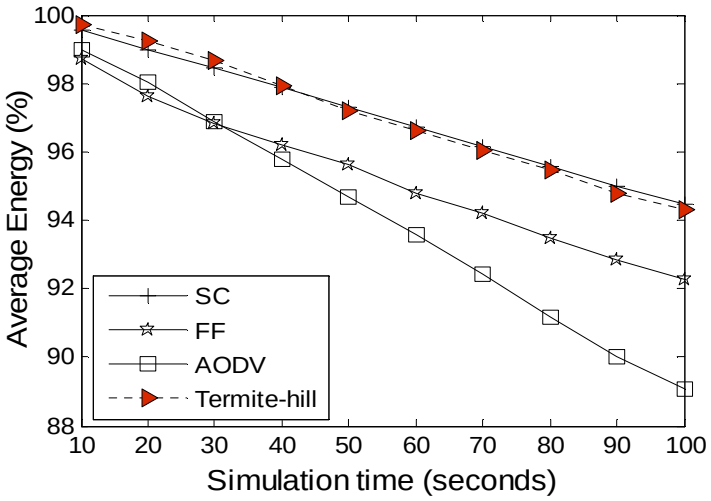
4.5.1 Performance of Termite-hill with Static Sink

First, we evaluate the performance of Termite-hill and compare with other routing protocols with static sink. In this scenario, we assumed that the sink node is fixed at a particular destination. In our simulation, results as reported in Fig. 4.7 show the performance in term of the success rate of events generated by the network, average residual energy, energy utilization efficiency, standard deviation of nodes in the network for the respective algorithms with different simulation time, and other performance with respect to variation in network density. In terms of successful packet delivered at the sink node, it was observed that Termite-hill has its maximum success rate when the simulation time was still at the initial level (at t =10s) corresponding to the value of 94.4%. It maintains almost that value till the end of the simulation period. But AODV performance in that case keeps decreasing as the simulation time. The poor performance was due to flooding of route discovery packets each time of the routing process, as most of its data packets do not actually get to sink even when generated by the source nodes. It was also observed that Termite-hill performance was higher as compared to the entire algorithm under investigation. Even with the high reliability (high success rate), its performance in term of energy consumption was better than other

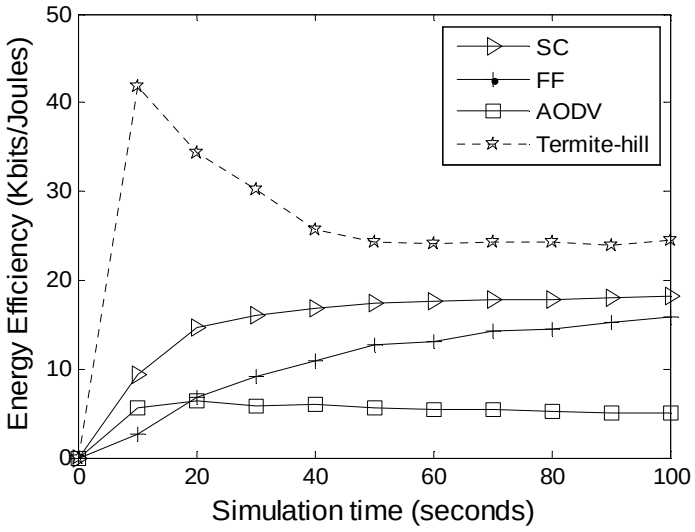
algorithms which in turn, makes it the most energy efficient. Our proposed algorithm achieves both high packet successful delivery, and energy utilization efficiency as compared to SC, FF, and AODV due to some of its important features as, first, the launch of its soldier carrying the first generated event in which most cases it is able to find routes to the destination in the first attempt; second, it makes use of restrictive flooding which results in quick convergence of the algorithm; third, it maintains a small event cache to queue events while route discovery is in progress; fourth, it utilize a simple packet switching model in which intermediate nodes do not perform complex routing table lookup as in others, rather packets are switched using a simple forwarding table at a faster rate; and lastly, the updating rule takes into consideration the paths energy, hence the probability of route selection is also a function of paths remaining energy. Though, generally, for variation in network density, energy consumption increase with increases in network nodes, while the success rate decreases slightly with increases in the network nodes which in turn affect the energy utilization efficiency. Though, Termite-hill still has the best performance with slightly higher delay.



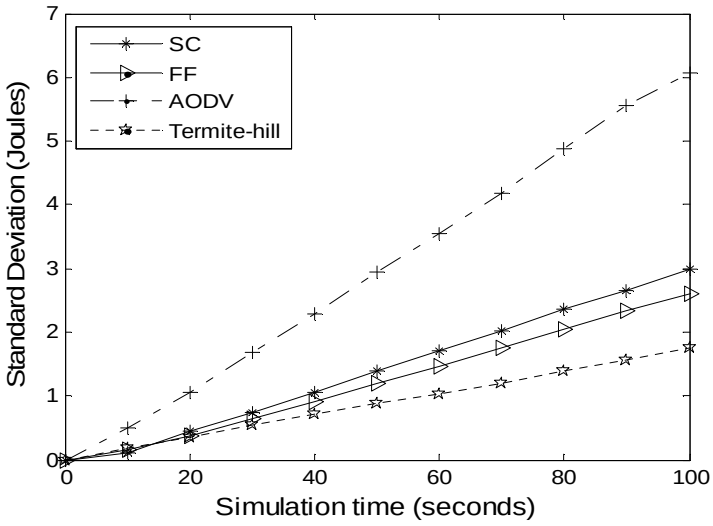
(a)


Average Energy (%)
Simulation time (seconds)
SC
FF
AODV
Termite-hill

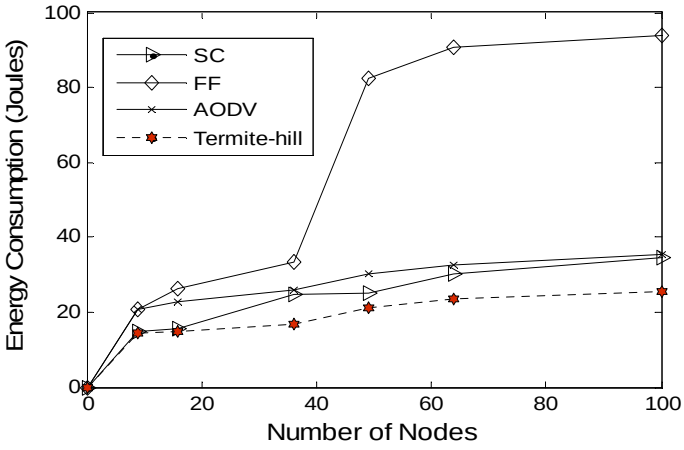
(b)


Energy Efficiency (Kbits/Joules)
Simulation time (seconds)
SC
FF
AODV
Termite-hill

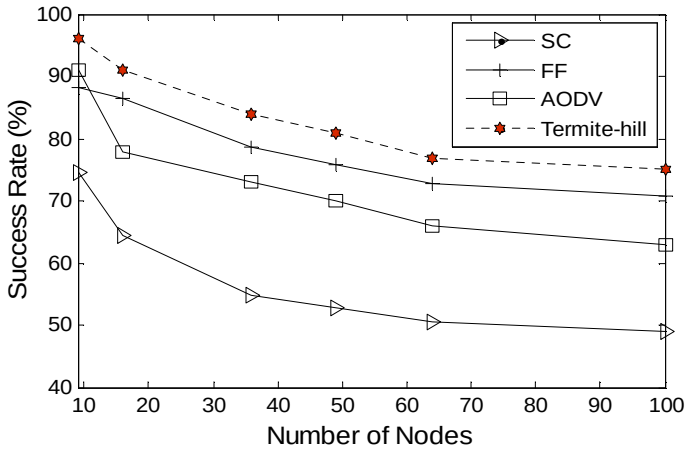
(c)


Standard Deviation (Joules)
Simulation time (seconds)
SC
FF
AODV
Termite-hill

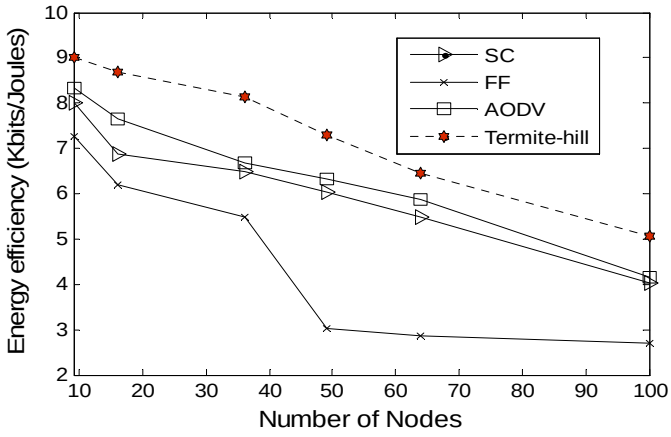
(d)



(e)



(f)



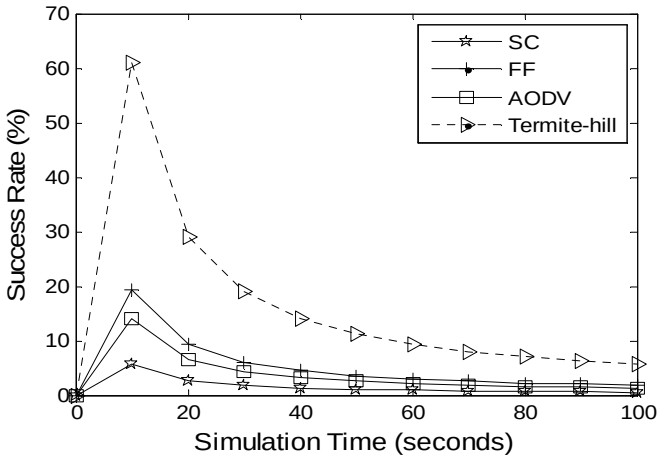
(g)

Fig. 4.7: Performance evaluation in static scenario: (a) Success Rate (b) Average Energy (c) Energy Efficiency (d) Standard Deviation (e) Energy consumption for different network densities (f) Success rate for different network densities (g) Energy efficiency for different network densities.

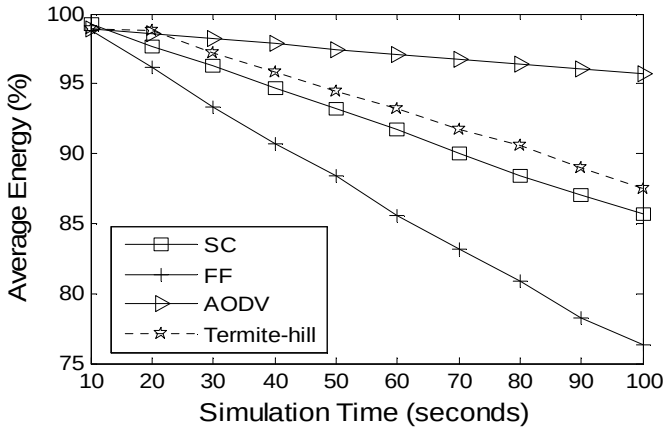
4.5.2 Performance of Termite-hill using Dynamic Sink (Target Tracking)

In this section, we study and evaluate the performance of Termite-hill with other routing algorithms in dynamic networks. In this scenario, we also assumed that the sink can change its location at any given time. The change is not along a path, but in any direction making it different from the mobility scenario. This is basically target tracking scenario. The target in the region of interest has to be monitored, but sometimes, it gets out of transmission range of almost all the nodes, hence the use of dynamic sink became very important as also, sensor nodes need fewer hops to get to the sink so as to limit energy consumption. In the first part, we simulate the entire algorithms over long duration of time with a fixed speed of sink as shown in Fig. 4.8. In that scenario, Termite-hill performance in terms of successful packet delivery was still higher than the other algorithms. Though, it consumes slighter higher energy as compared to AODV due to its higher packet delivery rate in fewer nodes. But with its high packet delivery rate, it has the highest energy utilization efficiency as compared to all the algorithms. But when the network grows higher with many nodes, the performance of AODV degrades in terms of energy consumption, which in turn affects its energy utilization efficiency. To further test its performance, we adapt all the routing algorithms in the dynamic scenario with varying speed of sink as shown in Fig. 4.9. In that case, Termite-hill performance in terms of successful packet delivery rate and energy utilization efficiency is higher, with less standard deviation. It will also be observed that though the success rate of each of the routing protocols tends to increase with variable speed of the sink, the energy consumption of all the algorithms also increases as more packets are delivered at the sink node since the average remaining energy keeps on dropping. The poor performance of FF in terms of high energy consumption is due to its pure

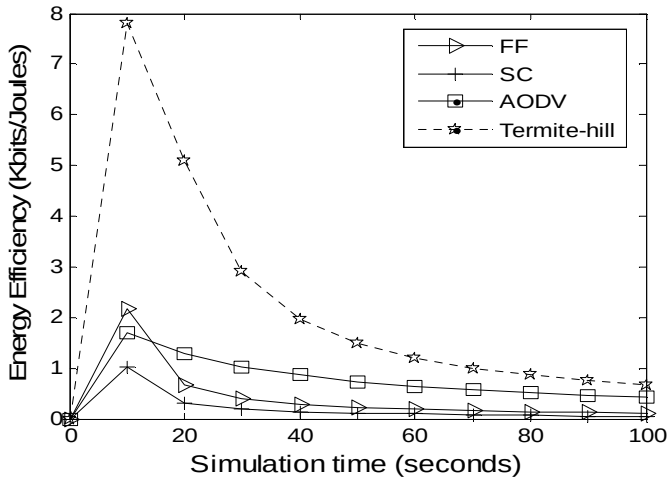
flooding of RREQ packets (ants), which make it to have unnecessary overhead in the network. Towards the end of this section, we compare the reliability of individual protocol with itself for the different scenarios.



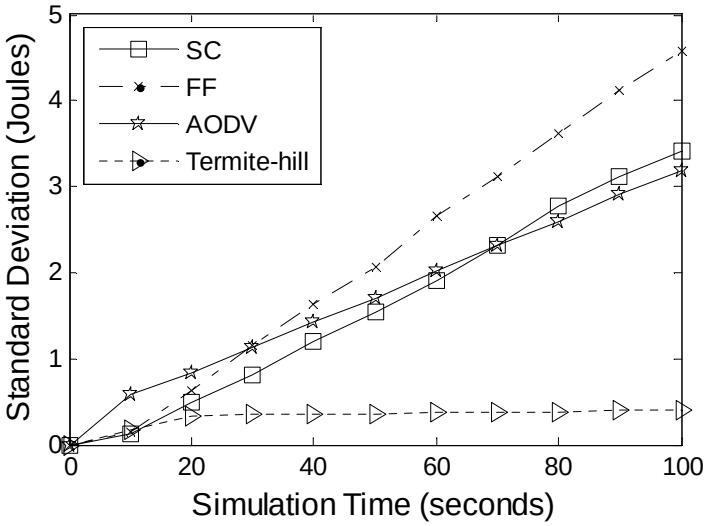
(a)



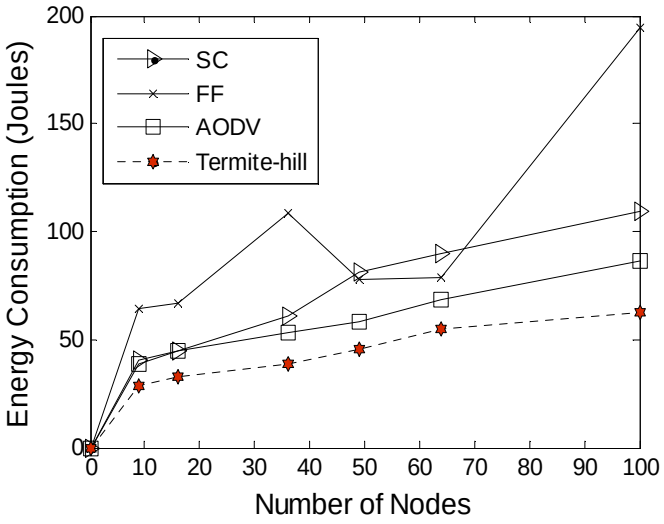
(b)



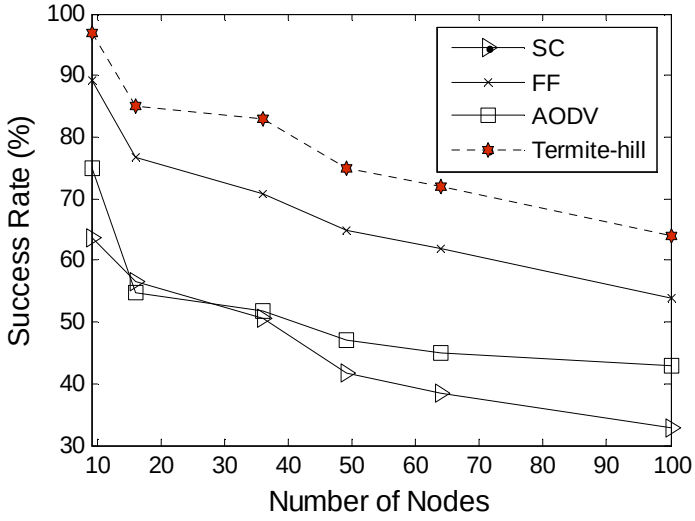
(c)


SC
FF
AODV
Termite-hill
Standard Deviation (Joules)
Simulation Time (seconds)
0
1
2
3
4
5
0
20
40
60
80
100

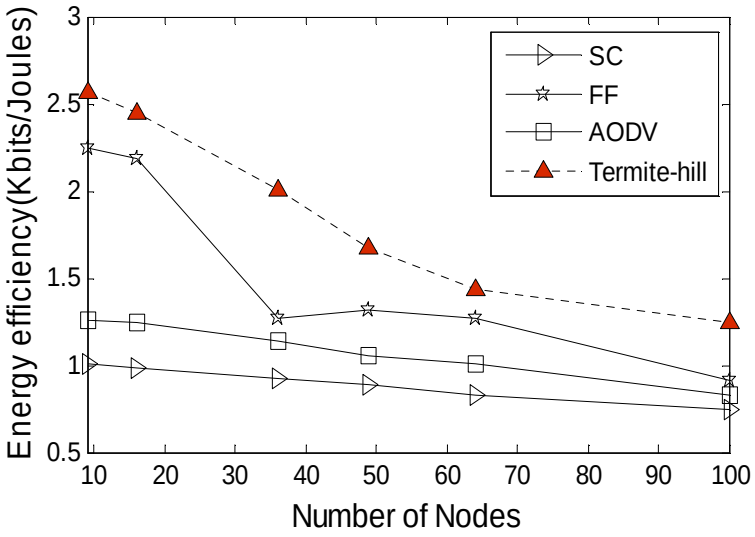
(d)


SC
FF
AODV
Termite-hill
Energy Consumption (Joules)
Number of Nodes
0
50
100
150
200
0
10
20
30
40
50
60
70
80
90
100

(e)

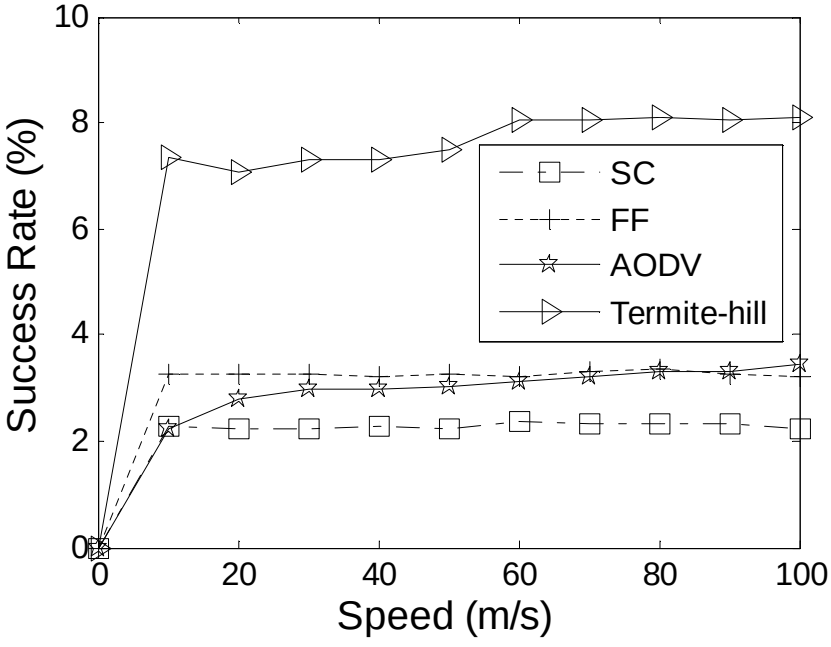

SC
FF
AODV
Termite-hill
Success Rate (%)
Number of Nodes
30
40
50
60
70
80
90
100
10
20
30
40
50
60
70
80
90
100

(f)

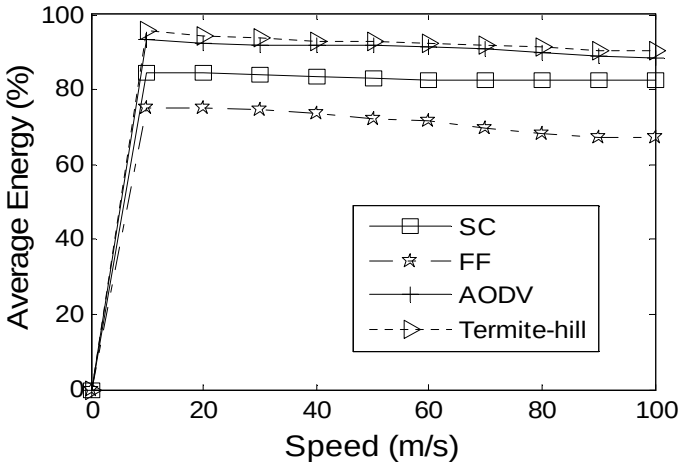


(g)

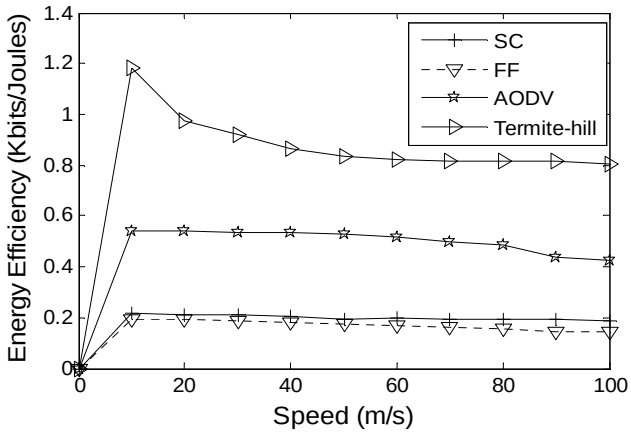
Fig. 4.8: Performance evaluation in dynamic scenario among routing protocols: (a) Success rate (b) Average energy (c) Energy efficiency (d) Standard deviation (e) Energy consumption for different network densities (f) Success rate for different network densities (g) Energy efficiency for different network densities.



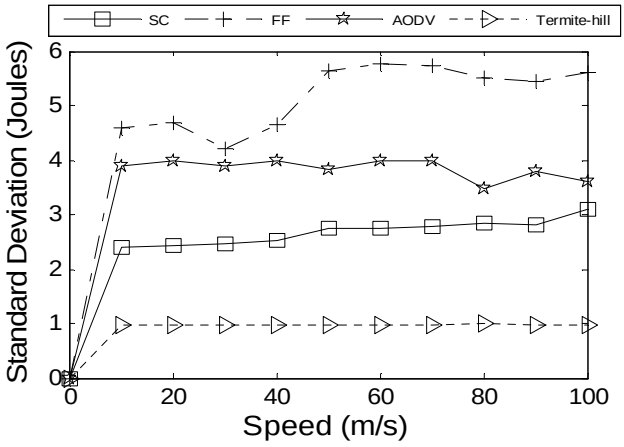
(a)



(b)



(c)



(d)

Fig. 4.9: Performance evaluation in dynamic scenario with varying speed among routing protocols: (a) Success Rate (b) Average Energy (c) Energy Efficiency (d) Standard Deviation.

4.5.3 Performance of Termite-hill with a Mobile Sink

In this section, we compare the performance of the proposed routing algorithm with other state-of-the-art routing algorithms in term of sink mobility. In this scenario, we assumed that only the sink is mobile, the network has only one data collection center (sink), and the sink moved along the border of the monitored area as described in Chapter 3, and shown in Fig. 3.1(b) and (c). In this scenario as in the static scenario, we also show the performance of Termite-hill in relation to the other state-of-the-art routing algorithms in term of network reliability (throughput), energy consumption, and energy utilization efficiency of the respective algorithms with mobile sink and a variation of sink speed. In terms of successful packet delivery per unit time, it should be observed in Fig. 4.10 (a) that Termite-hill did not compromise it as against network lifetime improvement. As shown in Table 4.2, is also the impact of sink mobility on network lifetime among different routing protocols. This table highlights the elongation of network lifetime as the sink becomes mobile to monitor the area of interest. In Fig. 4.10 (b), it should be observed that even with the increase in speed of the sink there is less effect on the energy consumption unlike in the static scenario whereby energy consumption increases with simulation time. The low energy consumption of almost all the algorithms was due to the fact that the sink is mobile, and also moved along the monitored area thereby help in shorting the number of hops to get to it, which in turns reduces the energy consumption of the participating nodes during the routing and the path discovering process. Though, all the algorithms did quite well in the mobility scenario, but our proposed routing algorithm did very well as compared to all in both the metrics used for the evaluation process. It will also be observed that unlike in the static scenario whereby energy utilization efficiency decreases with simulation time, in mobility, energy utilization of especially our proposed algorithm

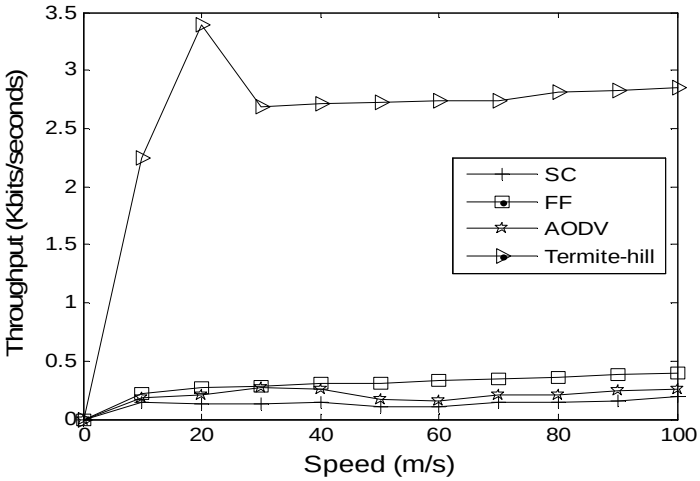
rises rapidly with increase in speed of the sink. This is due to less number of route discovery packets and limitation of the maximum number of retransmission of packets to three (3) as described in Chapter 5 of energy consumption of Termite-hill.

Table 4.2: Impact of sink mobility on network lifetime among different routing protocols

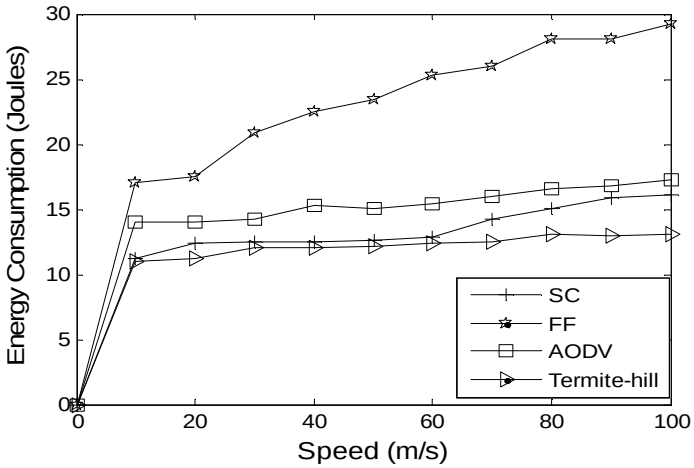
PROTOCOLS	Static Lifetime (%)	Mobile Lifetime (%)	Difference (%)
Termite-hill	98.4648	98.8747	0.4099
AODV	96.5408	98.1824	1.6416
SC	98.2818	98.8741	0.5923
FF	98.1738	98.2185	0.0447

Table 4.3: Comparison of the Routing Protocols based on different Metrics in Dynamic Scenario

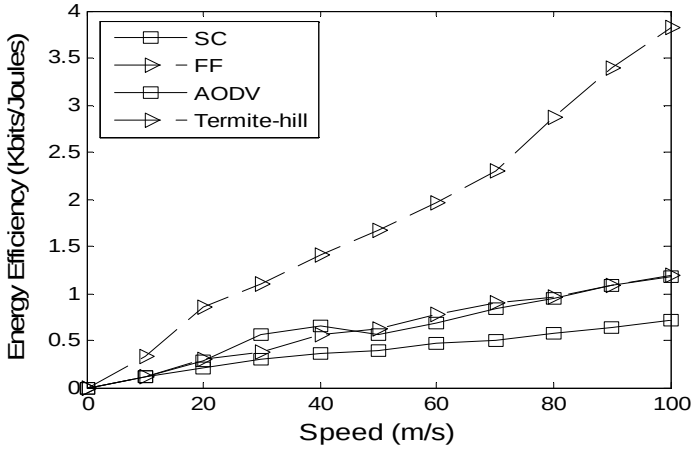
Routing Protocols	Latency (s)	Success Rate (%)	Energy Consumption (J)	Energy Efficiency (Kbits/J)	Energy Deviation (J)	Lifetime (days)
FF	0.0860	3.2152	88.9360	0.1432	5.5980	254.52
SC	0.1082	2.2101	47.5456	0.1841	3.1139	261.60
AODV	0.0306	3.4260	31.4720	0.4267	3.6085	262.89
Termite-hill	0.1687	8.0778	39.6736	0.8063	0.9758	264.62



(a)



(b)

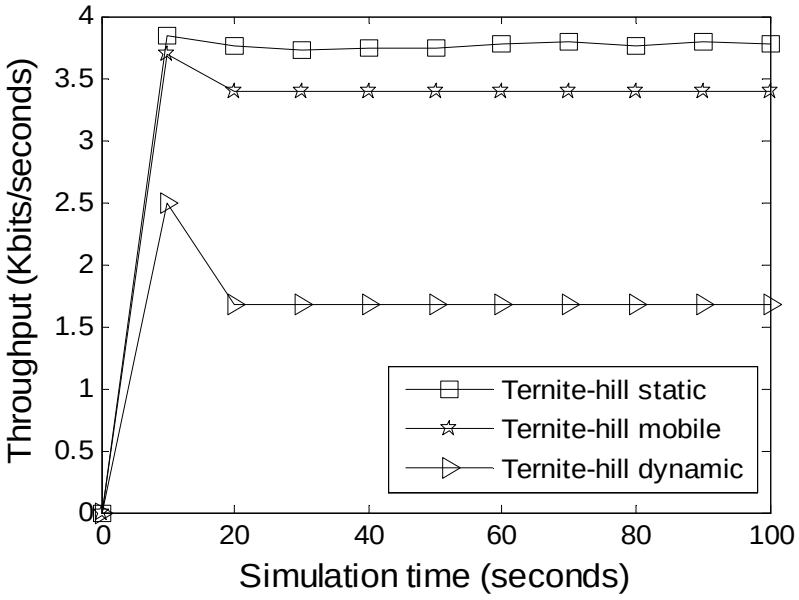


(c)

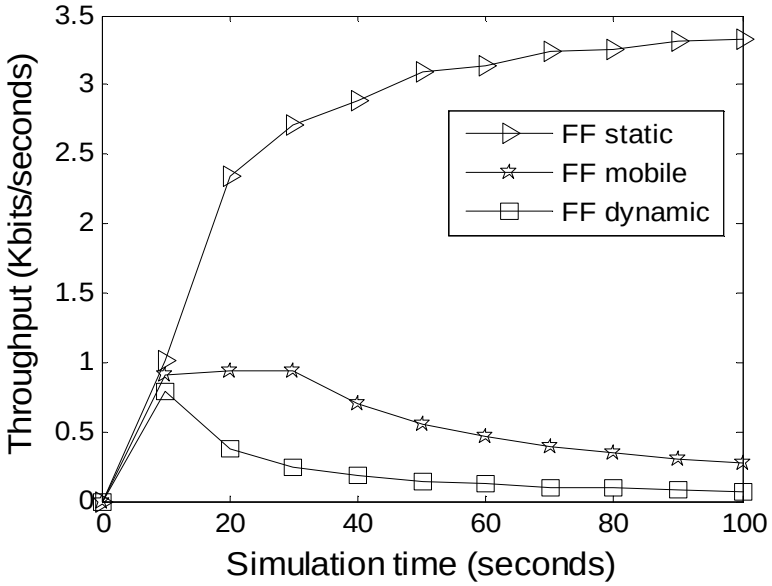
Fig. 4.10: Performance evaluation in mobile sink scenario with varying speed among routing protocols: (a) Throughput (b) Energy consumption (c) Energy Efficiency.

4.5.4 Performance of Termite-hill in terms of Network Reliability

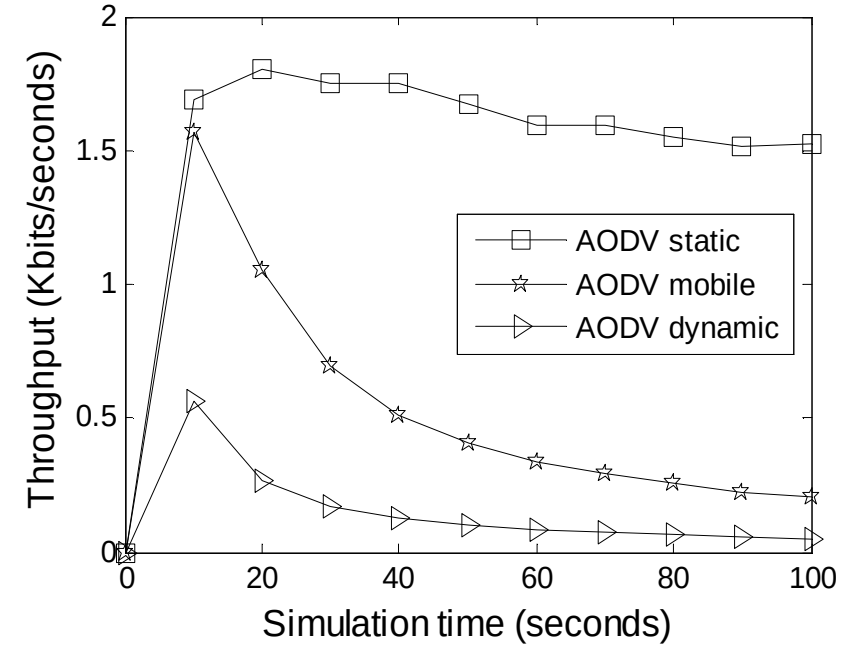
In this section, we compare the reliability of the individual routing algorithm in the different scenario tested. It should be clearly seen that, the proposed algorithm termed Termite-hill was able to balance the network reliability even as it improved on the network lifetime in the mobility scenario as observed in Table 4.2 and Fig. 4.11(a).



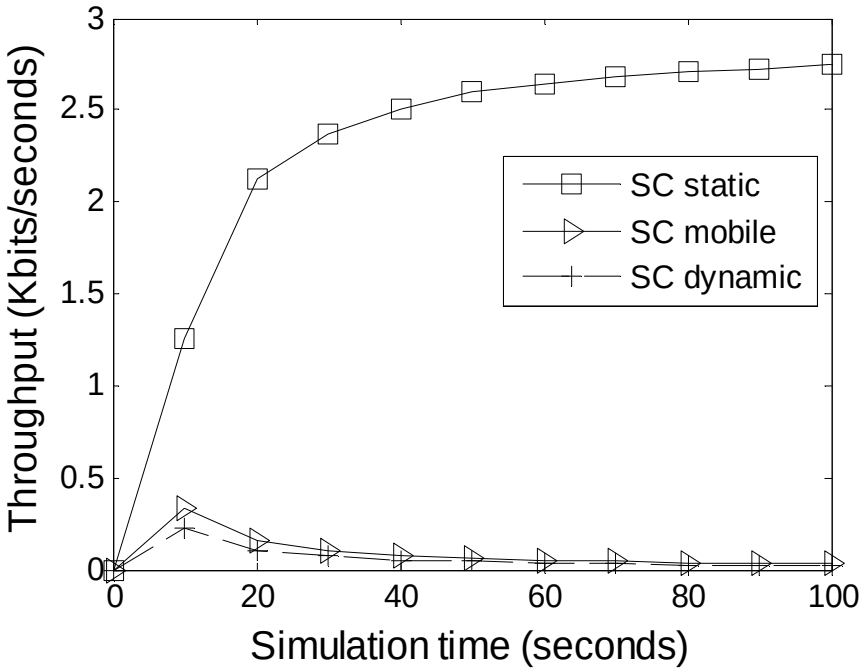
(a)



(b)



(c)



(d)

Fig. 4.11: Evaluation of impact of sink mobility on network reliability for different routing protocols: (a) Termite-hill (b) FF (c) AODV (d) SC.

4.6 Conclusions

In this chapter, we studied the natural behaviors of termites as regards the way by which they build their hills. The study inspires us to convert the real termite to artificial termite which helps us see how we could use them to solve the routing problem in WSN. Afterward, Termite inspired routing algorithm emerged as the result of the study, of which many factors and improvements inspired by the features of WSNs which includes limited available energy, constraint in terms of memory capacity, and its ability to process information were considered and implemented. The resulting Termite-hill algorithm was designed to function in three modules; route discovery, route maintenance (seed), and data packet module. The algorithm uses backward and forward soldiers for route discovery and updating between source nodes and the destination node, and the route is optimized in terms of distance, memory, and energy level of each path. The algorithm minimizes network overhead by on-demand routing, and maximizes network reliability and energy savings, which contribute to improving the sensor network lifetime. From the simulation results gathered, Termite-hill had a good performance in terms of energy utilization efficiency and reliability. Towards the end, we also investigated the impact of static sink, sink mobility and dynamic sink on network performance in a WSN using static, dynamic and mobility scenarios of the network. We analyzed the packet arrival rate, energy consumption, energy utilization efficiency and network lifetime of AODV, FF, SC, and our proposed routing algorithm termed Termite-hill. From the results gathered, it was seen that the performance of the algorithms as regards to energy utilization efficiency, network reliability, and network lifetime had a mutual relationship between sink mobility, dynamic sink and simulation time. Our proposed routing protocol Termite-hill has shown and demonstrated good performance in terms of energy utilization efficiency, throughput, and

network lifetime. With the introduction of mobile sink, the nodes around the sink always changes, thus balancing the energy consumption in the network and improving the network lifetime. From the results, the network lifetime was improved in all the algorithms, but many tend to sacrifice network reliability with network lifetime. Our proposed algorithm was able to balance the improvement of the network lifetime with the reliability of the network. SC and FF performances were below average due to their flooding nature of control packets which is the cause of the low packet delivery and high energy consumption leading to low energy utilization efficiency. The high performance in terms of low energy consumption of Termite-hill and AODV was due to the fact that they both try to increase their energy savings since nodes can only participate in the routing process only when there is query from the sink node or when they have sensed an event and they wanted to get it across to the destination node. Though, a little delay in event transmission is incurred, but it is neglected as compared to their efficiency in other metrics as the delay is also comparable with other protocols.

Chapter 5

A Formal Mathematical Framework for Modeling and Simulation of Wireless Sensor Network Environments

5.1 Introduction

The current practice in modeling and simulation of wireless sensor network (WSN) environments is to develop functional WSN systems for event gathering, and optimize the necessary performance metrics using heuristics and intuition. WSN typically consists of sensor nodes with sensing, computing, and communication devices. The main goal of the WSN is to gather event from the environment and transmit the event to a sink node. WSNs are usually self-configured ad-hoc networks with static source nodes. WSN technology achieved high application demands due to its versatility in areas of applications in addition to its wireless mode of communication. As we move ahead to design and implement such complex communication networks and systems, however, an important issue comes up. In the area of wireless sensor network, formal understanding is lagging behind in comparison with practice due to the fact that much attention is given to system designs and protocol development while leaving behind their analysis. This network system analysis is traditionally based on simulation tools. Also, due to vast applications of WSN, the creation of the network to perform each of the functionality may require a significant financial investment. For this reason, a priori knowledge of setting up a real system

is required. It is desirable to check the system behavior by means of mathematical analysis and simulation. In this study, we use formal techniques to analyze sensor network systems. Formal techniques not only provide a more abstract model, but make it possible to prove system and protocol correctness in addition to simulating its behavior.

The evaluation and validation of the WSN system are mostly done using simulation approaches and practical implementations. Simulation studies despite their wide use and merits of network systems and algorithm validation, has some drawbacks like long simulation times, and some results reported by several prominent protocols cannot be repeated due to lack of mathematical backings. We therefore, argue that simulation based validation of WSN systems and environments should be further strengthened through mathematical analysis. Simulation-based tools can provide useful statistical results from the system and protocol behavior, but since it is practically impossible to test exhaustively all the behaviors of the networks, these tools cannot prove the correctness of a protocol. Besides, simulation tools have limited scalability, and the results are application specific and require long simulation times which slow down the system engineering cycle. By using formal techniques, we can inspect all reachable states of a system and prove its correctness. Much of the current literature in WSNs deals with practical implementation of the recent algorithms for event gathering. Though, simulation experiments are conducted to verify their performance using some well-known metrics or metrics that favors the new algorithm. Even when compared with other algorithms, it is usually a selection approach of some metrics. However, if a formal understanding of the new algorithm is known, there will be more factors to be used in comparing with other protocols and as well as some areas that needs improvement after the analysis. In other words, simulation studies should be complemented by mathematical analysis so as to overcome some of

their shortcomings. In accordance with (Moreno et al., 2012; Saleem et al., 2008; Torres, et al., 2010; Kazemeyni et al., 2012), which argues for more clarification and repeatability of proposals, mathematical analysis is needed to widen the scope and throw more light on the achievable performance.

Little attention has been paid in the area of formal mathematical analysis for WSN topologies and event gathering. Such analysis is needed so as to throw more light on the theory behind their performance and in turn to know how variation in topology, network density, available energy, mobility, traffic load, and reliability affect the system design. To this end, we developed our modeling framework for WSN topology and information extraction in a grid based and line based randomly distributed sensor network. We strengthen the work with a model of the effect of node mobility on energy consumption of Termite-hill routing algorithm as a function of event success rate and occasional change in topology. The results of our mathematical analysis were also compared with the simulation results.

5.1.1 Organization of the chapter

Other sections contained in this chapter are organized as follows: Section 5.2 discusses some related work that deals with modeling of WSN systems and routing protocols. Section 5.3 describes the network system, gives some basic definition of some terms and model the system topology and energy consumption. We model the system in terms of information extraction with the limited available energy of sensor nodes in Section 5.4. In Section 5.5, we present the analysis and simulation results. We give concluding remarks and discussions in Section 5.6.

5.2 Related Work

The formal modeling of wireless sensor network (WSN) systems and environments has received little attention in the literature. In (Moreno et

al., 2012), a formal model of wireless sensor grids is presented. The authors model and design a simulator of systems that integrates sensors in a grid, which is an extension of GridSim. In a related work, Roth (2007) study was the design of a mathematical framework for the analysis of routing algorithms which are based on probabilities. In (Hamida and Chelius, 2007), the authors present a line based data dissemination algorithm and compare it with other dissemination architectures using an analytical analysis based on average distances between source and sink nodes. In a similar work (Shin et al., 2005; Khan et al., 2009), the authors model the energy consumption of a typical grid based sensor network. This work is similar to our proposed work, but the authors concentrated on the energy consumption of a grid based sensor network without considering information extraction on the network with the limited available energy of sensor nodes.

However, there are other studies that are similar to our proposed work in terms of event extraction modeling such as the work of Toumpis and Goldsmith (2003), and Johansson et al. (2003). WSN is energy intensive and much information is expected to be delivered to the end device (sink) at a high success rate with limited energy resource sensor nodes. Unlike the above literatures that concentrated on topology modeling or event extraction modeling, we model the network system in terms of information extraction with limited energy resources, and also consider the minimum transmission cost of a typical grid based sensor network. Towards the end of the work, we described an agent based reactive routing protocol (Termite-hill), and analyze the expected energy consumption of the protocol. We further strengthen our work with a comparative study of the analytical results and simulation results.

5.3 Network System Description and Definitions

5.3.1 Network description

Considering a network of *N* sensor nodes ranging from 1, 2, 3 . . . N, and a sink node (base station) *D*, distributed over a region, the location of the sensor nodes is fixed, of which is the case in most of the application of wireless sensor networks (WSNs). Each sensor is allowed to monitor the vicinity or area of interest of the sink. We assume that each sensor generates a data packet per time with size *n*-bits, and each of the sensor nodes has the ability to transmit its sensed event either directly or using other sensor nodes as routers to the sink node. For typical WSNs, each sensor node has a battery with finite and non-rechargeable energy. When an event is received or transmitted by a sensor node, part of its energy is utilized in the process. In this model, we assumed that the sink node has an unlimited amount of energy available to it. We shall also consider a grid based network and a randomly distributed network in a ring. To represent and describe the bandwidth and limited energy constraints in WSNs, some basic fundamental models are needed to actually describe the system. 1. Mathematical analysis (model) for node distribution and connectivity. 2. A model for network lifetime using the energy consumption of the network nodes. 3. A model for reliability of the system.

With this network of *N* sensor nodes independently and randomly distributed on a two dimensional simulation area A, if we use a uniform random distribution such that for a large number of nodes *N* and a wide space area *A*, the node density in the network can be given as $\rho = \frac{N}{A}$. This represents the expected number of nodes in a given area.

To analyze the connectivity between nodes in the network, a simple radio link model is assumed such that each node in the network has a certain transmission range d_{tx} and uses omnidirectional antennas. This represents

the bandwidth constraints of the sensor networks since the transmission range is limited. As shown in Fig. 5.1, two nodes can communicate with each other via a wireless communication link, only if they are within the communication range of each other, else, they cannot communicate. The communication is vice-versa, i.e. bi-directional. If we consider a propagation model with a certain signal attenuation (path loss) (Zhang, 2005) with $P_{tx} = P\ (d = 0)$ as the transmitted signal power at the sending node, and $P_{rx}(d)$ the received power at the receiving node of distance *d* from the source (sender), the received power can be denoted as $P_{rx}(d) \propto d^{-\gamma} P_{tx}$ where γ is the path loss exponent, and typically depends on the environment $(2 \leq \gamma \leq 4)$ (Zhang, 2005). We can then map the wireless transmission range d_{tx} to the equivalent transmission power P_{tx} using a threshold for receiver sensitivity $P_{rx(s)}$. A node in the network can receive information from other nodes adequately if and only if $P\ (d = d_{tx}) \geq P_{rx(s)}$. In this system, all nodes are static in the network area, but the sink node may be allowed to move into the system area according to some application requirements. To handle such network types, it is important to investigate the fundamental properties and parameters that characterize the network topology. These are mostly the possible connectivity, reliability and the behavior of the underlying algorithms or protocols, and above all, the energy consumptions of the participating nodes.

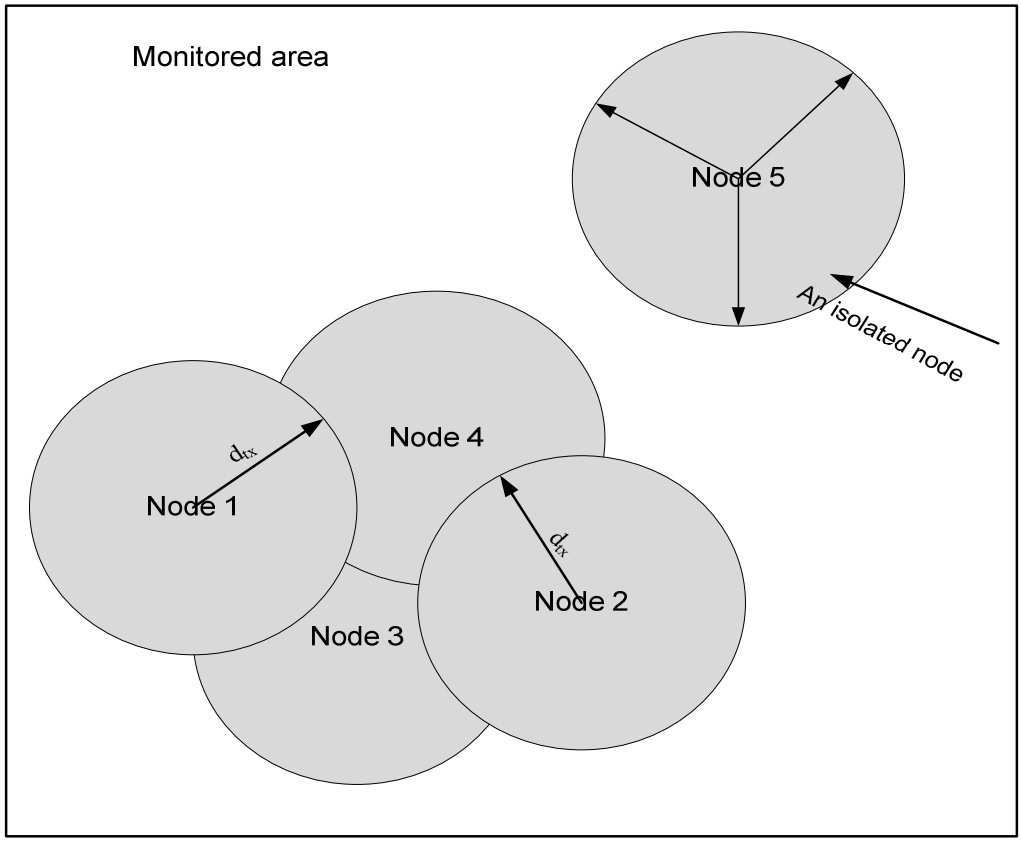


Fig. 5.1: Modeling the topology of a wireless sensor network.

5.3.2 Network topology

The topology model primarily specifies the topology of the network, i.e. the relative placement of nodes. In this model, we intend to analyze the basic topology of the Routing Modeling Application Simulation Environment (RMASE), since that is the environment in which Termite-hill routing algorithm was simulated for performance evaluation. The basic topology of RMASE (Zhang, 2005) is a rectangular x-y grid. The grid size, spacing, shift, density, and offset parameters in the simulation environment allows users to specify a variety of topologies including the triangular grids, long parallel lines, and random networks of various kinds. As pointed out above, we shall concentrate on the grid based network. We therefore define some parameters here for more understanding of the network topology.

Grid size (S_x, S_y): this is the number of grid points in *x* and *y* directions in the topology.

Grid distance (d_x, d_y): this represents the distance or spacing of the grid points in *x* and *y* directions of which the covered area is $S_x d_x$ by $S_y d_y$

Grid density (P_x, P_y): this specifies the number of nodes per side from grid point to grid point (extending out from the last grid point in each row and column). The total number of nodes is $S_x S_y * (P_x + P_y - 1)$.

Considering an ideal basic graph theory and interconnection networks (Hsu and Lin, 2009), we can model a WSN as an undirected graph $G = G(W, Z)$. This graph consists of *w* set of nodes (vertices) and *z* set of node pairs (wireless links, edges). The set of nodes denoted by $W(G) = \{w_1, w_2, \ldots\ldots, w_n\}$ represents the WSN enabled devices; and the set of node pairs denoted by $Z(G) = \{z_1, z_2, \ldots\ldots, z_m\}$, represent the wireless communication links. An element in $W(G)$ is called a node or vertex of *G*, and an element in $Z(G)$ is called a link or edge of *G*. A link between any pairs of nodes exists only if they lie within the transmission of each other, otherwise, no link exists. That is to say that, there is no connection between them. However, even with the existence of links between pairs of nodes, there is no guarantee of packet delivery between them due to channel errors and contention.

A graph can be considered as a connected graph, if for every pair of nodes or vertex, there is at least a single link between them, otherwise, it is a disconnected graph. A graph can also be regarded as a *n*-connected graph $(n = 1, 2, \ldots\ldots, n)$, if for every pair of vertex, there exist at least *n* mutually independent paths connecting them, and this is illustrated in Fig. 5.2. The number of connections of a node *a,* $d(a)$ is the number of nodes or node neighbors directly connected with it. A node is also isolated if it has a zero degree, i.e. $d(a) = 0$ (see Fig. 5.1). The minimum node degree of a graph $G(W, Z)$ is:

$$deg_{min}(G) = \min_{\forall a \epsilon G}\{\deg(a)\} \tag{5.1}$$

The average node degree of *G* is:

$$d_{avg}(G) = \frac{1}{n}\sum_{a=1}^{n} d(a) \quad (5.2)$$

In an undirected graph, $d_{avg}(G) = \frac{2m}{n}$ (Hsu and Lin, 2009)

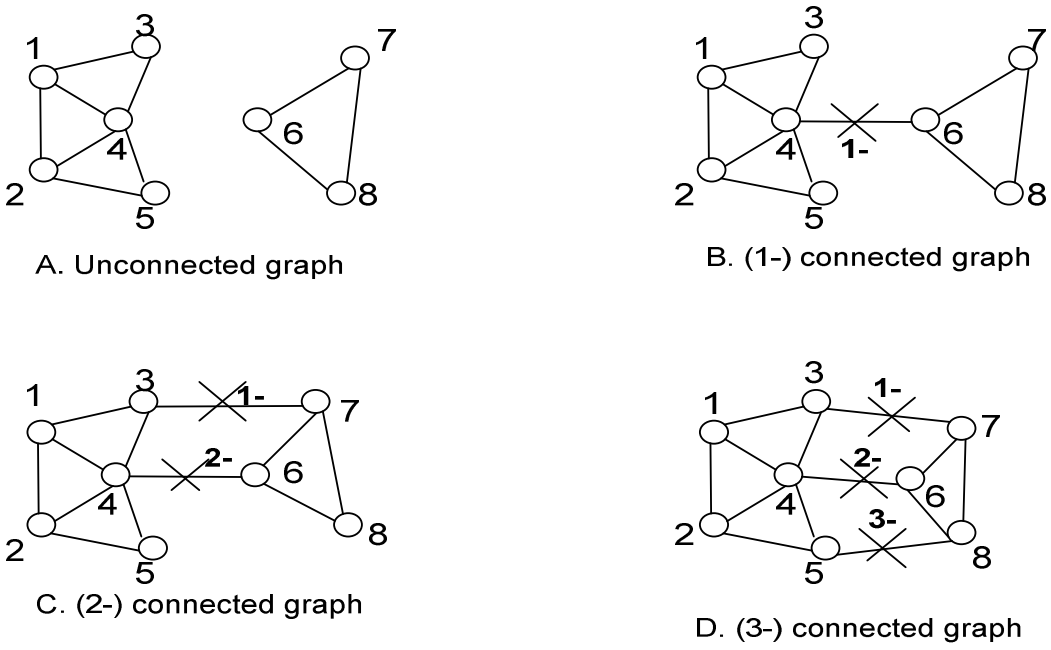


Fig. 5.2: Illustration of graph connectivity.

5.3.2.1 Grid distance

In the network of grid size S_x, S_y in the *x* and *y* directions, sensor nodes are relatively placed over the area or haphazardly scattered over the network within the grid area. Whenever there is a query broadcast from the sink node which is of interest to the sink to all the sensor nodes, the node with the desired information upon receiving the query, relay the information to the sink. The information relaying process can be directed to the sink node using just one hop or relayed using other intermediate nodes (using multiple hops). This node with the interested information by the sink node is called the source node. Or, in some cases, when nodes are placed within a monitored area of the grid, any of the sensor nodes that has information to send to the sink node at any point in time, is the source of the information, hence termed as the source node.

In a grid based approach, a grid topology is constructed with grid nodes from source nodes to sink node for event forwarding or relaying. A typical example of a grid topology is shown in Fig. 5.3.

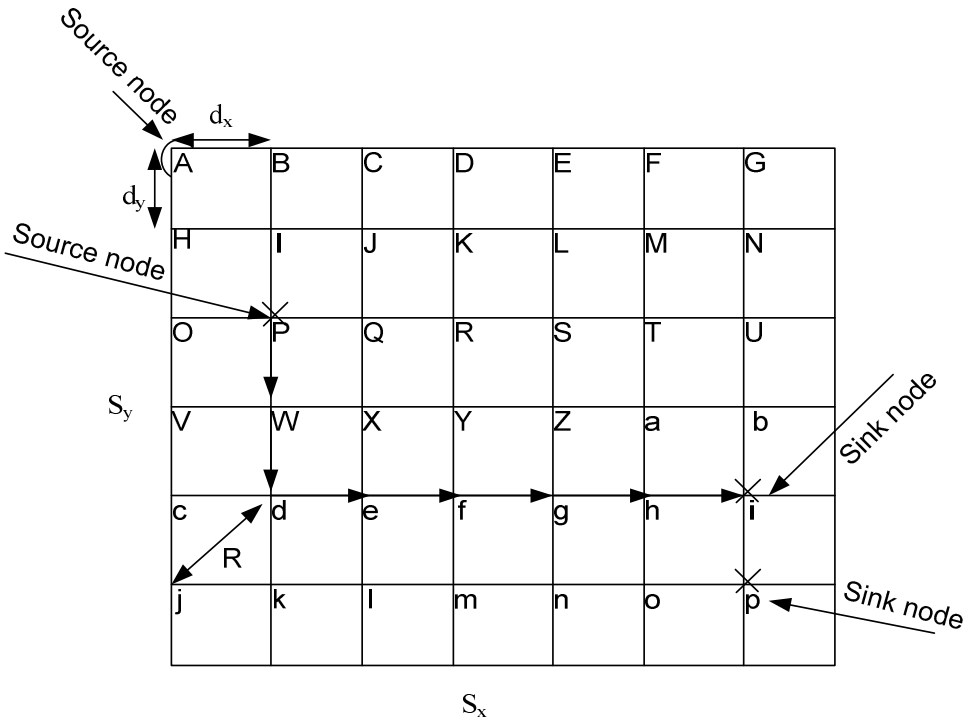


Fig. 5.3: A rectangular x-y grid topology in wireless sensor network.

If we construct a grid topology with sensor nodes placed randomly at each grid point as shown in Fig. 5.3 above, to determine the distance between two neighboring nodes for the construction of a grid topology, a trade-off between distance and energy consumption is required. If the distance selected between two consecutive neighboring nodes is too short, the number of forwarding hops to the sink node increases with a corresponding increase in energy consumption. On the other hand, if the distance selected is too high, more energy will be required for the node in the grid to forward the intended information to its neighbors. Also, the amount of energy needed transmission to forward a packet of information between nodes of the same communication range is proportional to the volume of information, measured in bits, and the square of the distance between

them. The energy required by a grid node for event forwarding using a single hop, involves the energy consumption by the sensor node for transmitting a packet of information E_{tx} and the energy consumption for receiving the packet by the receiver, E_{rx} (Meghji and Habibi, 2011). That is:

$$E_p = E_{tx} + E_{rx} \tag{5.3}$$

In a typical grid based topology where the source node is at an extreme end of the grid topology, and the sink node is also at the other end of the grid topology, the number of horizontal and vertical hops for any path from the source node to the sink node are the same. That is, for source node at point *A*, and sink node at point *p*, the path hops in any direction is 11. Now, assuming that the source node is at point *P*, and sink node at point *i*, taking the grid distance coordinate as d_x and d_y respectively for *x* and *y* direction. For two consecutive grid nodes, the distance between them *R* is:

$$R = ((d_{x1} - d_{x2})^2 + (d_{y1} - d_{y2})^2)^{\frac{1}{2}} \tag{5.4}$$

For a grid topology of width S_x and height S_y, the number of horizontal hops *X* and vertical hops *Y* from the source node to the sink node will be $\frac{S_x}{R}$ and $\frac{S_y}{R}$ respectively. For appropriate grid distance selection, the average transmission cost, T_{cav}, is defined in such a way as to indicate the average energy consumed per unit distance as:

$$T_{cav} = \frac{\sum_{(X+Y)hops}(E_{tx}+E_{rx})}{d} \tag{5.5}$$

and,

$$d = ((RX)^2 + (RY)^2)^{\frac{1}{2}} = R(X^2 + Y^2)^{\frac{1}{2}} \tag{5.6}$$

Where *d* is the distance from the source node to the sink node (direct distance to the sink). In accordance with (Heinzelman et al., 2000; Panda and Patra, 2010; Moreno et al., 2012), we assume a simple radio model as shown in Fig. 5.4, where the radio dissipation, E_{elec} *(J/bit)* to run the transmitter or receiver circuitry and the transmit amplifier energy, E_{amp}

*(J/bit*m²)* are required for sending information. Though, only the radio energy, E_{elec} is required for the receiver.

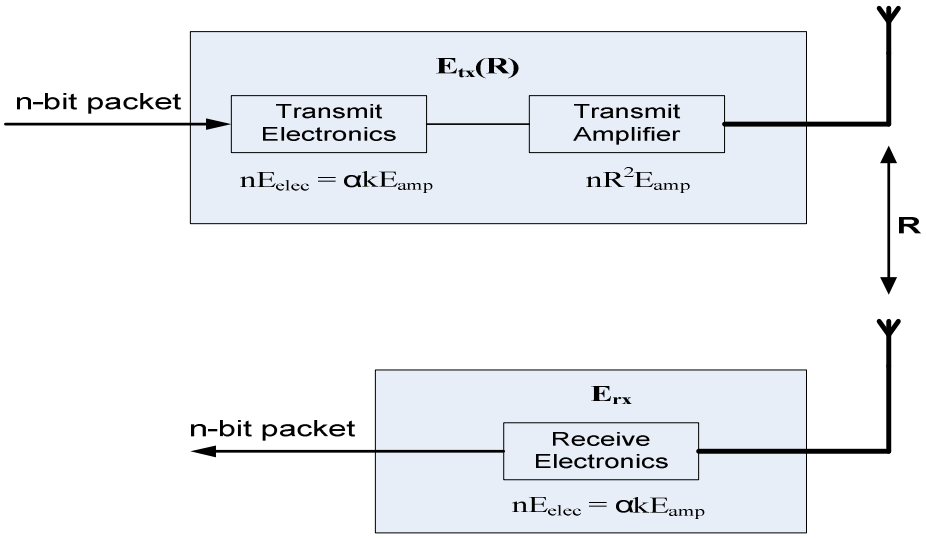


Fig. 5.4: First order radio model (single hop transmission).

Also, an R^2 energy loss due to channel transmission is assumed. Hence, to transmit an *n*-bit of information over a distance *R* using the simple radio model (Heinzelman et al., 2000; Panda and Patra, 2010), the energy expended in transmitting an information of *n*-bits of data in a single hop is$(E_{tx} + E_{rx})$, and

$$E_{tx}(n,R) = E_{tx-elec}(n) + E_{tx-amp}(n,R) \tag{5.7}$$

Or $$E_{tx}(n,R) = nE_{elec} + nR^{\lambda}E_{amp} = n\alpha E_{amp} + nR^{\lambda}E_{amp} \tag{5.8}$$

Where λ is the path-loss exponent such that $(2 \leq \lambda \leq 4)$. If we assumed $\lambda = 2$ for a single hop transmission (minimum loss) or minimum transmission cost,

$$E_{tx}(n,R) = nE_{amp}(\alpha + R^2) \tag{5.9}$$

Where α is an amplification factor such that $\alpha = \frac{E_{elec}}{E_{amp}}$. And to receive the *n*-bit of a message, the radio energy expended is:

$$E_{rx}(n) = E_{rx-elec}(n) = nE_{elec} = \alpha nE_{amp} \tag{5.10}$$

But the average energy consumed per unit distance in (5), can then be re-arranged as:

$$T_{cav} = \frac{\sum_{(X+Y)hops}(E_{tx}+E_{rx})}{d} = \frac{(X+Y)[\alpha n E_{amp}+nR^2E_{amp}+\alpha n E_{amp}]}{R*(X^2+Y^2)^{\frac{1}{2}}} \quad (5.11)$$

$$= \frac{(X+Y)nE_{amp}(2\alpha+R^2)}{R*(X^2+Y^2)^{\frac{1}{2}}} \quad (5.12)$$

For the static grid based network used in RMASE (Zhang, 2005), *x*, *y*, *n*, and E_{amp} are constant in the S_x by S_y grid topology. The minimum transmission cost can be achieved if the expression (5.12) above is set to the worst case scenario as:

$$T'_{cav} = \frac{d}{dR}\left[\frac{(X+Y)nE_{amp}(2\alpha+R^2)}{R*(X^2+Y^2)^{\frac{1}{2}}}\right] = 0 \quad (5.13)$$

That is, $\frac{(X+Y)nE_{amp}}{(X^2+Y^2)^{\frac{1}{2}}}\left[\frac{d}{dR}\left[\frac{(2\alpha+R^2)}{R}\right]\right] = 0$

or $\frac{d}{dR}[2\alpha R^{-1} + R] = 0$

And $-2\alpha R^{-2} + 1 = 0 \quad (5.14)$

→ $-2\alpha = -R^2$

∴ $R = (2\alpha)^{1/2} \quad (5.15)$

It then implies that, the average transmission cost occurred when $R = (2\alpha)^{1/2}$.

5.3.3 Energy model

Limiting the energy consumption of a WSN is as important as improvements to the reliability of the network. It is of important to know the theory and mathematical formulation behind the energy available and energy consumption of any component of a given system. WSN generally, has limited available energy for the nodes. The way in which this limited energy is consumed by any component of the system and also in the course of event transmission in the system is of paramount interest to the designer and the users of sensor network systems. As such, we intend to analyze the behavior of the sensor network system and its protocols by varying some of the important parameters that may lead to energy

consumption so that we can arrive at an optimal point and then find a way to improve on any system with the aid of the derived formulation.

5.3.3.1 Basic static network energy consumption for direct and multi-hop routing

In this section, we will be more focused on the energy consumption analysis of termite based routing protocol for WSN as proposed in (Zungeru et al., 2012), and described in detail in Chapter 4. Though, it is also of importance to examine the energy consumption of the two modes of communication, that is, the direct communication with the sink node, and multi-hop routing using the termite based routing algorithm.

In the first case, using the direct communication protocol, sensor nodes send their information to the sink node without using any intermediate node; in this case, one-hop is utilized. But if the sink node is at a far distance from the source node, much energy will be required to transmit a single information or packet to the sink node. In other words, the network lifetime will be reduced since the participating node's battery will be drained at a faster rate as the distance R in equation (5.9) is large. Now considering the case whereby intermediate nodes have to be used for the relaying function, that is, serving as routers and at the same time sources (multi-hop routing), this serves as minimum energy routing. In this case, nodes route information destined for the sink node via intermediate nodes. Thus, the source nodes also act as routers for other source nodes in addition to sensing functionality.

However, even among the multi-hop routing protocols, there still exist some differences in the way the routes are chosen. For example, as surveyed and studied in (Zungeru et al., 2012), there are proactive, reactive, and hybrid routing protocols. In reactive protocols, each protocol computes routes when they are needed. In this class, each node store routes only to its immediate neighbors, and determine multi-hop routes as

required. Routing table maintenance overhead is drastically reduced in lieu of the time required to send a message. Determination of the path to the sink node is needed whenever a packet is to be transmitted across multi-hop to the sink. In proactive protocols, routes are computed before they are actually needed, which are also stored in a table format called the routing table at each node. The settling time for a network utilizing proactive routing protocol is extremely high, and the number of messages exchanged in order to maintain routing information does grow at an increasing rate, hence, limiting the scalability of the protocols. In hybrid protocols, the strength of proactive and reactive protocols is combined. The protocols utilize a proactive scheme within a given radius, while using the reactive scheme to determine the routes to nodes outside the radius. The radius is always a function of some metrics like the number of hops. From these analyses of the three classes of path establishment, reactive path establishment looks more promising in terms of energy efficiency since, fewer control packets (overhead) will be utilized in the route discovery process. To this end, we analyze the energy consumption of a promising reactive routing protocol (termite-hill) for WSN. Detailed description of the protocol is given in Chapter 4.

Termite-hill routing protocol utilizes the multi-hop routing approach. Unlike the direct communication approach between sensor node and a sink node where a high energy is needed to transmit a single packet of information, Termite-hill algorithm utilizes multiple hops to send its packets by evenly distributing the energy consumption between intermediate nodes. Each data packet must go through n (low energy) transmits and n receives. But depending on the relative costs of the amplifier at the transmitter and radio electronics, the total energy required in the transmission and reception might be greater using this multiple hop routing than direct transmission to the sink node. Though, the energy consumption

for nodes participating in direct communication increases rapidly with transmission distance. But the major problem is that the bandwidth available for WSN is limited and the transmission power is low, hence, direct communication is not feasible in most cases for a typical sensor network. In some cases, each sensor node serving as a source drained faster in energy thereby creating an energy-hole in the network and in turn, limiting the network lifetime. Therefore, to illustrate this statement, we consider a simple linear network as shown in Fig. 5.5. Considering the linear network, we looked at the energy consumption in transmitting a *b-bit* of information from a source node at a distance $(n)R$ away from the sink node, where *R* is the distance between the nodes in the network, and *n*, the number of hops. In this multiple hop routing approach, every node in the network sends information only to its nearby node among its neighbors towards the destination node. This then means that a node of distance $(n)R$ to the sink node will needs *n* number of transmissions, with distance *R*, and *n-1* receives times. Therefore, the energy consumption for a multi-hop routing protocol is:

$$E_{CMH} = E_{tx} + E_{rx} \tag{5.16}$$

$$= nE_{tx}(n_1, R) + (n-1)E_{rx}(n_1) \tag{5.17}$$

But $E_{rx} = \alpha n E_{amp}$ from (8), where $n_1 = b$ in this case, and $E_{tx} = nE_{amp}(\alpha + R^2)$ from (5.5), that is,

$$E_{tx}(n_1, d) = bE_{amp}(\alpha + R^2) \tag{5.18}$$

$$\therefore E_{CMH} = n\left(bE_{amp}(\alpha + R^2)\right) + (n-1)\alpha bE_{amp} = nbE_{amp}(\alpha + R^2) + (n-1)\alpha bE_{amp}$$

$$= nb\alpha E_{amp} + nbR^2E_{amp} + nb\alpha E_{amp} - \alpha bE_{amp} = bE_{amp}(2\alpha n + nR^2 - \alpha) \tag{5.19}$$

But if the communication between the source nodes and the sink is the direct one, i.e. without using intermediate nodes, the expected energy consumption will vary slightly from the expression in (5.19) above as:

$$E_{CD} = E_{tx}(b, R_d)$$

$$= bE_{amp}(\alpha + R^2) \tag{5.20}$$

Where R_d represent the distance between the source node and the sink node. It is also a function of *n-by-R*-distance from the sink node. That is $R_d = nR$.

$$\therefore \quad E_{CD} = bE_{amp}(\alpha + (nR)^2) = bE_{amp}(\alpha + n^2R^2) \tag{5.21}$$

Comparing equations (5.19) and (5.21), the energy consumption in the network will be less for multi-hop transmission if $E_{CMH} < E_{CD}$ that is:

$$bE_{amp}(2\alpha n + nR^2 - \alpha) < bE_{amp}(\alpha + n^2R^2) \tag{5.22}$$

Now, if we consider the network in terms of the strength of transmit electronics, amplifiers at each node, and the receiver electronics, the expected energy consumption for direct transmission to the sink will be described as:

$$E_{tx}(b, R_d) = E_{tx}(b, (nR)) \tag{5.23}$$

Where $R_d = nR$ as defined above and the direct transmission energy consumption will then be,

$$E_{txD} = bE_{elec} + b(nR)^2E_{amp}$$

$$\therefore \quad E_{CD} = bE_{elec} + bn^2R^2E_{amp} = b(E_{elec} + n^2R^2E_{amp}) \tag{5.24}$$

For the multi-hop, energy consumption involves the *n* transmission and *(n-1)* reception. That is:

$$\begin{aligned} E_{CMH} &= nE_{tx}(b, R) + (n-1)E_{rx}(b) \\ &= n\left(bE_{elec} + bR^2E_{amp}\right) + (n-1)bE_{elec} \\ &= b\left[\left(nE_{elec} + nR^2E_{amp} + nE_{elec} - E_{elec}\right)\right] \\ &= b\left[2nE_{elec} - E_{elec} + nR^2E_{amp}\right] \\ &= b\left[nR^2E_{amp} + (2n-1)E_{elec}\right] \end{aligned} \tag{5.25}$$

also, comparing equations (5.24) and (5.25) for the energy consumption using the direct and multiple hops approach, the energy consumption using multi-hop transmission will be less than direct transmission if $E_{CMH} < E_{CD}$, that is:

$$b\left[nR^2E_{amp} + (2n-1)E_{elec}\right] < b(E_{elec} + n^2R^2E_{amp}) \quad (5.26)$$

or, $nR^2E_{amp} + (2n-1)E_{elec} < E_{elec} + n^2R^2E_{amp}$

That is, $2E_{elec} - 2nE_{elec} > nR^2E_{amp} - n^2R^2E_{amp}$

And $2E_{elec}(1-n) > nR^2E_{amp}(1-n)$

$$\therefore \quad \frac{E_{elec}}{E_{amp}} > \frac{nR^2}{2} \quad (5.27)$$

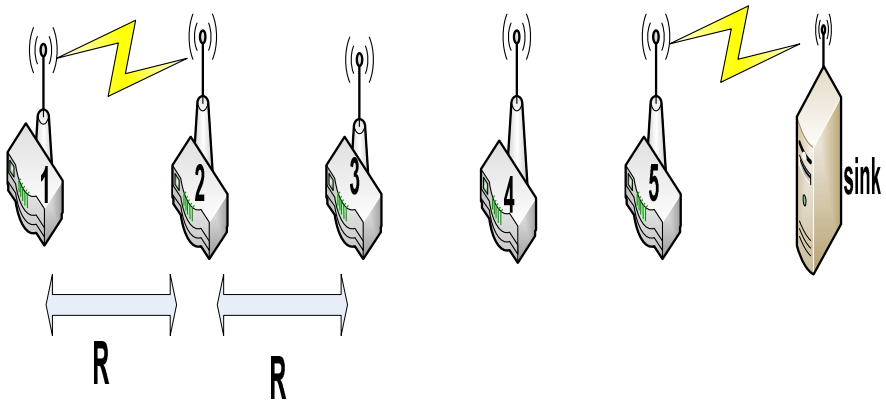


Fig. 5.5: A Linear Network.

5.3.3.2 Analysis of energy consumption of Termite-hill routing algorithm with mobile sink

In this section, we focused more on energy consumption of the termite-hill routing algorithm as an example of a multi-hop, reactive and swarm intelligence based routing protocol. Detail explanation of the algorithm is given in Chapter 4 above. The expected energy consumption per unit time of Termite-hill is the summation of the energy required to discover active paths towards the sink node in the network (overhead energy consumption), and the energy required to transmit a useful information to the sink node. The total energy consumption is:

$$E_{CT}(\lambda_p, \lambda_T) = E_{overhead}(\lambda_p, \lambda_T) + E_{data}(\lambda_p, \lambda_T) \quad (5.28)$$

Where λ_P and λ_T are the packet arrival rate at each node, and the occasional change in topology respectively, $E_{overhead}$, represents the energy consumed in detecting a valid route, and E_{data} is the energy consumption in transmitting the sensed event towards the sink node. Whenever an event

is sensed by a sensor node, and the node want to pass the information about the sensed event to the sink node, it will first search its routing table for a valid route to the destination node. If the route is active in its cache memory, it will then have to transmit the information without generating a *soldier* (overhead message) to discover any active path. Otherwise, it will have no option than to launch a *forward soldier* to search for a valid route towards the destination node. The *forward soldier* is retransmitted within nodes in the network that received it until it reaches the destination node. The energy consumed in this process is measured by:

$$E_{FSoldier} = N \times b_f \times (E_{tx}(r) + \frac{\pi r^2}{A}(N-1)E_{rx}) \quad (5.29)$$

Where *A* is the square area of the topology, b_f represents the bit value for a *forward soldier* message, *N* equals to the total number of network nodes, $E_{tx}(r)$, and $E_{rx}(r)$ are the energy consumption in the transmission and reception respectively. Then, the destination node responds by propagating backward *soldier* back to the source node in a one way manner, hence, the energy consumed by the *backward soldier* can be estimated by:

$$E_{BSoldier} = b_b \times h \times (E_{tx}(r) + E_{rx} + \frac{\pi r^2}{A}(N-2)E_{listen}) \quad (5.30)$$

Where *h* represents hops count between the source node and the destination node, and b_b the number of bits for a *backward soldier*. Though, in a situation whereby a link fails in the process of discovering a route to the destination node, the nodes which serves as routers between the source of the information and the destination node (intermediate node) that detect the link failure, will have to send a seeding message (route error) along the path to the source node indicating a link failure. And the energy consumed by the seeding message is:

$$E_{SEED} = b_L \times h \times (E_{tx}(r) + E_{rx} + \frac{\pi r^2}{A}(N-2)E_{listen}) \quad (5.31)$$

b_L is the number of bits of link reply message (seeding). In order to describe the packet flow in WSNs, we assumed a constant inter-arrival

period of the packets at time $T = \frac{1}{\lambda_p}$. When a sensor node detects an event and has a valid route to the destination during time *T*, it immediately forward the information about the detected event along the valid route without creating a forward soldier, otherwise, it creates a forward soldier which also carries the first detected event to search for a valid route. The process of sending the soldier in the network is a repetitive process until it eventually gets to the sink and back to the node where it was created for the valid route. And the energy consumed in the process by Termite-hill is:

$$E_{overhead}(\lambda_p, \lambda_T) = N\lambda_p((1 - P_h(T, \lambda_T)) \times E_{max.discovery}(n, T_d(h), \lambda_T)) \quad (5.32)$$

Where $P_h(T, \lambda_T) = e^{-\lambda_T hT}$ represents the validity of path with regards to the probability of availability of the path with *h* hops in time *T* of topology change will take place. Interested readers can see (La and Han, 2007) for the derivation. If by any chance a *backward soldier* does not get back to the node where it was created originally, the originator creates another *forward soldier* for the same purpose, but during this time, a limit is set for it regarding the number of hops (*n*), it is expected to take on a round movement back to the source node, and this is computed as: $E_{max.discovery}(n, T_d(h), \lambda_T)$

$$= \sum_{k=1}^{n} \left\{ \left(1 - P_h(T_d(h), \lambda_T)\right)^{k-1} P_h(T_d(h), \lambda_T) \right.$$
$$\left. \times \left(kE_{FSoldier} + (k-1)\left(\frac{E_{BSoldier} - E_{SEED}}{2}\right) + E_{BSoldier} \right) \right\}$$
$$= \sum_{k=1}^{n} \left\{ \left(1 - P_h(T_d(h), \lambda_T)\right)^{k-1} P_h(T_d(h), \lambda_T) \times (kE_{FSoldier} + kE_{BSoldier}) \right\}$$
$$= (E_{FSoldier} + E_{BSoldier}) \sum_{k=1}^{n} k \left\{ \left(1 - P_h(T_d(h), \lambda_T)\right)^{k-1} P_h(T_d(h), \lambda_T) \right\}$$
$$= (E_{FSoldier} + E_{BSoldier}) \left(\frac{1}{P_h(T_d(h), \lambda_T)} - \frac{\left(1 - P_h(T_d(h), \lambda_T)\right)^n}{P_h(T_d(h), \lambda_T)} - n\left(1 - P_h(T_d(h), \lambda_T)\right)^n \right) \quad (5.33)$$

Where *n* is a number that represents the limit of hops in terms of retransmissions set for the soldier. The total energy consumed for a successful reception of *the backward soldier message* is:

$$lim_{n \to \infty} E_{max.discovery}(n, T_d(h), \lambda_T) = \frac{(E_{FSoldier} + E_{BSoldier})}{P_h(T_d(h), \lambda_T)} \quad (5.34)$$

However, the retransmission process will add to energy consumption in the network. But in our case, we limit the retransmission attempts to a

maximum of three (3) as will be seen in the simulation parameters. If a route is successfully discovered, the node transmits the sensed information towards the destination node based on what it has learned and stored in the routing table to the sink node, and in return, the sink node acknowledges the reception of the information. The energy consumed in this process is:

$$E_{c.d} = (b_{data} + b_{ack} + b_f + b_b) \times h \times (E_{tx}(r) + E_{rx} + \frac{\pi r^2}{A}(N-2)E_{listen}) \quad (5.35)$$

b_{data}, and b_{ack} are the bits length of data and acknowledgement message respectively. If for instance a node tried retransmission for r-information using the maximum limit of retransmission defined by *n*, so as to get successful delivery of the information at the sink node, for *h*-hops distance, the duration of the retransmission is assumed a mean value $T_d(h)$, which is randomly distributed, and is achieved using the expression:

$$\mathrm{E_{r-data}}(\lambda_p, \lambda_T)$$

$$= \mathrm{N}\lambda_p \sum_{k=1}^{r} \left\{ \left(\left(1 - \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T)\right)^{k-1} \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T) \times \left(\mathrm{E_{c.d}} + (k-1)\left(\frac{\mathrm{E_{c.d}}}{2}\right)\right)\right) + \right.$$

$$\left. \left(\left(1 - \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T)\right)^{r}\right) \frac{\mathrm{E_{c.d}}}{2} r \right\}$$

$$= \mathrm{N}\lambda_p \sum_{k=1}^{r} \left\{ \left(\left(1 - \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T)\right)^{k-1} \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T) \times \left(\left(\frac{\mathrm{E_{c.d}}}{2} k + \frac{\mathrm{E_{c.d}}}{2}\right)\right)\right) + \left(\left(1 - \right.\right.\right.$$

$$\left. \left. \left. \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T)\right)^{r}\right) \frac{\mathrm{E_{c.d}}}{2} r \right\}$$

$$= \mathrm{N}\lambda_p \frac{\mathrm{E_{c.d}}}{2} \sum_{k=1}^{r} \left\{ \left(\left(1 - \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T)\right)^{k-1} \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T)(k+1)\right) + \left(1 - \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T)\right)^{r} r \right\}$$

$$= \mathrm{N}\lambda_p \frac{\mathrm{E_{c.d}}}{2} \left\{ \left(\frac{1 - \left(1 - \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T)\right)^{r}}{\mathrm{P_h}(\mathrm{T_d}(h), \lambda_T)} \right) + 1 - \left(1 - \mathrm{P_h}(\mathrm{T_d}(h), \lambda_T)\right)^{r} \right\} \quad (5.36)$$

Where $N\lambda_p$ represents the average of a number of information send across the network over time, and as *r* tends to infinity ($r \to \infty$), the used energy for retransmission of the information is approximated using:

$$lim_{r\to\infty} E_{r-data}(\lambda_p, \lambda_T) = N\lambda_p \frac{E_{c.d}}{2}\left(\frac{1}{P_h(T_d(h), \lambda_T)} + 1\right) \quad (5.37)$$

It will be observed in the later section (Section 5.5.3) that due to data retransmission and several route discovery processes when the network topology changes, our protocol consumes more energy on the process of the route discovery, but its reliability of packet delivery gets higher. It will also be observed that for a lighter data packet with higher occasional change in topology, Termite-hill performance is excellent, though it tends to degrade with increase in size of data packet and lower change in network topology. This implies that, Termite-hill is designed to balance both static and dynamic networks as seen in the results of Chapter 4.

5.4 Event Extraction Analysis in Wireless Sensor Network

This section models the system such that maximum information is expected to be extracted from the limited energy resource wireless sensor network (WSN) nodes. We assume that all sensor nodes are randomly placed on a grid sensor network as shown in Fig. 5.3, which are normally static in their respective position. Each of the sensor nodes is also equipped with an energy of E_{batt}, and each pair of sensor nodes on the grid as defined earlier in Section 5.3.1, is d_{ab}, where d_{ab} is the actual distance between nodes *a* and *b* on the sensor grid. Assuming that much information is needed to be extracted in the network and to be delivered to the sink node with unlimited energy resource (most sink nodes normally plug into the regular power supply). Each node in the network consumes E_{rx} of energy per bit received, E_{tx}/bit transmits, and E_{ss}/bit sensed. Now, if we use the additive Gaussian noise (AWGN) channel (Emans, 2012) for capacity expression. As adopted in (Moreno et al., 2012), and in accordance with (Sadagopan, 2004), we also assumed that there is no event aggregation with this model, that is to say that, all information is relayed to the sink node either directly or using multi-hops. Assuming the total energy consumed at node *a* denoted by E_a, equals $\sum(E_{rx} + E_{tx} + E_{ss})$,

and this energy under normal circumstance, should not exceed the total amount of energy available for the node *a*. From the above assumptions, we represent the problem as a nonlinear program as:

$$Max \sum_{b=1}^{N} I_{bN+1}$$

Such that,

$$\sum_{b=1}^{N+1} I_{ab} - \sum_{b=1}^{N} I_{ba} \geq 0 \quad (5.38)$$

$for\ a = 1{:}N$

That is, $$\sum_{b=1}^{N+1} I_{ab} - \sum_{b=1}^{N} I_{ba} \leq k \sum_{b=1}^{N} I_{bN+1} \quad (5.39)$$

K represents the number of bits of information that is generated and sent to the sink node by each sensor node in the network. Now, for a signal decay d_{ab}^{-2}, and noise of factor ζ, we can model the system such that the total energy consumed at node a, *Ea* should be less than the total energy available for the node. That is $E_a \leq E_{batt}$. That is to say that:

$$\sum_{b=1}^{N+1} E_{ab} + E_{rx} \sum_{b=1}^{N} I_{ba} + E_{ss}\left(\sum_{b=1}^{N+1} I_{ab} - \sum_{b=1}^{N} I_{ba}\right) \leq E_{batt} \quad (5.40)$$

$$for\ a = 1{:}N$$

and $$I_{ab} \leq log\left(1 + \frac{E_{ab} d_{ab}^{-2}}{\zeta}\right) \quad (5.41)$$

$$for\ a = 1{:}N, b = 1{:}N+1$$

Such that, $$I_{ab} \geq 0,\ E_{ab} \geq 0 \quad (5.42)$$

$$for\ a = 1{:}N, b = 1{:}N+1$$

Expression (5.40) has a nice interpretation, such that the total energy consumed is a function of "E_{ss}", transmitting cost of information among all pairs of nodes *a*-*b*, which is a function of the energy node *a* spent in transmitting (E_{ab}) and the cost node *b* spend in receiving $E_{rx}I_{ba}$. Also, the total energy available on each node should be greater than the sum of energy expended in transmission, sensing and reception at each node as shown in (Eqn. 5.40) at the end of every information processing. This is a good assumption for network lifetime elongation and maximum throughput. If we sum the total energy consumption of node *a* in the network with the aim of also extracting as much information as possible in the sensor network, we will then have the overall energy consumed by the

sensor nodes as a function of information. The summation of all the energy consumption for every node *a* in the network is given as:

$$\sum_{a=1}^{N} E_a$$

$$= \sum_{a=1}^{N}\left[E_{ss}\left(\sum_{b=1}^{N+1} I_{ab} - \sum_{b=1}^{N} I_{ba}\right) + \sum_{b=1}^{N+1} E_{ab} + \sum_{b=1}^{N} E_{rx} I_{ba}\right]$$

$$= E_{ss}\sum_{a=1}^{N} I_{aN+1} + \sum_{a=1}^{N}\sum_{b=1}^{N+1} E_{ab} + E_{rx}\sum_{a=1}^{N}\sum_{b=1}^{N} I_{ab}$$

$$= \sum_{a=1}^{N}(E_{ss} I_{aN+1} + E_{aN+1}) + \sum_{a=1}^{N}\sum_{b=1}^{N}(E_{rx} I_{ab} + E_{ab}) \quad (5.43)$$

From (5.43), it is clearly seen that the amount of message achievable using the limited resource WSN, is a function of the total available energy of the network nodes. If we now try to minimize the total energy usage in the network and extract as much information as possible, we will have at least I_{min} information delivered to the sink node. The above statement can be represented analytically and replacing the energy bound constraint (5.40) by a more generalized consumption as:

$$min \sum_{a=1}^{N}(E_{ss} I_{aN+1} + E_{aN+1}) + \sum_{a=1}^{N}\sum_{b=1}^{N}(E_{rx} I_{ab} + E_{ab})$$

Such that,

$$\sum_{b=1}^{N+1} I_{ab} - \sum_{b=1}^{N} I_{ba} \geq 0 \quad (5.44)$$

$$for\ a = 1{:}N$$

And, $$\sum_{b=1}^{N+1} I_{ab} - \sum_{b=1}^{N} I_{ba} \leq k \sum_{b=1}^{N} I_{bN+1} \quad (5.45)$$

$$for\ a = 1{:}N$$

That is, $$\sum_{b=1}^{N} I_{bN+1} \geq I_{min} \quad (5.46)$$

$$for\ a = 1{:}N$$

It then implies that, $$I_{ab} \leq log\left(1 + \frac{E_{ab} d_{ab}^{-2}}{\zeta}\right) \quad (5.47)$$

$$for\ a = 1{:}N, b = 1{:}N+1$$

Such that, $$I_{ab} \geq 0,\ E_{ab} \geq 0 \quad (5.48)$$

$$for\ a = 1{:}N, b = 1{:}N+1$$

5.5 Computational and Simulation Experiments

In this study, we considered (1) a line based sensor network topology, and (2) a rectangular randomly distributed grid based sensor network for our analysis. In the first and second part of the analytical experiment, we used the parameters in (Heinzelman et al., 2000; Panda and Patra, 2010; Moreno et al., 2012). While in the last part of the experiment, we compared the analytical results with the simulation results, we used the energy model of Waspmote (Libelium, 2008, 2010) sensor nodes. The change in the parameters is to aid us in fair comparison between the analytical results and simulation results. For each pair of communication, both the current drawn in reception and transmission, data rate, minimum supply voltage are given in the literature. We used RMASE simulation environment. In the first topology, we assume that each pair of nodes is separated by a distance R, with all nodes are on line L and the sink node is at a distance nR from the source, where n represents the number of hops or nodes in-between the source and the sink node. The grid topology considers a rectangular grid network where nodes are randomly distributed on a grid sensor network of size S_x, S_y in the x and y directions, and each pair of nodes are also separated by a distance R equal to their transmission range.

In our experiments comprising of computation as well as simulation, we illustrate different possible models to analyze transmission cost, energy consumption for both direct communication and multihop communication with variable packet size in bits, network size, and transmission distance. We further strengthen the analysis by modeling the energy consumption of a typical reactive routing protocol for wireless sensor network with respect to network size, topology change rate and packet arrival rate (light traffic load and heavy traffic load). The analytical results were then compared with the simulation results. The parameters used in the first part of our

analytical results in accordance with (Heinzelman et al., 2000; Panda and Patra, 2010) where $E_{elec} = 100nJ/bit$, $E_{amp} = 500\mu J/(bit * m^2)$, transmission distance, packet size, and number of nodes used in the experiments were varied accordingly as shown in Fig. 5.6 and Fig. 5.7. In Table 5.1, are the analytical and simulation parameters setting for the last part of our experiment.

Table 5.1: Analytical and Simulation Parameters

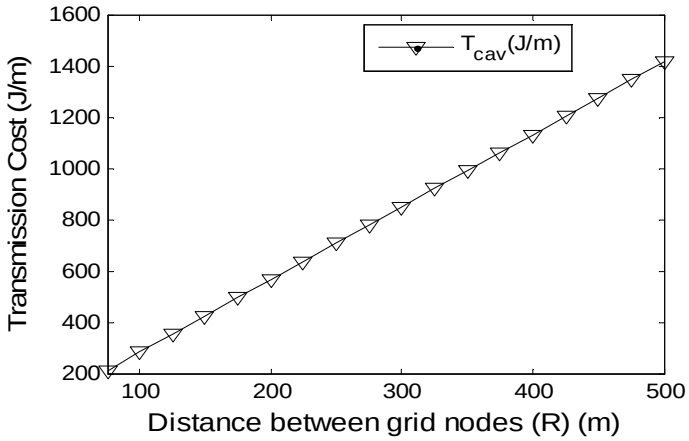
Parameters	Values
Routing Protocol	Termite-hill
Size of Topology (A)	100 x 100
Nodes Distribution	Random distribution
Maximum number of Retransmission (n)	3
Transmission distance (R)	35 m
Data Traffic	Constant Bit Rate (CBR)
Data Rate	250 kbps
Propagation model	Probabilistic
Energy consumption	Waspmote-802.15.4
Time of topology change	2 s
Simulation Time, Average Simulation times	360s, 10
Number of Nodes (N)	441

5.5.1 Average Transmission Cost of a Grid Based Sensor Network

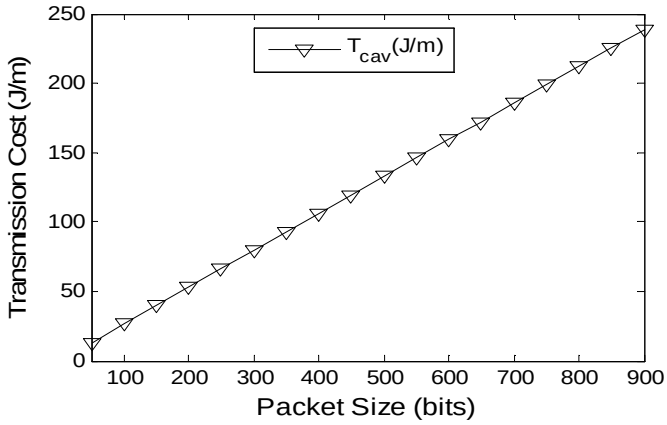
In this section, we analyze the average transmission cost of a grid based wireless sensor network. From Fig. 5.6, derived from the computational results of Eqn. (5.12), we found out that the transmission cost in joules per unit meter, is directly proportional to both the distance between the nodes (R), packet size (n), and the number of nodes (hops) in the X and Y directions. Fig. 5.6 (a) is a representation of the effect of transmission distance on the average cost of transmission. As it is seen, for every increase in transmission distance (R), there is a high rise in energy

consumption per node. The rise in transmission cost is as a result of the proportionality between the transmission cost and the transmission distance (R) as described in Eqn. (5.12). The high rise in transmission cost is not favorable for sensor network, since the sensor nodes have limited energy as well as limited bandwidth. For a typical wireless sensor node of the Libelium (Libelium, 2009) as described in (Zungeru et al., 2011), each sensor node can transmit at a maximum distance of 500m, Crossbow MICAz for 100m, and Intel IMote2 for 30m. Even the ones that can transmit at a higher distance has the problem of limited available energy for them, hence it is encouraged to adapt to the multihop means of communication to evenly distribute the energy cost and also limit the transmission distance. Shown in Fig. 5.6(b) is the variation of transmission cost with packet size. There is also an increase in transmission cost for every increase in the number of bits transmitted on the wireless link. The increase in transmission cost is as a result of increase in packet size, this can also be seen in Eqn. (5.12) that the transmission cost is a function of the number of bits transmitted (n). Also, a typical sensor node has low memory and storage capacity, and a sensor frame occupies an average of approximate 100bytes (800bits) of information. With the low memory of 8KB SRAM of Waspmote, it then becomes necessary to limit the overhead cost of transmission so as to save space for data transmission. As can be seen, we compute the transmission cost to a maximum of 900bits of information, of which every increase in bits also corresponds to an increase in transmission cost. In Fig. 5.6(c), we show the relationship between the transmission cost of grid based sensor network and the number of nodes in the network in the form of the number of X and Y hops. Also, the figure shows an increase in the transmission cost with a corresponding increase in the network density. This is true because as shown in Eqn. (5.12), the

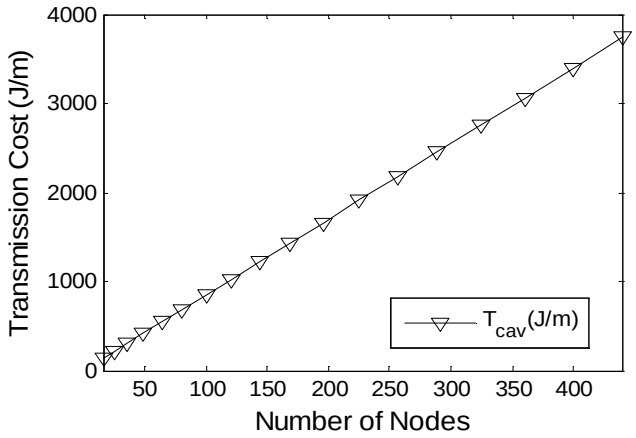
average transmission cost is proportional to the number of hops in the *X* and *Y* directions.



(a)



(b)



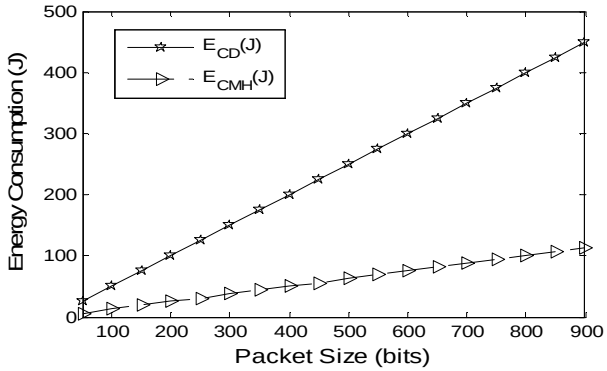
(c)

Fig. 5.6: Effect of (a) Transmission distance (b) Packet Size, and (c) Number of Nodes on the Transmission cost of a grid based sensor network.

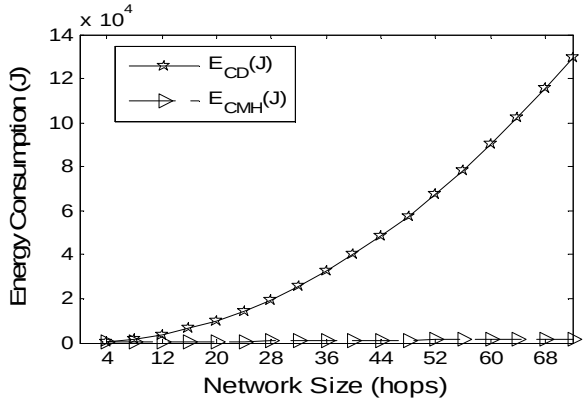
5.5.2 Energy Consumption of a Direct and Multihop Communication in WSN

In this section, we explore how the energy consumption given in Equation (5.19) and (5.21) for multi-hop and direct communication react to an increase in packet size, network size, and transmission distance between source nodes and sink node in a wireless sensor network. The behavior of the network in terms of increase in energy consumption for the direct communications as against the multihop communication due to variable packet size is expected, as shown in Fig. 5.7(a). This is because, the energy consumed in direct transmission is proportional to the square of the number of bits transmitted, and it is measured in joules per bits. This then means that, for multihop communications, the number of bits transmitted is at a short distance between nodes, hence saving more energy. We can see that, as the packet size increases up to a value of 900bits, the energy consumed per node for the direct communication rises up to the value of 450, while it is 113 for multihop. This is a high difference when compared to the energy available per node for network lifetime. Also, as can be seen in Fig. 5.7(b), even with an increase in the number of nodes in the network, the energy consumed by the multihop communication tends to remain the same as the initial consumption in the fewer nodes, whereas, it increases sharply and higher in the case of direct communication per node. This is as expected since the network size increase, the transmission range increases, which is not favorable for direct communication. As for multihop, as the network size increases with an increase in the number of nodes, it has not much effect as neighboring nodes serve as routers and hence, same or equivalent energy is expected to be consumed. Fig. 5.7 (c) shows the effect of transmission distance on energy consumption of grid based sensor network. As can be seen, though, there is a little increase in energy consumption of multihop communication, but the increase in that of the

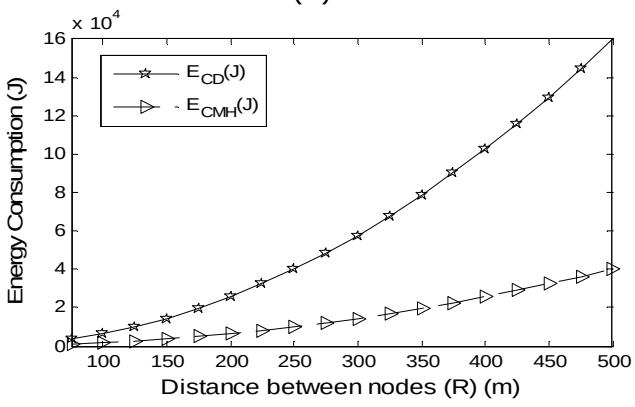
direct communication is much higher. The increase in the node per distance is an indirect way of increasing the network area. The increase in the network area means that much energy will be required per node for transmitting a bit of information as the energy consumption is a function of transmission distance (R) between nodes as shown in Equation (5.19).



(a)



(b)



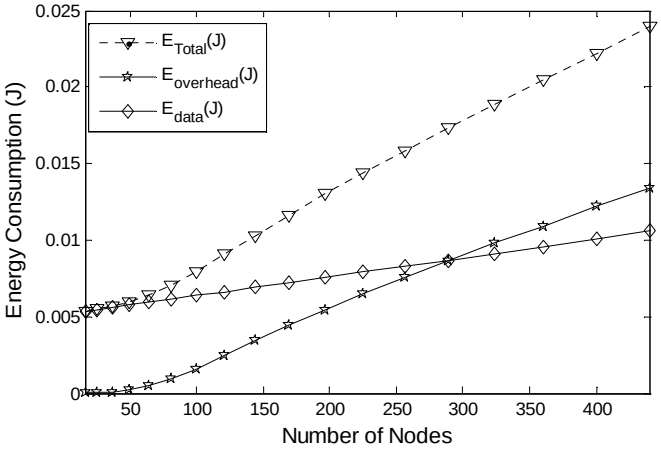
(c)

Fig. 5.7: Energy Consumption of a Direct and Multihop Communication in WSN with respect to (a) Packet Size (b) Network Size (c) Transmission distance.

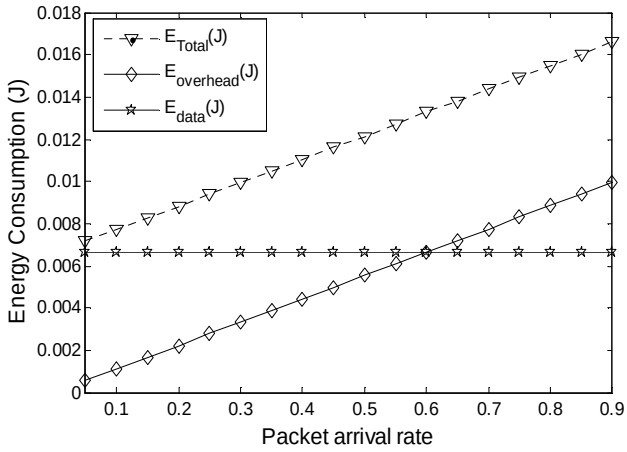
5.5.3 Energy Consumption of a Termite-hill Routing Protocol in WSN

In this section, we compare the computational results and the simulation results of Termite-hill routing algorithm in terms of energy consumption with respect to network size, topology change rate, and packet arrival rate. Equation (5.28), (5.32), and (5.35) were used for the analytical results. Fig. 5.8(a)-(c) shows the effect of high number of retransmissions on the energy consumption of Termite-hill routing algorithm. As shown in Fig. 5.8 (a), as we vary the mobility from 0.05 to 0.9 for every step of 0.05, the total energy consumption increases, also, at the point of high network density of about 289 nodes corresponding to high mobility of 0.6, we have the energy consumed due to control traffic equals the energy consumed in transmitting the data packets. After that point, the energy consumption as a result of control traffic grows above that of the data, hence, many packets were dropped and the system encounters unnecessary and wasted energy. Also shown in Fig. 5.8 (b) and (c), there is a point at which the energy consumption due to control packets becomes higher than the energy used to transmit the data packets. But in Fig. 5.8 (c), with a very high number of retransmission of about 1000, the energy consumption of the system due to control packets approaches the total energy consumed in the system, and at that instance, then it shows that it is necessary to limit the number of retransmission so as to save energy for transmitting the real information (data packets). Fig. 5.9 (a)-(c) shows a similar behavior to that of Fig. 5.8, but the main difference here is that, the number of retransmission was varied for each variation of number of nodes, mobility and traffic load. The most important observation is in Fig. 5.9 (a), which shows that, for high network density, the energy consumption due to control traffic approaches the total energy consumption as we increase the number of retransmission of the control

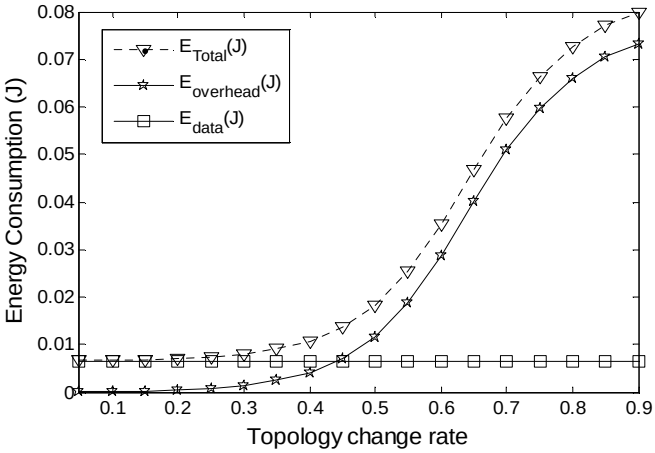
packets. Fig. 5.10 (a) shows the relationship between the total energy consumed in the network using the analytical method, overhead energy consumption, energy consumption for data transmission, and energy consumed by the termite-hill using the RMASE simulation environment with respect to variation of network density. Fig. 5.10 (a) shows that the total energy consumption is comparable to the energy consumed in data transmission since the overhead energy consumption by the termite-hill protocol is negligible. This is the point at which the retransmission was reduced to optimal value (three). This behavior is expected because, Termite-hill routing algorithm utilizes control packets when needed, that is, forward soldiers are sent when the path to the sink is not known. But if the path is valid in the routing table, the event to be forwarded is sent to any of the neighbors who have the knowledge of the location of the sink node. That is to say that, less overhead is utilized by Termite-hill as a reactive routing protocol due to less number of route discovery packets, and limitation of the maximum number of retransmission of packets to three as described in Equation (5.28) through (5.37) of energy consumption of Termite-hill. It is also worth noting that, the analytical results varied slightly with the simulation results due to the environment of simulation and other simulation parameters which are prone to errors and the limitation of retransmission of *forward soldiers* and data packets. Though, our main goal is to evaluate the effect of overhead cost on the total energy consumption of Termite-hill algorithm. As shown in Fig. 5.10(b), and 5.10(c), the topology change rate as well as the packet arrival rate has less effect on the energy consumption when the number of retransmission is reduced to minima. Though for high traffic load and at high mobility, the energy consumption tends to increase slowly, but the increase is minimal.



(a) $n = 50,\ \lambda_p = 0.5,\ \lambda_T = 0.05 - 0.9\ at\ step\ 0.05$

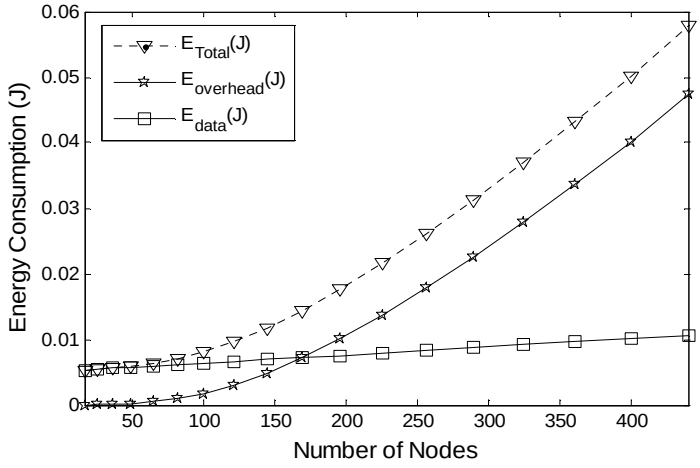


(b) $n = 100,\ \lambda_T = 0.5,\ N = 120$

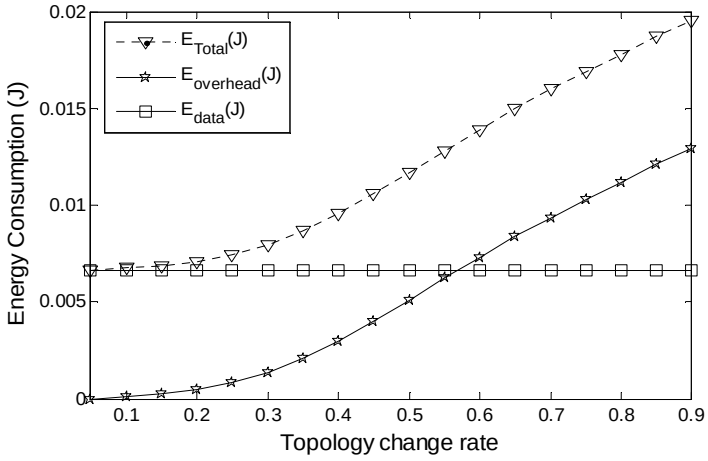


(c) $n = 1000,\ \lambda_P = 0.5,\ N = 120$

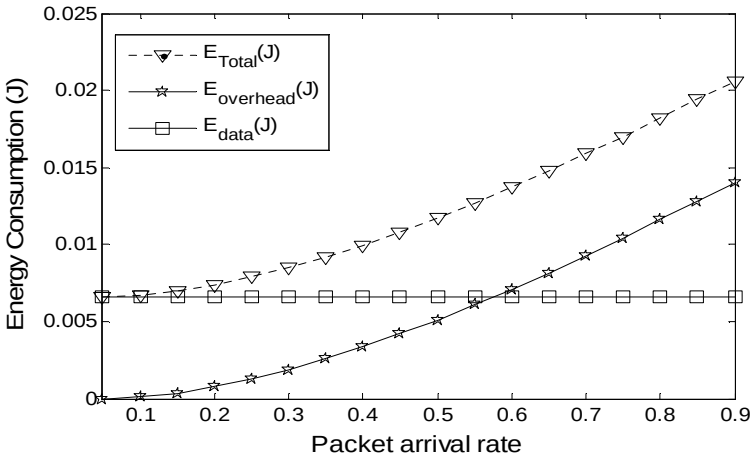
Fig. 5.8: Analytical results for Effect of high number of retransmission on Energy Consumption of termite-hill in WSN with respect to (a) Number of nodes (b) Packet arrival rate (c) Mobility.



(a) $n = 1 - 170\ at\ step\ 10,\ \lambda_P = 0.5,\ \lambda_T = 0.05 - 0.9\ step\ 0.05$

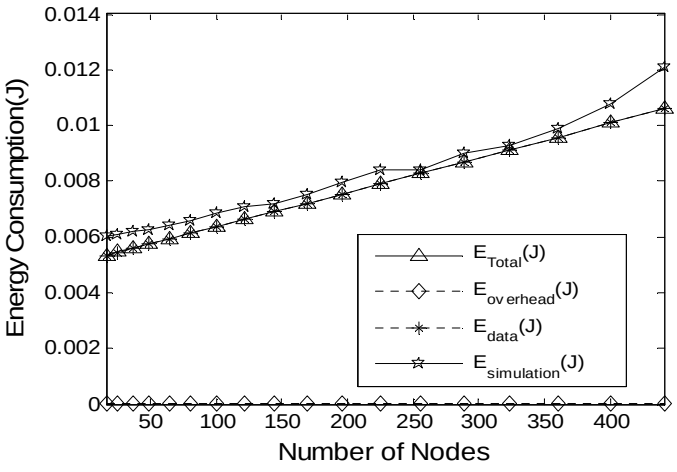


(b) $n = 1 - 170\ at\ step\ 10,\ \lambda_P = 0.5,\ N = 120$

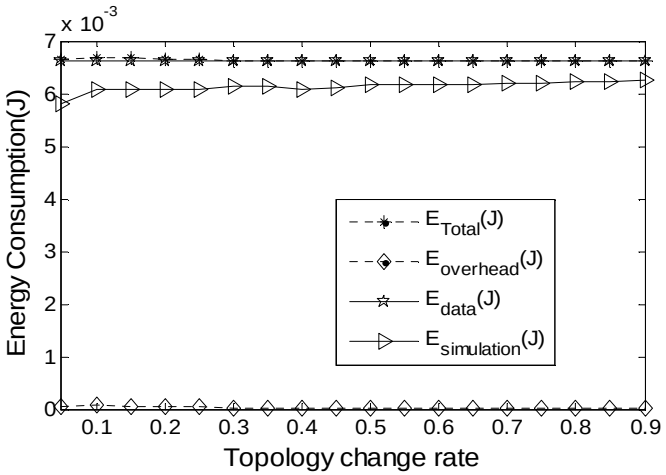


(c) $n = 1 - 170\ at\ step\ 10,\ \lambda_T = 0.5,\ N = 120$

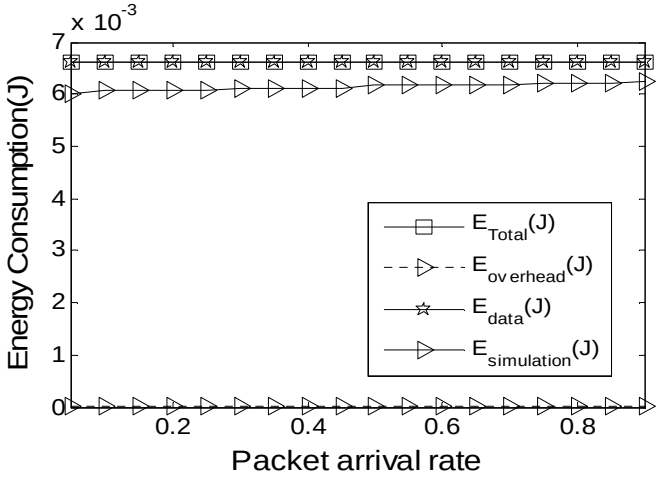
Fig. 5.9: Analytical results for Effect of varying number of retransmission on Energy Consumption of termite-hill in WSN with respect to (a) Number of nodes (b) Mobility (c) Packet arrival rate.



(a)



(b)



(c)

Fig. 5.10: Analytical and Simulation Energy Consumption of termite-hill in WSN with respect to (a) Number of nodes (b) Mobility (c) Packet arrival rate.

5.6 Conclusions

In this chapter, we addressed the need for a systematic modeling methodology by developing an analytical framework for energy consumption in WSN environments and information extraction in a grid based and line based randomly distributed sensor network. The work was further illustrated and strengthened through a model of mobility impact on routing energy consumption by deriving the expected energy consumption of a reactive routing protocol for WSN; Termite-hill routing algorithm as a function of packet arrival rate and topology change rate. The results of our mathematical analysis were also compared with the simulation results. Through the analysis, it was found that the performance of WSN systems and its protocols in terms of energy consumption had a strong correlation between transmission distance, mobility, retransmission of control packets and traffic conditions. In the average transmission cost analysis of a grid based sensor network, we found out that the transmission cost in joules per unit meter is directly proportional to both the distance between nodes, packet size, and the number of nodes in the X and Y directions. Hence, for every increase in transmission distance, number of bits transmitted on the wireless link or number of nodes in the network, there is a corresponding increase in transmission cost. But it is more pronounce with an increase in transmission distance as the energy consumed per meter tends to 1400 for 500 meters distance. Also, in comparison between the energy cost of direct and multi-hop communication, with variable packet size, when the packet size increases up to a value of 900bits, the energy consumed per node for the direct communication rises up to the value of 450, whereas it is 113 per node for multi-hop communication. When we vary the mobility of termite-hill algorithm from 0.05 to 0.9 for every step of 0.05, we observed an increase in energy cost of transmission. But at the point of 0.6 corresponding to high mobility and 289 nodes in the network, we have

energy consumed due to control packets equals the energy consumed in transmitting the data packet. It then shows that it is not safe to increase the mobility beyond the point of 0.6 so as to conserve energy for data transmission instead of for control packets. Also, at the point of increase of retransmission of packets up to 1000, the energy consumed due to control packet approaches the total energy cost of the system, which means that it is necessary to limit the number of retransmission so as to save energy. In the course of the analysis, we presented a reactive routing protocol for WSNs to enable us to verify the analytical results. It was also shown from the analytical and simulation results, that the energy consumption of a routing protocol is dependent on the network size, node mobility, and traffic condition. The analytical model for the transmission cost and the energy consumption, helps to strengthen and deepen our understanding of the effect of network density, transmission range, mobility, and traffic load on energy consumption using the termite-hill routing protocol. There are a number of natural extensions of this work we plan to undertake in the future such as further extensive modeling of the pheromone update and evaporation of the routing protocol.

Chapter 6

Radio Frequency Energy Harvesting and Management for Wireless Sensor Networks

6.1 Introduction

Radio Frequency (RF) energy harvesting holds a promising future for generating a small amount of electrical power to drive partial circuits in wireless communicating electronic devices. Reducing power consumption has become a major challenge in WSNs. As a vital factor affecting system cost and lifetime, energy consumption in WSNs is an emerging and active research area. Finite electrical battery life is encouraging companies and researchers to come up with new ideas and technologies to drive wireless mobile devices for an enhanced period of time. Batteries add to size and their disposal adds to environmental pollution. For mobile and miniature electronic devices, a promising solution is available in capturing and storing the energy from external ambient sources, a technology known as energy harvesting. Other names for this type of technology are power harvesting, energy scavenging and free energy, which are derived from renewable energy (Little et al., 1998). In recent years, the use of wireless devices increases at an alarming rate due to its wide area of applications likes mobile phones and WSNs (Bouchouicha et al., 2010). With the increase in this application area, the use of batteries for the wireless devices has come on board since they are mostly mobile, and researchers are also working tirelessly for making sure that the battery lifetime is given utmost attention

by the reduction of power consumption of the devices. Though, some researchers have taken another dimension by recycling ambient energy like the Micro-electromechanical Systems (MEMS) (Beeby et al., 2006). The charging of mobile devices is convenient due to the fact that the mobile user can do that at his/her convenience. Unlike other applications for example WSNs which are situated in inaccessible locations, to change or even replace their batteries remain a difficult task for the users. This problem arises due to the high network constituents, and the distribution is randomly done in a high density environment or the environment of a distribution is hard to access. The research on RF energy harvesting provides reasonable techniques of overcoming the constraints mentioned above.

The rectification of RF signals research has been proposed for several applications. The rectified power strictly depends on the RF power available at the signal generator, the choice of antenna and frequency band. An Energy harvesting technique using electromagnetic energy specifically radio frequency is the focus of this chapter. Generally, Communication gadgets have different types of antennas which aims is to propagate RF energy in most directions, and as such helps in the maximization of connectivity for mobile applications. The energy transmitted from the wireless sources is much higher up to 30W for 10GHz frequency (Hamdesign, 2010), but only a small amount can be scavenged in the real environment. The rest is dissipated as heat or absorbed by other materials. The RF power harvesting technique is also used in Radio Frequency Identification (RFID) tags and implantable electronics devices. Most commonly used wireless sensor nodes consumes few μW in sleep mode and hundreds μW in active mode. A great factor contributing for energy harvesting research and development is an ultra-low-power component.

The management of power available for sensor nodes has been dealt with to an extent using ant-based routing as surveyed in (Zungeru et al., 2012), which utilizes the behavior of real ants searching for food through pheromone deposition while dealing with problems that need to find paths to goals. The simulation behavior of ant colony leads to optimization of network parameters for the WSN routing process to provide a maximum network lifetime.

The main goal of this chapter is to propose practical harvesting of radio frequency energy using Powercast harvesters while managing the harvested and available energy of the sensor networks using our proposed Algorithm, Improved Energy Efficient Ant Based Routing, which help in the optimization of the available energy. The objective is to power efficiently sensor network nodes with or without batteries to maintain network lifetime at a maximum without performance degradation.

This chapter presents a practical approach to RF Energy harvesting and management of the harvested and available energy for wireless sensor networks using the Improved Energy Efficient Ant Based Routing Algorithm (IEEABR) as reported in Chapter 3. The chapter looks at the measurement of the RF power density, calculation of the received power, storage of the harvested power, and management of the power in wireless sensor networks. The routing process uses an IEEABR technique for energy management. Practical and real-time implementations of the RF Energy using Powercast harvesters and simulations using the energy model of a Libelium Waspmote to verify the approach were performed. Finally, the chapter concludes with performance analysis of the harvested energy, comparison of IEEABR and other traditional energy management techniques for wireless sensor networks.

6.2 Review of Energy Harvesting Systems and Power Consumption in WSNs

For proper operation of sensor networks, a reliable energy harvesting techniques are needed. Over the years, so much work has been done on the research from both academic and industrial researchers in large scale energy from various renewable energy sources. Less attention has been paid to small scale energy harvesting techniques. Though, quite a number of works have been carried out on energy scavenging for WSNs. The Efficient far-field energy harvesting (Le et al., 2008) uses a passively powered RF-DC conversion circuit operating at 906MHz to achieve a power of up to 5.5μW. In related works (Jabbar et al., 2010; Lu et al., 2010; Hagerty et al., 2003; Tang and Guy, 2009), all the above considered little available RF energy and use it to power sensor nodes. In a related work, Bouchouicha et al. (2010) studied ambient RF energy harvesting using broadband and narrow band to recover the RF energy. Among but not all of the available Energy harvesting system for Wireless sensors are; solar power, electromagnetic energy, thermal energy, wind energy, kinetic energy, piezoelectric, thermo-electric, electrostatic, blood sugar and others. These could be further classified into three (Seah and Tan, 2010); Thermal energy, Radiant energy, and Mechanical energy. Based on these, a table showing the comparison of the different and common energy scavenging techniques is as shown in Table 6.1. Beside the harvested energy for the sensor network, the consumption of the harvested power for the different mode of the network has to be looked upon before choosing power harvesting source. A review of some power consumption in some selected sensor nodes can be found in (Gilbert and Balouchi, 2008). For some commercial sensor network nodes, the consumption differs, as shown in Table 6.2; power consumption of the nodes differs from manufacturers.

Table 6.1: Comparison of Energy Harvesting Sources for WSNs

Energy Source	Classification	Performance (power density)	Weakness	Strength
Solar Power	Radiant Energy	100mW/cm^3	Require exposure to light, and low efficiency if device is in the building	Can use without limit
RF Waves	Radiant Energy	0.02μW/cm^2 at 5Km from AM Radio	Low efficiency inside a building	Can use without limit
RF Energy	Radiant Energy	40μW/cm^2 at 10m	Low efficiency if out of line of sight	Can use without limit
Body Heat	Thermal Energy	60μW/cm^2 at 5°C	Available only when the temperature difference is high	Easy to build using a thermocouple
External Heat	Thermal Energy	135μW/cm^2 at 10°C	Available only when the temperature difference is high	Easy to build using a thermocouple
Body Motion	Mechanical Energy	800μW/cm^3	Dependent on motion	High power density, not limited to interior and exterior
Blood Flow	Mechanical Energy	0.93W at 100mmHg	Energy conversion efficiency is low	High power density, not limited to interior and exterior
Air Flow	Mechanical Energy	177μW/cm^3	Efficiency is low inside a building	High power density,
Vibrations	Mechanical Energy	4μW/cm^3	Has to exist in surrounding	High power density, not limited to interior and exterior
Piezoelectric	Mechanical Energy	50μJ/N	Has to exist in surrounding	High power density, not limited to interior and exterior

Table 6.2: Comparison of Power consumption of some selected sensor network nodes

Operating conditions	Manufacturers			
	Crossbow MICAz (Crossbow, 2008)	Waspmote (Libelium, 2008, 2009)	Intel IMote2 (crossbow, 2007)	Jennic JN5139 (NXP, 2010)
Radio standard	IEEE 802.15.4/Zigbee	IEEE 802.15.4/Zigbee	IEEE 802.15.4	IEEE 802.15.4/Zigbee
Typical range	100m (outdoor), 30m (indoor)	500m	30m	1km
Data rate	250 kbps	250 kbps	250 kbps	250 kbps
Sleep mode (deep sleep)	15 µA	62 µA	390 µA	2.8 µA
Processor consumption	8 mA active mode	9 mA	31-53 mA	2.7+0.325 mA/MHz
Transmission	17.4 mA (+0 dBm)	50.26 mA	44 mA	34 mA (+3 dBm)
Reception	19.7 mA	49.56 mA	44 mA	34 mA
Supply voltage (Min)	2.7 V	3.3 V	3.2 V	2.7 V
Average power	2.8 mW	1 mW	12 mW	3 mW

6.2.1 Ambient RF Sources and Available Power

A possible source of energy comes from ubiquitous radio transmitters. Radio waves, a part of the electromagnetic spectrum consists of magnetic and electrical component. They carry messages/information by adopting the principle of variation of amplitude, frequency and phase of the wave within a certain frequency band. On contact with a conductor such as an antenna, the Electromagnetic (EM) radiation induces electrical current on the conductor's surface, known as the skin effect. The communication devices use an antenna for transmission and/or reception of data by

utilizing the different frequency spectrum from 10Kz to 30Kz. The maximum theoretical power available for RF energy harvesting is 7.0μW and 1.0μW for 2.4GHz and 900MHz frequency respectively for free space distance of 40 meters. The path loss of signals will be different in environments other than free space (Le et al., 2008). Though for our work using the Powercast harvester, the power available for P2110 which operate at 915MHz is 3.5mW before conversion and 1.93mW after conversion at a distance of 0.6 meters, and 1μW at a distance of 11 meters (Powercast, 2011).

6.3 Radio frequency energy harvesting and the use of powercast harvester

Radio frequency energy harvesting is a process whereby radio frequency energy emitted by sources that generate high electromagnetic fields such as TV signals, wireless radio networks and cell phone towers, but through power generating circuit linked to a receiving antenna, is captured and converted into a usable DC voltage. Most commonly used as an application for radio frequency identification tags, in which the sensing device wirelessly sends a radio frequency to a harvesting device which supplies just enough power to send back identification information specific to the item of interest. The circuit systems which receive the detected radio frequency of the antenna are made on a fraction of a micrometer scale but can convert the propagated electromagnetic waves to low voltage DC power at a distance of up to 100 meters. Depending on concentration levels which can differ through the day, the power conversion circuit may be attached to a capacitor which can disperse a constant required voltage for the sensor and circuit when there isn't any sufficient supply of incoming energy. Most circuits use a floating gate transistor as the diode which converts the signal into generated power but in linked to the drain of the

transistor and a second floating gate transistor linked to a second capacitor can enable a higher output voltage once the capacitors reach full potential (Dixon, 2010).

Three categories of ambient energy harvesting are available, which includes: intentional, anticipated, and unknown as shown in Fig. 6.1.



Fig. 6.1: Pictorial view of intentional, anticipated, and unknown energy sources.

6.3.1 Intentional Energy Harvesting

The design solemnly depends on the active constituents of the system. The constituents are basically the transmitter in the system which main function is to provide the needed energy into the environment, but this is applicable when the energy is actually needed. This work relies basically on the intentional energy harvesting using the Powercast harvester. Powercast support this approach with an energy source of 3W 915MHz RF transmitters, the P1110 and P2110 also use along with it as a receiver. This method of energy harvesting can also be used for other types of energy, which includes placing an energy harvesting device on a piece of industrial equipment which principle of operation is by vibration, that is to say that the energy can be harvested when the system is its full operation. This method of energy harvesting helps designers to engineers' series of stable energy solution. A quick look at the basic operation of the transmitter and receiving circuit is as discussed below.

6.3.2 The Powercast TX91501 Powercaster Transmitter

The Powercast TX91501 is a radio frequency power transmitter specifically designed to provide both power and data to end devices containing the Powercast P2110 or P1110 power harvester receivers (Powercast, 2011). The transmitter is housed in a durable plastic case with mounting holes. It is powered by a regulated 5V DC voltage mostly from a power source of 240V AC, rectified and regulated to its accommodated voltage of 5V DC from its inbuilt internal circuitry. The transmitter has a factory set, fixed power output and no user adjustable settings. Also a beautiful and control feature of it is the status LED which provide a feedback on functional state. It provides a maximum of 3Watts EIRP (Equivalent or Effective Isotropic Radiated Power). A side view and real view of its transmission state are as shown in Fig. 6.2.



(a)



(b)

Fig. 6.2: (a), a side view, and (b) a real view of a TX91501 Powercaster Transmitter in its transmission state.

The Powercast transmitter transmits power in the form of Direct Sequence Spread Spectrum (DSSS) and Data in the Amplitude Shift Keying (ASK) modulation and at a center frequency of 915MHz. The power output is 3watts EIRP and vertically polarizes for optimal transmission. For data communication, it has an 8-bit factory set, TX91501 identification (ID) number broadcast with random intervals up to 10ms using Amplitude Shift Keying (ASK) modulation. Its operating temperature is within the range of -20^{o}C to 50^{o}C at the power input from Mains of 5VDC/1A.

6.3.3 The Powercast Power Harvester Receiver

The Powercast Receivers have the ability to harvest energy directly from ambient RF energy, and the harvested energy is mostly converted to DC power which can be used to charge mobile devices using batteries or battery less devices. The two modules available for our research are P1110 and P2110 and both have similarities and differences in their area of applications as shown in Table 6.3.

Table 6.3: Comparison of the Two RF Energy Powercast receivers

Receivers	Differences	Similarities
P2110	1. Designed for battery charging and direct power applications. 2. Provide intermittent/pulsed power output 3. Provide regulated output voltage up to 5.25V, and configurable voltage 4. Optimization for system power management and control of input/output	1. Harvesting range from 850-950 MHz 2. Works with standard 50-ohm antennas
P1110	1. Mainly design for battery charging applications with also direct power applications 2. Configurable over voltage protection up to 4.2V 3. It can connect directly to rechargeable batteries.	1. Harvesting range from 850-950 MHz 2. Works with standard 50-ohm antennas

The Powercast P2110 power harvester receiver is a radio frequency energy harvesting device, it mainly converts the RF energy into consumable DC voltage. Having a wide range of operating frequencies, it provides radio frequency energy harvesting and power management for battery free, micro power devices. The device helps in the conversion of the radio frequency energy to DC and stores in a capacitor as well as boosting the voltage to the set output voltage level and enables the voltage output.

6.4 Measurement of RF Power Received and Gains

Power meters which provide more accurate measurement of RF power of any of the types of RF measurement equipment, and the simplified Friis Equation that provides a reasonable estimate of the amount of power that is received and available for use were utilized in our experiment.

6.4.1 Friis Transmission Equation

Friis Transmission Formula is mostly used to study RF communication links (Shaw, 2005). The formula can be used in situations where the distance between two antennas is known, and a suitable antenna need to be found. Using the Friis transmission equation, one can solve for the antenna gains needed at either the transmitter or receiver in order to meet certain design specifications.

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi R}\right)^2 \quad (6.1)$$

Where P_r is the received power in Watts (W), P_t is the transmitted power, G_t is the transmitting antenna's gain, G_r is the receiving antenna's gain, λ is the wavelength of the transmitted and received signal in meters, and R is the distance between the antennas in meters. The gain of the antennas, usually measured in decibels, can be converted to power ratio using;

$$G = 10^{\frac{G_{dB}}{10}} \quad (6.2)$$

$$\lambda = \frac{c}{f} \quad (6.3)$$

Where C is the speed of light in meters per second, and f is the frequency in Hz. Hence, C is equal to 3×10^8 m/s

A simplified version of the Friis equation (Powercast, 2010) is provided by the Powercast company for quick and easiest calculation on a spreadsheet, where a reasonable estimate of the amount of power generated, received and available for use are calculated.

6.4.2 Power Density

The power density is the rate of energy transfer per unit volume of a material. The Power density of a hypothetical antenna radiating the same intensity of radio waves in all directions at any distance, is the rate by which power is transmitted by the surface area of a sphere ($4\pi R^2$) of the antenna at that distance. The surface area of the spherical material increases as the radius of the square, it then implies that the power

density, P_D, measured in watts per meter squared, tends to decrease by the square of the radius. That is:

$$P_D = \frac{P_t}{4\pi R^2} \tag{6.4}$$

Where, P_t denotes the peak value of power, P_D represents the power density whereby *R* is the distance between the transmitter and the receiving antenna. The Gain (G_t) of an antenna is calculated using the expression below as:

$$G_t = \frac{\text{Maximum radiation intensity of actual antenna}}{\text{Radiation intensity of isotropic antenna with same power input}} \tag{6.5}$$

The power density at any point from an object detection system of G_t gain is the product of power density from a hypothetical antenna with the object detecting system antenna gain. P_D from an object detection system is therefore:

$$P_D = \frac{P_t G_t}{4\pi R^2} \tag{6.6}$$

6.5 Energy storage

The most common energy storage device used in a sensor node is a battery, either non rechargeable or rechargeable. A non-rechargeable battery (e.g., alkaline) is suitable for a micro sensor with very low power consumption (e.g., 50 μW). Alternatively, a rechargeable battery (e.g., lithium ion) is used widely in sensor nodes with energy harvesting technology (Dusit et al., 2007). A battery is used not only for storage of energy generated by the harvesting device but also to regulate the supply of energy to a sensor node. As wireless sensor node is powered by exhaustible batteries (Waltenegus and Christian, 2010). Several factors affect the quality of these batteries, but the main factor is cost. In a large-scale deployment, the cost of hundreds and thousands of batteries is a serious deployment constraint. Batteries are specified by a rated current capacity, *C*, expressed in ampere-hour. This quantity describes the rate at

which a battery discharges without significantly affecting the prescribed supply voltage (or potential difference). Practically, as the discharge rate increases, the rated capacity decreases. Most portable batteries are rated at 1*C*. This means a 1000mAh battery provides 1000mA for 1 hour, if it is discharged at a rate of 1*C*. Ideally, the same battery can discharge at a rate of 0.5*C*, providing 500mA for 2 hours; and at 2*C*, 2000mA for 30 minutes and so on. 1*C* is often referred to as a 1-hour discharge. Likewise, a 0.5*C* would be a 2-hour and a 0.1*C* a 10-hour discharge. In reality, batteries perform at less than the prescribed rate. Often, the Peukert Equation is applied to quantifying the capacity offset (i.e., how long a battery lasts in reality):

$$T = \frac{C}{I^n} \tag{6.7}$$

I represent the current drawn in Ampere (A), C represents the theoretical capacity of the battery, and it is expressed in ampere-hours; where T is the time of discharge in seconds, and n represent the Peukert number which is a constant that directly relates to the internal resistance of the battery. The value of the Peukert number indicates how well a battery performs under continuous heavy current. If the value approaches unity (1), it shows that the battery performance is good. If the number is high, it shows a high capacity lost whenever the battery is discharged at a higher current. The number is normally determined using empirical means. For example, for lead acid batteries, the number is typically between 1.3, and1.4. Drawing current at a rate greater than the discharge rate results in a current consumption rate higher than the rate at which the active elements in the electrolyte diffuse. If this process continues for a long time, the electrodes run out of active material even though the electrolyte has not yet exhausted its active materials. This situation can be overcome by

intermittently drawing current from the battery and also proper power management techniques.

6.6 Energy Management in WSNs

Despite the fact that energy scavenging mechanisms can be adopted to recharge batteries, e.g., through Powercast harvesters (Powercast, 2011), solar panels (Bouchouicha et al., 2010), piezoelectric or acoustic transducers (Hagerty, et al., 2003), energy are a limited resource and must be used judiciously. Hence, energy utilization efficiency methods need to be looked into at the sensor nodes so as to help in the maximization of network lifetime. Lots of routing algorithms, energy management and data compression algorithms have been designed specifically for WSNs for the purpose of the reduction of energy consumption and management of the sensor networks (Kansal et al., 2007). The EAGRP (Elrahim et al., 2010), An Enhanced AODV (Li et al., 2009), and Pheromone based energy aware directed diffusion algorithm for wireless sensor network (Zhu, 2007), all have developed different protocols in order to manage the available energy in WSNs. In a related work (Alippi et al., 2009); use Energy-hungry Sensors in trying to manage the available energy in WSNs. Reducing power consumption has been a major challenge in WSNs. As a vital factor affecting system cost and lifetime, energy consumption in WSNs is an emerging and active research area. The energy consumption of WSNs is of crucial concern due to the limited availability of energy. Whereas energy is a scarce resource in every wireless device, the problem in WSNs is more severe for the following reasons (Waltenegus and Christian, 2010):

1. Compared to the complexity of the task they carry out; sensing, processing, self-managing, and communication, the nodes being very small in size to accommodate high-capacity power supplies.

2. While the research community is investigating the contribution of renewable energy and self-recharging mechanisms, the size of nodes is still a constraining factor.
3. Ideally, a WSN consists of a large number of nodes. This makes manually changing, replacing or recharging batteries almost impossible.
4. The failure of a few nodes may cause the entire network to fragment premature.

The problem of power consumption can be approached from two angles. One is to develop energy efficient communication protocols (self-organization, medium access, and routing protocols) that take the peculiarities of WSNs into account. The other is to identify activities in the networks that are both wasteful and unnecessary and mitigate their impact. Wasteful and unnecessary activities can be described as local (limited to a node) or global (having a scope network-wide). In either case, these activities can be further considered as an accidental side-effects or results of non-optimal software and hardware implementations (configurations). For example, observations based on field deployment reveal that some nodes exhausted their batteries prematurely because of unexpected overhearing of traffic that caused the communication subsystem to become operational for a longer time than originally intended (Jiang et al., 2007). Similarly, some nodes exhausted their batteries prematurely because they aimlessly attempted to establish links with a network that had become no longer accessible to them. Most inefficient activities are, however, the results of non-optimal configurations in hardware and software components. For example, a considerable amount of energy is wasted by an idle processing or a communication subsystem. A radio that aimlessly senses the media or overhears while neighboring nodes communicate with each other consumes a significant amount of power. A dynamic power management (DPM) control strategy aimed at

adapting the power/performance of a system to its workload. The DPM has a local or global scope or both aims at minimizing power consumption of individual nodes by providing each subsystem with the amount of power that is sufficient to carry out a task at hand (Waltenegus and Christian, 2010). Hence, it does not consider the residual energy of neighboring nodes. IEEABR as the proposed Algorithm consider the available power of nodes and the energy consumption of each path as the reliance of routing selection. It improves memory usage, utilizes the self-organization, self-adaptive and dynamic optimization capability of an ant colony system to find the optimal path and multiple candidate paths from source nodes to sink nodes. The protocol avoids using up the energy of nodes on the optimal path and prolongs the network lifetime while preserving network connectivity. This is necessary since for any WSN protocol design, the important issue is the energy efficiency of the underlying algorithm due to the fact that the network under investigation has strict power requirements. Our proposed algorithm used for the energy management is fully described and presented in Chapter 3.

6.7 Experiment and Simulation Results

Different experiments were conducted to measure the circuit's parameter and influence of the RF power source. Simulation results based on the performance of the circuit with differences in distance of the harvester from the power sources, the energy usage, and energy management using our proposed IEEABR, are all analyzed below while also, showing the harvesting set-up and the simulation environment.

6.7.1 Experimental Results

Using the Powercast Calculator, and components settings; P2110 at 1.2V-915 MHz, Battery Capacity at 1150 mAh for P2110 and P1110 at 4.0V-915 MHz for the same battery capacity, while varying the distance between the

transmitter, the readings are as shown in Table 6.4 (a-d) with differences in the receiver and antenna used in the experiment. The behavior of the packets received with time is shown in Fig 6.3(a-b), while for the packet received with distance for different harvesters and antennas are compared in Fig. 6.4(a-b).

Table 6.4: (a) Amount of Power harvested by P2110 harvester using Dipole Antenna

Distance (ft)	P (μW)	I (μA)	Recharge Time
2	3687	3073	22.08
5	523	436	155.04
10	135	112	602.64
12	85	71	952.32
15	37	31	2169.12
18	11	9	7360.56
20	1	1	68339.28

Table 6.4: (b) Amount of Power harvested by P2110 harvester using Patch Antenna

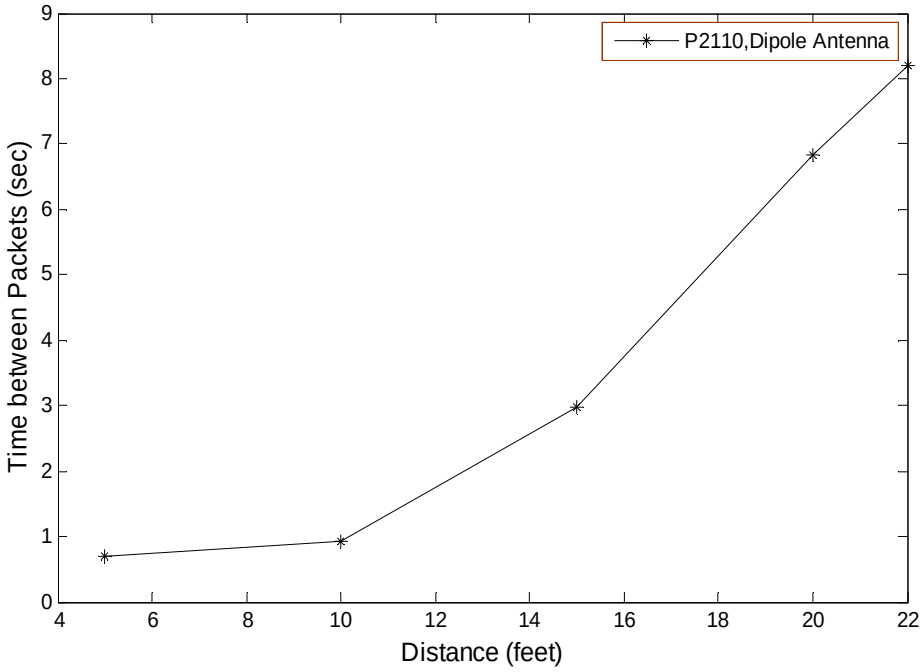
Distance (ft)	P (μW)	I (μA)	Recharge Time (hrs)
5	1925	1604	42.24
10	386	322	210.50
15	189	158	429.40
18	131	109	618.5
20	102	85	797.50
25	50	41	1639.00
30	19	16	4353.00
35	5	4	15517.00
36	1	1	70019.00

Table 6.4: (c) Amount of Power harvested by P1110 harvester using Dipole Antenna

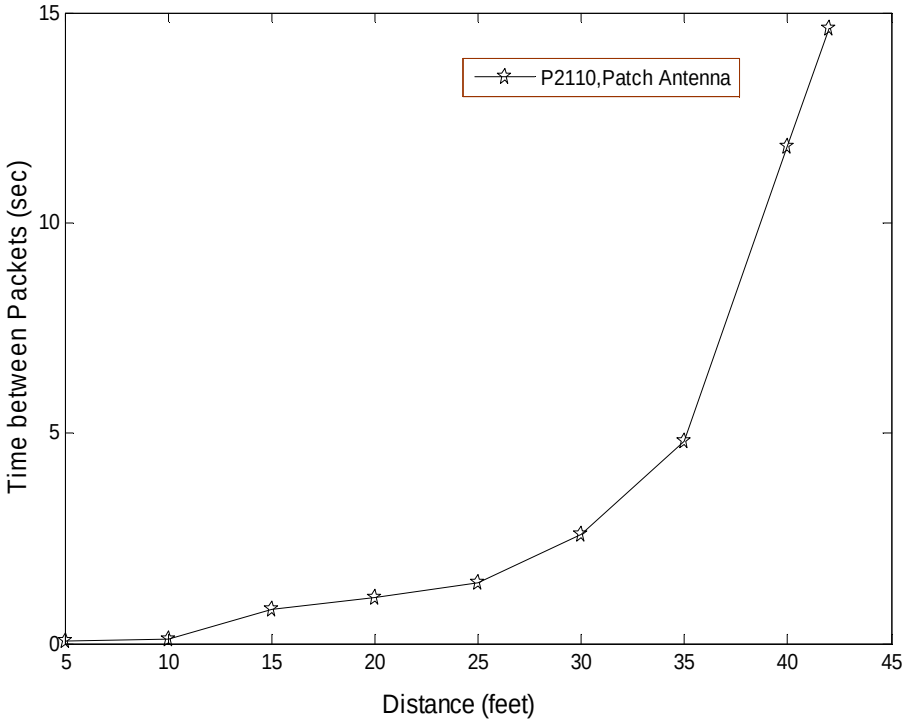
Distance (ft.)	P (μW)	I (μA)	Recharge Time (hrs)
2	3688	922	62.40
4	1085	271	211.92
6	259	65	888.72
7	86	22	2659.92

Table 6.4: (d) Amount of Power harvested by P1110 harvester using Patch Antenna

Distance (ft)	P (μW)	I (μA)	Recharge Time (hrs)
2	16115	4029	14.16
4	3070	768	74.88
6	1551	388	148.30
8	810	203	283.90
10	366	92	627.60
12	93	23	2475.00
13	26	7	8750.00

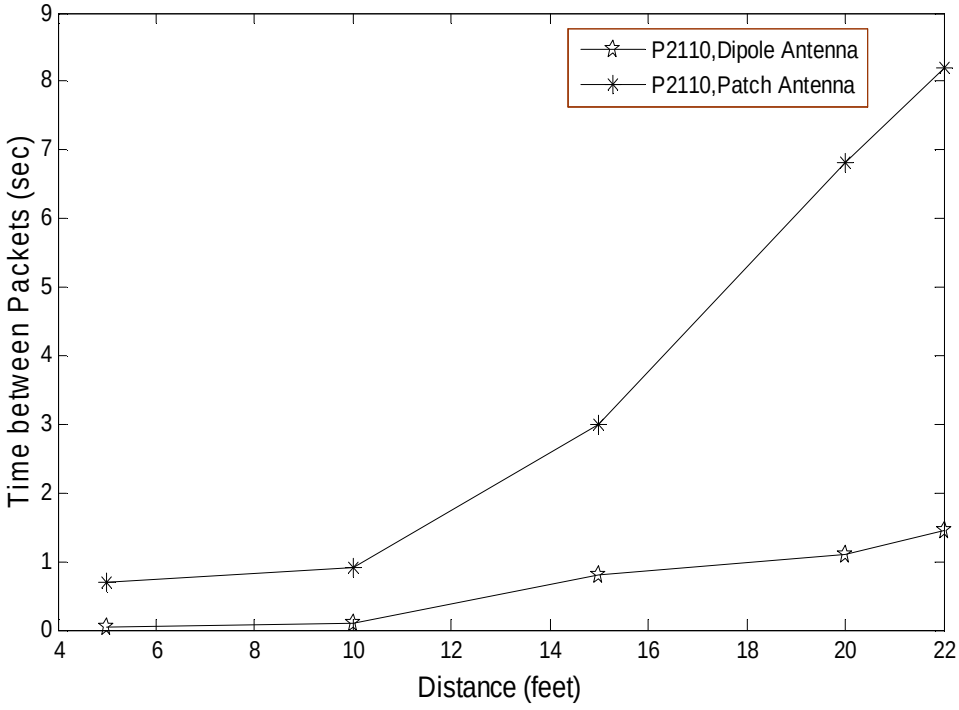


(a) TX91501-3W EIRP,915MHz Power Transmitter

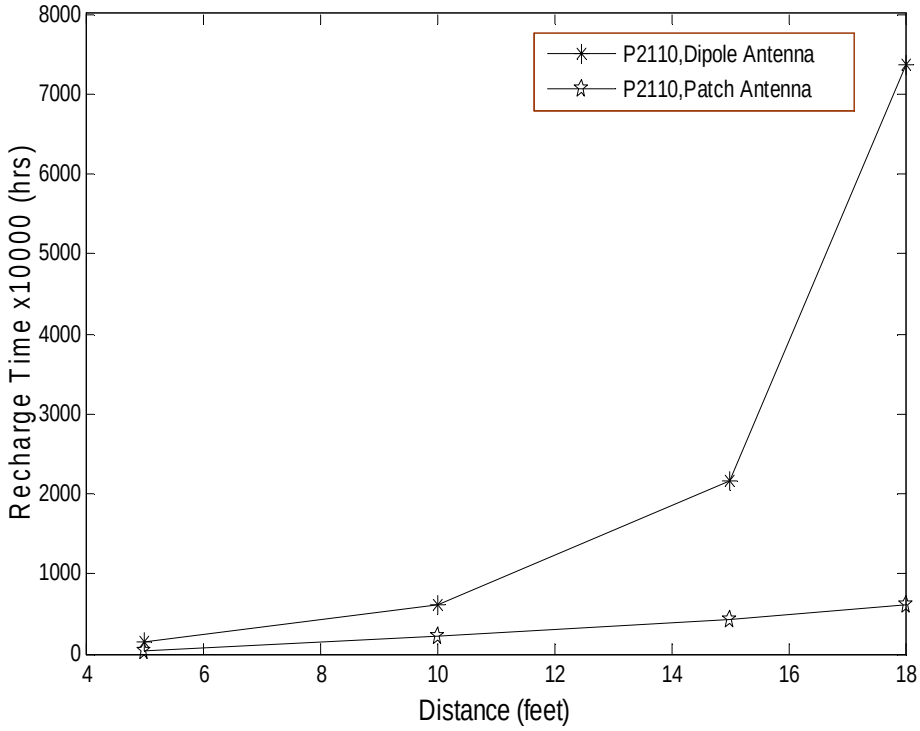


(b) TX91501-3W EIRP,915MHz Power Transmitter

Fig. 6.3: Variation in Time between Packets received (charging time) and Distance of harvesting.



(a) Comparison of packet received with distance using different Antenna



(b) Comparison of Battery Recharge Time, using Dipole and Patch Antennas with P2110

Fig. 6.4: Comparison of Power harvesting using Dipole and Patch Antenna with P2110.

6.7.2 Simulation Results

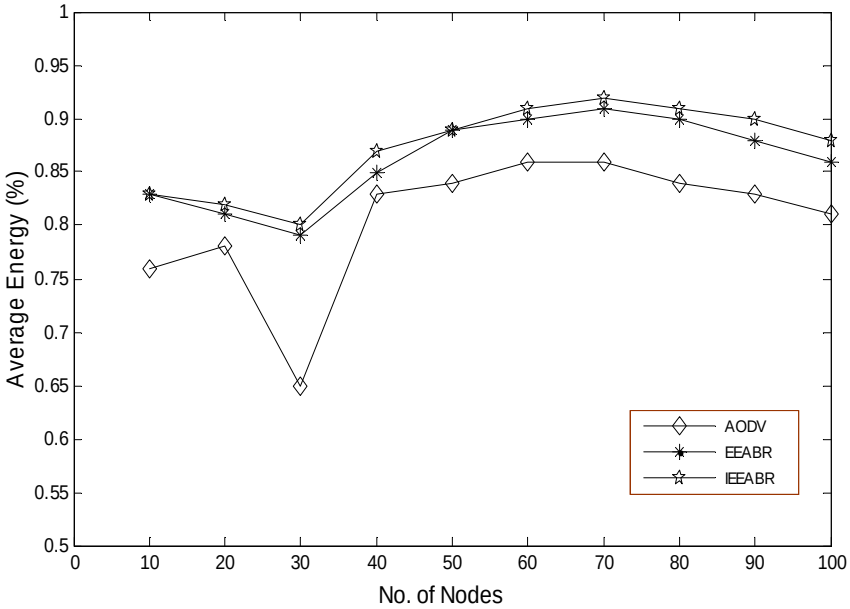
We use an event driven network simulator-2 (NS-2) based on the network topology so as to evaluate the performance of the algorithm used for the energy management. NS-2 provides a high simulation environment for wireless communication along with well-defined propagation, MAC and radio layers. AntSense (an NS-2 module for Ant Colony Optimization) (Camilo, 2007) was used for the EEABR. The simulation parameters are as shown in Table 6.5. We assume that all nodes have no mobility since the nodes are fixed in the application of most wireless sensor networks. Simulations were run for 60 minutes (3600 seconds) for each of the experiments performed, and the remaining energy of all nodes was taken and recorded at the end of each simulation. The average energy calculated while also noting the minimum energy of the nodes. This helps in taking tracks of the performance of the management protocols in term of network's energy consumption.

Table 6.5: Simulation Parameters for AODV, EEABR and IEEABR

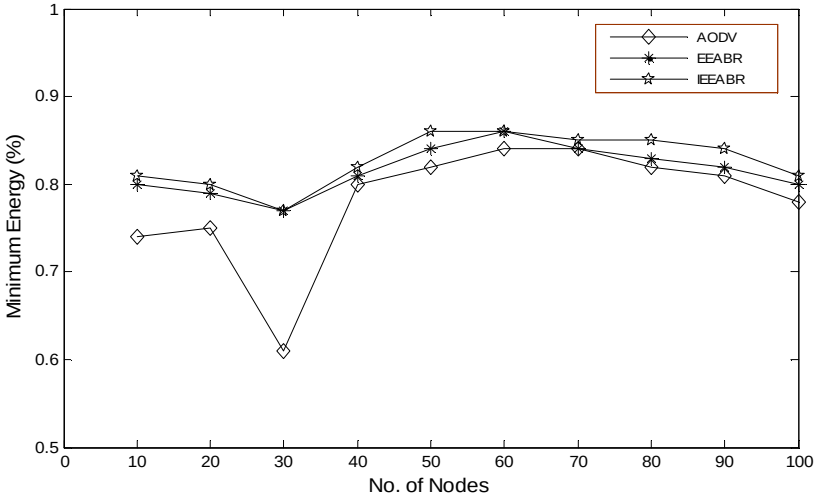
Parameters	Values
Routing Protocols	AODV, EEABR, IEEABR
MAC Layer, Frequency	IEEE 802.15.4, 2.4GHz
Area of Deployment	200x200 m^2 (10 nodes), 300x300 m^2 (20 nodes), 400x400 m^2 (30
Data Traffic, Simulation Time	Constant Bit Rate (CBR), 3600 sec.
Battery power	1150mAH, 3.7V
Propagation Model	Two-ray ground reflection
Data rate	250 Kbps
Current draw in Sleep Mode	62μA
Current draw in Transmitting Mode	50.26mA
Current draw in Receiving Mode	49.56mA
Current draw in Idle Mode	9mA

As Energy is the key parameter to be considered when designing protocol for power management to enhance the maximum lifetime of sensor networks, we therefore use the performance metrics (Section 3.7.1) as that of Chapter 3 for comparison purpose. The simulation was done on a static WSN, and the actual point where the sensed event and the sink node where unknown. Each participating node was given the responsibility of monitoring its immediate environment and the sensed event have to be sent to the sink node in which nodes near the target will depreciate easily in their energy content as they will have to assist in routing the sensed event from other nodes which are far away from the sink node. Simulations were run for 60 minutes (3600 seconds) for each experiment performed, and the remaining energy of all nodes was taken and recorded at the end of each simulation. The average energy calculated while also noting the minimum energy of the nodes. Fig. 6.5 gave the results of the experiments for the performance metrics used for the comparison purpose; The Average Energy, and Minimum Energy of AODV, EEABR and IEEABR. As it can be

seen from the results presented in the Fig. 6.5, the IEEABR protocol had better results in both Average energies of the nodes and the Minimum energy of node experienced at the end of the simulation. The AODV as compared to the EEABR perform worst in all cases. In terms of Average energy levels of the network, IEEABR as compared with EEABR average energy values, varies between 2% and 8%, while for AODV is in the range of 15% to 22% also to the minimum energy of the nodes.



(a) Average Energy of Network Nodes



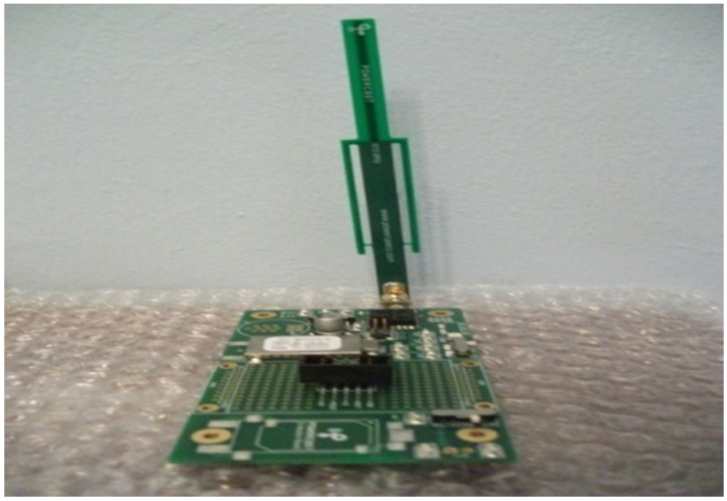
(b) Energy of Node with Minimum Energy

Fig. 6.5: Performance analysis of AODV, EEABR and IEEABR Energy Management Protocols.

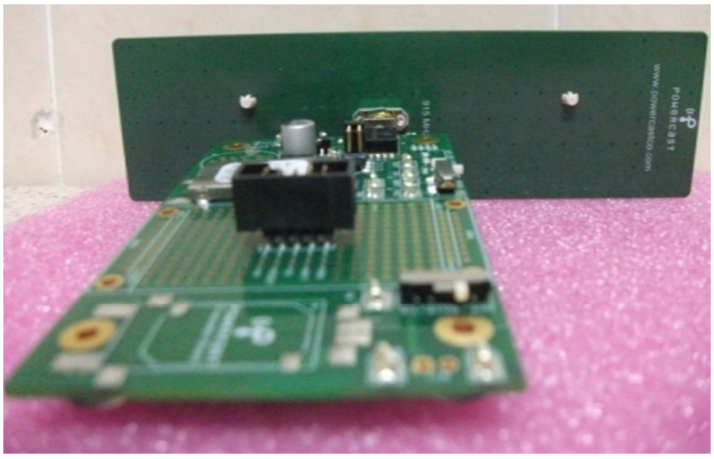
6.7.3 Real-time Implementation of RF Powercast Energy Harvester

The real-time implementation in Harvesting Radio Frequency Energy from Powercast Harvesters and the management of the power harvested using the Improved Energy Efficient Ant Based Routing Algorithm is presented as shown in Fig. 6.6. The experiment was set up and measurement of available power measured while also varying the distance between the harvester and transmitter. The time between each packet delivery that is, the harvesting period was noticed and recorded. The power consumption of the Waspmote under consideration can be found in (Libelium, 2008, 2010). The battery powering the Waspmote is 1150mAh at 3.7V, which can sufficiently power each of the nodes separately under the constant transmission or reception for 19.39 hrs. For our management protocol applied, the maximum energy consumed was found to be 23% of the supply energy amounting to total current drawn to be 264.5mA for 1hr. It then means that, the battery can sufficiently powered the node with the minimum energy for 4.35hours without recharging. For the recharging of the battery at 15 feet as shown in Table 6.4b, it takes 429.4hrs for fully recharge when empty, and 91.9hrs to replenish the drain current of 264.5mA. But with constant harvesting, it then means that the total energy of the battery remains without reduction, which can then sustain the network for the required number of years needed for sensing. A quick look at the receiver in its receiving and conversion state with Dipole, and Patch antenna connected to the application of harvesting from the Powercast transmitter and the harvesting mode of the receiver 3 feet (0.914m) away from the transmitter respectively are shown in Fig 6.6 (a), (b), and (c). P2110 Powercast harvester Receiver with (a) Dipole (omnidirectional) antenna, Fig. 6.6 (b) Patch (directional) antenna, (c) TX95101 Powercaster transmitter in its harvesting mode, (d) with Waspmote, and (e) Gateway

connected to the Sink. The results of the measurements of the harvested RF energy are as presented in Table 6.4 (a-d) and Fig. 6.3 (a-b).



(a) P2110 Powercast harvester Receiver with Dipole (omnidirectional) antenna



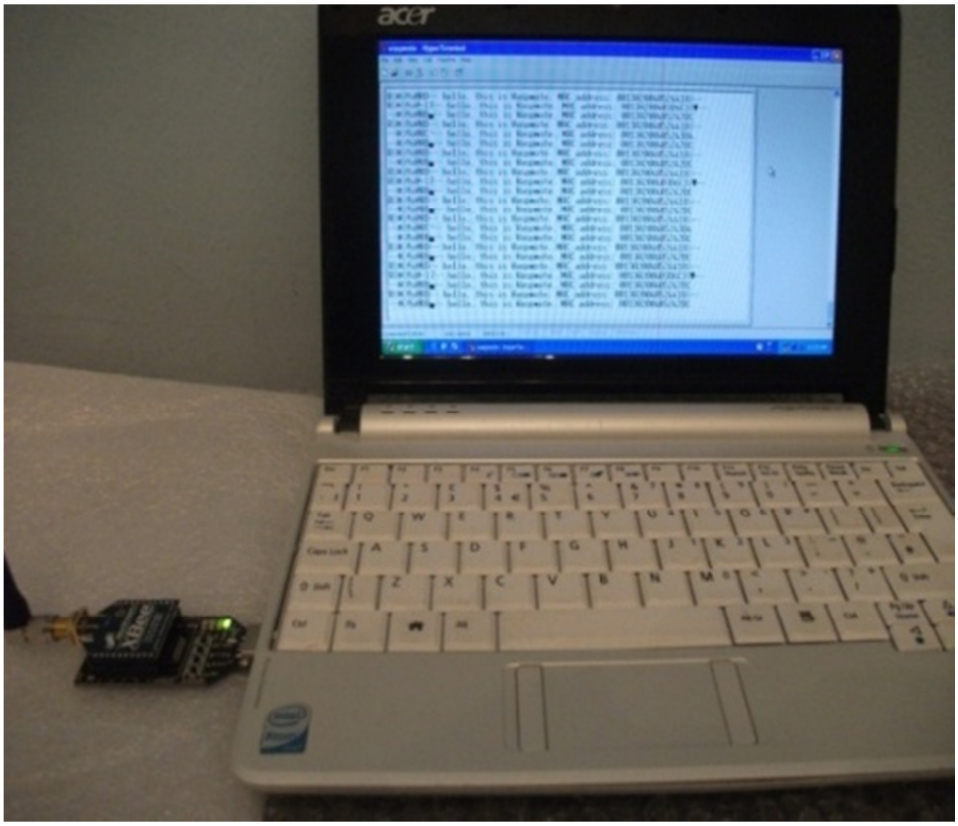
(b) P2110 Powercast harvester Receiver with a patch (directional) antenna



(c) TX95101 Powercaster transmitter in its harvesting mode



(d) TX95101 Powercaster transmitter with Waspmote



(e) Gateway/Sink node connected to the the User PC

Fig. 6.6: Hardware setup of the real-time implementation.

6.8 Conclusions

In this chapter, research based on the application of radio frequency energy harvesting, using Powercast harvesters to support the limited available energy of wireless sensor networks, and its management using Ant Colony Optimization metaheuristic was adopted. In this work we proposed an Improved Energy Efficient Ant Based routing Algorithm energy management technique, which improves the lifetime of sensor networks. From the experimental study, the experimental results clearly showed that IEEABR has a good performance in WSNs applications considering the metrics used, hence better at energy management. Also looking at the harvested energy, the time of charging the battery powering the sensor nodes drastically reduced, while requiring time intervals of 91.9hrs to recharge the battery. The protocol considers the residual energy of nodes in the network after each simulation period. Based on NS-2 simulation, the IEEABR approach has effectively balances the WSN nodes' energy consumption which in turn helps in maximization of the network lifetime. Consequently, IEEABR as the proposed algorithm can efficiently extend the network lifetime without performance degradation. This Algorithm focused mainly on energy management and the lifetime of wireless sensor networks. Furthermore, the chapter concludes with the possibility of wireless sensor network to operate perpetually in the presence of energy harvesting sources and reliable energy management algorithm.

As future work, we intend to build a linking circuit so as to directly charge the Waspmote battery from the Powercast harvesters. Some other protocols based on low network overhead, delay constraint and quality of service may be needed for further studies, and this will then increase the efficiency of the network performance.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This chapter concludes the thesis. The original contributions to the state-of-the-art of the field of swarm intelligence to routing problems in wireless sensor network were presented. In this thesis, we have surveyed the promising state of the art routing protocols which we arrived at the most efficient among the surveyed in terms of energy efficiency and low memory utilization; the EEABR. The first improved version of the EEABR was evaluated through simulation in the same environment in which EEABR was evaluated; NS-2. The improved version shows a high performance as against its predecessor. The improvements to the original EEABR were based on: (1) a new scheme to intelligently initialize the routing tables giving priority to neighboring nodes that simultaneously could be the destination, (2) intelligent update of routing tables in case of a node or link failure, and (3) reducing the flooding ability of ants for congestion control. The energy efficiency improvements are significant particularly for dynamic routing environments.

Further experiments were made and the algorithm simulated in a Matlab based simulation environment; RMASE. The performance shows a high difference between IEEABR algorithm and EEABR in terms of the metrics used for the evaluation. This show and proves beyond reasonable doubt that the proposed algorithm IEEABR perform quite well in all the metrics used for evaluation purpose, while also showing reasonable differences between itself and its predecessor. IEEABR outperforms its predecessor, EEABR with the value of 31% in terms of energy consumption

of the nodes in the networks at static scenario, while 29.66% in dynamic scenario, and energy efficiency of 9% and 64.22% respectively. It was also found to outperform other protocols used for the performance comparison.

To elaborate on the adaptive nature and the energy management ability of the IEEABR, we evaluate its performance in the presence of energy harvesters, and comparison was done with EEABR and AODV algorithms, and found to outperform them in terms of energy efficiency. Its performance in this case for sensor network utilizing the Waspmote power consumption parameters, clearly indicates that it can be used for energy management in WSN which can lead towards the perpetual operation of the network.

After careful study of the improved version of EEABR termed IEEABR, we find it necessary to design a new routing algorithm which counteracts some of the constraints of the IEEABR such as high consumption of energy due to high number of control packets, low throughput and application specific to event-driven (proactive path establishment). Due to the high communication overhead of IEEABR protocol, we then design a new routing algorithm termed Termite-hill which is an on-demand and reactive in path establishment, so as to limit the control packets generated when not in need during the routing process. The Termite-hill routing algorithm is an agent based algorithm which takes an inspiration from the natural system of termites' behaviors due to their adaptability, robustness, and scalability. In this contest, we demonstrated through extensive simulation that the proposed routing algorithm for WSN termed Termite-hill, is simple, scalable, robust, energy efficient, and reliable in information routing in WSN as compared to the existing state-of-the-art routing protocols for WSNs. The Termite-hill was compared with both classical and swarm intelligence based routing protocols, and performance wise was very good. The performance of the Termite-hill was tested in all the three scenarios of

WSN; static, dynamic, and mobility, and it was found to have good performance.

Furthermore, we modeled the environment in which the newly designed protocols were to operate with consideration of the constraints of wireless sensor networks. We developed a formal mathematical framework for modeling and simulating WSN environments utilizing the hill building behavior of termites. The mathematical analysis covers (1) a sound model for WSN topology and information extraction including the grid based and line based topology, (2) the derivation of the expected energy consumption of our agent based routing algorithm for WSN: Termite-hill, as a function of event success rate and occasional change in topology. The results of our mathematical analysis were also compared with the simulation results.

In general, we have engineered an event-driven, simple, scalable, reliable, decentralized and energy-efficient multi-hop routing protocol for WSNs through nature-inspired simple ants and termite agents.

7.2 Future works

In a dynamic environment, no research is really ever completely done. There still exist a number of opportunities for future work. Clearly, there are many areas of improvements to the two (2) algorithms presented in this thesis, i.e. (1) the Improved Energy Efficient Ant-Based Routing algorithm, and (2) Termite-hill routing algorithm. Based on the empirical and formal evaluations of the algorithms, issues including packet delivery rate, energy consumption to end-to-end delay, and everything in-between need to be improved upon. More generally, there are also open issues regarding the development of a mathematical framework for other parameters that involves quality of service aside the energy consumption analysis we have given in this thesis. Future work regarding the real-time implementation of the Termite-hill algorithm on real WSN nodes to verify the assumptions made during the design of the routing protocol is also needed. Though, in a flat network where a pair of source-sink is separated with an arbitrary large number of hops, Termite-hill algorithm fails to discover path, which is an area that need to be addressed. And also, there are a number of natural extensions of modeling of the pheromone update and evaporation of the routing protocols designed in this thesis. Also, in the next generation of ad-hoc networks, the mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs) will cohesively integrate to provide a better performance of the ad hoc network framework. In such a hybrid network, due to conflicting operational environments will definitely presents unique challenges for routing protocols. A recent proposed Termite-hill routing protocol design to operate in the WSN environment can be tuned to operate in the MANET environment, if this happen, more challenges are bound to occur with the environments in terms of energy cost, security of the information to be routed to the sink node, and quality of service factors. As such, we intend to look into the challenging issues.

REFERENCES

Akkaya, K and Younis, M. (2005) A Survey on Routing Protocols for Wireless Sensor Networks. *Ad Hoc Networks (Elsevier)*. **3** (3), pp. 325-349.

Akkaya, K. and Younis, M. (2003) An energy-aware QoS routing protocol for wireless sensor networks. *In: Proceedings of the IEEE workshop on mobile and wireless networks (MWN '03)*. Providence (RI) 2003.

Akyildiz, I.F., Su, W. Sankarasubramaniam, Y. and Cayirci, E. (2002) Wireless sensor networks: A Survey*. Computer Networks*, **38** (4), pp. 393–422.

Alippi, C., Anastasi, G., Francesco, M.D. and Roveri, M. (2009) Energy management in wireless sensor networks with energy-hungry sensors. *Instrumentation and Measurement Magazine.* 12(2), pp. 16-23.

Anandamoy, S. (2006) *Swarm Intelligence Based Optimization of MANET Cluster Formation*. A Thesis submitted to the Faculty of Electrical Engine ering, The University of Arizona. [online]. Available from: http://www.acims.arizona.edu/PUBLICATIONS/PDF/Thesis_Sen.pdf. [Accessed 21 March 2011].

Babulal, K.S., Tewari, R.R. (2010) E2XLRADR (Energy Efficient Cross Layer Routing Algorithm with Dynamic Retransmission for Wireless Sensor Networks). International Journal of Wireless and Mobile Network. **2**(3), pp. 167-177.

Baranidharan, B. and Shanti, B. (2010) A Survey on Energy Efficient Protocols for Wireless Sensor Networks. International Journal of Computer Applications. **11**(10), pp. 35-40.

Beeby, S.P., Tudor, M.J. and White, N.M. (2006) Energy Harvesting Vibration Sources for Microsystems Applications. *Measurements Science and Technology*. **17**, pp. 175-195.

Bi, J., Li, Z. and Wang, R. (2010) An ant colony optimization-based load balancing routing algorithm for wireless multimedia sensor networks. *In: Proceedings of the 12th IEEE international conference on communication technology (ICCT)* 2010. pp. 584–587.

Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999) Swarm Intelligence: From Natural to Artificial Systems. London, U.K.: *Oxford Univ. Press*, pp. 1-278.

Bouchouicha, D., Dupont, F., Latrach, M. and Ventura, L. (2010) Ambient RF Energy Harvesting. *In: International Conference on Renewable Energies and Power Quality (ICREPQ'10)*, Granada, Spain 2010.

Bouchouicha, D., Dupont, F., Latrach, M. and Ventura, L. (2012) Ambient RF energy harvesting. [online]. Available from: http://www.icrepq.com/icrepq%2710/652-Bouchouicha.pdf [Access 04 April 2012].

Boukerche A, Cheng X, Linus J. Energy aware data-centric routing in micro-sensor networks. *In: Proceedings of the ACM MSWiM, in conjunction with ACM MobiCom*. San Diego (CA) 2003. pp. 42–49.

Braginsky, D. and Estrin, D. (2002) Rumor routing algorithm for sensor networks. *In: Proceedings of the first workshop on sensor networks and applications (WSNA)*. Atlanta (GA) 2002.

Buczak, A. and Jamalabad, V. (1998) Self-Organization of a Heterogeneous Sensor Network by Genetic Algorithms. In: C.H. Dagli, C.L. Chen, and M. Akay, M. (Eds.), *Intelligent Engineering Systems through Artificial Neural Networks*. ASME Press, New York, 1998. vol. 8, pp. 259–264.

Bulusu, N., Heidemann, J. and Estrin D. (2000) GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communication Magazine*. **7** (5), pp. 28–34.

Buttyan, L. and Hubaux, J.-P. (2000) Enforcing service availability in mobile ad-hoc WANs. In: *Mobile and Ad Hoc Networking and Computin,* pp. 87-96.

Cai, W., Jin, X., Zhang, Y., Chen, K. and Wang, R. (2006) ACO based QoS routing algorithm for wireless sensor networks*. In: Proceedings of the 3rd international conference on ubiquitous intelligence and computing (UIC).* **4159**, pp. 419–428.

Camilo, T., Carreto, C., Silva, J.S. and Boavida, F. (2006) An energy efficient ant-based routing algorithm for wireless sensor networks. *In: Proceedings of 5th International Workshop on Ant Colony Optimization and Swarm Intelligence*, Brussels, Belgium 2006. pp. 49–59.

Camilo, T., Carreto, C., Silva, J.S., and Bounvida, F. (2007). *AntSense: A N S-2 module for Ant Colony Optimization*. [online]. Available from: http://eden.dei.uc.pt/~tandre/antsense/. [Accessed 05 November 2010].

Çelik, F., Zengin, A. and Tuncel, S. (2010) A Survey on swarm intelligence based routing protocols in wireless sensor networks. *International Journal of the Physical Sciences*. **5** (14), pp. 2118-2126.

Chang, J.-H. and Tassiulas, L. (2000) Maximum lifetime routing in wireless sensor networks. *In: Proceedings of the advanced telecommunications and information distribution research program (ATIRP_00)*. College Park (MD) 2000.

Chen, W., Li, C., Chiang, F. and Chao, H. (2007) Jumping ant routing algorithm for sensor networks. *Computer Communications*. **30**, pp. 2892–2903.

Chong, C.-Y. and Kumar, S.P. (2003) Sensor Networks: Evolution, Opportunities, and Challenges*. In: Proceedings of the IEEE*. **91**(8), pp. 1247 – 1256.

Chu, M., Haussecker, H. and Zhao, F. (2002) Scalable information driven sensor querying and routing for ad hoc heterogeneous sensor networks. *The International Journal of High Performance Computing Applications*. **16**(3), pp. 293–313.

Cobo, L., Quintero, A. and Pierre, S. (2010) Ant based routing for wireless multimedia sensor networks using multiple QoS metrics. *Computer Networks*. **54**, pp. 2991–3010.

Crossbow. (2007). *Imote2*. [Online]. Available from: http://www.cse.wustl.edu/wsn/images/e/e3/Imote2_Datasheet.pdf. [Accessed 8 January 2011].

Crossbow. (2008). *MICAz*. [Online]. Available from: http://courses.ece.ubc.ca/494/files/MICAz_Datasheet.pdf. [Accessed 08 January 2011].

Cui, X., Liu, Z. (2009) BCEE: A balanced-clustering, energy-efficient hierarchical routing protocol in Wireless Sensor Networks. *In: IEEE International Conference on Network Infrastructure and Digital Content*, November 2009. pp. 26-30.

Dasgupta, K., Kalpakis, K. and Namjoshi, P. (2003) An efficient clustering-based heuristic for data gathering and aggregation in sensor networks. *In: Proceedings of the IEEE wireless communications and networking conference (WCNC_03)*. New Orleans (LA) 2003.

De-min, G., Huan-yan, Q., Xiao-yong, Y. and Xiao-nan, W. (2008). *Based on ant colony multicast trees of wireless sensor network routing research.* [online]. Available at: http://www.techrepublic.com/whitepapers/based-on-ant-colony-multicast-trees-of-wireless-sensor-network-routing-research/3785751. [Accessed 03 January 2011].

Dhurandher, S.K., Misra, S., Obaidat, M.S. and Gupta, N. (2008) QDV: a quality-of-security-based distance vector routing protocol for wireless sensor networks using ant colony optimization. *In: Proceedings of 2008 IEEE international conference on wireless and mobile computing, networking and communications 2008*. Pp. 598–602.

Di Caro, G., Ducatelle, F. and Gambardelle, L.M. (2004) AntHocNet: An Ant-based hybrid routing algorithm for mobile ad hoc networks. In: Proceedings of the 8th International Conference on parallel problem solving from nature (PPSN VIII). Pp. 461-470.

Dickinson, R. M. (1975) Evaluation of a microwave high-power reception-conversion array for wireless power transmission. *Jet Propulsion Laboratory, California Institute of Technology*, Pasadena, CA, Tech. Memo pp. 33-741.

Dixon, B. (2010). *Radio Frequency Energy Harvesting*. [Online]. Available f rom: http://rfenergyharvesting.com/. [Accessed 06 February 2011].

Dorigo, M. (2001) Solve difficult optimization problems. Lecture notes in computer science. *In: Proceedings of the 6th European conference on advances in artificial life table of contents*. **2159**, pp. 11–22.

Dorigo, M. and Caro, D. (1998) AntNet: distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*. **9**, pp. 317–365.

Dulman, S.O. (2005) Data-centric architecture for wireless sensor networks. *PhD dissertation*, Romania.

Dusit, N., Ekram, H., Mohammad, M.R. and Vijay, K.B. (2007) Wireless Sensor Networks with Energy Harvesting Technologies: A Game-Theoretic Approach to Optimal Energy Management. *IEEE Wireless Communications*. **14**(4), pp.90-96.

East, T.W.R. (1991) Self-steering, Self-focusing phased array for SHARP. *Antennas and Propagation Society International Symposium,* AP-S, Digest 24-28. **3**, pp. 1732 – 1735.

Elrahim, A.G.A., Elsayed, H.A., Ramly, S.E, and Ibrahim, M.M. (2010) Energy aware wireless sensor network geographic routing protocol. *Universal Journal of Computer Science and Engineering Technology.* **1**(2), pp. 105–11.

Emans, B. (2012). *Additive White Gaussian Noise (AWGN)*. [Online]. Availa ble from: http://en.wikipedia.org/wiki/Additive_white_Gaussian_noise. [Accessed April 2012].

Ghasemaghaei, R., Rahman, M.A., Abdur Rahman, M., Gueaieb, W. and El Saddik, A. (2008) Ant colony-based many-to-one sensory data routing in wireless sensor networks. *In: Proceedings of IEEE/ACS international conference on computer systems and applications 2008*. pp. 1005–1010.

Ghasemaghaei, R., Rahman, M.A., Gueaieb, W. and El-Saddik A. (2007) Ant colony-based reinforcement learning algorithm for routing in wireless sensor networks. *In: Proceedings of IEEE instrumentation and measurement technology conference, IMTC 2007*. pp. 1–6.

Gilbert, J.M. and Balouchi, F. (2008) Comparison of Energy Harvesting Systems for Wireless Sensor Networks. *International Journal of Automation and Computing*. **5**(4), pp. 334-347.

Guo, W., Zhang, W. and Lu, G.A. (2010) Comprehensive routing protocol in wireless sensor network based on ant colony algorithm. *In: Proceedings of NSWCTC'10, Proceedings of the 2010 second international conference on networks security, wireless communications and trusted computing 2010*. **1**, pp. 41-44.

Hagerty, J. A., Zhao, T., Zane, R. and Popovic, Z. (2003) Efficient Broadband RF Energy Harvesting for Wireless Sensors. *In: Colorado Power Electronics Center (COPEC) publications*, Boulder Colorado, USA.

HAMDESIGN. (2010). *RF HAMDESIGN: Microwave equipments and parts*. [online]. Available from: http://www.rfhamdesign.com/. [Accessed 02 January 2010].

Hamida, E.B. and Chelius, G. (2007) Analytical Evaluation of Virtual Infrastructures for Data Dissemination in Wireless Sensor Networks with Mobile Sink. *In: 1st ACM International Workshop on Sensor Actor Networks, Montreal, Canada*. pp. 3-10.

Hawaii. (2003). *Power Density*. [Online]. Available at: http://www.phys.hawaii.edu/~anita/new/papers/militaryHandbook/pwr-dens.pdf. [Accessed 28 February 2011].

Hayami, H., Nakamura, M. and Yoshioka, K. (2005) The Life Cycle CO2 Emission Performance of the DOE/NASA Solar Power Satellite System: A Comparison of Alternative Power Generation Systems in Japan. *IEEE Transactions on Systems, Man, and Cybernetics Part C*: Applications and Reviews. 35(3), pp. 391-400.

He, T., Stankovic, J.A., Lu, C. and Abdelzaher, T. (2003) SPEED: a stateless protocol for real-time communication in sensor networks. *In: Proceedings of the international conference on distributed computing systems*. Providence (RI) 2003.

Hedetniemi, S.M, Hedetniemi, S.T. and Liestman, A.L.A. (1988) Survey of gossiping and broadcasting in communication networks. *Networks*. **18**, pp. 319–49.

Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H. (2000) Energy-efficient communication protocol for wireless microsensor network. *In: Proceedings of the 33rd annual Hawaii international conference on systems science*. **2**, pp. 3005-3014.

Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H. (2002) An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*. **1**(4), pp. 660–670.

Heinzelman, W.R., Kulik, J. and Balakrishnan, H. (1999) Adaptive protocols for information dissemination in wireless sensor networks. *In: Proceedings of ACM MobiCom*. Seattle (WA) August 1999. pp. 174–185.

Hölldobler, B. and Wilson, E.O. (1990) *The Ant*. In: Harvard University Press, 1990.

Hsu, L-H. and Lin, C-H. (2009) Graph Theory and Interconnection Networks. *CRC Press, Taylor and Francis Group*.

Hussain, S. and Islam, O. (2008) Genetic Algorithm for Energy Efficient Trees in Wireless Sensor Networks. *Advanced Intelligent Environments, Springer*. pp. 1-14.

Intanagonwiwat, C., Govindan, R. and Estrin, D. (2000) Directed diffusion: a scalable and robust communication paradigm for sensor networks. *In: Proceedings of ACM MobiCom*. Boston (MA). pp. 56–67.

Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J. and Silva F. (2003) Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking*. **11**, pp. 2–16.

Iyengar, S., Wu, H.-C., Balakrishnan, N. and Chang, S.Y. (2007) Biologically Inspired Cooperative Routing for Wireless Mobile Sensor Networks. *IEEE Systems*. **1**(2), pp. 181-183.

Jabbar, H., Song, Y.S. and Jeong, T.D. (2010) RF Energy Harvesting System and Circuits for Charging of Mobile Devices. *IEEE Transaction on Consumer Electronics*. **56**(1), pp. 247-253.

Jiang, X., Taneja, J., Ortiz, J., Tavakoli, A., Dutta, P., Jeong, J., Culler, D., Levis, P. and Shenker, A. (2007) Architecture for Energy Management in Wireless Sensor Networks*. ACM SIGBED Review-Special issue on the workshop on wireless sensor network architecture.* **4**(3), pp.31-36.

Jietai, W., Jiadong, X.U., Mantian, X. (2009) EAQR: an energy-efficient ACO based QoS routing algorithm in wireless sensor networks. *Chinese Journal of Electronics*. **18**, pp. 113–6.

JLH Labs. (2008). *Background Perspective on Wireless Micro-networking*. [Online]. Available from: http://www.jlhlabs.com/LowPowerWireless.htm. [Accessed 18 January 2011].

Johansson, M., Xiao, L. and Boyd, S. (2003) Simultaneous routing and resource allocation in CDMA wireless data networks. *In: Proceedings of the IEEE International Conference on Communications.* **1**, pp. 51–55.

Johnson, D.B., Maltz, D.A. and Broch, J. (1996) Dynamic source routing in ad hoc wireless networks. *In: Imielinski V, Korth H, (Eds.), Mobile computing, Dordrecht: Kluwer Academic Publishers* 1996. Ch. 5, pp. 153–81.

Kalpakis, K., Dasgupta, K. and Namjoshi, P. (2002) Maximum lifetime data gathering and aggregation in wireless sensor networks. *In: Proceedings of the IEEE international conference on networking (NETWORKS _02).* Atlanta (GA).

Kansal, A., Hsu, J., Zahedi, S. and Srivastava, M.B. (2007) Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems*. **6**, pp. 32-66.

Karaki, J.N. and Kamal, A.E. (2004) Routing Techniques in Wireless Sensor Networks: a Survey. *Wireless Communications, IEEE*. **11** (6), pp. 6-28.

Karp, B. and Kung, H.T. (2000) GPSR: greedy perimeter stateless routing for wireless sensor networks. *In: Proceedings of the 6th annual ACM/IEEE international conference on mobile computing and networking (MobiCom _00)*. Boston (MA).

Katz, R.H., Kahn, J.M. and Pister, K.S.J. (1999) Mobile networking for smart dust. *In: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom99)*, Seattle, Washington, pp. 271-278.

Kazemeyni, F., Johnsen, E.B., Owe, O. and Balasingham, I. (2012) Formal modeling and validation of a power-efficient grouping protocol for WSNs. *Journal of Logic and Algebraic Programming*. **81**(3), pp. 284-297.

Khan, Z. A., Belleudy, C. and Auguin, M. (2009) Analytical Model for Energy Consumption Analysis in Grid Based Wireless Sensor Networks. *In: Proceedings of the International Conference on New Technologies, Mobility and Security.* pp. 1-5.

Kiri, Y., Sugano, M. and Murata, M. (2007) Self-organized data-gathering scheme for multi sink sensor networks inspired by swarm intelligence.

In: Proceedings of first international conference on self-adaptive and self-organizing systems (SASO) 2007. pp. 161–172.

Krishnamachari, B., Estrin, D. and Wicker, S. (2002) Modeling data centric routing in wireless sensor networks. *In: Proceedings of IEEE INFOCOM*, June 2002, New York.

Kulik, J., Heinzelman, W. and Balakrishnan, H. (2002) Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks*. **8**, pp. 169–85.

La, R.J. and Han, Y. (2007) Distribution of path durations in mobile Ad-hoc networks and path selection. *IEEE/ACM Transactions on Networking*. **15**(5), pp. 993-1006.

Le, T., Mayaram, K. and Fiez, T. (2008) Efficient Far-Field Radio Frequency Energy Harvesting for Passively Powered Sensor Networks. *IEE Journal of Solid-State Circuits*. **43**(5), pp. 1287 – 1302.

Li, K., Torres, C.E., Thomas, K., Rossi, L.F. and Shen, C.-C. (2011) Slime mold inspired routing protocols for wireless sensor networks. *Swarm Intelligence*. ***5***(3-4), pp. 183-223.

Li, L. and Halpern, J.Y. (2001) Minimum-energy mobile wireless networks revisited. *In: Proceedings of the IEEE ICC*, Helsinki Finland June 2001. pp. 278–83.

Li, W., Chen, M. and Li, M.-M. (2009) An Enhanced AODV Route Protocol Applying in the Wireless Sensor Networks*. Fuzzy Information and Engineering*. 2, AISC 62, pp. 1591-1600.

Libelium. (2009). *Waspmote 802.15.4 Networking guide*. [Online]. Available at: http://www.libelium.com/documentation/waspmote/waspmote-802.15.4-networking_guide.pdf. [Accessed November 2010].

Libelium. (2008). *Waspmote Technical Guide*. [Online]. Available from: http://www.libelium.com/documentation/waspmote/waspmote-technical_guide_eng.pdf. [Accessed 17 October 2010].

Libelium. (2010). *Waspmote Datasheet.* [online]. Available from: http://www.libelium.com/documentation/waspmote/waspmote-datasheet_eng.pdf. [Accessed November 2010].

Lin, C.R. and Gerla, M. (1997) Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*. **15**(7), pp. 1265–75.

Lindsey, S., Raghavendra, C.S. (2002) PEGASIS: power-efficient gathering in sensor information system. *In: Proceedings of the IEEE aerospace conference*. **3,** pp. 1125–30.

Lindsey, S., Raghavendra, C.S. and Sivalingam, K. (2001) Data gathering in sensor networks using the energy*delay metric. *In: Proceedings of the IPDPS workshop on issues in wireless networks and mobile computing*, San Francisco, CA 2001.

Little, F.E., McSpadden, J.O., Chang, K. and Kaya, N. (1998) Toward Space Solar Power: Wireless Energy Transmission Experiments Past, Present and Future. *In: Space technology and applications international forum, AIP Conference Proceedings*. **420**, pp. 1225-1233.

Liu, M., Xu, S. and Sun, S. (2012) An agent-assisted QoS-based routing algorithm for wireless sensor networks. *Journal of Network and Computer Applications*. **35**(1), pp. 29–36.

Liu, P.X. (2004) Data gathering communication in wireless sensor networks using ant colony optimization. *In: Proceedings of IEEE international conference on robotics and biomimetics 2004*. pp. 822–7.

Liu, Z., Kwiatkowska, M.Z. and Constantinou, C.A. (2005) Biologically Inspired QOS Routing Algorithm for Mobile Ad Hoc Networks. *In: 19th International Conference on Advanced Information Networking and Applications*. pp. 426–431.

Lou, W. (2005) An efficient N-to-1 multipath routing protocol in wireless sensor networks. *In: Proceedings of IEEE MASS*. Washington, DC 2005. pp. 1–8.

Lu, C., Raghunathan, V. and Roy, K. (2010) Micro-scale energy harvesting: A system design perspective. *In: Proceedings of the 2010 Asia and South Pacific Design Automation Conference (ASPDAC'10)*, IEEE Press Piscataway, NJ, USA, 2010, pp. 89-94.

Luo, J., Panchard, J., Piorkowski, M., Grossglauser, M. and Hubaux, J. (2006) MobiRoute: Routing towards a Mobile Sink for Improving Lifetime in Sensor Networks. *In Proceedings Second IEEE/ACM Int'l Conf. Distributed Computing in Sensor Systems (DCOSS) 2006*. pp. 480-497.

MacKenzie, A.B. and Wicker, S.B. (2001) Game theory in communications: motivation, explanation, and application to power control. In: *GLOBECOM'01, IEEE Global Telecommunications Conference*. **2**, pp. 821-826.

Mahadevan, V. and Chiang, F. (2010) iACO: a bio-inspired power efficient routing scheme for sensor networks. *International Journal of Computer Theory and Engineering*. **2**, pp. 972–977.

Manjeshwar, A. and Agrawal, D.P. (2002) APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. *In: the Proceedings of the 2nd international workshop on parallel and distributed computing issues in wireless networks and mobile computing. San Francisco (CA) 2002*. pp. 195–202.

Marta, M. and Cardei, M. (2008) Using Sink Mobility to Increase Wireless Sensor Networks Lifetime. In: *Proceedings Ninth IEEE Int'l Symposium World of Wireless, Mobile and Multimedia Networks (WoWMoM) 2008*. pp. 1-10.

Meghji, M. and Habibi, D. (2011) Transmission power control in multihop wireless sensor network. *In: 2011 third international conference on ubiquitous and future networks*. pp. 25-30.

Min, R., Bhardwaj, M., Cho, S., Shih, E., Sinha, A., Wang, A. and Chandrakasan, A. (2001) Low Power Wireless Sensor Networks. *In: Proceedings of International Conference on VLSI Design, Bangalore, India, 2001*. pp. 221-226.

Misra, R. and Mandal, C. (2006) Ant-aggregation: ant colony algorithm for optimal data aggregation in wireless sensor networks. *In: Proceedings of IEEE international conference on wireless and optical communications networks 2006*. pp. 1–5.

Moreno, R., Antonio, R-G., Aurelio, B. and Rafael, C. (2012) SensGrid: modeling and simulation for wireless sensor grids. *Simulation: Transactions of the society for modeling and simulation international*. pp. 1-16.

Nat, V. and Niculescu, D. (2003) Routing on a curve. *ACM/SIGCOMM Computer Communication Review*. **33**(1), pp. 155–60.

NXP. (2010). *JN5139 Wireless Microcontrollers*. [Online]. Available from: http://www.jennic.com/products/wireless_microcontrollers/jn5139. [Accessed 08 March 2011].

Olugbemi, B.O. (2010) Influence of food on recruitment pattern in the termite, Microcerotermes fuscotibialis. *Journal of insect science.* **10**(154), pp. 1-10.

Panda, M. and Patra, M.R. (2010) Modeling radio channels for CSMA-MAC based wireless sensor networks. International journal of computer applications. **9**(6), pp. 6-11.

Paone, M., Paladina, L., Scarpa, M. and Puliafito, A. (2009) A multi-sink swarm-based routing protocol for wireless sensor networks. *In:*

Proceedings of IEEE symposium on computers and communications 2009. pp. 28–33.

PARC (2006). RMASE: Routing Modeling Application Simulation Environment. Available at: http://webs.cs.berkeley.edu/related.html [Accessed 22 October 2010].

Patra, C., Bhaumik, P. and Chakroborty, M.D. (2010) Modified rumor routing for wireless sensor networks. *Journal of Computer Science*. **7**, pp. 7–10.

Peng, S., Yang, S.X., Gregori, S. and Tian, F. (2008) An adaptive QoS and energy-aware routing algorithm for wireless sensor networks. *Network.* pp. 578–583.

Perkins, C.E. and Royer, E.M. (1999) Ad-hoc on-demand distance vector routing. *In: Proceedings of the 2nd IEEE workshop on mobile computing systems and applications* 1999. pp. 90–100.

Powercast. (2010). *Wireless Power Calculator*. [Online]. Available from: **power**castco.com/**wireless**-**power**-**calculator**.xls [Accessed 08 January 2011].

Powercast. (2011). *Powercast Documentation*. [Online] Available from: http://Powercastco.com/ [Accessed 08 February 2011].

Rabaey, J.M., Ammer, M.J., Silver, J.L.D., Patel, D. and Roundy, S. (2000) PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*. **33** (7), pp. 42–48.

Ramachandran, C., Misra, S. and Obaidat, M.S. (2008) Probabilistic zonal approach for swarm inspired wildfire detection using sensor networks. *International Journal of Communication Systems.* pp. 1047–1073.

Ramin, H., Foad, A.S., Manijeh, K., Reza, A. and Koorush, Z. (2010) Termite Colony Optimization: A novel approach for optimizing continues problems. In: Proceedings of the ICEE 2010. pp. 553-558.

Reinhard, J. and Kaib, M. (2001) Trail communication during foraging and recruitment in the subterranean termite Reticulitermes santonensis De Feytaud (Isoptera, Rhinotermitidae). *Journal of Insect Behavior.* **14**(2), pp. 157-171.

Resnick, M. (1994) Learning About Life. *Artificial Life*. **1**(1-3), pp. 229-242.

Resnick, M. (1997) Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds. *Cambridge MA: MIT Press*.

Rodoplu, V. and Ming, T.H. (1999) Minimum Energy Mobile Wireless Networks. *IEEE Journal of Selected Areas in Communications*. **17**(8), pp. 1333–1344.

Roth, M. (2007) The Markovian Termite: A soft routing framework. *In: Proceedings of IEEE Swarm Intelligence Symposium, SIS 2007.* pp. 213-220.

Roth, M. and Wicker, S. (2003) Termite: ad-hoc networking with stigmergy. In: GLOBECOM '03, *IEEE Global Telecommunications Conference.* **5**, pp. 2937-2941.

Roychowdhury, S. and Patra, C. (2010) Geographic adaptive fidelity and geographic energy aware routing in ad hoc routing. *In: Special issue of IJCCT 2010, for international conference (ACCTA-'10)*. **1**, pp. 3–5.

Sadagopan, N. and Krishnamachari, B. (2004) Maximizing data extraction in energy-limited sensor networks. In: *Proc. IEEE INFOCOM.* **3**, pp. 1717-1727.

Sadagopan, N., Krishnamachari, B. and Helmy, A. (2003) The ACQUIRE mechanism for efficient querying in sensor networks. *In: Proceedings of the first international workshop on sensor network protocol and applications (SNPA'03)* 2003. pp. 149–55.

Saleem, M. and Farooq, M. (2007) A framework for empirical evaluation of nature inspired routing protocols for wireless sensor networks*. In: Proceedings of IEEE Congress on Evolutionary Computation 2007*. pp. 751–758.

Saleem, M., Di Caro, G. and Farooq, M. (2011) Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions. *Information Sciences*. **181**, pp. 4597-4624.

Saleem, M., Khayam, S. and Farooq, M. (2008) Formal modeling of Bee Ad Hoc: a bio-inspired mobile ad hoc network routing protocol. *In: 6th international conference on Ant Colony Optimization and Swarm Intelligence*. pp. 315–322.

Saleem, M., Ullah, I. and Farooq, M. (2012) BeeSensor: An energy-efficient and scalable routing protocol for wireless sensor networks. *Information Science*. 200, pp. 38-56.

Saliba, C., and Farrugia, R.A. (2010) Quality of Service Aware Ant Colony Optimization Routing Algorithm. *Ad Hoc Networks*. pp. 343-347.

Schurgers, C. and Srivastava, M.B. (2001) Energy efficient routing in wireless sensor networks. *In: the MILCOM Proceedings on communications for network-centric operations: creating the information force*, McLean (VA) 2001.

Seah, W. and Tan, Y.K. (2010) Review of Energy Harvesting Technologies for Sustainable WSN. *Sustainable Wireless Sensor Networks,* In Y.K. Tan (Ed.), InTech Publishing. pp. 15-43.

Sendra, S., Lloret, J., Garcia, M. and Toledo, J.F. (2011) Power saving and energy optimization techniques for Wireless Sensor Networks. Journal of Communication. 6(5), pp. 439-459.

Shah, R. and Rabaey, J. (2002) Energy aware outing for low energy ad hoc sensor networks. *In: Proceedings of the IEEE wireless communications and networking conference (WCNC),* Orlando (FL) 2002.

Shaw, J. A. (2005). *Radiometry and the Friis Transmission Equation*. [Online]. Available at: http://www.coe.montana.edu/ee/rwolff/EE548/sring05%20papers/Friis_Radiometric_2005Feb9.pdf. [Accessed 08 January 2011].

Shin, J., Kim, J., Park, K. and Park, D. (2005) Railroad: virtual infrastructure for data dissemination in wireless sensor networks. *In: Proceedings of the 2nd ACM international Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*. pp. 168-174.

Shrivastava, N., Buragohain, C., Agrawal, D. and Suri, S. (2004) Medians and beyond: new aggregation techniques for sensor networks. In: Proceedings of 2nd ACM International Conference on Embedded Networked Sensor Systems (SenSys) 2004.

Singh, S.K. Singh, M.P. and Singh, D.K. (2010) Routing Protocols in Wireless Sensor Networks – A Survey. *International Journal of Computer Science and Engineering Survey (IJCSES)*. **1**(2), pp. 63-83.

Sohrabi, K., Gao, J., Ailawadhi, V. and Pottie, G.J. (2000) Protocols for Self-Organization of a Wireless Sensor Network. *IEEE Personal Communications*. **7** (5), pp. 16–27.

Sohraby, K., Minoli, D. and Znati, T. (2007) *Wireless Sensor Networks: Technology, Protocols, and Applications.* Wiley.

Somasundara, A., Kansal, A., Jea, D., Estrin, D. and Srivastava, M. (2006) Controllably Mobile Infrastructure for Low Energy Embedded Networks. *IEEE Transactions on Mobile Computing*. **5**(8), pp. 958-973.

Struzak, R. (2008). *Basic Antenna Theory*. [Online]. Available from: http://wirelessu.org/uploads/units/2008/08/12/39/5Anten_theor_basics.pdf. [Accessed 04 January 2011].

Subramanian, L. and Katz, R.H. (2000) An architecture for building self configurable systems. *In: Proceedings of the IEEE/ACM workshop on mobile ad hoc networking and computing*. Boston (MA) 2000.

Sun, Y., Ma, H., Liu, L. and Zheng, Y. (2008) ASAR: an ant-based service-aware routing algorithm for multimedia sensor networks. *Frontiers of Electrical and Electronic Engineering in China*. **3**(1), pp. 25–33.

Sztipanovits, J. (2004). *Probabilistic wireless network simulator (Prowler)*. [online]. Available from: http://www.isis.vanderbilt.edu/Projects/nest/prowler/. [Accessed 12 October 2010].

Tang, L. and Guy, C. (2009) Radio frequency energy harvesting in wireless sensor networks. *In: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing (ICWCMC'09): Connecting the World Wirelessly*, New York, NY, USA, 2009. pp. 644-648.

Thanigaivelu, K. and Murugan, K. (2009) Impact of Sink Mobility on Network Performance in Wireless Sensor Networks. In: *First International Conference on Networks and Communications, 2009*. pp. 7-11.

Torres, C. E., Rossi, L. F., Keffer, J., Li, K. and Shen, C. C. (2010) Modeling, analysis and simulation of ant-based network routing protocols. *Swarm Intelligence*. **4**(3), pp. 221–244.

Toumpis, S. and Goldsmith, A.J. (2003) Capacity regions for wireless ad hoc networks. IEEE Transaction on Wireless Communication. **2**, pp. 736–748.

Uchhula, V. and Bhatt, B. (2010) Comparison of different Ant Colony Based Routing Algorithms. *International Journal of Computer Applications, Special Issue on MANETs*. **2**, pp. 97–101.

Villalba, L.J.G., Orozco, A.L.S., Cabrera, A.T. and Abbas, C.J.B. (2009) Routing Protocol in Wireless Sensor Networks. *Sensors.* **9**, pp. 8399-8421.

Waltenegus, D. and Christian, P. (2010) Fundamental of Wireless Sensor Networks: theory and practice. *Wiley Series on Wireless Communication and Mobile Computing*. pp. 207-213.

Wang, G., Wang, Y. and Tao, X. (2009a) An ant colony clustering routing algorithm for wireless sensor networks. *In: Proceedings of third international conference on genetic and evolutionary computing 2009*. pp. 670–73.

Wang, L. and Zhang, R. Model, A.N. (2009b) An energy-balanced ant-based routing protocol for wireless sensor networks. *Networks*. pp. 1–4.

Wang, X., Qiaoliang, L., Naixue, X. and Yi, P. (2008) Ant colony optimization based location-aware routing for wireless sensor networks. *In: Proceedings of the third international conference on wireless algorithms, systems, and applications (WASA'08). Berlin, Heidelberg: Springer-Verlag 2008*. **5258**, pp. 109–120.

Wen, Y.-F., Chen, Y.-Q., Pan, M. (2008) Adaptive ant-based routing in wireless sensor networks using energy*delay metrics. *Journal of Zhejiang University Science A*. **9**, pp.531–8.

White, T. and Pagurek, B. (1998b) Toward Multi-Swarm Problem Solving in Networks. *In: the International Conference on Multi Agent Systems.* pp. 333–340.

White, T., Pagurek, B. and Oppacher, F. (1998a) Connection management using adaptive mobile agents. *In: Proceedings of international conference on parallel distributed processing techniques and applications, CSREA Press*. pp. 802–809.

Xia, S. and Wu, S. (2009) Ant colony-based energy-aware multipath routing algorithm for wireless sensor networks. *In: Proceedings of second international symposium on knowledge acquisition and modeling 2009*. pp. 198–201.

Xiu-li, R., Hong-wei, L. and Yu, W. (2008) Multipath routing based on ant colony system in wireless sensor networks. *In: Proceedings of international conference on computer science and software engineering 2008*. pp. 202–205.

Xu, Y., Heidemann, J. and Estrin, D. (2001) Geography-informed energy conservation for ad-hoc routing. *In: Proceedings of the 7th annual ACM/IEEE international conference on MobiCom'01. Rome, Italy.* pp. 70–84.

Yang, J., Lin, Y., Xiong, W. and Xu, B. (2008) Ant colony-based multi-path routing algorithm for wireless sensor networks. *In: Proceedings of international conference on computer science and software engineering 2008*. pp. 2–5.

Yang, Z. and Mohammed, A. (2005) A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*. **3**, pp. 325-349.

Yao, Y. and Gehrke, J. (2002) The cougar approach to in-network query processing in sensor networks. *SIGMOD record*. New York, USA: ACM 2002.

Younis, M., Youssef, M. and Arisha, K. (2002) Energy-aware routing in cluster-based sensor networks. *In: Proceedings of the 10th IEEE/ACM international symposium on modeling, analysis and simulation of computer and telecommunication systems (MASCOTS'02)*. Fort Worth (TX) 2002.

Younis, O. and Fahmy, S. (2004) Heed: a hybrid, energy-efficient, distributed clustering approach for ad-hoc networks. *IEEE Transactions on Mobile Computing*. **3**(4), pp. 366–9.

Yu, Y., Govindan, R. and Estrin, D. (2001) Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. *Technical reports UCLA/CSD-TR-01-0023. UCLA Computer Science Department*.

Zaman, N. and Abdullah, A.B. (2010) Energy efficient routing in wireless sensor network: Research issues and challenges. *In: IEEE International Conference on Intelligence and Information Technology, United Kingdom, August 2010*. pp. 1-6.

Zamree C.-A., Wajdi, A.-K. and Farhat, A. (2010) The Enhanced Fault-Tolerance Mechanism of AODV Routing Protocol for Wireless Sensor Network. *International Journal of Computer Science and Network Security*. **10**(6), pp. 41-50.

Zhang Y. (2005)
Routing Modeling Application Simulation Environment (RMASE),
Available at: https://docs.google.com/file/d/0B-29IhEITY3bbGY2VVo2SGxxRFE/edit

Zhang, Y. and Fromherz, M. (2004) Message-initiated constraint-based routing for wireless ad-hoc sensor networks. In: *Proceedings of 1st IEEE the Consumer Communication and Networking Conference, 2004*. pp. 648-650.

Zhang, Y., Fromherz, M.P.J. and Kuhn, L.D. (2004) smart routing with learning based QoS-aware Meta-strategies. In: *Proceedings of the 5th International Workshop on Quality of Future Internet Services 2004*. pp. 298-307.

Zhang, Y., Kuhn, L.D. and Fromherz, M.P.J. (2004) Improvements on ant routing for sensor networks. *In: Ant Colony Optimization and Swarm Intelligence, Lecture Notes Computer Science*. **3172**, pp. 289-313.

Zhang, Y., Simon, G. and Balogh, G. (2006) High-Level Sensor Network Simulations for Routing Performance Evaluations. In: *Proceedings of*

3*rd International Conference on Networked Sensing Systems*, Chicago, 31 May-2 June 2006, pp. 1-4.

Zhu, X. (2007) Pheromone based energy aware directed diffusion algorithm for wireless sensor network", *In: D.-S et al. (Eds.), ICIC 2007, LNCS 4681*, pp. 281-291, *Springer-Verlag Berlin Heidelberg,* 2007.

Ziyadi, M., Yasami, K. and Abolhassani, B. (2009) Adaptive clustering for energy efficient wireless sensor networks based on ant colony optimization. In: Proceedings of the seventh annual communication networks and services research conference 2009. pp. 330–334.

Zungeru, A.M., Ang, L.-M. & Seng, K.P. (2012) Classical and swarm intelligence routing protocols for wireless sensor networks: A survey and comparison. *Journal of Network and Computer Applications*, **35** (5), pp.1508-1536.

Zungeru, A.M., Ang, L.-M., Prabaharan, S.R.S. and Seng, K. P. (2011) Radio Frequency Energy Harvesting and Management for Wireless Sensor Networks. *In: Energy Scavenging and Optimization Techniques for Mobile Devices. V. Hrishikesh, and G -M. Mountean (Eds.)* CRC Press, Taylor and Francis Group 2011, USA. 13, pp. 341-367.

Zungeru, A.M., Ang, L.-M., Prabaharan, S.R.S. and Seng, K.P. (2011) Ant Based Routing Protocol for Visual Sensors. *In: A. Abd Manaf et al. (Eds.), ICIEIS Springer-Verlag Berlin Heidelberg 2011.* **252**, pp. 250-264.

Zungeru, A.M., Ang, L-M. and Seng, K. P. (2012) Termite-Hill: Routing towards a mobile sink for improving network lifetime in wireless sensor networks. *In: 2012 Third International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*. pp. 622-627.